User's Manual

32

RH850/U2A-EVA Group

User's Manual: Hardware

Renesas microcontroller

RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.50 Aug. 2024

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.
5. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
6. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
7. Power ON/OFF sequence
   In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual. The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The following documents apply to the RH850/U2A-EVA Group. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

Document Type	Description	Document Title	Document No.
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description Note: Refer to the application notes for details on using peripheral functions.	RH850/U2A-EVA Group User's Manual: Hardware	This User's manual
User's manual for Hardware	Hardware specifications of flash memory programming and erasure	RH850/U2A-EVA Group Flash Memory User's Manual: Hardware	R01UH0832EJxxxx
User's manual for Hardware	Hardware specifications of intelligent cryptographic unit/master and Secure Watchdog Timer	RH850/U2A-EVA Group Security User's Manual: Hardware	R01UH0868EJxxxx
User's manual for Hardware	Hardware specifications of virtualization support function	RH850G4MH Virtualization User's Manual: Hardware	R01UH0865EJxxxx
User's manual for Software	Description of CPU instruction set	RH850G4MH User's Manual: Software	R01US0209EJxxxx
User's manual for Software	Description of CPU instruction set of virtualization function	RH850G4MH Virtualization User's Manual: Software	R01US0432EJxxxx
User's manual for Emulation	Description of emulation functions	RH850/U2A-EVA Group User's Manual: Emulation	R01UH0834EJxxxx
User's manual for Emulation	Description of emulation functions in hardware security module	RH850/U2A-EVA Group Security User's Manual: Emulation	R01UH0835EJxxxx
Application Note	Information on using peripheral functions and application examples Sample programs Information on writing programs in assembly language and C	Available from Renesas Electronics Web site.	
Renesas Technical Update	Product specifications, updates on documents, etc.		

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
Note:	Footnote for item marked with Note in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or $xxxx_B$
	Decimal ... xxxx
	Hexadecimal ... $xxxx_H$

Prefix indicating power of 2 (address space, memory capacity):

K (kilo): $2^{10} = 1,024$

M (mega): $2^{20} = 1,024^2$

G (giga): $2^{30} = 1,024^3$

Description of Registers

Each register description includes register access, register address, and register value after a reset, a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables are described below.

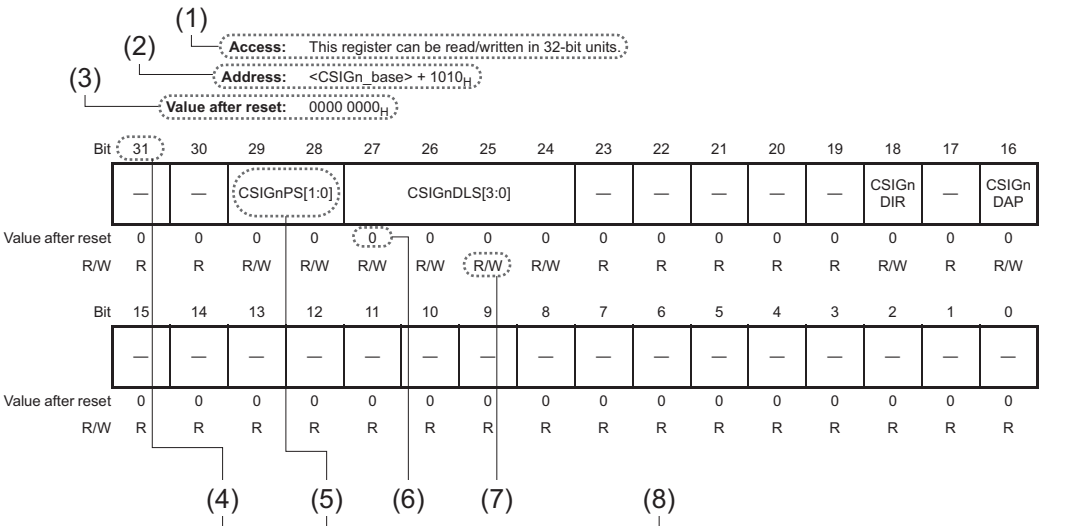


Table 14.19 CSIGnCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.
27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** Do not set bits CSIGnCFG0.CSIGnDLS[3:0] for a value 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.
23 to 19	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	No parity is waited for.
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.
1	0	Add odd parity	Odd parity bit is waited for.
1	1	Add even parity	Even parity bit is waited for.

(1) Access

The register can be accessed in the bit unit indicated here.

(2) Address

This is the register address.
For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.
The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.
When clearly identifying the digits of a bit field is required, do so by using a form such as CSIGnDLS[3:0] above.
Indicate reserved bits by using a dash (—).

(6) Value after a reset (in binary notation)

This is the bit values after a reset.

0	:	The value after a reset is 0.
1	:	The value after a reset is 1.
—	:	The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W	:	The bit or field is readable and writable.
R	:	The bit or field is readable. Note that all reserved bits are indicated as R. When written, the value specified in the bit chart or the value after a reset should be written. In case of writing to writable registers that also include non-reserved bits with the R-attribute, writing to the R-attribute bits will be ignored unless otherwise specified.
W	:	This bit or field is writable. When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

Table of Contents

Section 1 Overview

The RH850/U2A-EVA is an emulation device of RH850/U2A16 (516 pins), RH850/U2A16 (373 pins), RH850/U2A16 (292 pins), RH850/U2A8 (373 pins), RH850/U2A8 (292 pins), RH850/U2A6 (292 pins), RH850/U2A6 (176 pins), RH850/U2A6 (156 pins) and RH850/U2A6 (144 pins) production devices which are the single-chip microcontroller RH850 series from Renesas Electronics.

This section describes an overview of the RH850/U2A-EVA (516 pins), RH850/U2A16 (516 pins), RH850/U2A16 (373 pins), RH850/U2A16 (292 pins), RH850/U2A8 (373 pins), RH850/U2A8 (292 pins), RH850/U2A6 (292 pins), RH850/U2A6 (176 pins), RH850/U2A6 (156 pins) and RH850/U2A6 (144 pins).

1.1 Outline

This product is a 32-bit single-chip microcontroller that incorporates multiple CPUs of the RH850 Series, code flash, data flash, RAM, DMA controllers, high-speed communication interfaces including CAN, FlexRay, Ethernet, RHSIF, LIN, SENT, PSI5S, PSI5, peripheral functions including A/D converters, timer units and many communication interfaces that are used in the automotive applications. This product also conforms to the Automotive Safety Integrity Level (ASIL) that is highly demanded in the recent automotive field (ASIL D level).

(1) Includes multiple RH850 cores

This product contains multi RH850G4MH2 cores (each CPU is referred to as CPU0 to CPU3 (U2A8, U2A6: CPU0 to CPU1) hereafter). Each CPU supports RISC-type instruction sets and have significantly improved the instruction execution speed with basic instructions (one clock cycle per instruction) and the optimized 10-stage pipeline configurations. Furthermore, this product also supports multiplication instructions using a 32-bit hardware multiplier, saturated product-sum operation instructions, and bit manipulation instructions as instructions best suited for various fields. In addition, this product also support CPU virtualization function.

Two-byte basic instructions and high-level language instructions improve object code efficiency for the C compiler and reduce the program size. Furthermore, this product is suited for advanced real-time control applications by offering a high-speed response time including the processing time of the on-chip interrupt controller.

(2) On-chip code flash and data flash

This product incorporates a 16-MB (U2A8: 8MB, U2A6: 6MB) code flash allowing high-speed accesses, which enables each CPU to access this flash memory efficiently. This memory can be reprogrammed while it is placed on an application system. This can shorten the system development period and significantly improve the serviceability after the system is delivered.

This product also has a 576-KB (U2A8: 320-KB, U2A6: 256-KB) data flash that is available for storing EEPROM data.

(3) Rich peripheral functionality

This microcontroller supports common communication interfaces such as SPI (CSI), I^2C as well as automotive oriented communication interfaces such as Ethernet, RHSIF, FlexRay, CAN-FD, LIN, CXPI, SENT, PSI5, PSI5S. As internal peripheral modules, this microcontroller have A/D Converter, Long-Term System Counter, Generic Timer Module, timer units and dedicated Peripheral Interconnect module which connects the functionalities of these peripherals. It also has the global standard on-chip Nexus JTAG as a debug interface. This allows construction of systems without the need to provide these functions externally, which reduces cost, quantity of components, and PCB footprint.

(4) Functional safety support

This product equips several dedicated functionalities including the Dual Core Lock Step configuration for the CPU, the memory protection with ECC, the bus protection with ECC/EDC, the peripheral module protection, and voltage / clock monitors to support the functional safety standard (ISO26262) required in the automotive applications.

(5) Security support

This product provides various security features and utilizes the Intelligent Cryptographic Unit/Master (ICUMHA) as an on-chip Hardware Security Module (HSM).

1.2 Features

CPU0/1/2/3 core	RH850G4MH2:	4 units [For U2A-EVA and U2A16 Only] 2 units [For U2A8 and U2A6 Only] (for high-speed operation and control)
CPU0/1/2/3 instruction cache memory (for Code Flash)	16 KB	
CPU0/1/2/3 data buffer (for Code Flash)	4 lines (256 bit/line)	
Minimum CPU0/1/2/3 instruction execution time	2.5ns (during internal 400 MHz operation)	
General CPU0/1/2/3 registers	Thirty-two 32-bit registers	
CPU0/1/2/3 instruction sets	Signed multiplication (32 bits × 32 bits → 64 bits): 1 to 2 CPU clocks Saturated operation instructions (with overflow/underflow detection function) 32-bit arithmetic/logical shift instructions: 1 CPU clock Bit manipulation instructions Load/store instructions with long/short formats Signed load instructions	
Inter-Processor Interrupt (IPIR)	Support of inter-Processor interrupt function of 4 channels Support level detection of interrupts Accessible from all clusters and all PEs identification of interrupt request source PE is possible. Unintended inter-PE interrupts can be prevented by masking interrupt requests. SET1, CLR1, and NOT1 can be executed as atomic operation instructions to IPIR.	
Barrier-Synchronization (BARR)	The 16-ch barrier synchronization registers are provided. Barrier synchronization can be implemented using the same code for all the cores. Accessible from all clusters and all PEs within the system	
Time Protection Timer (TPTM)	Interval timer × 2ch (down counter), free-run timer × 1ch (up counter), up timer × 2ch (up counter) Start, Stop and Restart of counter for the interval timers, the free-run timer and the up timers. Divided counter of the timers can be configured for the interval timers, the free-run timer, up timer respectively. Simultaneous count control for the interval timers belonging to the same PE. Simultaneous count control for the up timers belonging to the same PE using registers prepared for each PE. Globally simultaneous count control registers are prepared. A PE can control all up counters including ones owned by other PEs. Underflow interrupt for the interval timers. Comparison value matching interrupts for the up timers. Each timer set has 3 control signals to stop timers while in debug mode. One is for interval timers and the free-run timer, another is for up timer 0, the other is for up timer 1.	
Memory space	4-GB address space (common to program and data)	
Code flash	• Two types of memory area – User area: 16 MB (common to each CPU, ICUMHA) [For U2A-EVA and U2A16 Only] – User area: 8 MB (common to each CPU, ICUMHA) [For U2A8 Only] – User area: 6 MB (common to each CPU, ICUMHA) [For U2A6 Only] – User boot area: 64 KB high-speed reading through cache enabled. • OTA (Over-the-Air) update support	
Data flash	512 KB + 64 KB (dedicated to ICUMHA) [For U2A-EVA and U2A16 Only] 256 KB + 64 KB (dedicated to ICUMHA) [For U2A8 Only] 192 KB + 64 KB (dedicated to ICUMHA) [For U2A6 Only]	
RAM	Local RAM: 64 KB (CPU0/1/2/3 [For U2A-EVA and U2A16 Only]) Local RAM: 64 KB (CPU0/1 [For U2A8 and U2A6 Only]) Cluster RAM: 3328 KB (common to each CPU) (256 KB: Retention RAM) [For U2A-EVA and U2A16 Only] Cluster RAM: 1664 KB (common to each CPU) (128 KB: Retention RAM) [For U2A8 Only] Cluster RAM: 640 KB (common to each CPU) (128 KB: Retention RAM) [For U2A6 Only] Cluster Emulation RAM: 2 MB (1 MB for each cluster 0 & 1) [For U2A-EVA Only] Cluster Emulation RAM: 32 KB (cluster 0) [For U2A6 Only] Global Emulation RAM: 2 MB [For U2A-EVA Only]	

Serial Flash Memory I/F (SFMA)	1 unit incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292) and U2A6 (QFP-176) Only] One serial Flash Memory device can be connected. A data bus width of 1 bit, 2 bits, or 4 bits can be selected. 4 GB address space Efficient data reception due to built-in read cache (64-bit line × 16 entries) Arbitrary bit rate settable by the on-chip baud rate generator
Multimedia Card Interface (eMMC)	1 unit incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292) and U2A6 (QFP-176) Only] • Compliant with JEDEC STANDARD JESD84-A441 (neither DDR mode nor 1.8-V operation is supported). • Supports 1-/4-/8-bit MMC bus widths. • Supports the backward-compatible mode. • High-speed mode is supported. • MMC Clock frequency = MMCA module clock frequency/2k (k = 1 to 10). • Supports block transfer. • Supports boot operation. • Supports high priority interrupts (HPI). • Supports background operation. • Interrupt requests: normal operation and error/timeout. • DMA transfer requests: buffer write and buffer read.
Interrupts/exceptions	1 non-maskable interrupt (NMI pin) 1 FE level interrupt 768 maskable interrupts (high-speed: 32, low-speed: 736) Simultaneous distribution of interrupt sources to multiple cores (each CPU) • Applicable sources: non-maskable interrupt (NMI pin), FE level interrupt, 32 high-speed maskable interrupts External interrupt input function (IRQ pins) Software interrupt function (SINT) Inter-processor interrupt function (IPIR) 64-level priority specifiable for maskable interrupts For RH850G4MH2 exceptions, see **Section 3.2.4, Exceptions and Interrupts**.
sDMA controller	32 channels incorporated (16 channels × 2 units) Transfer data length: 1 byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, 32 bytes, 64 bytes Parallel reads and writes (fly-by) Address mode: dual address mode Transfer requests: auto request, peripheral hardware request. Bus modes: normal speed mode, slow speed mode Arbitration modes: fixed priority mode, round-robin mode Interrupt requests: termination of descriptor step, the termination of data transfer, the occurrence of address error. Descriptor memory: 8 KB (shared at all channels). Scatter-gather transfer Transfer target: On-chip memory, on-chip peripheral modules (excluding DTS and sDMAC)
DTS controller	128 channels incorporated Transfer unit: 8 bits/16 bits/32 bits/64 bits/128 bits 64-bit × 2-burst transfer Dual-address transfer mode Address reloading function Chain transfer function Three transfer modes: Single transfer, block transfer 1 (specified by number of transfer times), and block transfer 2 (specified by address count) Transfer target: On-chip memory, on-chip peripheral modules (excluding the DTS and sDMAC) Transfer requests can be set by interrupt sources and the software.
I/O	Output driving ability of specific input/output pins is selectable Inversion or non-inversion of output values of specific input/output pins is selectable Pull-up or pull-down off of specific input/output pins is selectable

Safety functions	Flash memory ECC error detection function RAM ECC error detection function Peripheral module RAM ECC error detection function (e.g. FlexRay, CAN, GTM) Clock monitor Error Control Module (ECM) Duplexing of modules (e.g. CPUs, ECM, error output pins) Automatic Power-on BIST execution after reset Standby Resume BIST (SR-BIST) execution selection after wake-up from DeepSTOP mode.
Error Control Module (ECM)	Collects information for each error check system and safety function and indicates error status. When an error is detected, an error signal can be output from the error pin to the external. Interrupts and internal reset signals can be generated upon detection of an error. Provided with a function to generate a pseudo-error for debugging and self-diagnosis.
Data CRC Function (KCRC)	The data CRC (Cyclic Redundancy Check) function can verify or generate data streams protected by a CRC with various lengths and different bit widths.
Window Watchdog Timer (WDTB)	5 units incorporated [For U2A-EVA and U2A16 Only] 3 units incorporated [For U2A8 and U2A6 Only] Can generate a signal to the ECM when a counter overflows (timer expires). Can generate an interrupt at 75% of the counter overflow value. An interrupt request can be generated at any function of the counter value. A window open period can be set to any function of the counter value.
Long-Term System Counter (LTSC)	1 unit incorporated 64-bit counter without overflow. • Free-run up counting • Atomic read/write access to all registers • Anytime read access to counter registers • Application reset (SW reset) can be masked. When masked, counter keeps running on reset occurrence and counter register will not be initialized.
OS Timer (OSTM)	10 units incorporated [For U2A-EVA and U2A16 Only] 8 units incorporated [For U2A8 and U2A6 Only] • A 32-bit timer assuming use by OS • Interval timer mode or free-running timer mode selectable • Synchronous start between units available
Timer Array Unit D (TAUD)	3 units incorporated The TAUD has the following functions: • 16 channels, 16-bit counter and 16-bit data register per channel • Independent channel operation • Synchronous channel operation (master and slave operation) • Generation of different types of output signal • Real-time output • Counter can be triggered by external signal • Interrupt generation The TAUD can operate independently or synchronously (combine with other channels)
Timer Array Unit J (TAUJ)	4 units incorporated The TAUJ has the following functions: • Independent channel operation function (operated using a single channel) • Synchronous channel operation function (operated using a master channel and multiple slave channels) The TAUJ can operate independently or synchronously (combine with other channels)

Motor Control Timer (TSG3)	2 units incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292) and U2A6 (QFP-176) Only] 1 unit incorporated [For U2A6 (BGA-156) and U2A6 (QFP-144) Only] • 18-bit timer counter • Count clock resolution: Minimum 12.5 ns (count clock = 80 MHz) • Operating mode corresponding to various motor control methods • Compare registers with reload buffer • 10-bit dead time counter • A/D conversion trigger signal generation • Forced output stop function by TAPA • Reload (simultaneous rewrite) or anytime rewrite • HT-PWM mode with 0-100% duty cycles output • Semi-automatic cruise function • Three-phase encoder function (hall sensor signals can be input). • Fail-safe function (warning interrupt or error interrupt can be generated) – Simultaneous active output detect function for positive and inverse phase. – Abnormal input detection function of the three-phase encoder
Timer Option (TAPA)	4 units incorporated [For U2A-EVA, U2A16, U2A8 and U2A6 (BGA-292) Only] 3 units incorporated [For U2A6 (QFP-176) Only] 2 units incorporated [For U2A6 (BGA-156) and U2A6 (QFP-144) Only] Combine with the peripheral interconnect (PIC) to provide the following functions: • Asynchronous Hi-Z control function • Interrupt signal output function • A/D conversion start trigger selection function
Timer Pattern Buffer (TPBA)	2 units incorporated • Count clock resolution: Minimum 12.5 ns (count clock = 80 MHz) • 16-bit counter • 16-bit duty register • 16-bit period setting register • 7-bit address counter register • 7-bit pattern number setting register • Interrupt request signals – Period-matched detection interrupt – Duty-cycle-matched detection interrupt – Number-of-patterns matched detection interrupt • Number of duty patterns – 64 patterns (16 bits) or 128 patterns (8 bits) • Automatic duty generation according to the number of patterns • Output control by software • The count clock can be selected from PCLK, PCLK/2, PCLK/4, and PCLK/8 according to the prescaler set value. • Synchronous start with another timer
PWM Output/Diagnostic (PWM-Diag)	1 PWBA block for generating clock signals. • Generates a count clock signal for PWGC 96 PWGC blocks generate PWM signals. [For U2A-EVA, U2A16, U2A8 and U2A6 (BGA-292) Only] 76 PWGC blocks generate PWM signals. [For U2A6 (QFP-176) Only] 50 PWGC blocks generate PWM signals. [For U2A6 (BGA-156) Only] 64 PWGC blocks generate PWM signals. [For U2A6 (QFP-144) Only] • Outputs PWM waveforms and A/D conversion trigger to PWSD 1 PWSD block for generating triggers for A/D conversion. • Transmits the required setting information to the A/D converter and outputs the A/D conversion start trigger

Real-Time Clock (RTCA)	1 unit incorporated Count clock selection from 240 kHz to 2.5 MHz Counters for years, months, day of the month, day of the week, hours, minutes, seconds, and a subcounter. One Hz pulse output function Fixed interval interrupt function Alarm interrupt function
Peripheral Interconnect function (PIC1)	1 unit incorporated • Simultaneous start trigger function • INT signal output selection function • PWM/Delay pulse output function with dead time • Trigger pulse width measurement function • Encoder capture trigger select function • Two-phase encoder control function • Three-phase pulse input control function • Three-phase encoder control function • TAUD input select function • Hi-Z control function • Timer output monitor function (PWM-Diag) • Timer input monitor function • TSG3 Synchronous Clear Function
Peripheral Interconnect function (PIC2)	3 units incorporated • ADCJ trigger select function • Signal routing function for GTM: – Baud Rate Measurement for an UART (RLIN3) – Hi-Z Control Function Over External Pin for GTM Output – GTM Output Monitor for PWM Diagnostic – ENCA Trigger Selection Function – PSI5S Timestamp and the Sync Pulse Signal Selection Function – GTM Timer Input (TIM) Selection Function – ENCA Encoder Input Selection
Serial Communication Interface 3 (SCI3)	3 units incorporated [For U2A-EVA, U2A16 and U2A8 Only] • Clock synchronization or start-stop system selectable • Full-duplex communication enabled • Arbitrary bit rate selectable by the on-chip baud rate generator • LSB first or MSB first selectable
Multi-channel Serial Peripheral Interface (MSPI)	10 units incorporated [For U2A-EVA (BGA-516) and U2A16 (BGA-516) Only] 9 units incorporated [For U2A16 (BGA-373) and U2A8 (BGA-373) Only] 6 units incorporated [For U2A16 (BGA-292), U2A8 (BGA-292), U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (BGA-156) Only] 4 units incorporated [For U2A6 (QFP-144) Only] • Chip select: Up to 8 for each unit • Three-wire serial synchronous data transfer • Master mode or slave mode selectable • Settable up to eight channels with phase of clock and data settable • Transmission rate up to 40 Mbps • Arbitrary bit rate settable by the on-chip baud rate generator

CANFD interface (RS-CANFD)	16 channels incorporated [For U2A-EVA, U2A16 and U2A8 Only] 12 channels incorporated [For U2A6 (BGA-292) Only] 11 channels incorporated [For U2A6 (QFP-176) Only] 8 channels incorporated [For U2A6 (BGA-156) Only] 7 channels incorporated [For U2A6 (QFP-144) Only] • Classical CAN mode (RS-CAN software compatibility mode) – Conforming to CAN-FD ISO 11898-1 (2015) – Transfer speed up to 1 Mbps – A total of 16588 message buffers (Individual 1024 buffers + Shared 15564 buffers when 8 byte data payload) provided for 16 channels [For U2A-EVA, U2A16 and U2A8 Only] – A total of 3301 message buffers (Individual 384 buffers + Shared 2917 buffers when 8 byte data payload) provided for 12 channels [For U2A6 (BGA-292) Only] – A total of 3026 message buffers (Individual 352 buffers + Shared 2674 buffers when 8 byte data payload) provided for 11 channels [For U2A6 (QFP-176) Only] – A total of 2200 message buffers (Individual 256 buffers + Shared 1944 buffers when 8 byte data payload) provided for 8 channels [For U2A6 (BGA-156) Only] – A total of 1926 message buffers (Individual 224 buffers + Shared 1702 buffers when 8 byte data payload) provided for 7 channels [For U2A6 (QFP-144) Only] – Reception filtering • CAN FD mode – Conforming to CAN-FD ISO 11898-1 (2015) – Transfer speed up to 8 Mbps – A total of 5120 message buffers (Individual 1024 buffers + Shared 4096 buffers when 64 byte data payload) provided for 16 channels [For U2A-EVA, U2A16 and U2A8 Only] – A total of 1152 message buffers (Individual 384 buffers + Shared 768 buffers when 64 byte data payload) provided for 12 channels [For U2A6 (BGA-292) Only] – A total of 1056 message buffers (Individual 352 buffers + Shared 704 buffers when 64 byte data payload) provided for 11 channels [For U2A6 (QFP-176) Only] – A total of 768 message buffers (Individual 256 buffers + Shared 512 buffers when 64 byte data payload) provided for 8 channels [For U2A6 (BGA-156) Only] – A total of 672 message buffers (Individual 224 buffers + Shared 448 buffers when 64 byte data payload) provided for 7 channels [For U2A6 (QFP-144) Only] – Reception filtering
FlexRay (FLXA)	2 units incorporated [For U2A-EVA, U2A16 and U2A8 Only] 1 unit incorporated [For U2A6 only] • Conforming to Protocol Specification v2.1 • Buffer size: A 8-KB space is divided into up to 128 sections (for transmission, reception, and receive FIFO) • Message filtering: slot counter filter, channel filter, cycle counter filter • Bit rate: 10 Mbps
LIN/UART interface (RLIN3)	24 units incorporated [For U2A-EVA (BGA-516), U2A16 (BGA-516), U2A16 (BGA-373) and U2A8 (BGA-373) Only] 12 units incorporated [For U2A16 (BGA-292), U2A8 (BGA-292), U2A6 (BGA-292) and U2A6 (QFP-176) Only] 8 units incorporated [For U2A6 (BGA-156) Only] 10 units incorporated [For U2A6 (QFP-144) Only] • Conforming to LIN Protocol Spec versions 1.3, 2.0, 2.1, 2.2, and SAE J2602 • Three operating modes – LIN Master mode – LIN Slave mode – UART mode (half-duplex, full-duplex) • Arbitrary bit rate is selectable by the on-chip baud rate generator • LIN Self-test mode with internal data loop back
Clock Extension Peripheral Interface (CXPI)	4 units incorporated [For U2A-EVA, U2A16 and U2A8 Only] Baud rate: 9.6 kbps/10.4 kbps/19.2 kbps/20.0 kbps Master or Slave Mode Event trigger method / Polling method Output and sample the PWM waveform by PWM encoding/decoding function

I^2C Bus Interface (RIIC)	2 units incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (BGA-156) Only] 1 unit incorporated [For U2A6 (QFP-144) Only] I^2C bus format with master mode or slave mode selectable Transfer rate up to 400 kbps
Single Edge Nibble Transmission (RSENT)	8 units incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (BGA-156) Only] 6 units incorporated [For U2A6 (QFP-144) Only] • Conforming to the SENT (Single Edge Nibble Transmission) protocol specified in the SAE J2716_201604 standard and the SPC (Short PWM Code) extension to the SENT specification • Unidirectional or bidirectional transfer is possible through a single pin • Bidirectional transfer is possible through two pins • Data transfer protected by a CRC is possible
Peripheral Sensor Interface 5 (PSI5)	4 channels incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (BGA-156) Only] 3 channels incorporated [For U2A6 (QFP-144) Only] • Conformance with PSI5 protocol specification V2.0 • Bit rates: Low speed(125 kbps), High speed(189 kbps), PAS compatibility mode(250 kbps) • Communication Mode selectable
Peripheral Sensor Interface 5 serial communication module (PSI5S)	2 units incorporated [For U2A-EVA, U2A16, U2A8 and U2A6 (BGA-292) Only] 1 unit incorporated [For U2A6 (QFP-176), U2A6 (BGA-156) and U2A6 (QFP-144) Only] • Support the UART based communication for PSI5 transceiver • Conformance with PSI5 protocol specification V2.2 • Generate a PSI5 message from the UART transfer data • The bit rate of the UART can be set by the built-in baud rate generator
RHSIF	1 unit incorporated [For U2A-EVA, U2A16 and U2A8 Only] • Asynchronous high speed LVDS interface based on IEEE 1596.3-1996 reduced range link • Asynchronous high speed LVDS interface supporting maximum data rates of 320 Mbps • Four channels, including one channel with data streaming capability • Bus master interface which is used by a target node to access shared memory
Ethernet Controller (ETNB)	1 channel of 100Mbps ethernet incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (QFP-144) Only] 1 channel of 1Gbps ethernet incorporated [For U2A-EVA, U2A16 and U2A8 Only] • Conformance with the IEEE 802.3 MAC layer standard • PHY interface: MII (Media Independent Interface) and RMII (Reduced Media Independent Interface) for 100Mbps; SGMII (Serial Gigabit Media Independent Interface) is for 1Gbps • Supports 1Gbps and 100 Mbps or 10 Mbps • Supports full-duplex mode • built-in DMA transfer function

Analog to Digital Converter (ADCJ)	94 channels incorporated [For U2A-EVA (BGA-516), U2A16 (BGA-516), U2A16 (BGA-373) and U2A8 (BGA-373) Only] 79 channels incorporated [For U2A16 (BGA-292), U2A8 (BGA-292) and U2A6 (BGA-292) Only] 64 channels incorporated [For U2A6 (QFP-176) Only] 29 channels incorporated [For U2A6 (BGA-156) Only] 39 channels incorporated [For U2A6 (QFP-144) Only] • A/D conversion method: Successive approximation • Configuration of analog input pins – ADCJ0/ADCJ1/ADCJ2 high accuracy inputs: 20/20/20 [For U2A-EVA (BGA-516) and U2A16 (BGA-516), U2A16 (BGA-373) and U2A8 (BGA-373) Only] – ADCJ0/ADCJ1/ADCJ2 high accuracy inputs: 20/20/5 [For U2A16 (BGA-292), U2A8 (BGA-292) and U2A6 (BGA-292) Only] – ADCJ0/ADCJ1/ADCJ2 high accuracy inputs: 14/14/5 [For U2A6 (QFP-176) Only] – ADCJ0/ADCJ1/ADCJ2 high accuracy inputs: 8/8/0 [For U2A6 (BGA-156) Only] – ADCJ0/ADCJ1/ADCJ2 high accuracy inputs: 10/10/0 [For U2A6 (QFP-144) Only] – ADCJ0/ADCJ1/ADCJ2 low accuracy inputs: 10/14/10 [For U2A-EVA, U2A16, U2A8 and U2A6 (BGA-292) Only] – ADCJ0/ADCJ1/ADCJ2 low accuracy inputs: 10/14/7 [For U2A6 (QFP-176) Only] – ADCJ0/ADCJ1/ADCJ2 low accuracy inputs: 6/7/0 [For U2A6 (BGA-156) Only] – ADCJ0/ADCJ1/ADCJ2 low accuracy inputs: 3/10/6 [For U2A6 (QFP-144) Only] • Resolution: 12-bit • Conversion speed: 1.0 µs • Scan groups for five systems for each converter • Two scan modes (multicycle scan mode and continuous scan mode) • ADCJ0: Up to 64 virtual channels • ADCJ1: Up to 64 virtual channels • ADCJ2: Up to 64 virtual channels • Addition mode A/D conversion functions incorporated • Can enter data directly to the Generic Timer Module. • Safety functions • Supporting an upper / lower-limit-excess-notice-function for the ADC Voltage Monitor Secondary Error Generator in each virtual channel • Track & Hold (T&H) input channels 4 channel inputs per unit can select T&H circuit for synchronize conversion.
Power supply	Supports both of 5 V, 3.3 V and 1.09 V power supplies with the exception of the core supply. The power on / off sequence has no constraints.
Power supply voltage monitor	• The power supply voltage monitor is used for monitoring power domain E0VCC, VCC, ISOVDD and AWOVDD. • The power supply voltage monitor has High-side (HDET) and Low-side (LDET) voltage detectors, which detect if the monitored voltage is more or less than the specified voltage. • The power supply voltage monitor have two types detection function of Primary detection and Secondary detection. • The Primary detection function is performed by Voltage Monitor. The Secondary detection function is performed by the ADCJ. • The Delay Monitor (DMON) assists the VMON which detects the Low-side voltage of ISOVDD. • Primary power supply voltage monitor can control on or off for the VMON reset generated by high level/low level detection of ISOVDD, VCC and E0VCC. • The Primary detection voltage value is fixed, secondary detection voltage value can be set by the ADCJ.
Secure Watchdog Timer (SWDT)	1 unit incorporated Can generate an error signal for the Interrupt Controller in response to an error. Confirmation of matching with a specified Program Counter (PC) value of the CPU0.
Intelligent Cryptographic Unit Master (ICUMHA)	The ICUMHA is an on-chip Hardware Security Module (HSM). The ICUMHA supports user-defined security services to the overall system based on cryptographic primitives.

Debugging and Calibration	Nexus JTAG: One channel incorporated LPD (4-pin): One channel incorporated AUDR: One channel incorporated [For U2A-EVA Only] Aurora Trace Interface: One channel incorporated [For U2A-EVA Only]
Boundary scan	Supports boundary scan conforming to the IEEE1149.1 standard
Clock controller	The user is able to select the crystal resonance frequency (16 MHz or 20 MHz or 24 MHz or 40 MHz). Incorporates a crystal resonation circuit (Main OSC), which is used as a reference clock for the PLL. Incorporates High Speed Internal Oscillator (HS IntOSC) and Low Speed Internal Oscillator (LS IntOSC), which are used as the start-up clock and backup clock. Incorporates High Voltage Internal Oscillator (HV IntOSC), which is used as digital noise filter clock for VMON. Incorporates a PLL circuit to generate high speed internal clocks by multiplying the Main OSC input. Generates clock pulses used inside the chip from the internal oscillator, main oscillator and PLL. Software configurable external clock output.
Operating modes	Operating modes ● Normal Operating Mode ● User Boot Mode ● Serial Programming Mode ● Boundary scan mode
Standby controller	This product supports various power-down modes. The power consumption of the MCU can be reduced by selecting one of the modes: • RUN mode RUN mode is a normal operation mode where the CPU is operating and all of other modules can operate. The CPU can enter “HALT” state by executing the “HALT” instruction to stop its operation in this mode. • STOP mode STOP mode is a chip-level stand-by mode in which the clock supply to a certain clock domain can be stopped. • DeepSTOP mode DeepSTOP mode is a chip-level stand-by mode to reduce power consumption further than STOP mode. In addition to the clock supply stop, the power supply to the Isolated area is switched off. • Cyclic RUN mode Cyclic RUN mode is a low-power operation mode in which limited modules can operate at low speed. • Cyclic STOP mode Cyclic STOP mode is a STOP mode in cyclic operation, and one CPU halts its operation. • Module standby
Reset controller	7 reset functions • Power On Reset • System Reset 1 • System Reset 2 • Application Reset • DeepSTOP Reset • Module Reset • JTAG Reset External Reset output pin: $\overline{\text{RESETOUT}}$ Automatic RAM initialization after reset (include DTSRAM, sDMAC Descriptor RAM, GTM RAM, MMCA RAM and MSPI RAM).
Clock monitor	• Up to 10 clock monitors depending on the device configuration. • Detects clock disturbances that results in lower or higher frequency than target frequency, and sends an error notification to the ECM. Supports the self-diagnosis function.
Low-Power Sampler (LPS)	1 unit incorporated [For U2A-EVA, U2A16, U2A8, U2A6 (BGA-292), U2A6 (QFP-176) and U2A6 (QFP-144) Only] Support checking the digital input ports and analog input ports to monitor the external input without consuming CPU resources.

Temperature sensor	1 sensor incorporated. • Out of range detection of temperature. • Operating modes – Single measurement mode – Continuous measurement mode • Interrupt generation – Temperature Measurement End Interrupt (INTOTSOTI) – Temperature Rise/Drop Interrupt (INTOTSOTULI) – Abnormal temperature error signal (OTABE) – Temperature Sensor Error (INTOTSOTE) • Support self-diagnosis function
Generic Timer Module (GTM)	1 unit incorporated GTM v3.5 is a modular timer unit and consists of the following submodules. • Advanced Routing Unit (ARU) • Clock Management Unit (CMU) • Cluster Configuration Module (CCM) • Time Base Unit (TBU) • Timer Input Module (TIM) • Advanced Timer Output (ATOM) • Dead Time Module (DTM) • Multi Channel Sequencer (MCS) • Interrupt Concentrator Module (ICM) • Output Compare Unit (CMP) • Monitoring Unit (MON)
Encoder Timer (ENCA)	2 units incorporated [For U2A-EVA, U2A16, U2A8 and U2A6 (BGA-292) Only] 1 unit incorporated [For U2A6 (QFP-176) and U2A6 (QFP-144) Only] • Generation of the counter control signal from the encoder input signal, and count operation. • Capture function for capturing the counter value with an external trigger signal • Compare function for compare match judgment with the counter value • Two capture compare registers that can be set separately for capture operation and for compare operation • Interrupt mask function for masking the interrupt request signal output as a result of the compare match judgment during compare operation • Function for loading the value of the capture compare register to the counter upon underflow occurrence • The Encoder input signal can be applied to the timer counter clear condition • Edge or level can be selected for clearing the encoder input signal of the timer counter clear condition • Detection of counter overflow and underflow and output of error flags and error occurrence interrupts • Five interrupts: two capture compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt.
Package	[For U2A-EVA (BGA-516) and U2A16 (BGA-516) Only] 516-pin plastic FBGA (0.8 mm ball pitch) (25mm × 25mm package size) [For U2A16 (BGA-373) and U2A8 (BGA-373) Only] 373-pin plastic FBGA (0.8 mm ball pitch) (21mm × 21mm package size) [For U2A16 (BGA-292), U2A8 (BGA-292) and U2A6 (BGA-292) Only] 292-pin plastic FBGA (0.8 mm ball pitch) (17mm × 17mm package size) [For U2A6 (QFP-176) Only] 176-pin plastic HLQFP (0.5 mm pin pitch) (24mm × 24mm package size) [For U2A6 (BGA-156) Only] 156-pin plastic LFBGA (0.65 mm ball pitch) (10mm × 10mm package size) [For U2A6 (QFP-144) Only] 144-pin plastic HLQFP (0.4 mm pin pitch) (16mm × 16mm package size)

1.3 Application Fields

- Automotive field (including body control and chassis & safety)

1.4 Ordering Information

Table 1.1 Product Name List (1/2)

Name	Package	On-Chip ROM	Operating Temperature (Tj)	External Oscillator	Maximum Operating Frequency	MSPI restriction *1	SVR restriction *3	Note
R7F702Z19AEDBG (RH850/U2A-EVA)	Plastic FBGA-516 0.8-mm ball pitch 25 mm × 25 mm (RENESAS code*2 = PRBG0516GC-A)	16 MB	max. 160°C	16/20/24/40 MHz	400 MHz	Yes	No	*4
R7F702Z19BFDBG (RH850/U2A-EVA)	Plastic FBGA-516 0.8-mm ball pitch 25 mm × 25 mm (RENESAS code*2 = PRBG0516GC-A)	16 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702300EBBG-C (RH850/U2A16)	Plastic FBGA-516 0.8-mm ball pitch 25 mm × 25 mm (RENESAS code*2 = PRBG0516GC-A)	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	Yes	Yes	*4
R7F702300EBBB-C (RH850/U2A16)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	Yes	Yes	*4
R7F702300EABA-C (RH850/U2A16)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	16 MB	max. 160°C	16/20/24/40 MHz	400 MHz	Yes	Yes	*4
R7F702300AEBBC-C (RH850/U2A16)	Plastic FBGA-516 0.8-mm ball pitch 25 mm × 25 mm (RENESAS code*2 = PRBG0516GD-A	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	Yes	*4
R7F702300AEBBB-C (RH850/U2A16)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	Yes	*4
R7F702300AFABA-C (RH850/U2A16)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	16 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	Yes	*4
R7F702300BEBBC-C (RH850/U2A16)	Plastic FBGA-516 0.8-mm ball pitch 25 mm × 25 mm (RENESAS code*2 = PRBG0516GD-A)	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702300BEBBB-C (RH850/U2A16)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	16 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702300BFABA-C (RH850/U2A16)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	16 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702301EBBA-C (RH850/U2A8)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	8 MB	max. 150°C	16/20/24/40 MHz	400 MHz	Yes	Yes	*4
R7F702301EABG-C (RH850/U2A8)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	8 MB	max. 160°C	16/20/24/40 MHz	400 MHz	Yes	Yes	*4
R7F702301AEBBA-C (RH850/U2A8)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	8 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	Yes	*4
R7F702301AFABG-C (RH850/U2A8)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	8 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	Yes	*4
R7F702301BEBBA-C (RH850/U2A8)	Plastic FBGA-373 0.8-mm ball pitch 21 mm × 21 mm	8 MB	max. 150°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702301BFABG-C (RH850/U2A8)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	8 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702302FABB-C (RH850/U2A6)	Plastic FBGA-292 0.8-mm ball pitch 17 mm × 17 mm	6 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—

Table 1.1 Product Name List (2/2)

Name	Package	On-Chip ROM	Operating Temperature (Tj)	External Oscillator	Maximum Operating Frequency	MSPI restriction *1	SVR restriction *3	Note
R7F702302FAFK-C (RH850/U2A6)	Plastic HLQFP-176 0.5 mm pin pitch 24 mm × 24 mm	6 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702302FABD-C (RH850/U2A6)	Plastic LFBGA-156 0.65 mm ball pitch 10 mm × 10 mm	6 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—
R7F702302FAFM-C (RH850/U2A6)	Plastic HLQFP-144 0.4 mm pin pitch 16 mm × 16 mm	6 MB	max. 160°C	16/20/24/40 MHz	400 MHz	No	No	—

Note 1. Refer to **Section 19.8, MSPI restrictions**.

Note 2. Refer to **Section 54.1, Package Outline**.

Note 3. Refer to **Section 10.10, SVR restriction**.

Note 4. As sales of these products is limited, please contact your sales representative first when considering these products for purchasing and project development.

FBGA is hereafter referred to as BGA unless the complete abbreviation is required.

The RH850/U2A-EVA series has 21 variants. Select the proper product according to the table below.

Table 1.2 RH850/U2A-EVA List

Frequency	Package					
	FBGA-516	FBGA-373	FBGA-292	HLQFP-176	LFBGA-156	HLQFP-144
400 MHz	R7F702Z19AEDBG (RH850/U2A-EVA)*1 R7F702Z19BFDBG (RH850/U2A-EVA) R7F702300EBBG-C (RH850/U2A16)*1 R7F702300AEBBC-C (RH850/U2A16)*1 R7F702300BEBBC-C (RH850/U2A16)	R7F702300EBBB-C (RH850/U2A16)*1 R7F702300AEBBB-C (RH850/U2A16)*1 R7F702300BEBBB-C (RH850/U2A16) R7F702301EBBA-C (RH850/U2A8)*1 R7F702301AEBBA-C (RH850/U2A8)*1 R7F702301BEBBA-C (RH850/U2A8)	R7F702300EABA-C (RH850/U2A16)*1 R7F702300AFABA-C (RH850/U2A16)*1 R7F702300BFABA-C (RH850/U2A16) R7F702301EABG-C (RH850/U2A8)*1 R7F702301AFABG-C (RH850/U2A8)*1 R7F702301BFABG-C (RH850/U2A8) R7F702302FABB-C (RH850/U2A6)	R7F702302FAFK-C (RH850/U2A6)	R7F702302FABD-C (RH850/U2A6)	R7F702302FAFM-C (RH850/U2A6)

Note 1. As sales of these products is limited, contact your sales representative first when considering these products for purchasing and project development.

1.5 Differences in the Specifications of RH850/U2A-EVA Products

The table below lists the differences in the specifications of RH850/U2A-EVA products.

(1/3)

Product			RH850/U2A-EVA 516 pins	RH850/U2A16 516 pins	RH850/U2A16 373 pins	RH850/U2A16 292 pins	RH850/U2A8 373 pins	RH850/U2A8 292 pins	RH850/U2A6 292 pins	RH850/U2A6 176 pins	RH850/U2A6 156 pins	RH850/U2A6 144 pins
CPU	Frequency		400	400	400	400	400	400	400	400	400	400
	Main Core		4	4	4	4	2	2	2	2	2	2
	Lockstep		4	4	4	4	2	2	2	2	2	2
	FPU		4	4	4	4	2	2	2	2	2	2
	Instruction Cache per core (for Code Flash)		16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)	16 KB (PBS 4w)
	Data Buffer per core (for Code Flash)		4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)	4 lines (256 bit/ line)
	MPU regions per core		32	32	32	32	32	32	32	32	32	32
	IPIR	Unit	1	1	1	1	1	1	1	1	1	1
	BARR	Unit	1	1	1	1	1	1	1	1	1	1
	TPTM	Unit	1	1	1	1	1	1	1	1	1	1
RAM	Total RAM (Local RAM + Cluster RAM)		3584 KB	3584 KB	3584 KB	3584 KB	1792 KB	1792 KB	768 KB	768 KB	768 KB	768 KB
	Local RAM per core		64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
	Cluster RAM (CRAM)	Total Size (Retention RAM included)	3328 KB	3328 KB	3328 KB	3328 KB	1664 KB	1664 KB	640 KB	640 KB	640 KB	640 KB
		Retention RAM (included in CRAM)	256 KB	256 KB	256 KB	256 KB	128 KB	128 KB	128 KB	128 KB	128 KB	128 KB
	Cluster Emulation RAM		2 MB (1 MB for each cluster 0 & 1)	No	No	No	No	No	32 KB (cluster 0)	32 KB (cluster 0)	32 KB (cluster 0)	32 KB (cluster 0)
	Global Emulation RAM		2 MB	No	No	No	No	No	No	No	No	No
	Instrumentation RAM		96 KB	No	No	No	No	No	No	No	No	No
	Trace RAM		64 KB	No	No	No	No	No	32 KB	32 KB	32 KB	32 KB
Flash	Code Flash		16 MB	16 MB	16 MB	16 MB	8 MB	8 MB	6 MB	6 MB	6 MB	6 MB
	Data Flash	Total Size	576 KB	576 KB	576 KB	576 KB	320 KB	320 KB	256 KB	256 KB	256 KB	256 KB
		Dedicated ICUMHA	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
DMA	sDMAC	Channels	32	32	32	32	32	32	32	32	32	32
	DTS	Channels	128	128	128	128	128	128	128	128	128	128
I/O port			301	301	221	165	221	165	165	129	77	97
Timer	ENCA	Units	2	2	2	2	2	2	2	1	No	1
	OSTM	Units	10	10	10	10	8	8	8	8	8	8
	WDTB	Units	5	5	5	5	3	3	3	3	3	3
	LTSC	Unit	1	1	1	1	1	1	1	1	1	1
	SWDT	Unit	1	1	1	1	1	1	1	1	1	1
	GTM	Unit	1	1	1	1	1	1	1	1	1	1
	TAPA	Units	4	4	4	4	4	4	4	3	2	2
	TPBA	Units	2	2	2	2	2	2	2	2	2	2
	TAUD	Units	3	3	3	3	3	3	3	3	3	3
	TAUJ	Units	4	4	4	4	4	4	4	4	4	4
	TSG3	Units	2	2	2	2	2	2	2	2	1	1
	PWM-Diag	Units	96	96	96	96	96	96	96	76	50	64
	RTCA	Unit	1	1	1	1	1	1	1	1	1	1

(2/3)

Product			RH850/U2A-EVA 516 pins	RH850/U2A16 516 pins	RH850/U2A16 373 pins	RH850/U2A16 292 pins	RH850/U2A8 373 pins	RH850/U2A8 292 pins	RH850/U2A6 292 pins	RH850/U2A6 176 pins	RH850/U2A6 156 pins	RH850/U2A6 144 pins
Communication	SCI3	Units	3	3	3	3	3	3	No	No	No	No
	RLIN3	Units	24	24	24	12	24	12	12	12	8	10
	CXPI	Units	4	4	4	4	4	4	No	No	No	No
	MSPI	Units	10	10	9	6	9	6	6	6	6	4
	RHSIF	Unit	1	1	1	1	1	1	No	No	No	No
	RS-CANFD	Units [Channels]	2 [16ch]	2 [16ch]	2 [16ch]	2 [16ch]	2 [16ch]	2 [16ch]	2 [12ch]	2 [11ch]	2 [8ch]	2 [7ch]
	FlexRay	Units [Channels]	2 [4ch]	2 [4ch]	2 [4ch]	2 [4ch]	2 [4ch]	2 [4ch]	1 [2ch]	1 [2ch]	1 [2ch]	1 [2ch]
	Ethernet	Channel (100Mb/1Gb)	1/1	1/1	1/1	1/1	1/1	1/1	1/0	1/0	No	1/0
		ETNB0 (MII/RMII/SGMII)	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	Yes/Yes/No	No/No/No	Yes/Yes/No
		ETNB1 (MII/RMII/SGMII)	Yes/No/Yes	Yes/No/Yes	No/No/Yes	No/No/Yes	No/No/Yes	No/No/Yes	No/No/No	No/No/No	No/No/No	No/No/No
	RSENT	Units	8	8	8	8	8	8	8	8	8	6
	PSI5	Channels	4	4	4	4	4	4	4	4	4	3
	PSI5S	Units	2	2	2	2	2	2	2	1	1	1
	RIIC	Units	2	2	2	2	2	2	2	2	2	1
	eMMC	Unit	1	1	1	1	1	1	1	1	No	No
	SFMA	Unit	1	1	1	1	1	1	1	1	No	No
Safety	ASIL level		N/A	D	D	D	D	D	D	D	D	D
	ECM		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	CRC(KCRC)	Units	8	8	8	8	8	8	8	8	8	8
	Clock Monitor		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Error injection for self-diagnosis of several safety mechanisms		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	LBIST		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	MBIST		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Bus ECC		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ADC	ADCJ	Units	3	3	3	3	3	3	3	3	2	3
		ADCJ0/1/2 (Channels) (High accuracy inputs)	20/20/20	20/20/20	20/20/20	20/20/5	20/20/20	20/20/5	20/20/5	14/14/5	8/8/0	10/10/0
		ADCJ0/1/2 (Low accuracy inputs)	10/14/10	10/14/10	10/14/10	10/14/10	10/14/10	10/14/10	10/14/10	10/14/7	6/7/0	3/10/6
		Total inputs	94	94	94	79	94	79	79	64	29	39
		Virtual channels per unit	64	64	64	64	64	64	64	64	64	64
		ADC timers	2	2	2	2	2	2	2	2	2	2
Debug	Nexus-JTAG		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Trace I/F (Aurora)		4 lanes	No	No	No	No	No	No	No	No	No
	Low Pin Debug I/F (4-pin)		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Bypass I/F (AUDR)		Yes	No	No	No	No	No	No	No	No	No
Security	Basic Hardware Protection (BHP)		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ICUMHA		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Secure RAM		64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
	Secure Data Flash		64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB	64 KB
	Code Flash Protection		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
System Control	Temp Sensor		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Voltage Monitor		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	LPS		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes

(3/3)

Product		RH850/ U2A-EVA 516 pins	RH850/ U2A16 516 pins	RH850/ U2A16 373 pins	RH850/ U2A16 292 pins	RH850/ U2A8 373 pins	RH850/ U2A8 292 pins	RH850/ U2A6 292 pins	RH850/ U2A6 176 pins	RH850/ U2A6 156 pins	RH850/ U2A6 144 pins
Package*1	BGA516	Yes	Yes	No	No	No	No	No	No	No	No
	BGA373	No	No	Yes	No	Yes	No	No	No	No	No
	BGA292	No	No	No	Yes	No	Yes	Yes	No	No	No
	HLQFP176	No	No	No	No	No	No	No	Yes	No	No
	BGA156	No	No	No	No	No	No	No	No	Yes	No
	HLQFP144	No	No	No	No	No	No	No	No	No	Yes

Note 1. For product names, refer to **Section 1.4, Ordering Information**.

1.6 Pin Connection Diagram (Top View)

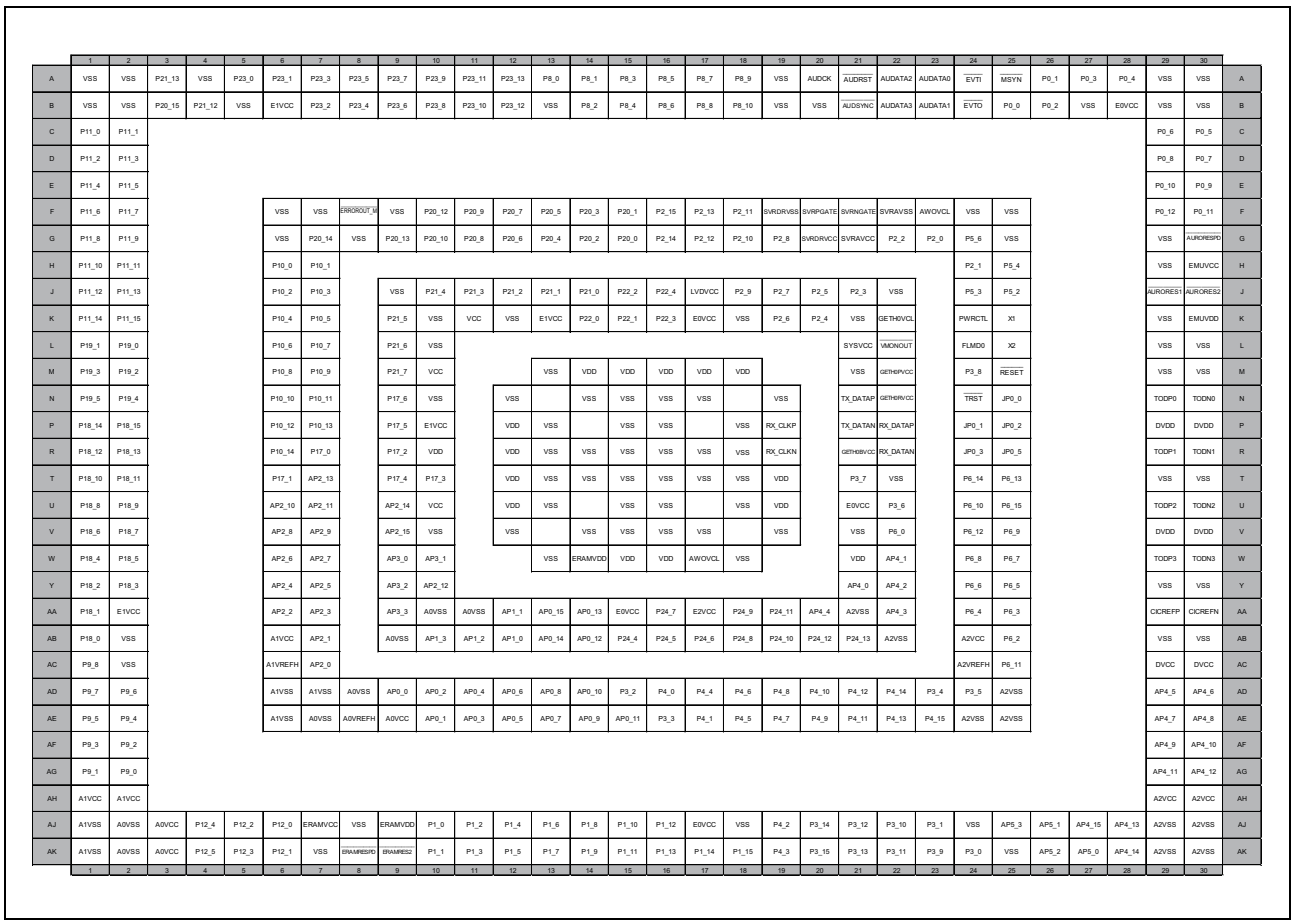
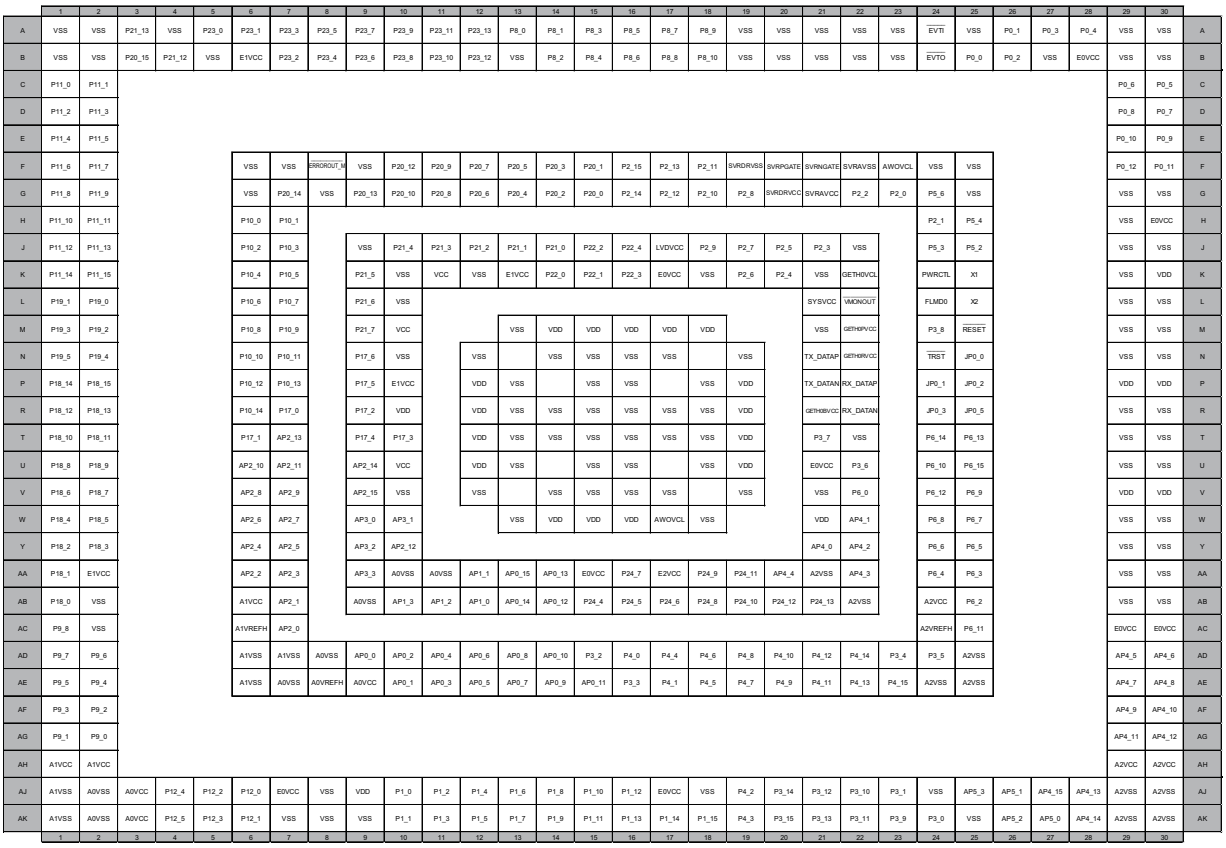


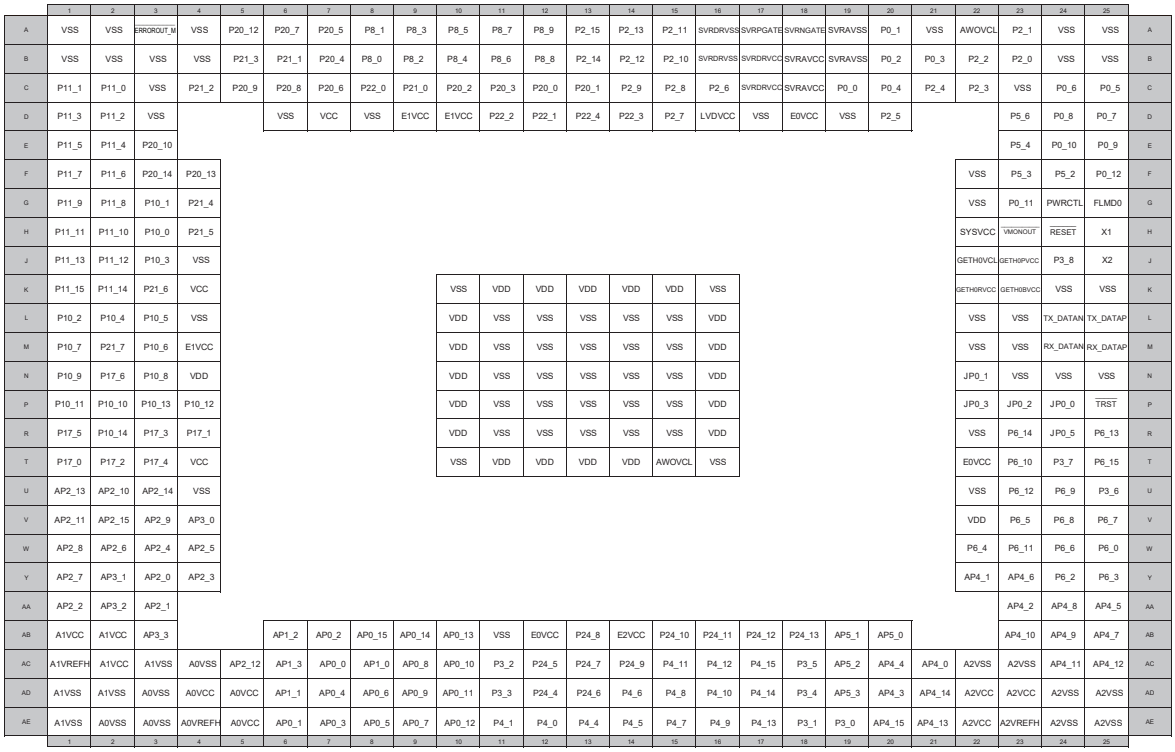
Figure 1.1 Pin Connection Diagram (BGA516 U2A-EVA)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
A	VSS	VSS	P21_13	VSS	P23_0	P23_1	P23_3	P23_5	P23_7	P23_9	P23_11	P23_13	P8_0	P8_1	P8_3	P8_5	P8_7	P8_9	VSS	VSS	VSS	VSS	VSS	EVTI	VSS	P0_1	P0_3	P0_4	VSS	VSS	A
B	VSS	VSS	P20_15	P21_12	VSS	E1VCC	P23_2	P23_4	P23_6	P23_8	P23_10	P23_12	VSS	P8_2	P8_4	P8_6	P8_8	P8_10	VSS	VSS	VSS	VSS	VSS	EVTO	P0_0	P0_2	VSS	E0VCC	VSS	VSS	B
C	P11_0	P11_1																											P0_6	P0_5	C
D	P11_2	P11_3																											P0_8	P0_7	D
E	P11_4	P11_5																											P0_10	P0_9	E
F	P11_6	P11_7																											P0_12	P0_11	F
G	P11_8	P11_9																											VSS	VSS	G
H	P11_10	P11_11																											VSS	E0VCC	H
J	P11_12	P11_13																											VSS	VSS	J
K	P11_14	P11_15																											VSS	VDD	K
L	P19_1	P19_0																											VSS	VSS	L
M	P19_3	P19_2																											VSS	VSS	M
N	P19_5	P19_4																											VSS	VSS	N
P	P18_14	P18_15																											VDD	VDD	P
R	P18_12	P18_13																											VSS	VSS	R
T	P18_10	P18_11																											VSS	VSS	T
U	P18_8	P18_9																											VSS	VSS	U
V	P18_6	P18_7																											VDD	VDD	V
W	P18_4	P18_5																											VSS	VSS	W
Y	P18_2	P18_3																											VSS	VSS	Y
AA	P18_1	E1VCC																											VSS	VSS	AA
AB	P18_0	VSS																											VSS	VSS	AB
AC	P9_8	VSS																											E0VCC	E0VCC	AC
AD	P9_7	P9_6																											AP4_5	AP4_6	AD
AE	P9_5	P9_4																											AP4_7	AP4_8	AE
AF	P9_3	P9_2																											AP4_9	AP4_10	AF
AG	P9_1	P9_0																											AP4_11	AP4_12	AG
AH	A1VCC	A1VCC																											A2VCC	A2VCC	AH
AJ	A1VSS	A0VSS	A0VCC	P12_4	P12_2	P12_0	E0VCC	VSS	VDD	P1_0	P1_2	P1_4	P1_6	P1_8	P1_10	P1_12	E0VCC	VSS	P4_2	P3_14	P3_12	P3_10	P3_1	VSS	AP5_3	AP5_1	AP4_15	AP4_13	A2VSS	A2VSS	AJ
AK	A1VSS	A0VSS	A0VCC	P12_5	P12_3	P12_1	VSS	VSS	VSS	P1_1	P1_3	P1_5	P1_7	P1_9	P1_11	P1_13	P1_14	P1_15	P4_3	P3_15	P3_13	P3_11	P3_9	P3_0	VSS	AP5_2	AP5_0	AP4_14	A2VSS	A2VSS	AK
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	

Note: The pin location of RX_DATAP, RX_DATAN, TX_DATAP and TX_DATAN is defined by the Option Byte 14. For details, refer to **Section 51.12.20, OPBT14 — Option Byte 14**.

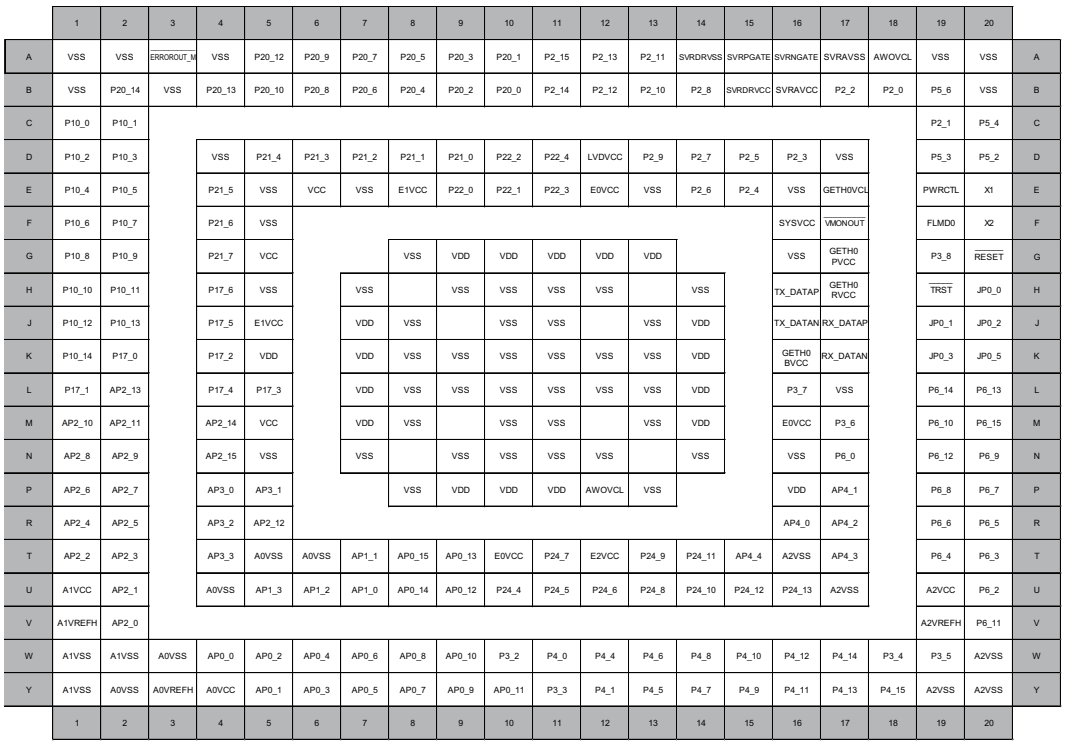
Figure 1.2 Pin Connection Diagram (BGA516 U2A16)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
A	VSS	VSS	ERROROUT_M	VSS	P20_12	P20_7	P20_5	P8_1	P8_3	P8_5	P8_7	P8_9	P2_15	P2_13	P2_11	SVRDRVSS	SVRPGATE	SVRNGATE	SVRAVSS	P0_1	VSS	AWOVCL	P2_1	VSS	VSS	A
B	VSS	VSS	VSS	VSS	P21_3	P21_1	P20_4	P8_0	P8_2	P8_4	P8_6	P8_8	P2_14	P2_12	P2_10	SVRDRVSS	SVRDRVCC	SVRAVCC	SVRAVSS	P0_2	P0_3	P2_2	P2_0	VSS	VSS	B
C	P11_1	P11_0	VSS	P21_2	P20_9	P20_8	P20_6	P22_0	P21_0	P20_2	P20_3	P20_0	P20_1	P2_9	P2_8	P2_6	SVRDRVCC	SVRAVCC	P0_0	P0_4	P2_4	P2_3	VSS	P0_6	P0_5	C
D	P11_3	P11_2	VSS			VSS	VCC	VSS	E1VCC	E1VCC	P22_2	P22_1	P22_4	P22_3	P2_7	LVDVCC	VSS	E0VCC	VSS	P2_5			P5_6	P0_8	P0_7	D
E	P11_5	P11_4	P20_10																				P5_4	P0_10	P0_9	E
F	P11_7	P11_6	P20_14	P20_13																		VSS	P5_3	P5_2	P0_12	F
G	P11_9	P11_8	P10_1	P21_4																		VSS	P0_11	PWRCTL	FLMD0	G
H	P11_11	P11_10	P10_0	P21_5																		SYSVCC	VMONOUT	RESET	X1	H
J	P11_13	P11_12	P10_3	VSS																		GETH0VCL	GETH0PVCC	P3_8	X2	J
K	P11_15	P11_14	P21_6	VCC						VSS	VDD	VDD	VDD	VDD	VDD	VSS						GETH0RVCC	GETH0BVCC	VSS	VSS	K
L	P10_2	P10_4	P10_5	VSS						VDD	VSS	VSS	VSS	VSS	VSS	VDD						VSS	VSS	TX_DATAN	TX_DATAP	L
M	P10_7	P21_7	P10_6	E1VCC						VDD	VSS	VSS	VSS	VSS	VSS	VDD						VSS	VSS	RX_DATAN	RX_DATAP	M
N	P10_9	P17_6	P10_8	VDD						VDD	VSS	VSS	VSS	VSS	VSS	VDD						JP0_1	VSS	VSS	VSS	N
P	P10_11	P10_10	P10_13	P10_12						VDD	VSS	VSS	VSS	VSS	VSS	VDD						JP0_3	JP0_2	JP0_0	TRST	P
R	P17_5	P10_14	P17_3	P17_1						VDD	VSS	VSS	VSS	VSS	VSS	VDD						VSS	P6_14	JP0_5	P6_13	R
T	P17_0	P17_2	P17_4	VCC						VSS	VDD	VDD	VDD	VDD	AWOVCL	VSS						E0VCC	P6_10	P3_7	P6_15	T
U	AP2_13	AP2_10	AP2_14	VSS																		VSS	P6_12	P6_9	P3_6	U
V	AP2_11	AP2_15	AP2_9	AP3_0																		VDD	P6_5	P6_8	P6_7	V
W	AP2_8	AP2_6	AP2_4	AP2_5																		P6_4	P6_11	P6_6	P6_0	W
Y	AP2_7	AP3_1	AP2_0	AP2_3																		AP4_1	AP4_6	P6_2	P6_3	Y
AA	AP2_2	AP3_2	AP2_1																				AP4_2	AP4_8	AP4_5	AA
AB	A1VCC	A1VCC	AP3_3			AP1_2	AP0_2	AP0_15	AP0_14	AP0_13	VSS	E0VCC	P24_8	E2VCC	P24_10	P24_11	P24_12	P24_13	AP5_1	AP5_0			AP4_10	AP4_9	AP4_7	AB
AC	A1VREFH	A1VCC	A1VSS	A0VSS	AP2_12	AP1_3	AP0_0	AP1_0	AP0_8	AP0_10	P3_2	P24_5	P24_7	P24_9	P4_11	P4_12	P4_15	P3_5	AP5_2	AP4_4	AP4_0	A2VSS	A2VSS	AP4_11	AP4_12	AC
AD	A1VSS	A1VSS	A0VSS	A0VCC	A0VCC	AP1_1	AP0_4	AP0_6	AP0_9	AP0_11	P3_3	P24_4	P24_6	P4_6	P4_8	P4_10	P4_14	P3_4	AP5_3	AP4_3	AP4_14	A2VCC	A2VCC	A2VSS	A2VSS	AD
AE	A1VSS	A0VSS	A0VSS	A0VREFH	A0VCC	AP0_1	AP0_3	AP0_5	AP0_7	AP0_12	P4_1	P4_0	P4_4	P4_5	P4_7	P4_9	P4_13	P3_1	P3_0	AP4_15	AP4_13	A2VCC	A2VREFH	A2VSS	A2VSS	AE
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	

Note: The pin location of RX_DATAP, RX_DATAN, TX_DATAP and TX_DATAN is defined by the Option Byte 14. For details, refer to **Section 51.12.20, OPBT14 — Option Byte 14**.

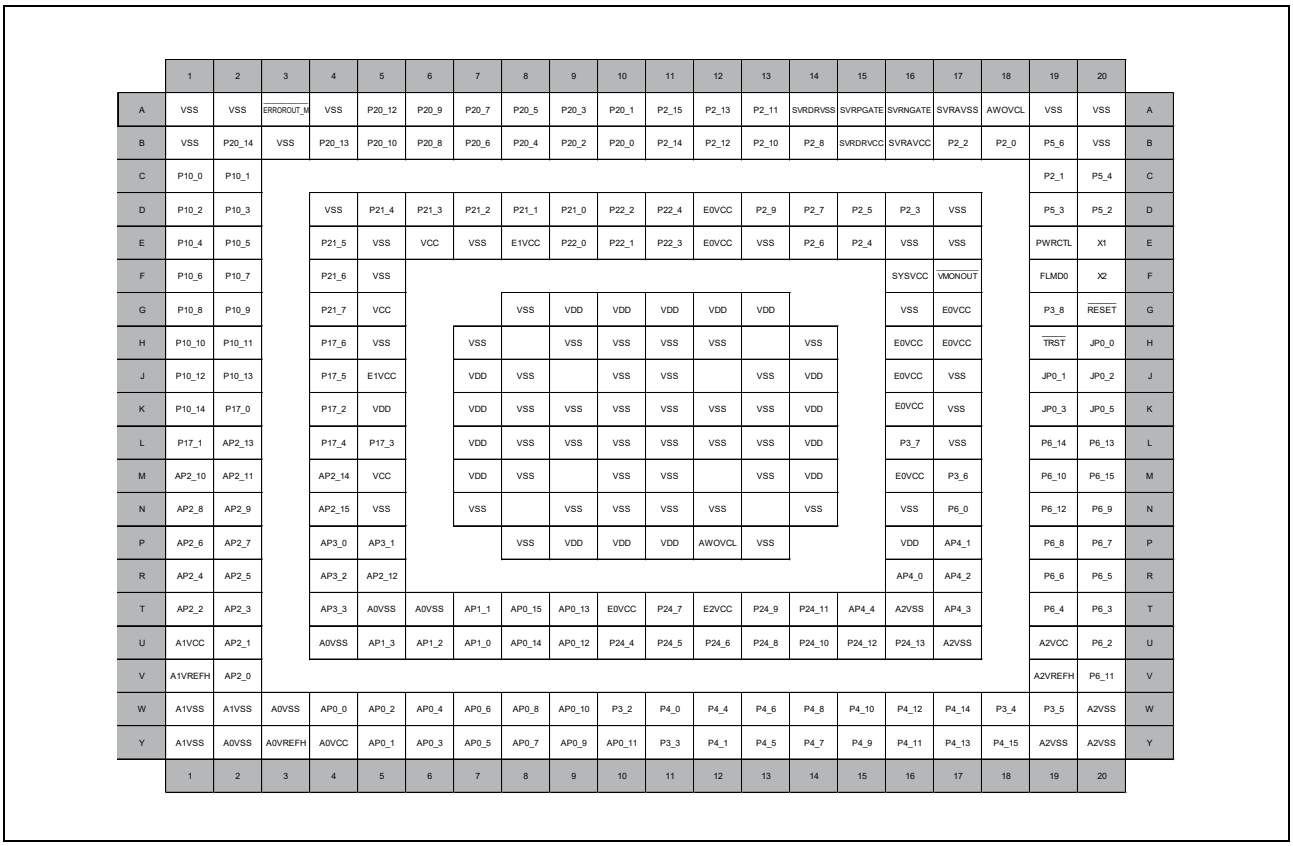
Figure 1.3 Pin Connection Diagram (BGA373 U2A16/U2A8)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	VSS	VSS	ERROROUT_M	VSS	P20_12	P20_9	P20_7	P20_5	P20_3	P20_1	P2_15	P2_13	P2_11	SVRDRVSS	SVRPGATE	SVRNGATE	SVRAVSS	AWOVCL	VSS	VSS	A
B	VSS	P20_14	VSS	P20_13	P20_10	P20_8	P20_6	P20_4	P20_2	P20_0	P2_14	P2_12	P2_10	P2_8	SVRDRVCC	SVRAVCC	P2_2	P2_0	P5_6	VSS	B
C	P10_0	P10_1																	P2_1	P5_4	C
D	P10_2	P10_3		VSS	P21_4	P21_3	P21_2	P21_1	P21_0	P22_2	P22_4	LVDVCC	P2_9	P2_7	P2_5	P2_3	VSS		P5_3	P5_2	D
E	P10_4	P10_5		P21_5	VSS	VCC	VSS	E1VCC	P22_0	P22_1	P22_3	E0VCC	VSS	P2_6	P2_4	VSS	GETH0VCL		PWRCTL	X1	E
F	P10_6	P10_7		P21_6	VSS											SYSVCC	VMONOUT		FLMD0	X2	F
G	P10_8	P10_9		P21_7	VCC			VSS	VDD	VDD	VDD	VDD	VDD			VSS	GETH0 PVCC		P3_8	RESET	G
H	P10_10	P10_11		P17_6	VSS		VSS		VSS	VSS	VSS	VSS		VSS		TX_DATAP	GETH0 RVCC		TRST	JP0_0	H
J	P10_12	P10_13		P17_5	E1VCC		VDD	VSS		VSS	VSS		VSS	VDD		TX_DATAN	RX_DATAP		JP0_1	JP0_2	J
K	P10_14	P17_0		P17_2	VDD		VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD		GETH0 BVCC	RX_DATAN		JP0_3	JP0_5	K
L	P17_1	AP2_13		P17_4	P17_3		VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD		P3_7	VSS		P6_14	P6_13	L
M	AP2_10	AP2_11		AP2_14	VCC		VDD	VSS		VSS	VSS		VSS	VDD		E0VCC	P3_6		P6_10	P6_15	M
N	AP2_8	AP2_9		AP2_15	VSS		VSS		VSS	VSS	VSS	VSS		VSS		VSS	P6_0		P6_12	P6_9	N
P	AP2_6	AP2_7		AP3_0	AP3_1			VSS	VDD	VDD	VDD	AWOVCL	VSS			VDD	AP4_1		P6_8	P6_7	P
R	AP2_4	AP2_5		AP3_2	AP2_12											AP4_0	AP4_2		P6_6	P6_5	R
T	AP2_2	AP2_3		AP3_3	A0VSS	A0VSS	AP1_1	AP0_15	AP0_13	E0VCC	P24_7	E2VCC	P24_9	P24_11	AP4_4	A2VSS	AP4_3		P6_4	P6_3	T
U	A1VCC	AP2_1		A0VSS	AP1_3	AP1_2	AP1_0	AP0_14	AP0_12	P24_4	P24_5	P24_6	P24_8	P24_10	P24_12	P24_13	A2VSS		A2VCC	P6_2	U
V	A1VREFH	AP2_0																	A2VREFH	P6_11	V
W	A1VSS	A1VSS	A0VSS	AP0_0	AP0_2	AP0_4	AP0_6	AP0_8	AP0_10	P3_2	P4_0	P4_4	P4_6	P4_8	P4_10	P4_12	P4_14	P3_4	P3_5	A2VSS	W
Y	A1VSS	A0VSS	A0VREFH	A0VCC	AP0_1	AP0_3	AP0_5	AP0_7	AP0_9	AP0_11	P3_3	P4_1	P4_5	P4_7	P4_9	P4_11	P4_13	P4_15	A2VSS	A2VSS	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

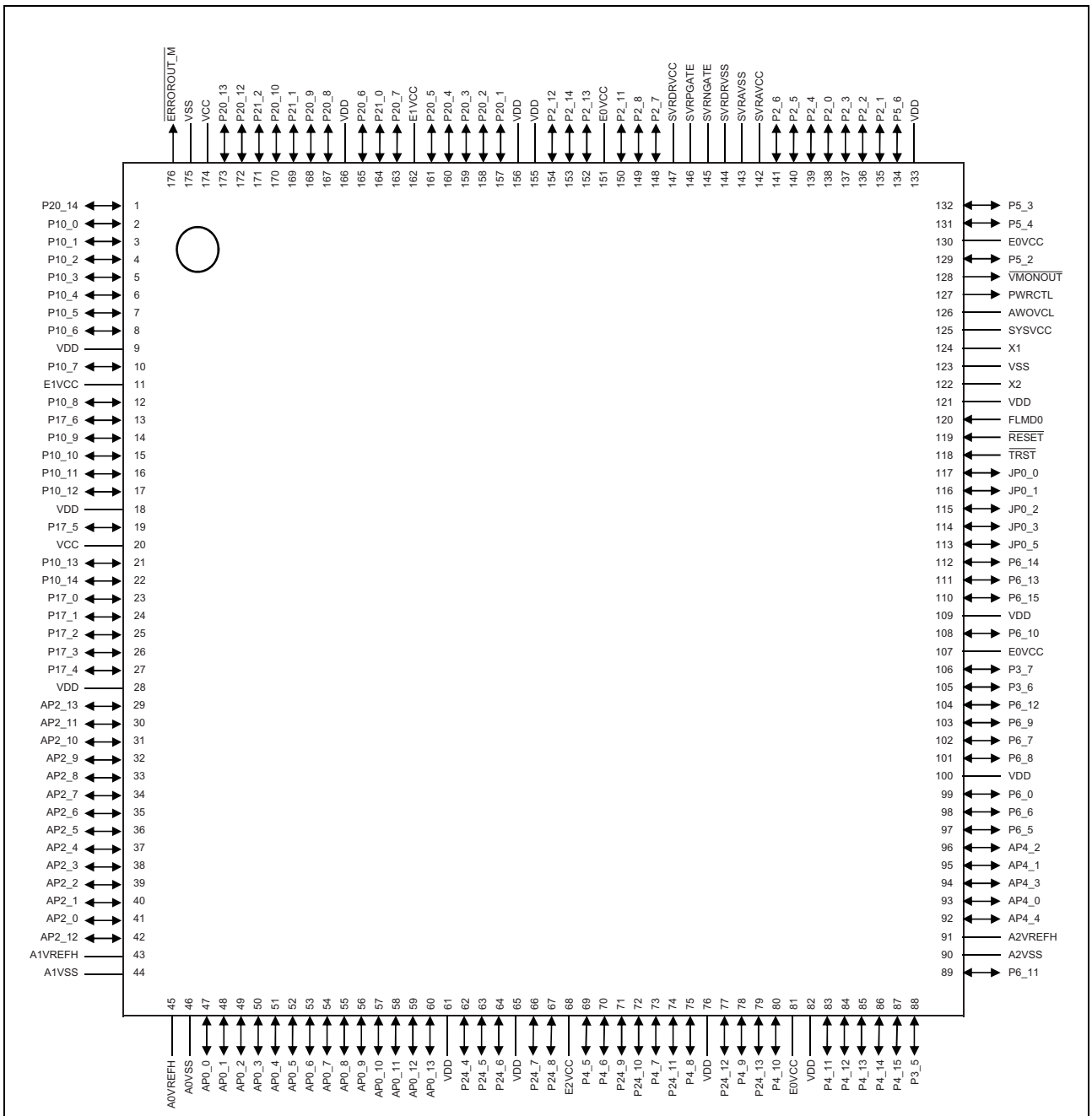
Note: The pin location of RX_DATAP, RX_DATAN, TX_DATAP and TX_DATAN is defined by the Option Byte 14. For details, refer to **Section 51.12.20, OPBT14 — Option Byte 14**.

Figure 1.4 Pin Connection Diagram (BGA292 U2A16/U2A8)



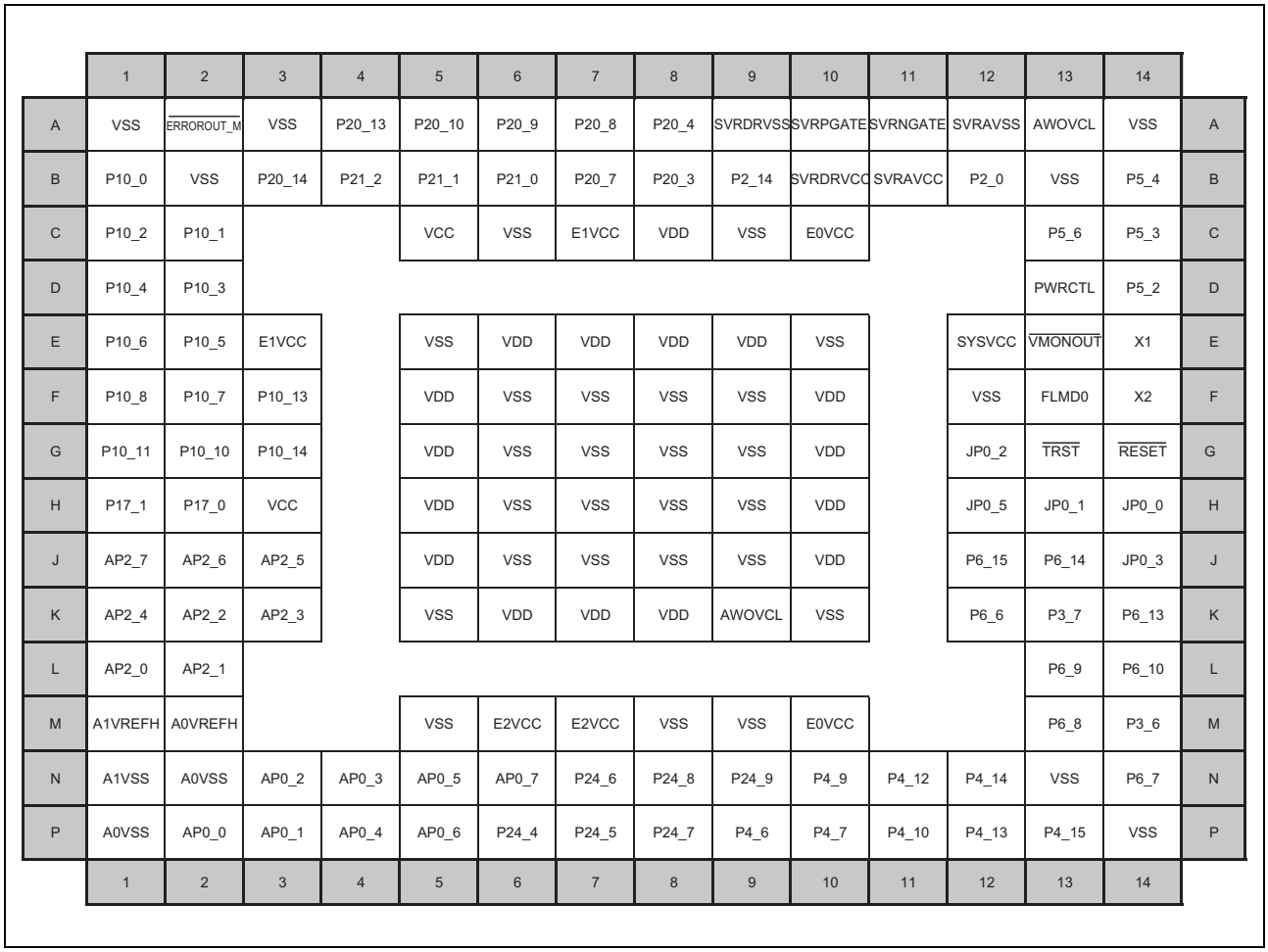
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A	VSS	VSS	ERROROUT_M	VSS	P20_12	P20_9	P20_7	P20_5	P20_3	P20_1	P2_15	P2_13	P2_11	SVRDRVSS	SVRPGATE	SVRNGATE	SVRAVSS	AWOVCL	VSS	VSS	A
B	VSS	P20_14	VSS	P20_13	P20_10	P20_8	P20_6	P20_4	P20_2	P20_0	P2_14	P2_12	P2_10	P2_8	SVRDRVCC	SVRAVCC	P2_2	P2_0	P5_6	VSS	B
C	P10_0	P10_1																	P2_1	P5_4	C
D	P10_2	P10_3		VSS	P21_4	P21_3	P21_2	P21_1	P21_0	P22_2	P22_4	E0VCC	P2_9	P2_7	P2_5	P2_3	VSS		P5_3	P5_2	D
E	P10_4	P10_5		P21_5	VSS	VCC	VSS	E1VCC	P22_0	P22_1	P22_3	E0VCC	VSS	P2_6	P2_4	VSS	VSS		PWRCTL	X1	E
F	P10_6	P10_7		P21_6	VSS											SYSVCC	VMONOUT		FLMD0	X2	F
G	P10_8	P10_9		P21_7	VCC			VSS	VDD	VDD	VDD	VDD	VDD			VSS	E0VCC		P3_8	RESET	G
H	P10_10	P10_11		P17_6	VSS		VSS		VSS	VSS	VSS	VSS		VSS		E0VCC	E0VCC		TRST	JP0_0	H
J	P10_12	P10_13		P17_5	E1VCC		VDD	VSS		VSS	VSS		VSS	VDD		E0VCC	VSS		JP0_1	JP0_2	J
K	P10_14	P17_0		P17_2	VDD		VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD		E0VCC	VSS		JP0_3	JP0_5	K
L	P17_1	AP2_13		P17_4	P17_3		VDD	VSS	VSS	VSS	VSS	VSS	VSS	VDD		P3_7	VSS		P6_14	P6_13	L
M	AP2_10	AP2_11		AP2_14	VCC		VDD	VSS		VSS	VSS		VSS	VDD		E0VCC	P3_6		P6_10	P6_15	M
N	AP2_8	AP2_9		AP2_15	VSS		VSS		VSS	VSS	VSS	VSS		VSS		VSS	P6_0		P6_12	P6_9	N
P	AP2_6	AP2_7		AP3_0	AP3_1			VSS	VDD	VDD	VDD	AWOVCL	VSS			VDD	AP4_1		P6_8	P6_7	P
R	AP2_4	AP2_5		AP3_2	AP2_12											AP4_0	AP4_2		P6_6	P6_5	R
T	AP2_2	AP2_3		AP3_3	A0VSS	A0VSS	AP1_1	AP0_15	AP0_13	E0VCC	P24_7	E2VCC	P24_9	P24_11	AP4_4	A2VSS	AP4_3		P6_4	P6_3	T
U	A1VCC	AP2_1		A0VSS	AP1_3	AP1_2	AP1_0	AP0_14	AP0_12	P24_4	P24_5	P24_6	P24_8	P24_10	P24_12	P24_13	A2VSS		A2VCC	P6_2	U
V	A1VREFH	AP2_0																	A2VREFH	P6_11	V
W	A1VSS	A1VSS	A0VSS	AP0_0	AP0_2	AP0_4	AP0_6	AP0_8	AP0_10	P3_2	P4_0	P4_4	P4_6	P4_8	P4_10	P4_12	P4_14	P3_4	P3_5	A2VSS	W
Y	A1VSS	A0VSS	A0VREFH	A0VCC	AP0_1	AP0_3	AP0_5	AP0_7	AP0_9	AP0_11	P3_3	P4_1	P4_5	P4_7	P4_9	P4_11	P4_13	P4_15	A2VSS	A2VSS	Y
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

Figure 1.5 Pin Connection Diagram (BGA292 U2A6)



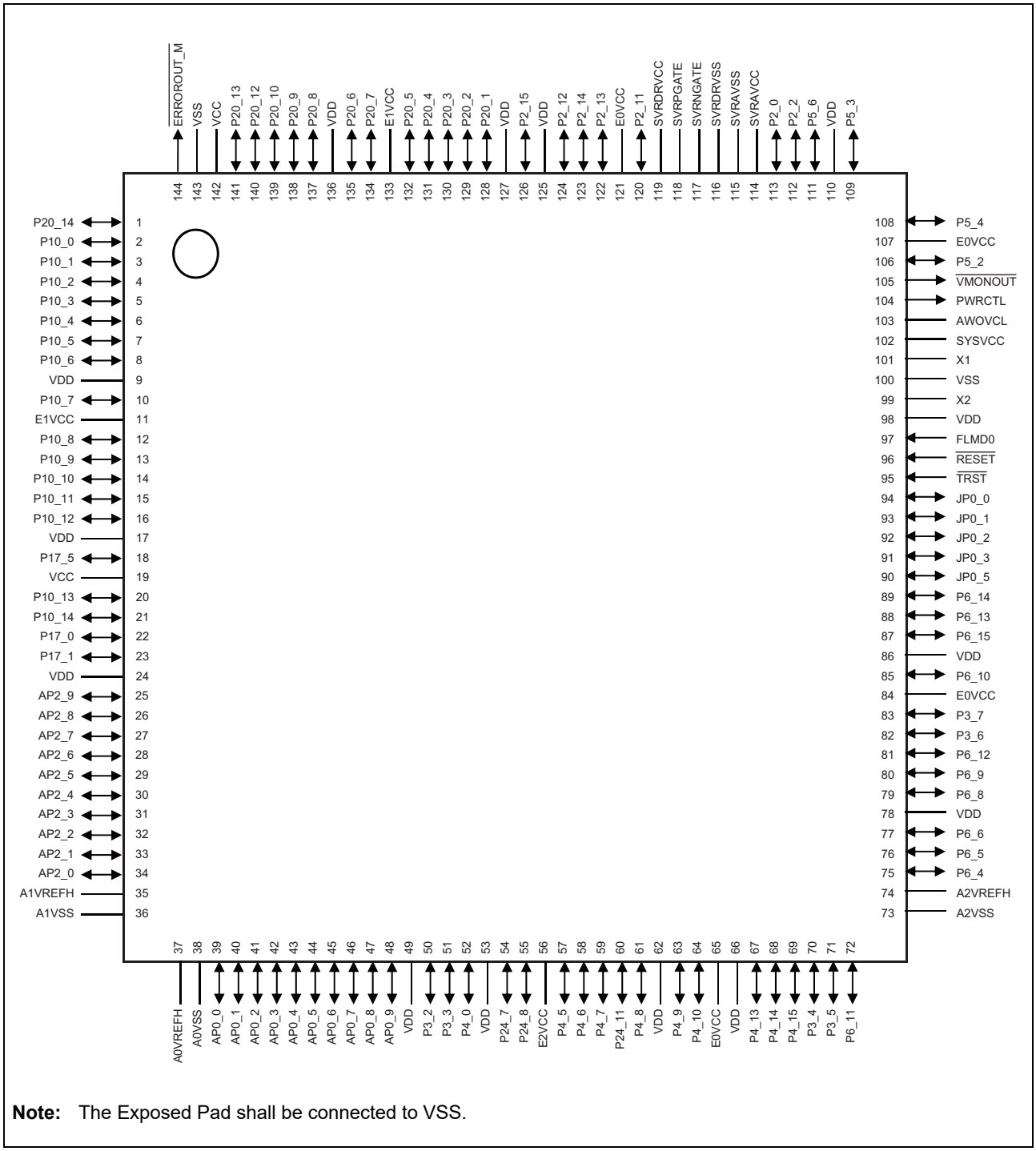
Note: The Exposed Pad shall be connected to VSS.

Figure 1.6 Pin Connection Diagram (QFP176 U2A6)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
A	VSS	$\overline{\text{ERROROUT\_M}}$	VSS	P20_13	P20_10	P20_9	P20_8	P20_4	SVRDRVSS	SVRPGATE	SVRNGATE	SVRAVSS	AWOVCL	VSS	A
B	P10_0	VSS	P20_14	P21_2	P21_1	P21_0	P20_7	P20_3	P2_14	SVRDRVCC	SVRAVCC	P2_0	VSS	P5_4	B
C	P10_2	P10_1			VCC	VSS	E1VCC	VDD	VSS	E0VCC			P5_6	P5_3	C
D	P10_4	P10_3											PWRCTL	P5_2	D
E	P10_6	P10_5	E1VCC		VSS	VDD	VDD	VDD	VDD	VSS		SYSVCC	$\overline{\text{VMONOUT}}$	X1	E
F	P10_8	P10_7	P10_13		VDD	VSS	VSS	VSS	VSS	VDD		VSS	FLMD0	X2	F
G	P10_11	P10_10	P10_14		VDD	VSS	VSS	VSS	VSS	VDD		JP0_2	$\overline{\text{TRST}}$	$\overline{\text{RESET}}$	G
H	P17_1	P17_0	VCC		VDD	VSS	VSS	VSS	VSS	VDD		JP0_5	JP0_1	JP0_0	H
J	AP2_7	AP2_6	AP2_5		VDD	VSS	VSS	VSS	VSS	VDD		P6_15	P6_14	JP0_3	J
K	AP2_4	AP2_2	AP2_3		VSS	VDD	VDD	VDD	AWOVCL	VSS		P6_6	P3_7	P6_13	K
L	AP2_0	AP2_1											P6_9	P6_10	L
M	A1VREFH	A0VREFH			VSS	E2VCC	E2VCC	VSS	VSS	E0VCC			P6_8	P3_6	M
N	A1VSS	A0VSS	AP0_2	AP0_3	AP0_5	AP0_7	P24_6	P24_8	P24_9	P4_9	P4_12	P4_14	VSS	P6_7	N
P	A0VSS	AP0_0	AP0_1	AP0_4	AP0_6	P24_4	P24_5	P24_7	P4_6	P4_7	P4_10	P4_13	P4_15	VSS	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 1.7 Pin Connection Diagram (BGA156 U2A6)



Note: The Exposed Pad shall be connected to VSS.

Figure 1.8 Pin Connection Diagram (QFP144 U2A6)

1.7 Functional Block Configuration

Internal Block Diagram

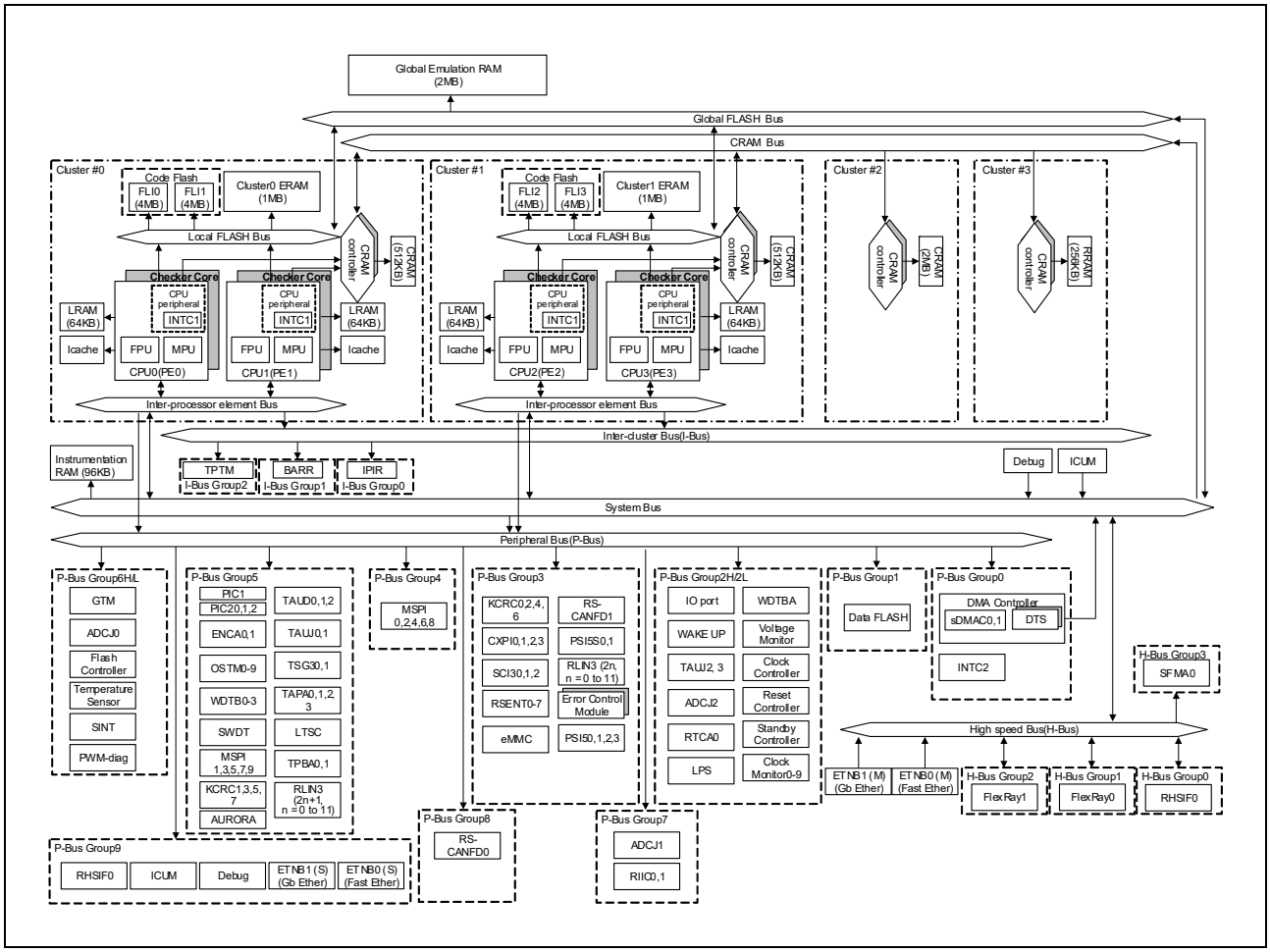


Figure 1.9 Internal Block Diagram (U2A-EVA 516 pins)

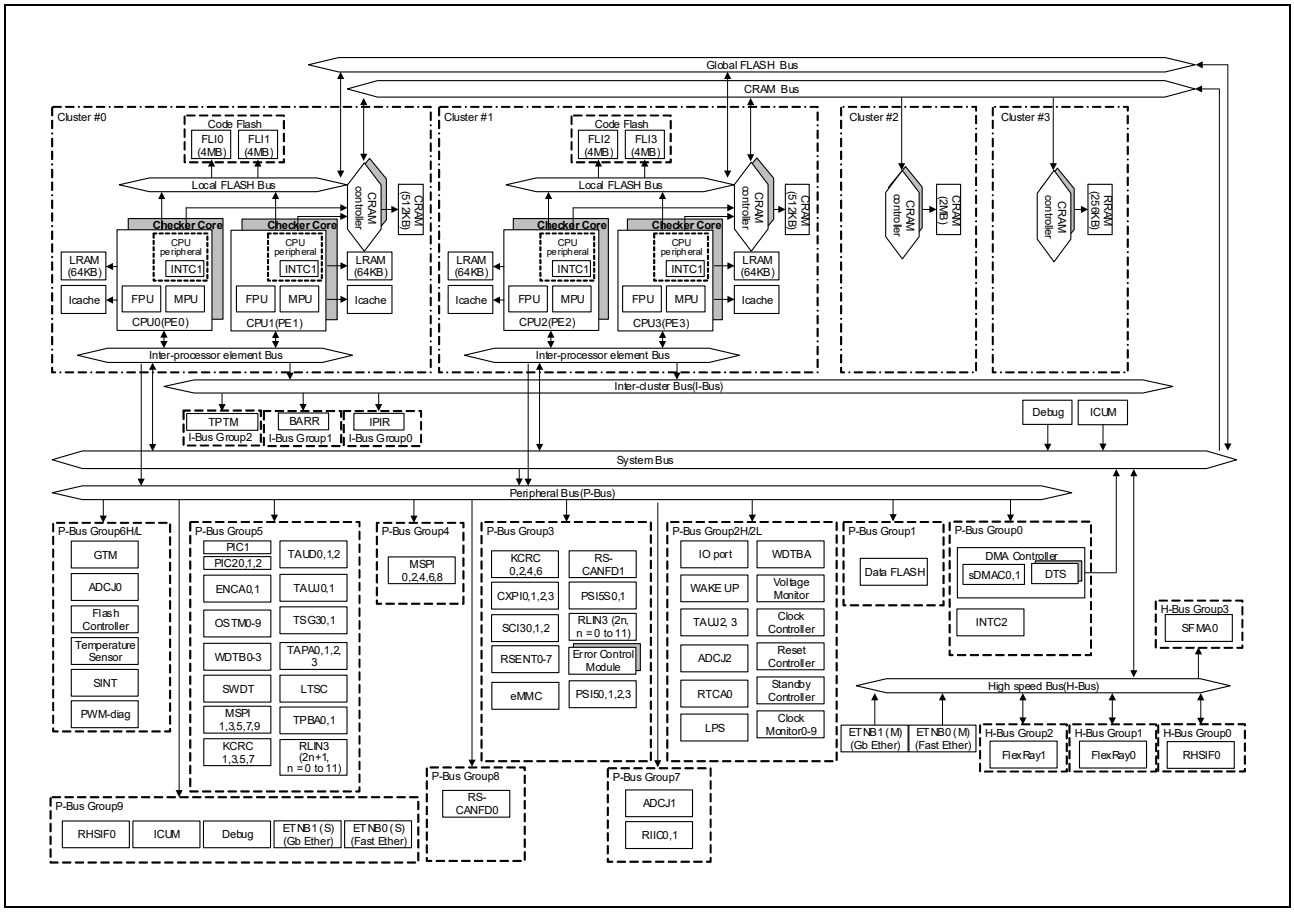


Figure 1.10 Internal Block Diagram (U2A16 516 pins)

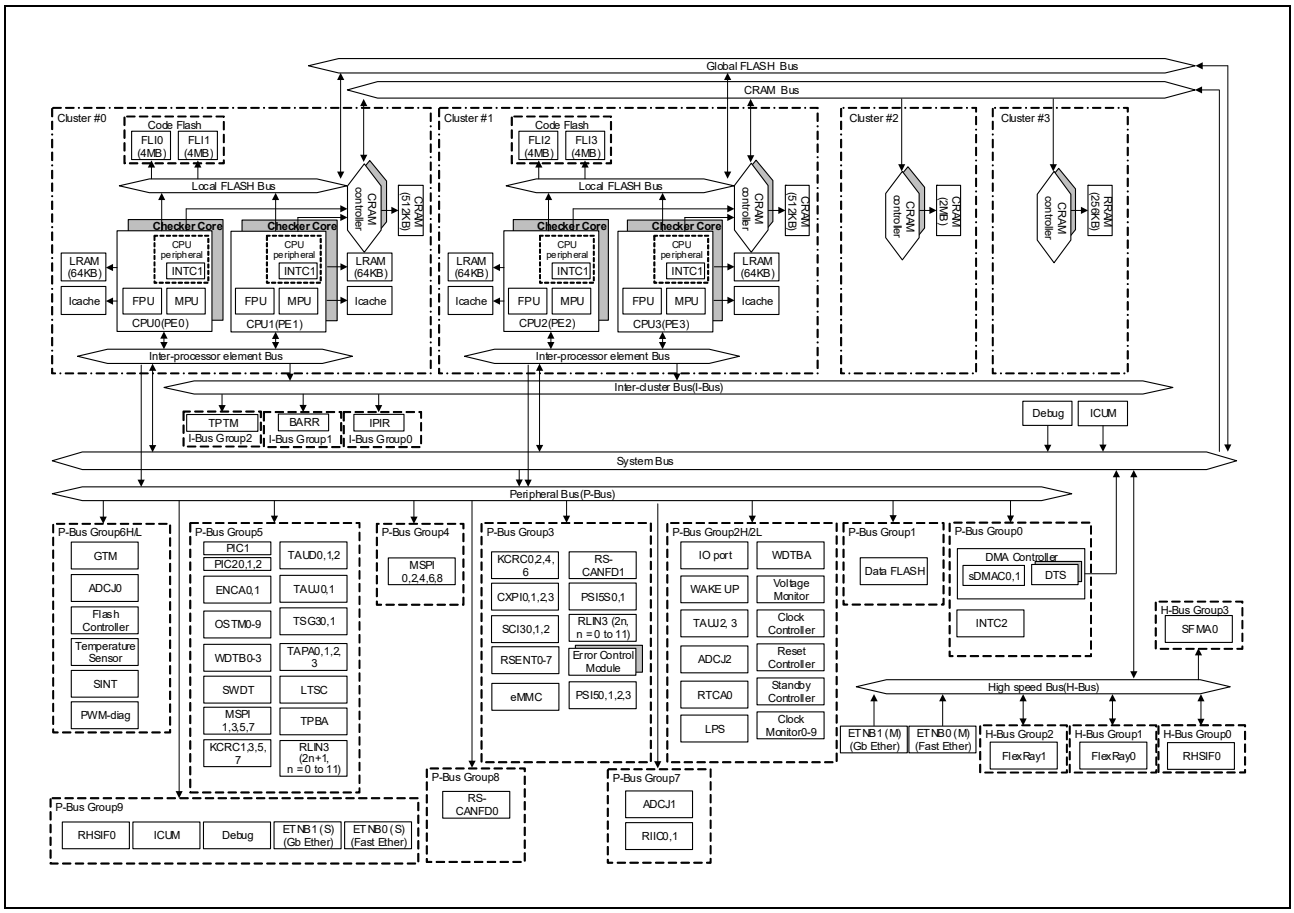


Figure 1.11 Internal Block Diagram (U2A16 373 pins)

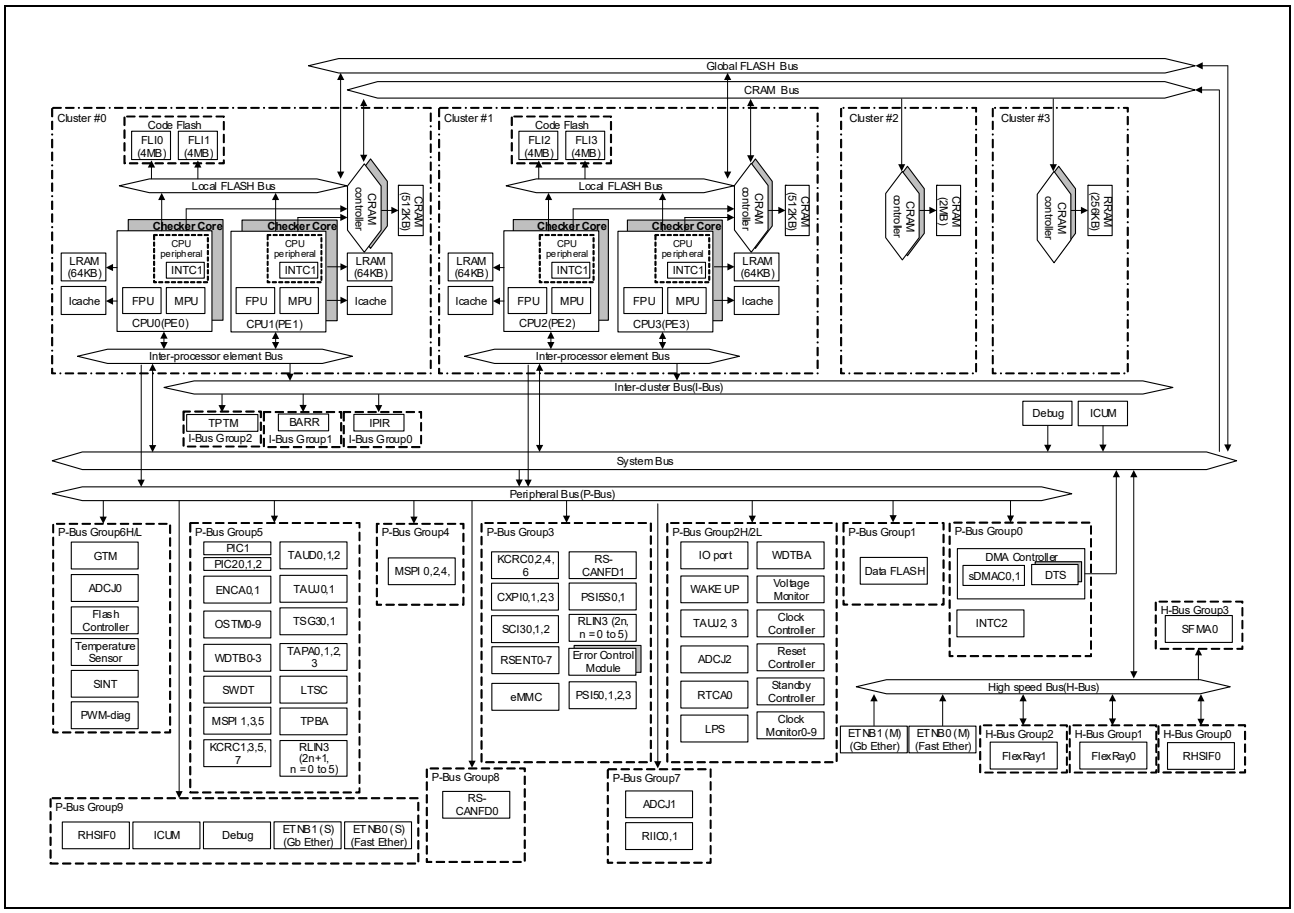


Figure 1.12 Internal Block Diagram (U2A16 292 pins)

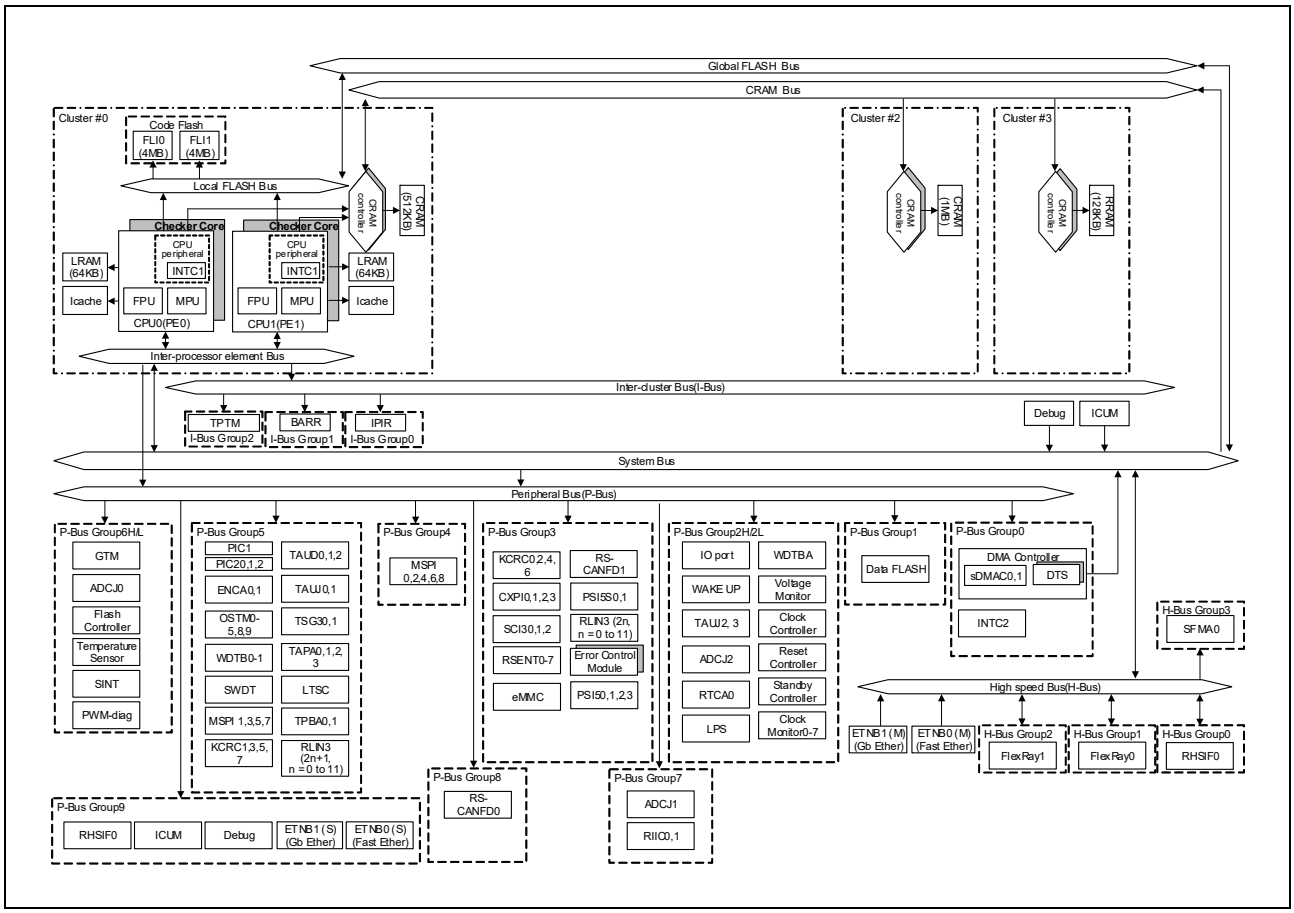


Figure 1.13 Internal Block Diagram (U2A8 373 pins)

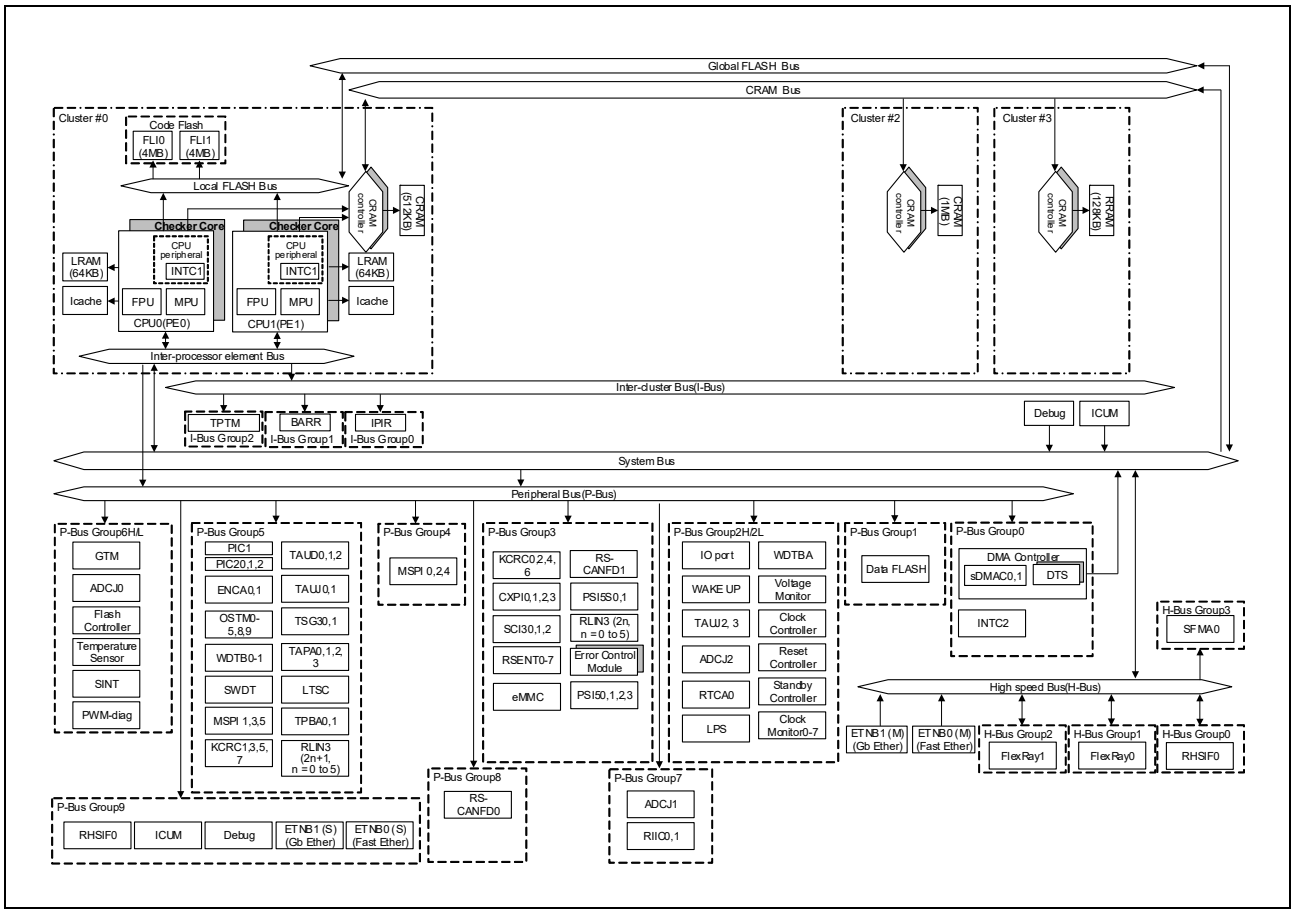


Figure 1.14 Internal Block Diagram (U2A8 292 pins)

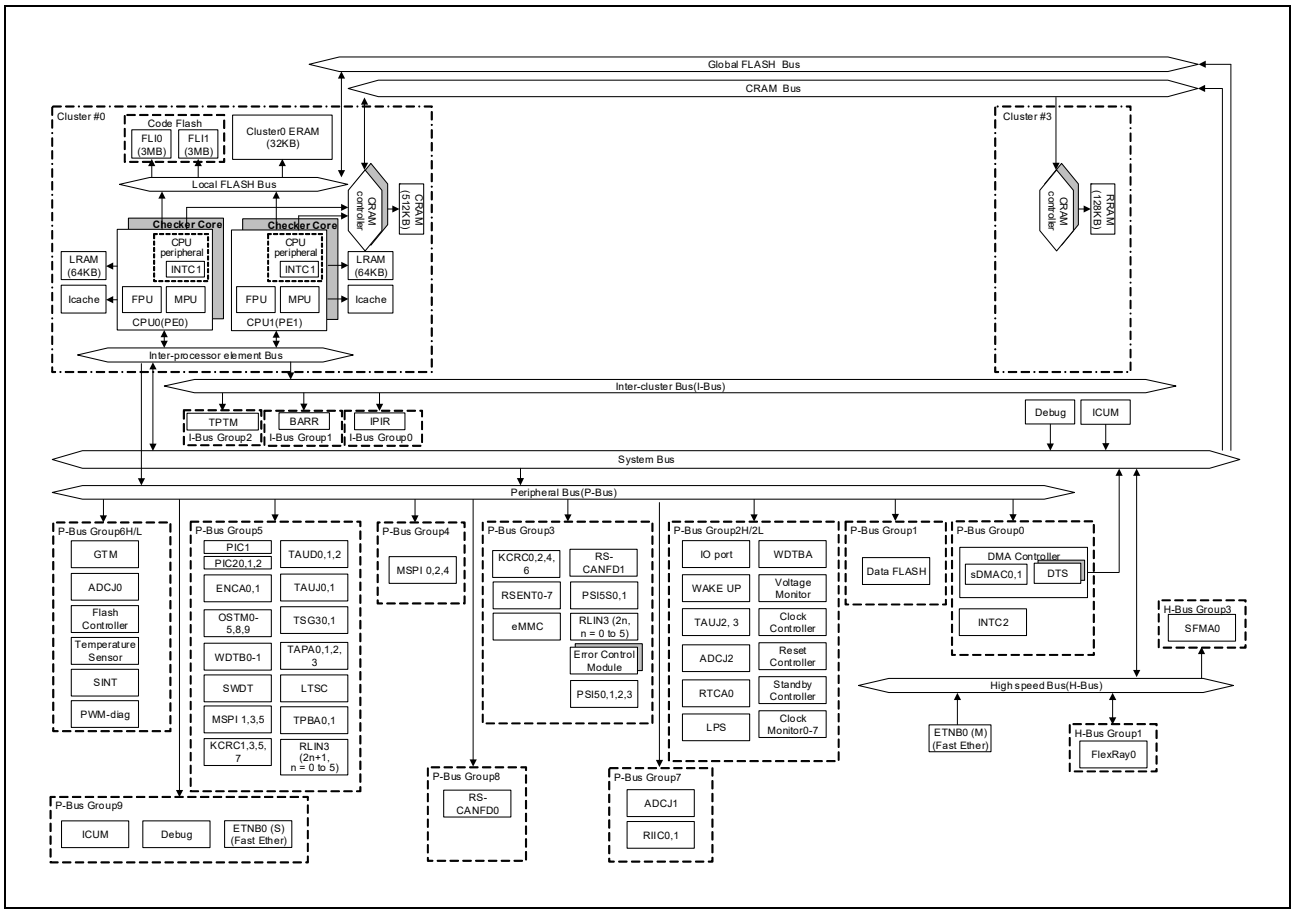


Figure 1.15 Internal Block Diagram (U2A6 292 pins)

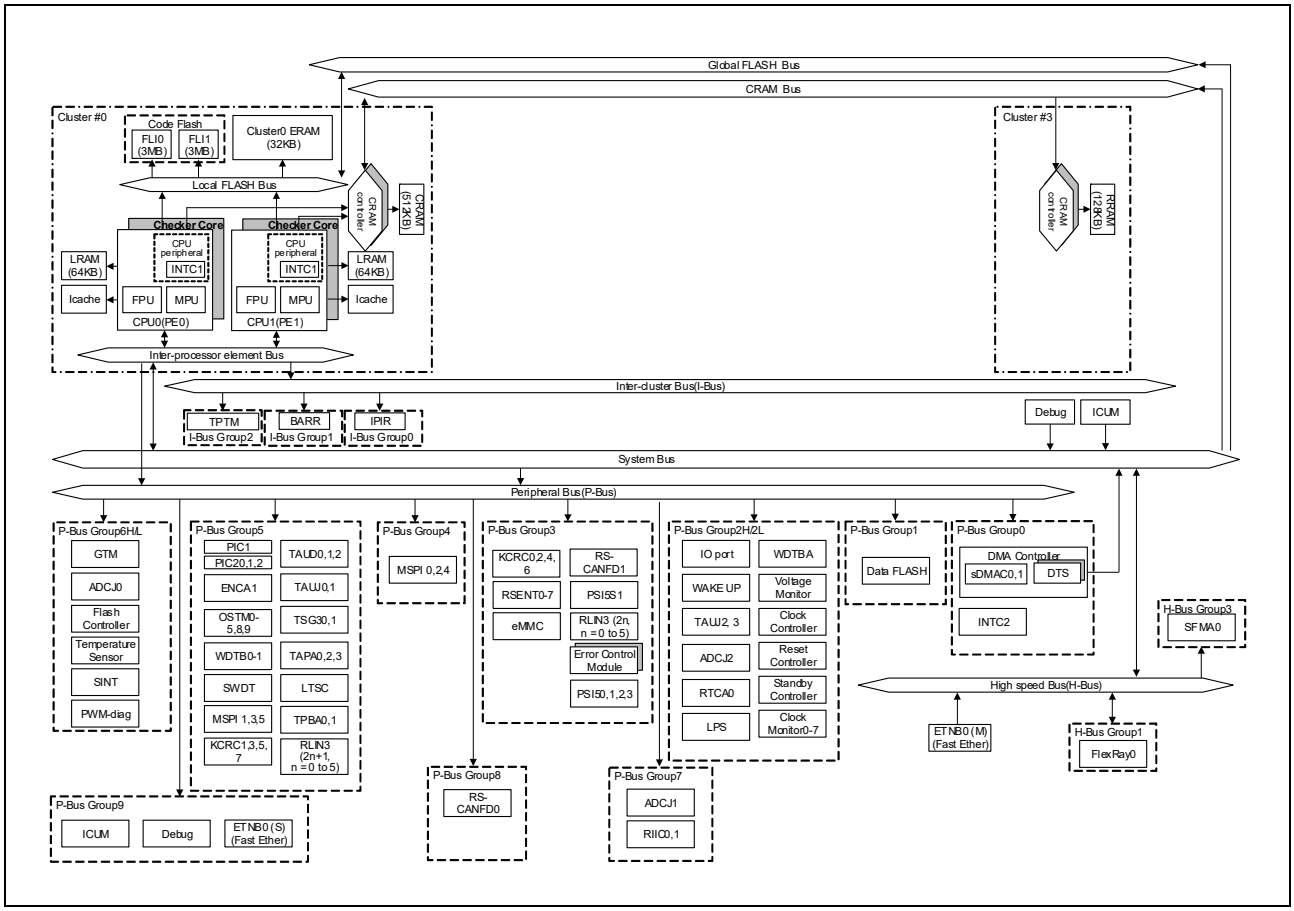


Figure 1.16 Internal Block Diagram (U2A6 176 pins)

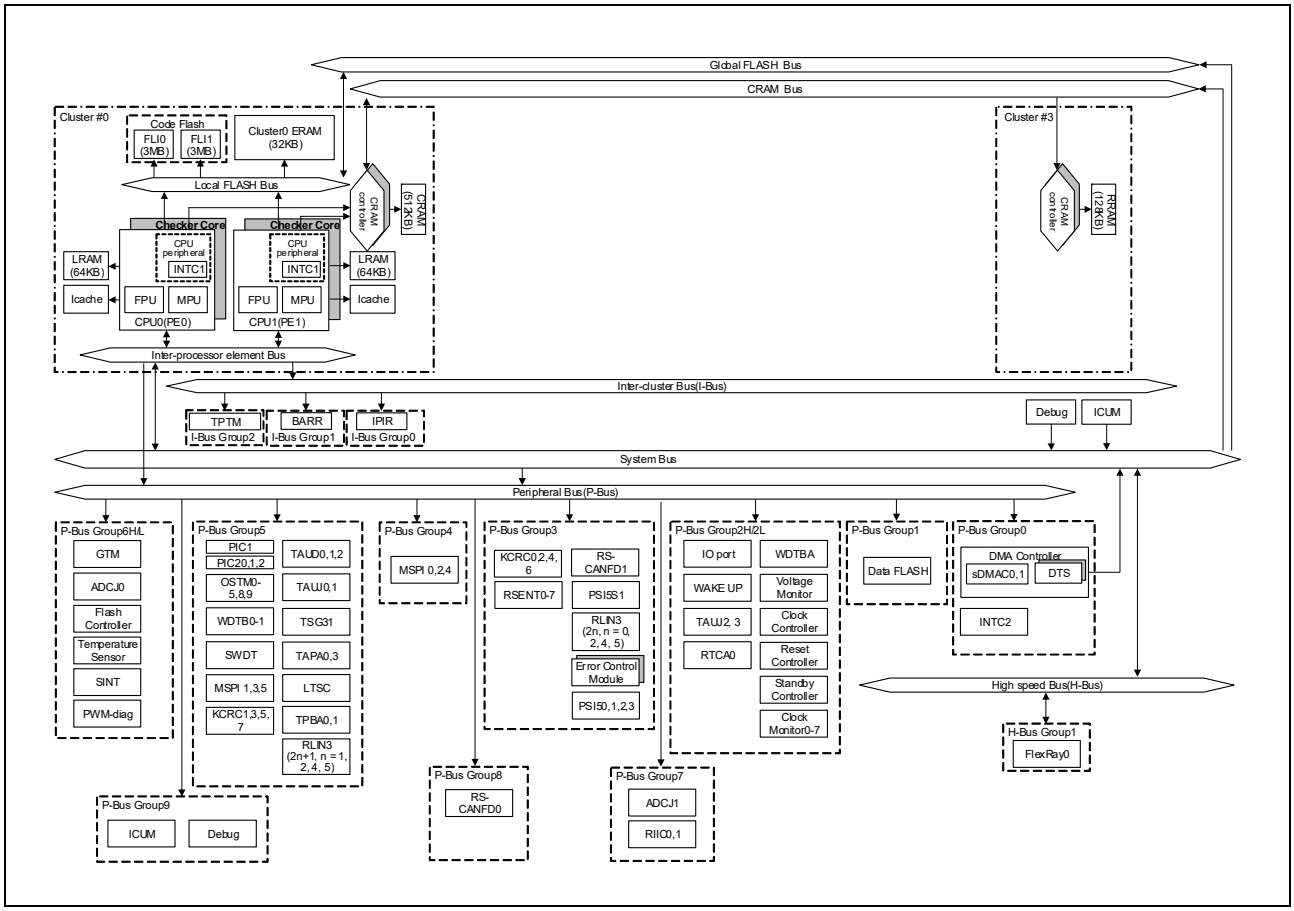


Figure 1.17 Internal Block Diagram (U2A6 156 pins)

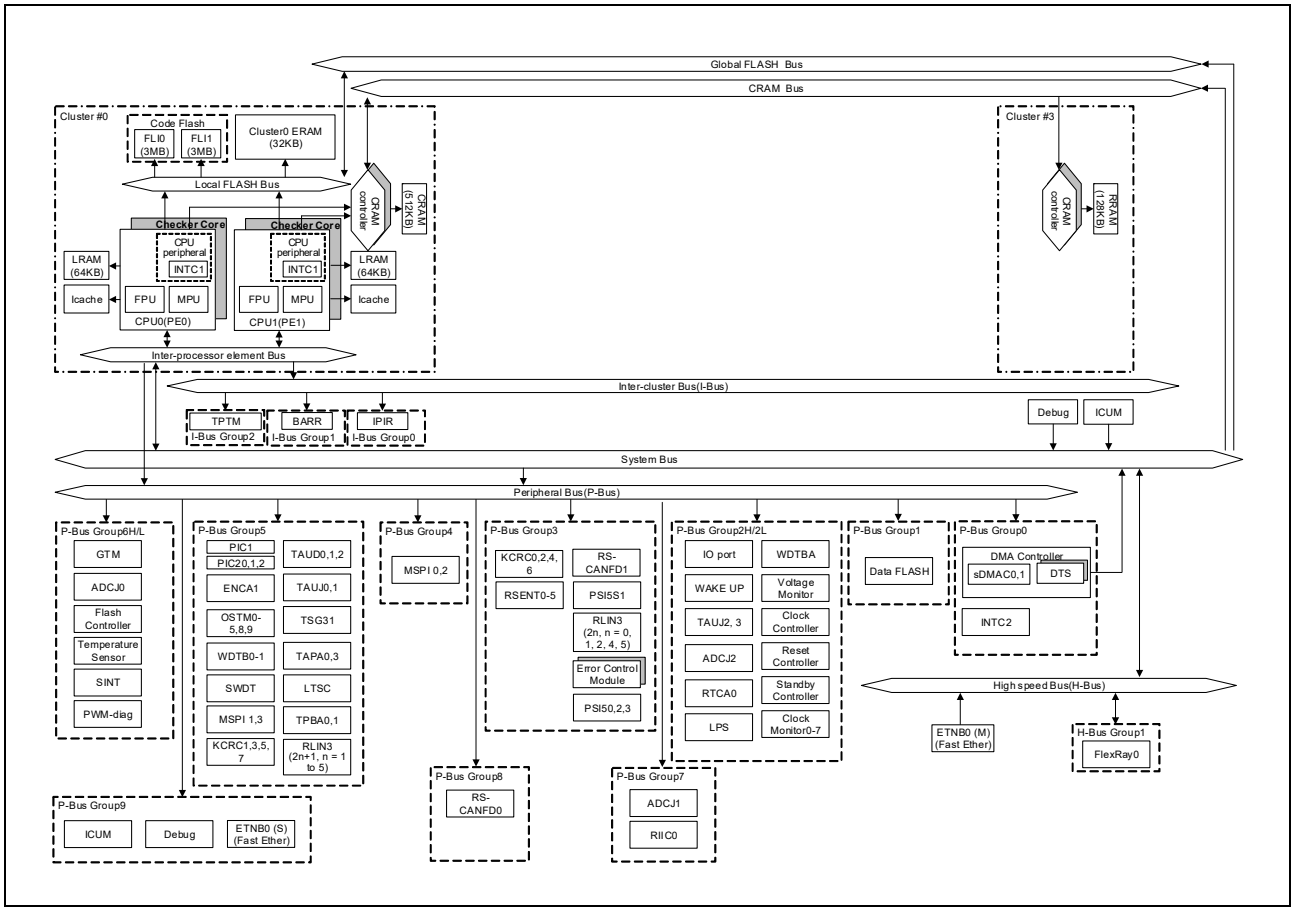


Figure 1.18 Internal Block Diagram (U2A6 144 pins)

Each CPU has its own set of CPU peripherals. These CPU peripherals are assigned to the common address: CPU peripheral (self), separately from their respective address. For example, access by CPU0 to a register of a CPU peripheral (self) is to the register in the CPU peripheral of CPU0; access by CPU1 to a register of a CPU peripheral (self) is to the register in the CPU peripheral of CPU1.

Section 2 Pin Functions

This section describes the pin connections and respective pin functions.

2.1 Term Definition

The following terms are used in this section:

Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

This product provides the following port groups, indicated by the numbers in the table below.

Table 2.1 Port Group

Package	BGA516	BGA373	BGA292	QFP176	BGA156	QFP144
Number of port group	27	21	17	14	13	12
Port Group Name	JP0, P0 to P6, P8 to P12, P17 to P24, AP0 to AP5	JP0, P0, P2 to P6, P8, P10, P11, P17, P20 to P22, P24, AP0 to AP5	JP0, P2 to P6, P10, P17, P20 to P22, P24, AP0 to AP4	JP0, P2 to P6, P10, P17, P20, P21, P24, AP0, AP2, AP4	JP0, P2 to P6, P10, P17, P20, P21, P24, AP0, AP2	JP0, P2 to P6, P10, P17, P20, P24, AP0, AP2

Port group index n

Each port group is identified by its own index “n” throughout this section; e.g.

PMCn for the port mode control register of the Pn port.

Port mode and ports

A pin in port mode works as a general purpose input/output pin. It is then called “port”.

The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions.

It is such as MSPI and INTP.

SHMT1/SHMT4/TTL

Denotes input buffer types. Each types have a different DC characteristics.

For details, refer to **Section 55, Electrical Characteristics**.

2.2 Pin List

2.2.1 Pin List and Function assignment

For detailed information, refer to Appendix “E02_01_List_of_Pin_Assignment.xlsx”.

2.2.2 Pin Status

Table 2.2 Pin Status (1/3)

Pin Function		Pin Status					
			RESET = H				
Category	Pin Name	RESET = L	During Internal Reset	FBIST0 *1	After Internal Reset	DeepStop mode	FBIST1/ FBIST2*1
System Control	X1	I	I	I	I	I	I
	X2	O	O	O	O	O	O
	RESET	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
	P6_10 (RESETOUT)	O	O	O	O	O	O
	FLMD0	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
	P6_13 (FLMD1)	I *6	Z	Z	Z	Z	Z
	P6_14 (FLMD2)	I *7	Z	Z	Z	Z	Z
	P5_3 (MODE0)	I *8	Z	Z	Z	Z *3	Z *3
	PWRCTL	O	O	O	O	O	O
	SVRPGATE	O	O	O	O	O	O
	SVRNGATE	O	O	O	O	O	O
	VMONOUT	O	O	O	O	O	O
ECM	ERROROUT_M	Z	O	O	O	Z	O
PORT	Pn_m (n=0,1,6,8) excluding P6_10, P6_13, P6_14	Z	Z	Z	Z	Z	Z
	Pn_m (n=2-5,9-12,17-24) excluding P5_3	Z	Z	Z	Z	Z *3*4	Z *3*4
	APn_m (n=0-3)	Z	Z	Z	Z	Z *3	Z *3
	APn_m (n=4-5)	Z	Z	Z	Z	Z	Z
ADC	AnVREFH (n=0-2)	I	I	I	I	I	I
SGMII	RX_CLKN	Z	Z	Z	Z	Z	Z
	RX_CLKP	Z	Z	Z	Z	Z	Z
	RX_DATAN	Z	Z	Z	Z	Z	Z
	RX_DATAP	Z	Z	Z	Z	Z	Z
	TX_DATAN	Z	Z	Z	Z	Z	Z
	TX_DATAP	Z	Z	Z	Z	Z	Z
AUDR	AUDRST	I (pull-down)	I (pull-down)	pull-down	I (pull-down)	I (pull-down)	pull-down
	AUDCK	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	pull-up
	AUDSYNC	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	pull-up
	AUDATAn (n=0-3)	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	pull-up

Table 2.2 Pin Status (2/3)

Pin Function			Pin Status					
			$\overline{\text{RESET}}$ = L	$\overline{\text{RESET}}$ = H				
Category		Pin Name	$\overline{\text{RESET}}$ = L	During Internal Reset	FBIST0 *1	After Internal Reset	DeepStop mode	FBIST1/ FBIST2*1
Aurora		CICREFP	Z	Z	Z	Z	Z	Z
		CICREFN	Z	Z	Z	Z	Z	Z
		TODP0	Z	Z	Z	Z	Z	Z
		TODN0	Z	Z	Z	Z	Z	Z
		TODP1	Z	Z	Z	Z	Z	Z
		TODN1	Z	Z	Z	Z	Z	Z
		TODP2	Z	Z	Z	Z	Z	Z
		TODN2	Z	Z	Z	Z	Z	Z
		TODP3	Z	Z	Z	Z	Z	Z
		TODN3	Z	Z	Z	Z	Z	Z
		$\overline{\text{AURORES1}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{AURORESPD}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{AURORES2}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{MSYN}}$	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)
ERAM		$\overline{\text{ERAMRESPD}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		$\overline{\text{ERAMRES2}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
Debug	Nexus (without $\overline{\text{RDY}}$)	JP0_0	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_1	Z	Z	—	Z	Z	—
		JP0_2	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_3	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	—	I (pull-down)	I (pull-down)	—
		JP0_5	Z	Z	—	Z	Z	—
	Nexus (with $\overline{\text{RDY}}$)	JP0_0	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_1	Z	Z	—	Z	Z	—
		JP0_2	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_3	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	—	I (pull-down)	I (pull-down)	—
		JP0_5	Z *5	Z *5	—	O	O	—
Debug	LPD4	JP0_0	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_1	Z *5	Z*5	—	O	O	—
		JP0_2	I (pull-up)	I (pull-up)	—	I (pull-up)	I (pull-up)	—
		JP0_3	Z	Z	—	Z	Z	—
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	—	I (pull-down)	I (pull-down)	—
		JP0_5	Z *5	Z *5	—	O	O	—
	GPIO	JP0_0	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)
		JP0_1	Z	Z	Z	Z	Z	Z
		JP0_2	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	I (pull-up)
		JP0_3	Z	Z	Z	Z	Z	Z
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)	I (pull-down)
		JP0_5	Z	Z	Z	Z	Z	Z

Table 2.2 Pin Status (3/3)

Pin Function			Pin Status					
				$\overline{\text{RESET}}$ = H				
Category		Pin Name	$\overline{\text{RESET}}$ = L	During Internal Reset	FBIST0 *1	After Internal Reset	DeepStop mode	FBIST1/ FBIST2*1
Debug	BSCAN	JP0_0	I (pull-up)	I (pull-up)	—	I (pull-up)	—	—
		JP0_1	Z	Z	—	Z	—	—
		JP0_2	I (pull-up)	I (pull-up)	—	I (pull-up)	—	—
		JP0_3	I (pull-up)	I (pull-up)	—	I (pull-up)	—	—
		$\overline{\text{TRST}}$	I (pull-down)	I (pull-down)	—	I (pull-down)	—	—
		JP0_5	Z	Z	—	Z	—	—
	—	$\overline{\text{EVTI}}$	I (pull-up)	I (pull-up)	pull-up	I (pull-up)	I (pull-up)	pull-up
		$\overline{\text{EVTO}}$	O	O	O	O	Z *2	O

Note: I: Input
O: Output
Z: Hi-Z
pull-up: On-chip pull-up resistor
pull-down: On-chip pull-down resistor

Note 1. FBIST0/FBIST1/FBIST2 are executed only when $\overline{\text{TRST}}$ = L

Note 2. If $\overline{\text{TRST}}$ = H, the pin keeps output

Note 3. The pin is in I/O buffer hold state. For details, refer to **Section 15, Standby Controller (STBC)**

Note 4. LVDS input/output function become disable even if LVDSCTRLA or LVDSCTRLB register is set.

Note 5. If reset mask function is enable, the pin keeps output. For details, refer to **Section 50, Debugging and Calibration**.

Note 6. The pin is Hi-Z when FLMD0 = L

Note 7. The pin is Hi-Z when FLMD0 = L or P6_13 (FLMD1) = L

Note 8. The pin is Hi-Z when FLMD0 = L or P6_13 (FLMD1) = L or P6_14 (FLMD2) = L

2.2.3 Pin Function name

Table 2.3 Pin Function Name Definition (1/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	JP0_0	IO	Port of JTAG port group 0	√	√	√	√	√	√	√	√	√	√
	JP0_1	IO		√	√	√	√	√	√	√	√	√	√
	JP0_2	IO		√	√	√	√	√	√	√	√	√	√
	JP0_3	IO		√	√	√	√	√	√	√	√	√	√
	JP0_5	IO		√	√	√	√	√	√	√	√	√	√
	P0_0	IO	Port of port group 0	√	√	√		√					
	P0_1	IO		√	√	√		√					
	P0_2	IO		√	√	√		√					
	P0_3	IO		√	√	√		√					
	P0_4	IO		√	√	√		√					
	P0_5	IO		√	√	√		√					
	P0_6	IO		√	√	√		√					
	P0_7	IO		√	√	√		√					
	P0_8	IO		√	√	√		√					
	P0_9	IO		√	√	√		√					
	P0_10	IO		√	√	√		√					
	P0_11	IO		√	√	√		√					
	P0_12	IO		√	√	√		√					
	P1_0	IO	Port of port group 1	√	√								
	P1_1	IO		√	√								
	P1_2	IO		√	√								
	P1_3	IO		√	√								
	P1_4	IO		√	√								
	P1_5	IO		√	√								
	P1_6	IO		√	√								
	P1_7	IO		√	√								
	P1_8	IO		√	√								
	P1_9	IO		√	√								
	P1_10	IO		√	√								
	P1_11	IO		√	√								
	P1_12	IO		√	√								
	P1_13	IO		√	√								
	P1_14	IO		√	√								
	P1_15	IO		√	√								
	P2_0	IO	Port of port group 2	√	√	√	√	√	√	√	√	√	√
	P2_1	IO		√	√	√	√	√	√	√	√		
	P2_2	IO		√	√	√	√	√	√	√	√		√
	P2_3	IO		√	√	√	√	√	√	√	√		
	P2_4	IO		√	√	√	√	√	√	√	√		
	P2_5	IO		√	√	√	√	√	√	√	√		
	P2_6	IO		√	√	√	√	√	√	√	√		
	P2_7	IO		√	√	√	√	√	√	√	√		
	P2_8	IO		√	√	√	√	√	√	√	√		

Table 2.3 Pin Function Name Definition (2/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	P2_9	IO	Port of port group 2	√	√	√	√	√	√	√			
	P2_10	IO		√	√	√	√	√	√	√			
	P2_11	IO		√	√	√	√	√	√	√	√		√
	P2_12	IO		√	√	√	√	√	√	√	√		√
	P2_13	IO		√	√	√	√	√	√	√	√		√
	P2_14	IO		√	√	√	√	√	√	√	√	√	√
	P2_15	IO		√	√	√	√	√	√	√			√
	P3_0	IO	Port of port group 3	√	√	√		√					
	P3_1	IO		√	√	√		√					
	P3_2	IO		√	√	√	√	√	√	√			√
	P3_3	IO		√	√	√	√	√	√	√			√
	P3_4	IO		√	√	√	√	√	√	√			√
	P3_5	IO		√	√	√	√	√	√	√	√		√
	P3_6	IO		√	√	√	√	√	√	√	√	√	√
	P3_7	IO		√	√	√	√	√	√	√	√	√	√
	P3_8	IO		√	√	√	√	√	√	√			
	P3_9	IO		√	√								
	P3_10	IO		√	√								
	P3_11	IO		√	√								
	P3_12	IO		√	√								
	P3_13	IO		√	√								
	P3_14	IO		√	√								
	P3_15	IO		√	√								
	P4_0	IO	Port of port group 4	√	√	√	√	√	√	√			√
	P4_1	IO		√	√	√	√	√	√	√			
	P4_2	IO		√	√								
	P4_3	IO		√	√								
	P4_4	IO		√	√	√	√	√	√	√			
	P4_5	IO		√	√	√	√	√	√	√	√		√
	P4_6	IO		√	√	√	√	√	√	√	√	√	√
	P4_7	IO		√	√	√	√	√	√	√	√	√	√
	P4_8	IO		√	√	√	√	√	√	√	√		√
	P4_9	IO		√	√	√	√	√	√	√	√	√	√
	P4_10	IO		√	√	√	√	√	√	√	√	√	√
	P4_11	IO		√	√	√	√	√	√	√	√		
	P4_12	IO		√	√	√	√	√	√	√	√	√	
	P4_13	IO		√	√	√	√	√	√	√	√	√	√
	P4_14	IO		√	√	√	√	√	√	√	√	√	√
	P4_15	IO		√	√	√	√	√	√	√	√	√	√
	P5_2	IO	Port of port group 5	√	√	√	√	√	√	√	√	√	√
	P5_3	IO		√	√	√	√	√	√	√	√	√	√
	P5_4	IO		√	√	√	√	√	√	√	√	√	√
	P5_6	IO		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (3/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	P6_0	IO	Port of port group 6	√	√	√	√	√	√	√	√		
	P6_2	IO		√	√	√	√	√	√	√			
	P6_3	IO		√	√	√	√	√	√	√			
	P6_4	IO		√	√	√	√	√	√	√			√
	P6_5	IO		√	√	√	√	√	√	√	√		√
	P6_6	IO		√	√	√	√	√	√	√	√	√	√
	P6_7	IO		√	√	√	√	√	√	√	√	√	
	P6_8	IO		√	√	√	√	√	√	√	√	√	√
	P6_9	IO		√	√	√	√	√	√	√	√	√	√
	P6_10	IO		√	√	√	√	√	√	√	√	√	√
	P6_11	IO		√	√	√	√	√	√	√	√		√
	P6_12	IO		√	√	√	√	√	√	√	√		√
	P6_13	IO		√	√	√	√	√	√	√	√	√	√
	P6_14	IO		√	√	√	√	√	√	√	√	√	√
	P6_15	IO		√	√	√	√	√	√	√	√	√	√
	P8_0	IO	Port of port group 8	√	√	√		√					
	P8_1	IO		√	√	√		√					
	P8_2	IO		√	√	√		√					
	P8_3	IO		√	√	√		√					
	P8_4	IO		√	√	√		√					
	P8_5	IO		√	√	√		√					
	P8_6	IO		√	√	√		√					
	P8_7	IO		√	√	√		√					
	P8_8	IO		√	√	√		√					
	P8_9	IO		√	√	√		√					
	P8_10	IO		√	√								
	P9_0	IO	Port of port group 9	√	√								
	P9_1	IO		√	√								
	P9_2	IO		√	√								
	P9_3	IO		√	√								
	P9_4	IO		√	√								
	P9_5	IO		√	√								
	P9_6	IO		√	√								
	P9_7	IO		√	√								
	P9_8	IO		√	√								
	P10_0	IO	Port of port group 10	√	√	√	√	√	√	√	√	√	√
	P10_1	IO		√	√	√	√	√	√	√	√	√	√
	P10_2	IO		√	√	√	√	√	√	√	√	√	√
	P10_3	IO		√	√	√	√	√	√	√	√	√	√
	P10_4	IO		√	√	√	√	√	√	√	√	√	√
	P10_5	IO		√	√	√	√	√	√	√	√	√	√
	P10_6	IO		√	√	√	√	√	√	√	√	√	√
	P10_7	IO		√	√	√	√	√	√	√	√	√	√
	P10_8	IO		√	√	√	√	√	√	√	√	√	√
	P10_9	IO		√	√	√	√	√	√	√	√		√

Table 2.3 Pin Function Name Definition (4/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	P10_10	IO	Port of port group 10	√	√	√	√	√	√	√	√	√	√
	P10_11	IO		√	√	√	√	√	√	√	√	√	√
	P10_12	IO		√	√	√	√	√	√	√	√		√
	P10_13	IO		√	√	√	√	√	√	√	√	√	√
	P10_14	IO		√	√	√	√	√	√	√	√	√	√
	P11_0	IO	Port of port group 11	√	√	√		√					
	P11_1	IO		√	√	√		√					
	P11_2	IO		√	√	√		√					
	P11_3	IO		√	√	√		√					
	P11_4	IO		√	√	√		√					
	P11_5	IO		√	√	√		√					
	P11_6	IO		√	√	√		√					
	P11_7	IO		√	√	√		√					
	P11_8	IO		√	√	√		√					
	P11_9	IO		√	√	√		√					
	P11_10	IO		√	√	√		√					
	P11_11	IO		√	√	√		√					
	P11_12	IO		√	√	√		√					
	P11_13	IO		√	√	√		√					
	P11_14	IO		√	√	√		√					
	P11_15	IO		√	√	√		√					
	P12_0	IO	Port of port group 12	√	√								
	P12_1	IO		√	√								
	P12_2	IO		√	√								
	P12_3	IO		√	√								
	P12_4	IO		√	√								
	P12_5	IO		√	√								
	P17_0	IO	Port of port group 17	√	√	√	√	√	√	√	√	√	√
	P17_1	IO		√	√	√	√	√	√	√	√	√	√
	P17_2	IO		√	√	√	√	√	√	√	√		
	P17_3	IO		√	√	√	√	√	√	√	√		
	P17_4	IO		√	√	√	√	√	√	√	√		
	P17_5	IO		√	√	√	√	√	√	√	√		√
	P17_6	IO		√	√	√	√	√	√	√	√		
	P18_0	IO	Port of port group 18	√	√								
	P18_1	IO		√	√								
	P18_2	IO		√	√								
	P18_3	IO		√	√								
	P18_4	IO		√	√								
	P18_5	IO		√	√								
	P18_6	IO		√	√								
	P18_7	IO		√	√								
	P18_8	IO		√	√								
	P18_9	IO		√	√								
	P18_10	IO		√	√								

Table 2.3 Pin Function Name Definition (5/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	P18_11	IO	Port of port group 18	√	√								
	P18_12	IO		√	√								
	P18_13	IO		√	√								
	P18_14	IO		√	√								
	P18_15	IO		√	√								
	P19_0	IO	Port of port group 19	√	√								
	P19_1	IO		√	√								
	P19_2	IO		√	√								
	P19_3	IO		√	√								
	P19_4	IO		√	√								
	P19_5	IO		√	√								
	P20_0	IO	Port of port group 20	√	√	√	√	√	√	√			
	P20_1	IO		√	√	√	√	√	√	√	√		√
	P20_2	IO		√	√	√	√	√	√	√	√		√
	P20_3	IO		√	√	√	√	√	√	√	√	√	√
	P20_4	IO		√	√	√	√	√	√	√	√	√	√
	P20_5	IO		√	√	√	√	√	√	√	√		√
	P20_6	IO		√	√	√	√	√	√	√	√		√
	P20_7	IO		√	√	√	√	√	√	√	√	√	√
	P20_8	IO		√	√	√	√	√	√	√	√	√	√
	P20_9	IO		√	√	√	√	√	√	√	√	√	√
	P20_10	IO		√	√	√	√	√	√	√	√	√	√
	P20_12	IO		√	√	√	√	√	√	√	√		√
	P20_13	IO		√	√	√	√	√	√	√	√	√	√
	P20_14	IO		√	√	√	√	√	√	√	√	√	√
	P20_15	IO		√	√								
	P21_0	IO	Port of port group 21	√	√	√	√	√	√	√	√	√	
	P21_1	IO		√	√	√	√	√	√	√	√	√	
	P21_2	IO		√	√	√	√	√	√	√	√	√	
	P21_3	IO		√	√	√	√	√	√	√			
	P21_4	IO		√	√	√	√	√	√	√			
	P21_5	IO		√	√	√	√	√	√	√			
	P21_6	IO		√	√	√	√	√	√	√			
	P21_7	IO		√	√	√	√	√	√	√			
	P21_12	IO		√	√								
	P21_13	IO		√	√								
	P22_0	IO	Port of port group 22	√	√	√	√	√	√	√			
	P22_1	IO		√	√	√	√	√	√	√			
	P22_2	IO		√	√	√	√	√	√	√			
	P22_3	IO		√	√	√	√	√	√	√			
	P22_4	IO		√	√	√	√	√	√	√			
	P23_0	IO	Port of port group 23	√	√								
	P23_1	IO		√	√								
	P23_2	IO		√	√								
	P23_3	IO		√	√								

Table 2.3 Pin Function Name Definition (6/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	P23_4	IO	Port of port group 23	√	√								
	P23_5	IO		√	√								
	P23_6	IO		√	√								
	P23_7	IO		√	√								
	P23_8	IO		√	√								
	P23_9	IO		√	√								
	P23_10	IO		√	√								
	P23_11	IO		√	√								
	P23_12	IO		√	√								
	P23_13	IO		√	√								
	P24_4	IO	Port of port group 24	√	√	√	√	√	√	√	√	√	
	P24_5	IO		√	√	√	√	√	√	√	√	√	
	P24_6	IO		√	√	√	√	√	√	√	√	√	
	P24_7	IO		√	√	√	√	√	√	√	√	√	√
	P24_8	IO		√	√	√	√	√	√	√	√	√	√
	P24_9	IO		√	√	√	√	√	√	√	√	√	
	P24_10	IO		√	√	√	√	√	√	√	√		
	P24_11	IO		√	√	√	√	√	√	√	√		√
	P24_12	IO		√	√	√	√	√	√	√	√		
	P24_13	IO		√	√	√	√	√	√	√	√		
	AP0_0	IO	Port of analog port group 0	√	√	√	√	√	√	√	√	√	√
	AP0_1	IO		√	√	√	√	√	√	√	√	√	√
	AP0_2	IO		√	√	√	√	√	√	√	√	√	√
	AP0_3	IO		√	√	√	√	√	√	√	√	√	√
	AP0_4	IO		√	√	√	√	√	√	√	√	√	√
	AP0_5	IO		√	√	√	√	√	√	√	√	√	√
	AP0_6	IO		√	√	√	√	√	√	√	√	√	√
	AP0_7	IO		√	√	√	√	√	√	√	√	√	√
	AP0_8	IO		√	√	√	√	√	√	√	√		√
	AP0_9	IO		√	√	√	√	√	√	√	√		√
	AP0_10	IO		√	√	√	√	√	√	√	√		
	AP0_11	IO		√	√	√	√	√	√	√	√		
	AP0_12	IO		√	√	√	√	√	√	√	√		
	AP0_13	IO		√	√	√	√	√	√	√	√		
	AP0_14	IO		√	√	√	√	√	√	√			
	AP0_15	IO		√	√	√	√	√	√	√			
	AP1_0	IO	Port of analog port group 1	√	√	√	√	√	√	√			
	AP1_1	IO		√	√	√	√	√	√	√			
	AP1_2	IO		√	√	√	√	√	√	√			
	AP1_3	IO		√	√	√	√	√	√	√			
	AP2_0	IO	Port of analog port group 2	√	√	√	√	√	√	√	√	√	√
	AP2_1	IO		√	√	√	√	√	√	√	√	√	√
	AP2_2	IO		√	√	√	√	√	√	√	√	√	√
	AP2_3	IO		√	√	√	√	√	√	√	√	√	√
	AP2_4	IO		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (7/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Port	AP2_5	IO	Port of analog port group 2	√	√	√	√	√	√	√	√	√	√
	AP2_6	IO		√	√	√	√	√	√	√	√	√	√
	AP2_7	IO		√	√	√	√	√	√	√	√	√	√
	AP2_8	IO		√	√	√	√	√	√	√	√		√
	AP2_9	IO		√	√	√	√	√	√	√	√		√
	AP2_10	IO		√	√	√	√	√	√	√	√		
	AP2_11	IO		√	√	√	√	√	√	√	√		
	AP2_12	IO		√	√	√	√	√	√	√	√		
	AP2_13	IO		√	√	√	√	√	√	√	√		
	AP2_14	IO		√	√	√	√	√	√	√			
	AP2_15	IO		√	√	√	√	√	√	√			
	AP3_0	IO	Port of analog port group 3	√	√	√	√	√	√	√			
	AP3_1	IO		√	√	√	√	√	√	√			
	AP3_2	IO		√	√	√	√	√	√	√			
	AP3_3	IO		√	√	√	√	√	√	√			
	AP4_0	IO	Port of analog port group 4	√	√	√	√	√	√	√	√		
	AP4_1	IO		√	√	√	√	√	√	√	√		
	AP4_2	IO		√	√	√	√	√	√	√	√		
	AP4_3	IO		√	√	√	√	√	√	√	√		
	AP4_4	IO		√	√	√	√	√	√	√	√		
	AP4_5	IO		√	√	√		√					
	AP4_6	IO		√	√	√		√					
	AP4_7	IO		√	√	√		√					
	AP4_8	IO		√	√	√		√					
	AP4_9	IO		√	√	√		√					
	AP4_10	IO		√	√	√		√					
	AP4_11	IO		√	√	√		√					
	AP4_12	IO		√	√	√		√					
	AP4_13	IO		√	√	√		√					
	AP4_14	IO		√	√	√		√					
	AP4_15	IO		√	√	√		√					
	AP5_0	IO	Port of analog port group 5	√	√	√		√					
	AP5_1	IO		√	√	√		√					
	AP5_2	IO		√	√	√		√					
	AP5_3	IO		√	√	√		√					

Table 2.3 Pin Function Name Definition (8/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Interrupts	NMI	I	Non maskable interrupt	√	√	√	√	√	√	√	√	√	√
	INTP0	I	Maskable external interrupt	√	√	√	√	√	√	√	√	√	√
	INTP1	I		√	√	√	√	√	√	√	√	√	√
	INTP2	I		√	√	√	√	√	√	√	√	√	√
	INTP3	I		√	√	√	√	√	√	√	√	√	√
	INTP4	I		√	√	√	√	√	√	√	√	√	√
	INTP5	I		√	√	√	√	√	√	√	√	√	
	INTP6	I		√	√	√	√	√	√	√	√	√	√
	INTP7	I		√	√	√	√	√	√	√	√	√	√
	INTP8	I		√	√	√	√	√	√	√	√	√	√
	INTP9	I	Maskable external interrupt	√	√	√	√	√	√	√	√	√	
	INTP10	I		√	√	√	√	√	√	√	√	√	√
	INTP11	I		√	√	√	√	√	√	√	√	√	
	INTP12	I		√	√	√	√	√	√	√	√		√
	INTP13	I		√	√	√	√	√	√	√	√		
	INTP14	I		√	√	√	√	√	√	√	√		
	INTP15	I		√	√	√	√	√	√	√	√		
	INTP16	I		√	√	√	√	√	√	√	√	√	√
	INTP17	I		√	√	√	√	√	√	√	√	√	√
	INTP18	I		√	√	√	√	√	√	√	√		√
	INTP19	I		√	√	√	√	√	√	√	√	√	√
	INTP20	I		√	√	√	√	√	√	√	√	√	√
	INTP21	I		√	√	√	√	√	√	√	√	√	√
	INTP22	I		√	√	√	√	√	√	√	√		√
	INTP23	I		√	√	√	√	√	√	√	√	√	√
	INTP24	I		√	√	√	√	√	√	√	√	√	√
	INTP25	I		√	√	√	√	√	√	√	√	√	√
	INTP26	I		√	√	√	√	√	√	√	√	√	√
	INTP27	I		√	√	√	√	√	√	√	√	√	√
	INTP28	I		√	√	√	√	√	√	√			
	INTP29	I		√	√	√	√	√	√	√	√		√
	INTP30	I		√	√	√	√	√	√	√	√	√	√
	INTP31	I		√	√	√	√	√	√	√	√	√	√
	INTP32	I		√	√	√	√	√	√	√	√		√
	INTP33	I		√	√	√	√	√	√	√	√	√	√
	INTP34	I		√	√	√	√	√	√	√	√	√	√
	INTP35	I		√	√	√	√	√	√	√	√		
	INTP36	I		√	√	√	√	√	√	√	√	√	√
	INTP37	I		√	√	√	√	√	√	√	√	√	√
	INTP38	I		√	√	√		√					
	INTP39	I		√	√	√		√					

Table 2.3 Pin Function Name Definition (9/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
System Control	FLMD0	I	Operating mode select pin	√	√	√	√	√	√	√	√	√	√
	FLMD1	I		√	√	√	√	√	√	√	√	√	√
	FLMD2	I		√	√	√	√	√	√	√	√	√	√
	MODE0	I		√	√	√	√	√	√	√	√	√	√
	$\overline{\text{RESET}}$	I	Terminal reset	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{RESETOUT}}$	O	Reset output	√	√	√	√	√	√	√	√	√	√
	X1	I	Main oscillator resonator connections	√	√	√	√	√	√	√	√	√	√
	X2	O		√	√	√	√	√	√	√	√	√	√
	EXTCLK0O	O	Clock controller output	√	√	√	√	√	√	√	√	√	√
	EXTCLK1O	O		√	√	√	√	√	√	√	√	√	√
	PWRCTL	O	ISOVDD power control output signal	√	√	√	√	√	√	√	√	√	√
	SVRPGATE	O	SVR gate drive for the high-side PMOSFET	√	√	√	√	√	√	√	√	√	√
	SVRNGATE	O	SVR gate drive for the low-side NMOSFET	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{VMONOUT}}$	O	Voltage monitor error detection output signal	√	√	√	√	√	√	√	√	√	√
LPS	DPO	O	Port output signal for digital input	√	√	√	√	√	√	√	√		√
	APO	O	Port output signal for analog input	√	√	√	√	√	√	√	√		√
	SELDP0	O	External multiplexer select output signal for digital port	√	√	√	√	√	√	√	√		√
	SELDP1	O		√	√	√	√	√	√	√	√		√
	SELDP2	O		√	√	√	√	√	√	√	√		√
	DPIN0	I	Digital port input signal	√	√	√	√	√	√	√	√		√
	DPIN1	I		√	√	√	√	√	√	√	√		√
	DPIN2	I		√	√	√	√	√	√	√	√		√
	DPIN3	I		√	√	√		√					
	DPIN4	I		√	√	√		√					
	DPIN5	I		√	√	√		√					
	DPIN6	I		√	√	√		√					
	DPIN7	I		√	√	√		√					
	DPIN8	I		√	√	√		√					
	DPIN9	I		√	√	√		√					
	DPIN10	I		√	√	√		√					
	DPIN11	I		√	√	√	√	√	√	√	√		√
	DPIN12	I		√	√	√	√	√	√	√	√		√
	DPIN13	I		√	√	√		√					
	DPIN14	I		√	√	√		√					
	DPIN15	I		√	√								
	DPIN16	I		√	√								
	DPIN17	I		√	√								
	DPIN18	I		√	√								
	DPIN19	I		√	√								

Table 2.3 Pin Function Name Definition (10/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
LPS	DPIN20	I	Digital port input signal	√	√								
	DPIN21	I		√	√								
	DPIN22	I		√	√								
	DPIN23	I		√	√								
SFMA	SFMA0CLK	O	SFMA0 clock	√	√	√	√	√	√	√	√		
	SFMA0SSL	O	SFMA0 slave select	√	√	√	√	√	√	√	√		
	SFMA0IO0	IO	SFMA0 master data input / output	√	√	√	√	√	√	√	√		
	SFMA0IO1	IO		√	√	√	√	√	√	√	√		
	SFMA0IO2	IO		√	√	√	√	√	√	√	√		
	SFMA0IO3	IO		√	√	√	√	√	√	√	√		
MMCA	MMCA0CLK	O	MMCA0 clock	√	√	√	√	√	√	√	√		
	MMCA0CMD	IO	MMCA0 command / response	√	√	√	√	√	√	√	√		
	MMCA0DAT0	IO	MMCA0 data	√	√	√	√	√	√	√	√		
	MMCA0DAT1	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT2	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT3	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT4	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT5	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT6	IO		√	√	√	√	√	√	√	√		
	MMCA0DAT7	IO		√	√	√	√	√	√	√	√		
MSPI	MSPI0SC	IO	MSPI0 serial clock input/output	√	√	√	√	√	√	√	√	√	√
	MSPI0SI	I	MSPI0 receive data input	√	√	√	√	√	√	√	√	√	√
	MSPI0SO	O	MSPI0 transmit data output	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{MSPI0SSI}}$	I	MSPI0 serial peripheral chip select input signal	√	√	√	√	√	√	√			
	MSPI0CSS0	O	MSPI0 serial peripheral chip select output signal	√	√	√	√	√	√	√	√	√	√
	MSPI0CSS1	O		√	√	√	√	√	√	√	√		√
	MSPI0CSS2	O		√	√	√	√	√	√	√	√	√	√
	MSPI0CSS3	O		√	√	√	√	√	√	√	√	√	√
	MSPI0CSS4	O		√	√	√	√	√	√	√	√	√	√
	MSPI0CSS5	O		√	√	√	√	√	√	√	√	√	√
	MSPI0CSS6	O		√	√	√	√	√	√	√	√	√	√
	MSPI0CSS7	O		√	√	√	√	√	√	√	√	√	√
	MSPI0DCS	I	MSPI0 data consistency check signal	√	√	√	√	√	√	√	√	√	√
	MSPI0_SCKP	IO	MSPI0 serial clock differential input/ output	√	√	√	√	√	√				
	MSPI0_SCKN	IO		√	√	√	√	√	√				
	MSPI0_SOP	O	MSPI0 transmit data differential output	√	√	√	√	√	√				
	MSPI0_SON	O		√	√	√	√	√	√				
	MSPI0_SIP	I	MSPI0 receive data differential input	√	√	√	√	√	√				
	MSPI0_SIN	I		√	√	√	√	√	√				
	MSPI1SC	IO	MSPI1 serial clock input/output	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (11/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
MSPI	MSPI1SI	I	MSPI1 receive data input	√	√	√	√	√	√	√	√	√	√
	MSPI1SO	O	MSPI1 transmit data output	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{MSPI1SSI}}$	I	MSPI1 serial peripheral chip select input signal	√	√	√	√	√	√	√	√	√	√
	MSPI1CSS0	O	MSPI1 serial peripheral chip select output signal	√	√	√	√	√	√	√	√	√	√
	MSPI1CSS1	O		√	√	√	√	√	√	√	√	√	√
	MSPI1CSS2	O		√	√	√	√	√	√	√	√		√
	MSPI1CSS3	O		√	√	√	√	√	√	√	√	√	√
	MSPI1CSS4	O		√	√	√	√	√	√	√	√	√	√
	MSPI1CSS5	O		√	√	√	√	√	√	√	√	√	√
	MSPI1CSS6	O		√	√	√	√	√	√	√	√	√	√
	MSPI1CSS7	O		√	√	√	√	√	√	√	√	√	√
	MSPI1DCS	I	MSPI1 data consistency check signal	√	√	√	√	√	√	√	√	√	√
	MSPI1_SCKP	IO	MSPI1 serial clock differential input/ output	√	√	√	√	√	√	√			
	MSPI1_SCKN	IO		√	√	√	√	√	√	√			
	MSPI1_SOP	O	MSPI1 transmit data differential output	√	√	√	√	√	√	√			
	MSPI1_SON	O		√	√	√	√	√	√	√			
	MSPI1_SIP	I	MSPI1 receive data differential input	√	√	√	√	√	√	√			
	MSPI1_SIN	I		√	√	√	√	√	√	√			
	MSPI2SC	IO	MSPI2 serial clock input/output	√	√	√	√	√	√	√	√	√	√
	MSPI2SI	I	MSPI2 receive data input	√	√	√	√	√	√	√	√	√	√
	MSPI2SO	O	MSPI2 transmit data output	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{MSPI2SSI}}$	I	MSPI2 serial peripheral chip select input signal	√	√	√	√	√	√	√	√		√
	MSPI2CSS0	O	MSPI2 serial peripheral chip select output signal	√	√	√	√	√	√	√	√	√	√
	MSPI2CSS1	O		√	√	√	√	√	√	√	√		√
	MSPI2CSS2	O		√	√	√	√	√	√	√	√		√
	MSPI2CSS3	O		√	√	√	√	√	√	√	√	√	
	MSPI2CSS4	O	MSPI2 serial peripheral chip select output signal	√	√	√	√	√	√	√	√		√
	MSPI2CSS5	O		√	√	√	√	√	√	√	√		√
	MSPI2CSS6	O		√	√	√	√	√	√	√	√		√
	MSPI2CSS7	O		√	√	√	√	√	√	√	√	√	√
	MSPI2DCS	I	MSPI2 data consistency check signal	√	√	√	√	√	√	√	√	√	√
	MSPI3SC	IO	MSPI3 serial clock input/output	√	√	√	√	√	√	√	√	√	√
	MSPI3SI	I	MSPI3 receive data input	√	√	√	√	√	√	√	√	√	√
	MSPI3SO	O	MSPI3 transmit data output	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (12/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
MSPI	$\overline{\text{MSPI3SSI}}$	I	MSPI3 serial peripheral chip select input signal	√	√	√	√	√	√	√			
	MSPI3CSS0	O	MSPI3 serial peripheral chip select output signal	√	√	√	√	√	√	√	√		√
	MSPI3CSS1	O		√	√	√	√	√	√	√	√		√
	MSPI3CSS2	O		√	√	√	√	√	√	√	√	√	√
	MSPI3CSS3	O		√	√	√	√	√	√	√	√		
	MSPI3CSS4	O		√	√	√	√	√	√	√			
	MSPI3CSS5	O		√	√	√	√	√	√	√			
	MSPI3CSS6	O		√	√	√	√	√	√	√			
	MSPI3CSS7	O		√	√	√	√	√	√	√			
	MSPI3DCS	I	MSPI3 data consistency check signal	√	√	√	√	√	√	√	√	√	√
	MSPI4SC	IO	MSPI4 serial clock input/output	√	√	√	√	√	√	√	√	√	
	MSPI4SI	I	MSPI4 receive data input	√	√	√	√	√	√	√	√	√	
	MSPI4SO	O	MSPI4 transmit data output	√	√	√	√	√	√	√	√	√	
	$\overline{\text{MSPI4SSI}}$	I	MSPI4 serial peripheral chip select input signal	√	√	√	√	√	√	√			
	MSPI4CSS0	O	MSPI4 serial peripheral chip select output signal	√	√	√	√	√	√	√			
	MSPI4CSS1	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS2	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS3	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS4	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS5	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS6	O		√	√	√	√	√	√	√	√	√	
	MSPI4CSS7	O		√	√	√	√	√	√	√	√		
	MSPI4DCS	I	MSPI4 data consistency check signal	√	√	√	√	√	√	√	√	√	
	MSPI5SC	IO	MSPI5 serial clock input/output	√	√	√	√	√	√	√	√	√	
	MSPI5SI	I	MSPI5 receive data input	√	√	√	√	√	√	√	√	√	
	MSPI5SO	O	MSPI5 transmit data output	√	√	√	√	√	√	√	√	√	
	$\overline{\text{MSPI5SSI}}$	I	MSPI5 serial peripheral chip select input signal	√	√	√	√	√	√	√	√	√	
	MSPI5CSS0	O	MSPI5 serial peripheral chip select output signal	√	√	√	√	√	√	√	√	√	
	MSPI5CSS1	O	MSPI5 slave select signal output	√	√	√	√	√	√	√	√	√	
	MSPI5CSS2	O		√	√	√	√	√	√	√	√	√	
	MSPI5CSS3	O		√	√	√	√	√	√	√	√		
	MSPI5DCS	I	MSPI5 data consistency check signal	√	√	√	√	√	√	√	√	√	
	MSPI6SC	IO	MSPI6 serial clock input/output	√	√	√		√					

Table 2.3 Pin Function Name Definition (13/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
MSPI	MSPI6SI	I	MSPI6 receive data input	√	√	√		√					
	MSPI6SO	O	MSPI6 transmit data output	√	√	√		√					
	$\overline{\text{MSPI6SSI}}$	I	MSPI6 serial peripheral chip select input signal	√	√	√		√					
	MSPI6CSS0	O	MSPI6 serial peripheral chip select output signal	√	√	√		√					
	MSPI6CSS1	O		√	√	√		√					
	MSPI6CSS2	O		√	√	√		√					
	MSPI6CSS3	O		√	√	√		√					
	MSPI6DCS	I	MSPI6 data consistency check signal	√	√	√		√					
	MSPI7SC	IO	MSPI7 serial clock input/output	√	√	√		√					
	MSPI7SI	I	MSPI7 receive data input	√	√	√		√					
	MSPI7SO	O	MSPI7 transmit data output	√	√	√		√					
	$\overline{\text{MSPI7SSI}}$	I	MSPI7 serial peripheral chip select input signal	√	√	√		√					
	MSPI7CSS0	O	MSPI7 serial peripheral chip select output signal	√	√	√		√					
	MSPI7CSS1	O		√	√	√		√					
	MSPI7CSS2	O		√	√	√		√					
	MSPI7CSS3	O		√	√	√		√					
	MSPI7DCS	I	MSPI7 data consistency check signal	√	√	√		√					
	MSPI8SC	IO	MSPI8 serial clock input/output	√	√	√		√					
	MSPI8SI	I	MSPI8 receive data input	√	√	√		√					
	MSPI8SO	O	MSPI8 transmit data output	√	√	√		√					
	$\overline{\text{MSPI8SSI}}$	I	MSPI8 serial peripheral chip select input signal	√	√	√		√					
	MSPI8CSS0	O	MSPI8 serial peripheral chip select output signal	√	√	√		√					
	MSPI8CSS1	O		√	√	√		√					
	MSPI8CSS2	O		√	√	√		√					
	MSPI8CSS3	O		√	√	√		√					
	MSPI8DCS	I	MSPI8 data consistency check signal	√	√	√		√					
	MSPI9SC	IO	MSPI9 serial clock input/output	√	√								
	MSPI9SI	I	MSPI9 receive data input	√	√								
	MSPI9SO	O	MSPI9 transmit data output	√	√								
	$\overline{\text{MSPI9SSI}}$	I	MSPI9 serial peripheral chip select input signal	√	√								

Table 2.3 Pin Function Name Definition (14/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
MSPI	MSPI9CSS0	O	MSPI9 serial peripheral chip select output signal	√	√								
	MSPI9CSS1	O		√	√								
	MSPI9CSS2	O		√	√								
	MSPI9CSS3	O		√	√								
	MSPI9DCS	I	MSPI9 data consistency check signal	√	√								
SCI3	SCI30RXD	I	SCI30 data input signal	√	√	√	√	√	√				
	SCI30TXD	O	SCI30 data output signal	√	√	√	√	√	√				
	SCI30SCK	IO	SCI30 serial clock input/output	√	√	√	√	√	√				
	SCI31RXD	I	SCI31 data input signal	√	√	√	√	√	√				
	SCI31TXD	O	SCI31 data output signal	√	√	√	√	√	√				
	SCI31SCK	IO	SCI31 serial clock input/output	√	√	√	√	√	√				
	SCI32RXD	I	SCI32 data input signal	√	√	√	√	√	√				
	SCI32TXD	O	SCI32 data output signal	√	√	√	√	√	√				
	SCI32SCK	IO	SCI32 serial clock input/output	√	√	√	√	√	√				
RLIN3	RLIN30RX	I	RLIN30 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN30TX	O	RLIN30 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN31RX	I	RLIN31 receive data input	√	√	√	√	√	√	√	√		
	RLIN31TX	O	RLIN31 transmit data output	√	√	√	√	√	√	√	√		
	RLIN32RX	I	RLIN32 receive data input	√	√	√	√	√	√	√	√		√
	RLIN32TX	O	RLIN32 transmit data output	√	√	√	√	√	√	√	√		√
	RLIN33RX	I	RLIN33 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN33TX	O	RLIN33 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN34RX	I	RLIN34 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN34TX	O	RLIN34 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN35RX	I	RLIN35 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN35TX	O	RLIN35 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN36RX	I	RLIN36 receive data input	√	√	√	√	√	√	√	√		
	RLIN36TX	O	RLIN36 transmit data output	√	√	√	√	√	√	√	√		
	RLIN37RX	I	RLIN37 receive data input	√	√	√	√	√	√	√	√		√

Table 2.3 Pin Function Name Definition (15/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
RLIN3	RLIN37TX	O	RLIN37 transmit data output	√	√	√	√	√	√	√	√		√
	RLIN38RX	I	RLIN38 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN38TX	O	RLIN38 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN39RX	I	RLIN39 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN39TX	O	RLIN39 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN310RX	I	RLIN310 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN310TX	O	RLIN310 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN311RX	I	RLIN311 receive data input	√	√	√	√	√	√	√	√	√	√
	RLIN311TX	O	RLIN311 transmit data output	√	√	√	√	√	√	√	√	√	√
	RLIN312RX	I	RLIN312 receive data input	√	√	√		√					
	RLIN312TX	O	RLIN312 transmit data output	√	√	√		√					
	RLIN313RX	I	RLIN313 receive data input	√	√	√		√					
	RLIN313TX	O	RLIN313 transmit data output	√	√	√		√					
	RLIN314RX	I	RLIN314 receive data input	√	√	√		√					
	RLIN314TX	O	RLIN314 transmit data output	√	√	√		√					
	RLIN315RX	I	RLIN315 receive data input	√	√	√		√					
	RLIN315TX	O	RLIN315 transmit data output	√	√	√		√					
	RLIN316RX	I	RLIN316 receive data input	√	√	√		√					
	RLIN316TX	O	RLIN316 transmit data output	√	√	√		√					
	RLIN317RX	I	RLIN317 receive data input	√	√	√		√					
	RLIN317TX	O	RLIN317 transmit data output	√	√	√		√					
	RLIN318RX	I	RLIN318 receive data input	√	√	√		√					
	RLIN318TX	O	RLIN318 transmit data output	√	√	√		√					
	RLIN319RX	I	RLIN319 receive data input	√	√	√		√					
	RLIN319TX	O	RLIN319 transmit data output	√	√	√		√					
	RLIN320RX	I	RLIN320 receive data input	√	√	√		√					
	RLIN320TX	O	RLIN320 transmit data output	√	√	√		√					
	RLIN321RX	I	RLIN321 receive data input	√	√	√		√					

Table 2.3 Pin Function Name Definition (16/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
RLIN3	RLIN321TX	O	RLIN321 transmit data output	√	√	√		√					
	RLIN322RX	I	RLIN322 receive data input	√	√	√		√					
	RLIN322TX	O	RLIN322 transmit data output	√	√	√		√					
	RLIN323RX	I	RLIN323 receive data input	√	√	√		√					
	RLIN323TX	O	RLIN323 transmit data output	√	√	√		√					
RIIC	RIIC0SCL	IO	RIIC0 serial clock	√	√	√	√	√	√	√	√	√	√
	RIIC0SDA	IO	RIIC0 serial data	√	√	√	√	√	√	√	√	√	√
	RIIC1SCL	IO	RIIC1 serial clock	√	√	√	√	√	√	√	√	√	
	RIIC1SDA	IO	RIIC1 serial data	√	√	√	√	√	√	√	√	√	
RS-CANFD	CAN0RX	I	CAN0 receive data input	√	√	√	√	√	√	√	√	√	√
	CAN0TX	O	CAN0 transmit data output	√	√	√	√	√	√	√	√	√	√
	CAN1RX	I	CAN1 receive data input	√	√	√	√	√	√	√	√	√	√
	CAN1TX	O	CAN1 transmit data output	√	√	√	√	√	√	√	√	√	√
	CAN2RX	I	CAN2 receive data input	√	√	√	√	√	√	√			√
	CAN2TX	O	CAN2 transmit data output	√	√	√	√	√	√	√			√
	CAN3RX	I	CAN3 receive data input	√	√	√	√	√	√	√	√		
	CAN3TX	O	CAN3 transmit data output	√	√	√	√	√	√	√	√		
	CAN4RX	I	CAN4 receive data input	√	√	√	√	√	√	√	√	√	√
	CAN4TX	O	CAN4 transmit data output	√	√	√	√	√	√	√	√	√	√
	CAN5RX	I	CAN5 receive data input	√	√	√	√	√	√	√	√		
	CAN5TX	O	CAN5 transmit data output	√	√	√	√	√	√	√	√		
	CAN6RX	I	CAN6 receive data input	√	√	√	√	√	√	√	√	√	√
	CAN6TX	O	CAN6 transmit data output	√	√	√	√	√	√	√	√	√	√
	CAN7RX	I	CAN7 receive data input	√	√	√	√	√	√	√	√		√
	CAN7TX	O	CAN7 transmit data output	√	√	√	√	√	√	√	√		√
	CAN8RX	I	CAN8 receive data input	√	√	√	√	√	√	√	√	√	√
	CAN8TX	O	CAN8 transmit data output	√	√	√	√	√	√	√	√	√	√
	CAN9RX	I	CAN9 receive data input	√	√	√	√	√	√	√	√	√	
	CAN9TX	O	CAN9 transmit data output	√	√	√	√	√	√	√	√	√	
	CAN10RX	I	CAN10 receive data input	√	√	√	√	√	√	√	√	√	

Table 2.3 Pin Function Name Definition (17/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
RS-CANFD	CAN10TX	O	CAN10 transmit data output	√	√	√	√	√	√	√	√	√	
	CAN11RX	I	CAN11 receive data input	√	√	√	√	√	√	√	√	√	
	CAN11TX	O	CAN11 transmit data output	√	√	√	√	√	√	√	√	√	
	CAN12RX	I	CAN12 receive data input	√	√	√	√	√	√				
	CAN12TX	O	CAN12 transmit data output	√	√	√	√	√	√				
	CAN13RX	I	CAN13 receive data input	√	√	√	√	√	√				
	CAN13TX	O	CAN13 transmit data output	√	√	√	√	√	√				
	CAN14RX	I	CAN14 receive data input	√	√	√	√	√	√				
	CAN14TX	O	CAN14 transmit data output	√	√	√	√	√	√				
	CAN15RX	I	CAN15 receive data input	√	√	√	√	√	√				
	CAN15TX	O	CAN15 transmit data output	√	√	√	√	√	√				
FLXA	FLXA0RXDA	I	FLXA0 channel A receive data input	√	√	√	√	√	√	√	√	√	√
	FLXA0TXDA	O	FLXA0 channel A transmit data output	√	√	√	√	√	√	√	√	√	√
	FLXA0TXENA	O	FLXA0 channel A transmit enable	√	√	√	√	√	√	√	√	√	√
	FLXA0RXDB	I	FLXA0 channel B receive data input	√	√	√	√	√	√	√	√	√	√
	FLXA0TXDB	O	FLXA0 channel B transmit data output	√	√	√	√	√	√	√	√	√	√
	FLXA0TXENB	O	FLXA0 channel B transmit enable	√	√	√	√	√	√	√	√	√	√
	FLXA0STPWT	I	FLXA0 stop watch trigger input	√	√	√	√	√	√	√	√	√	√
	FLXA1RXDA	I	FLXA1 channel A receive data input	√	√	√	√	√	√				
	FLXA1TXDA	O	FLXA1 channel A transmit data output	√	√	√	√	√	√				
	FLXA1TXENA	O	FLXA1 channel A transmit enable	√	√	√	√	√	√				
	FLXA1RXDB	I	FLXA1 channel B receive data input	√	√	√	√	√	√				
	FLXA1TXDB	O	FLXA1 channel B transmit data output	√	√	√	√	√	√				
	FLXA1TXENB	O	FLXA1 channel B transmit enable	√	√	√	√	√	√				
	FLXA1STPWT	I	FLXA1 stop watch trigger input	√	√	√	√	√	√				

Table 2.3 Pin Function Name Definition (18/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ETNB	ETNB0TXCLK	I	MII transmit clock	√	√	√	√	√	√	√	√		√
	ETNB0RXCLK	I	MII receive clock	√	√	√	√	√	√	√	√		√
	ETNB0TXEN	O	MII/RMII transmit data enable	√	√	√	√	√	√	√	√		√
	ETNB0TXD0	O	MII/RMII transmit data	√	√	√	√	√	√	√	√		√
	ETNB0TXD1	O		√	√	√	√	√	√	√	√		√
	ETNB0TXD2	O		√	√	√	√	√	√	√	√		√
	ETNB0TXD3	O		√	√	√	√	√	√	√	√		√
	ETNB0TXER	O	MII transmit data error	√	√	√	√	√	√	√	√		√
	ETNB0RXDV	I	MII receive data valid	√	√	√	√	√	√	√	√		√
	ETNB0RXD0	I	MII/RMII receive data	√	√	√	√	√	√	√	√		√
	ETNB0RXD1	I		√	√	√	√	√	√	√	√		√
	ETNB0RXD2	I		√	√	√	√	√	√	√	√		√
	ETNB0RXD3	I		√	√	√	√	√	√	√	√		√
	ETNB0RXER	I	MII/RMII receive data error	√	√	√	√	√	√	√	√		√
	ETNB0REFCLK	I	RMII reference clock	√	√	√	√	√	√	√	√		√
	ETNB0CRS_DV	I	RMII receive data valid	√	√	√	√	√	√	√	√		√
	ETNB0MDC	O	PHY management clock	√	√	√	√	√	√	√	√		√
	ETNB0MDIO	IO	PHY management transfer data	√	√	√	√	√	√	√	√		√
	ETNB0LINKSTA	I	PHY link status	√	√	√	√	√	√	√	√		√
	ETNB0WOL	O	Wake-On-LAN (Magic Packet detection)	√	√	√	√	√	√	√	√		√
	TX_DATAN	O	SGMII transmit serial data differential output	√	√	√	√	√	√				
	TX_DATAP	O		√	√	√	√	√	√				
	RX_CLKN	I	SGMII receive ddr clock differential input	√									
	RX_CLKP	I		√									
	RX_DATAN	I	SGMII receive serial data differential input	√	√	√	√	√	√				
	RX_DATAP	I		√	√	√	√	√	√				
	ETNB1REFCLK	I	SGMII reference clock	√	√	√	√	√	√				
	ETNB1TXCLK	I	MII transmit clock	√	√								
	ETNB1RXCLK	I	MII receive clock	√	√								
	ETNB1TXEN	O	MII transmit data enable	√	√								
	ETNB1TXD0	O	MII transmit data	√	√								
	ETNB1TXD1	O		√	√								
	ETNB1TXD2	O		√	√								
	ETNB1TXD3	O		√	√								
	ETNB1TXER	O	MII transmit data error	√	√								
	ETNB1RXDV	I	MII receive data valid	√	√								

Table 2.3 Pin Function Name Definition (19/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ETNB	ETNB1RXD0	I	MII receive data	√	√								
	ETNB1RXD1	I		√	√								
	ETNB1RXD2	I		√	√								
	ETNB1RXD3	I		√	√								
	ETNB1RXER	I	MII receive data error	√	√								
	ETNB1MDC	O	PHY management clock	√	√	√	√	√	√				
	ETNB1MDIO	IO	PHY management transfer data	√	√	√	√	√	√				
	ETNB1LINKSTA	I	PHY link status signal	√	√								
	ETNB1PHYINT	I	PHY interrupt information	√	√	√	√	√	√				
	ETNB1WOL	O	Wake-On-LAN signal (Magic Packet detection)	√	√	√	√	√	√				
RSENT	RSENT0RX	I	RSENT0 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT0SPCO	O	RSENT0 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT1RX	I	RSENT1 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT1SPCO	O	RSENT1 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT2RX	I	RSENT2 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT2SPCO	O	RSENT2 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT3RX	I	RSENT3 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT3SPCO	O	RSENT3 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT4RX	I	RSENT4 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT4SPCO	O	RSENT4 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT5RX	I	RSENT5 receive data input	√	√	√	√	√	√	√	√	√	√
	RSENT5SPCO	O	RSENT5 SPC extension output	√	√	√	√	√	√	√	√	√	√
	RSENT6RX	I	RSENT6 receive data input	√	√	√	√	√	√	√	√	√	
	RSENT6SPCO	O	RSENT6 SPC extension output	√	√	√	√	√	√	√	√	√	
	RSENT7RX	I	RSENT7 receive data input	√	√	√	√	√	√	√	√	√	
	RSENT7SPCO	O	RSENT7 SPC extension output	√	√	√	√	√	√	√	√	√	

Table 2.3 Pin Function Name Definition (20/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
PSI5	PSI50RX	I	PSI50 receive data input	√	√	√	√	√	√	√	√	√	√
	PSI50TX	O	PSI50 transmit data output	√	√	√	√	√	√	√	√	√	√
	PSI51RX	I	PSI51 receive data input	√	√	√	√	√	√	√	√	√	
	PSI51TX	O	PSI51 transmit data output	√	√	√	√	√	√	√	√	√	
	PSI52RX	I	PSI52 receive data input	√	√	√	√	√	√	√	√	√	√
	PSI52TX	O	PSI52 transmit data output	√	√	√	√	√	√	√	√	√	√
	PSI53RX	I	PSI53 receive data input	√	√	√	√	√	√	√	√	√	√
	PSI53TX	O	PSI53 transmit data output	√	√	√	√	√	√	√	√	√	√
	PSI5S0RX	I	UART Rx data	√	√	√	√	√	√	√			
	PSI5S0TX	O	UART Tx data	√	√	√	√	√	√	√			
	PSI5S0CLK	O	UART clock	√	√	√	√	√	√	√			
	PSI5S1RX	I	UART Rx data	√	√	√	√	√	√	√	√	√	√
	PSI5S1TX	O	UART Tx data	√	√	√	√	√	√	√	√	√	√
	PSI5S1CLK	O	UART clock	√	√	√	√	√	√	√	√	√	√
RHSIF	HSIF0_REFCLK	IO	RHSIF0 reference clock input/output	√	√	√	√	√	√				
	HSIF0_RXDN	I	RHSIF0 receive data differential input	√	√	√	√	√	√				
	HSIF0_RXDP	I		√	√	√	√	√	√				
	HSIF0_TXDN	O	RHSIF0 transmit data differential output	√	√	√	√	√	√				
	HSIF0_TXDP	O		√	√	√	√	√	√				
CXP1	CXP10RX	I	CXP10 receive data input	√	√	√	√	√	√				
	CXP10TX	O	CXP10 transmit data output	√	√	√	√	√	√				
	CXP11RX	I	CXP11 receive data input	√	√	√	√	√	√				
	CXP11TX	O	CXP11 transmit data output	√	√	√	√	√	√				
	CXP12RX	I	CXP12 receive data input	√	√	√	√	√	√				
	CXP12TX	O	CXP12 transmit data output	√	√	√	√	√	√				
	CXP13RX	I	CXP13 receive data input	√	√	√	√	√	√				
	CXP13TX	O	CXP13 transmit data output	√	√	√	√	√	√				
OSTM	OSTM8O	O	OSTM8 timer output	√	√	√	√	√	√	√	√	√	√
	OSTM9O	O	OSTM9 timer output	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (21/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
TAUD	TAUD0I0	I	TAUD0 channel input	√	√	√	√	√	√	√	√	√	
	TAUD0I1	I		√	√	√	√	√	√	√	√	√	
	TAUD0I2	I		√	√	√	√	√	√	√	√	√	
	TAUD0I3	I		√	√	√	√	√	√	√	√	√	√
	TAUD0I4	I		√	√	√	√	√	√	√	√	√	√
	TAUD0I5	I		√	√	√	√	√	√	√	√	√	
	TAUD0I6	I		√	√	√	√	√	√	√	√		
	TAUD0I7	I		√	√	√	√	√	√	√	√		√
	TAUD0I8	I		√	√	√	√	√	√	√	√		
	TAUD0I9	I		√	√	√	√	√	√	√	√		
	TAUD0I10	I		√	√	√	√	√	√	√	√	√	√
	TAUD0I11	I		√	√	√	√	√	√	√	√	√	√
	TAUD0I12	I		√	√	√	√	√	√	√	√		
	TAUD0I13	I		√	√	√	√	√	√	√	√		
	TAUD0I14	I		√	√	√	√	√	√	√	√		
	TAUD0I15	I		√	√	√	√	√	√	√	√		√
	TAUD0O0	O	TAUD0 channel output	√	√	√	√	√	√	√	√	√	√
	TAUD0O1	O		√	√	√	√	√	√	√	√		
	TAUD0O2	O		√	√	√	√	√	√	√	√		√
	TAUD0O3	O		√	√	√	√	√	√	√	√		
	TAUD0O4	O		√	√	√	√	√	√	√	√		
	TAUD0O5	O		√	√	√	√	√	√	√	√		
	TAUD0O6	O		√	√	√	√	√	√	√	√		
	TAUD0O7	O		√	√	√	√	√	√	√	√		
	TAUD0O8	O		√	√	√	√	√	√	√	√		
	TAUD0O9	O		√	√	√	√	√	√	√			
	TAUD0O10	O		√	√	√	√	√	√	√			
	TAUD0O11	O		√	√	√	√	√	√	√	√		√
	TAUD0O12	O		√	√	√	√	√	√	√	√		√
	TAUD0O13	O		√	√	√	√	√	√	√	√		√
	TAUD0O14	O		√	√	√	√	√	√	√	√	√	√
	TAUD0O15	O		√	√	√	√	√	√	√			√
	TAUD1I0	I	TAUD1 channel input	√	√	√	√	√	√	√	√		√
	TAUD1I1	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I2	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I3	I		√	√	√	√	√	√	√	√		
	TAUD1I4	I		√	√	√	√	√	√	√	√	√	
	TAUD1I5	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I6	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I7	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I8	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I9	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I10	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I11	I		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (22/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
TAUD	TAUD1I12	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I13	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I14	I		√	√	√	√	√	√	√	√	√	√
	TAUD1I15	I		√	√	√	√	√	√	√	√	√	√
	TAUD1O0	O	TAUD1 channel output	√	√	√	√	√	√	√			
	TAUD1O1	O		√	√	√	√	√	√	√	√		√
	TAUD1O2	O		√	√	√	√	√	√	√	√		√
	TAUD1O3	O		√	√	√	√	√	√	√	√	√	√
	TAUD1O4	O		√	√	√	√	√	√	√	√	√	√
	TAUD1O5	O		√	√	√	√	√	√	√	√		√
	TAUD1O6	O		√	√	√	√	√	√	√	√		√
	TAUD1O7	O		√	√	√	√	√	√	√	√	√	√
	TAUD1O8	O		√	√	√	√	√	√	√	√	√	
	TAUD1O9	O		√	√	√	√	√	√	√	√	√	
	TAUD1O10	O		√	√	√	√	√	√	√	√	√	
	TAUD1O11	O		√	√	√	√	√	√	√			
	TAUD1O12	O	TAUD1 channel output	√	√	√	√	√	√	√			
	TAUD1O13	O		√	√	√	√	√	√	√			
	TAUD1O14	O		√	√	√	√	√	√	√			
	TAUD1O15	O		√	√	√	√	√	√	√			
	TAUD2I0	I	TAUD2 channel input	√	√	√	√	√	√	√	√	√	√
	TAUD2I1	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I2	I		√	√	√	√	√	√	√	√		√
	TAUD2I3	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I4	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I5	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I6	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I7	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I8	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I9	I		√	√	√	√	√	√	√	√		√
	TAUD2I10	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I11	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I12	I		√	√	√	√	√	√	√	√		√
	TAUD2I13	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I14	I		√	√	√	√	√	√	√	√	√	√
	TAUD2I15	I		√	√	√	√	√	√	√	√		

Table 2.3 Pin Function Name Definition (23/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
TAUD	TAUD2O0	O	TAUD2 channel output	√	√	√	√	√	√	√			√
	TAUD2O1	O		√	√	√	√	√	√	√			√
	TAUD2O2	O		√	√	√	√	√	√	√			√
	TAUD2O3	O		√	√	√	√	√	√	√			
	TAUD2O4	O		√	√	√	√	√	√	√			
	TAUD2O5	O		√	√	√	√	√	√	√	√		√
	TAUD2O6	O		√	√	√	√	√	√	√	√	√	√
	TAUD2O7	O		√	√	√	√	√	√	√	√	√	√
	TAUD2O8	O		√	√	√	√	√	√	√	√		√
	TAUD2O9	O		√	√	√	√	√	√	√	√		
	TAUD2O10	O		√	√	√	√	√	√	√			
	TAUD2O11	O		√	√	√	√	√	√	√			
	TAUD2O12	O		√	√	√	√	√	√	√			√
	TAUD2O13	O		√	√	√	√	√	√	√	√		√
	TAUD2O14	O		√	√	√	√	√	√	√	√	√	√
	TAUD2O15	O		√	√	√	√	√	√	√	√	√	
TAUJ	TAUJ0I0	I	TAUJ0 channel input	√	√	√	√	√	√	√			√
	TAUJ0I1	I		√	√	√	√	√	√	√	√		√
	TAUJ0I2	I		√	√	√	√	√	√	√	√	√	√
	TAUJ0I3	I		√	√	√	√	√	√	√	√	√	√
	TAUJ0O0	O	TAUJ0 channel output	√	√	√	√	√	√	√			√
	TAUJ0O1	O		√	√	√	√	√	√	√	√		√
	TAUJ0O2	O		√	√	√	√	√	√	√	√	√	√
	TAUJ0O3	O		√	√	√	√	√	√	√	√	√	√
	TAUJ1I0	I	TAUJ1 channel input	√	√	√	√	√	√	√	√	√	√
	TAUJ1I1	I		√	√	√	√	√	√	√	√	√	√
	TAUJ1I2	I		√	√	√	√	√	√	√	√	√	√
	TAUJ1I3	I		√	√	√	√	√	√	√	√	√	√
	TAUJ1O0	O	TAUJ1 channel output	√	√	√	√	√	√	√	√	√	√
	TAUJ1O1	O		√	√	√	√	√	√	√	√	√	√
	TAUJ1O2	O		√	√	√	√	√	√	√	√	√	√
	TAUJ1O3	O		√	√	√	√	√	√	√	√	√	√
	TAUJ2I0	I	TAUJ2 channel input	√	√	√	√	√	√	√	√	√	√
	TAUJ2I1	I		√	√	√	√	√	√	√	√	√	√
	TAUJ2I2	I		√	√	√	√	√	√	√	√	√	√
	TAUJ2I3	I		√	√	√	√	√	√	√	√		√
	TAUJ2O0	O	TAUJ2 channel output	√	√	√	√	√	√	√	√	√	√
	TAUJ2O1	O		√	√	√	√	√	√	√	√	√	√
	TAUJ2O2	O		√	√	√	√	√	√	√	√	√	√
	TAUJ2O3	O		√	√	√	√	√	√	√	√		√

Table 2.3 Pin Function Name Definition (24/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
TAUJ	TAUJ3I0	I	TAUJ3 channel input	√	√	√	√	√	√	√	√	√	√
	TAUJ3I1	I		√	√	√	√	√	√	√	√	√	√
	TAUJ3I2	I		√	√	√	√	√	√	√	√	√	√
	TAUJ3I3	I		√	√	√	√	√	√	√	√	√	√
	TAUJ3O0	O	TAUJ3 channel output	√	√	√	√	√	√	√	√	√	√
	TAUJ3O1	O		√	√	√	√	√	√	√	√	√	√
	TAUJ3O2	O		√	√	√	√	√	√	√	√	√	√
	TAUJ3O3	O		√	√	√	√	√	√	√	√	√	√
TSG3	TSG30O0	O	TSG30 timer up / down status output	√	√	√	√	√	√	√	√		
	TSG30O1	O		√	√	√	√	√	√	√	√		
	TSG30O2	O		√	√	√	√	√	√	√	√		
	TSG30O3	O		√	√	√	√	√	√	√	√		
	TSG30O4	O		√	√	√	√	√	√	√	√		
	TSG30O5	O		√	√	√	√	√	√	√	√		
	TSG30O6	O		√	√	√	√	√	√	√	√		
	TSG30O7	O		√	√	√	√	√	√	√	√		
	TSG30PTSI0	I	TSG30 hall sensor input	√	√	√	√	√	√	√			
	TSG30PTSI1	I		√	√	√	√	√	√	√			
	TSG30PTSI2	I		√	√	√	√	√	√	√			
	TSG30CLKI	I	TSG30 external clock input enable	√	√	√	√	√	√	√			
	TSG31O0	O	TSG31 timer up / down status output	√	√	√	√	√	√	√	√	√	√
	TSG31O1	O		√	√	√	√	√	√	√	√	√	√
	TSG31O2	O		√	√	√	√	√	√	√	√	√	√
	TSG31O3	O		√	√	√	√	√	√	√	√	√	√
	TSG31O4	O		√	√	√	√	√	√	√	√	√	√
	TSG31O5	O		√	√	√	√	√	√	√	√	√	√
	TSG31O6	O		√	√	√	√	√	√	√	√	√	√
	TSG31O7	O		√	√	√	√	√	√	√	√	√	√
	TSG31PTSI0	I	TSG31 hall sensor input	√	√	√	√	√	√	√	√		√
	TSG31PTSI1	I		√	√	√	√	√	√	√	√		√
	TSG31PTSI2	I		√	√	√	√	√	√	√	√		√
	TSG31CLKI	I	TSG31 external clock input enable	√	√	√	√	√	√	√	√	√	√
TPBA	TPBA0O	O	TPBA0 timer pattern buffer output	√	√	√	√	√	√	√	√	√	√
	TPBA1O	O	TPBA1 timer pattern buffer output	√	√	√	√	√	√	√	√	√	√
GTM	GTM0I0	I	GTM0 timer input signals for TIM0	√	√	√	√	√	√	√	√	√	√
	GTM0I1	I		√	√	√	√	√	√	√	√	√	√
	GTM0I2	I		√	√	√	√	√	√	√	√	√	√
	GTM0I3	I		√	√	√	√	√	√	√	√	√	√
	GTM0I4	I		√	√	√	√	√	√	√	√	√	√
	GTM0I5	I		√	√	√	√	√	√	√	√	√	√
	GTM0I6	I		√	√	√	√	√	√	√	√	√	√
	GTM0I7	I		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (25/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
GTM	GTM1I0	I	GTM0 timer input signals for TIM1	√	√	√	√	√	√	√	√	√	√
	GTM1I1	I		√	√	√	√	√	√	√	√	√	√
	GTM1I2	I		√	√	√	√	√	√	√	√	√	√
	GTM1I3	I		√	√	√	√	√	√	√	√	√	√
	GTM1I4	I		√	√	√	√	√	√	√	√	√	√
	GTM1I5	I		√	√	√	√	√	√	√	√	√	√
	GTM1I6	I		√	√	√	√	√	√	√	√	√	√
	GTM1I7	I		√	√	√	√	√	√	√	√	√	√
	GTM2I0	I	GTM0 timer input signals for TIM2	√	√	√	√	√	√	√	√		√
	GTM2I1	I		√	√	√	√	√	√	√	√	√	√
	GTM2I2	I		√	√	√	√	√	√	√	√	√	√
	GTM2I3	I		√	√	√	√	√	√	√	√		√
	GTM2I4	I		√	√	√	√	√	√	√	√	√	√
	GTM2I5	I		√	√	√	√	√	√	√	√	√	√
	GTM2I6	I		√	√	√	√	√	√	√	√	√	√
	GTM2I7	I		√	√	√	√	√	√	√	√	√	√
	GTM3I0	I	GTM0 timer input signals for TIM3	√	√	√	√	√	√	√			√
	GTM3I1	I		√	√	√	√	√	√	√			√
	GTM3I2	I		√	√	√	√	√	√	√			√
	GTM3I3	I		√	√	√	√	√	√	√			√
	GTM3I4	I		√	√	√	√	√	√	√	√	√	√
	GTM3I5	I		√	√	√	√	√	√	√	√	√	√
	GTM3I6	I		√	√	√	√	√	√	√	√	√	√
	GTM3I7	I		√	√	√	√	√	√	√	√	√	√
	GTMAT0O0	O	GTM0 timer output signals for ATOM0	√	√	√	√	√	√	√	√	√	√
	GTMAT0O1	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O2	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O3	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O4	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O5	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O6	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O7	O		√	√	√	√	√	√	√	√		√
	GTMAT1O0	O	GTM0 timer output signals for ATOM1	√	√	√	√	√	√	√	√	√	√
	GTMAT1O1	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O2	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O3	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O4	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O5	O		√	√	√	√	√	√	√	√		
	GTMAT1O6	O		√	√	√	√	√	√	√	√		
	GTMAT1O7	O		√	√	√	√	√	√	√	√		

Table 2.3 Pin Function Name Definition (26/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
GTM	GTMAT2O0	O	GTM0 timer output signals for ATOM2	√	√	√	√	√	√	√	√		√
	GTMAT2O1	O		√	√	√	√	√	√	√	√	√	
	GTMAT2O2	O		√	√	√	√	√	√	√	√	√	√
	GTMAT2O3	O		√	√	√	√	√	√	√	√	√	
	GTMAT2O4	O		√	√	√	√	√	√	√	√		
	GTMAT2O5	O		√	√	√	√	√	√	√			√
	GTMAT2O6	O		√	√	√	√	√	√	√			
	GTMAT2O7	O		√	√	√	√	√	√	√			
	GTMAT3O0	O	GTM0 timer output signals for ATOM3	√	√	√	√	√	√	√	√		√
	GTMAT3O1	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O2	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O3	O		√	√	√	√	√	√	√	√		√
	GTMAT3O4	O		√	√	√	√	√	√	√	√		√
	GTMAT3O5	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O6	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O7	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O0N	O	GTM0 Inverted timer output signals for ATOM0	√	√	√	√	√	√	√	√	√	√
	GTMAT0O1N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O2N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O3N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O4N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O5N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O6N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT0O7N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O0N	O	GTM0 Inverted timer output signals for ATOM1	√	√	√	√	√	√	√	√	√	√
	GTMAT1O1N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O2N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O3N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O4N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O5N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O6N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT1O7N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT2O0N	O	GTM0 Inverted timer output signals for ATOM2	√	√	√	√	√	√	√	√	√	
	GTMAT2O1N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT2O2N	O		√	√	√	√	√	√	√	√	√	
	GTMAT2O3N	O		√	√	√	√	√	√	√	√	√	
	GTMAT2O4N	O		√	√	√	√	√	√	√	√		√
	GTMAT2O5N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT2O6N	O		√	√	√	√	√	√	√			
	GTMAT2O7N	O		√	√	√	√	√	√	√	√		√
	GTMAT3O0N	O	GTM0 Inverted timer output signals for ATOM3	√	√	√	√	√	√	√	√	√	√
	GTMAT3O1N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O2N	O		√	√	√	√	√	√	√	√		
	GTMAT3O3N	O		√	√	√	√	√	√	√	√	√	
	GTMAT3O4N	O		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (27/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
GTM	GTMAT3O5N	O	GTM0 Inverted timer output signals for ATOM3	√	√	√	√	√	√	√	√		√
	GTMAT3O6N	O		√	√	√	√	√	√	√	√	√	√
	GTMAT3O7N	O		√	√	√	√	√	√	√	√	√	√
	GTMECLK0	O	GTM0 external clock	√	√	√	√	√	√	√	√		
	GTMECLK1	O		√	√	√	√	√	√	√	√		√
	GTMECLK2	O		√	√	√	√	√	√	√	√		
RTCA	RTCA0OUT	O	RTCA0 1-Hz pulse output	√	√	√	√	√	√	√	√	√	√
ENCA	ENCA0E0	I	ENCA0 encoder input	√	√	√	√	√	√	√			
	ENCA0E1	I		√	√	√	√	√	√	√			
	ENCA0EC	I	ENCA0 encoder clear input	√	√	√	√	√	√	√			
	ENCA0TIN0	I	ENCA0 capture trigger input	√	√	√	√	√	√	√			
	ENCA0TIN1	I		√	√	√	√	√	√	√			
	ENCA1E0	I	ENCA1 encoder input	√	√	√	√	√	√	√	√		√
	ENCA1E1	I		√	√	√	√	√	√	√	√		√
	ENCA1EC	I	ENCA1 encoder clear input	√	√	√	√	√	√	√	√		√
	ENCA1TIN0	I	ENCA1 capture trigger input	√	√	√	√	√	√	√	√		√
	ENCA1TIN1	I		√	√	√	√	√	√	√	√		√
PIC	TAPA0UN	O	Motor control output U phase (negative)	√	√	√	√	√	√	√	√		
	TAPA0UP	O	Motor control output U phase (positive)	√	√	√	√	√	√	√	√		
	TAPA0VN	O	Motor control output V phase (negative)	√	√	√	√	√	√	√	√		
	TAPA0VP	O	Motor control output V phase (positive)	√	√	√	√	√	√	√	√		
	TAPA0WN	O	Motor control output W phase (negative)	√	√	√	√	√	√	√	√		
	TAPA0WP	O	Motor control output W phase (positive)	√	√	√	√	√	√	√	√		
	TAPA0ESO	I	TAUD0/TSG30 emergency shut-off	√	√	√	√	√	√	√	√		
	TAPA1ESO	I	TAUD1/TSG31 emergency shut-off	√	√	√	√	√	√	√			
	$\overline{\text{ESO0}}$	I	ATOM emergency shut-off	√	√	√	√	√	√	√	√		
	$\overline{\text{ESO1}}$	I		√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ESO2}}$	I		√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ESO3}}$	I		√	√	√	√	√	√	√	√		
PWM-Diag	PWGC0O	O	PWGC0 output	√	√	√	√	√	√	√	√	√	√
	PWGC1O	O	PWGC1 output	√	√	√	√	√	√	√	√		
	PWGC2O	O	PWGC2 output	√	√	√	√	√	√	√	√		√
	PWGC3O	O	PWGC3 output	√	√	√	√	√	√	√	√		
	PWGC4O	O	PWGC4 output	√	√	√	√	√	√	√	√		
	PWGC5O	O	PWGC5 output	√	√	√	√	√	√	√	√		
	PWGC6O	O	PWGC6 output	√	√	√	√	√	√	√	√		
	PWGC7O	O	PWGC7 output	√	√	√	√	√	√	√	√		
	PWGC8O	O	PWGC8 output	√	√	√	√	√	√	√	√		

Table 2.3 Pin Function Name Definition (28/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
PWM-Diag	PWGC9O	O	PWGC9 output	√	√	√	√	√	√	√			
	PWGC10O	O	PWGC10 output	√	√	√	√	√	√	√			
	PWGC11O	O	PWGC11 output	√	√	√	√	√	√	√	√		√
	PWGC12O	O	PWGC12 output	√	√	√	√	√	√	√	√		√
	PWGC13O	O	PWGC13 output	√	√	√	√	√	√	√	√		√
	PWGC14O	O	PWGC14 output	√	√	√	√	√	√	√	√	√	√
	PWGC15O	O	PWGC15 output	√	√	√	√	√	√	√			√
	PWGC16O	O	PWGC16 output	√	√	√	√	√	√	√			√
	PWGC17O	O	PWGC17 output	√	√	√	√	√	√	√			√
	PWGC18O	O	PWGC18 output	√	√	√	√	√	√	√			√
	PWGC19O	O	PWGC19 output	√	√	√	√	√	√	√	√		√
	PWGC20O	O	PWGC20 output	√	√	√	√	√	√	√	√	√	√
	PWGC21O	O	PWGC21 output	√	√	√	√	√	√	√	√	√	√
	PWGC22O	O	PWGC22 output	√	√	√	√	√	√	√			
	PWGC23O	O	PWGC23 output	√	√	√	√	√	√	√			√
	PWGC24O	O	PWGC24 output	√	√	√	√	√	√	√			
	PWGC25O	O	PWGC25 output	√	√	√	√	√	√	√			
	PWGC26O	O	PWGC26 output	√	√	√	√	√	√	√	√		√
	PWGC27O	O	PWGC27 output	√	√	√	√	√	√	√	√	√	√
	PWGC28O	O	PWGC28 output	√	√	√	√	√	√	√	√	√	√
	PWGC29O	O	PWGC29 output	√	√	√	√	√	√	√	√		√
	PWGC30O	O	PWGC30 output	√	√	√	√	√	√	√	√	√	√
	PWGC31O	O	PWGC31 output	√	√	√	√	√	√	√	√	√	√
	PWGC32O	O	PWGC32 output	√	√	√	√	√	√	√	√		
	PWGC33O	O	PWGC33 output	√	√	√	√	√	√	√	√	√	
	PWGC34O	O	PWGC34 output	√	√	√	√	√	√	√	√	√	√
	PWGC35O	O	PWGC35 output	√	√	√	√	√	√	√	√	√	√
	PWGC36O	O	PWGC36 output	√	√	√	√	√	√	√	√	√	√
	PWGC37O	O	PWGC37 output	√	√	√	√	√	√	√	√	√	√
	PWGC38O	O	PWGC38 output	√	√	√	√	√	√	√	√	√	√
	PWGC39O	O	PWGC39 output	√	√	√	√	√	√	√	√	√	√
	PWGC40O	O	PWGC40 output	√	√	√	√	√	√	√	√	√	√
	PWGC41O	O	PWGC41 output	√	√	√	√	√	√	√	√	√	
	PWGC42O	O	PWGC42 output	√	√	√	√	√	√	√	√	√	
	PWGC43O	O	PWGC43 output	√	√	√	√	√	√	√	√	√	
	PWGC44O	O	PWGC44 output	√	√	√	√	√	√	√	√	√	√
	PWGC45O	O	PWGC45 output	√	√	√	√	√	√	√	√	√	√
	PWGC46O	O	PWGC46 output	√	√	√	√	√	√	√	√	√	
	PWGC47O	O	PWGC47 output	√	√	√	√	√	√	√	√		
	PWGC48O	O	PWGC48 output	√	√	√	√	√	√	√	√		√
	PWGC49O	O	PWGC49 output	√	√	√	√	√	√	√	√	√	√
	PWGC50O	O	PWGC50 output	√	√	√	√	√	√	√	√		√
	PWGC51O	O	PWGC51 output	√	√	√	√	√	√	√	√		√
	PWGC52O	O	PWGC52 output	√	√	√	√	√	√	√	√	√	√
	PWGC53O	O	PWGC53 output	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (29/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
PWM-Diag	PWGC54O	O	PWGC54 output	√	√	√	√	√	√	√	√	√	√
	PWGC55O	O	PWGC55 output	√	√	√	√	√	√	√	√	√	√
	PWGC56O	O	PWGC56 output	√	√	√	√	√	√	√	√	√	√
	PWGC57O	O	PWGC57 output	√	√	√	√	√	√	√	√	√	√
	PWGC58O	O	PWGC58 output	√	√	√	√	√	√	√	√	√	√
	PWGC59O	O	PWGC59 output	√	√	√	√	√	√	√	√	√	√
	PWGC60O	O	PWGC60 output	√	√	√	√	√	√	√	√	√	√
	PWGC61O	O	PWGC61 output	√	√	√	√	√	√	√	√	√	√
	PWGC62O	O	PWGC62 output	√	√	√	√	√	√	√	√	√	√
	PWGC63O	O	PWGC63 output	√	√	√	√	√	√	√	√	√	√
	PWGC64O	O	PWGC64 output	√	√	√	√	√	√	√	√		√
	PWGC65O	O	PWGC65 output	√	√	√	√	√	√	√	√	√	√
	PWGC66O	O	PWGC66 output	√	√	√	√	√	√	√	√	√	√
	PWGC67O	O	PWGC67 output	√	√	√	√	√	√	√	√		√
	PWGC68O	O	PWGC68 output	√	√	√	√	√	√	√	√	√	√
	PWGC69O	O	PWGC69 output	√	√	√	√	√	√	√	√	√	√
	PWGC70O	O	PWGC70 output	√	√	√	√	√	√	√			
	PWGC71O	O	PWGC71 output	√	√	√	√	√	√	√	√		√
	PWGC72O	O	PWGC72 output	√	√	√	√	√	√	√	√		√
	PWGC73O	O	PWGC73 output	√	√	√	√	√	√	√	√	√	√
	PWGC74O	O	PWGC74 output	√	√	√	√	√	√	√	√	√	√
	PWGC75O	O	PWGC75 output	√	√	√	√	√	√	√	√		√
	PWGC76O	O	PWGC76 output	√	√	√	√	√	√	√	√		√
	PWGC77O	O	PWGC77 output	√	√	√	√	√	√	√	√	√	√
	PWGC78O	O	PWGC78 output	√	√	√	√	√	√	√	√	√	√
	PWGC79O	O	PWGC79 output	√	√	√	√	√	√	√	√	√	√
	PWGC80O	O	PWGC80 output	√	√	√	√	√	√	√	√	√	√
	PWGC81O	O	PWGC81 output	√	√	√	√	√	√	√	√		√
	PWGC82O	O	PWGC82 output	√	√	√	√	√	√	√	√	√	√
	PWGC83O	O	PWGC83 output	√	√	√	√	√	√	√	√	√	√
	PWGC84O	O	PWGC84 output	√	√	√	√	√	√	√	√	√	
	PWGC85O	O	PWGC85 output	√	√	√	√	√	√	√	√	√	
	PWGC86O	O	PWGC86 output	√	√	√	√	√	√	√	√	√	
	PWGC87O	O	PWGC87 output	√	√	√	√	√	√	√			
	PWGC88O	O	PWGC88 output	√	√	√	√	√	√	√			
	PWGC89O	O	PWGC89 output	√	√	√	√	√	√	√			
	PWGC90O	O	PWGC90 output	√	√	√	√	√	√	√			
	PWGC91O	O	PWGC91 output	√	√	√	√	√	√	√			
	PWGC92O	O	PWGC92 output	√	√	√	√	√	√	√			
	PWGC93O	O	PWGC93 output	√	√	√	√	√	√	√			
	PWGC94O	O	PWGC94 output	√	√	√	√	√	√	√			
	PWGC95O	O	PWGC95 output	√	√	√	√	√	√	√			

Table 2.3 Pin Function Name Definition (30/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ADCJ	ADCJ0I0	I	ADCJ0 input channel with high accuracy	√	√	√	√	√	√	√	√	√	√
	ADCJ0I1	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I2	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I3	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I4	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I5	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I6	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I7	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I8	I		√	√	√	√	√	√	√	√		√
	ADCJ0I9	I		√	√	√	√	√	√	√	√		√
	ADCJ0I10	I	ADCJ0 input channel with high accuracy	√	√	√	√	√	√	√	√		
	ADCJ0I11	I		√	√	√	√	√	√	√	√		
	ADCJ0I12	I		√	√	√	√	√	√	√	√		
	ADCJ0I13	I		√	√	√	√	√	√	√	√		
	ADCJ0I14	I		√	√	√	√	√	√	√			
	ADCJ0I15	I		√	√	√	√	√	√	√			
	ADCJ0I16	I		√	√	√	√	√	√	√			
	ADCJ0I17	I		√	√	√	√	√	√	√			
	ADCJ0I18	I		√	√	√	√	√	√	√			
	ADCJ0I19	I		√	√	√	√	√	√	√			
	ADCJ0I0S	I	ADCJ0 input channel with low accuracy	√	√	√	√	√	√	√	√	√	
	ADCJ0I1S	I		√	√	√	√	√	√	√	√	√	
	ADCJ0I2S	I		√	√	√	√	√	√	√	√	√	
	ADCJ0I3S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I4S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0I5S	I		√	√	√	√	√	√	√	√	√	
	ADCJ0I6S	I		√	√	√	√	√	√	√	√		
	ADCJ0I7S	I		√	√	√	√	√	√	√	√		√
	ADCJ0I8S	I		√	√	√	√	√	√	√	√		
	ADCJ0I9S	I		√	√	√	√	√	√	√	√		
	ADCJ0SEL0	O	ADCJ0 external analog multiplexer (MPX) output	√	√	√	√	√	√	√	√		√
	ADCJ0SEL1	O		√	√	√	√	√	√	√	√		√
	ADCJ0SEL2	O		√	√	√	√	√	√	√	√	√	√
	ADCJ0TRG0	I	ADCJ0 external trigger	√	√	√	√	√	√	√	√	√	√
	ADCJ0TRG1	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0TRG2	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0TRG3	I		√	√	√	√	√	√	√	√		√
	ADCJ0TRG4	I		√	√	√	√	√	√	√	√	√	√
	ADCJ0CNV0	O	ADCJ0 conversion timing monitor	√	√	√	√	√	√	√	√		√
	ADCJ0CNV1	O		√	√	√	√	√	√	√			
	ADCJ0CNV2	O		√	√	√	√	√	√	√			
	ADCJ0CNV3	O		√	√	√	√	√	√	√			
	ADCJ0CNV4	O		√	√	√	√	√	√	√			
	A0VREFH	I	ADCJ0 reference voltage pin for the analog part	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (31/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ADCJ	ADCJ1I0	I	ADCJ1 input channel with high accuracy	√	√	√	√	√	√	√	√	√	√
	ADCJ1I1	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I2	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I3	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I4	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I5	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I6	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I7	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I8	I		√	√	√	√	√	√	√	√		√
	ADCJ1I9	I		√	√	√	√	√	√	√	√		√
	ADCJ1I10	I		√	√	√	√	√	√	√	√		
	ADCJ1I11	I	ADCJ1 input channel with high accuracy	√	√	√	√	√	√	√	√		
	ADCJ1I12	I		√	√	√	√	√	√	√	√		
	ADCJ1I13	I		√	√	√	√	√	√	√	√		
	ADCJ1I14	I		√	√	√	√	√	√	√			
	ADCJ1I15	I		√	√	√	√	√	√	√			
	ADCJ1I16	I		√	√	√	√	√	√	√			
	ADCJ1I17	I		√	√	√	√	√	√	√			
	ADCJ1I18	I		√	√	√	√	√	√	√			
	ADCJ1I19	I		√	√	√	√	√	√	√			
	ADCJ1I0S	I	ADCJ1 input channel with low accuracy	√	√	√	√	√	√	√	√	√	√
	ADCJ1I1S	I		√	√	√	√	√	√	√	√		
	ADCJ1I2S	I		√	√	√	√	√	√	√	√		
	ADCJ1I3S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I4S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I5S	I		√	√	√	√	√	√	√	√		
	ADCJ1I6S	I		√	√	√	√	√	√	√	√		√
	ADCJ1I7S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I8S	I		√	√	√	√	√	√	√	√		√
	ADCJ1I9S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I10S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I11S	I		√	√	√	√	√	√	√	√		
	ADCJ1I12S	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1I13S	I		√	√	√	√	√	√	√	√		√
	ADCJ1SEL0	O	ADCJ1 external analog multiplexer (MPX) output	√	√	√	√	√	√	√	√	√	√
	ADCJ1SEL1	O		√	√	√	√	√	√	√	√	√	√
	ADCJ1SEL2	O		√	√	√	√	√	√	√	√		
	ADCJ1TRG0	I	ADCJ1 external trigger	√	√	√	√	√	√	√	√		√
	ADCJ1TRG1	I		√	√	√	√	√	√	√	√		√
	ADCJ1TRG2	I		√	√	√	√	√	√	√	√		√
	ADCJ1TRG3	I		√	√	√	√	√	√	√	√	√	√
	ADCJ1TRG4	I		√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (32/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ADCJ	ADCJ1CNV0	O	ADCJ1 conversion timing monitor	√	√	√	√	√	√	√			
	ADCJ1CNV1	O		√	√	√	√	√	√	√			
	ADCJ1CNV2	O		√	√	√	√	√	√	√			
	ADCJ1CNV3	O		√	√	√	√	√	√	√			
	ADCJ1CNV4	O		√	√	√	√	√	√	√			
	A1VREFH	I	ADCJ1 reference voltage pin for the analog part	√	√	√	√	√	√	√	√	√	√
	ADCJ2I0	I	ADCJ2 input channel with high accuracy	√	√	√	√	√	√	√	√		
	ADCJ2I1	I		√	√	√	√	√	√	√	√		
	ADCJ2I2	I		√	√	√	√	√	√	√	√		
	ADCJ2I3	I		√	√	√	√	√	√	√	√		
	ADCJ2I4	I		√	√	√	√	√	√	√	√		
	ADCJ2I5	I		√	√	√		√					
	ADCJ2I6	I		√	√	√		√					
	ADCJ2I7	I		√	√	√		√					
	ADCJ2I8	I	ADCJ2 input channel with high accuracy	√	√	√		√					
	ADCJ2I9	I		√	√	√		√					
	ADCJ2I10	I		√	√	√		√					
	ADCJ2I11	I		√	√	√		√					
	ADCJ2I12	I		√	√	√		√					
	ADCJ2I13	I		√	√	√		√					
	ADCJ2I14	I		√	√	√		√					
	ADCJ2I15	I		√	√	√		√					
	ADCJ2I16	I		√	√	√		√					
	ADCJ2I17	I		√	√	√		√					
	ADCJ2I18	I		√	√	√		√					
	ADCJ2I19	I		√	√	√		√					
	ADCJ2I0S	I	ADCJ2 input channel with low accuracy	√	√	√	√	√	√	√	√		√
	ADCJ2I1S	I		√	√	√	√	√	√	√			
	ADCJ2I2S	I		√	√	√	√	√	√	√			
	ADCJ2I3S	I		√	√	√	√	√	√	√			√
	ADCJ2I4S	I		√	√	√	√	√	√	√	√		√
	ADCJ2I5S	I		√	√	√	√	√	√	√	√		√
	ADCJ2I6S	I		√	√	√	√	√	√	√	√		
	ADCJ2I7S	I		√	√	√	√	√	√	√	√		√
	ADCJ2I8S	I		√	√	√	√	√	√	√	√		√
	ADCJ2I9S	I		√	√	√	√	√	√	√	√		
	ADCJ2SEL0	O	ADCJ2 external analog multiplexer (MPX) output	√	√	√	√	√	√	√	√		√
	ADCJ2SEL1	O		√	√	√	√	√	√	√	√		
	ADCJ2SEL2	O		√	√	√	√	√	√	√	√		√
	ADCJ2TRG0	I	ADCJ2 external trigger	√	√	√	√	√	√	√	√		
	ADCJ2TRG1	I		√	√	√	√	√	√	√	√		
	ADCJ2TRG2	I		√	√	√	√	√	√	√	√		
	ADCJ2TRG3	I		√	√	√	√	√	√	√	√		
	ADCJ2TRG4	I		√	√	√	√	√	√	√	√		

Table 2.3 Pin Function Name Definition (33/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
ADCJ	ADCJ2CNV0	O	ADCJ2 conversion timing monitor	√	√	√	√	√	√	√	√		
	ADCJ2CNV1	O		√	√	√	√	√	√	√	√		
	ADCJ2CNV2	O		√	√	√	√	√	√	√	√		√
	ADCJ2CNV3	O		√	√	√	√	√	√	√	√		√
	ADCJ2CNV4	O		√	√	√	√	√	√	√	√		√
	A2VREFH	I	ADCJ2 reference voltage pin for the analog part	√	√	√	√	√	√	√	√		√
ECM	$\overline{\text{ERROROUT\_M}}$	O	Error output master signal	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ERROROUT\_C}}$	O	Error output checker signal	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ERRORIN0}}$	I	External error input signal	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ERRORIN1}}$	I		√	√	√	√	√	√	√	√	√	√
	$\overline{\text{ERRORIN2}}$	I		√	√	√	√						
	$\overline{\text{ERRORIN3}}$	I		√	√	√	√						
FLASH	FPDT	O	Transmit data output (2-wire UATR, CSI)	√	√	√	√	√	√	√	√	√	√
	FPDR	I	Receive data input (2-wire UATR, CSI)	√	√	√	√	√	√	√	√	√	√
	FPCK	I	Serial clock input (CSI)	√	√	√	√	√	√	√	√	√	√
Debug	$\overline{\text{TRST}}$	I	Test reset input	√	√	√	√	√	√	√	√	√	√
	TDI	I	Test data input	√	√	√	√	√	√	√	√	√	√
	TDO	O	Test data output	√	√	√	√	√	√	√	√	√	√
	TCK	I	Test clock input	√	√	√	√	√	√	√	√	√	√
	TMS	I	Test mode select input	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{RDY}}$	O	Ready output	√	√	√	√	√	√	√	√	√	√
	$\overline{\text{EVTI}}$	I	Event trigger input	√	√								
	$\overline{\text{EVTO}}$	O	Even trigger output	√	√								
	$\overline{\text{LPDRST}}$	I	Test reset input	√	√	√	√	√	√	√	√	√	√
	LPDCLKO	O	LPD 4 pin clock output	√	√	√	√	√	√	√	√	√	√
	LPDI	I	LPD 4 pin data input	√	√	√	√	√	√	√	√	√	√
	LPDO	O	LPD 4 pin data output	√	√	√	√	√	√	√	√	√	√
	LPDCLKI	I	LPD 4 pin clock input	√	√	√	√	√	√	√	√	√	√
AUDR	$\overline{\text{AUDRST}}$	I	AUDR reset signal	√									
	$\overline{\text{AUDSYNC}}$	I	AUDR synchronous signal	√									
	AUDCK	I	AUDR clock signal	√									
	AUDATA0	IO	AUDR data signal	√									
	AUDATA1	IO		√									
	AUDATA2	IO		√									
	AUDATA3	IO		√									

Table 2.3 Pin Function Name Definition (34/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Aurora	CICREFP	I	Trace I/F clock pos	√									
	CICREFN	I	Trace I/F clock neg	√									
	TODP0	O	Trace I/F data pos 0	√									
	TODN0	O	Trace I/F data neg 0	√									
	TODP1	O	Trace I/F data pos 1	√									
	TODN1	O	Trace I/F data neg 1	√									
	TODP2	O	Trace I/F data pos 2	√									
	TODN2	O	Trace I/F data neg 2	√									
	TODP3	O	Trace I/F data pos 3	√									
	TODN3	O	Trace I/F data neg 3	√									
	$\overline{\text{MSYN}}$	I	Trace I/F Synchronization request	√									
	$\overline{\text{AURORES1}}$	I	EMUVDD domain reset signal	√									
	$\overline{\text{AURORES2}}$	I	EMUVDD domain core signal isolation	√									
	$\overline{\text{AURORESPD}}$	I	Core domain EMUVDD signal isolation	√									
ERAM	$\overline{\text{ERAMRESPD}}$	I	Core domain ERAMVDD signal isolation	√									
	$\overline{\text{ERAMRES2}}$	I	ERAMVDD domain core signal isolation	√									
Power	SYSVCC	-	Power supply for System Logic, Internal voltage regulator and SVR power	√	√	√	√	√	√	√	√	√	√
	VCC	-	Power supply for FLASH	√	√	√	√	√	√	√	√	√	√
	VDD	-	Power supply terminal for core	√	√	√	√	√	√	√	√	√	√
	VSS	-	Common Ground	√	√	√	√	√	√	√	√	√	√
	SVRAVCC	-	Power supply for SVR	√	√	√	√	√	√	√	√	√	√
	SVRDRVCC	-		√	√	√	√	√	√	√	√	√	√
	SVRAVSS	-	Ground for SVR	√	√	√	√	√	√	√	√	√	√
	SVRDRVSS	-		√	√	√	√	√	√	√	√	√	√
	AWOVCL	-	External buffer capacitance of regulator	√	√	√	√	√	√	√	√	√	√

Table 2.3 Pin Function Name Definition (35/35)

Category	Function name	I/O	Explanation	RH850/ U2A-EVA (BGA516)	RH850/ U2A16 (BGA516)	RH850/ U2A16 (BGA373)	RH850/ U2A16 (BGA292)	RH850/ U2A8 (BGA373)	RH850/ U2A8 (BGA292)	RH850/ U2A6 (BGA292)	RH850/ U2A6 (QFP176)	RH850/ U2A6 (BGA156)	RH850/ U2A6 (QFP144)
Power	E0VCC	-	Power supply for I/O port	√	√	√	√	√	√	√	√	√	√
	E1VCC	-		√	√	√	√	√	√	√	√	√	√
	E2VCC	-		√	√	√	√	√	√	√	√	√	√
	LVDVCC	-	Power supply for LVDS	√	√	√	√	√	√				
	A0VCC	-	Power supply for A/D converters	√	√	√	√	√	√	√			
	A1VCC	-		√	√	√	√	√	√	√			
	A2VCC	-		√	√	√	√	√	√	√			
	A0VSS	-	Ground for A/D converters	√	√	√	√	√	√	√	√	√	√
	A1VSS	-		√	√	√	√	√	√	√	√	√	√
	A2VSS	-		√	√	√	√	√	√	√	√		√
	GETH0PVCC	-	Power supply for Ethernet	√	√	√	√	√	√				
	GETH0RVCC	-		√	√	√	√	√	√				
	GETH0BVCC	-		√	√	√	√	√	√				
	GETH0VCL	-		√	√	√	√	√	√				
	DVCC	-	Power supply for Aurora	√									
	DVDD	-		√									
	EMUVCC	-		√									
	EMUVDD	-		√									
	ERAMVDD	-	Power supply for ERAM	√									
	ERAMVCC	-		√									
ICUMHA	ICUMGPIO0	O	ICUMHA GPIO control	√	√	√	√	√	√	√	√	√	√
	ICUMGPIO1	O		√	√	√	√	√	√	√	√	√	√
	ICUMGPIO2	O		√	√	√	√	√	√	√	√	√	√
	ICUMGPIO3	O		√	√	√	√	√	√	√	√	√	√

2.2.4 Handling of Unused Pins

Handling of unused pins are listed in following table.

Table 2.4 Handling of Unused Pins (1/2)

Category	Pin Name	Handling of Unused Pins
System Control	X1 RESET FLMD0 SYSVCC VCC VDD VSS AWOVCL	(These pins will always be used.)
	X2 PWRCTL VMONOUT	These pins can be left open.
	P6_10 (RESETOUT)	[Input mode] This pin can be left open and the settings are to disable input (when PMCn_m = 0, PMn_m = 1, PIBCn_m = 0) This pin can be left open and the settings are to enable on-chip pull-down resistors (use PDn_m) This pin can be connected to VSS via individual resistor. [Output mode] This pin can be left open.
SVR	SVRPGATE SVRNGATE	These pins can be left open.
	SVRAVCC SVRDRVCC	These pins must be connected to SYSVCC.
	SVRAVSS SVRDRVSS	These pins must be connected to VSS.
ECM	ERROROUT_M	This pin can be left open.
PORT	Pn_m (excluding P6_10)	[Input mode] These pins can be left open and the settings are to disable input (when PMCn_m = 0, PMn_m = 1, PIBCn_m = 0) These pins can be left open and the settings are to enable on-chip pull-up/pull-down resistors (use PUn_m, PDn_m) These pins can be connected to power supply or ground of each pin via individual resistor. [Output mode] These pins can be left open.
	APn_m	[Input mode] These pins can be left open and the settings are to disable input (when PMn_m = 1, PIBCn_m = 0) These pins can be connected to power supply or ground of each pin via individual resistor. [Output mode] These pins can be left open.
	E0VCC E1VCC E2VCC	(These pins will always be used.)
	LVDVCC	This pin must be connected to E0VCC.
Debug	JP0_0 JP0_1 JP0_2 JP0_3 JP0_5 EVTI EVTO	These pins can be left open.
	TRST	This pin must be connected to VSS via individual resistor.

Table 2.4 Handling of Unused Pins (2/2)

Category	Pin Name	Handling of Unused Pins
Aurora	CICREFP CICREFN TODPn TODNn MSYN	These pins can be left open.
	$\overline{\text{AURORES1}}$ $\overline{\text{AURORES2}}$ $\overline{\text{AURORESPD}}$	These pins must be connected to VSS via individual resistor.
	EMUVCC EMUVDD DVCC DVDD	(These pins will always be used.)
ERAM	$\overline{\text{ERAMRES2}}$ $\overline{\text{ERAMRESPD}}$	These pins must be connected to VSS via individual resistor.
	ERAMVDD ERAMVCC	(These pins will always be used.)
AUDR	AUDATAn	These pins can be left open.
	$\overline{\text{AUDSYNC}}$ AUDCK	These pins can be left open These pins can be connected to E0VCC via individual resistor.
	$\overline{\text{AUDRST}}$	This pin can be left open This pin can be connected to VSS via individual resistor.
SGMII	RX_CLKN RX_CLKP RX_DATAN RX_DATAP TX_DATAN TX_DATAP	These pins can be left open.
	GETH0PVCC GETH0RVCC	These pins must be connected to E0VCC. (When E0VCC = E1VCC = E2VCC then these pins can be connected to each of them.)
	GETH0BVCC	This pin must be connected to 3.0V to 3.6V voltage power supply, or VSS with 1 kΩ or more pull-down resistance.
	GETH0VCL	This pin can be left open.
ADC	A0VCC A0VREFH	These pins must be connected to E2VCC.
	A1VCC A1VREFH	These pins must be connected to E1VCC.
	A2VCC A2VREFH	These pins must be connected to E0VCC.
	A0VSS A1VSS A2VSS	These pins must be connected to VSS.

2.3 PORT Overview

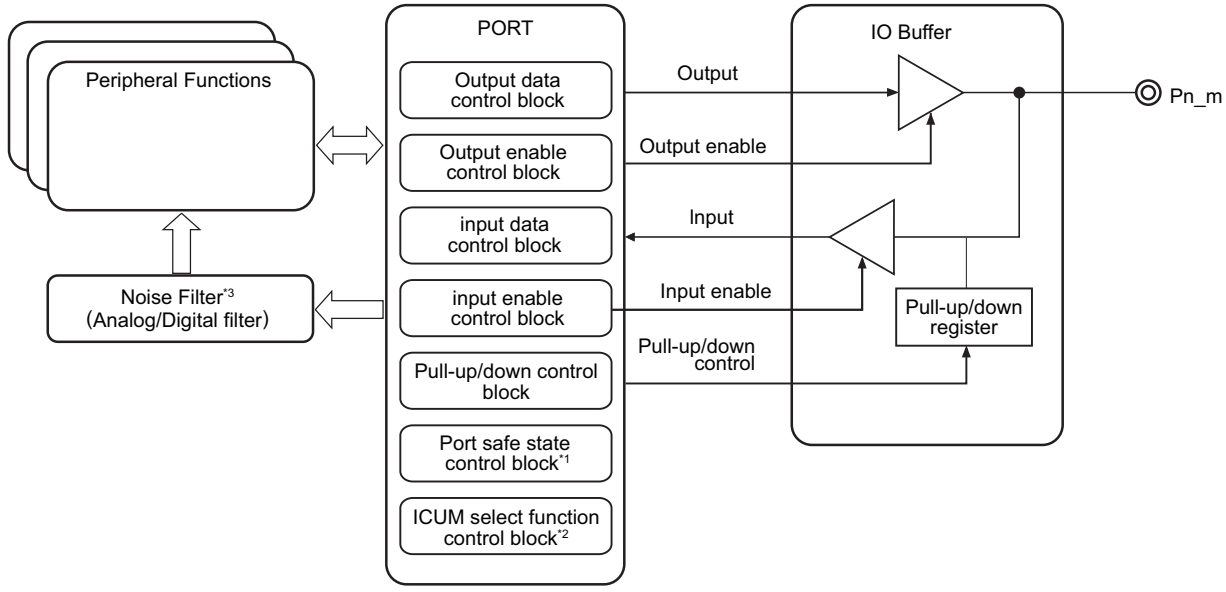
The product has various pins for input/output functions, known as ports. The ports are organized in port groups. Each pin is configurable to be used as GPIO or input/output function of internal IP.

Features summary:

- Configurable for individual pins.
- Each pin is multiplexed GPIO and many input/output functions of internal IPs.
- Electrical characteristics is configurable for most of the pins.
- Support port safe state function.
- Noise filter (Analog or Digital filter) is available for some functional signals of internal IPs.
- Supports LVDS function for high-speed interface.
- Shared analog pins with some GPIO pins are available for ADC.
- Write access protection is available for specific registers or whole port group.

2.3.1 Introduction

The port is an interface between external devices and the internal resources (IPs) and usually consists of IO buffer and the control logic as it is shown below.



Note 1. For details of port safe state function, see **Section 2.4.2, Port Safe State Function**
Note 2. For details of ICUM select function, see **Section 2.4.3, ICUM Select Function**
Note 3. For details of Noise Filter, see **Section 2.7, Noise Filter & Edge/Level Detector**

Figure 2.1 Functional Block Diagram

2.3.2 Register Base Addresses

All port addresses are given as an offset from the individual base addresses, <JPORT_base> and <PORT_base>.

Table 2.5 Port Group

Base Address Name	Base Address	Bus Group
<PORT_base>	FFD9 0000$_H$	P-Bus Group 2L
<JPORT_base>	FFDA 0000$_H$	P-Bus Group 2L

2.3.3 Clock Supply

The clock supply to ports is shown in the following table.

Table 2.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
PORT	Register access clock	CLK_LSB

2.3.4 Operation Mode

Pins can operate in three operation modes.

- Port mode (PMCn.PMCn_m = 0)
  A pin in port mode operates as a general purpose input/output pin. The input / output mode is selected by setting the PMn.PMn_m bit.
- Alternative mode software I/O control (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 0)
  In this mode, the pins operate as alternative functions. The input / output mode is selected by setting the PMn.PMn_m bit.
- Alternative mode direct I/O control (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 1)
  In this mode, the pins operate as alternative functions. Unlike the alternative mode software I/O control, however, the input / output direction is selected by the alternative function.

The following is register effect related to operation modes and pin I/O direction.

- PMCn.PMCn_m bit
  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).
- PMn.PMn_m bit
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) when the port mode (PMCn_m = 0) and alternative mode software I/O control (PMCn_m = 1, PIPCn_m = 0) have been selected.
- PIBCn.PIBCn_m bit
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0, PMn_m = 1).
- PIPCn.PIPCn_m bit
  This bit selects alternative mode software I/O control or alternative mode direct I/O control.
- PBDCn.PBDCn_m bit
  In output mode or output enabled by alternative function, when this bit is set to 1, the pin enters the bidirectional mode.

The input/output direction and each modes by register setting is shown below **Table 2.7, Pin register setting**.

Table 2.7 Pin register setting

PMCn_m	PMn_m	PIBCn_m	PIPCn_m	PBDCn_m	Modes	I/O Direction
0	0	X	X	0	Port mode Output mode	Output
				1	Port mode Bi-directional mode	Input/Output
	1	0		X	Port mode Input mode (input disabled)	–
		1			Port mode Input mode (input enabled)	Input
1	0	X	0	0	Alternative mode Software I/O control Output mode	Output
		X*2		1	Alternative mode Software I/O control Bi-directional mode	Input/Output
	1	0*1		X	Alternative mode Software I/O control Input mode	Input
	X		1	0	Alternative mode Direct I/O control Input or output	Controlled by the alternative function
		X*2		1	Alternative mode Direct I/O control Input or bi-direction	Controlled by the alternative function

Note 1. Setting PIBC 1 is prohibited.

Note 2. Setting PIBC 0 in Bi-directional mode makes propagate the pin level to alternative input function.
Setting PIBC 1 in Bi-directional mode does not make propagate the pin level to alternative input function.

Table 2.8 Alternative mode selection table

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1 (ALT-IN1)
			0	Alternative output mode 1 (ALT-OUT1)
		1	1	Alternative input mode 2 (ALT-IN2)
			0	Alternative output mode 2 (ALT-OUT2)
	1	0	1	Alternative input mode 3 (ALT-IN3)
			0	Alternative output mode 3 (ALT-OUT3)
		1	1	Alternative input mode 4 (ALT-IN4)
			0	Alternative output mode 4 (ALT-OUT4)
1	0	0	1	Alternative input mode 5 (ALT-IN5)
			0	Alternative output mode 5 (ALT-OUT5)
		1	1	Alternative input mode 6 (ALT-IN6)
			0	Alternative output mode 6 (ALT-OUT6)
	1	0	1	Alternative input mode 7 (ALT-IN7)
			0	Alternative output mode 7 (ALT-OUT7)
		1	1	Alternative input mode 8 (ALT-IN8)
			0	Alternative output mode 8 (ALT-OUT8)

NOTE

Some input or output functions is assigned to more than one pin. Only activate one single pin to one given alternative input function. Do not activate a input function on multiple pins at the same time.
Do not set alternative mode selection to a place where the alternative function is not assigned.

2.3.5 Pin data input/output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.

(1) Output data

In the port mode (PMCn.PMCn_m = 0), the value of the Pn.Pn_m bit is output to the Pn_m pin.

(2) Input data

When the PPRn register is read, either the value of the Pn_m pin, the value of the corresponding bit of the port register Pn.Pn_m, or the value output by the alternative function is returned.

Which value is returned depends on the pin mode and setting of several control bits. The different PPRn read modes are shown in the **Table 2.9, PPRn_m Read Values**.

Table 2.9 PPRn_m Read Values

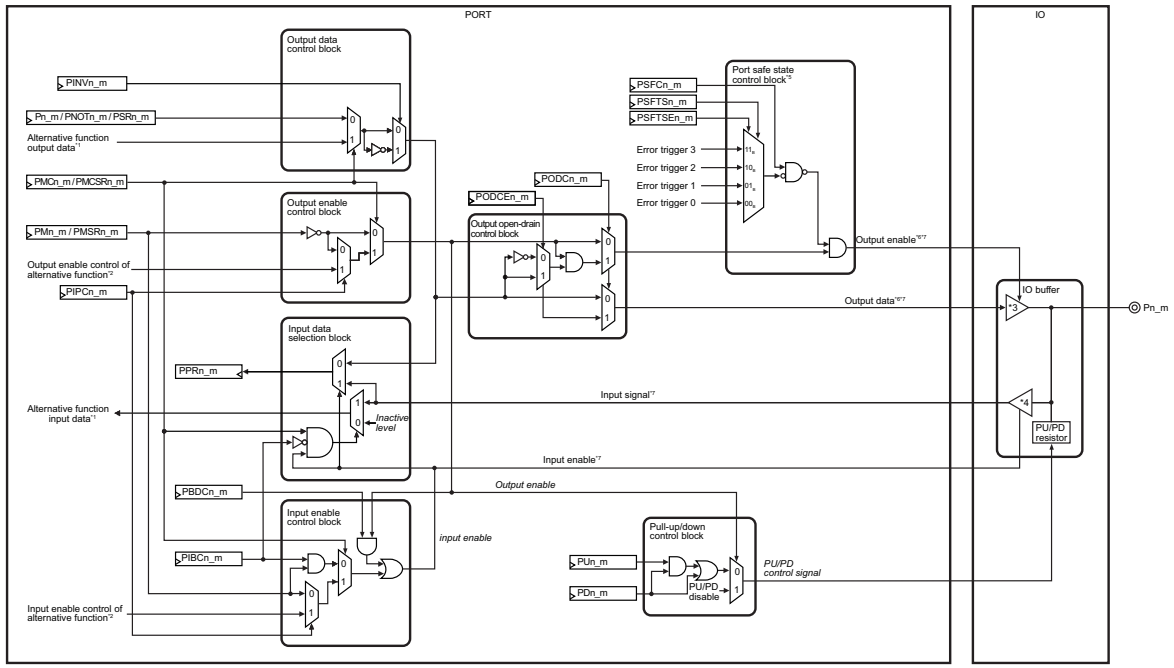
PMCn_m	PMn_m	PIBCn_m	PIPCn_m	PBDCn_m	Mode	PPRn_m Read Value
0	0	X	X	0	Port Mode Output Mode	Pn.Pn_m bit
				1	Port Mode Bi-directional Mode	Pn_m pin
	1	0		X	Port Mode Input Mode, (Input disabled)	Pn.Pn_m bit
		1			Port Mode Input Mode (Input enabled)	Pn_m pin
1	0	X	0	0	Alternative Mode Software I/O control Output Mode	Alternative-function internal output signal
				1	Alternative Mode Software I/O control Bi-directional Mode	Pn_m pin
	1			X	Alternative Mode Software I/O control Input Mode	Pn_m pin
	X		1	0	Alternative Mode Direct I/O control Input or output	I/O port in alternative mode: Input: Pn_m pin Output: Alternative-function internal output signal
				1	Alternative Mode Direct I/O control Input or bi-direction	I/O port in alternative mode: Input: Pn_m pin Bi-direction: Pn_m pin

2.4 Port Functions

This section explains port control logic functions.

2.4.1 Port Control Logic

The following figure shows the logical circuitry of the port control functions. The diagram is only a logical reference and does not show the real circuitry.



Note 1. Alternative function signal selected by PFCn_m, PFCEn_m and PFCAEn_m.

Note 2. Only functions exist in **Table 2.29, Alternative functions require "Direct I/O Control" (Must Set PIPCn_m=1)** are supported this configuration.

Note 3. The output drive strength can be set by PDSCn_m and PUCCn_m.

Note 4. The input buffer type can be set by PISn_m and PISAn_m.

Note 5. Detail of Port Safe State function is described in **Section 2.4.2, Port Safe State Function.**

Note 6. P10_0, P10_1, P17_0 and P17_1 are target pins for ICUM select function which requires additional higher priority control. For details, refer to **Section 2.4.3, ICUM Select Function**.

Note 7. The alternative functions which are supported LVDS buffers have additional control logic from LVDS registers. For details, refer to **Section 2.4.5, Functions incorporated LVDS Buffers**.

Figure 2.2 Port Control Logic Block Diagram

2.4.2 Port Safe State Function

Pn_m and APn_m have Port Safe State Function which can operate the output enable of each pins individually by selecting error triggers for the defined application. This product supports up to 4 error triggers named ERROROUT0Z, ERROROUT1Z, ERROROUT2Z, and ERROROUT3Z, and the trigger can be selected by setting of PSFTSn_m and PSFTSEn_m registers. For details of error triggers behavior, refer to **Section 45, Error Control Module (ECM)**.

The output enable control of a pin is forcibly disabled in case that an error trigger is issued when Port Safe State Function is enabled by setting PSFCn_m register. Even Bi-directional mode output data is suppressed by Port Safe State function during an error trigger is issued.

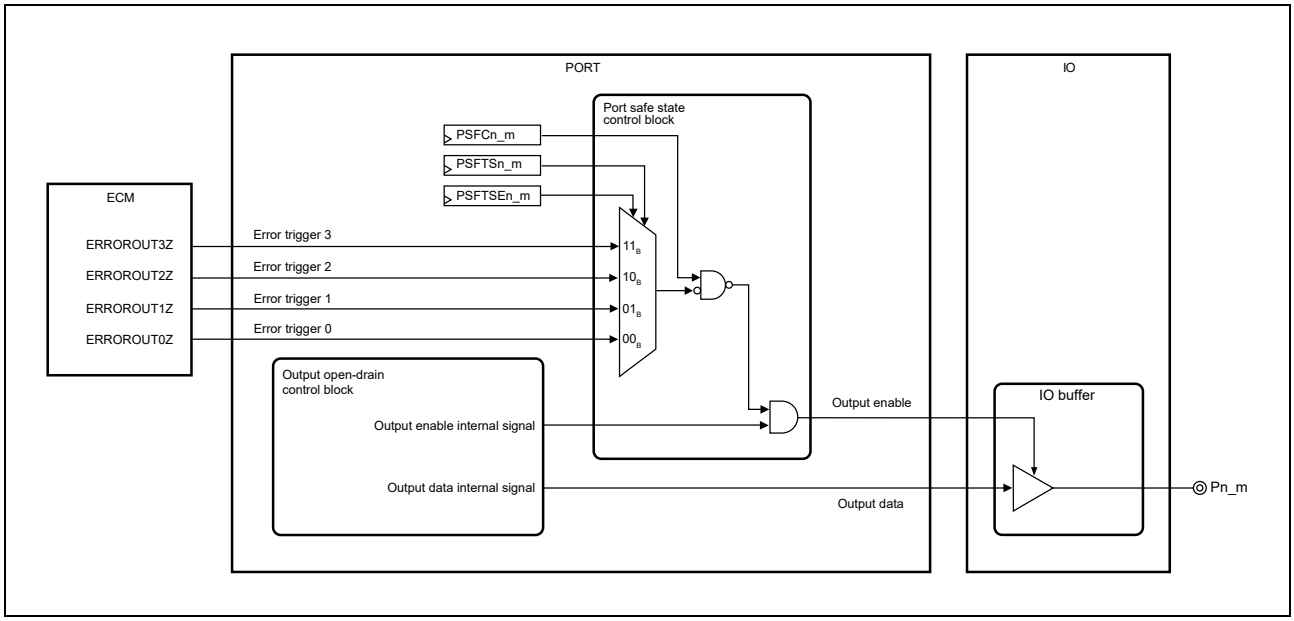


Figure 2.3 Block Diagram of Port Safe State Function

The following table shows the register configuration for Port Safe State Function

Table 2.10 Configuration for Port Safe State Function

PSFCn_m	PSFTSEn_m	PSFTSn_m	Function
0	X	X	Port Safe State function is disabled
1	0	0	Port Safe State function is enabled. Error Trigger 0 is selected for Port Safe State function.
		1	Port Safe State function is enabled. Error Trigger 1 is selected for Port Safe State function.
	1	0	Port Safe State function is enabled. Error Trigger 2 is selected for Port Safe State function.
		1	Port Safe State function is enabled. Error Trigger 3 is selected for Port Safe State function.

CAUTION

Error triggers from ECM are triggered by DeepSTOP Reset and the status of a pin located to AWO area becomes high impedance when Port Safe State Function is enabled.

2.4.3 ICUM Select Function

Target pins: P10_0, P10_1, P17_0 and P17_1.

The ICUM Select Function takes highest priority in port functions to control output value of these target pins.

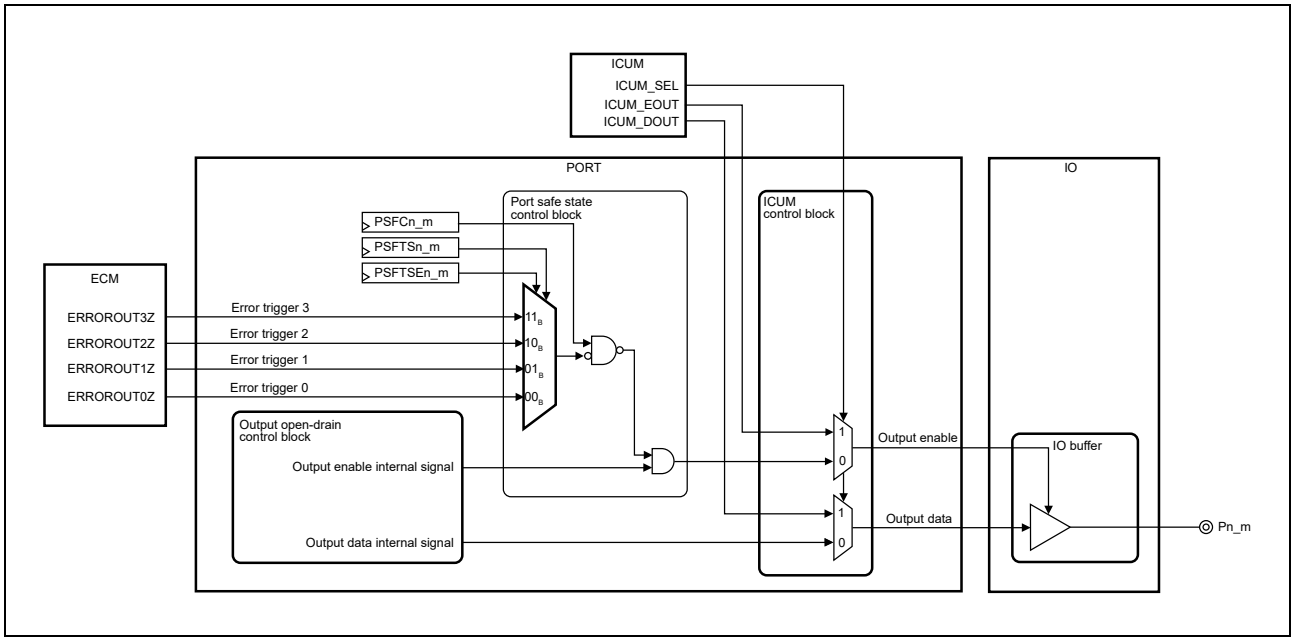


Figure 2.4 Block Diagram of ICUM Select Function

For more details how to use the ICUM function, refer to *RH850/U2A-EVA Group Security User's Manual: Hardware*.

2.4.4 $\overline{\text{RESETOUT}}$ Function

P6_10 has $\overline{\text{RESETOUT}}$ function to handshake with external device. To support this, P6_10 will drive out low level to external device during and after reset which is effective for any kind of resets (except DeepSTOP Reset, Module Reset and JTAG Reset).

This function is realized by having a special reset value of PM6[10], PDSC6[10] and PUCC6[10], which are inverted in contrast to the others.

P6_10 keeps driving out low level until the related registers are changed by user program.

The following figure shows behavior of $\overline{\text{RESETOUT}}$.

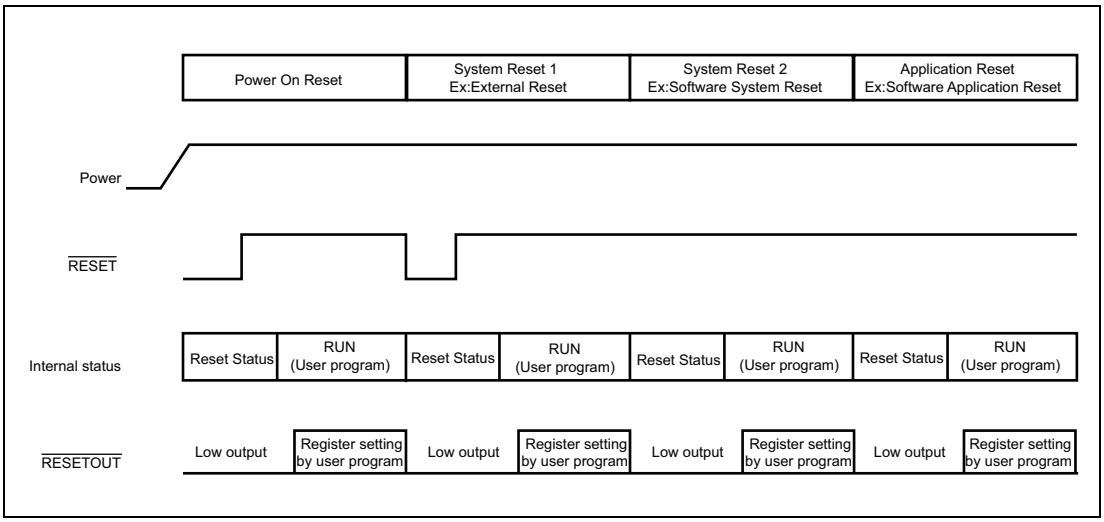


Figure 2.5 Timing Chart of $\overline{\text{RESETOUT}}$ function for each reset factor

CAUTION

To avoid data collision, the outside circuit connected to this pin must not drive in high level at any case.

2.4.5 Functions incorporated LVDS Buffers

When the LVDS buffer is used, set the LVDSCTRL[A,B] register to enable input/output. Also, set whether the supply voltage of LVDS buffer is 3V or 5V.

Table 2.11 LVDS Control Register Assignment

Control Register		Power Domain	I/F	Pin Name (A pair of P-ch, N-ch)	I/O	Control	Related Function (A pair of P-ch, N-ch)
Register	Bit						
LVDSCTRLA	[3]	LVDVCC	IEEE	P2_13/P2_12	OUT	OE	MSPI0_SOP/MSPI0_SON
							HSIF0_TXDP/HSIF0_TXDN
	[0]			P2_11/P2_10	IN	IE	MSPI0_SIP/MSPI0_SIN
							HSIF0_RXDP/HSIF0_RXDN
	[16]					VSEL	MSPI0_SIP/MSPI0_SIN
							HSIF0_RXDP/HSIF0_RXDN
	[5]			P2_15/P2_14	OUT	OE	MSPI0_SCKP/MSPI0_SCKN
	[4]				IN	IE	MSPI0_SCKP/MSPI0_SCKN
	[20]					VSEL	MSPI0_SCKP/MSPI0_SCKN
LVDSCTRLB	[5]	E0VCC	IEEE	P4_9/P4_10	OUT	OE	MSPI1_SCKP/MSPI1_SCKN
	[4]				IN	IE	MSPI1_SCKP/MSPI1_SCKN
	[20]					VSEL	MSPI1_SCKP/MSPI1_SCKN
	[3]			P4_7/P4_8	OUT	OE	MSPI1_SOP/MSPI1_SON
	[0]			P4_5/P4_6	IN	IE	MSPI1_SIP/MSPI1_SIN
	[16]					VSEL	MSPI1_SIP/MSPI1_SIN

CAUTION

Input/output from the LVDS buffer actually being enabled/disable takes at least 10 µs after setting LVDSCTRLA, LVDSCTRLB, IOHOLD0.IOHOLD_P2, and IOHOLD0.IOHOLD_P4.

For pins have not only functions incorporated LVDS buffers but also GPIO and alternative functions. These cannot be used at the same time. Refer to Section 2.6.4, LVDS Function Setting in case LVDS buffers are used.

2.4.6 Debug Interface

The switch between debug interface and port mode is controlled by a combination of mode pins and option byte settings. For details, see **Section 50, Debugging and Calibration**.

2.5 Port Register Description

For detailed list control registers of each port group, refer to Appendix “E02-02_List of Port Functions.xlsx”.

In the bitmap field, “x” means an effective bit and “—” means a reserved bit. Reserved areas are always read as the values after reset. The write value also should be the value after reset. The registers of unimplemented pins should not be modified after reset.

The list of port control register types are described below.

Table 2.12 Port Register Overview (1/2)

Port Register Name	Port Symbol	Address	Access Protection		
			PBG	Other	
				Excluding P2, P6, JP0	P2, P6 *1
Port register	Pn	<PORT_base> + 0000_H + 40_H*n	Refer to Appendix “E02-02_List of Port Functions.xlsx”	—	PWE
	APn	<PORT_base> + $0C80_H$ + 40_H*n		—	—
	JPn	<JPORT_base> + 0000_H + 40_H*n		—	—
Port Set Reset register	PSRn	<PORT_base> + 0004_H + 40_H*n		—	PWE
	APSRn	<PORT_base> + $0C84_H$ + 40_H*n		—	—
	JPSRn	<JPORT_base> + 0004_H + 40_H*n		—	—
Port NOT register	PNOTn	<PORT_base> + 0008_H + 40_H*n		—	PWE
	APNOTn	<PORT_base> + $0C88_H$ + 40_H*n		—	—
	JPNOTn	<JPORT_base> + 0008_H + 40_H*n		—	—
Port Pin Read register	PPRn	<PORT_base> + $000C_H$ + 40_H*n		—	—
	APPRn	<PORT_base> + $0C8C_H$ + 40_H*n		—	—
	JPPRn	<JPORT_base> + $000C_H$ + 40_H*n		—	—
Port Mode register	PMn	<PORT_base> + 0010_H + 40_H*n		—	PWE
	APMn	<PORT_base> + $0C90_H$ + 40_H*n		—	—
	JPMn	<JPORT_base> + 0010_H + 40_H*n		—	—
Port Mode Control register	PMCn	<PORT_base> + 0014_H + 40_H*n		—	PWE
	JPMCn	<JPORT_base> + 0014_H + 40_H*n		—	—
Port Function Control register	PFCn	<PORT_base> + 0018_H + 40_H*n		—	PWE
	JPFCn	<JPORT_base> + 0018_H + 40_H*n		—	—
Port Function Control Expansion register	PFCEn	<PORT_base> + $001C_H$ + 40_H*n		—	PWE
	JPFCEn	<JPORT_base> + $001C_H$ + 40_H*n		—	—
Port Mode Set Reset register	PMSRn	<PORT_base> + 0020_H + 40_H*n		—	PWE
	APMSRn	<PORT_base> + $0CA0_H$ + 40_H*n		—	—
	JPMSRn	<JPORT_base> + 0020_H + 40_H*n		—	—
Port Mode Control Set Reset register	PMCSRn	<PORT_base> + 0024_H + 40_H*n		—	PWE
	JPMCSRn	<JPORT_base> + 0024_H + 40_H*n		—	—
Port Function Control Additional Expansion register	PFCAEn	<PORT_base> + 0028_H + 40_H*n		—	PWE
Port output value Inversion register	PINVn	<PORT_base> + 0030_H + 40_H*n		PWE	PWE
	APINVn	<PORT_base> + $0CB0_H$ + 40_H*n		PWE	—
	JPINVn	<JPORT_base> + 0030_H + 40_H*n		—	—
Port Input Buffer Control register	PIBCn	<PORT_base> + 4000_H + 40_H*n		—	PWE
	APIBCn	<PORT_base> + $4C80_H$ + 40_H*n		—	—
	JPIBCn	<JPORT_base> + 4000_H + 40_H*n		—	—

Table 2.12 Port Register Overview (2/2)

Port Register Name	Port Symbol	Address	Access Protection		
			PBG	Other	
				Excluding P2, P6, JP0	P2, P6 *1
Port Bi-Direction Control register	PBDCn	<PORT_base> + 4004_H + 40_H*n	Refer to Appendix "E02-02_List of Port Functions.xlsx"	—	PWE
	APBDCn	<PORT_base> + $4C84_H$ + 40_H*n		—	—
	JPBDCn	<JPORT_base> + 4004_H + 40_H*n		—	—
Port IP Control register	PIPCn	<PORT_base> + 4008_H + 40_H*n		—	PWE
Pull-Up option register	PUn	<PORT_base> + $400C_H$ + 40_H*n		—	PWE
	JPUn	<JPORT_base> + $400C_H$ + 40_H*n		—	—
Pull-Down option register	PDn	<PORT_base> + 4010_H + 40_H*n		—	PWE
	JPDn	<JPORT_base> + 4010_H + 40_H*n		—	—
Port Open Drain Control register	PODCn	<PORT_base> + 4014_H + 40_H*n		PWE	PWE
	APODCn	<PORT_base> + $4C94_H$ + 40_H*n		PWE	—
	JPODCn	<JPORT_base> + 4014_H + 40_H*n		—	—
Port Drive Strength Control register	PDSCn	<PORT_base> + 4018_H + 40_H*n		PWE	PWE
	APDSCn	<PORT_base> + $4C98_H$ + 40_H*n		PWE	—
	JPDSCn	<JPORT_base> + 4018_H + 40_H*n		—	—
Port Input buffer Selection register	PISn	<PORT_base> + $401C_H$ + 40_H*n		—	PWE
	JPISn	<JPORT_base> + $401C_H$ + 40_H*n		—	—
Port Input buffer Selection Advanced register	PISAn	<PORT_base> + 4024_H + 40_H*n		—	PWE
	JPISAn	<JPORT_base> + 4024_H + 40_H*n		—	—
Port Universal Characteristic Control register	PUCCn	<PORT_base> + 4028_H + 40_H*n		PWE	PWE
	JPUCCn	<JPORT_base> + 4028_H + 40_H*n		—	—
Port Open Drain Control Expansion register	PODCEn	<PORT_base> + 4038_H + 40_H*n		PWE	PWE
	APODCEn	<PORT_base> + $4CB8_H$ + 40_H*n		PWE	—
	JPODCEn	<JPORT_base> + 4038_H + 40_H*n		—	—
Port Control register	PCRn_m	<PORT_base> + 2000_H + 40_H*n + 4_H*m		PWE	PWE
	APCRn_m	<PORT_base> + $2C80_H$ + 40_H*n + 4_H*m		PWE	—
	JPCRn_m	<JPORT_base> + 2000_H + 40_H*n + 4_H*m		—	—
Port Safe State Control register	PSFCn	<PORT_base> + 6000_H + 40_H*n		PWE	PWE
	APSFCn	<PORT_base> + $6C80_H$ + 40_H*n		PWE	—
Port Safe State Trigger Selection register	PSFTSn	<PORT_base> + 6010_H + 40_H*n		PWE	PWE
	APSFTSn	<PORT_base> + $6C90_H$ + 40_H*n		PWE	—
Port Safe State Trigger Selection Expansion register	PSFTSEn	<PORT_base> + 6014_H + 40_H*n		PWE	PWE
	APSFTSEn	<PORT_base> + $6C94_H$ + 40_H*n		PWE	—
Port Keycode Protection Register	PKCPROT	<PORT_base> + $2F40_H$	PBG20#12	—	—
Port Write Enable register	PWE	<PORT_base> + $2F44_H$	PBG20#12	PKCPROT	PKCPROT
LVDS control A register	LVDSCTRLA	<PORT_base> + $2F50_H$	PBG20#12	PWE	—
LVDS control B register	LVDSCTRLB	<PORT_base> + $2F54_H$	PBG20#12	PWE	—

Note 1. P2 is intended to protect the LVDS outputs which belong to LVDVCC. P6 is intended to protect the $\overline{\text{RESETOUT}}$ output.

2.5.1 Pn/APn/JP0 — Port Register

This register defines port pins output levels for port output mode.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.13 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of pin m (m = 0 to 15): 0: Port pin drives low level 1: Port pin drives high level

NOTES

1. Reading Pn returns the register value independent from other register settings.
2. The value on this register bit is reflected to a pin level in the following conditions.
   Case : Port Mode (PMCn_m=0) & Output Mode (PMn_m=0)

2.5.2 PPRn/APPRn/JPPR0 — Port Pin Read Register

This register reflects the actual pin level when the input buffer is active.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.14 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	This register reflects the actual pin level when the input buffer is active 0: Port pin is at low level 1: Port pin is at high level

2.5.3 PMn/APMn/JPM0 — Port Mode Register

This register selects the pin direction as input or output.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.15 PMn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

2.5.4 PMCn/JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCn_15	PMCn_14	PMCn_13	PMCn_12	PMCn_11	PMCn_10	PMCn_9	PMCn_8	PMCn_7	PMCn_6	PMCn_5	PMCn_4	PMCn_3	PMCn_2	PMCn_1	PMCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.16 PMCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

2.5.5 PFCn/JPFC0 — Port Function Control Register

This register selects the alternative peripheral functions together with PFCEn, PFCAEn and PMn in Control Mode (PMCn = 1).

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCn_15	PFCn_14	PFCn_13	PFCn_12	PFCn_11	PFCn_10	PFCn_9	PFCn_8	PFCn_7	PFCn_6	PFCn_5	PFCn_4	PFCn_3	PFCn_2	PFCn_1	PFCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.17 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of a pin. See **Table 2.20, Setting Alternative Functions**.

2.5.6 PFCEn / JPFCE0 — Port Function Control Expansion Register

This register selects the alternative peripheral functions together with PFCn, PFCAEn and PMn in Control Mode (PMCn = 1).

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.18 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies an alternative function of a pin. See **Table 2.20, Setting Alternative Functions**.

2.5.7 PFCAEn — Port Function Control Additional Expansion Register

This register selects the alternative peripheral functions together with PFCn, PFCEn and PMn in Control Mode (PMCn = 1).

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAEn_15	PFCAEn_14	PFCAEn_13	PFCAEn_12	PFCAEn_11	PFCAEn_10	PFCAEn_9	PFCAEn_8	PFCAEn_7	PFCAEn_6	PFCAEn_5	PFCAEn_4	PFCAEn_3	PFCAEn_2	PFCAEn_1	PFCAEn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.19 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies an alternative function of a pin. See **Table 2.20** for details.

Table 2.20 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative peripheral function 1 (Control Mode 1) Input
			0	Alternative peripheral function 1 (Control Mode 1) Output
		1	1	Alternative peripheral function 2 (Control Mode 2) Input
			0	Alternative peripheral function 2 (Control Mode 2) Output
	1	0	1	Alternative peripheral function 3 (Control Mode 3) Input
			0	Alternative peripheral function 3 (Control Mode 3) Output
		1	1	Alternative peripheral function 4 (Control Mode 4) Input
			0	Alternative peripheral function 4 (Control Mode 4) Output
1	0	0	1	Alternative peripheral function 5 (Control Mode 5) Input
			0	Alternative peripheral function 5 (Control Mode 5) Output
		1	1	Alternative peripheral function 6 (Control Mode 6) Input
			0	Alternative peripheral function 6 (Control Mode 6) Output
	1	0	1	Alternative peripheral function 7 (Control Mode 7) Input
			0	Alternative peripheral function 7 (Control Mode 7) Output
		1	1	Alternative peripheral function 8 (Control Mode 8) Input
			0	Alternative peripheral function 8 (Control Mode 8) Output

2.5.8 PNOTn/APNOTn/JPNOT0 — Port NOT Register

This register provided a method to flip the bit values of Pn register. The bits of Pn register are flipped if the PNOTn register is written with the corresponding bit values being 1.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn_15	PNOTn_14	PNOTn_13	PNOTn_12	PNOTn_11	PNOTn_10	PNOTn_9	PNOTn_8	PNOTn_7	PNOTn_6	PNOTn_5	PNOTn_4	PNOTn_3	PNOTn_2	PNOTn_1	PNOTn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.21 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	0: No effect on the value of Pn_m bit 1: The value of Pn_m bit is flipped

2.5.9 PSRn/APSRn/JPSR0 — Port Set/Reset Register

This register provides an alternative method to write/read data on Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn_m is set by the corresponding bit in the lower 16 bits of PSRn.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.22 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m value is written to Pn_m. 0: Pn_m is not affected by PSRn_m 1: Pn_m is PSRn_m Example: When PSRn.PSRn_31 = 1, the value of PSRn.PSRn_15 is written to bit Pn.Pn_15 and output. When read, 0000_H is always returned.
15 to 0	PSRn_[15:0]	Sets the output level of pin Pn_m (m = 0 to 15). 0: Low level is written on Pn_m when it is enabled by PSRn_(m+16) 1: High level is written on Pn_m when it is enabled by PSRn_(m+16) When read, Pn register value is returned.

2.5.10 PMSRn/APMSRn/JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write/read data on PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn_m is set by the corresponding bit in the lower 16 bits of PMSRn.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.23 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Specifies whether the value of the corresponding lower bit PMSRn_m value is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m Example: When PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15. When read, 0000_H is always returned.
15 to 0	PMSRn_[15:0]	Data bits that specify the PMn_m value if the corresponding upper bit (PMSRn_[31:16]) PMSRn_m is 1. 0: PMn_m = 0 1: PMn_m = 1 When read, PMn register value is returned.

2.5.11 PMCSRn/JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn_31	PMCSRn_30	PMCSRn_29	PMCSRn_28	PMCSRn_27	PMCSRn_26	PMCSRn_25	PMCSRn_24	PMCSRn_23	PMCSRn_22	PMCSRn_21	PMCSRn_20	PMCSRn_19	PMCSRn_18	PMCSRn_17	PMCSRn_16
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn_15	PMCSRn_14	PMCSRn_13	PMCSRn_12	PMCSRn_11	PMCSRn_10	PMCSRn_9	PMCSRn_8	PMCSRn_7	PMCSRn_6	PMCSRn_5	PMCSRn_4	PMCSRn_3	PMCSRn_2	PMCSRn_1	PMCSRn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.24 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_ [31:16]	Specifies whether the value of the corresponding lower bit PMCSRn_m value is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m 1: PMCn_m is PMCSRn_m Example: When PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15. When read, 0000_H is always returned.
15 to 0	PMCSRn_ [15:0]	Data bits that specify the PMCn_m value if the corresponding upper bit (PMCSRn_[31:16]) PMCSRn_m is 1. 0: PMCn_m = 0 1: PMCn_m = 1 When read, PMCn register value is returned.

2.5.12 PINVn/APINVn/JPINV0 — Port Output value Inversion Register

This register inverts the output value of the port.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINVn_15	PINVn_14	PINVn_13	PINVn_12	PINVn_11	PINVn_10	PINVn_9	PINVn_8	PINVn_7	PINVn_6	PINVn_5	PINVn_4	PINVn_3	PINVn_2	PINVn_1	PINVn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.25 PINVn Register Contents

Bit Position	Bit Name	Function
15 to 0	PINVn_[15:0]	This register inverts the output value of the port. 0: No effect 1: Inverted value is output

2.5.13 PIBCn/APIBCn/JPIBC0 — Port Input Buffer Control Register

When a pin is being used in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables or disables the input buffer. However, when the pin is used as an input pin in S/W I/O control alternative mode (PMCn.PMCn_m = 1 and PIPCn.PIPCn_m = 0) or direct I/O control alternative mode (PMCn.PMCn_m = 1 and PIPCn.PIPCn_m = 1), set PIBCn.PIBCn_m = 0.

When pins are in bidirectional mode (PBDCn.PBDCn_m = 1), the shared output level loop-back function and pin output level-read function can be selected by setting PIBCn.PIBCn_m. Refer to **Section 2.5.14, PBDCn/APBDCn/JPBDC0 — Port Bi-Direction Control Register**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBCn_15	PIBCn_14	PIBCn_13	PIBCn_12	PIBCn_11	PIBCn_10	PIBCn_9	PIBCn_8	PIBCn_7	PIBCn_6	PIBCn_5	PIBCn_4	PIBCn_3	PIBCn_2	PIBCn_1	PIBCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.26 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer: 0: Input buffer is disabled. 1: Input buffer is enabled.

NOTES

1. To enable port pin's input buffer, the IO direction must be set as input mode by PM=1 during Port Mode (PMC=0).
2. By keeping this register at a reset value of 0, port pin's input buffer does not consumes current even when the pin level is at an intermediate voltage. When the input buffer is disabled, in port mode, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.
3. During this register set 1 with Port Mode (PMC=0), the values for peripheral macro are fixed.
4. During "Software I/O control alternative-function input" Mode (PMC=1, PM=1, PIPC=0), this register bit must be set to 0.

2.5.14 PBDCn/APBDCn/JPBDC0 — Port Bi-Direction Control Register

This register enables the input buffer when a pin is used in output mode, and permits bidirectional mode. The Pn_m pin level is read via PPRn.PPRn_m in bidirectional mode.

- Alternative output level loopback function
  When the Pn_m pin is used as the alternative output function, the actual pin output level based on the alternative output function can be looped back to the alternative input side by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 0. For example, the pin output level based on the first alternative function can be looped back to the same alternative input side. Also the pin output level can be read via PPRn.PPRn_m.
- Pin output level read function
  When the Pn_m pin is used as the general output port function or the alternative output function, the actual pin output level can be read via PPRn.PPRn_m by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 1. Under this setting, the pin output level will never be looped back to the alternative input side even in alternative output mode.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"
Address: Refer to **Table 2.12, Port Register Overview**
Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.27 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn_[15:0]	Enables/disables bi-direction mode of the corresponding pin. 0: Bi-direction mode disabled 1: Bi-direction mode enabled

NOTE

Loopback is enabled after four cycles of APB Low freq clock (CLK_LSB) since this register is written.

2.5.15 PIPCn — Port IP Control Register

This register specifies whether the I/O direction of pin Pn_m is controlled by the port mode register PMn.PMn_m or by an alternative function. If pin Pn_m is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn_m setting.

The list of alternative functions that require direct control of the I/O direction is **Table 2.29, Alternative functions require "Direct I/O Control" (Must Set PIPCn_m=1)**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.28 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O control mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

Table 2.29 Alternative functions require "Direct I/O Control" (Must Set PIPCn_m=1) (1/2)

Category	Pin Name	I/O	Function description
PIC	TAPA0UN	O	Motor control output U phase (negative)
	TAPA0UP	O	Motor control output U phase (positive)
	TAPA0VN	O	Motor control output V phase (negative)
	TAPA0VP	O	Motor control output V phase (positive)
	TAPA0WN	O	Motor control output W phase (negative)
	TAPA0WP	O	Motor control output W phase (positive)
TAUD	TAUD1O10	O	TAUD1 output for channel 10
	TAUD1O11	O	TAUD1 output for channel 11
	TAUD1O12	O	TAUD1 output for channel 12
	TAUD1O13	O	TAUD1 output for channel 13
	TAUD1O14	O	TAUD1 output for channel 14
	TAUD1O15	O	TAUD1 output for channel 15

Table 2.29 Alternative functions require “Direct I/O Control” (Must Set PIPCn_m=1) (2/2)

Category	Pin Name	I/O	Function description
TSG3	TSG30O1	O	TSG30 timer output 1
	TSG30O2	O	TSG30 timer output 2
	TSG30O3	O	TSG30 timer output 3
	TSG30O4	O	TSG30 timer output 4
	TSG30O5	O	TSG30 timer output 5
	TSG30O6	O	TSG30 timer output 6
	TSG31O1	O	TSG31 timer output 1
	TSG31O2	O	TSG31 timer output 2
	TSG31O3	O	TSG31 timer output 3
	TSG31O4	O	TSG31 timer output 4
	TSG31O5	O	TSG31 timer output 5
	TSG31O6	O	TSG31 timer output 6
SCI3	SCI3nRXD	I	SCI3n receive data
	SCI3nTXD	O	SCI3n transmit data
	SCI3nSCK	I/O	SCI3n serial clock input/output
MSPI	MSPInSO	O	MSPI serial data output
	MSPInDCS	I	MSPI data consistency check
GTM	GTMATnOm	O	GTM output signals for ATOM
	GTMATnOmN	O	GTM inverted timer output signals for ATOM
MMCA	MMCAnCMD	I/O	MMCA command/response
	MMCAnDATm	I/O	MMCA data
SFMA	SFMAnIOm	I/O	SFMA transmit/receive data
ETH	ETNB0MDIO	I/O	PHY management transfer data signal
GETH	ETNB1MDIO	I/O	PHY management transfer data signal

2.5.16 PUn/JPU0 — Pull-up Option Register

This register specifies whether pull-up resistor is connected to an input pin. For the electrical characteristics for each setting, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.30 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether a pull-up resistor is connected to the corresponding pin. 0: No pull-up resistor connected 1: Pull-up resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The pull-up resistor has no effect when the pin is operated in output mode.

2.5.17 PDn/JPD0 — Pull-down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin. For the electrical characteristics for each setting, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"
Address: Refer to **Table 2.12, Port Register Overview**
Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.31 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin: 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.

2.5.18 PODCn/APODCn/JPODC0 — Port Open-drain Control Register

This register selects push-pull or open-drain as the output buffer function.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.32 PODCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

2.5.19 PODCEn/APODCEn/JPODCE0 — Port Open-drain Control Expansion Register

This register selects the emulated P-channel Open-drain together with PODCn.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCEn_15	PODCEn_14	PODCEn_13	PODCEn_12	PODCEn_11	PODCEn_10	PODCEn_9	PODCEn_8	PODCEn_7	PODCEn_6	PODCEn_5	PODCEn_4	PODCEn_3	PODCEn_2	PODCEn_1	PODCEn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.33 PODCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PODCEn_ [15:0]	Refer to **Table 2.34, Port Open Drain Control Expansion** for the detailed operation on PODCE.

Table 2.34 Port Open Drain Control Expansion

PODCEn_m	PODCn_m	Function
0	0	Push-pull
0	1	Emulated N-channel Open Drain
1	0	Push-pull
1	1	Emulated P-channel Open Drain

2.5.20 PDSCn/APDSCn/JPDSC0 — Port Drive Strength Control Register

This register, together with the PUCCn registers, specifies the output buffer drive strength of a pin. Specific driving ability settings may be required, depending on the pin function to be used. For details, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSCn_15	PDSCn_14	PDSCn_13	PDSCn_12	PDSCn_11	PDSCn_10	PDSCn_9	PDSCn_8	PDSCn_7	PDSCn_6	PDSCn_5	PDSCn_4	PDSCn_3	PDSCn_2	PDSCn_1	PDSCn_0
Value after reset	*1	*1	*1	*1	*1	*1*2	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Note 2. PDSCn except PDSC6: 0
PDSC6: 1

Table 2.35 PDSCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. See **Table 2.37, Port Output Buffer drive Strength Selection**.

2.5.21 PUCCn/JPUCC0 — Port Universal Characteristic Control Register

This register, together with the PDSCn registers, specifies the output buffer drive strength of a pin. Specific driving ability settings may be required, depending on the pin function to be used. For details, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUCCn_15	PUCCn_14	PUCCn_13	PUCCn_12	PUCCn_11	PUCCn_10	PUCCn_9	PUCCn_8	PUCCn_7	PUCCn_6	PUCCn_5	PUCCn_4	PUCCn_3	PUCCn_2	PUCCn_1	PUCCn_0
Value after reset	*1	*1	*1	*1	*1	*1*2	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Note 2. PUCCn except PUCC6: 0
PUCC6: 1

Table 2.36 PUCCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUCCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. See **Table 2.37, Port Output Buffer drive Strength Selection**.

Table 2.37 Port Output Buffer drive Strength Selection

PUCCn_m	PDSCn_m	Output Buffer drive Strength
0	0	Drive strength = 5 (very low)
	1	Drive strength = 4 (low)
1	0	Drive strength = 3 (medium)
	1	Drive strength = 2 (high) Drive strength = 1*1 (very high)

Note 1. Only P2_0, P2_5, P22_0, P22_4

NOTES

1. For details of selectable input characteristics, refer to the appendix "E02_01_List_of_Pin_Assignment.xlsx".
2. JP0_1 is selected to "Drive strength = 3" when the pin is used as Nexus, LPD-4 pin and Boundary SCAN interface.
3. JP0_5 is selected to "Drive strength = 3" when the pin is used as Nexus, LPD-4 pin interface.

2.5.22 PISn/JPIS0 — Port input buffer selection register

This register and the PISAn specifies the input buffer characteristics. Specific input characteristic settings may be required, depending on the pin function to be used. For details, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.38 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer characteristic. See **Table 2.40, Port Input buffer Characteristics Selection**.

2.5.23 PISAn/JPISA0 – Port Input buffer Selection Advanced register

This register and the PISn specifies the input buffer characteristics. Specific input characteristic settings may be required, depending on the pin function to be used. For details, refer to **Section 55, Electrical Characteristics**.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISAn_15	PISAn_14	PISAn_13	PISAn_12	PISAn_11	PISAn_10	PISAn_9	PISAn_8	PISAn_7	PISAn_6	PISAn_5	PISAn_4	PISAn_3	PISAn_2	PISAn_1	PISAn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.39 PISAn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISAn_[15:0]	Specifies the input buffer characteristic: See **Table 2.40**.

Table 2.40 Port Input buffer Characteristics Selection

PISAn_m	PISn_m	Function
0	0	SHMT1 input buffer is selected
	1	SHMT4 input buffer is selected
1	X	TTL input buffer is selected

NOTES

1. For details of selectable input characteristics, refer to the appendix "E02_01_List_of_Pin_Assignment.xlsx".
2. Input buffer type of JP0_0, JP0_2 and JP0_3 is selected to TTL type when these pins are used as Nexus, LPD-4 pin and Boundary SCAN interface.

2.5.24 PCRn_m/APCRn_m/JPCR0_m — Port Control Register

By going through this register, it is possible to have access to the registers of each port group, and the individual pins is specified all functions by 1 PCR register setting.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PINV	—	PODC	PODCE	—	PUCC	PDSC	—	PISA	—	PIS	PU	PD	PBDC	PIBC
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	P	—	—	—	PPR	—	PMC	PIPC	PM	—	PFCAE	PFCE	PFC
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.41 PCRn_m Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	PINV	Same as the m bit of PINVn/APINVn/JPINV0 register.
29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	PODC	Same as the m bit of PODCn/APODCn/JPODC0 register.
27	PODCE	Same as the m bit of PODCEn/APODCEn/JPODCE0 register.
26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	PUCC	Same as the m bit of PUCCn/JPUCC0 register.
24	PDSC	Same as the m bit of PDSCn/APDSCn/JPDSC0 register.
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	PISA	Same as the m bit of PISAn/JPISA0 register.
21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	PIS	Same as the m bit of PISn/JPIS0 register.
19	PU	Same as the m bit of PUn/JPU0 register.
18	PD	Same as the m bit of PDn/JPD0 register.
17	PBDC	Same as the m bit of PBDCn/APBDCn/JPBDC0 register.
16	PIBC	Same as the m bit of PIBCn/APIBCn/JPIBC0 register.
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	P	Same as the m bit of Pn/APn/JP0 register.
11 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PPR	Same as the m bit of PPRn/APPRn/JPPR0 register.

Table 2.41 PCRn_m Register Contents (2/2)

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PMC	Same as the m bit of PMCn/JPMC0 register.
5	PIPC	Same as the m bit of PIPCn register.
4	PM	Same as the m bit of PMn/APMn/JPM0 register.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PFCAE	Same as the m bit of PFCAEn register.
1	PFCE	Same as the m bit of PFCEn/JPFCE0 register.
0	PFC	Same as the m bit of PFCn/JPFC0 register.

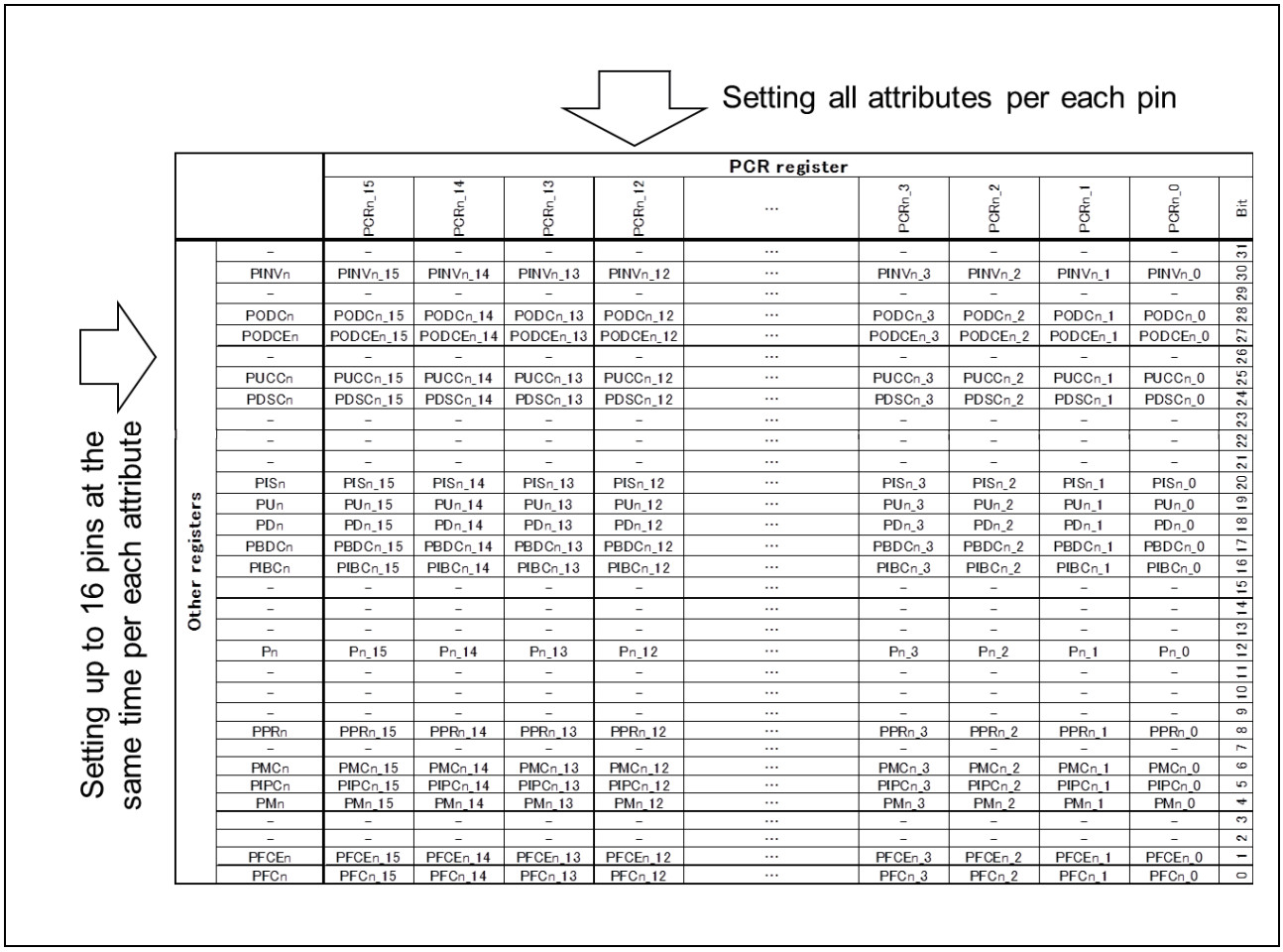


Figure 2.6 Relationship between Other Registers and PCR(Port Control Register)

2.5.25 PSFCn/APSFCn — Port Safe State Control register

This register controls Port Safe State function.

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFCn_15	PSFCn_14	PSFCn_13	PSFCn_12	PSFCn_11	PSFCn_10	PSFCn_9	PSFCn_8	PSFCn_7	PSFCn_6	PSFCn_5	PSFCn_4	PSFCn_3	PSFCn_2	PSFCn_1	PSFCn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.42 PSFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFCn_[15:0]	Control Port Safe State function: 0: Port Safe State function is disabled 1: Port Safe State function is enabled

2.5.26 PSFTSn/APSFTSn — Port Error Trigger Selection register

This register selects the defined application in Port Safe State mode (PSFCn_m = 1).

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFTSn_15	PSFTSn_14	PSFTSn_13	PSFTSn_12	PSFTSn_11	PSFTSn_10	PSFTSn_9	PSFTSn_8	PSFTSn_7	PSFTSn_6	PSFTSn_5	PSFTSn_4	PSFTSn_3	PSFTSn_2	PSFTSn_1	PSFTSn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.43 PSFTSn Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFTSn_[15:0]	Select Error trigger for Port Safe State function. See **Table 2.10, Configuration for Port Safe State Function**.

2.5.27 PSFTSEn/APSFTSEn — Port Error Trigger Selection Expansion register

This register and the PSFTSn select the defined application in Port Safe State mode (PSFCn_m = 1).

Access: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Address: Refer to **Table 2.12, Port Register Overview**

Value after reset: Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSFTSEn_15	PSFTSEn_14	PSFTSEn_13	PSFTSEn_12	PSFTSEn_11	PSFTSEn_10	PSFTSEn_9	PSFTSEn_8	PSFTSEn_7	PSFTSEn_6	PSFTSEn_5	PSFTSEn_4	PSFTSEn_3	PSFTSEn_2	PSFTSEn_1	PSFTSEn_0
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Refer to Appendix "*E02-02_List of Port Functions.xlsx*"

Table 2.44 PSFTSEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PSFTSEn_[15:0]	Select Error trigger for Port Safe State function. See **Table 2.10, Configuration for Port Safe State Function**.

2.5.28 PKCPROT — Port Keycode Protection Register

This register is used for protection against writing operation to the PWE register.

Access: This register can be read or written in 32-bit units.

Address: <PORT_base> + $2F40_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PKCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PKCPROT[15:1]															PWEE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 2.45 PKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	PKCPROT[31:1]	Enable or disable modification of the PWEE bit. The value written is not retained. These bits are always read as 0.*1
0	PWEE	PWE Enable bit 0: Disables write access of PWE register 1: Enables write access of PWE register

Note 1. Write $A5A5A500_H$ to this register to disable writes to PWE register.
Write $A5A5A501_H$ to this register to enable writes to PWE register.

2.5.29 PWE — Port Write Enable register

This register is used to control which protected registers could be written. When PKCPROT.PWEE = 1, this register could be written. When PKCPROT.PWEE = 0, this register could not be written. For the target register, refer to **Table 2.12, Port Register Overview**.

Access: This register can be read or written in 32-bit units.

Address: <PORT_base> + $2F44_H$

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LVDSCTRLAB	PWEAP5*1	PWEAP4*4	PWEAP3*3	PWEAP2	PWEAP1*3	PWEAP0	PWEP24	PWEP23*1*2	PWEP22*3	PWEP21*5	PWEP20
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWEP19*1*2	PWEP18*1*2	PWEP17	PWEP12*1*2	PWEP11*1	PWEP10	PWEP9*1*2	PWEP8*1	PWEP6	PWEP5	PWEP4	PWEP3	PWEP2	PWEP1*1*2	PWEP0*1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 2.46 PWE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	LVDSCTRLAB	Port Write Enable for LVDSCTRLA and LVDSCTRLB 0: Disables write access 1: Enables write access
26	PWEAP5*1	Port Write Enable for AP5 0: Disables write access 1: Enables write access
25	PWEAP4*4	Port Write Enable for AP4 0: Disables write access 1: Enables write access
24	PWEAP3*3	Port Write Enable for AP3 0: Disables write access 1: Enables write access
23	PWEAP2	Port Write Enable for AP2 0: Disables write access 1: Enables write access
22	PWEAP1*3	Port Write Enable for AP1 0: Disables write access 1: Enables write access
21	PWEAP0	Port Write Enable for AP0 0: Disables write access 1: Enables write access
20	PWEP24	Port Write Enable for P24 0: Disables write access 1: Enables write access
19	PWEP23*1*2	Port Write Enable for P23 0: Disables write access 1: Enables write access
18	PWEP22*3	Port Write Enable for P22 0: Disables write access 1: Enables write access
17	PWEP21*5	Port Write Enable for P21 0: Disables write access 1: Enables write access

Table 2.46 PWE Register Contents (2/2)

Bit Position	Bit Name	Function
16	PWEP20	Port Write Enable for P20 0: Disables write access 1: Enables write access
15	PWEP19*1*2	Port Write Enable for P19 0: Disables write access 1: Enables write access
14	PWEP18*1*2	Port Write Enable for P18 0: Disables write access 1: Enables write access
13	PWEP17	Port Write Enable for P17 0: Disables write access 1: Enables write access
12	PWEP12*1*2	Port Write Enable for P12 0: Disables write access 1: Enables write access
11	PWEP11*1	Port Write Enable for P11 0: Disables write access 1: Enables write access
10	PWEP10	Port Write Enable for P10 0: Disables write access 1: Enables write access
9	PWEP9*1*2	Port Write Enable for P9 0: Disables write access 1: Enables write access
8	PWEP8*1	Port Write Enable for P8 0: Disables write access 1: Enables write access
7	PWEP6	Port Write Enable for P6 0: Disables write access 1: Enables write access
6	PWEP5	Port Write Enable for P5 0: Disables write access 1: Enables write access
5	PWEP4	Port Write Enable for P4 0: Disables write access 1: Enables write access
4	PWEP3	Port Write Enable for P3 0: Disables write access 1: Enables write access
3	PWEP2	Port Write Enable for P2 0: Disables write access 1: Enables write access
2	PWEP1*1*2	Port Write Enable for P1 0: Disables write access 1: Enables write access
1	PWEP0*1	Port Write Enable for P0 0: Disables write access 1: Enables write access
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. Not support in BGA292, BGA156, QFP176, QFP144
Note 2. Not support in BGA373
Note 3. Not support in QFP176, BGA156, QFP144
Note 4. Not support in BGA156, QFP144
Note 5. Not support in QFP144

2.5.30 LVDSCTRLA — LVDS control A register

This register specifies whether to use LVDS function.

Access: This register can be read or written in 32-bit units.

Address: <PORT_base> + $2F50_H$

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	LVDSCTRLA20	—	—	—	LVDSCTRLA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	LVDSCTRLA5	LVDSCTRLA4	LVDSCTRLA3	—	—	LVDSCTRLA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W

Table 2.47 LVDSCTRLA Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	LVDSCTRLA20	Specify the voltage for the LVDS function: 0: LVDS function of P2_12, P2_13, P2_14 and P2_15 is used in 5V mode 1: LVDS function of P2_12, P2_13, P2_14 and P2_15 is used in 3V mode **CAUTION** Be sure to specify the same values with LVDSCTRLA[16].
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	LVDSCTRLA16	Specify the voltage for the LVDS input function: 0: LVDS input function of P2_10 and P2_11 is used in 5V mode 1: LVDS input function of P2_10 and P2_11 is used in 3V mode **CAUTION** Be sure to specify the same values with LVDSCTRLA[20].
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LVDSCTRLA5	Specify whether to use LVDS function: 0: Disable LVDS output function of P2_14 and P2_15 1: Enable LVDS output function of P2_14 and P2_15
4	LVDSCTRLA4	Specify whether to use LVDS function: 0: Disable LVDS input function of P2_14 and P2_15 1: Enable LVDS input function of P2_14 and P2_15
3	LVDSCTRLA3	Specify whether to use the LVDS function: 0: Disable LVDS output function of P2_12 and P2_13 1: Enable LVDS output function of P2_12 and P2_13
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVDSCTRLA0	Specify whether to use LVDS function: 0: Disable LVDS input function of P2_10 and P2_11 1: Enable LVDS input function of P2_10 and P2_11

2.5.31 LVDSCTRLB — LVDS control B register

This register specifies whether to use LVDS function.

Access: This register can be read or written in 32-bit units.

Address: <PORT_base> + $2F54_H$

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	LVDSCTRLB20	—	—	—	LVDSCTRLB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	LVDSCTRLB5	LVDSCTRLB4	LVDSCTRLB3	—	—	LVDSCTRLB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W

Table 2.48 LVDSCTRLB Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	LVDSCTRLB20	Specify the voltage for the LVDS function: 0: LVDS function of P4_7, P4_8, P4_9 and P4_10 is used in 5V mode 1: LVDS function of P4_7, P4_8, P4_9 and P4_10 is used in 3V mode **CAUTION** Be sure to specify the same values with LVDSCTRLB[16].
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	LVDSCTRLB16	Specify the voltage for the LVDS input function: 0: LVDS input function of P4_5 and P4_6 is used in 5V mode 1: LVDS input function of P4_5 and P4_6 is used in 3V mode **CAUTION** Be sure to specify the same values with LVDSCTRLB[20].
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LVDSCTRLB5	Specify whether to use the LVDS function: 0: Disable LVDS output function of P4_9 and P4_10 1: Enable LVDS output function of P4_9 and P4_10
4	LVDSCTRLB4	Specify whether to use the LVDS function: 0: Disable LVDS input function of P4_9 and P4_10 1: Enable LVDS input function of P4_9 and P4_10
3	LVDSCTRLB3	Specify whether to use the LVDS function: 0: Disable LVDS output function of P4_7 and P4_8 1: Enable LVDS output function of P4_7 and P4_8
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVDSCTRLB0	Specify whether to use LVDS function: 0: Disable LVDS input function of P4_5 and P4_6 1: Enable LVDS input function of P4_5 and P4_6

2.6 Port Setting Flow Example

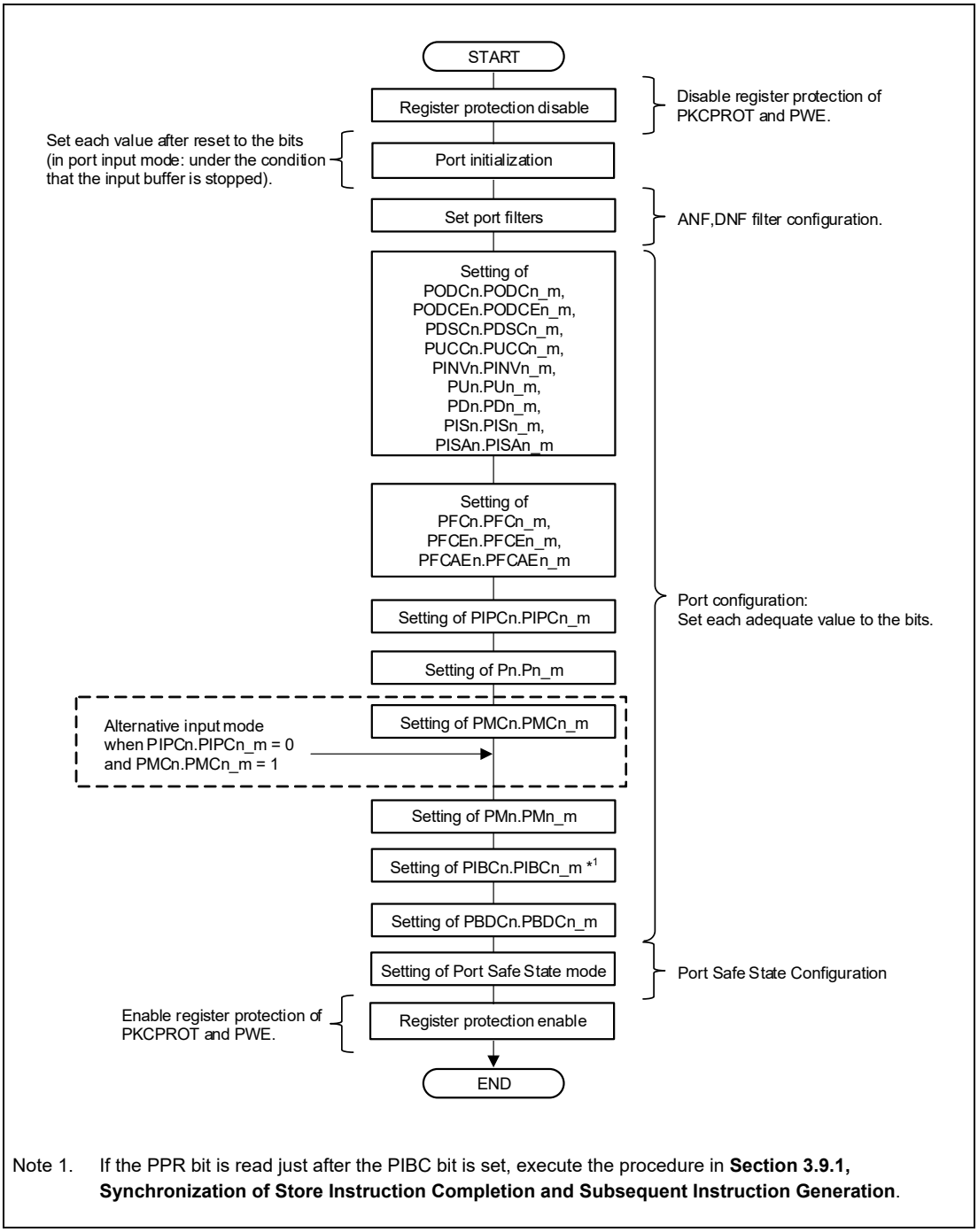
Port setting flow examples are shown in this section.

CAUTIONS

1. **When a pin is set to an alternative output mode while the PIPCn.PIPCn_m bit is 0, it may be temporarily switched to an alternative input mode. This occurs during the period between setting of the PMCn.PMCn_m bit to 1 and setting of the PMn.PMn_m bit to 0. Due to this possibility of a pin being temporarily switched to an alternative input mode, if a pin has an interrupt related signal as an alternative function, disable the interrupt or make sure that it is ignored. Also, when using ERRORINx (x = 0 to 3), set port later than the pins which other ERRORINx (x = 0 to 3) are assigned.**
2. **In PCR configuration, do not change the value of the invert value register (PINVn_m) and output value register (Pn_m) at the same time after output enable is active. If these registers are changed, set them again from the start of the register setting flow.**
3. **When returning to the "Port initializing" state after port setting has been configured, set port register again in the inverse order with Port Group Unit register. At that case, it cannot be used with PCR configuration.**

2.6.1 Batch Setting

The following figure indicates an example of setting a port group collectively.



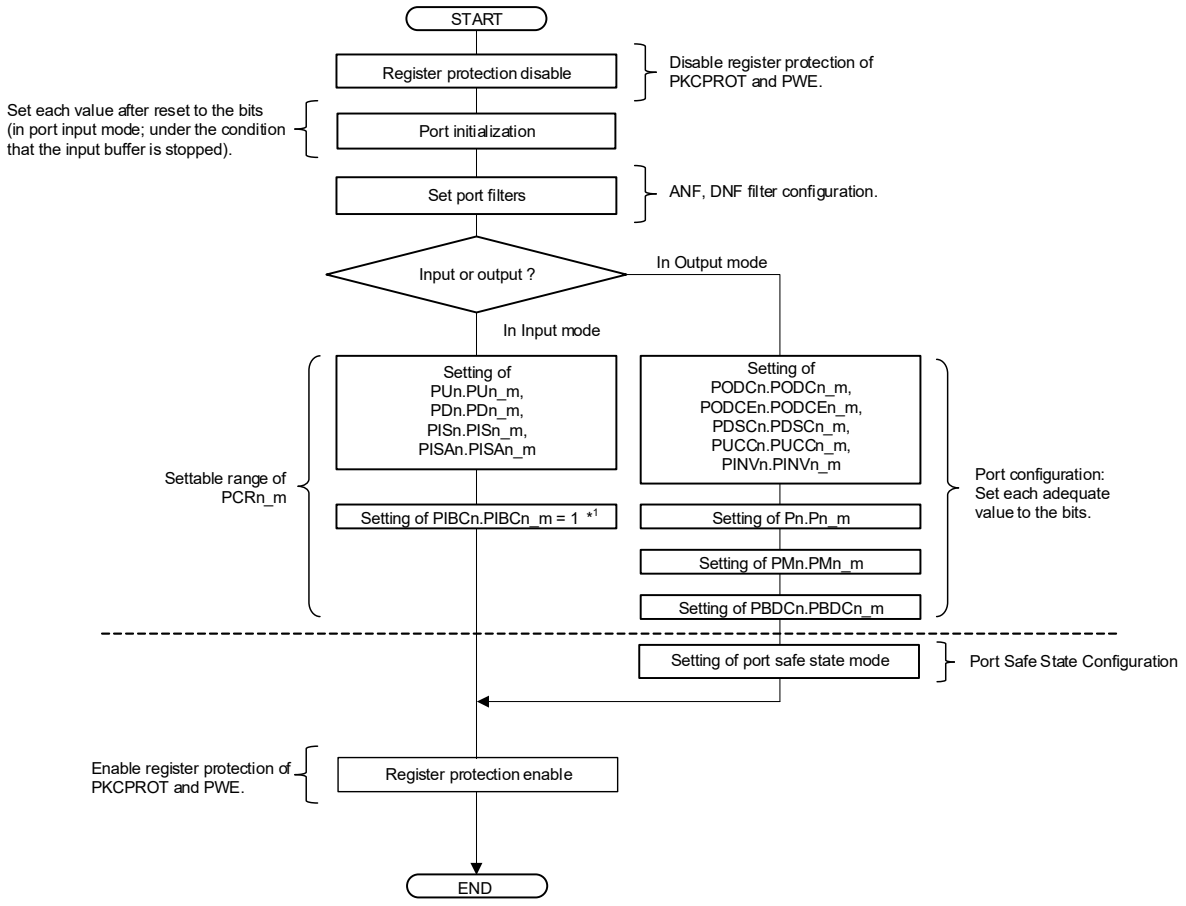
Note 1. If the PPR bit is read just after the PIBC bit is set, execute the procedure in **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation**.

Figure 2.7 Port setting flow example (setting collectively)

2.6.2 Individual Setting

The following figure indicates an example of setting an individual port.

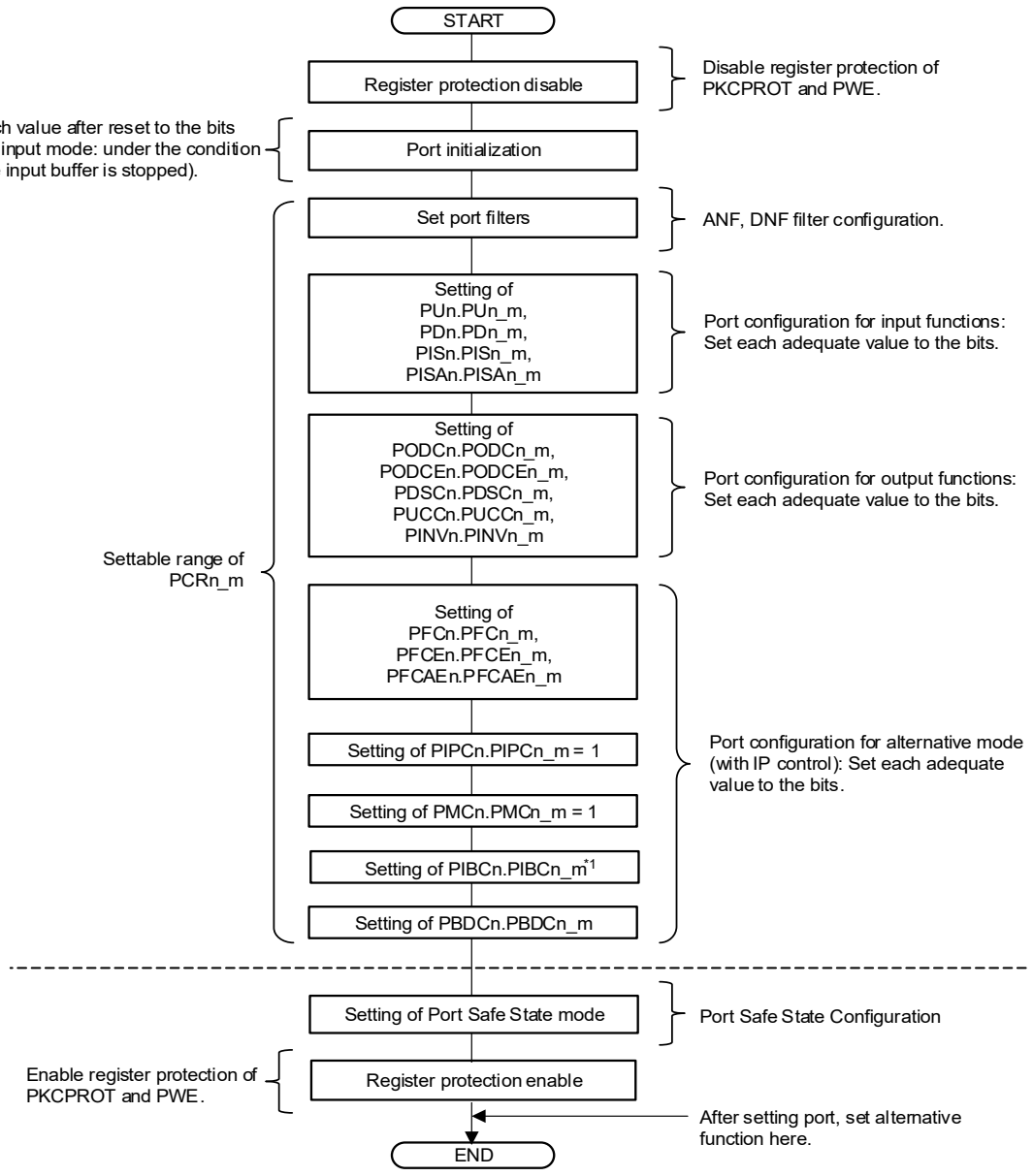
By going through this register, it is possible to have access to the registers of each port group, and the individual pins is specified all functions by 1 PCR register setting.



Note: The individual pin state changes less than 100ns after setting this flow.

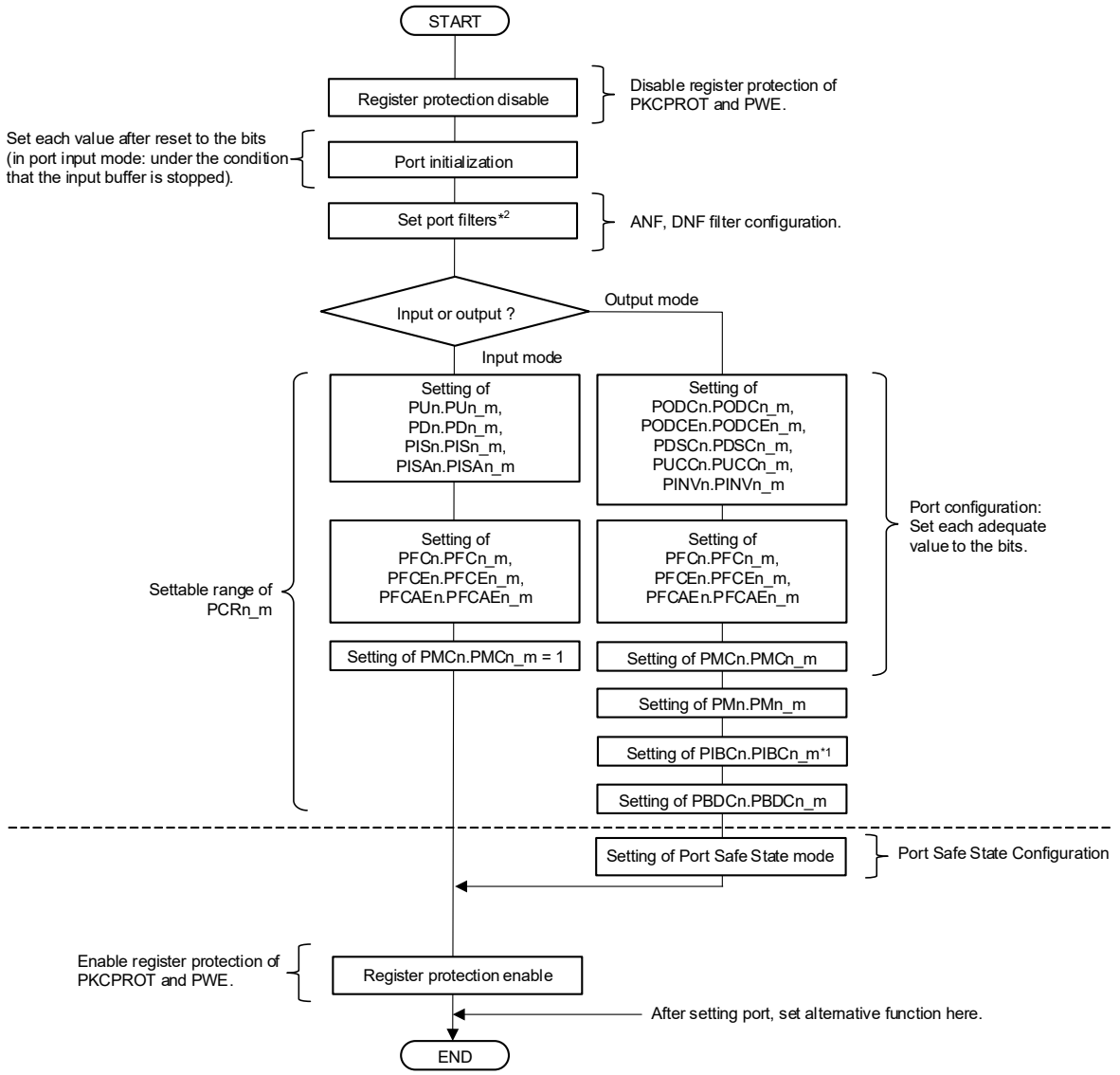
Note 1. If the PPR bit is read just after the PIBC bit is set, execute the procedure in **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation**.

Figure 2.8 Port setting flow example (Port mode)



Note 1. If the PPR bit is read just after the PIBC bit is set, execute the procedure in **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation**.

Figure 2.9 Port setting flow example (Alternative mode with IP control)



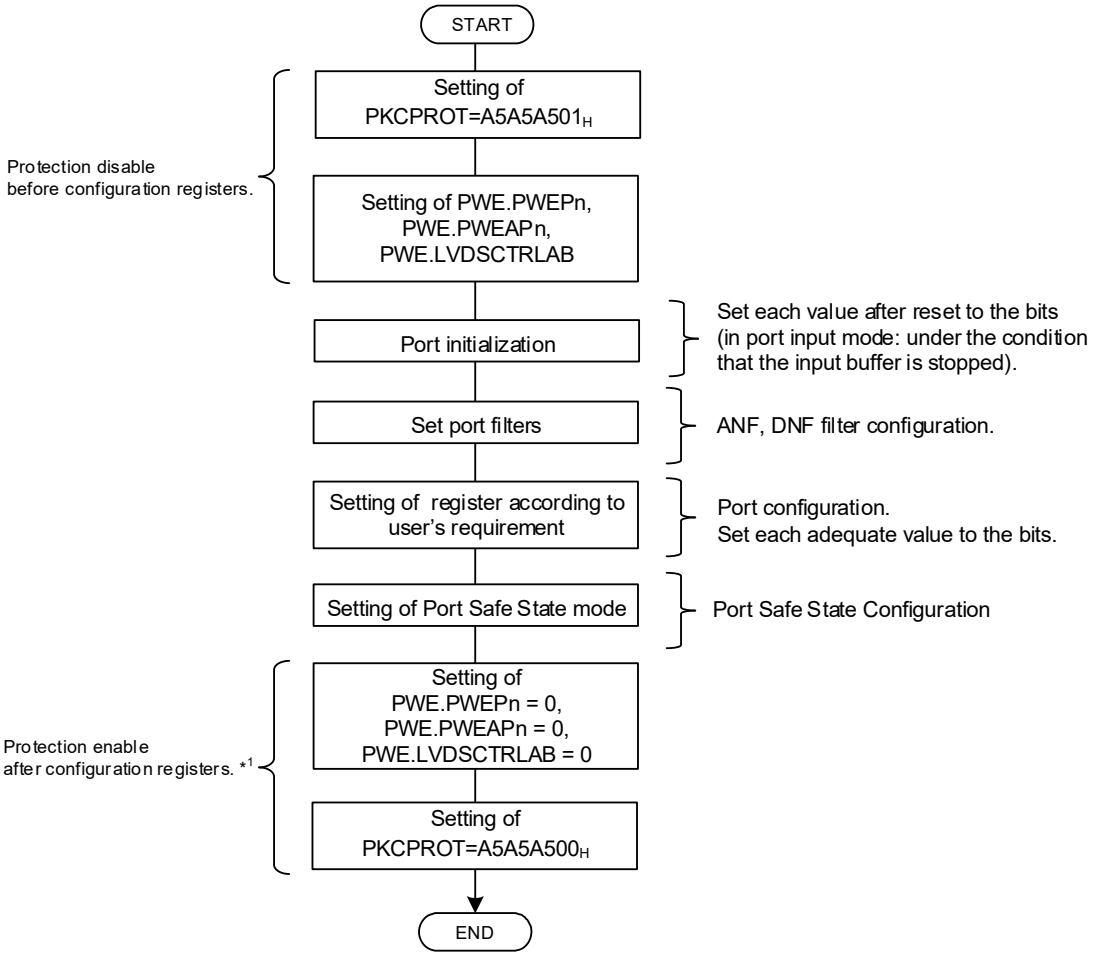
Note 1. If the PPR bit is read just after the PIBC bit is set, execute the procedure in **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation**.

Note 2. In order to avoid the unintended interrupt occurrence, NMI and INTPx (x = 0 to 39) should be set according to **Section 6.5.3, External Interrupt Processing Flow**.

Figure 2.10 Port setting flow example (Alternative mode without IP control)

2.6.3 Protection Setting

The following shows an example of setting register protection.



Note 1. Disable the protection sequence of PKCPROT and PWE as needed. (If not needed, this sequence can be skipped.) When PKCPROT is set to $A5A5A500_H$ to disable write access but PWE remains set to 1, access to the specific Pn group registers remains enabled.

Figure 2.11 Protection setting flow

2.6.4 LVDS Function Setting

The following shows the flow to use LVDS function.

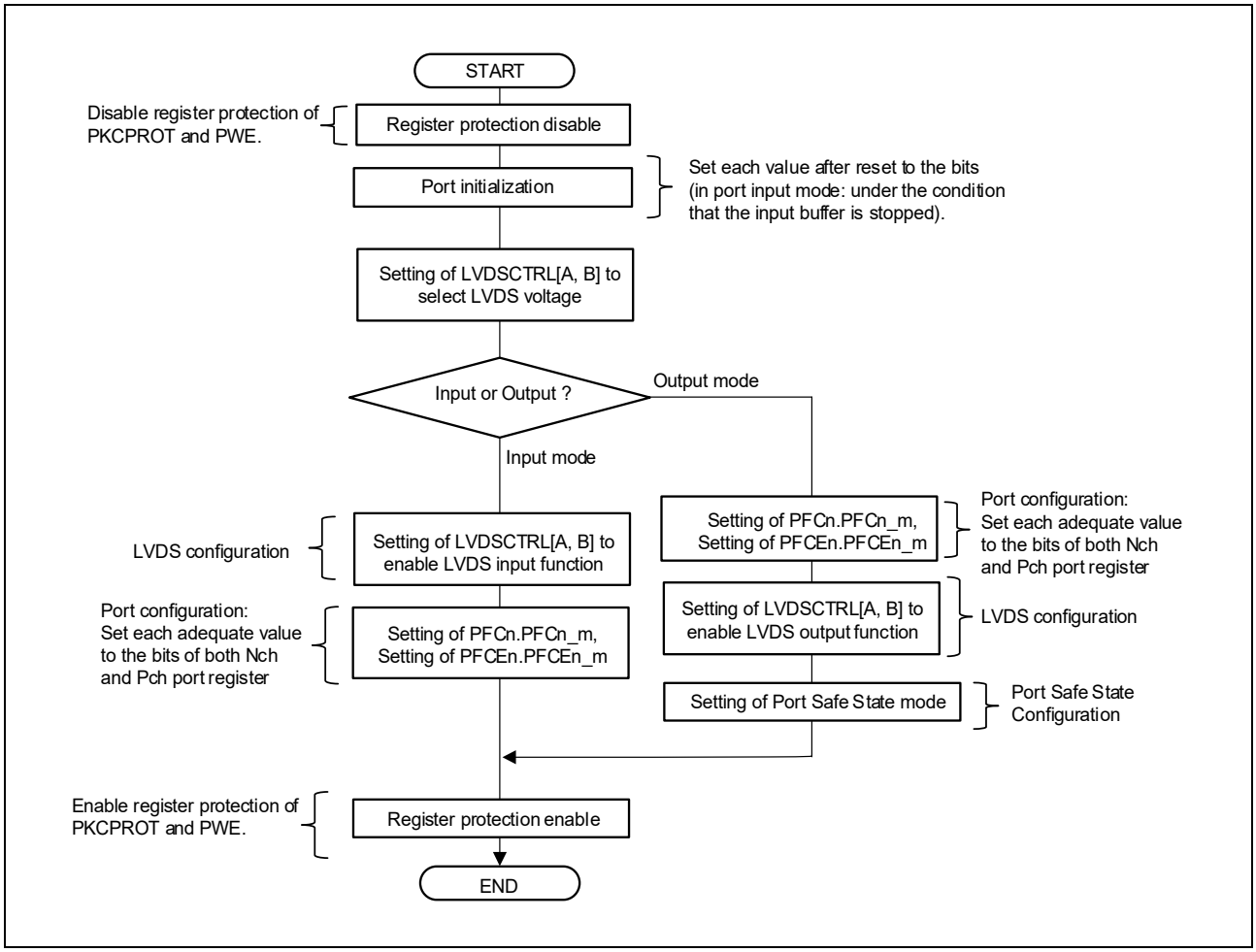
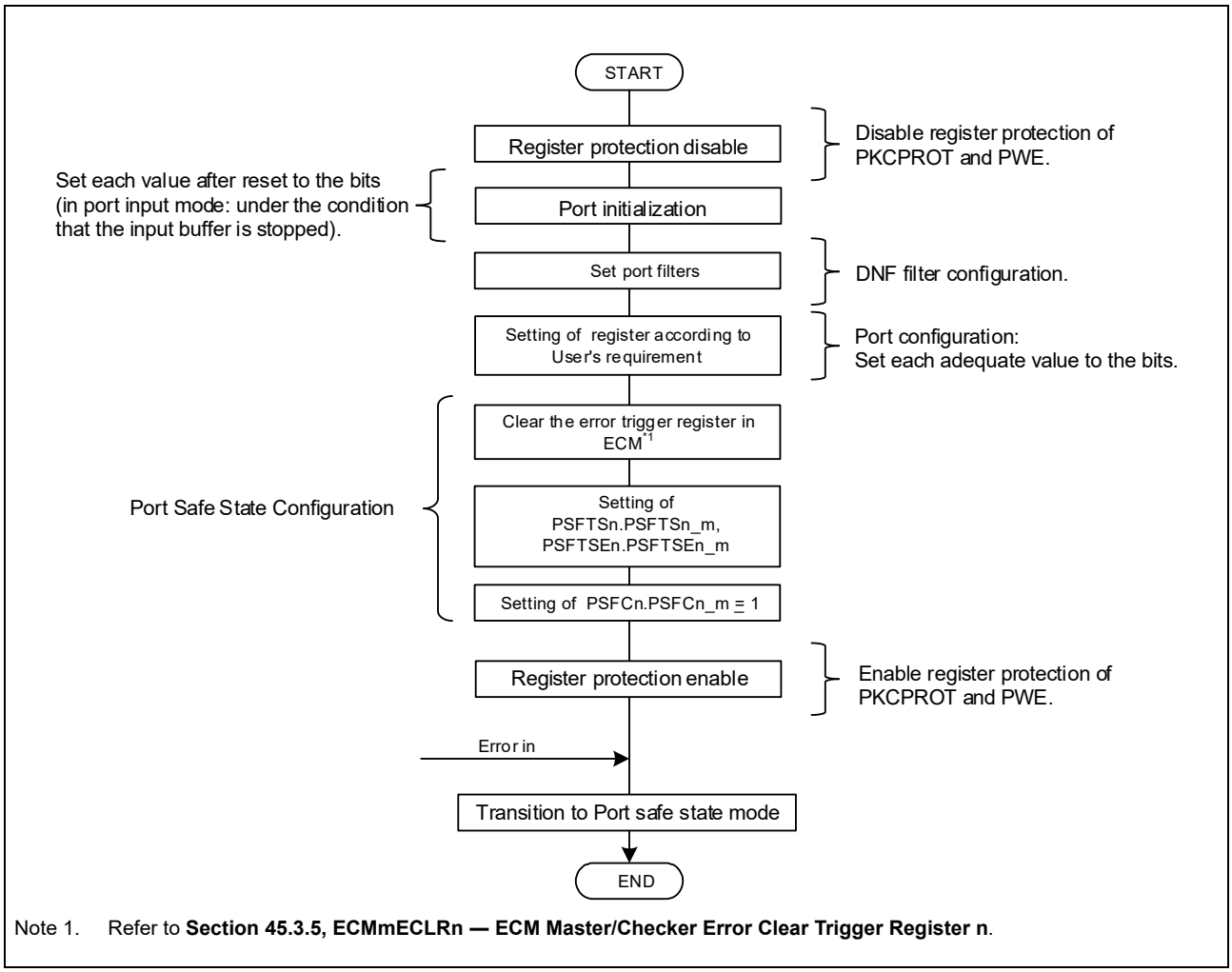


Figure 2.12 LVDS function setting flow example

2.6.5 Port Safe State Function Setting

The following shows the flow to use port safe state mode. If the function is not used, port safe state configuration sequence can be skipped.



Note 1. Refer to **Section 45.3.5, ECMmECLRn — ECM Master/Checker Error Clear Trigger Register n**.

Figure 2.13 Port safe state function setting flow

2.7 Noise Filter & Edge/Level Detector

The input signals at some pins are passed through a filter to remove noise and glitches. This device supports both analog and digital filters. It also supports the function for edge and level detection after the signals have passed through a filter. The first part of this section provides an overview of port input signals that are equipped with a filter and the filter type, noise filter & edge detection control registers and bits, and register addresses.

2.7.1 Noise Filter Assignment

The noise filter type of each function is shown below.

Table 2.49 Noise filter Assignment (1/5)

Function		Noise Filter									Detector Type				Clock Supply
			Filter Control Register		Digital noise elimination control register		Digital noise elimination enable register								
Group	Name	Filter type	Register	Address	Register	Address	Register	Address	Control bit		Rise	Fall	Both	Level	Sampling clock
System Control	$\overline{\text{RESET}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{TRST}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	FLMD0	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	FLMD1	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	FLMD2	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	MODE0	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{AURORES1}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{AURORES2}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{AURORESPD}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{ERAMRESPD}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
	$\overline{\text{ERAMRES2}}$	ANF (Type B1)	–	–	–	–	–	–	–		–	–	–	–	–
Wakeup (RLIN)	INTDNFA1WUF0 (RLIN30RX)	DNF (Type C2)	FCLACTL0_ RLIN3	FFED 4000$_H$	DNFACTL_ RLIN3	FFED 0000$_H$	DNFAEN_ RLIN3 (DNFAENL_ RLIN3)	FFED 0004$_H$ (FFED 000C$_H$)	NFENL0		√	√	√	–	CLK_LSB
	INTDNFA1WUF1 (RLIN31RX)		FCLACTL1_ RLIN3	FFED 4004$_H$					NFENL1		√	√	√	–	
Wakeup (RSCAN)	INTDNFA2WUF0 (CAN0RX)	DNF (Type C2)	FCLACTL0_ RSCAN	FFED 4100$_H$	DNFACTL_ RSCAN	FFED 0100$_H$	DNFAEN_ RSCAN (DNFAENL_ RSCAN)	FFED 0104$_H$ (FFED 010C$_H$)	NFENL0		√	√	√	–	CLK_LSB
	INTDNFA2WUF1 (CAN1RX)		FCLACTL1_ RSCAN	FFED 4104$_H$					NFENL1		√	√	√	–	
Wakeup (FLX)	INTDNFA0WUF0 (FLX0RXDA)	DNF (Type C2)	FCLACTL0_ FLXA	FFED 4200$_H$	DNFACTL_ FLXA	FFED 0200$_H$	DNFAEN_ FLXA (DNFAENL_ FLXA)	FFED 0204$_H$ (FFED 020C$_H$)	NFENL0		√	√	√	–	CLK_LSB
	INTDNFA0WUF1 (FLX1RXDA)		FCLACTL1_ FLXA	FFED 4204$_H$					NFENL1		√	√	√	–	
SENT	SENT0RX	ANF/DNF (Type E1)	FCLACTL0_ RSENT	FFED 4300$_H$	DNFACTL_ RSENT	FFED 0300$_H$	DNFAEN_ RSENT (DNFAENL_ RSENT)	FFED 0304$_H$ (FFED 030C$_H$)	NFENL0		–	–	–	–	CLK_HSB
	SENT1RX		FCLACTL1_ RSENT	FFED 4304$_H$					NFENL1		–	–	–	–	
	SENT2RX		FCLACTL2_ RSENT	FFED 4308$_H$					NFENL2		–	–	–	–	
	SENT3RX		FCLACTL3_ RSENT	FFED 430C$_H$					NFENL3		–	–	–	–	
	SENT4RX		FCLACTL4_ RSENT	FFED 4310$_H$					NFENL4		–	–	–	–	
	SENT5RX		FCLACTL5_ RSENT	FFED 4314$_H$					NFENL5		–	–	–	–	
	SENT6RX		FCLACTL6_ RSENT	FFED 4318$_H$					NFENL6		–	–	–	–	
	SENT7RX		FCLACTL7_ RSENT	FFED 431C$_H$					NFENL7		–	–	–	–	

Table 2.49 Noise filter Assignment (2/5)

Function		Noise Filter								Detector Type				Clock Supply
			Filter Control Register		Digital noise elimination control register		Digital noise elimination enable register							
Group	Name	Filter type	Register	Address	Register	Address	Register	Address	Control bit	Rise	Fall	Both	Level	Sampling clock
PSI5	PSI50RX	ANF/DNF (Type E1)	FCLACTL0_PSI5	FFED 4400$_H$	DNFACTL_PSI5	FFED 0400$_H$	DNFAEN_PSI5 (DNFAENL_PSI5)	FFED 0404$_H$ (FFED 040C$_H$)	NFENL0	–	–	–	–	CLK_HSB
	PSI51RX		FCLACTL1_PSI5	FFED 4404$_H$					NFENL1	–	–	–	–	
	PSI52RX		FCLACTL2_PSI5	FFED 4408$_H$					NFENL2	–	–	–	–	
	PSI53RX		FCLACTL3_PSI5	FFED 440C$_H$					NFENL3	–	–	–	–	
PIC	INTTAPA0ESO / TAPA0ESO	ANF/DNF (Type F1)	FCLACTL0_TAPA	FFED 4500$_H$	DNFACTL_TAPA	FFED 0500$_H$	DNFAEN_TAPA (DNFAENL_TAPA)	FFED 0504$_H$ (FFED 050C$_H$)	NFENL0	√	√	√	–	CLK_HSB
	INTTAPA1ESO / TAPA1ESO		FCLACTL1_TAPA	FFED 4504$_H$					NFENL1	√	√	√	–	
ADCJ	ADCJ0TRG0	DNF (Type D1)	–	–	DNFACTL_ADCJ0	FFED 0600$_H$	DNFAEN_ADCJ0 (DNFAENL_ADCJ0)	FFED 0604$_H$ (FFED 060C$_H$)	NFENL0	*1	*1	*1	–	CLK_ADC
	ADCJ0TRG1		–	–					NFENL1	*1	*1	*1	–	
	ADCJ0TRG2		–	–					NFENL2	*1	*1	*1	–	
	ADCJ0TRG3		–	–					NFENL3	*1	*1	*1	–	
	ADCJ0TRG4		–	–					NFENL4	*1	*1	*1	–	
	ADCJ1TRG0	DNF (Type D1)	–	–	DNFACTL_ADCJ1	FFED 0700$_H$	DNFAEN_ADCJ1 (DNFAENL_ADCJ1)	FFED 0704$_H$ (FFED 070C$_H$)	NFENL0	*1	*1	*1	–	
	ADCJ1TRG1		–	–					NFENL1	*1	*1	*1	–	
	ADCJ1TRG2		–	–					NFENL2	*1	*1	*1	–	
	ADCJ1TRG3		–	–					NFENL3	*1	*1	*1	–	
	ADCJ1TRG4		–	–					NFENL4	*1	*1	*1	–	
	ADCJ2TRG0	DNF (Type C1)	FCLACTL0_ADCJ2	FFED 4C00$_H$	DNFACTL_ADCJ2	FFED 0800$_H$	DNFAEN_ADCJ2 (DNFAENL_ADCJ2)	FFED 0804$_H$ (FFED 080C$_H$)	NFENL0	√	√	√	–	CLKA_ADC
	ADCJ2TRG1		FCLACTL1_ADCJ2	FFED 4C04$_H$					NFENL1	√	√	√	–	
	ADCJ2TRG2		FCLACTL2_ADCJ2	FFED 4C08$_H$					NFENL2	√	√	√	–	
	ADCJ2TRG3		FCLACTL3_ADCJ2	FFED 4C0C$_H$					NFENL3	√	√	√	–	
	ADCJ2TRG4		FCLACTL4_ADCJ2	FFED 4C10$_H$					NFENL4	√	√	√	–	
ENCA	ENCA0TIN0	DNF (Type D1)	–	–	DNFACTL_ENCA	FFED 0900$_H$	DNFAEN_ENCA (DNFAENL_ENCA)	FFED 0904$_H$ (FFED 090C$_H$)	NFENL0	*2	*2	*2	–	CLK_HSB
	ENCA0TIN1		–	–					NFENL1	*2	*2	*2	–	
	ENCA1TIN0		–	–					NFENL2	*2	*2	*2	–	
	ENCA1TIN1		–	–					NFENL3	*2	*2	*2	–	
TAUD	TAUD0I0	DNF (Type D1)	–	–	DNFACTL_TAUD0	FFED 0A00$_H$	DNFAEN_TAUD0 (DNFAENH_TAUD0/ DNFAENL_TAUD0)	FFED 0A04$_H$ (FFED 0A08$_H$/ FFED 0A0C$_H$)	NFENL0	*3	*3	*3	–	CLK_HSB
	TAUD0I1		–	–					NFENL1	*3	*3	*3	–	
	TAUD0I2		–	–					NFENL2	*3	*3	*3	–	
	TAUD0I3		–	–					NFENL3	*3	*3	*3	–	
	TAUD0I4		–	–					NFENL4	*3	*3	*3	–	
	TAUD0I5		–	–					NFENL5	*3	*3	*3	–	
	TAUD0I6		–	–					NFENL6	*3	*3	*3	–	
	TAUD0I7		–	–					NFENL7	*3	*3	*3	–	
	TAUD0I8		–	–					NFENH0	*3	*3	*3	–	
	TAUD0I9		–	–					NFENH1	*3	*3	*3	–	
	TAUD0I10		–	–					NFENH2	*3	*3	*3	–	
	TAUD0I11		–	–					NFENH3	*3	*3	*3	–	
	TAUD0I12		–	–					NFENH4	*3	*3	*3	–	
	TAUD0I13		–	–					NFENH5	*3	*3	*3	–	
	TAUD0I14		–	–					NFENH6	*3	*3	*3	–	
	TAUD0I15		–	–					NFENH7	*3	*3	*3	–	

Table 2.49 Noise filter Assignment (3/5)

Function		Noise Filter								Detector Type				Clock Supply
			Filter Control Register		Digital noise elimination control register		Digital noise elimination enable register							
Group	Name	Filter type	Register	Address	Register	Address	Register	Address	Control bit	Rise	Fall	Both	Level	Sampling clock
TAUD	TAUD1I0	DNF (Type D1)	–	–	DNFACTL_TAUD1	FFED 0B00$_H$	DNFAEN_TAUD1 (DNFAENH_TAUD1/ DNFAENL_TAUD1)	FFED 0B04$_H$ (FFED 0B08$_H$/ FFED 0B0C$_H$)	NFENL0	*3	*3	*3	–	CLK_HSB
	TAUD1I1		–	–					NFENL1	*3	*3	*3	–	
	TAUD1I2		–	–					NFENL2	*3	*3	*3	–	
	TAUD1I3		–	–					NFENL3	*3	*3	*3	–	
	TAUD1I4		–	–					NFENL4	*3	*3	*3	–	
	TAUD1I5		–	–					NFENL5	*3	*3	*3	–	
	TAUD1I6		–	–					NFENL6	*3	*3	*3	–	
	TAUD1I7		–	–					NFENL7	*3	*3	*3	–	
	TAUD1I8		–	–					NFENH0	*3	*3	*3	–	
	TAUD1I9		–	–					NFENH1	*3	*3	*3	–	
	TAUD1I10		–	–					NFENH2	*3	*3	*3	–	
	TAUD1I11		–	–					NFENH3	*3	*3	*3	–	
	TAUD1I12		–	–					NFENH4	*3	*3	*3	–	
	TAUD1I13		–	–					NFENH5	*3	*3	*3	–	
	TAUD1I14		–	–					NFENH6	*3	*3	*3	–	
	TAUD1I15		–	–					NFENH7	*3	*3	*3	–	
	TAUD2I0		–	–	DNFACTL_TAUD2	FFBF 6900$_H$	DNFAEN_TAUD2 (DNFAENH_TAUD2/ DNFAENL_TAUD2)	FFBF 6904$_H$ (FFBF 6908$_H$/ FFBF 690C$_H$)	NFENL0	*3	*3	*3	–	
	TAUD2I1		–	–					NFENL1	*3	*3	*3	–	
	TAUD2I2		–	–					NFENL2	*3	*3	*3	–	
	TAUD2I3		–	–					NFENL3	*3	*3	*3	–	
	TAUD2I4		–	–					NFENL4	*3	*3	*3	–	
	TAUD2I5		–	–					NFENL5	*3	*3	*3	–	
	TAUD2I6		–	–					NFENL6	*3	*3	*3	–	
	TAUD2I7		–	–					NFENL7	*3	*3	*3	–	
	TAUD2I8		–	–					NFENH0	*3	*3	*3	–	
	TAUD2I9		–	–					NFENH1	*3	*3	*3	–	
	TAUD2I10		–	–					NFENH2	*3	*3	*3	–	
	TAUD2I11		–	–					NFENH3	*3	*3	*3	–	
	TAUD2I12		–	–					NFENH4	*3	*3	*3	–	
	TAUD2I13		–	–					NFENH5	*3	*3	*3	–	
	TAUD2I14		–	–					NFENH6	*3	*3	*3	–	
	TAUD2I15		–	–					NFENH7	*3	*3	*3	–	
TSG3	TSG30PTSI0/ ENCA0E0	DNF (Type D1)	–	–	DNFACTL_TSG3	FFED 0C00$_H$	DNFAEN_TSG3 (DNFAENL_TSG3)	FFED 0C04$_H$ (FFED 0C0C$_H$)	NFENL0	*4	*4	*4	–	CLK_HSB
	TSG30PTSI1/ ENCA0E1		–	–					NFENL1	*4	*4	*4	–	
	TSG30PTSI2/ ENCA0EC		–	–					NFENL2	*4	*4	*4	–	
	TSG30CLKI	DNF (Type D2)	–	–					NFENL3	–	–	–	–	
	TSG31PTSI0/ ENCA1E0	DNF (Type D1)	–	–					NFENL4	*4	*4	*4	–	
	TSG31PTSI1/ ENCA1E1		–	–					NFENL5	*4	*4	*4	–	
	TSG31PTSI2/ ENCA1EC		–	–					NFENL6	*4	*4	*4	–	
	TSG31CLKI	DNF (Type D2)	–	–					NFENL7	–	–	–	–	

Table 2.49 Noise filter Assignment (4/5)

Function		Noise Filter								Detector Type				Clock Supply
			Filter Control Register		Digital noise elimination control register		Digital noise elimination enable register							
Group	Name	Filter type	Register	Address	Register	Address	Register	Address	Control bit	Rise	Fall	Both	Level	Sampling clock
Interrupt	NMI	ANF (Type A1)	FCLACTL0_NMI	FFED 4600$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP0	ANF (Type A1)	FCLACTL0_INTP_0	FFED 4700$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP1	ANF (Type A1)	FCLACTL1_INTP_0	FFED 4704$_H$	–	–	–	–	–	√	√	√	√	
	INTP2	ANF (Type A1)	FCLACTL2_INTP_0	FFED 4708$_H$	–	–	–	–	–	√	√	√	√	
	INTP3	ANF (Type A1)	FCLACTL3_INTP_0	FFED 470C$_H$	–	–	–	–	–	√	√	√	√	
	INTP4	ANF (Type A1)	FCLACTL4_INTP_0	FFED 4710$_H$	–	–	–	–	–	√	√	√	√	
	INTP5	ANF (Type A1)	FCLACTL5_INTP_0	FFED 4714$_H$	–	–	–	–	–	√	√	√	√	
	INTP6	ANF (Type A1)	FCLACTL6_INTP_0	FFED 4718$_H$	–	–	–	–	–	√	√	√	√	
	INTP7	ANF (Type A1)	FCLACTL7_INTP_0	FFED 471C$_H$	–	–	–	–	–	√	√	√	√	
	INTP8	ANF (Type A1)	FCLACTL0_INTP_1	FFED 4800$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP9	ANF (Type A1)	FCLACTL1_INTP_1	FFED 4804$_H$	–	–	–	–	–	√	√	√	√	
	INTP10	ANF (Type A1)	FCLACTL2_INTP_1	FFED 4808$_H$	–	–	–	–	–	√	√	√	√	
	INTP11	ANF (Type A1)	FCLACTL3_INTP_1	FFED 480C$_H$	–	–	–	–	–	√	√	√	√	
	INTP12	ANF (Type A1)	FCLACTL4_INTP_1	FFED 4810$_H$	–	–	–	–	–	√	√	√	√	
	INTP13	ANF (Type A1)	FCLACTL5_INTP_1	FFED 4814$_H$	–	–	–	–	–	√	√	√	√	
	INTP14	ANF (Type A1)	FCLACTL6_INTP_1	FFED 4818$_H$	–	–	–	–	–	√	√	√	√	
	INTP15	ANF (Type A1)	FCLACTL7_INTP_1	FFED 481C$_H$	–	–	–	–	–	√	√	√	√	
	INTP16	ANF (Type A1)	FCLACTL0_INTP_2	FFED 4900$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP17	ANF (Type A1)	FCLACTL1_INTP_2	FFED 4904$_H$	–	–	–	–	–	√	√	√	√	
	INTP18	ANF (Type A1)	FCLACTL2_INTP_2	FFED 4908$_H$	–	–	–	–	–	√	√	√	√	
	INTP19	ANF (Type A1)	FCLACTL3_INTP_2	FFED 490C$_H$	–	–	–	–	–	√	√	√	√	
	INTP20	ANF (Type A1)	FCLACTL4_INTP_2	FFED 4910$_H$	–	–	–	–	–	√	√	√	√	
	INTP21	ANF (Type A1)	FCLACTL5_INTP_2	FFED 4914$_H$	–	–	–	–	–	√	√	√	√	
	INTP22	ANF (Type A1)	FCLACTL6_INTP_2	FFED 4918$_H$	–	–	–	–	–	√	√	√	√	
	INTP23	ANF (Type A1)	FCLACTL7_INTP_2	FFED 491C$_H$	–	–	–	–	–	√	√	√	√	
	INTP24	ANF (Type A1)	FCLACTL0_INTP_3	FFED 4A00$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP25	ANF (Type A1)	FCLACTL1_INTP_3	FFED 4A04$_H$	–	–	–	–	–	√	√	√	√	
	INTP26	ANF (Type A1)	FCLACTL2_INTP_3	FFED 4A08$_H$	–	–	–	–	–	√	√	√	√	
	INTP27	ANF (Type A1)	FCLACTL3_INTP_3	FFED 4A0C$_H$	–	–	–	–	–	√	√	√	√	
	INTP28	ANF (Type A1)	FCLACTL4_INTP_3	FFED 4A10$_H$	–	–	–	–	–	√	√	√	√	
	INTP29	ANF (Type A1)	FCLACTL5_INTP_3	FFED 4A14$_H$	–	–	–	–	–	√	√	√	√	

Table 2.49 Noise filter Assignment (5/5)

Function		Noise Filter								Detector Type				Clock Supply
			Filter Control Register		Digital noise elimination control register		Digital noise elimination enable register							
Group	Name	Filter type	Register	Address	Register	Address	Register	Address	Control bit	Rise	Fall	Both	Level	Sampling clock
Interrupt	INTP30	ANF (Type A1)	FCLACTL6_INTP_3	FFED 4A18$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP31	ANF (Type A1)	FCLACTL7_INTP_3	FFED 4A1C$_H$	–	–	–	–	–	√	√	√	√	
	INTP32	ANF (Type A1)	FCLACTL0_INTP_4	FFED 4B00$_H$	–	–	–	–	–	√	√	√	√	CLKA_LPS
	INTP33	ANF (Type A1)	FCLACTL1_INTP_4	FFED 4B04$_H$	–	–	–	–	–	√	√	√	√	
	INTP34	ANF (Type A1)	FCLACTL2_INTP_4	FFED 4B08$_H$	–	–	–	–	–	√	√	√	√	
	INTP35	ANF (Type A1)	FCLACTL3_INTP_4	FFED 4B0C$_H$	–	–	–	–	–	√	√	√	√	
	INTP36	ANF (Type A1)	FCLACTL4_INTP_4	FFED 4B10$_H$	–	–	–	–	–	√	√	√	√	
	INTP37	ANF (Type A1)	FCLACTL5_INTP_4	FFED 4B14$_H$	–	–	–	–	–	√	√	√	√	
	INTP38	ANF (Type A1)	FCLACTL6_INTP_4	FFED 4B18$_H$	–	–	–	–	–	√	√	√	√	
	INTP39	ANF (Type A1)	FCLACTL7_INTP_4	FFED 4B1C$_H$	–	–	–	–	–	√	√	√	√	
TAUJ	TAUJ0I0	ANF/DNF (Type G1)	–	–	DNFACTL_TAUJ0	FFED 0E00$_H$	DNFAEN_TAUJ0 (DNFAENL_TAUJ0)	FFED 0E04$_H$ (FFED 0E0C$_H$)	NFENL0	*5	*5	*5	–	CLK_HSB
	TAUJ0I1		–	–					NFENL1	*5	*5	*5	–	
	TAUJ0I2		–	–					NFENL2	*5	*5	*5	–	
	TAUJ0I3		–	–					NFENL3	*5	*5	*5	–	
	TAUJ1I0	ANF/DNF (Type G1)	–	–	DNFACTL_TAUJ1	FFED 0F00$_H$	DNFAEN_TAUJ1 (DNFAENL_TAUJ1)	FFED 0F04$_H$ (FFED 0F0C$_H$)	NFENL0	*5	*5	*5	–	CLK_HSB
	TAUJ1I1		–	–					NFENL1	*5	*5	*5	–	
	TAUJ1I2		–	–					NFENL2	*5	*5	*5	–	
	TAUJ1I3		–	–					NFENL3	*5	*5	*5	–	
	TAUJ2I0	ANF/DNF (Type G1)	–	–	DNFACTL_TAUJ2	FFED 1000$_H$	DNFAEN_TAUJ2 (DNFAENL_TAUJ2)	FFED 1004$_H$ (FFED 100C$_H$)	NFENL0	*5	*5	*5	–	CLKA_TAUJ
	TAUJ2I1		–	–					NFENL1	*5	*5	*5	–	
	TAUJ2I2		–	–					NFENL2	*5	*5	*5	–	
	TAUJ2I3		–	–					NFENL3	*5	*5	*5	–	
	TAUJ3I0	ANF/DNF (Type G1)	–	–	DNFACTL_TAUJ3	FFED 1100$_H$	DNFAEN_TAUJ3 (DNFAENL_TAUJ3)	FFED 1104$_H$ (FFED 110C$_H$)	NFENL0	*5	*5	*5	–	CLKA_TAUJ
	TAUJ3I1		–	–					NFENL1	*5	*5	*5	–	
	TAUJ3I2		–	–					NFENL2	*5	*5	*5	–	
	TAUJ3I3		–	–					NFENL3	*5	*5	*5	–	
ECM	$\overline{\text{ERRORIN0}}$	DNF (Type D2)	–	–	DNFACTL_ECM	FFED 0D00$_H$	DNFAEN_ECM (DNFAENL_ECM)	FFED 0D04$_H$ (FFED 0D0C$_H$)	NFENL0	–	–	–	–	CLK_LSB
	$\overline{\text{ERRORIN1}}$		–	–					NFENL1	–	–	–	–	
	$\overline{\text{ERRORIN2}}$		–	–					NFENL2	–	–	–	–	
	$\overline{\text{ERRORIN3}}$		–	–					NFENL3	–	–	–	–	

Note 1. For the setting for ADCJ0/ADCJ1 edge detection, see **Section 41.3.2.3, PIC2ADCJnEDGSEL — A/D Converter Trigger Edge Control Register (n = 0, 1)**.

Note 2. For the setting for ENCAn edge detection, see **Section 40.3.3, ENCAnIOC0 — ENCA I/O Control Register 0**.

Note 3. For the setting for TAUDn edge detection, see **Section 33.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register m**.

Note 4. For the setting for ENCAn edge detection, see **Section 40.3.4, ENCAnIOC1 — ENCA I/O Control Register 1**. And TSG3 do not have edge detector

Note 5. For the setting for TAUJn edge detection, see **Section 34.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register m**.

2.7.2 Description of Port Noise Filter & Edge/Level Detection

2.7.2.1 Analog Filters

(1) Analog Filter Characteristic

Analog filters have fixed characteristics. See **Section 55, Electrical Characteristics** for the input conditions for signals input to pins that incorporate an analog filter.

(2) Analog Filter Control Registers

A dedicated FCLACTLm_<name> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Section 2.7.1, Noise Filter Assignment**.

CAUTIONS

1. **When the output signal from the analog filter is set to an input for an alternative function, allow at least the period of pulse rejection to elapse after the port pin switch to the alternative function.**
   The period of pulse rejection is referred to Section 55 Electrical Characteristics.
2. **When enabling the edge detector in the analog filter, allow at least the period of pulse rejection to elapse after the port pin switch to the alternative function, and then set the edge detection mode.**

2.7.2.2 Digital Filters

(1) Digital Filter Characteristic

Digital filters allow adjusting the filter characteristics to the needs of the application.

The input signal is sampled with the sampling frequency f_s.

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- DNFACTL_<name>.PRS[2:0] allows to select the sampling frequency to

  $f_s = f_{supply}/2^{PRS[2:0]}$

  where f_{supply} is the sampling clock supply frequency. About “Sampling clock supply” of each DNF, refer to **Section 2.7.1, Noise Filter Assignment**.

- DNFACTL_<name>.NFSTS[1:0] determines the number of level samples, “s”, (2 to 5):

  $s = NFSTS[1:0] + 2$

External signal pulses shorter than the following are always suppressed.

$s \times 1/f_s$

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

$(s + 1) \times 1/f_s$

External signal pulses in the following range may be suppressed or judged as valid.

$s \times 1/f_s$ to $(s + 1) \times 1/f_s$

The filter operation is illustrated in the figure below with DNFACTL_<name>.NFSTS[1:0]= 01_B, i.e. s = 3 same level samples.

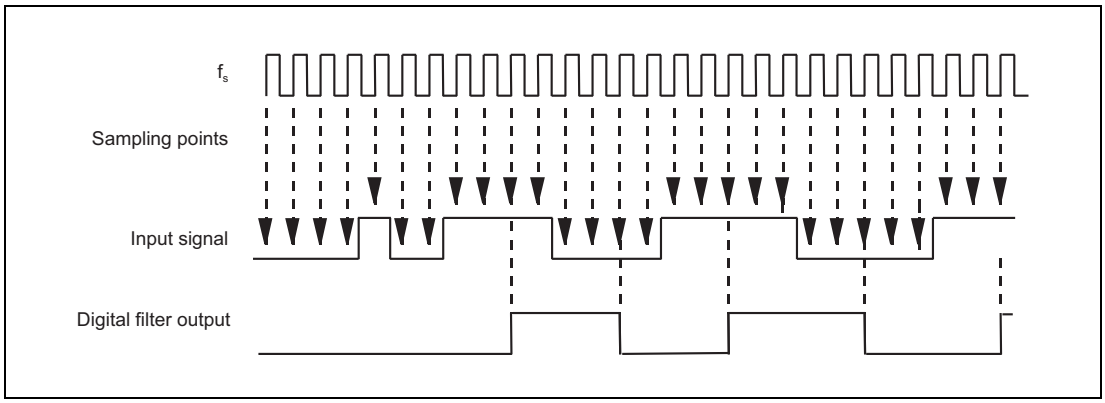


Figure 2.14 Digital Filter Function

(2) Digital Filter Groups

The input signals processed through digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by DNFACTL_<name>.PRS[2:0] and NFSTS[1:0] apply to the signals.

However, the digital filter for each signal can be enabled or disabled separately by DNFAEN_<name>.NFENHm (m = 0 to 7) and DNFAEN_<name>.NFENLm (m = 0 to 7).

CAUTIONS

1. **When the output signal from the digital filter is set to an input for an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFAEN_<name>.NFENHm (m = 0 to 7) = 1 and DNFAEN_<name>.NFENLm (m = 0 to 7) = 1) for the port pin to switch to the alternative function.**

   s = DNFACTL_<name>.NFSTS[1:0] + 2

   $s \times 1/f_s + 2 \times 1/f_{supply}$

2. **When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFAEN_<name>.NFENHm (m = 0 to 7) = 1 and DNFAEN_<name>.NFENLm (m = 0 to 7) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.**

   $s \times 1/f_s + 3 \times 1/f_{supply}$

(3) Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFACTL_<name> and digital noise elimination enable register DNFAEN_<name> are used to set all the filters in the same group (<name> = peripheral function group).

DNFACTL_<name> register specified he characteristics of the digital noise elimination filter for the digital filter of <name>.

DNFAEN_<name> register enables/disables each filter by setting the corresponding bit in DNFAEN_<name>.NFENHm (m = 0 to 7) and DNFAEN_<name>.NFENLm (m = 0 to 7).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The assignment of the input signals to the control registers and their addresses are given in **Section 2.7.1, Noise Filter Assignment**.

CAUTION

Do not change any control register settings, while the concerned digital filter is enabled by DNFAEN_<name>.NFENHm (m = 0 to 7) = 1 and DNFAEN_<name>.NFENLm (m = 0 to 7) = 1. Otherwise an unintended filter output may be generated.

2.7.2.3 ANF Type A1

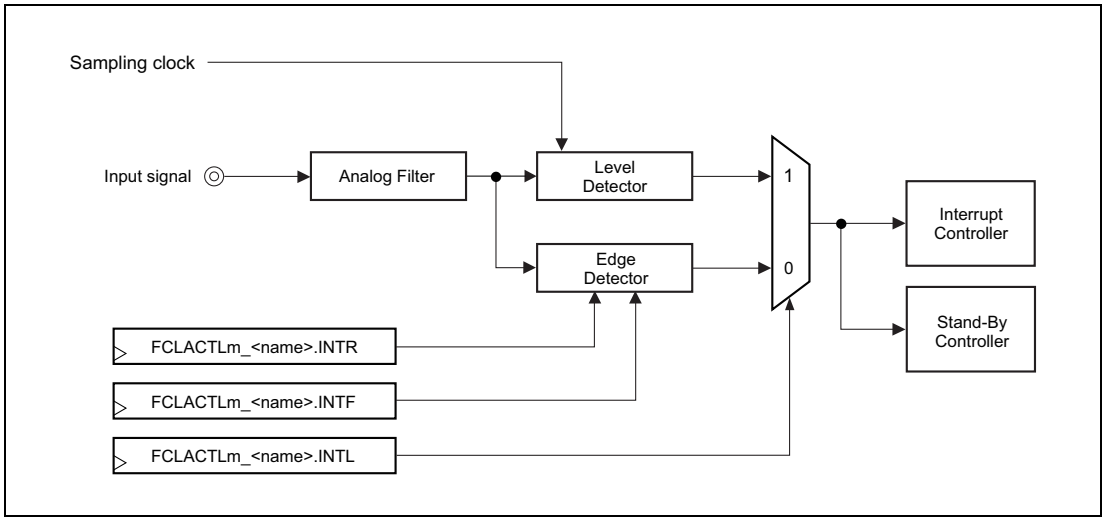


Figure 2.15 Block diagram of ANF type A1

2.7.2.4 ANF Type B1

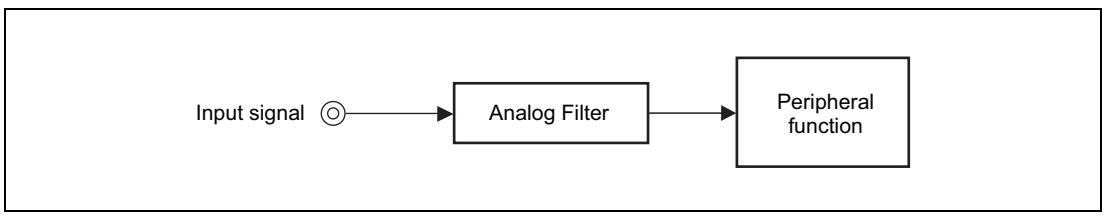


Figure 2.16 Block diagram of ANF type B1

2.7.2.5 DNF Type C1

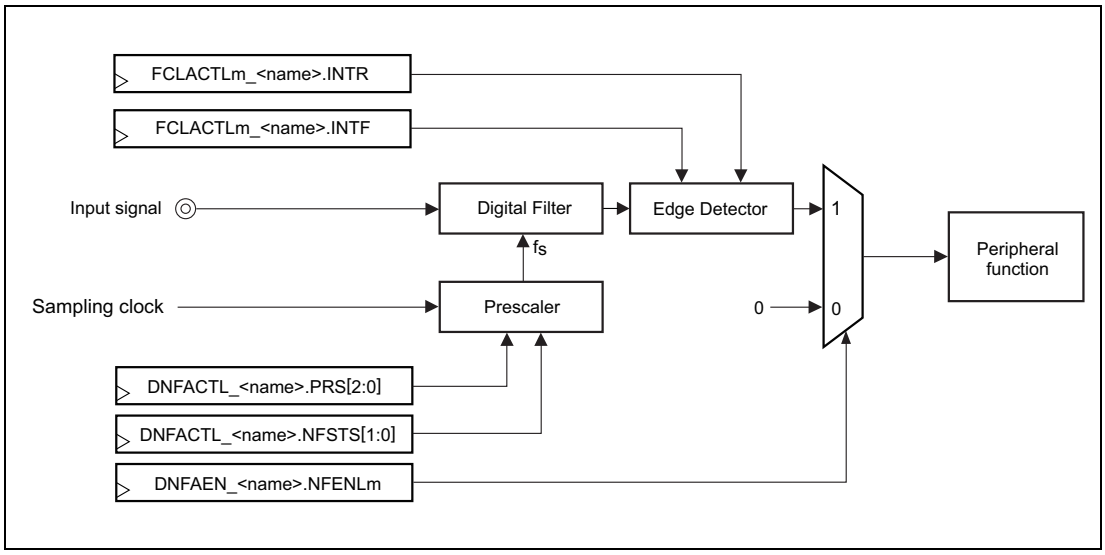


Figure 2.17 Block diagram of DNF type C1

2.7.2.6 DNF Type C2

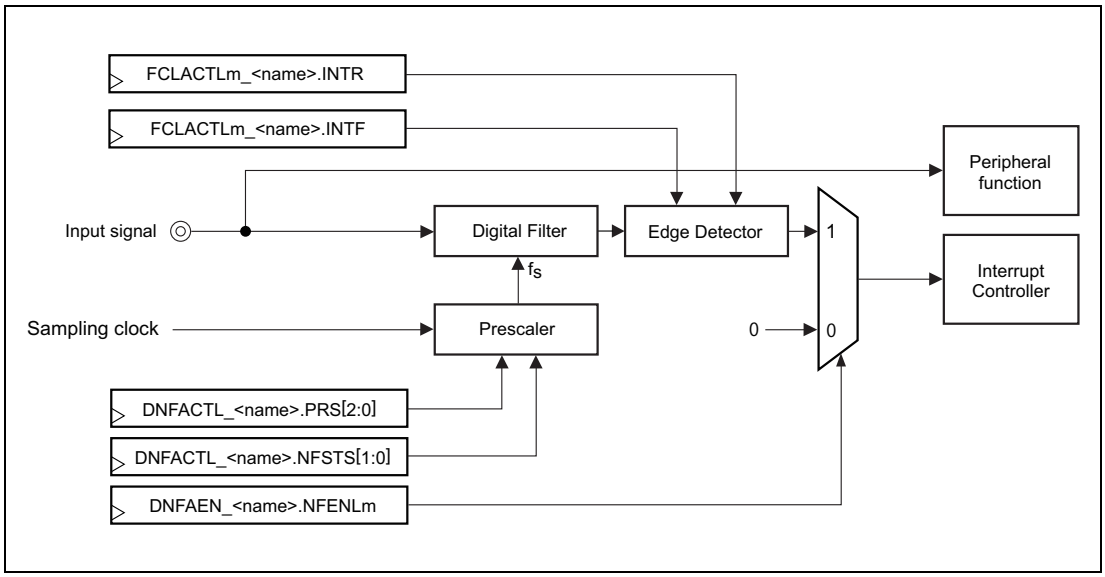


Figure 2.18 Block diagram of DNF type C2

2.7.2.7 DNF Type D1

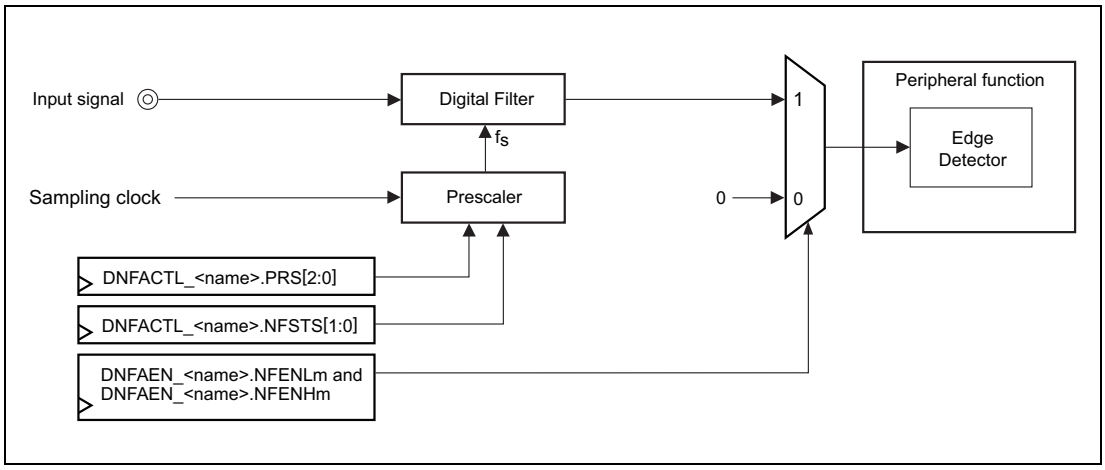


Figure 2.19 Block diagram of DNF type D1

2.7.2.8 DNF Type D2

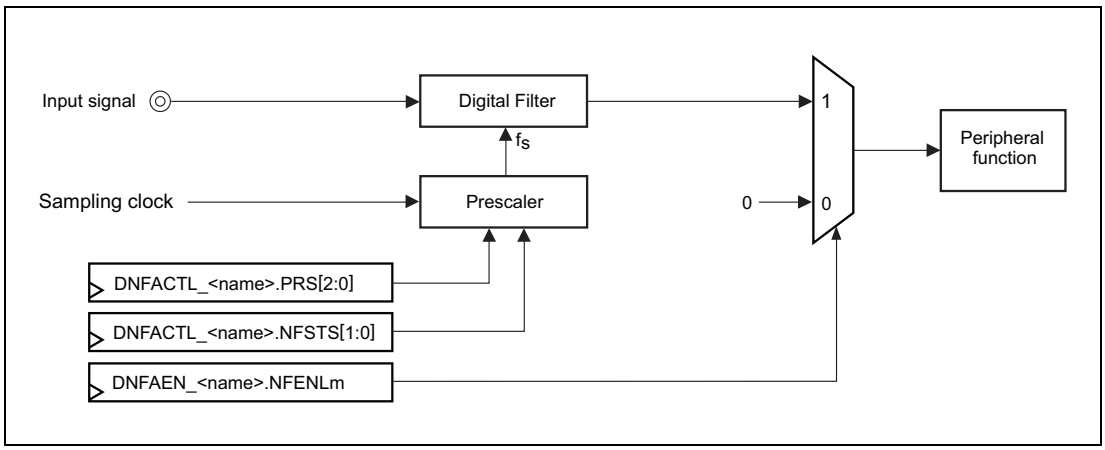


Figure 2.20 Block diagram of DNF type D2

2.7.2.9 ANF/DNF Type E1

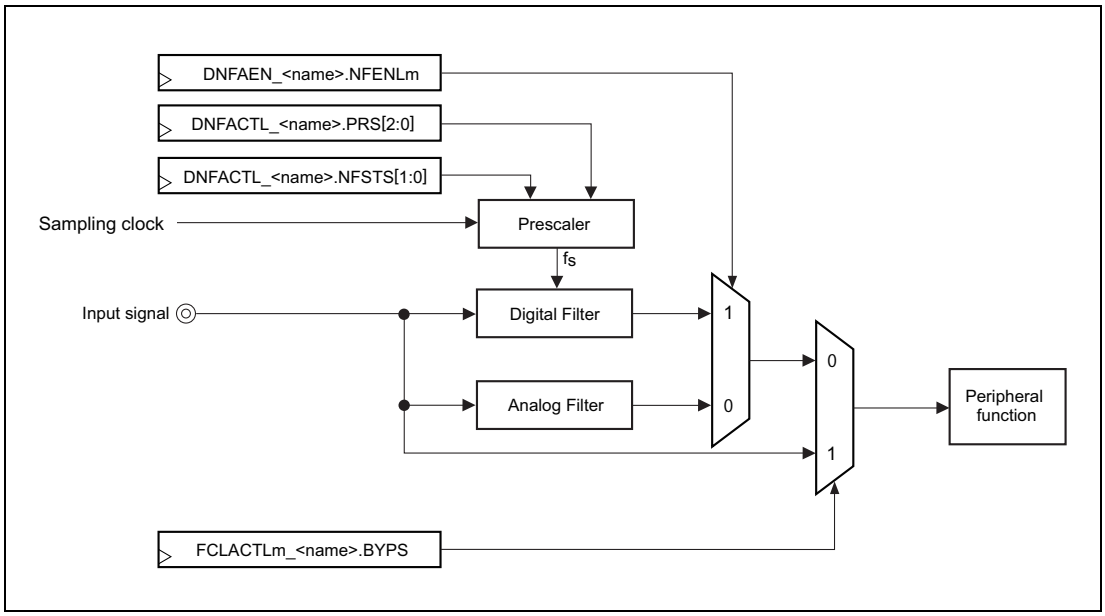


Figure 2.21 Block diagram of ANF/DNF type E1

2.7.2.10 ANF/DNF Type F1

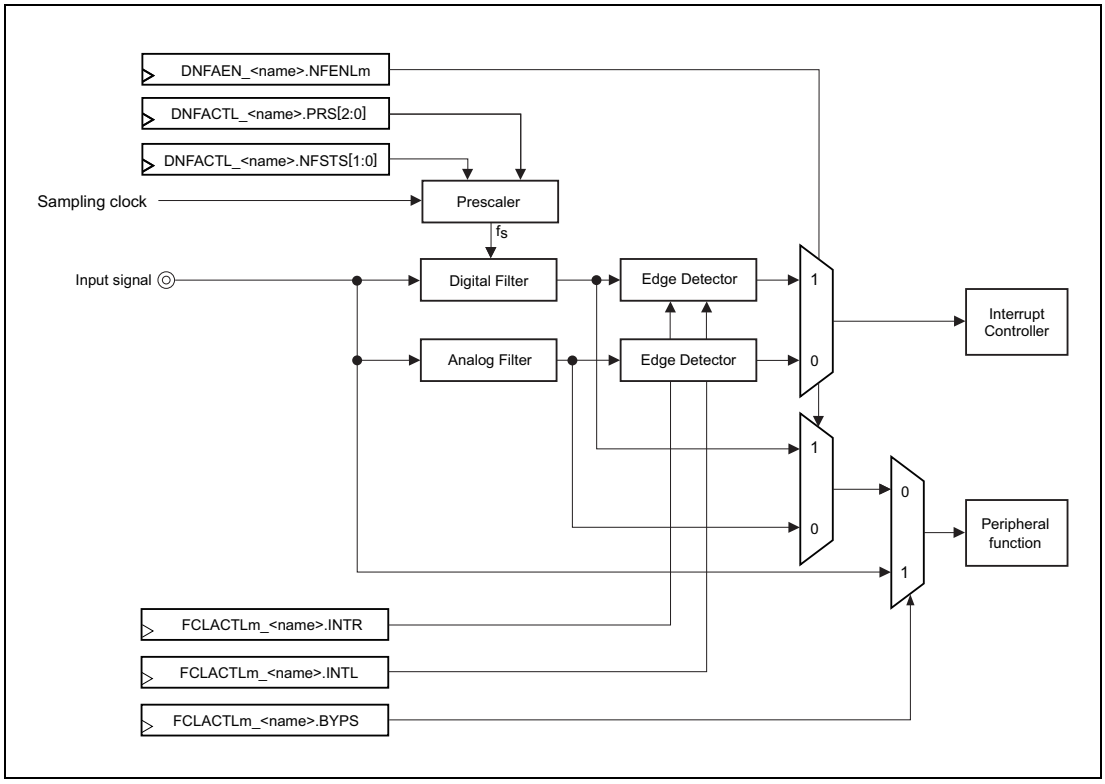


Figure 2.22 Block diagram of ANF/DNF type F1

2.7.2.11 ANF/DNF Type G1

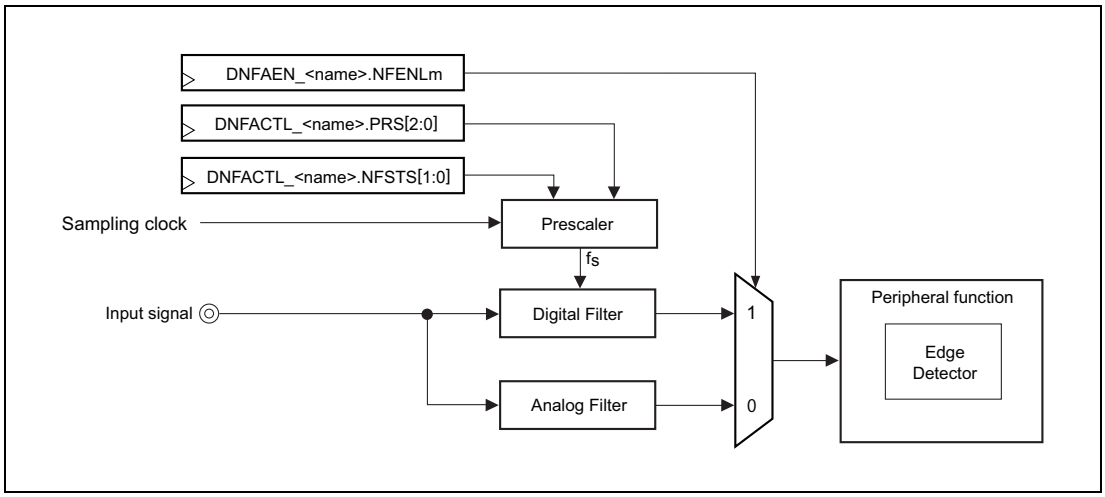


Figure 2.23 Block diagram of DNF type G1

2.7.3 Registers

2.7.3.1 List of Registers

The digital noise filter is controlled and operated by the following registers.

Table 2.50 List of Digital Noise Filter Registers

Register Name	Symbol	Address	Access Protection	
			PBG	Other
Filter control register	FCLACTLm_<name>	See **Section 2.7.1, Noise Filter Assignment**.	PBG20#13	—
Digital noise elimination control register	DNFACTL_<name>		*1	—
Digital noise elimination enable register	DNFAEN_<name>		*1	—
Digital noise elimination enable register H	DNFAENH_<name>		*1	—
Digital noise elimination enable register L	DNFAENL_<name>		*1	—

Note 1. PBG20#13 other than DNFACTL_TAUD2, DNFAEN_TAUD2, DNFAENH_TAUD2 and DNFAENL_TAUD2.
PBG51#3 for DNFACTL_TAUD2, DNFAEN_TAUD2, DNFAENH_TAUD2 and DNFAENL_TAUD2.

NOTE

<name> is a representative. For the actual name and existence of these registers, see **Section 2.7.1, Noise Filter Assignment**.

2.7.3.2 FCLACTLm_<name> — Filter Control Register

This register controls the analog and digital filter operation.

Access: This register can be read or written in 8-bit units.

Address: See **Table 2.49, Noise filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	BYPS	—	—	—	—	INTL	INTF	INTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Table 2.51 FCLACTLm_<name> Register Contents

Bit Position	Bit Name	Function
7	BYPS	Bypass mode selection 0: Select filter signal 1: Select filter-bypass signal **Note:** This bit is only valid for ANF/DNF Type E1 and F1. For other than these types, this bit should be the value after reset.
6 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	INTL	Detection mode selection: 0: Edge detection 1: Level detection **Note:** This bit is only for ANF Type A1. For other than these types, this bit should be the value after reset.
1	INTF	See **Table 2.52, Detection Mode Configuration**.
0	INTR	See **Table 2.52, Detection Mode Configuration**.

Table 2.52 Detection Mode Configuration

INTL	INTF	INTR	Detection Mode	Function
0	0	0	Edge detection	Disable edge detection
	0	1		Rising edge detection
	1	0		Falling edge detection
	1	1		Both rising and falling edge detection
1	X	0	Level detection	Low level detection
		1		High level detection

2.7.3.3 DNFACTL_<name> — Digital Noise Elimination Control Register

This register specifies characteristics of the digital noise elimination filter.

Access: This register can be read or written in 8-bit units.

Address: See **Table 2.49, Noise filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	NFSTS[1:0]		—	—	PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2.53 DNFACTL_<name> Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 5	NFSTS[1:0]	Specify the number of samples used to judge whether an external signal pulse is valid. **NFSTS[1:0]** / **Number of Samples** 00_B / 2 01_B / 3 10_B / 4 11_B / 5
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PRS[2:0]	Digital filter sampling clock selection **PRS[2:0]** / **Sampling clock Frequency** 000_B / Sampling clock supply/1 001_B / Sampling clock supply/2 010_B / Sampling clock supply/4 011_B / Sampling clock supply/8 100_B / Sampling clock supply/16 101_B / Sampling clock supply/32 110_B / Sampling clock supply/64 111_B / Sampling clock supply/128 About "*Sampling clock supply*" of each DNF, refer to **Table 2.49, Noise filter Assignment**.

2.7.3.4 DNFAEN_<name> — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

Access: This register can be read or written in 16-bit units.
The upper- and lower-order bytes (NFENH[7:0] and NFENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFAENH_<name> and DNFAENL_<name>.

Address: See **Section 2.7.1, Noise Filter Assignment**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NFENH7	NFENH6	NFENH5	NFENH4	NFENH3	NFENH2	NFENH1	NFENH0	NFENL7	NFENL6	NFENL5	NFENL4	NFENL3	NFENL2	NFENL1	NFENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.54 DNFAEN_<name> Register Contents

Bit Position	Bit Name	Function
15 to 0	NFENH[7:0] NFENL[7:0]	Enable/Disable digital noise elimination: 0: Disabled 1: Enabled **Note:** For the correspondence between these bits and input signals. See **Section 2.7.1, Noise Filter Assignment**.

2.7.3.5 DNFAENH_<name> — Digital Noise Elimination Enable H Register

Setting in this register correspond to those of the 8 upper-order bits of DNFAEN_<name> register.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: See **Section 2.7.1, Noise Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NFENH7	NFENH6	NFENH5	NFENH4	NFENH3	NFENH2	NFENH1	NFENH0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2.7.3.4, DNFAEN_<name> — Digital Noise Elimination Enable Register**.

2.7.3.6 DNFAENL_<name> — Digital Noise Elimination Enable L Register

Setting in this register correspond to those of the 8 lower-order bits of DNFAEN_<name> register.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: See **Section 2.7.1, Noise Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NFENL7	NFENL6	NFENL5	NFENL4	NFENL3	NFENL2	NFENL1	NFENL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

For details of the respective bit functions, see **Section 2.7.3.4, DNFAEN_<name> — Digital Noise Elimination Enable Register**.

Section 3 CPU System

3.1 Overview

3.1.1 Block Diagram

For the block diagram of the CPU system, see **Section 1, Overview, Figure 1.9** to **Figure 1.14**.

The CPU system consists of the following modules.

CPU0 (PE0), CPU1 (PE1), CPU2 (PE2), CPU3 (PE3)

The RH850G4MH2 core is included as the main CPU. The CPUs also include a checker core for safety assurance.

NOTES

1. **CPU2 (PE2), CPU3 (PE3) are not implemented in RH850/U2A8 and RH850/U2A6.**
2. **In this Section 3, "G4MH core" is understood as "G4MH2 core".**

ICUMHA

Refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Local RAM

Each PE has a high-speed accessible local RAM.

Cluster RAM

The cluster RAM is a mass on-chip RAM that all PEs share. Some areas of this RAM function as the retention RAM.

Code Flash

A mass code flash is included for program storage. All PEs and ICUMHA share the code flash and they are connected with each other via the Local/Global FLASH Bus.

Emulation RAM

This is RAM to emulate the code flash. The programs can be replaced by control from an external tool without rewriting the code flash. It is supported in RH850/U2A-EVA and RH850/U2A6.

Data Flash

This is a flash memory rewritable by the CPU.

Instrumentation RAM

Instrumentation RAM is data holding RAM area incorporated as the RH850 2nd generation debugging function for tools to output. It is supported only in RH850/U2A-EVA.

P-Bus and H-Bus and I-Bus

Peripheral IPs are connected on P-Bus, H-Bus and I-Bus.

U2A-EVA, U2A16, U2A8: The P-Bus consists of 10 groups, P-Bus Group 0 to 9.
The H-Bus consists of 4 groups, H-Bus Group 0 to 3.
The I-Bus consists of 3 groups, I-Bus Group 0 to 2.

U2A6: The P-Bus consists of 10 groups, P-Bus Group 0 to 9.
The H-Bus consists of 2 groups, H-Bus Group 1 and 3.
The I-Bus consists of 3 groups, I-Bus Group 0 to 2.

Refer to each section on peripheral IPs for information on bus groups.

INTC1, INTC2

INTC1 is an interrupt controller exclusive to each PE. INTC2 is a common interrupt controller that all PEs can share, which is able to set the destination PE for binding interrupt requests by the registers.

DMA

Two DMA transfer modules, sDMAC and DTS, are included. DTS includes the checker logic for safety assurance.

3.2 CPU

3.2.1 Core Functions

3.2.1.1 Features

Table 3.1 lists the features of the RH850G4MH core. See the *RH850G4MH User's Manual: Software.*

Table 3.1 Features of the RH850G4MH Core

Item	Feature
CPU	• High performance 32-bit architecture for embedded control • 32-bit internal data bus • Thirty-two 32-bit general-purpose registers • RISC-type instruction sets Long/short type load/store instructions Three-operand instructions Instruction sets based on C • Supports Hardware-assisted Virtualization – CPU operation mode, register access privileges, and instruction execution privileges for virtual machines – Independent exception processing/interrupt acceptance processing for virtual machines – Nested memory protection function for virtual machines – Context high-speed save/return function for virtual machines • CPU operating modes – User mode, supervisor mode • Address space: 4-GB linear space for both data and instructions • Out-of-Order execution
Coprocessor	• A floating point operation coprocessor (FPU) Supports single precision (32-bits) and double precision (64-bits). Supports IEEE754-compliant data types and exceptions. Rounding mode: Nearest, 0 direction, +∞ direction, and -∞ direction Handling of non-normalized numbers: These are truncated to 0, or an exception is reported to comply with IEEE754.
Exception/Interrupt	• 64-level interrupt priority that can be specified for each channel • Vector selection method that can be selected according to performance requirements and the amount of consumed memory Direct branch method exception vector (direct vector method) Address-table-referencing indirect branch method exception vector (table reference method) • Support for high-speed context save and restore processing on interrupt by using dedicated instructions (PUSHSP, POPSP) • Support for high-speed context save on interrupt by using the register bank feature • Support for restoration from the register bank using a dedicated instruction (RESBANK) • The following functions are added for Hardware assisted Virtualization – Exception handling for Host mode and Guest mode can be used independently. – Exception vector for Host mode and Guest mode can be used independently. – Interrupt assigned to Host Mode is given priority over any other interrupts assigned to Guest Mode. – Interrupt channel can be assigned to Guest mode. – High priority interrupt for background Guest can be accepted promptly.
Memory Management	• Memory protection function (MPU): 32 areas settable.
Cache	• Instruction cache 16 KB

Notes: • For details of Virtualization and Hypervisor, see the *RH850G4MH Virtualization User's Manual: Hardware.*
• FXU (extended floating-point operation unit) is not supported in RH850/U2A.

3.2.2 Processor Model

This CPU defines a processor model that has basic operation functions, registers, and an exception management function.

This section describes the unique features of the processor model of this CPU.

3.2.2.1 CPU Operating Modes

This CPU has two operating modes of the supervisor mode (SV) and the user mode (UM). Whether the system is in supervisor mode or user mode is indicated by the UM bit in the PSW register.

- Supervisor mode (PSW.UM = 0): All hardware functions can be managed or used.
- User mode (PSW.UM = 1): The usable hardware functions are restricted.

(1) Definition of CPU Operating Modes

(a) Supervisor Mode (SV)

All hardware functions can be managed or used in this mode. The system always starts up in supervisor mode after a reset.

(b) User Mode (UM)

This operating mode makes up a pair with the supervisor mode. In user mode, address spaces to which access is permitted by the supervisor and the system registers defined as user resources can be used. Supervisor-privileged instructions cannot be executed and result in exceptions.

Restriction in user mode (PSW.UM = 1)

- Privileged instruction violations due to SV-privileged-instruction operating restrictions (→ PIE exceptions)

For details about privileged-instruction operating restrictions, see **Section 3.2.2.1(3), CPU Operating Mode and Privileges**.

(2) CPU Operating Mode Transition

The CPU operating mode changes due to three events.

(a) Change due to Acknowledging an Exception

When an exception is acknowledged in user mode, the CPU operating mode changes to privileged mode.
When an exception is acknowledged in supervisor mode, the CPU operating mode remains as is.

(b) Change due to a Return Instruction

When a return instruction is executed, the PSW value is restored according to the value of the corresponding bit backed up to EIPSW and FEPSW.

(c) Change due to a System Register Instruction

The CPU operating mode changes when an LDSR instruction is used to directly overwrite the PSW operating mode bits.

CAUTION

The CPU operating mode cannot be changed in user mode because the higher-order 31 to 5 bits of the PSW register cannot be overwritten while the mode can be changed in supervisor mode. This CPU guarantees that if an LDSR instruction is used to update the PSW register, the new setting will be reflected when the subsequent instruction is executed. However, this CPU does not guarantee that the new setting will be reflected in the memory protection by the MPU for instruction fetch of the subsequent instruction. Therefore, for changing the higher-order 31 to 5 bits of the PSW register, it is recommended to use a return instruction. For details, see Section 3.2.7.3, Hazard Management after System Register Update.

(3) CPU Operating Mode and Privileges

In this CPU, the usable functions can be restricted according to usage permission settings for specific resources and the CPU operating mode. Specific instructions (including instructions that update specific system registers) can only be executed in the defined operating mode. The permissions necessary to execute these specific instructions are called “privileges”. In operating modes that do not have privileges, these instructions are not executed and exceptions occur.

This CPU defines the following two types of privileges (usage permission).

- Supervisor (SV) privilege: Privilege necessary for important system resources operation, fatal error processing, and user-mode program execution management
- Coprocessor use permissions: Permissions necessary to use a coprocessor

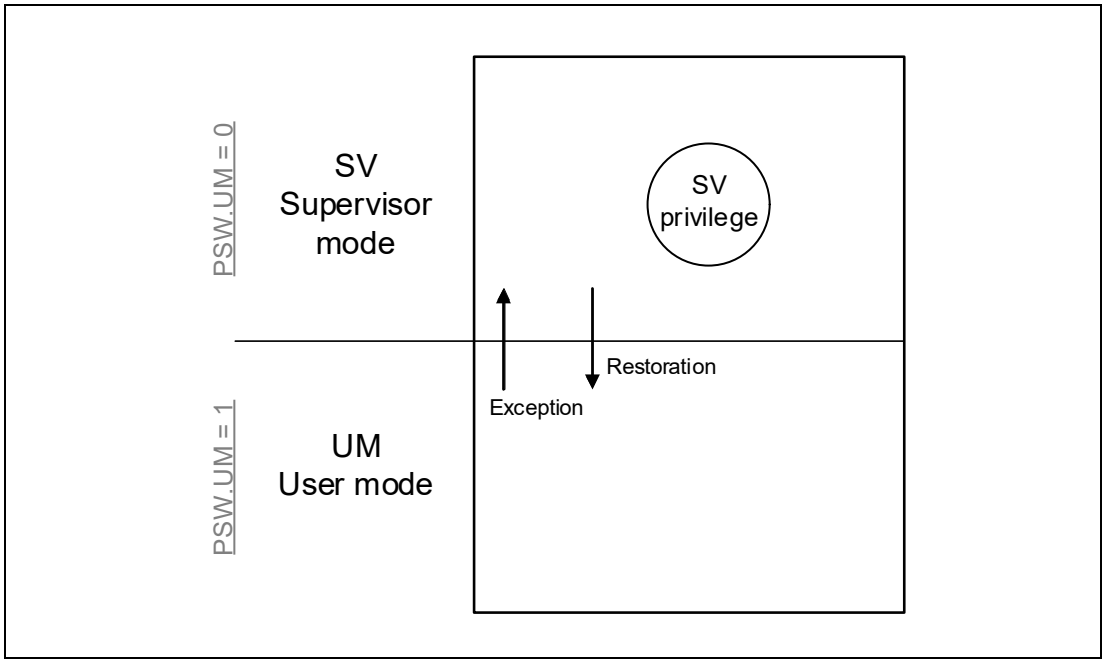


Figure 3.1 CPU Operating Modes and Privileges

(a) Supervisor Privilege (SV Privilege)

The privilege necessary to perform the operation for important system resources, fatal error processing, and user-mode program execution management is called the supervisor privilege (SV privilege). This privilege is available in supervisor mode. The SV privilege is generally necessary to execute instructions used to perform the operation for important system resources, and these instructions are sometimes called SV privileged instructions.

(b) Coprocessor Use Permissions

Regardless of the CPU operating mode, it is possible to specify whether coprocessors can be used.

The CU2 to CU0 bits in the PSW register are used by a supervisor to specify whether coprocessors can be used by each program. If the CU 2 to CU0 bits are not set to 1, a coprocessor unusable exception occurs when the corresponding coprocessor instruction is executed or the system register is accessed.

If no coprocessor is installed, it is not possible to set the corresponding CU bits to 1. The setting of the CU2 to CU0 bits is valid regardless of the CPU operating mode, and, if the supervisor accesses coprocessor system registers, it is necessary to set the CU2 to CU0 bits to enable coprocessor use.

(c) Operation when there is a Privilege Violation

When a privileged instruction is executed by someone who does not have the required privilege, or when a system register for which access permission is specified is accessed by someone who does not have the required permission, a PIE exception or UCPOP exception occurs.

The relationship between instructions that can be executed according to operating mode and use permission are shown in **Table 3.2**, and the relationship between system registers that can be accessed according to operating mode and use permission are shown in **Table 3.3**.

Table 3.2 Operation when Violating Execution Permission

Instruction		PSW				
Execution Permission	Classification	UM	CU2	CU1	CU0	Whether Instructions can be Executed
SV privilege	*1	0	—	—	—	Possible
		1	—	—	—	Not possible (PIE exception occurs)
User	Coprocessor 0 instruction	—	—	—	0	Not possible (UCPOP exception occurs)
		—	—	—	1	Possible
	Coprocessor 1 instruction	—	—	0	—	Not possible (UCPOP exception occurs)
		—	—	1	—	Possible
	Coprocessor 2 instruction	—	0	—	—	Not possible (UCPOP exception occurs)
		—	1	—	—	Possible
	Other than above	—	—	—	—	Possible

Note: —: 0 or 1

Note 1. Coprocessor instructions with SV privilege are not defined in this CPU.

Table 3.3 Operation When Violating Access Permission to System Registers

System Register		PSW				
Access Permission	Classification	UM	CU2	CU1	CU0	Whether Instructions can be Executed
SV permission	Coprocessor 0 Permission	0	—	—	0	Inaccessible (UCPOP exception occurs)
		0	—	—	1	Accessible
		1	—	—	0	Inaccessible (UCPOP exception occurs)
		1	—	—	1	Inaccessible (PIE exception occurs)
	Coprocessor 1 Permission	0	—	0	—	Inaccessible (UCPOP exception occurs)
		0	—	1	—	Accessible
		1	—	0	—	Inaccessible (UCPOP exception occurs)
		1	—	1	—	Inaccessible (PIE exception occurs)
	Coprocessor 2 Permission	0	0	—	—	Inaccessible (UCPOP exception occurs)
		0	1	—	—	Accessible
		1	0	—	—	Inaccessible (UCPOP exception occurs)
		1	1	—	—	Inaccessible (PIE exception occurs)
	Other than above	0	—	—	—	Accessible
		1	—	—	—	Inaccessible (PIE exception occurs)
UM permission	Coprocessor 0 Permission	—	—	—	0	Inaccessible (UCPOP exception occurs)
		—	—	—	1	Accessible
	Coprocessor 1 Permission	—	—	0	—	Inaccessible (UCPOP exception occurs)
		—	—	1	—	Accessible
	Coprocessor 2 Permission	—	0	—	—	Inaccessible (UCPOP exception occurs)
		—	1	—	—	Accessible
	Other than above	—	—	—	—	Accessible

Note: —: 0 or 1

(4) Halt State by a HALT Instruction

Executing a HALT instruction brings the CPU core to a HALT state.

A HALT state is terminated when a request for a terminating exception is made to the CPU. At this point, if the conditions for acknowledging the terminating exception are met, control is transferred to the exception handler address. Even if they are not met, the HALT state is terminated when the request is made; therefore, operation is resumed from the instruction following the HALT instruction.

(5) Temporary Halt State by a SNOOZE Instruction

A SNOOZE instruction is an instruction to reduce the use of bus bandwidth during a spinlock. After this instruction is completed, the CPU core is brought to a temporary halt state to restrict the execution of the subsequent instructions. Programmers can avoid the unnecessary use of bus bandwidth that results from short-term repetition of a locking process, by inserting this instruction into a spinlock loop.

A temporary halt state is terminated if any of the following conditions is met:

- After the time specified by the SNZCFG register elapsed, the execution of the subsequent instructions is resumed.
- The occurrence of a terminating exception causes control to be transferred to the exception handler address. If the conditions for acknowledging the exception are not met, execution is resumed from the instruction following the SNOOZE instruction. Also, if the exception is acknowledged, the instruction following the SNOOZE instruction is interrupted, and the PC restored indicates the instruction following the SNOOZE instruction.

(6) Timing Adjustment by Instruction

If there is a necessary to adjust timing for program execution, SYNCP can be used as a specific instruction. When execute SYNCP in a row, the number of cycle which costs for 1 instruction depends on the CPU core. In case of G4MH core, this cycle number is 3. To adjust timing, the other instructions are not guaranteed.

3.2.2.2 Instruction Execution

The instruction execution flow of this CPU is shown below.

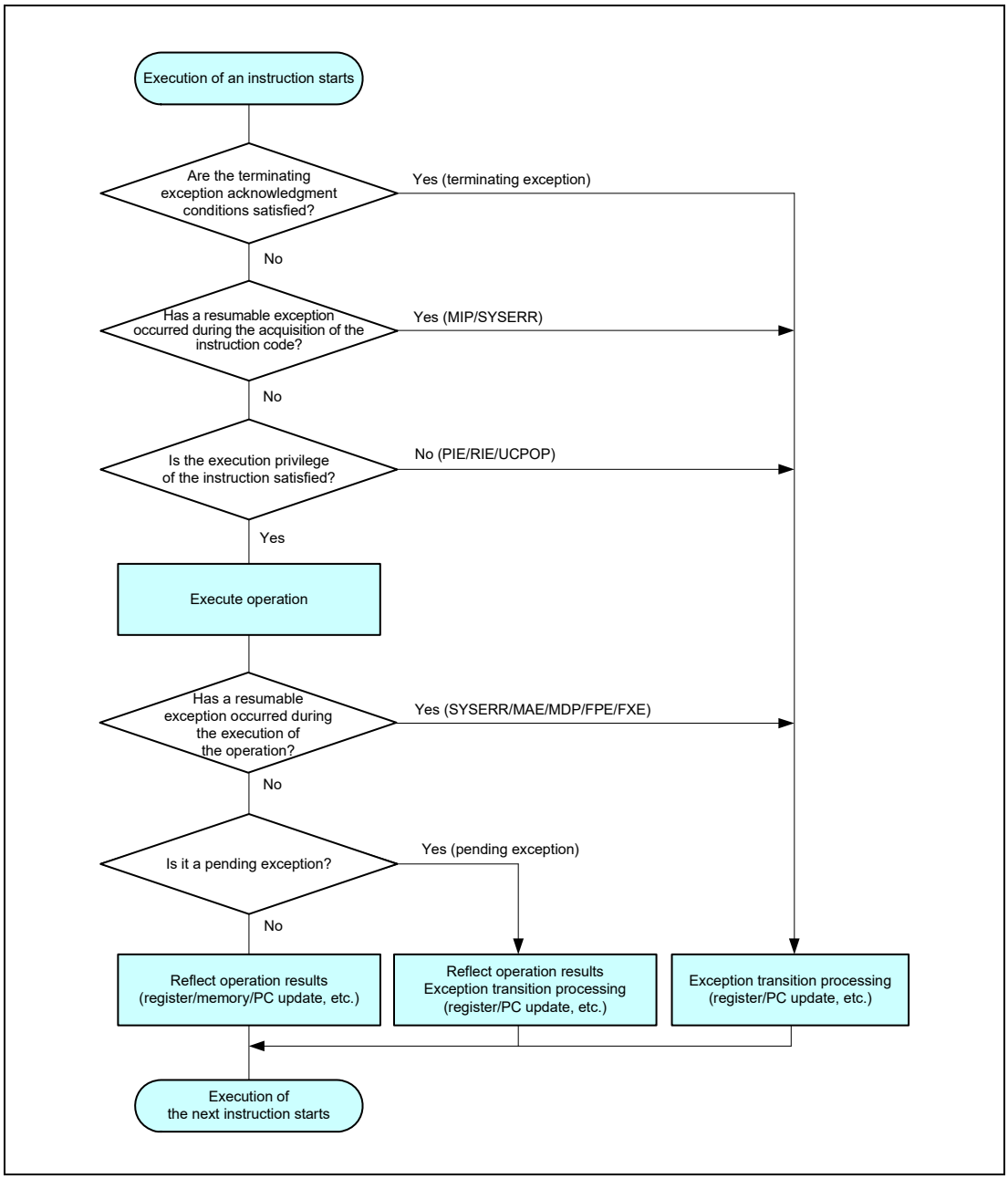


Figure 3.2 Instruction Execution Flow

If terminating exceptions can be acknowledged, if an exception is detected during the acquisition of instruction code, or if the execution privilege of the instruction is not satisfied, an exception occurs before the instruction is executed. If a resumable exception occurs while the CPU is executing the operation, it interrupts the execution of the operation and acknowledges the exception. In these cases, the result of instruction operation is, in principle, not reflected in registers or memory and the CPU retains its state that is established before executing the instruction.*1

For a pending exception such as a software exception, the exception is acknowledged after the result of instruction execution has been reflected.

For details of the types of exceptions, refer to **Section 3.2.2.3, Exceptions and Interrupts**.

Note 1. If an exception is acknowledged during the execution of the following instructions, intermediate results may be applied to memory or general-purpose registers. However, SP/EP is not updated.

- PREPARE, DISPOSE, PUSHSP, POPSP, STM.MP, LDM.MP

3.2.2.3 Exceptions and Interrupts

Exceptions and interrupts are exceptional events that cause the program under execution to branch to another program. Exceptions and interrupts are triggered by various sources, including interrupts from peripherals and program abnormalities.
For details, see **Section 3.2.2.3, Exceptions and Interrupts**.

(1) Types of Exceptions

The exceptions of this CPU are divided into the following three types according to the purpose of the exceptions.

- Terminating exception
- Resumable exception
- Pending exception

(a) Terminating Exception

A terminating exception is an exception that is acknowledged by interrupting an instruction before the execution of the operation for the instruction. Terminating exceptions include interrupts, etc.

This type of exceptions, such as an interrupt or hardware error, start another program irrelevant to the currently processed program.

(b) Resumable Exception

A resumable exception is an exception that occurs during the execution of the operation for an instruction and that is acknowledged without the completion of the instruction. It is also referred to as a precise exception because it is precisely acknowledged without the completion of the relevant instruction or the execution of the subsequent instruction.

Unlike a terminating exception, a resumable exception occurs during the execution of an instruction, and cancel the execution of the instruction, and so it allows the instruction to be executed again after exception handling. Therefore, it is possible to perform complex memory management while maintaining the consistency of the logical operation of programs, by executing the same instruction again after appropriate configuration by exception handling.

(c) Pending Exception

A pending exception is an exception that occurs as a result of the execution of the operation for an instruction, and that is acknowledged after the completion of the instruction. Pending exceptions include software exceptions, etc.

The occurrence of a pending exception is defined as a normal operation of an instruction; therefore, unlike a resumable exception, a pending exception normally completes the instruction that caused it, and never re-executes the instruction. This exception is intended mainly for use in call gates, for example, for calling a management program.

(2) Exception Level

In this CPU, if an exception with a high degree of urgency occurs while another exception is being processed, the urgent exception will be processed. To make it possible to return to the interrupted exception handling after acknowledging the urgent exception, even if the context had not been saved to the memory, exception causes are managed in the following two hierarchical levels.

- EI level exception
- FE level exception

EI level exceptions are used for processing such as regular user processing, interrupt servicing, and OS processing. FE level exceptions are used to enable interrupts with a high degree of urgency for the system or exceptions from the memory management function that might occur during OS processing to be acknowledged even while an EI level exception is being processed.

3.2.2.4 Coprocessors

In this CPU, single-precision and double-precision floating-point unit (FPU) and extended floating-point operation unit (FXU) are incorporated. Note that these coprocessors may not be available depending on the specification of the product.

(1) Coprocessor Use Permissions

To execute a coprocessor instruction, permission to use the corresponding coprocessor instruction is necessary. Coprocessor use permissions are specified by the PSW.CU2 to PSW.CU0 bits, and, if an attempt is made to execute an instruction for which the corresponding coprocessor use permission is cleared to 0, a coprocessor unusable exception (UCPOP) occurs.

In the following cases, the values of the PSW.CU2 to CU0 bits are fixed at 0 and cannot be changed.

- Coprocessor functions are not incorporated in the product
- Coprocessor functions are made unavailable according to the functions of the product

(2) Correspondences between Coprocessor Use Permissions and Coprocessors

This CPU defines coprocessor use permissions to control the availability of the coprocessor for each program during CPU operation. There are three coprocessor use permissions (CU0 to CU2), and their correspondences with the coprocessors are shown in the following table. This CPU does not have a coprocessor function with the coprocessor use permission CU2. The coprocessor use permission CU2 is a function reserved for future CPUs that have compatibility with this CPU.

Table 3.4 Correspondences between Coprocessor Use Permissions and Coprocessors

Coprocessor Use Permission	Coprocessor	Exception Cause Code
CU0	Single-precision FPU	80_H
	Double-precision FPU	
CU1	FXU	81_H
CU2	Reserved	82_H

(3) Coprocessor Unusable Exceptions

A coprocessor unusable exception occurs if an attempt is made to execute a coprocessor instruction or access a system register of the coprocessor without having the corresponding coprocessor use permission (PSW.CUn = 0).

For details about the opcodes of the coprocessor instructions, see the *RH850G4MH User's Manual: Software*. For details about the system registers of the coprocessors, see **Section 3.2.3.4, FPU Function Registers** and **Section 3.2.3.5, FXU Function Registers**.

If the register bank function is used, the FPSR register, which is a system register of the FPU, is accessed when the automatic context saving and the RESBANK instruction is executed. In this case, even if the coprocessor use permission is not given (PSW.CU0 = 0), a coprocessor unusable exception does not occur.

(4) System Registers

Some coprocessor functions have system registers as the part of their functions. The coprocessor use permission is necessary to access the system register of a coprocessor function. For some system registers, the supervisor privilege (SV privilege) is necessary in addition to the coprocessor use permission.

For details about the permissions necessary to access system registers, see **Section 3.2.2.5, Registers**.

3.2.2.5 Registers

This CPU defines program registers (general-purpose registers and the program counter PC) and system registers for controlling the status and storing exception information.

(1) Program Registers

The program registers include general-purpose registers (r0 to r31) and the program counter (PC).

Table 3.5 Program Registers

Category	Access Permission	Name
Program counter	UM	PC
General-purpose registers	UM	r0-r31

Note: Access to the registers with UM (user mode) access permission is always allowed.

For details about program registers, see **Section 3.2.3.1, Program Registers**.

(2) System Registers

System registers are placed in dedicated address spaces defined based on two types of address information: selection ID and register number. Up to 32 selection ID can be defined, and one selection ID includes up to 32 system registers. Therefore, up to 1024 system registers can be defined in the address spaces for system registers. Basically this CPU allocates selection ID as shown below:

Selection ID 0 to 3:	Registers related to basic functions
Selection ID 4 and 5:	Registers related to the memory management function
Selection ID 10 and 11:	Registers related to functions expanded from the legacy architecture
Selection ID 12 and 13:	Registers related to this CPU specific hardware functions
Other ID:	Reserved for future expansion of CPUs compatible with this CPU

For details about system registers, see the relevant sections in **Section 3.2.3, Register Set**.

(3) Register Updating

There are several methods used to update registers. Normally, no particular restrictions apply when updating register by using an instruction. However, when updating registers by using the following instructions, some restrictions might apply, depending on the operating mode.

- LDSR
- STSR

(a) LDSR and STSR

The LDSR and STSR instructions can access all the system registers.

However, if a system register is accessed without the proper permission, a PIE exception or UCPOP exception might occur. For details about the access permission for each register, see the description of system registers in **Section 3.2.3, Register Set**. For details about behaviors when a privilege violation occurs, see **Section 3.2.2.1(3), CPU Operating Mode and Privileges**.

Figure 3.3 shows the flow of executing the LDSR and STSR instructions.

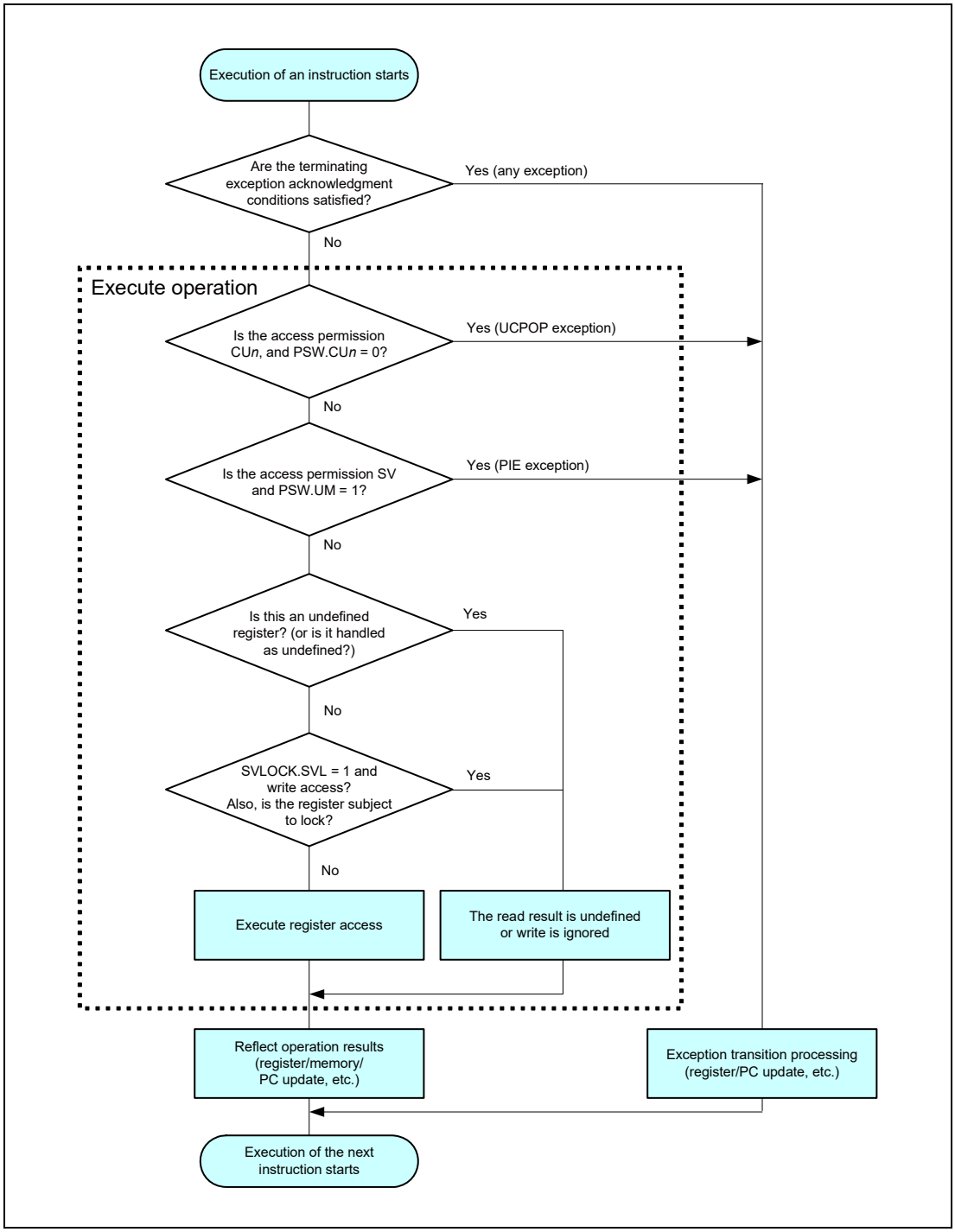


Figure 3.3 Flow of Executing the LDSR and STSR Instructions

(4) Accessing Undefined Registers

If a system register number without any register assigned is accessed or if an inaccessible register is accessed, the following results occur.

- Undefined registers are handled as having the SV permission. When they are accessed by an LDSR or STSR instruction in user mode (PSW.UM = 1), a PIE exception occurs.
- For a read operation, the read result is undefined. If the read value is used in a program, unexpected behaviors might occur.
- For a write operation, the write operation is ignored.

(5) Supervisor Lock Setting

If the SVLOCK.SVL bit is set (1), the following system registers cannot be updated even in supervisor mode:

SPID, MPM, MPLA, MPUA, MPAT, MPIDn, and MPBK

However, if the SVLOCK.SVL bit is set (1), an attempt to update the system registers above does not cause a PIE exception.

Also, even if the SVLOCK.SVL bit is set (1), it is possible to read the system registers above.

This function is intended for the safety of systems that incorporate this CPU. It prevents a program running in supervisor mode from abruptly changing memory protection setting due to a programming error, etc., to make illegal memory access possible, and to impair safety.

3.2.2.6 Data Types

(1) Data Formats

This CPU handles data in little endian format. This means that byte 0 of a halfword or a word is always the least significant (rightmost) byte.

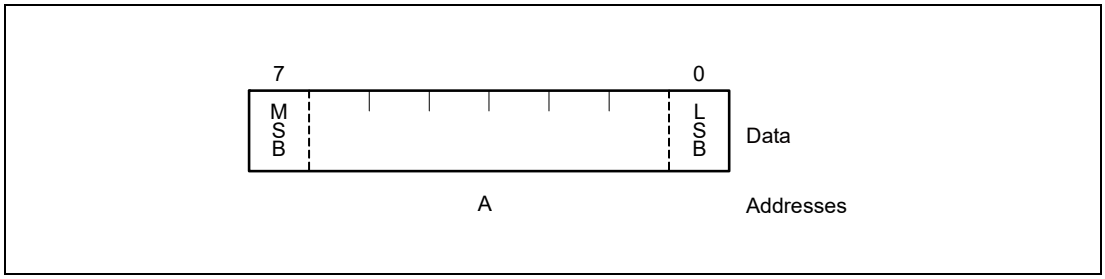
The supported data format is as follows.

- Byte (8-bit data)
- Halfword (16-bit data)
- Word (32-bit data)
- Double-word (64-bit data)
- Quad-word (128-bit data)
- Bit (1-bit data)

In this CPU, a memory access of double-word data is divided into two non-atomic memory accesses of word data, and a memory access of quad-word data is divided into four non-atomic memory accesses of word data.

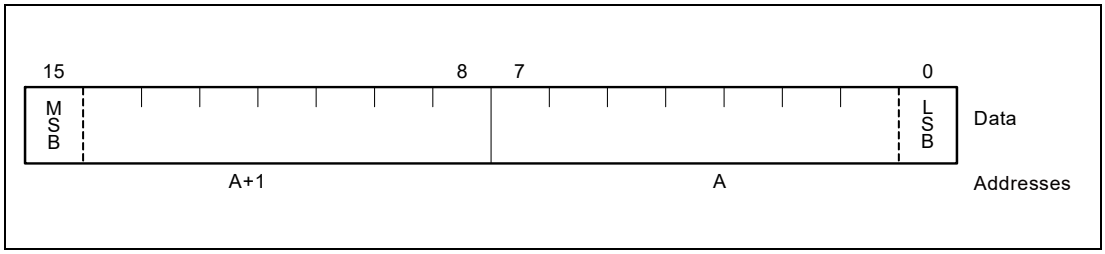
(a) Byte

A byte is 8 consecutive bits of data that starts from any byte boundary. Numbers from 0 to 7 are assigned to these bits, with bit 0 as the LSB (least significant bit) and bit 7 as the MSB (most significant bit). The byte address is specified as “A”.



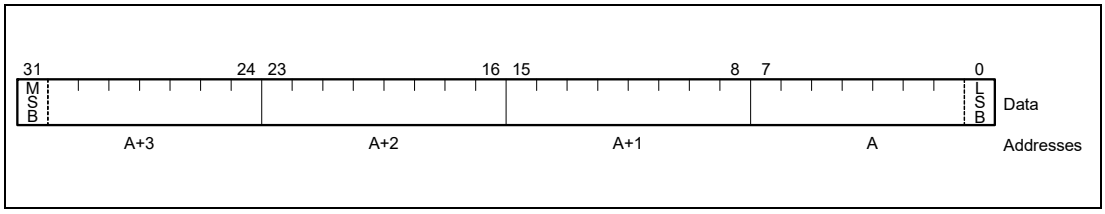
(b) Halfword

A halfword is two consecutive bytes (16 bits) of data that starts from any byte boundary. Numbers from 0 to 15 are assigned to these bits, with bit 0 as the LSB and bit 15 as the MSB. The bytes in a halfword are specified using address “A”, so that the two addresses comprise byte data of “A” and “A + 1”.



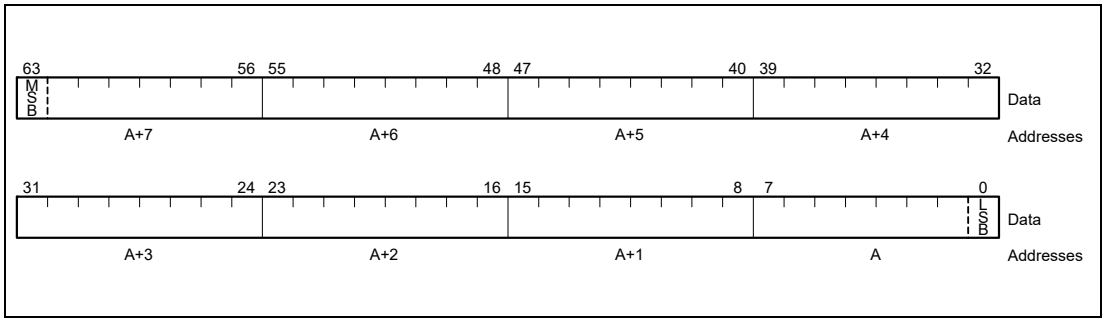
(c) Word

A word is four consecutive bytes (32 bits) of data that starts from any byte boundary. Numbers from 0 to 31 are assigned to these bits, with bit 0 as the LSB (least significant bit) and bit 31 as the MSB (most significant bit). A word is specified by address “A” and consists of byte data of four addresses: “A”, “A + 1”, “A + 2”, and “A + 3”.



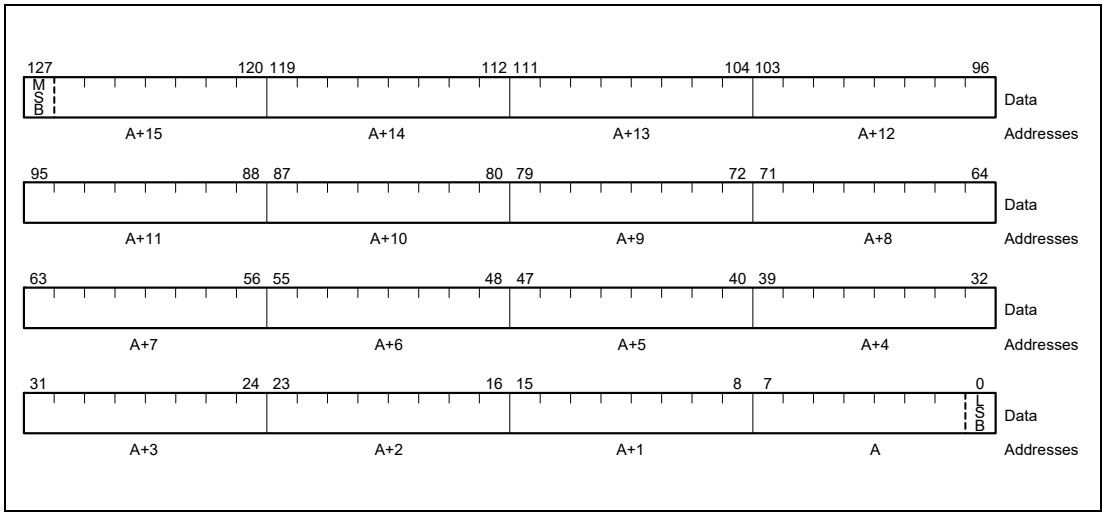
(d) Double-word

A double-word is eight consecutive bytes (64 bits) that start from any 4-byte boundary. Numbers from 0 to 63 are assigned to these bits, with bit 0 as the LSB and bit 63 as the MSB. A double-word is specified by address “A” and consists of byte data of eight addresses: “A”, “A + 1”, “A + 2”, “A + 3”, “A + 4”, “A + 5”, “A + 6”, and “A + 7”.



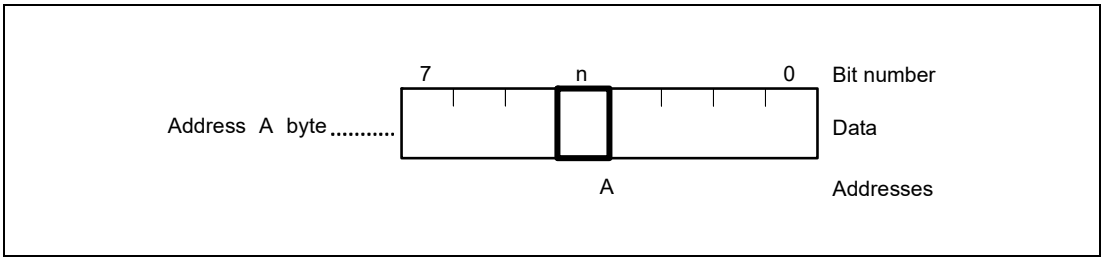
(e) Quad-word

A quad-word is sixteen consecutive bytes (128 bits) that start from any 8-byte boundary. Numbers from 0 to 127 are assigned to these bits, with bit 0 as the LSB and bit 127 as the MSB. A quad-word is specified by address “A” and consists of byte data of sixteen addresses: “A”, “A + 1”, …, and “A + 15”.



(f) Bit

A bit is bit data at the nth bit within 8-bit data that starts from any byte boundary. Each bit is specified using its byte address “A” and its bit number “n” (n = 0 to 7).



(2) Data Representation

(a) Integers

Integers are represented as binary values using 2's complement, and are used in one of four lengths: 64 bits, 32 bits, 16 bits, or 8 bits. Regardless of the length of an integer, its place uses bit 0 as the LSB, and this place gets higher as the bit number increases. Because this is a 2's complement representation, the MSB is used as a sign bit.

The integer ranges for various data lengths are as follows.

- Double-word (64 bits): −9223372036854775808 to +9223372036854775807
- Word (32 bits): −2147483648 to +2147483647
- Halfword (16 bits): −32768 to +32767
- Byte (8 bits): −128 to +127

Although data format of quad-word (128 bits) is defined, data representation of quad-word is not used in this CPU. This is because quad-word data consists of four-word data or two-double-word data, but this CPU does not process quad-word data directly.

(b) Unsigned Integers

In contrast to "integers" which are data that can take either a positive or negative sign, "unsigned integers" are never negative integers. Like integers, unsigned integers are represented as binary values, and are used in one of four lengths: 64 bits, 32 bits, 16 bits, or 8 bits. Also like integers, the positioning of unsigned integers uses bit 0 as the LSB and gets higher as the bit number increases. However, unsigned integers do not use a sign bit.

The unsigned integer ranges for various data lengths are as follows.

- Double-word (64 bits): 0 to 18446744073709551615
- Word (32 bits): 0 to 4294967295
- Halfword (16 bits): 0 to 65535
- Byte (8 bits): 0 to 255

(c) Bits

Bit data are handled as single-bit data with either of two values: cleared (0) or set (1). There are four types of bit-related operations (listed below), which target only single-byte data in the memory space.

- Set
- Clear
- Invert
- Test

(3) Data Alignment

In this CPU, misaligned data allocation is inhibited. When the result of address calculation is a misaligned address, a misalignment exception (MAE) occurs.

Misaligned access indicates the accesses to the data size with the addresses listed below:

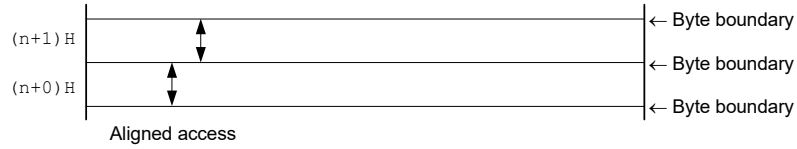
- Halfword size: The access to an address that is not at the halfword boundary (where LSB of the address = 0)
- Word size: The access to an address that is not at the word boundary (where the lowest two bits of the address = 0).
- Double-word size: The access to an address that is not at the double-word boundary (where the lowest three bits of the address = 0).
- Quad-word size: The access to an address that is not at the quad-word boundary (where the lowest four bits of the address = 0)

For the double-word format only, a misaligned access exception does not occur when data is placed at the word boundary but not at the double-word boundary, and data can be normally accessed in double word.

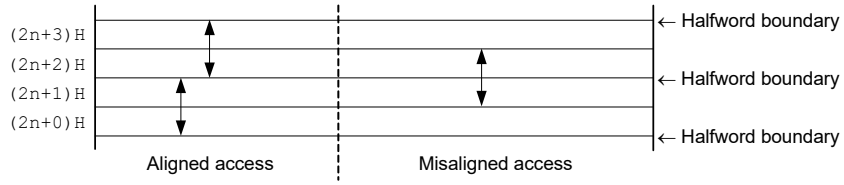
CAUTIONS

1. **The following instructions might possibly cause misaligned access. For details, see the relevant descriptions in the *RH850G4MH User's Manual: Software*.**
   - **LD.H, LD.HU, LD.W, LD.DW**
   - **SLD.H, SLD.HU, SLD.W**
   - **ST.H, ST.W, ST.DW**
   - **SST.H, SST.W**
   - **LDL.HU, LDL.W, STC.H, STC.W, CAXI**
   - **LDV.W, LDV.DW, LDV.QW, STV.W, STV.DW, STV.QW**
   - **LDVZ.H4, STVZ.H4**
2. **The following instructions do not cause misaligned access, because the address is rounded as the instruction specification when a misaligned address is specified.**
   - **PREPARE, DISPOSE**
   - **PUSHSP, POPSP**
   - **STM.MP, LDM.MP**

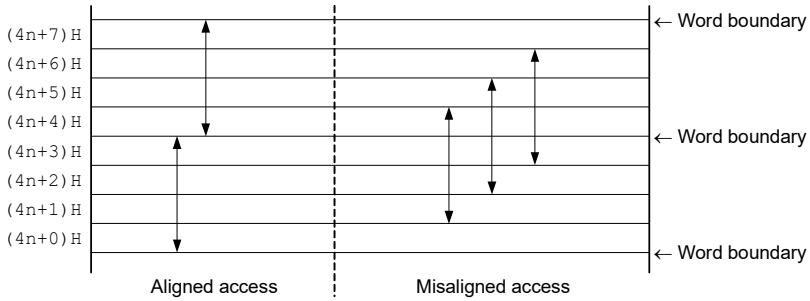
(a) Byte Access



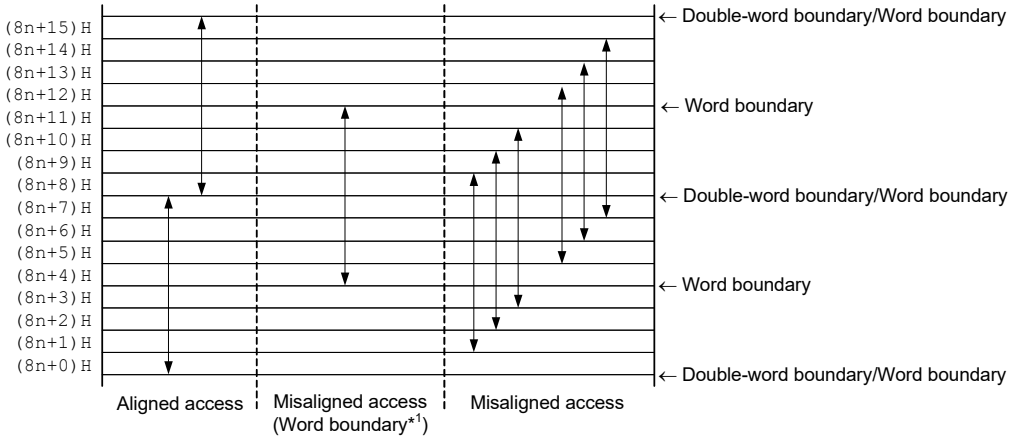
(b) Halfword Access



(c) Word Access



(d) Double-word Access



Note 1. No misalignment exception (MAE) occurs for LD.DW and ST.DW instructions. For details, see LD.DW and ST.DW in the *RH850G4MH User's Manual: Software*. An MAE occurs for other double-word access instructions.

Figure 3.4 Example of Data Placement for Misaligned Access (1/2)

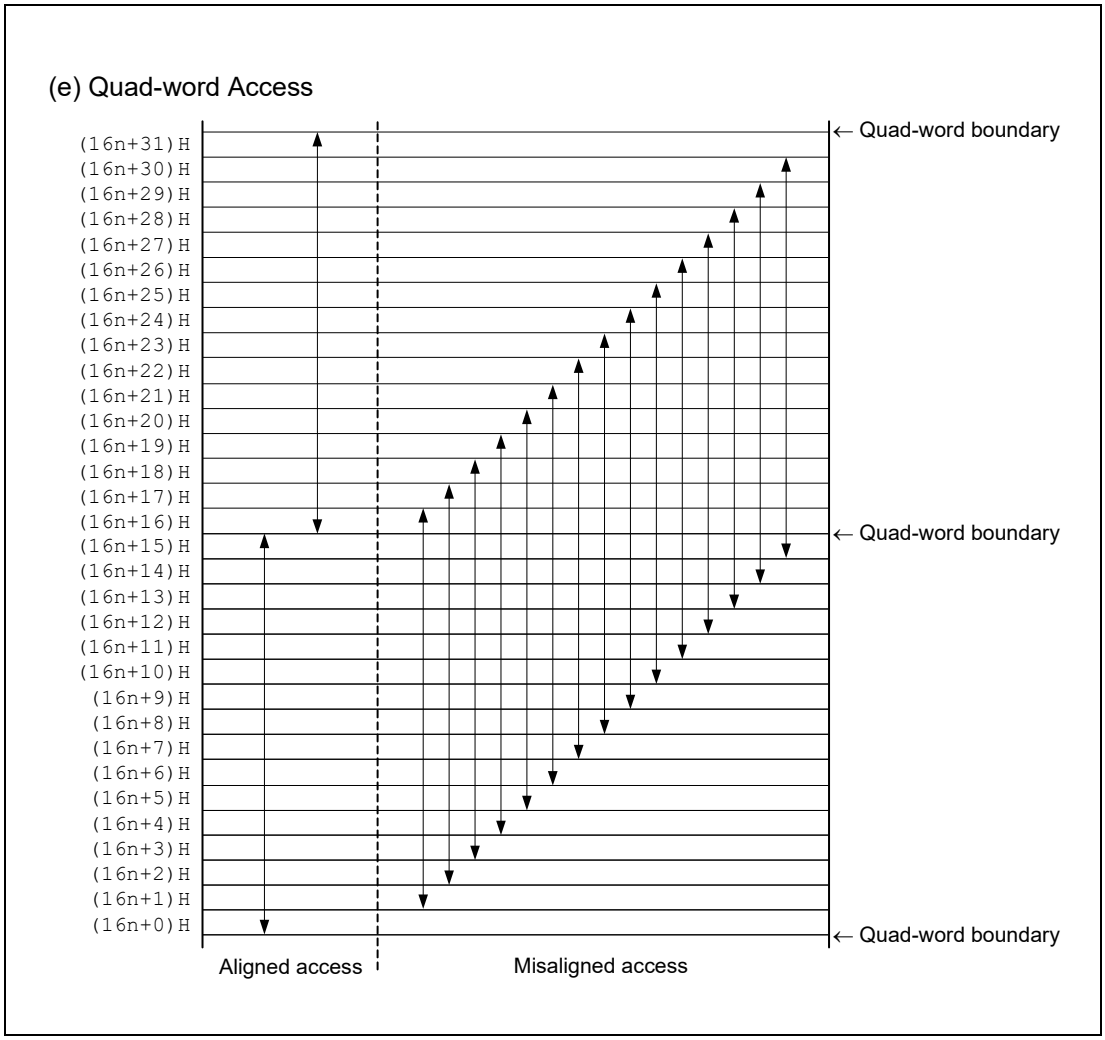


Figure 3.4 Example of Data Placement for Misaligned Access (2/2)

3.2.2.7 Address Space

This CPU supports a linear address space of up to 4 Gbytes. Both memory and I/O can be mapped to this address space (using the memory mapped I/O method). The CPU outputs a 32-bit address for memory and I/O, in which the highest address number is "$2^{32} - 1$".

The byte data placed at various addresses is defined with bit 0 as the LSB and bit 7 as the MSB. When the data is comprised of multiple bytes, it is defined so that the byte data at the lowest address is the LSB and the byte data at the highest address is the MSB (i.e., in little endian format).

This manual stipulates that, when representing data comprised of multiple bytes, the right edge must be represented as the lower-order addresses and the left side as the higher-order addresses, as shown below.

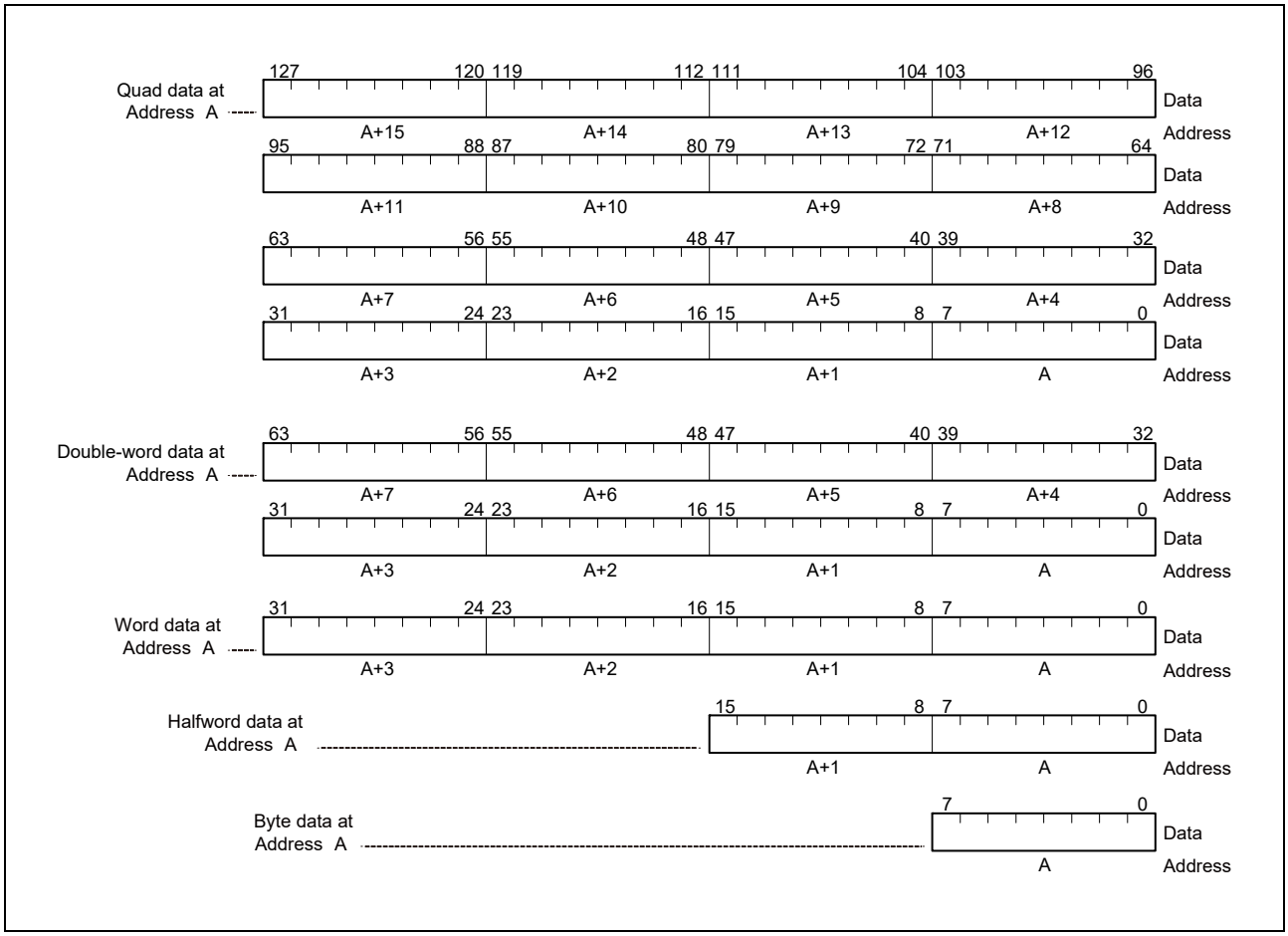


Figure 3.5 Address Space Byte Format

(1) Memory Map

This CPU is 32-bit architecture and supports a linear address space of up to 4 Gbytes. The whole range of this 4-Gbyte address space can be addressed by instruction addressing (instruction fetch access) and data addressing (data access).

A memory map is shown in **Figure 3.6**.

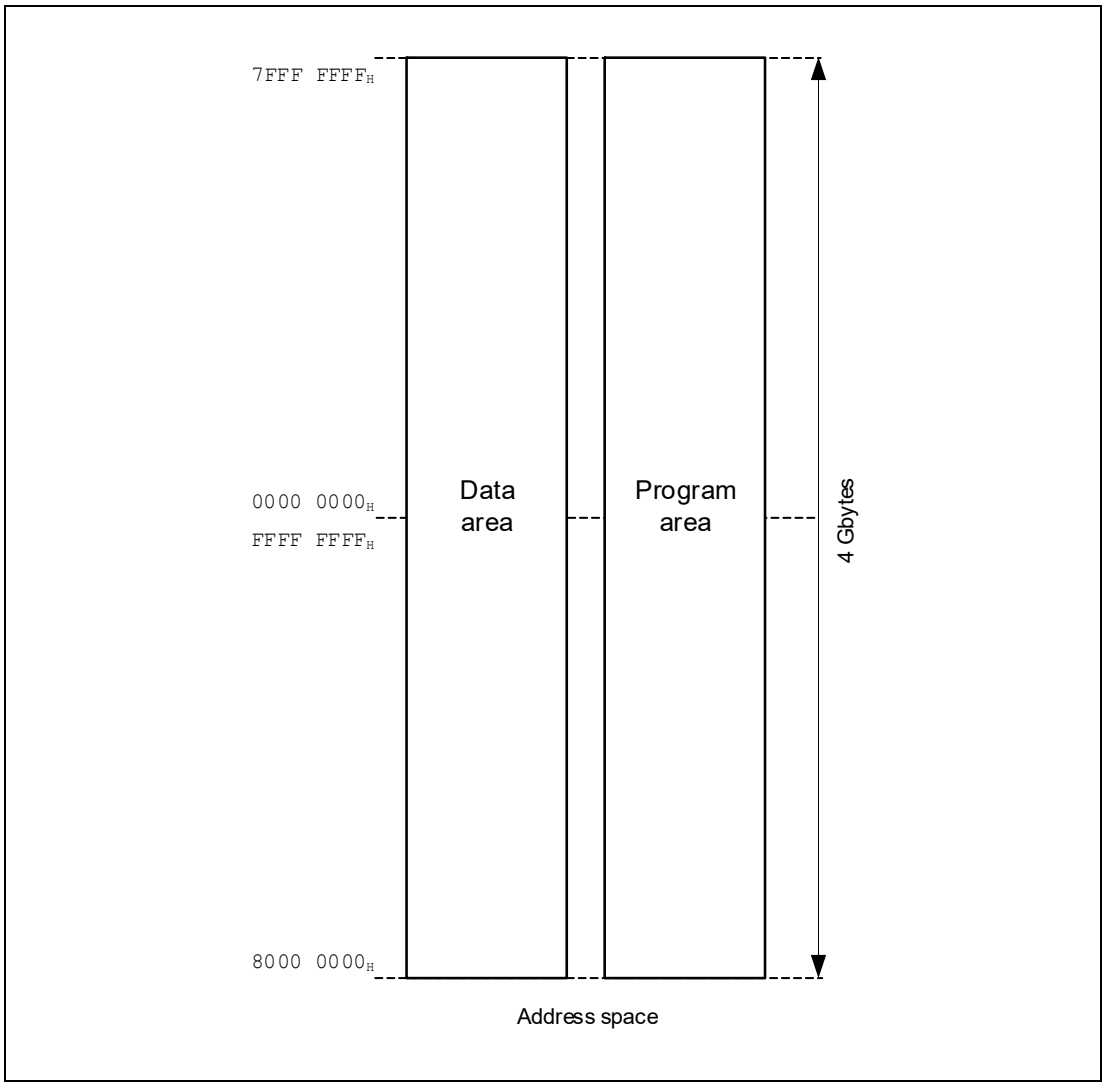


Figure 3.6 Memory Map (Address Space)

(2) Instruction Addressing

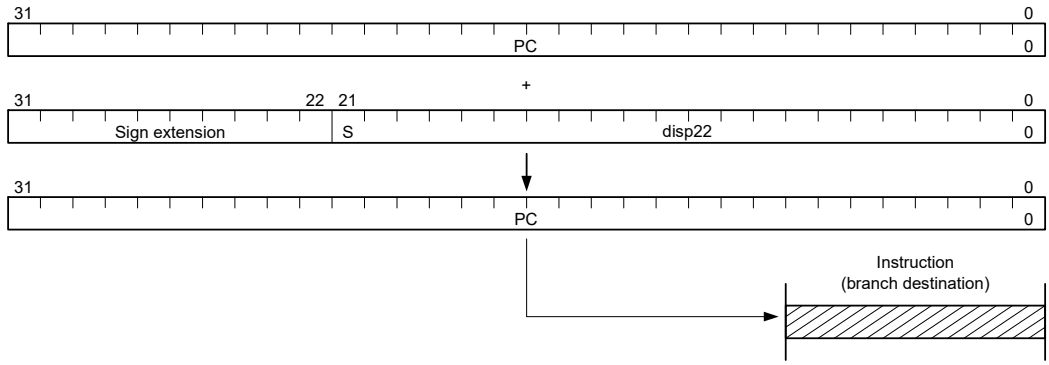
The instruction address is determined based on the contents of the program counter (PC), and is automatically incremented according to the number of bytes in the executed instruction. When a branch instruction is executed, the addressing shown below is used to set the branch destination address to the PC.

If the result of address calculation exceeds the positive maximum value FFFF $FFFF_H$ by addition, it is wrapped around to 0000 0000_H. If the result of address calculation falls below the positive minimum value 0000 0000_H by subtraction, it is wrapped around to FFFF $FFFF_H$.

(a) Relative Addressing (PC Relative)

Signed N-bit data (displacement: disp N) in the instruction code is added to the program counter (PC). In this case, displacement is handled as 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher-order bits are sign-extended (N differs from one instruction to another).

The JARL, JR, and Bcond instructions are used with this type of addressing.



Note: This is an example of 22-bit displacement.

Figure 3.7 Relative Addressing

(b) Register Addressing (Register Indirect)

The contents of the general-purpose register (reg1) or system register (regID) specified by the instruction are transferred to the program counter (PC).

The JMP, CTRET, EIRET, FERET, and DISPOSE instructions are used with this type of addressing.

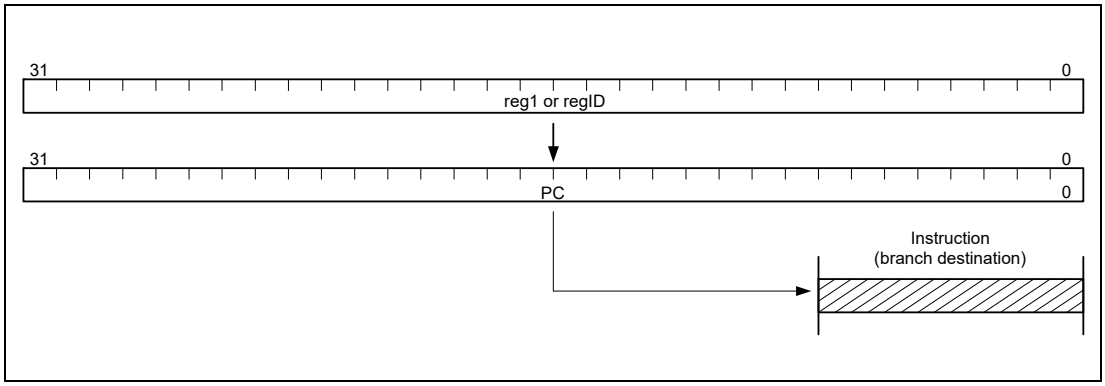


Figure 3.8 Register Addressing

(c) Based Addressing

The contents of the general-purpose register (reg1) and the N-bit displacement (dispN) specified by the instruction are added and transferred to the program counter (PC). At this time, the displacement is handled as a 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The JMP instruction is used with this type of addressing.

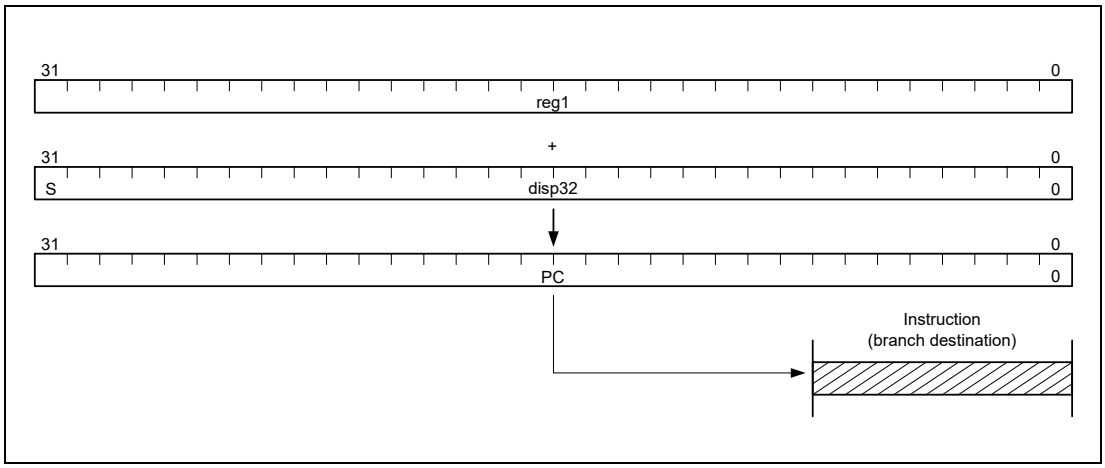


Figure 3.9 Based Addressing

(d) Other Addressing

A value specified by an instruction is transferred to the program counter (PC). How a value is specified is explained in [Operation] or [Description] of each instruction.

The CALLT, SYSCALL, TRAP, FETRAP, and RIE instructions, and branch in case of an exception are used with this type of addressing.

(3) Data Addressing

The following methods can be used to access the target registers or memory when executing an instruction.

If the result of address calculation exceeds the positive maximum value FFFF $FFFF_H$ by addition, it is wrapped around to 0000 0000_H. If the result of address calculation falls below the positive minimum value 0000 0000_H by subtraction, it is wrapped around to FFFF $FFFF_H$.

(a) Register Addressing

This addressing method accesses the general-purpose register or system register specified in the general-purpose register field as an operand.

Any instruction that includes the operand reg1, reg2, reg3, or regID is used with this type of addressing.

(b) Immediate Addressing

This address mode uses arbitrary size data as the operation target in the instruction code.

Any instruction that includes the operand imm5, imm16, vector, or cccc is used with this type of addressing.

NOTE

vector: This is immediate data that specifies the exception vector (00_H to $1F_H$), and is an operand used by the TRAP, FETRAP, and SYSCALL instructions. The data width differs from one instruction to another.

cccc: This is 4-bit data that specifies a condition code, and is an operand used in the Bcond instruction, CMOV instruction, SASF instruction, and SETF instruction.

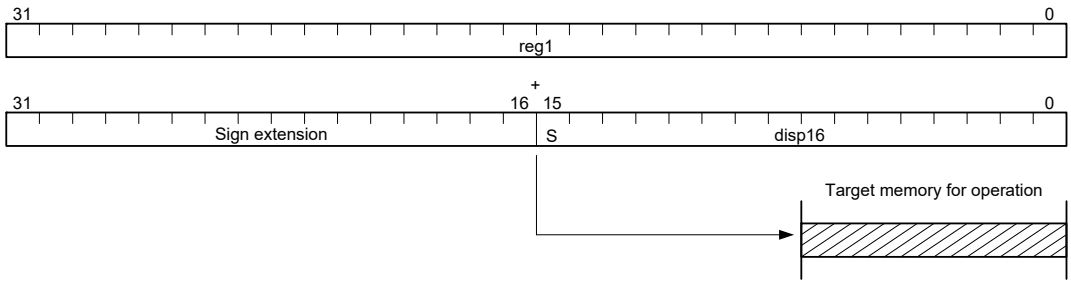
(c) Based Addressing

There are two types of based addressing, as described below.

Type 1

The contents of the general-purpose register (reg1) specified at the addressing specification field in the instruction code are added to the N-bit displacement (dispN) data sign-extended to word length to obtain the operand address, and the target memory is accessed using the operand address. At this time, the displacement is handled as a 2's complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The LD, ST, LDV and STV instructions are used with this type of addressing.



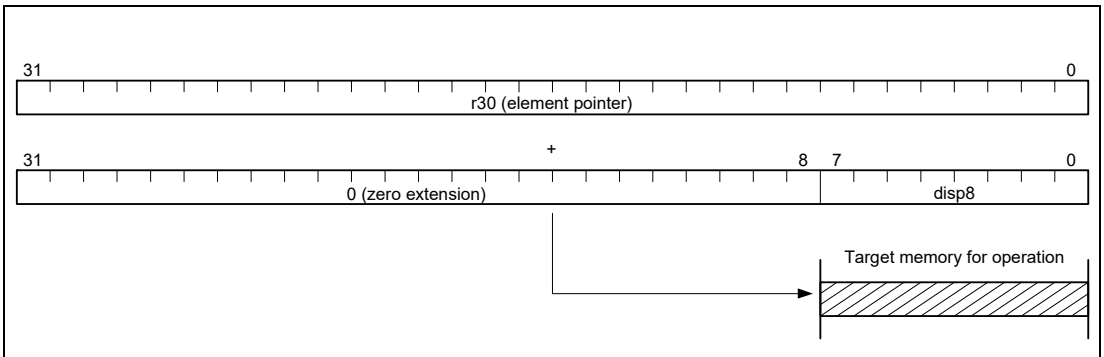
Note: This is an example of 16-bit displacement.

Figure 3.10 Based Addressing (Type 1)

Type 2

The contents of the element pointer (r30) are added to the N-bit displacement data (dispN) zero-extended to a word length to obtain the operand address, and target memory is accessed using the operand address. If the displacement is less than 32 bits, the higher bits are zero-extended (N differs from one instruction to another).

The SLD and SST instructions are used with this type of addressing.



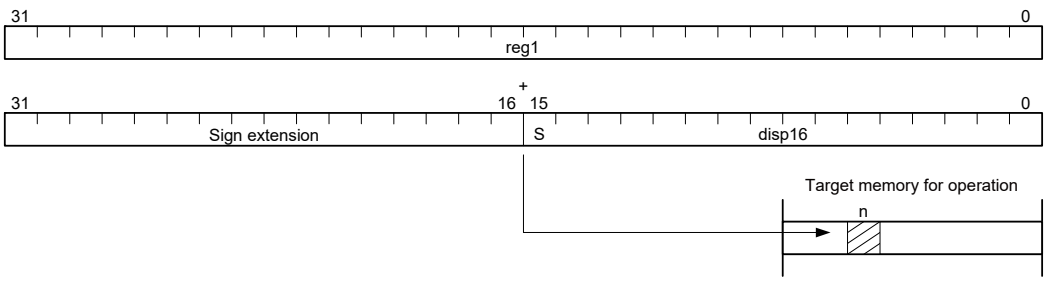
Note: This is an example of 8-bit displacement.

Figure 3.11 Based Addressing (Type 2)

(d) Bit Addressing

The contents of the general-purpose register (reg1) are added to the N-bit displacement (dispN) data sign-extended to word length to obtain the operand address, and one bit (as specified by 3-bit data “bit #3”) in one byte of the target memory is accessed using the operand address. At this time, the displacement is handled as a 2’s complement data, and the MSB is a sign bit (S). If the displacement is less than 32 bits, the higher bits are sign-extended (N differs from one instruction to another).

The CLR1, SET1, NOT1, and TST1 instructions are used with this type of addressing.



Note: n: Bit position specified by 3-bit data (bit #3) (n = 0 to 7)
This is an example of 16-bit displacement.

Figure 3.12 Bit Addressing

(e) Post Index Increment/Decrement Addressing

The contents of the general-purpose register (reg1) are used as an operand address to access the target memory, and then the general-purpose register (reg1) is updated. There are two types of the general-purpose register updating, by incrementing or decrementing.

If the result of incrementing the general-purpose register (reg1) value exceeds the positive maximum value FFFF $FFFF_H$, the result wraps around to 0000 0000_H, and, if the result of decrementing the general-purpose register value is less than the positive minimum value 0000 0000_H, the result wraps around to FFFF $FFFF_H$.

Type 1

The general-purpose register (reg1) is updated by adding a constant that depends on the type of accessed data (the size of the accessed data) to the contents of the general-purpose register (reg1). If the type of accessed data is a byte, 1 is added, if the type is a halfword, 2 is added, if the type is a word, 4 is added, and if the type is a double-word, 8 is added.

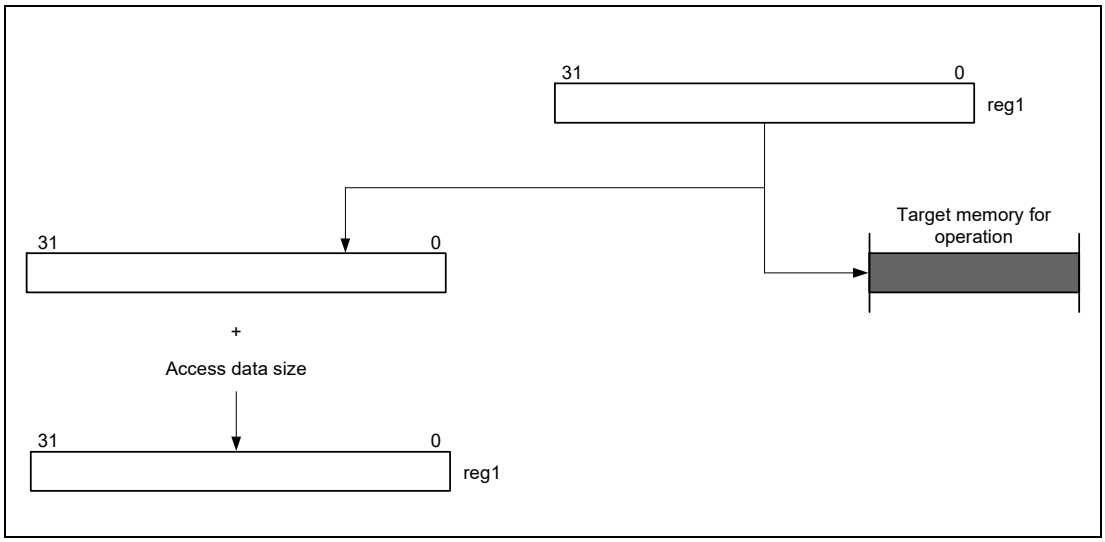


Figure 3.13 Post Index Increment/Decrement Addressing (Type 1)

Type 2

The general-purpose register (reg1) is updated by subtracting a constant that depends on the size of the accessed data from the contents of the general-purpose register (reg1). If the size of accessed data is a byte, 1 is subtracted, if the size is a halfword, 2 is subtracted, if the size is a word, 4 is subtracted, and if the size is a double-word, 8 is subtracted.

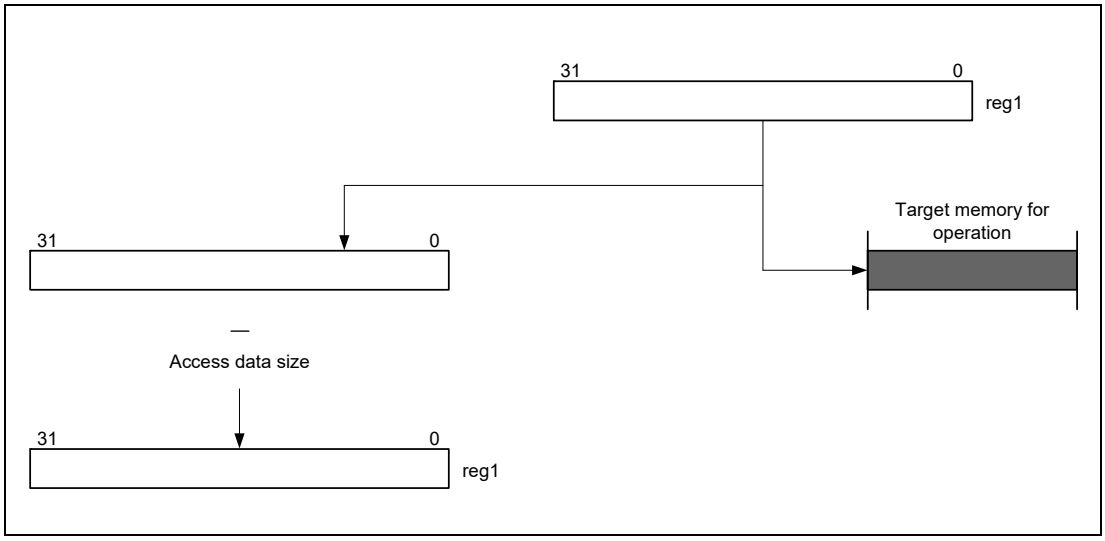


Figure 3.14 Post Index Increment/Decrement Addressing (Type 2)

(f) Other Addressing

The target memory is accessed using a value specified by an instruction as the operand address. How a value is specified is explained in [Operation] or [Description] of each instruction.

The SWITCH, CALLT, SYSCALL, PREPARE, DISPOSE, PUSHSP, POPSP, STM.MP, and LDM.MP instructions are used with this type of addressing.

3.2.2.8 Execution Timing of a Store Instruction

Completion of writing by a store instruction may differ depending on the type of the memory to be accessed. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

3.2.2.9 Memory Ordering

This CPU guarantees that memories are accessed in the programmed order. However, in the system that incorporates multiple bus masters such as the bus system with DMA or multi-core, the order of accesses to memories needs to be considered. For these cases, see **Section 3.2.7.1, Synchronization Processing**.

3.2.2.10 Acquiring the CPU Number

This CPU provides a method for identifying CPUs in a multi-processor system.

In the multi-processor configuration, you can identify which CPU core is running a program by referencing the PEID register. A unique number within a multi-processor system is assigned to the PEID register according to the specification of the product.

3.2.2.11 System Protection Identifier (SPID)

In this CPU, memory resources and peripheral devices are managed by system protection groups. By specifying the group to which the program being executed belongs, you can assign accessible memory resources and peripheral devices to the program.

The program being executed belongs to the group specified by the SPID, and whether the memory resources and peripheral devices are accessible is decided using the SPID. A value can be set to the SPID register by the supervisor.

CAUTION

According to the value of the SPID, how operations are assigned to memory resources and peripheral devices is determined by the specifications of the product.

3.2.2.12 Timestamp Counter

This CPU has a 64-bit timestamp counter. It can measure long time, and so can be used as specific information for time identification.

Since the timestamp counter is allocated to system registers, it can be accessed quickly by using a LDSR/STSR instruction.

If an overflow occurs during counting operation, no exception will occur.

(1) How to Operate the Timestamp Counter

The value of the timestamp counter is initialized to 0 by a reset. Therefore, if you want to retain the value of the counter across a reset, then before the reset, save the value of the counter in a memory that is not initialized by the reset, restore the value to the counter after the reset, and restart counting.

The counter is 64-bit width, so the counter consists of two 32-bit system registers, TSCOUNTL and TSCOUNTH. Both registers need to be accessed by using LDSR/STSR instructions.

When the counter is not running (the value of the TSCTRL.CEN bit is 0), no special care is needed to access the two registers.

However, if the counter is running (the value of the TSCTRL.CEN bit is 1), it is not recommended to update the counter by using LDSR instructions. In this case, the timing of update of the counter is not guaranteed. It is recommended to update the counter when it is not running.

Also, when reading the value of the counter by using STSR instructions while it is running, it is recommended to follow the procedure below.

```
TSCNTRD:
        STSR    1, r21, 11      - Read the upper side of the counter
        STSR    0, r20, 11      - Read the lower side of the counter
        STSR    1, r22, 11      - Read again the upper side of the counter
        CMP      r21, r22       - Compare the two values read from the upper side of the counter
        BNE      TSCNTRD        - If they are not identical, a carry has occurred. Read again.
```

Even if the CPU core is not in operation after the execution of a HALT or SNOOZE instruction, the timestamp counter continues counting.

3.2.2.13 Performance Measurement Function

This CPU has the performance measurement function. The performance measurement function can measure the performance of programs executed, the effects of interrupts generated during operation, etc. by counting the occurrence of the event specified by the PMCTRLn.CND bit.

The system registers used by the performance measurement function can be accessed only in supervisor mode after a reset. However, it can be accessed in user mode by changing the setting of the PMUMCTRL register.

The performance measurement function itself works in user mode regardless of the setting of the PMUMCTRL register. Even if all performance measurement channels are made inaccessible in user mode by using the PMUMCTRL register, configuration in supervisor mode allows performance measurement during operation in user mode.

This CPU has eight channels of system register set for the performance measurement function.

3.2.3 Register Set

This chapter describes the program register and system register mounted on this CPU.

3.2.3.1 Program Registers

Program registers includes general-purpose registers (r0 to r31) and the program counter (PC). r0 always retains 0. The values of the general-purpose registers r1 to r31 after a reset are undefined. The value of the PC after a reset is the value of RBASE register.

Table 3.6 Program Registers

Program Register	Name	Function	Description
General-purpose registers	r0	Zero register	Always retains 0
	r1	Assembler reserved register	Used as working register for generating addresses
	r2	Register for address and data variables (used when the real-time OS used does not use this register)	
	r3	Stack pointer (SP)	Used for generating a stack frame when a function is called
	r4	Global pointer (GP)	Used for accessing a global variable in the data area
	r5	Text pointer (TP)	Used as a register that indicates the start of the text area (area where program code is placed)
	r6 to r29	Register for addresses and data variables	
	r30	Element pointer (EP)	Used as a base pointer for generating addresses when accessing memory
	r31	Link pointer (LP)	Used when the compiler calls a function
Program counter	PC	Retains instruction addresses during execution of programs	

Note: For further descriptions of r1, r3 to r5, and r31 used for an assembler and/or C compiler, see the manual of each software development environment.

(1) General-purpose Registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables.

Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

(a) r0, r3, and r30

These registers are implicitly used by instructions.

r0 is a register that always retains 0. It is used for operations that use 0, addressing with base address being 0, etc.

r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.

r30 is used as a base pointer when the SLD instruction or SST instruction accesses memory.

(b) r1, r4, r5, and r31

These registers are implicitly used by the assembler and C compiler.

When using these registers, register contents must first be saved so they are not lost and can be restored after the registers are used.

(c) r2

This register is used by a real-time OS in some cases. If the real-time OS that is being used is not using r2, r2 can be used as a register for address variables or data variables.

(2) PC — Program Counter

The PC retains the address of the instruction being executed.

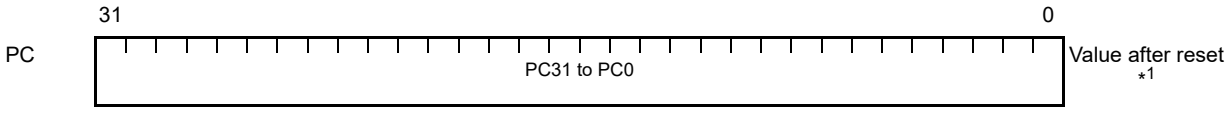


Table 3.7 PC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	PC31 to PC1	These bits indicate the address of the instruction being executed.	R/W	*1
0	PC0	This bit is fixed to 0, and branching to an odd number address is disabled.	R/W	0

Note 1. For details, see **Section 4, Address Space**.

3.2.3.2 Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

The basic system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.8 Basic System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 0	EIPC	Status save registers when acknowledging EI level exception	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	SV
SR2, 0	FEPC	Status save registers when acknowledging FE level exception	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	SV
SR5, 0	PSW	Program status word	*1
SR6, 0	FPSR	(See **Section 3.2.3.4, FPU Function Registers**)	CU0 and SV
SR7, 0	FPEPC	(See **Section 3.2.3.4, FPU Function Registers**)	CU0 and SV
SR8, 0	FPST	(See **Section 3.2.3.4, FPU Function Registers**)	CU0
SR9, 0	FPCC	(See **Section 3.2.3.4, FPU Function Registers**)	CU0
SR10, 0	FPCFG	(See **Section 3.2.3.4, FPU Function Registers**)	CU0
SR13, 0	EIIC	EI level exception cause	SV
SR14, 0	FEIC	FE level exception cause	SV
SR16, 0	CTPC	CALLT execution status save register	UM
SR17, 0	CTPSW	CALLT execution status save register	UM
SR20, 0	CTBP	CALLT base pointer	UM
SR21, 0	SNZCFG	SNOOZE control register	SV
SR28, 0	EIWR	EI level exception working register	SV
SR29, 0	FEWR	FE level exception working register	SV
SR0, 1	SPID	System protection identifier	SV
SR1, 1	SPIDLIST	List of system protection identifiers that can be specified in SPID	SV
SR2, 1	RBASE	Reset vector base address	SV
SR3, 1	EBASE	Exception handler vector address	SV
SR4, 1	INTBP	Base address of the interrupt handler “address” table	SV
SR5, 1	MCTL	CPU control	SV
SR6, 1	PID	Processor ID	SV
SR8, 1	SVLOCK	Supervisor lock	SV
SR11, 1	SCCFG	SYSCALL operation setting	SV
SR12, 1	SCBP	SYSCALL base pointer	SV
SR0, 2	PEID	Processor element identifier	UM
SR1, 2	BMID	Bus master identifier	UM
SR6, 2	MEA	Memory error address	SV
SR8, 2	MEI	Memory error information	SV
SR15, 2	RBCR0	Register bank control 0	SV
SR16, 2	RBCR1	Register bank control 1	SV
SR17, 2	RBNR	Register bank number	SV
SR18, 2	RBIP	Register bank initial pointer	SV

Note 1. The access permission differs depending on the bit. For details, see **Section 3.2.4.1(3), Types of Exceptions.**

(1) EIPC — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see **Section 3.2.4.1(3), Types of Exceptions**).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address must not be specified.

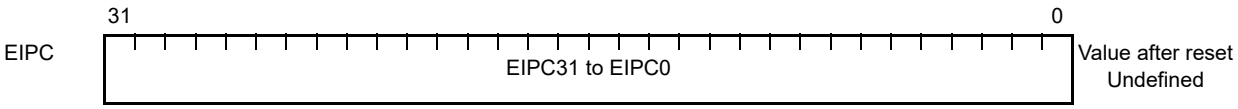


Table 3.9 EIPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	EIPC31 to EIPC1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC saved when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

(2) EIPSW — Status Save Register when Acknowledging EI Level Exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

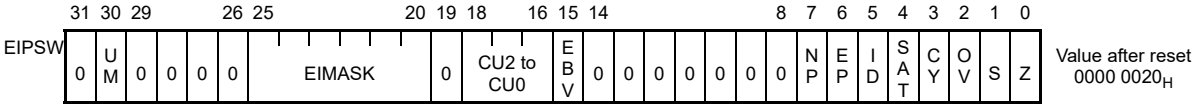


Table 3.10 EIPSW Register Contents

Bit	Name	Description	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 26	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
25 to 20	EIMASK	This bit stores the PSW.EIMASK bit setting when an EI level exception is acknowledged.*1	R/W	0
19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits store the PSW.CU2-0 field setting when an EI level exception is acknowledged.*2	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

Note 1. Only if INTCFG.EPL is set to 1, the value other than 0 can be set in this field. If INTCFG.EPL is cleared to 0, the value of this field becomes 0. Note that If INTCFG.EPL is cleared to 0 when the value of this field is other than 0, the value of this field becomes 0.

Note 2. CU2 is reserved for future CPUs that are to be made compatible with this CPU. It is always set to 0 in this CPU.

(3) FEPC — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see **Section 3.2.4.1(3), Types of Exceptions**). Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address must not be specified.

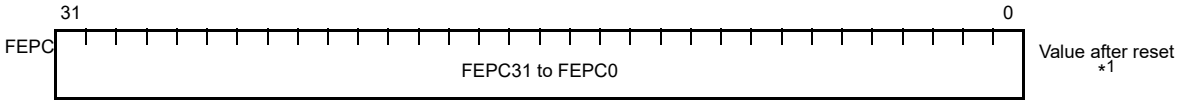


Table 3.11 FEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	FEPC31 to 1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	*1
0	FEPC0	This bit indicates the PC saved when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	*1

Note 1. When a reset occurs, among the instructions that completed execution before the reset occurred the value of the program counter of the instruction that was executed last is saved. If there is no execution completed instruction before the reset occurs, the value after reset is undefined. There is no information that identifies whether the value after reset is the value of the program counter of the execution completed instruction or undefined.

(4) FEPSW — Status Save Register when Acknowledging FE Level Exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

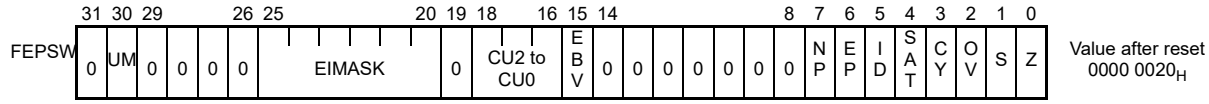


Table 3.12 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 26	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
25 to 20	EIMASK	These bits store the PSW.EIMASK field setting when an FE level exception is acknowledged*1.	R/W	0
19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits store the PSW.CU2-0 field setting when an FE level exception is acknowledged*2.	R/W	0
15	EBV	This bit stores the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

Note 1. Only if INTCFG.EPL is set to 1, the value other than 0 can be set in this field. If INTCFG.EPL is cleared to 0, the value of this field becomes 0. Note that If INTCFG.EPL is cleared to 0 when the value of this field is other than 0, the value of this field becomes 0.

Note 2. CU2 is reserved for future CPUs that are to be made compatible with this CPU. It is always set to 0 in this CPU.

(5) PSW — Program Status Word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by a condition instruction (Bcond, CMOV, etc.)).

CAUTIONS

1. **When the LDSR instruction is used to change the contents of this register, the changed contents become valid from the subsequent instruction. For details, see Section 3.2.7.3, Hazard Management after System Register Update.**
2. **The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See Table 3.13 for the access permission for each bit**

Table 3.13 Access Permission for PSW Register

Bit		Access Permission when Reading	Access Permission when Writing
30	UM	UM	SV[*1]
25 to 20	EIMASK		SV[*1]
18 to 16	CU2 to CU0		SV[*1]
15	EBV		SV[*1]
7	NP		SV[*1]
6	EP		SV[*1]
5	ID		SV[*1]
4	SAT		UM
3	CY		UM
2	OV		UM
1	S		UM
0	Z		UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

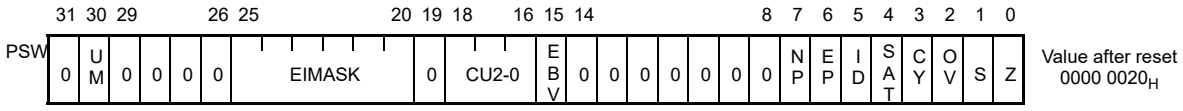


Table 3.14 PSW Register Contents (1/2)

Bit	Name	Description	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
30	UM	This bit indicates that the CPU is in user mode (in UM mode). 0: Supervisor mode 1: User mode	R/W	0
29 to 26	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
25 to 20	EIMASK	This field indicates the interrupt priority boundary between enabling and disabling an acknowlegement of an interrupt (EIINTn). For an interrupt (EIINTn) with higher priority than the value set in this field, the acknowlegement is enabled. For an interrupt (EIINTn) with priority less than or equal to the value set in this field, the acknowlegement is disabled. 0: All priorities are not acceptable 1: Acceptable priority is 0 2: Acceptable priority is 0 to 1 ... 62: Acceptable priority is 0 to 61 63: Acceptable priority is 0 to 62 Only if the INTCFG.EPL is set to 1, interrupt acknowledgment control is performed by the value of this field. If the INTCFG.EPL is cleared to 0, interrupt acknowledgment control by the value of this field is not performed*1. If the INTCFG.EPL is set to 1 and an interrupt (EIINTn) is acknowledged, the interrupt priority is saved in this field as part of PSW change due to acknowledgment of interrupt (EIINTn). The interrupt (EIINTn) specified as 63 as the priority is always disabled. However, the CPU halt state due to HALT or SNOOZE is also released by an interrupt (EIINTn) with priority 63.	R/W	0
19	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
18 to 16	CU2 to CU0	These bits indicate the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor unusable exception occurs if an instruction for the coprocessor is executed or a coprocessor resource (system register) is accessed. Bit 18 (CU2): Fixed to 0*2. Bit 17 (CU1): FXU Bit 16 (CU0): FPU CU2 to CU0 are fixed to 0 in the devices that do not have corresponding coprocessors.	R/W	000
15	EBV	This bit indicates the reset vector and exception vector operation. See **Section 3.2.3.2(16), RBASE — Reset Vector Base Address** and **Section 3.2.3.2(17), EBASE — Exception Handler Vector Address.**	R/W	0
14 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	NP	This bit disables the acknowledgement of FE level exception. When an FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level and FE level exceptions. As for the exceptions which the NP bit disables the acknowledgment, see **Table 3.102, Exception Cause List**. 0: The acknowledgement of FE level exception is enabled. 1: The acknowledgement of FE level exception is disabled.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0

Table 3.14 PSW Register Contents (2/2)

Bit	Name	Description	R/W	Value after Reset
5	ID	This bit disables the acknowledgement of EI level exception. When an EI level or FE level exception is acknowledged, this bit is set to 1 to disable the acknowledgement of EI level exception. As for the exceptions which the ID bit disables the acknowledgment, see **Table 3.102, Exception Cause List** This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. The change of the ID bit by the EI or ID instruction will be enabled from the next instruction. 0: The acknowledgement of EI level exception is enabled. 1: The acknowledgement of EI level exception is disabled.	R/W	1
4	SAT*3	This bit indicates that a saturation arithmetic operation instruction resulted in overflow and saturation processing is applied to the result. This is a cumulative flag, that is, it is set (1) once a saturation occurs and not cleared (0) by subsequent instructions with unsaturated results. This bit is cleared by the LDSR instruction. Note that execution of an arithmetic operation instruction neither set nor clear this flag. 0: The result was not saturated 1: The result was saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV*3	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S*3	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative.	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. Only if INTCFG.EPL = 1, the value other than 0 can be set in this field. If INTCFG.EPL = 0, the value of this field becomes 0. Note that if INTCFG.EPL is cleared to 0 while the value of this field is other than 0, the value of this field becomes 0.

Note 2. The coprocessor use permission CU2 is reserved for future CPUs that are to be made compatible with this CPU.

Note 3. Saturation processing is applied to the operation result in accordance with the contents of the OV and S flags. The SAT flag is set (1) only when the OV flag is set (1) in the saturation arithmetic operation.

Operation Result Status	Flag Status			Operation Result after Saturation Processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF$_H$
Exceeded negative maximum value	1	1	1	8000 0000$_H$
Positive (maximum value not exceeded)	Value prior to operation is retained	0	0	Operation result as is
Negative (maximum value not exceeded)			1	

(6) EIIC — EI Level Exception Cause

The EIIC register retains the cause of any EI level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception cause (see **Table 3.102, Exception Cause List**).

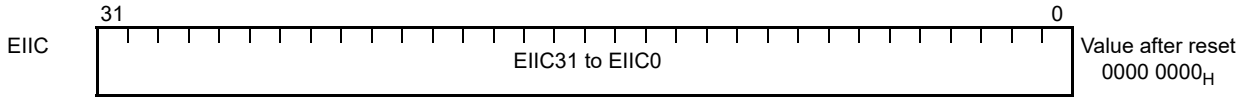


Table 3.15 EIIC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	EIIC31 to EIIC0	These bits store the exception cause code when an EI level exception occurs. The EIIC15-0 field stores the exception cause codes shown in **Table 3.102**. The EIIC31-16 field stores detailed exception cause codes defined individually for each exception. If there is no particular definition, these bits are set to 0.	R/W	0

(7) FEIC — FE Level Exception Cause

The FEIC register retains the cause of any FE level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception cause (see **Table 3.102, Exception Cause List**).

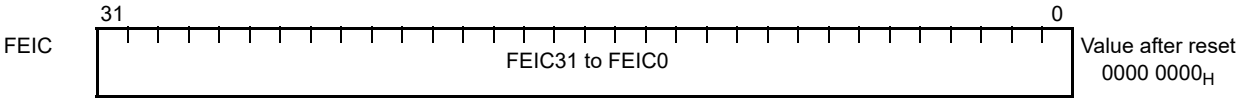


Table 3.16 FEIC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	FEIC31 to FEIC0	These bits store the exception cause code when an FE level exception occurs. The FEIC15-0 field stores the exception cause codes shown in **Table 3.102**. The FEIC31-16 field stores detailed exception cause codes defined individually for each exception. If there is no particular definition, these bits are set to 0.	R/W	0

(8) CTPC — Status Save Register when Executing CALLT

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC.

Be sure to set an even-numbered address to the CTPC register. An odd-numbered address must not be specified.

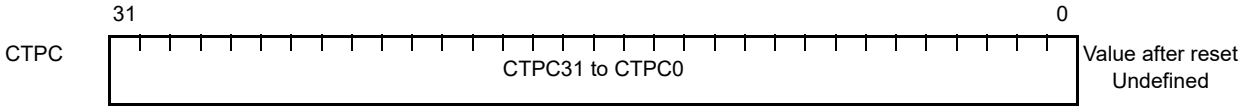


Table 3.17 CTPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	CTPC31 to CTPC1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

(9) CTPSW — Status Save Register when Executing CALLT

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

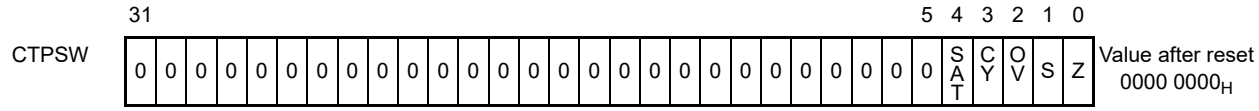


Table 3.18 CTPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4	SAT	This bit stores the PSW.SAT bit setting when the CALLT instruction is executed.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when the CALLT instruction is executed.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when the CALLT instruction is executed.	R/W	0
1	S	This bit stores the PSW.S bit setting when the CALLT instruction is executed.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when the CALLT instruction is executed.	R/W	0

(10) CTBP — CALLT Base Pointer

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

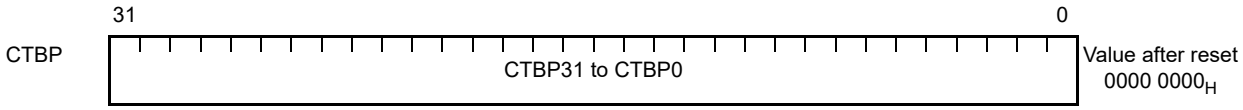


Table 3.19 CTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	CTBP31 to CTBP1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the start address of the table used by the CALLT instruction.	R/W	0
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the start address of the table used by the CALLT instruction. Always set this bit to 0.	R	0

(11) SNZCFG — SNOOZE Configuration

The SNZCFG register is used to configure the operation of the SNOOZE instruction.

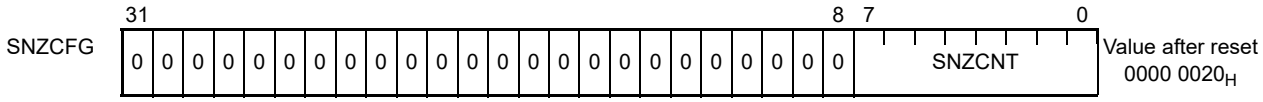


Table 3.20 SNZCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SNZCNT	These bits specify the number of clock cycles in which execution is to be temporarily halted by the SNOOZE instruction.	R/W	20_H

(12) EIWR — EI Level Exception Working Register

The EIWR register is used as a working register when an EI level exception has occurred.

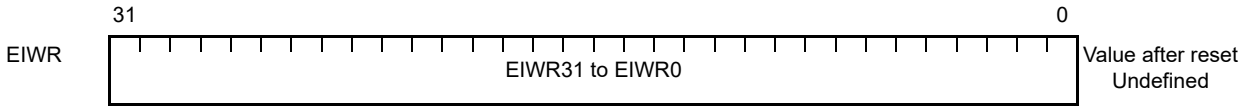


Table 3.21 EIWR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	EIWR31 to EIWR0	These bits constitute a working register that can be used for any purpose during the processing of an EI level exception. Use this register for purposes such as storing the values of general-purpose registers.	R/W	Undefined

(13) FEWR — FE Level Exception Working Register

The FEWR register is used as a working register when an FE level exception has occurred.

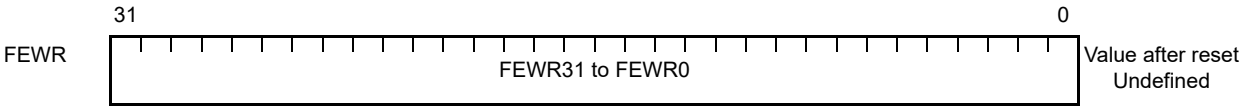


Table 3.22 FEWR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	FEWR31 to FEWR0	These bits constitute a working register that can be used for any purpose during the processing of an FE level exception. Use this register for purposes such as storing the values of general-purpose registers.	R/W	Undefined

(14) SPID — System Protection Identifier

The SPID register holds the system protection identifier of the CPU.

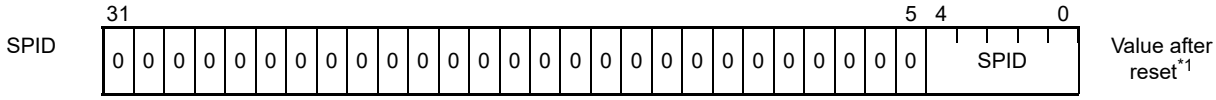


Table 3.23 SPID Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits indicate the system protection identifier. The system protection identifier is a variable ID that is used for access protection in a product which consists of two or more bus masters including this CPU. For its uses and constraints on its value, see **Section 3.9.7, Product information of initial value for G4MH register** . Within this CPU, the SPID is used to check for area matching by the MPU. It allows the system specifications defined for the product to be reflected in the MPU's protection feature. The settable system protection identifiers are given by the SPIDLIST register. If an attempt is made to set an illegal system protection identifier, the SPID register is not updated and retains the original value.	R/W	*1

Note 1. See **Section 3.9.7, Product information of initial value for G4MH register**.

(15) SPIDLIST — Legitimate System Protection Identifier List

The SPIDLIST register contains a list of system protection identifiers that can be set to the SPID register.

The bits corresponding to the settable system protection identifiers are set to 1. The bits corresponding to illegal system protection identifiers are cleared to 0. These values are set outside the CPU as system specifications and cannot be altered by this CPU.

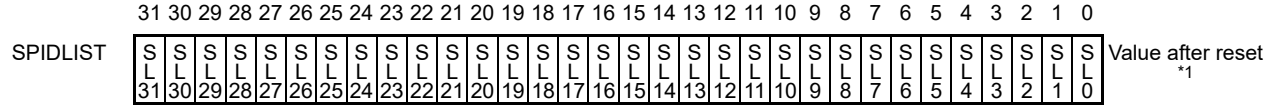


Table 3.24 SPIDLIST Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	SL31	This bit indicates whether or not 31 can be set as a system protection identifier.	R	*1
30	SL30	This bit indicates whether or not 30 can be set as a system protection identifier.	R	*1
29	SL29	This bit indicates whether or not 29 can be set as a system protection identifier.	R	*1
28	SL28	This bit indicates whether or not 28 can be set as a system protection identifier.	R	*1
27	SL27	This bit indicates whether or not 27 can be set as a system protection identifier.	R	*1
26	SL26	This bit indicates whether or not 26 can be set as a system protection identifier.	R	*1
25	SL25	This bit indicates whether or not 25 can be set as a system protection identifier.	R	*1
24	SL24	This bit indicates whether or not 24 can be set as a system protection identifier.	R	*1
23	SL23	This bit indicates whether or not 23 can be set as a system protection identifier.	R	*1
22	SL22	This bit indicates whether or not 22 can be set as a system protection identifier.	R	*1
21	SL21	This bit indicates whether or not 21 can be set as a system protection identifier.	R	*1
20	SL20	This bit indicates whether or not 20 can be set as a system protection identifier.	R	*1
19	SL19	This bit indicates whether or not 19 can be set as a system protection identifier.	R	*1
18	SL18	This bit indicates whether or not 18 can be set as a system protection identifier.	R	*1
17	SL17	This bit indicates whether or not 17 can be set as a system protection identifier.	R	*1
16	SL16	This bit indicates whether or not 16 can be set as a system protection identifier.	R	*1
15	SL15	This bit indicates whether or not 15 can be set as a system protection identifier.	R	*1
14	SL14	This bit indicates whether or not 14 can be set as a system protection identifier.	R	*1
13	SL13	This bit indicates whether or not 13 can be set as a system protection identifier.	R	*1
12	SL12	This bit indicates whether or not 12 can be set as a system protection identifier.	R	*1
11	SL11	This bit indicates whether or not 11 can be set as a system protection identifier.	R	*1
10	SL10	This bit indicates whether or not 10 can be set as a system protection identifier.	R	*1
9	SL9	This bit indicates whether or not 9 can be set as a system protection identifier.	R	*1
8	SL8	This bit indicates whether or not 8 can be set as a system protection identifier.	R	*1
7	SL7	This bit indicates whether or not 7 can be set as a system protection identifier.	R	*1
6	SL6	This bit indicates whether or not 6 can be set as a system protection identifier.	R	*1
5	SL5	This bit indicates whether or not 5 can be set as a system protection identifier.	R	*1
4	SL4	This bit indicates whether or not 4 can be set as a system protection identifier.	R	*1
3	SL3	This bit indicates whether or not 3 can be set as a system protection identifier.	R	*1
2	SL2	This bit indicates whether or not 2 can be set as a system protection identifier.	R	*1
1	SL1	This bit indicates whether or not 1 can be set as a system protection identifier.	R	*1
0	SL0	This bit indicates whether or not 0 can be set as a system protection identifier.	R	*1

Note 1. See **Section 3.9.7, Product information of initial value for G4MH register**.

(16) RBASE — Reset Vector Base Address

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this register indicates the exception handler vector address and the selection method of exception handler address.

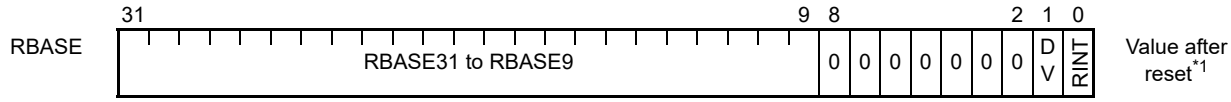


Table 3.25 RBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	RBASE31 to RBASE9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. The RBASE8 to RBASE0 bits are not assigned as names because these bits are always 0.	R	*1
8 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	DV	When the DV bit is set, the exception handler address for interrupt is determined by using the direct vector method. For details, see **Section 3.2.4.4(1)(b), Table Reference Method**. This bit is valid when PSW.EBV = 0.	R	*1
0	RINT	When the RINT bit is set, the exception handler address for interrupt processing is reduced. See **Section 3.2.4.4(1)(a), Direct Vector Method**. This bit is valid when PSW.EBV = 0.	R	*1

Note 1. See **Section 3.9.7, Product information of initial value for G4MH register**.

(17) EBASE — Exception Handler Vector Address

This register indicates the exception handler vector address and the selection method of exception handler address. This register is valid when the PSW.EBV bit is 1.

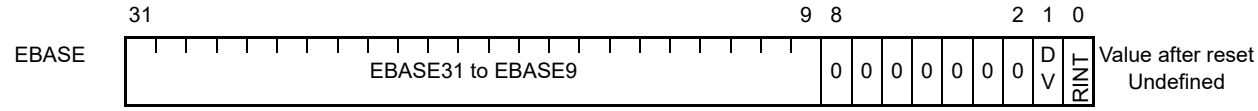


Table 3.26 EBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	EBASE31 to EBASE9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. The EBASE8 to EBASE0 bits are not assigned as names because these bits are always 0.	R/W	Undefined
8 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	DV	When the DV bit is set, the exception handler address for interrupt is determined by using the direct vector method. For details, see **Section 3.2.4.4(1)(b), Table Reference Method**.	R/W	Undefined
0	RINT	When the RINT bit is set, the exception handler address for interrupt processing is reduced. See **Section 3.2.4.4(1)(a), Direct Vector Method**.	R/W	Undefined

(18) INTBP — Base Address of the Interrupt Handler Address Table

This register indicates the base address of the table when the table reference method is selected as the interrupt handler address selection method.

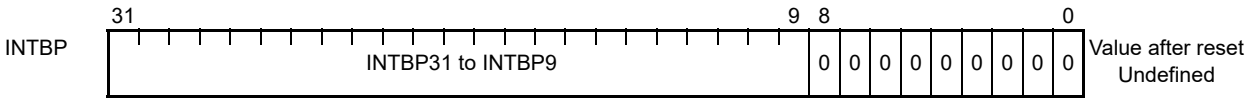


Table 3.27 INTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	INTBP31 to INTBP9	These bits indicate the base pointer address for an interrupt when the table reference method is used. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupt specified by the table reference method (EIINTn) is acknowledged. The INTBP8 to INTBP0 bits are not assigned as names because these bits are always 0.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(19) MCTL — Machine Control

The MCTL register is used to control the CPU.

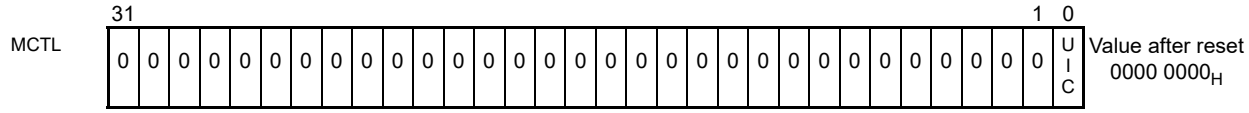


Table 3.28 MCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction in user mode become possible.	R/W	0

(20) PID — Processor ID

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

CAUTION

The PID register indicates information used to identify the incorporated CPU core and CPU core configuration. Usage such that the software behavior varies dynamically according to the PID register information is not assumed.

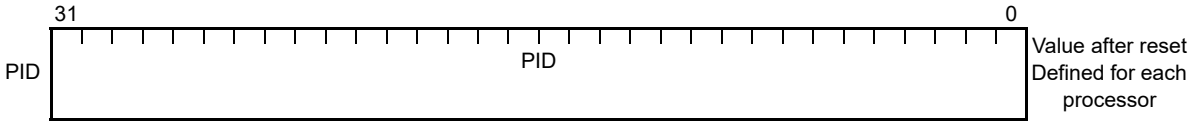


Table 3.29 PID Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.	R	*1
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bit 23 to 19: Reserved Bit 18: Register bank Bit 17 to 12: Reserved Bit 11: Extended floating-point operation function Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operation function Bit 8: Memory protection unit (MPU) function **Note:** If a double-precision floating-point operation function is implemented (when bit 10 is 1), a single-precision floating-point operation function is also always implemented (bit 9 is 1).	R	*1
7 to 0		Version Identifier This identifier indicates the version of the processor.	R	*1

Note 1. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

(21) SVLOCK — Supervisor Lock

The SVLOCK register is used to restrict the CPU operation in supervisor mode.

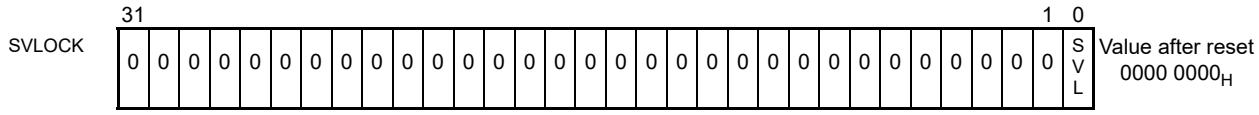


Table 3.30 SVLOCK Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	SVL	This bit specifies whether to restrict the CPU operation in supervisor mode. 0: Does not restrict the CPU operation in supervisor mode. 1: Restrict the CPU operation in supervisor mode. If the SVL bit is set to 1, the following system registers*1 cannot be updated even when the CPU is in supervisor mode: SPID, MPM, MPLA, MPUA, MPAT, MPIDn, MPBK For details, see **Section 3.2.2.5(5), Supervisor Lock Setting**.	R/W	0

Note 1. The target system registers are those registers that are associated with memory accessing. This register prevents these registers from being rewritten carelessly and unintentional memory access from being performed outside the CPU.

(22) SCCFG — SYSCALL Operation Setting

This register is used to set operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

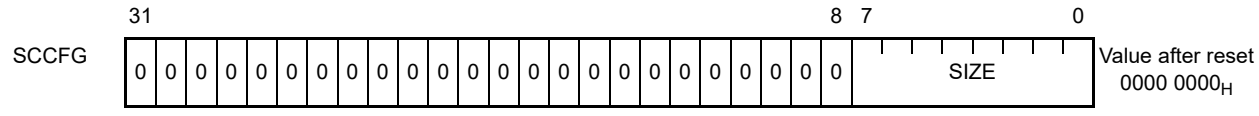


Table 3.31 SCCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	SIZE	These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If a vector exceeding the maximum number of entries is specified for the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry.	R/W	0

(23) SCBP — SYSCALL Base Pointer

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

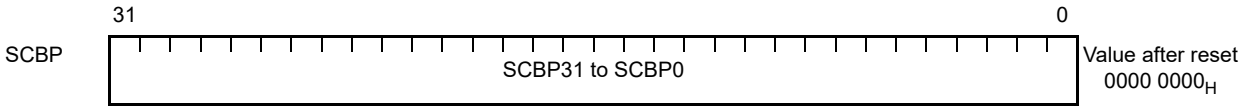


Table 3.32 SCBP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	SCBP31 to SCBP2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction.	R/W	0
1, 0	SCBP1, SCBP0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the start address of the table used by the SYSCALL instruction. Always set these bits to 0.	R	0

(24) PEID — Processor Element Identifier

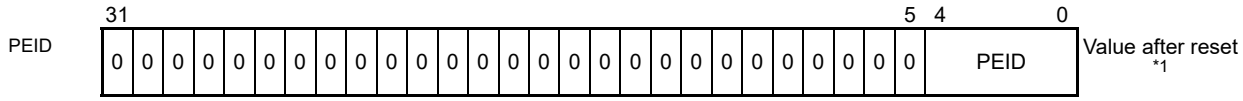


Table 3.33 PEID Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	PEID	These bits indicate the processor element identifier.	R	*1

Note 1. The processor element identifier of the CPU which is defined by the product specification is read. Writing to these bits is not possible. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

(25) BMID — Bus Master Identifier

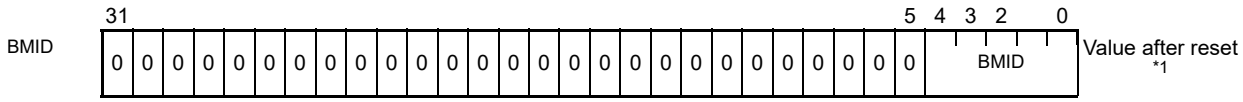


Table 3.34 BMID Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	BMID	These bits indicate the bus master identifier.	R	*1

Note 1. The bus master identifier of the CPU which is defined by the product specification is read. Writing to these bits is not possible. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

(26) MEA — Memory Error Address

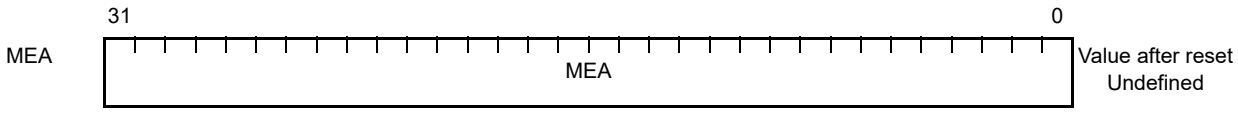


Table 3.35 MEA Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MEA	These bits holds the address in which an MAE (misalignment) or MPU violation occurred.	R/W	Undefined

(27) MEI — Memory Error Information

The MEI register holds the information about the instruction that caused a misalignment exception (MAE) or memory protection exception (MDP). The information can be used as hint information for the emulation by software.

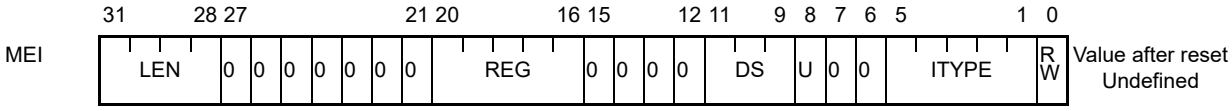


Table 3.36 MEI Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 28	LEN	These bits indicate the code size of the instruction that causes the exception. 0: Non-instruction factor 2: 16 bits 4: 32 bits 6: 48 bits 8 :64 bits Values other than those listed above are reserved for future use and never stored here. For details, see **Table 3.37**.	R/W	Undefined
27 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	REG	These bits indicate the source register number or destination register number of the instruction that caused the exception. For details, see **Table 3.37**.	R/W	Undefined
15 to 12	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
11 to 9	DS	These bits indicate the data type of the instruction that caused the exception*1. 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) 4: Quad-word (128 bits) Values other than those listed above are reserved for future use and never stored here. For details, see **Table 3.37**.	R/W	Undefined
8	U	This bit indicates the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see **Table 3.37**.	R/W	Undefined
7 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 1	ITYPE	These bits indicate the instruction that caused the exception. For details, see **Table 3.37**.	R/W	Undefined
0	RW	This bit indicates whether the operation performed by the instruction that caused the exception is a read (Load-memory) or a write (Store-memory). 0: Read (Load-memory) 1: Write (Store-memory) For details, see **Table 3.37**.	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3.37 Instructions Causing Exceptions and Values of MEI Register (1/2)

Instruction	LEN	REG	DS	U	RW	ITYPE
SLD.B	2 (16 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00000_B
SLD.BU	2 (16 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00000_B
SLD.H	2 (16 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00000_B
SLD.HU	2 (16 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00000_B
SLD.W	2 (16 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00000_B
SST.B	2 (16 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00000_B
SST.H	2 (16 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00000_B
SST.W	2 (16 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00000_B
LD.B (disp16)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00001_B
LD.BU (disp16)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00001_B
LD.H (disp16)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00001_B
LD.HU (disp16)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00001_B
LD.W (disp16)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00001_B
ST.B (disp16)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00001_B
ST.H (disp16)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00001_B
ST.W (disp16)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00001_B
LD.B (disp23)	6 (48 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00010_B
LD.BU (disp23)	6 (48 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00010_B
LD.H (disp23)	6 (48 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00010_B
LD.HU (disp23)	6 (48 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00010_B
LD.W (disp23)	6 (48 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00010_B
LD.DW (disp23)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	00010_B
ST.B (disp23)	6 (48 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00010_B
ST.H (disp23)	6 (48 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00010_B
ST.W (disp23)	6 (48 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00010_B
ST.DW (disp23)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	00010_B
LD.B (+)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00100_B
LD.BU (+)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00100_B
LD.H (+)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00100_B
LD.HU (+)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00100_B
LD.W (+)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00100_B
ST.B (+)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00100_B
ST.H (+)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00100_B
ST.W (+)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00100_B
LD.B (−)	4 (32 bits)	dst	0 (Byte)	0 (Signed)	0 (Read)	00101_B
LD.BU (−)	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00101_B
LD.H (−)	4 (32 bits)	dst	1 (Half-word)	0 (Signed)	0 (Read)	00101_B
LD.HU (−)	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00101_B
LD.W (−)	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00101_B
ST.B (−)	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00101_B
ST.H (−)	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00101_B
ST.W (−)	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00101_B
LDL.BU	4 (32 bits)	dst	0 (Byte)	1 (Unsigned)	0 (Read)	00111_B
LDL.HU	4 (32 bits)	dst	1 (Half-word)	1 (Unsigned)	0 (Read)	00111_B

Table 3.37 Instructions Causing Exceptions and Values of MEI Register (2/2)

Instruction	LEN	REG	DS	U	RW	ITYPE
LDL.W	4 (32 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	00111_B
STC.B	4 (32 bits)	src	0 (Byte)	0 (Signed)	1 (Write)	00111_B
STC.H	4 (32 bits)	src	1 (Half-word)	0 (Signed)	1 (Write)	00111_B
STC.W	4 (32 bits)	src	2 (Word)	0 (Signed)	1 (Write)	00111_B
CAXI	4 (32 bits)	dst*1	2 (Word)*1	0 (Signed)*1	0 (Read)*2	01000_B
SET1	4 (32 bits)	0*1	0 (Byte)*1	0 (Signed)*1	0 (Read)*2	01001_B
CLR1	4 (32 bits)	0*1	0 (Byte)*1	0 (Signed)*1	0 (Read)*2	01001_B
NOT1	4 (32 bits)	0*1	0 (Byte)*1	0 (Signed)*1	0 (Read)*2	01001_B
TST1	4 (32 bits)	0*1	0 (Byte)*1	0 (Signed)*1	0 (Read)	01001_B
PREPARE	*5,*6	src*1	2 (Word)*1	0 (Signed)*1	1 (Write)	01100_B
DISPOSE	4 (32 bits)	dst*1	2 (Word)*1	0 (Signed)*1	0 (Read)	01100_B
PUSHSP	4 (32 bits)	src*1	2 (Word)*1	0 (Signed)*1	1 (Write)	01101_B
POPSP	4 (32 bits)	dst*1,*3	2 (Word)*1	0 (Signed)*1	0 (Read)	01101_B
STM.MP	4 (32 bits)	0*1	2 (Word)*1	0 (Signed)*1	1 (Write)	01111_B
LDM.MP	4 (32 bits)	0*1	2 (Word)*1	0 (Signed)*1	0 (Read)	01111_B
SWITCH	2 (16 bits)	0*1	1 (Half-word)*1	0 (Signed)*1	0 (Read)	10000_B
CALLT	2 (16 bits)	0*1	1 (Half-word)*1	1 (Unsigned)*1	0 (Read)	10001_B
SYSCALL	4 (32 bits)	0*1	2 (Word)*1	0 (Signed)*1	0 (Read)	10010_B
CACHE	4 (32 bits)	0*1	0 (Byte)*1	0 (Signed)*1	0 (Read)	10100_B
Interrupt (table reference method)*4	0 (Non-instruction)	0*1	2 (Word)*1	0 (Signed)*1	0 (Read)	10101_B
Save onto register bank	0 (Non-instruction)	0*1	2 (Word)*1	0 (Signed)*1	1 (Write)	10110_B
RESBANK	4 (32 bits)	0*1	2 (Word)*1	0 (Signed)*1	0 (Read)	10110_B
LDV.W (disp 16)	6 (48 bits)	dst	2 (Word)	0 (Signed)	0 (Read)	11101_B
LDV.DW (disp 16)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	11101_B
LDV.QW (disp 16)	6 (48 bits)	dst	4 (Quad-word)	0 (Signed)	0 (Read)	11101_B
STV.W (disp 16)	6 (48 bits)	src	2 (Word)	0 (Signed)	1 (Write)	11101_B
STV.DW (disp 16)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	11101_B
STV.QW (disp 16)	6 (48 bits)	src	4 (Quad-word)	0 (Signed)	1 (Write)	11101_B
LDVZ.H4 (disp 16)	6 (48 bits)	dst	3 (Double-word)	0 (Signed)	0 (Read)	11111_B
STVZ.H4 (disp 16)	6 (48 bits)	src	3 (Double-word)	0 (Signed)	1 (Write)	11111_B

Note: dst: Destination register number; src: Source register number

Note 1. Specific to this CPU.

Note 2. This exception occurs when the instruction executes a read access.

Note 3. When the destination is r3, 0 is stored.

Note 4. When an exception occurs during a table reference that is triggered by an interrupt for which the table reference method is selected,

Note 5. In instruction format (1), “4 (32 bits)” is set.

Note 6. In instruction format (2), the value is selected as follows according to the value of the ff field in the instruction code:

ff = 00_B: 4 (32 bits)
ff = 01_B: 6 (48 bits)
ff = 10_B: 6 (48 bits)
ff = 11_B: 8 (64 bits)

(28) RBCR0 — Register Bank Control 0

The RBCR0 register controls the operation of the register bank function.

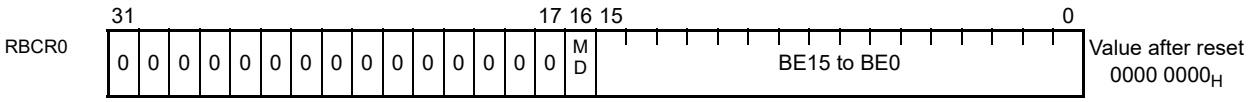


Table 3.38 RBCR0 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	MD	This bit specifies the save mode of the register bank. 0: Save mode 0 1: Save mode 1 List of Registers to be Saved Target Register / Save Mode 0 / Save Mode 1 PC ✓ ✓ PSW ✓ ✓ EIIC ✓ ✓ FPSR ✓ ✓ r1-r19 ✓ ✓ r20-r29 — ✓ r30 ✓ ✓ r31 — ✓ ✓: Saved —: Not saved	R/W	0
15	BE15	This bit specifies whether to use a register bank on an interrupt (EIINTn) with priority 15 to 63. 0: Do not use the register bank on an interrupt with the priority 15 to 63. 1: Use the register bank on an interrupt with the priority 15 to 63. While this bit is set to 1, if EI level interrupt (EIINTn) with priority 15 to 63 is acknowledged, the register bank function can be used.	R/W	0
14 to 0	BE14 to BE0	These bits specify the priority level of the interrupt on which the register bank is to be used. 0: Do not use the register bank on an interrupt of the level corresponding to this bit. 1: Use the register bank on an interrupt of the level corresponding to this bit. The bit positions correspond to the priority levels of the interrupts as follows: Bit / Priority Level 0 Level 0 (highest priority) 1 Level 1 : 14 Level 14 The register bank is used when an interrupt (EIINTn) of a level is accepted while the bit corresponding to that level is set (1).	R/W	0

Note 1. As the definition of this function, the interrupt (EIINTn) with priority 63 is included. However, since interrupts with priority 63 are always masked by PSW.EIMASK and PLMR registers, the register bank function is not used by the priority 63 interrupt (EIINTn).

(29) RBCR1 — Register Bank Control 1

The RBCR1 register controls the operation of the register bank function.

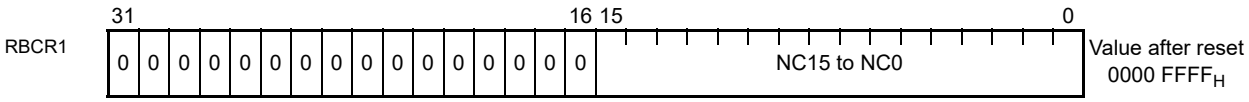


Table 3.39 RBCR1 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15	NC15	At the time accepting an interrupt (EIINTn) with priority 15 to 63*1 is acknowledged, the value to be set in the PSW.ID bit is specified. 0: Set PSW.ID = 0 for interrupt acknowledgment with priority 15 to 63 1: Set PSW.ID = 1 for interrupt acknowledgment with priority 15 to 63 (initial setting) However, this setting is valid only when the interrupt handler address selection method is Table Reference Method and the register bank is usable.	R/W	1
14 to 0	NC14 to NC0	These bits specify the values to be set to the PSW.ID bit when an interrupt (EIINTn) of the priority level corresponding to the bit position is accepted. 0: PSW.ID is set to 0 when an interrupt of the priority level corresponding to the bit position is accepted. 1: PSW.ID is set to 1 when an interrupt of the priority level corresponding to the bit position is accepted (initial value). This setting, however, is enabled only when the interrupt handler address selection method is set to the table reference method and the use of the register banks is enabled. The bit positions correspond to the priority levels of the interrupts as follows:	R/W	7FFF$_H$

Bit	Priority Level
0	Level 0 (highest priority)
1	Level 1
	:
14	Level 14

Note 1. As the definition of this function, the interrupt (EIINTn) with priority 63 is included. However, since interrupts with priority 63 are always masked by PSW.EIMASK and PLMR registers, the register bank function is not used by the priority 63 interrupt (EIINTn).

(30) RBNR — Register Bank Number

The RBNR register indicates the bank number of register bank function which is used next.

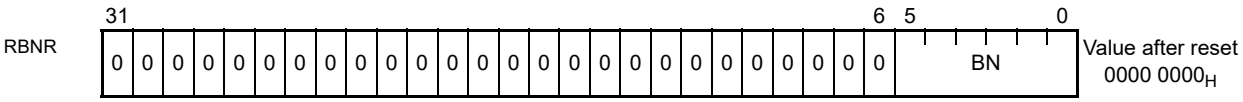


Table 3.40 RBNR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 0	BN	These bits indicate the number of the register bank to be used next. The value in these bits is incremented by 1 when an interrupt (EIINTn) that uses a register bank is accepted. If the value of BN is larger than INTCFG.ULNR, or if the value of BN is 63, if an interrupt (EIINTn) using the register bank occurs, a SYSERR exception is generated. That interrupt is not accepted, and it is put on hold. The value of BN is decremented by 1 when a RESBANK instruction is executed. If the RESBANK instruction is executed when the value of BN is 0, the instruction is not executed and a SYSERR exception is generated. In this case, BN is not updated and its value remains the same.	R/W	0

(31) RBIP — Register Bank Initial Pointer

The RBIP register indicates the start address of the memory area where the register bank is located.

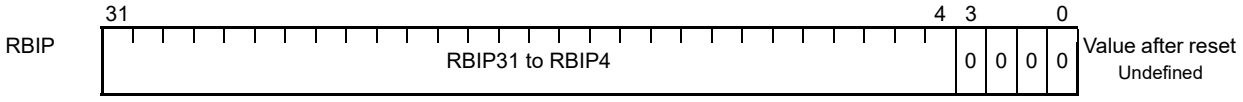


Table 3.41 RBIP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 4	RBIP31 to RBIP4	These bits indicate the initial pointer to the register bank. The values of RBCR0.MD and RBNR.BN together with this value determine the memory address to which the next register bank to be used is allocated as follows. • Save mode 0 (RBCR0.MD = 0): RBIP − RBNR.BN × 60_H • Save mode 1 (RBCR0.MD = 1): RBIP − RBNR.BN × 90_H	R/W	Undefined
3 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

3.2.3.3 Interrupt Function Registers

(a) Interrupt Function System Registers

Interrupt function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.42 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR10, 2	ISPR	Priority of interrupt being serviced	SV
SR11, 2	IMSR	Interrupt mask status	SV
SR12, 2	ICSR	Interrupt control status	SV
SR13, 2	INTCFG	Interrupt function setting	SV
SR14, 2	PLMR	Interrupt priority masking	SV

(1) ISPR — Priority of Interrupt being Serviced

The ISPR register holds the interrupt priority of the EI level interrupt (EIINTn) being processed by the CPU for each priority, priority ceiling by interrupt priority is performed when multiplexed interrupts occur.

For the ISPR register, the function becomes enabled if INTCFG.EPL is cleared to 0. When INTCFG.EPL is set to 1, the function of the ISPR register is disabled and acknowledgment of interrupt (EIINTn) is not controlled.

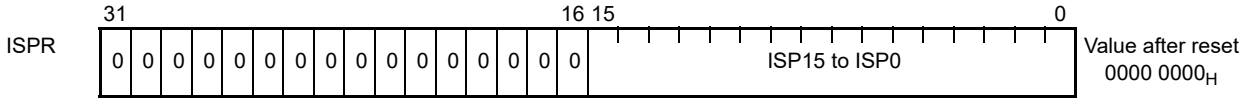


Table 3.43 ISPR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 0	ISP15 to ISP0	These bits indicate the acknowledgment status of an EIINTn interrupt with a priority*1 that corresponds to the relevant bit position. 0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged. 1: An interrupt request for an interrupt whose priority corresponds to the relevant position is being serviced by the CPU core. The bit positions correspond to the priority levels of the interrupts as follows: Bit / Priority 0 / Priority 0 (highest) 1 / Priority 1 : 14 / Priority 14 15 / Priority 15 (lowest) When an interrupt request (EIINTn) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1*4. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15 to ISP0 bits that are set (0 is the highest priority) is cleared to 0*2. While a bit in this register is set to 1, same or lower priority interrupts (EIINTn) are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged*5. For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking**. When performing software-based priority control using the PLMR register, be sure to clear this register by using the INTCFG.ISPC bit*6. If INTCFG.EPL is set to 1, the function of this field is disabled.	R*3	0

Note 1. For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking**.

Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.

Note 3. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

Note 4. If the priority of the acknowledged interrupt (EIINTn) is 16 to 63, neither bit of the ISP is set to 1.

Note 5. When an interrupt (EIINTn) with priority 16 to 63 is notified from the interrupt controller, if any bit of the ISP is set to 1, the interrupt (EIINTn) is masked.

Note 6. It is recommended that priority control by software using the PLMR register is performed with the INTCFG.ISPC bit set to 1 and automatic updating of ISPR disabled.

Note 7. The function of the ISPR register changes depending on the setting of INTCFG.EPL and INTCFG.ISPC.

INTCFG setting		Function of ISPR		
EPL	ISPC	Interrupt mask	ISP automatic update	Access by LDSR, STSR
0	0	Enabled	Enabled	Can Read Only
0	1	Enabled	Disabled	Can Read and Update
1	—	Disabled	Disabled	Treated as an undefined register

(2) IMSR — Interrupt Mask Status

The IMSR register is a register that indicates that interrupts notified to the CPU are masked for acceptance by the mask function in the CPU. Note that, for details of interrupt requests for updating each bit, see **Section 3.2.3.3 (a), Order of updating IMSR**.

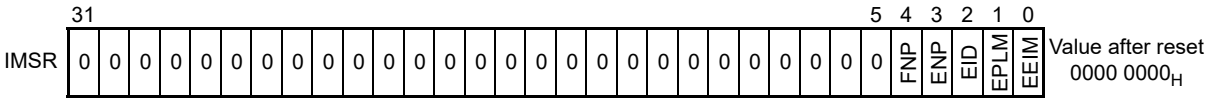


Table 3.44 IMSR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4	FNP	This bit specifies the existence of FEINT whose acceptance is masked by PSW.NP. 0: FEINT masked by PSW.NP does not exist 1: FEINT masked by PSW.NP does exists	R	0
3	ENP	This bit specifies the existence of EIINT whose acceptance is masked by PSW.NP. 0: EIINT masked by PSW.NP does not exist 1: EIINT masked by PSW.NP does exists	R	0
2	EID	This bit specifies the existence of EIINT whose acceptance is masked by PSW.ID. 0: EIINT masked by PSW.ID does not exist 1: EIINT masked by PSW.ID exists	R	0
1	EPLM	This bit specifies the existence of EIINT whose acceptance is masked by PLMR.PLM. 0: EIINT masked by PLMR.PLM does not exist 1: EIINT masked by PLMR.PLM exists	R	0
0	EEIM	This bit specifies the existence of EIINT whose acceptance is masked by ISPR.ISP when INTCFG.EPL is cleared to 0 or PSW.EIMASK when INTCFG.EPL is set to 1. 0: EIINT masked by ISPR.ISP or PSW.EIMASK does not exist 1: EIINT masked by ISPR.ISP or PSW.EIMASK exists	R	0

(a) Order of updating IMSR

In the interrupt notified to the CPU, the acceptance conditions are confirmed in the order of ISPR or PSW.EIMASK (selected by INTCFG.EPL), PLMR, PSW.ID, PSW.NP. If the acceptance condition is set to the value that masks the interrupt, the interrupt request is masked there, and the acceptance conditions that are in the subsequent order are not confirmed. Note that depending on the type of exception, the confirmation of conditions that are not acceptance conditions do not.

Since the interrupt (EIINTn) has multiple acceptance conditions, it is set one bit at a time according to the order of confirmation of the acceptance conditions as follows.

Table 3.45 Conditions which each bit of the IMSR are set (1) by masked of the interrupt (EIINTn)

Which bits are set (1)	Acceptance conditions				Value of IMSR			
	PSW.NP	PSW.ID	PLMR	*1	ENP	EID	EPLM	EEIM
ENP is set (1)	Disable	Enable	Enable	Enable	1	0*2	0*2	0*2
EID is set (1)	—	Disable	Enable	Enable	0*3	1	0*2	0*2
EPLM is set (1)	—	—	Disable	Enable	0*3	0*3	1	0*2
EEIM is set (1)	—	—	—	Disable	0*3	0*3	0*3	1

Note 1. If INTCFG.EPL is cleared (0), the acceptance condition is specified by ISPR, and if INTCFG.EPL is set (1), the acceptance condition is specified by PSW.EIMASK.

Note 2. If the interrupt (EIINTn) is masked under the acceptance condition, that interrupt is not masked by acceptance conditions at the right side of that condition. The bits corresponding to such unmasked acceptance conditions are cleared (0).

Note 3. If the interrupt (EIINTn) is masked under the acceptance condition, that interrupt is not confirmed by acceptance conditions at the left side of that condition. The bits corresponding to such unconfirmed acceptance conditions are cleared (0), even if the acceptance conditions is set to the value that masks the interrupt (EIINTn).

(3) ICSR — Interrupt Control Status

This register indicates the interrupt control status in the CPU.

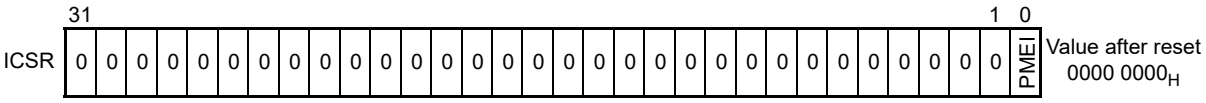


Table 3.46 ICSR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	PMEI	This bit indicates that an interrupt (EIINTn) with the priority level masked by the PLMR register exists. For details, see **Section 3.2.4.1(5), Interrupt Exception Priority and Priority Masking**.	R	0

(4) INTCFG — Interrupt Function Setting

This register is used to specify settings related to the CPU's internal interrupt function.

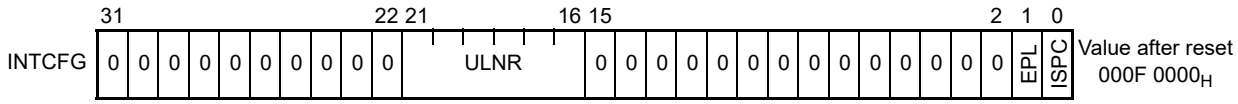


Table 3.47 INTCFG Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 22	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
21 to 16	ULNR	Specifying the maximum value of available register bank numbers. If the value of the RBNR.BN is bigger than the ULNR, or the value of the RBNR.BN is 63; and the interrupt (EIINTn) whose register bank function is enable occurs, the SYSERR exception will occur. Note that the interrupt (EIINTn) is not accepted and is held.	R/W	$0F_H$
15 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	EPL	For the interrupt (EIINTn), specify whether to enable interrupt priority level extension function. 0: Interrupt priority level extension function is disabled 1: Interrupt priority level extension function is enabled The following is the overview of interrupt operation by setting this bit. For details, see **Section 3.2.4.1 (5), Interrupt Exception Priority and Priority Masking** , **Section 3.2.4.4, Exception Handler Address**, **Section 3.2.4.5, Register Bank Function**.	R/W	0

Interrupt function		Value of EPL = 0	Value of EPL = 1
Number of interrupt priority level		16	64
Mask Function	ISPR	Function is enabled	Function is disabled
	PSW.EIMASK	Function is disabled	Function is enabled
	PLMR	Function is enabled	
	PSW.NP	Function is enabled	
	PSW.ID	Function is enabled	
INTCFG.ISPC		Settings can be changed	Fixed 0
ICSR.PMEI		Function is enabled	
IMSR		Function is enabled	
Handler generation	Direct vector method	Each vector can be used by each priority	Each vector can be used by priority 0 to 14. The priority 15 vector is shared with the priority is 15 to 63.
	Table reference method	Available	
Register bank		The usage can be set by each priority	Priority 0 to 14 can be individually settable. The priority 15 setting is shared with the priority is 15 to 63.

Table 3.47 INTCFG Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
0	ISPC	This bit changes how the ISPR register is written. 0: The ISPR register is automatically updated. Updates triggered by the program (via execution of LDSR instruction) are ignored. 1: The ISPR register is not automatically updated. Updates triggered by the program (via execution of LDSR instruction) are performed. If this bit is cleared to 0, the bits of the ISPR register are automatically set to 1 when an interrupt (EIINTn) is acknowledged, and cleared to 0 when the EIRET instruction is executed. In this case, the bits are not updated by an LDSR instruction executed by the program. If this bit is set to 1, the bits of the ISPR register are not updated by the acknowledgement of an interrupt (EIINTn) or by execution of the EIRET instruction. In this case, the bits can be updated by an LDSR instruction executed by the program. In normal cases, the ISPC bit should be cleared. When performing software-based priority control, however, set this bit (1). For details, see **Section 3.2.4.1(5)(b), Interrupt Priority Mask**.	R/W	0

(5) PLMR — Interrupt Priority Level Mask

This register masks the interrupts (EIINTn) whose priority level is not higher than the level specified by these bits.

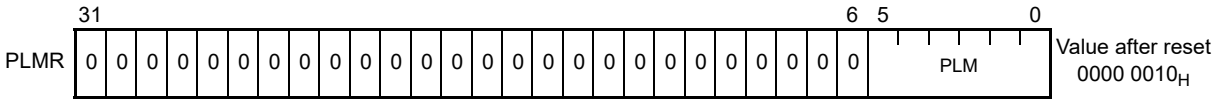


Table 3.48 PLMR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 0	PLM	These bits are used to mask the interrupts (EIINTn) whose priority level is not higher than the level specified by these bits. When an interrupt (EIINTn) is masked by this register, it is not accepted. The correspondence between the value of the PLM bit and the highest priority of interrupts to be masked is shown below. **Value of PLM Bit / Highest Priority of Interrupts to be Masked** 0 — Priority 0 (All priorities are not acceptable) 1 — Priority 1 (Acceptable of priority 0 only) : 14 — Priority 14 (Acceptable priority is 0 to 13) 15 — Priority 15 (Acceptable priority is 0 to 14) : 62 — Priority 62 (Acceptable priority is 0 to 61) 63 — Priority 63 (Acceptable priority is 0 to 62) Since the highest priority level of an interrupt that is defined for this CPU is 0, if 0 is specified in the PLM bit, all interrupts (EIINTn) are masked by this register. Since the lowest interrupt priority level defined by this CPU is 63, the interrupts (EIINTn) with the interrupt priority of 63 are always masked. Regardless of INTCFG.EPL setting, the interrupt mask by PLMR is done by the value of the PLM bit and the value of the interrupt priority notified from the interrupt controller. If the value of PLMR is altered by the LDSR instruction, the new PLMR value is reflected in the instructions following that LDSR instruction.	R/W	10_H

3.2.3.4 FPU Function Registers

(a) Floating-Point Registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations.

- Single-precision floating-point instruction:
  Thirty-two 32-bit registers can be specified. These general-purpose registers correspond to r0 to r31.
- Double-precision floating-point instruction:
  Sixteen 64-bit registers can be specified. Paired general-purpose registers are used as register pairs ({r1, r0}, {r3, r2} … {r31, r30}). Each register pair is specified in the instruction format with an even numbered register. Because r0 is a zero register (always holds 0), in principle {r1, r0} cannot be used by a double-precision floating-point instruction.

(b) Floating-Point Function System Registers

The FPU can use the following system registers to control floating-point operations. Floating-point function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

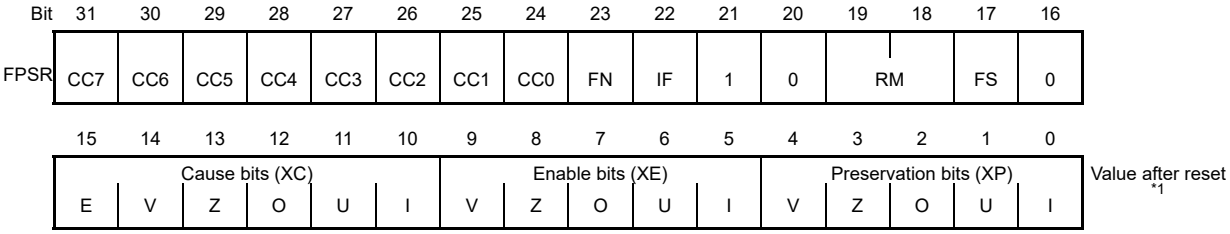
- FPSR: This register is used to control and monitor exceptions. It also holds the result of compare operations, and sets the FPU operation mode. Its bits are used to set condition code, subnormal number flush enable, rounding mode control, cause, exception enable, and preservation.
- FPEPC: This register stores the program counter value for the instruction where a floating-point operation exception has occurred.
- FPST: This register reflects the contents of the FPSR register bits related to the operation status.
- FPCC: This register reflects the contents of the FPSR.CC (7:0) bits.
- FPCFG: This register reflects the contents of the FPSR register bits related to the operation settings.

Table 3.49 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Floating-point operation configuration/status	CU0 and SV
SR7, 0	FPEPC	Floating-point operation exception program counter	CU0 and SV
SR8, 0	FPST	Floating point operation status	CU0
SR9, 0	FPCC	Floating-point operation comparison result	CU0
SR10, 0	FPCFG	Floating-point operation configuration	CU0

(1) FPSR — Floating-point Configuration/Status

This register indicates the execution status of floating-point operations and any exceptions that occur. For details about exception, see **Section 3.2.6.1 (5), Floating-Point Operation Exceptions**.



Note 1. See the descriptions of each bit.

Table 3.50 FPSR Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	CC[7:0]	These are the CC (condition) bits. They store the results of floating-point comparison instructions. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined
23	FN	This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number. For details, see **Section 3.2.6.1 (9), Flush to Nearest**.	R/W	0
22	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see **Section 3.2.6.1 (8), Flushing Subnormal Numbers**.	R/W	0
21	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
19, 18	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions.	R/W	00

RM Bits		Mnemonic	Description
19	18		
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.
1	0	RP	Rounds the result toward +∞. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.
1	1	RM	Rounds the result toward −∞. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.

Table 3.50 FPSR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th><th>R/W</th><th>Value After Reset</th></tr>
<tr><td>17</td><td>FS</td><td>This bit enables values that could not be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign. Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode.
<table><tr><th rowspan="2">Operation Result that is a Subnormal Number</th><th colspan="4">Rounding Mode and Value after Flushing</th></tr><tr><th>RN*1</th><th>RZ</th><th>RP</th><th>RM</th></tr><tr><td>Positive</td><td>+0</td><td>+0</td><td>$+2^{Emin}$</td><td>+0</td></tr><tr><td>Negative</td><td>−0</td><td>−0</td><td>−0</td><td>-2^{Emin}</td></tr></table>Note 1. If the rounding mode is RN and the FPSR.FN bit is set, flushing will occur in the direction of higher accuracy. For details, see Section 3.2.6.1 (9), Flush to Nearest.</td><td>R/W</td><td>1</td></tr>
<tr><td>16</td><td>—</td><td>(Reserved for future expansion. Be sure to set to 0.)</td><td>R</td><td>0</td></tr>
<tr><td>15 to 10</td><td>XC (E,V,Z,O,U,I)</td><td>These are the cause bits. For details, see Section 3.2.3.4(1), Cause Bits (XC).</td><td>R/W</td><td>Undefined</td></tr>
<tr><td>9 to 5</td><td>XE (V,Z,O,U,I)</td><td>These are the enable bits. For details, see Section 3.2.3.4(1), Enable Bits (XE).</td><td>R/W</td><td>0</td></tr>
<tr><td>4 to 0</td><td>XP (V,Z,O,U,I)</td><td>These are the preservation bits. For details, see Section 3.2.3.4(1), Preservation Bits (XP).</td><td>R/W</td><td>Undefined</td></tr>
</table>

Cause Bits (XC)

Bits 15 to 10 in the FPSR register are cause bits, which indicate the occurrence and cause of a floating-point operation exception. If an exception defined by IEEE754 is generated, when an enable bit is set to 1 corresponding to the exception, a cause bit is set, and the exception then occurs. When two or more exceptions occur during a single instruction, each corresponding bit is set to 1.

If two or more exceptions are detected, as long as the enable bit corresponding to one of the exceptions is set to 1, the exception occurs. In this case, the cause bits of all the detected exceptions, including exceptions whose enable bits are cleared to 0, are set to 1.

The cause bits are rewritten by a floating-point instruction (except the TRFSR instruction) where the floating-point operation exception occurred. The E bit is set to 1 when software emulation is required, otherwise it is cleared to 0. Other bits are set to 1 or cleared to 0 depending on whether or not an IEEE754-defined exception has occurred.

When a floating-point operation exception has occurred, the operation result is not stored, and only the cause bits are affected.

When the cause bits are set to 1 by an LDSR instruction, a floating-point operation exception does not occur.

Enable Bits (XE)

Bits 9 to 5 in the FPSR register are the enable bits, which enable floating-point operation exceptions. When an IEEE754-defined exception occurs, a floating-point operation exception occurs if the enable bit corresponding to the exception has been set to 1.

There are no enable bits corresponding to an unimplemented operation exception (E). An unimplemented operation exception (E) always occurs as a floating-point operation exception.

If the corresponding enable bit has not been set to 1, no exception occurs and the default result defined by IEEE754 is stored.

Preservation Bits (XP)

Bits 4 to 0 in the FPSR register are preservation bits. These bits store and indicate the detected exception after reset. An exception defined by IEEE754 occurs, and if a floating-point operation exception is not generated, the preservation bit is set to 1, otherwise it does not change. The preservation bits are not cleared to 0 by the floating-point operation. However, these bits can be set and cleared by software when an LDSR instruction is used to write a new value to the FPSR register.

There are no preservation bits corresponding to unimplemented operation exceptions (E). An unimplemented operation exception (E) always occurs as a floating-point operation exception.

NOTE

For details about the exception types and how they relate to particular bits, see **Figure 3.37, Cause, Enable, and Preservation Bits of FPSR Register**.

(2) FPEPC — Floating-point Exception Program Counter

When an exception that is enabled by an enable bit occurs, the program counter (PC) of the instruction that caused the exception is stored.

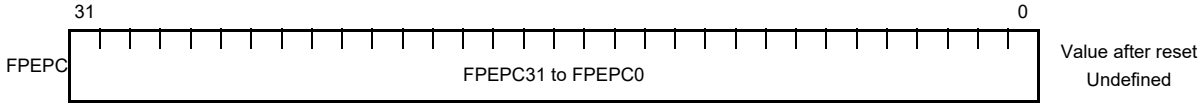


Table 3.51 FPEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	FPEPC31 to FPEPC1	These bits store the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R/W	Undefined
0	FPEPC0	This bit stores the program counter (PC) of the floating-point instruction that caused the exception when a floating-point operation exception that is enabled by an enable bit occurs.	R	0

(3) FPST — Floating-point Operation Status

This register reflects the contents of the FPSR register bits related to the operation status.

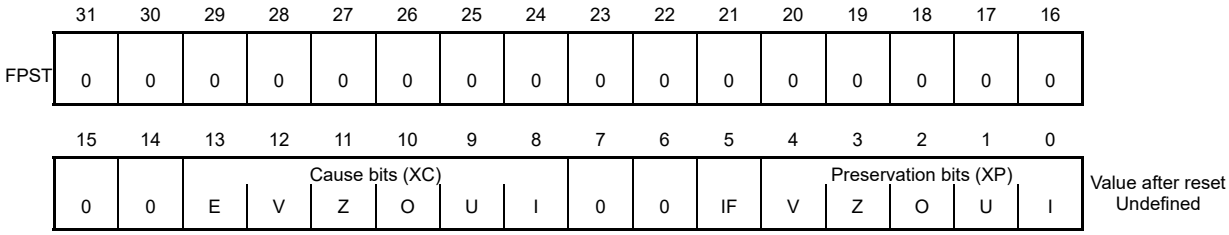


Table 3.52 FPST Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E, V, Z, O, U, I)	These are cause bits. For details, see **Section 3.2.3.4(1), Cause Bits (XC)**. Values written to these bits are reflected in FPSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see **Section 3.2.6.1 (8), Flushing Subnormal Numbers.** Value written to this bit is reflected in FPSR.IF bit.	R/W	0
4 to 0	XP (V, Z, O, U, I)	These are preservation bits. For details, see **Section 3.2.3.4(1), Preservation Bits (XP)**. Values written to these bits are reflected in FPSR.XP bits.	R/W	Undefined

(4) FPCC — Floating-point Operation Comparison Result

This register reflects the contents of the FPSR.CC7 to FPSR.CC0 bits.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
FPCC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	0	0	0	0	0	0	0	0	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0	Value after reset Undefined

Table 3.53 FPCC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7 to 0	CC[7:0]	These are CC (condition) bits. They store the result of a floating-point comparison instruction. The CC7 to CC0 bits are not affected by any instructions except the comparison instruction and LDSR instruction. Values written to these bits are reflected in the CC7 to CC0 bits of FPSR. 0: Comparison result is false 1: Comparison result is true	R/W	Undefined

(5) FPCFG — Floating-point Operation Configuration

This register reflects the contents of the FPSR register bits related to the operation settings.

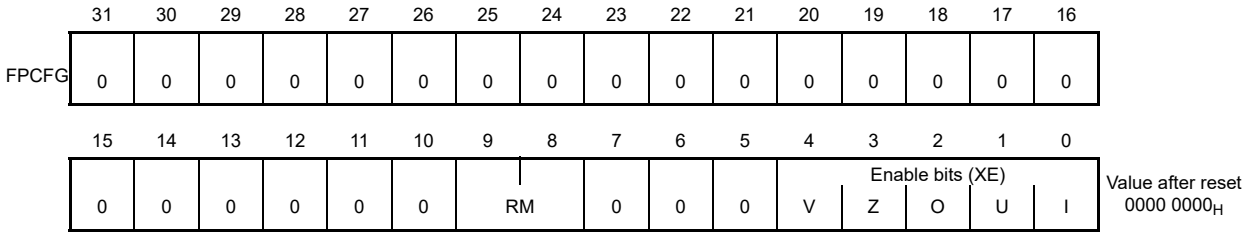


Table 3.54 FPCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
9, 8	RM	These are rounding mode control bits. The RM bits define the rounding mode that the FPU uses for all floating-point instructions. Values written to these bits are reflected in RM bits of FPSR.	R/W	0
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	XE (V,Z,O,U,I)	These are the enable bits. For details, see **Section 3.2.3.4(1), Enable Bits (XE)**. Values written to these bits are reflected in the FPSR.XE bits.	R/W	0

RM Bits 9	RM Bits 8	Mnemonic	Description
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two representable values, the result is rounded toward the value whose least significant bit is 0.
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.
1	0	RP	Rounds the result toward +∞. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.
1	1	RM	Rounds the result toward −∞. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.

3.2.3.5 FXU Function Registers

(a) Vector Registers

32 vector registers (wr0-wr31) are provided as data registers dedicated to the extended floating-point operation unit (FXU). All of these registers can be used to store data variables. The data width of these registers is 128 bits. The values of the vector registers wr0 to wr31 after reset are undefined.

As shown in **Figure 3.15** and **Table 3.55**, one vector register consists of 4 operation ways. One single-precision floating-point data is stored in one operation way. This allows four single-precision floating-point operations to be performed with one extended floating-point operation instruction.

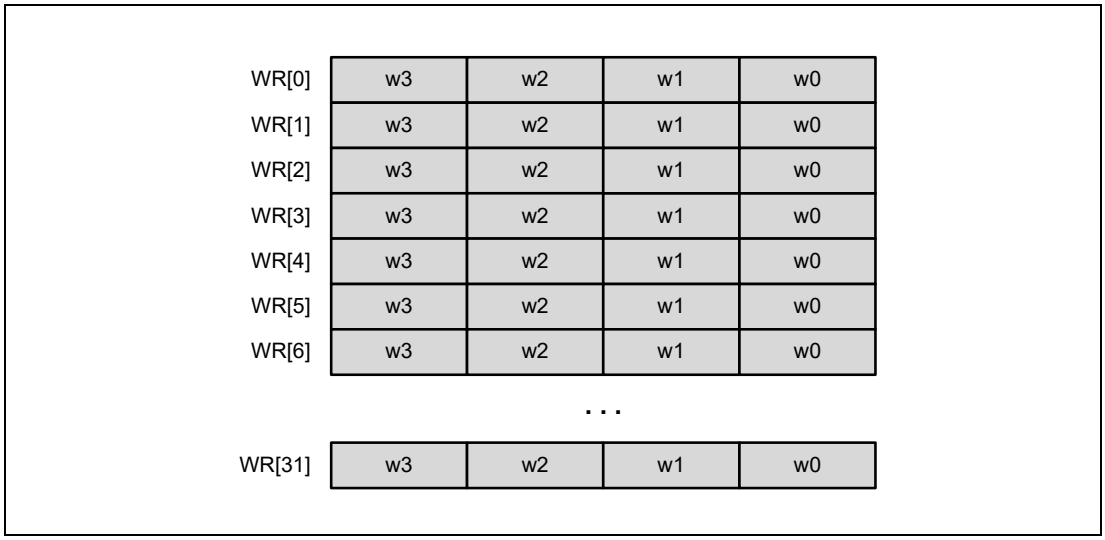


Figure 3.15 Vector Register Configuration

Table 3.55 Way Configuration of Vector Registers

Name	Bits	Function
W0	[31:0]	Way 0 data
W1	[63:32]	Way 1 data
W2	[95:64]	Way 2 data
W3	[127:96]	Way 3 data

(b) Extended Floating-point Function System Registers

The FXU can use the following system registers to control arithmetic operations. Extended floating-point function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

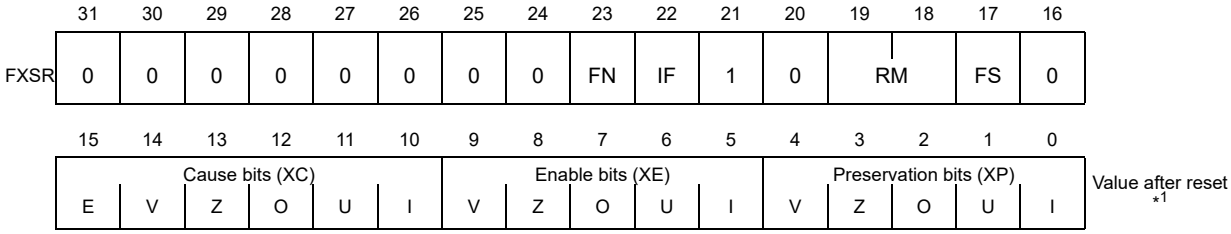
- FXSR: This register is used to control and monitor the extended floating-point operation exceptions. It is also used to set the operating mode of the FXU. It has bits for subnormal flush enabling, rounding mode control, exception cause, exception enabling and preservation.
- FXST: This register reflects the contents of the FXSR register bits related to the operation status.
- FXINFO: This register reflects the information about the configuration of the FXU.
- FXCFG: This register reflects the contents of the FXSR register bits related to the operation settings.
- FXXC: This register holds the XC (cause) bits of extended floating-point operation exceptions for each operation way.
- FXXP: This register holds the XP (preservation) bits of extended floating-point operation exceptions for each operation way.

Table 3.56 List of Extended Floating-point Function System Registers

Register Number (regID, selID)	Name	Function	Access Permission
SR6, 10	FXSR	Extended floating-point operation configuration/status	CU1 and SV
SR8, 10	FXST	Extended floating-point operation status	CU1
SR9, 10	FXINFO	FXU configuration information	CU1
SR10, 10	FXCFG	Extended floating-point operation configuration	CU1
SR12, 10	FXXC	XC (cause) bits for each operation way	CU1
SR13, 10	FXXP	XP (preservation) bits for each operation way	CU1

(1) FXSR — Extended Floating-point Operation Configuration/Status

The FXSR register indicates the execution status of extended floating-point operations and occurrence of exceptions.



Note 1. See the descriptions of each bit.

Table 3.57 FXSR Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
23	FN	This bit enables flush-to-nearest mode. When the FN bit is set to 1, if the rounding mode is RN and the operation result is a subnormal number, the number is flushed to the nearest number. For details, see **Section 3.2.6.1 (9), Flush to Nearest**.	R/W	0
22	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see **Section 3.2.6.2 Section 3.2.6.1 (8), Flushing Subnormal Numbers**.	R/W	0
21	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
20	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
19, 18	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FXU uses for all extended floating-point instructions.	R/W	00

RM Bits 19	RM Bits 18	Mnemonic	Description
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.
1	0	RP	Rounds the result toward +∞. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.
1	1	RM	Rounds the result toward −∞. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.

Table 3.57 FXSR Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
17	FS	This bit enables values that could not be normalized (subnormal numbers) to be flushed. If the FS bit is set, input operands and operation results that are subnormal numbers are flushed without causing an unimplemented operation exception (E). An input operand that is a subnormal number is flushed to 0 with the same sign. Operation results that are subnormal numbers either become 0 or the minimum normalized number, depending on the rounding mode. **Operation Result that is a Subnormal Number** / **Rounding Mode and Value after Flushing**: RN[*1] / RZ / RP / RM Positive: +0 / +0 / $+2^{Emin}$ / +0 Negative: −0 / −0 / −0 / -2^{Emin} Note 1. If the rounding mode is RN and the FXSR.FN bit is set to 1, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1 (9), Flush to Nearest**.	R/W	1
16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 10	XC (E,V,Z,O,U,I)	These are the cause bits. For details, see **Section 3.2.3.4(1), Cause Bits (XC)**.	R/W	Undefined
9 to 5	XE (V,Z,O,U,I)	These are the enable bits. For details, see **Section 3.2.3.4(1), Enable Bits (XE)**.	R/W	0
4 to 0	XP (V,Z,O,U,I)	These are the preservation bits. For details, see **Section 3.2.3.4(1), Preservation Bits (XP)**.	R/W	Undefined

Cause Bits (XC)

Bits 15 to 10 in the FXSR register are cause bits, which indicate the occurrence and cause of an extended floating-point operation exception. The FXSR.XC bits hold the OR of the detection results of extended floating-point operation exceptions that occurred in the operation ways. If an exception defined by IEEE754 is generated, when an enable bit is set to 1 corresponding to the exception, a cause bit is set, and the exception then occurs. When two or more exceptions occur during a single instruction, each corresponding bit is set to 1.

If two or more exceptions are detected, as long as the enable bit corresponding to one of the exceptions is set to 1, the exception occurs. In this case, the cause bits of all the detected exceptions, including exceptions whose enable bits are cleared to 0, are set to 1.

The cause bits are rewritten by the extended floating-point operation instruction that caused the extended floating-point operation exception. The E bit is set to 1 when software emulation is required, otherwise it is cleared to 0. Other bits are set to 1 or cleared to 0 depending on whether or not an IEEE754-defined exception has occurred.

When an extended floating-point operation exception has occurred, the operation result is not stored, and only the cause bits are affected.

When the cause bits are set to 1 by an LDSR instruction, no extended floating-point operation exception occurs.

Enable Bits (XE)

Bits 9 to 5 in the FXSR register are the enable bits, which enable extended floating-point operation exceptions. When an IEEE754-defined exception occurs, an extended floating-point operation exception occurs if the enable bit corresponding to the exception has been set to 1.

There are no enable bits corresponding to an unimplemented operation exception (E). An unimplemented operation exception (E) always occurs as an extended floating-point operation exception.

The FXSR.XE bits cannot be set on an operation way basis. The same setting is used for the all operation ways.

If the corresponding enable bit has not been set to 1, no exception occurs and the default result defined by IEEE754 is stored.

Preservation Bits (XP)

Bits 4 to 0 in the FXSR register are preservation bits and accumulate and indicate the exceptions that have been detected since a reset. An exception defined by IEEE754 occurs, and if an extended floating-point operation exception is not generated, the preservation bit is set to 1, otherwise it does not change. The preservation bits are not cleared to 0 by the extended floating-point operation. However, these bits can be set and cleared by software when an LDSR instruction is used to write a new value to the FXSR register.

There are no preservation bits corresponding to unimplemented operation exceptions (E). An unimplemented operation exception (E) always occurs as an extended floating-point operation exception.

NOTE

For details about the exception types and how they relate to particular bits, see **Figure 3.38, Relationship among the Cause and Enable Bits of the FXXC and FXSR Registers** and **Figure 3.39, Relationship among the Preservation and Enable Bits of the FXXP and FXSR Registers**.

(2) FXST — Extended Floating-point Operation Status

This register reflects the contents of the FXSR register bits related to the operation status.

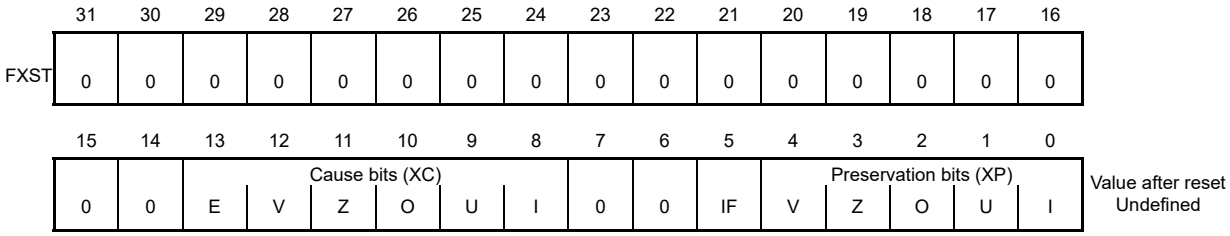


Table 3.58 FXST Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC (E,V,Z,O,U,I)	These are cause bits. For details, see **Section 3.2.3.4(1), Cause Bits (XC).** Values written to these bits are reflected in FXSR.XC bits.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	IF	This bit accumulates and indicates information about the flushing of input operands. For details about flushing subnormal numbers, see **Section 3.2.6.1 (8), Flushing Subnormal Numbers**. Value written to this bit is reflected in FXSR.IF bit.	R/W	0
4 to 0	XP (V,Z,O,U,I)	These are preservation bits. For details, see **Section 3.2.3.4(1), Preservation Bits (XP)**. Values written to these bits are reflected in FXSR.XP bits.	R/W	Undefined

(3) FXINFO — FXU Configuration Information

This register reflects the information about the configuration of the FXU.

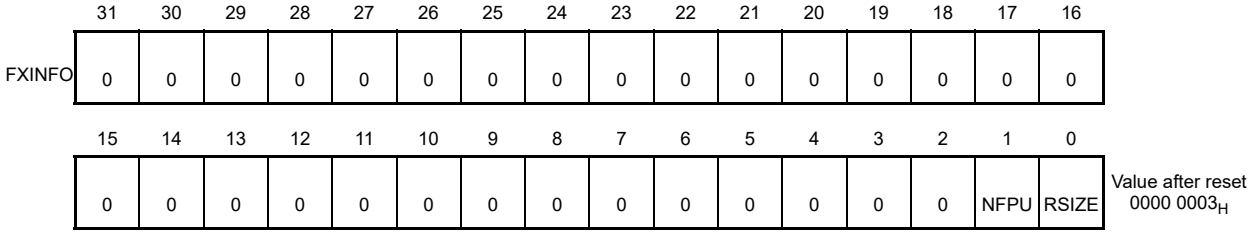


Table 3.59 FXINFO Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	NFPU	This bit indicates that 4-way SIMD arithmetic unit is provided.	R	1
0	RSIZE	This bit indicates that 32 128-bit vector registers are provided.	R	1

(4) FXCFG — Extended Floating-point Operation Configuration

This register reflects the contents of the FXSR register bits related to the operation settings.

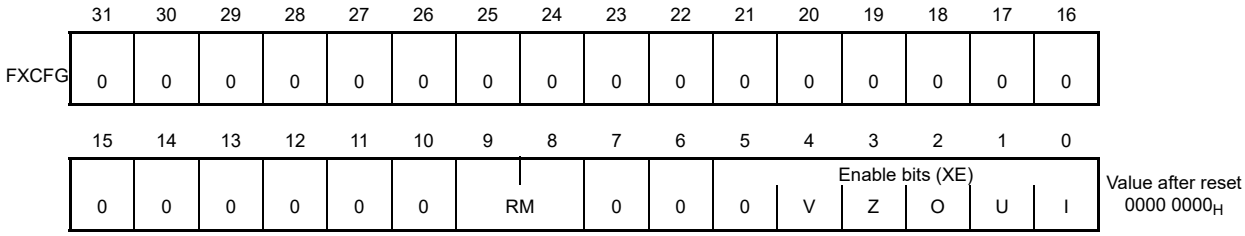


Table 3.60 FXCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 10	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
9, 8	RM	These are the rounding mode control bits. The RM bits define the rounding mode that the FXU uses for all extended floating-point instructions. Values written to these bits are reflected in FXSR.RM bits.	R/W	0
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	XE (V,Z,O,U,I)	These are the enable bits. For details, see **Section 3.2.3.4(1), Enable Bits (XE)**. Values written to these bits are reflected in FXSR.XE bits.	R/W	0

RM Bits		Mnemonic	Description
9	8		
0	0	RN	Rounds the result to the nearest representable value. If the value is exactly in-between the two nearest representable values, the result is rounded toward the value whose least significant bit is 0.
0	1	RZ	Rounds the result toward 0. The result is the nearest to the value that does not exceed the absolute value of the result with infinite accuracy.
1	0	RP	Rounds the result toward +∞. The result is nearest to a value equal to or greater than the accurate result with infinite accuracy.
1	1	RM	Rounds the result toward −∞. The result is nearest to a value equal to or less than the accurate result with infinite accuracy.

(5) FXXC — XC (Cause) Bits for Each operation way

This register holds the XC (cause) bits of extended floating-point operation exceptions for each operation way. Any attempts to write this register will not affect the value of the XC bits of the FXSR register.

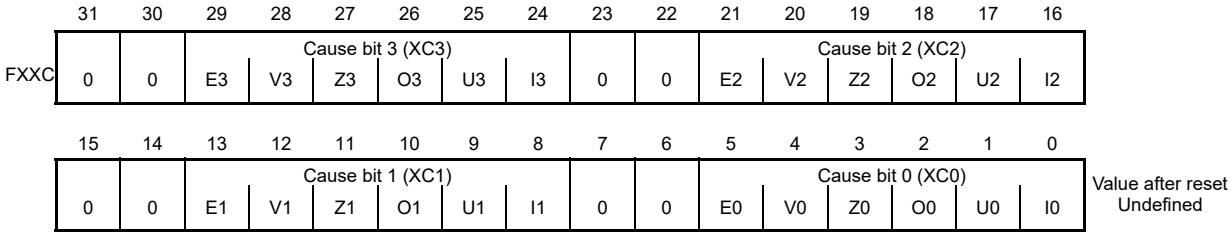


Table 3.61 FXXC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31, 30	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
29 to 24	XC3 (E3,V3,Z3,O3,U3,I3)	These bits hold the cause bits associated with operation way 3.	R/W	Undefined
23, 22	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
21 to 16	XC2 (E2,V2,Z2,O2,U2,I2)	These bits hold the cause bits associated with operation way 2.	R/W	Undefined
15, 14	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
13 to 8	XC1 (E1,V1,Z1,O1,U1,I1)	These bits hold the cause bits associated with operation way 1.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5 to 0	XC0 (E0,V0,Z0,O0,U0,I0)	These bits hold the cause bits associated with operation way 0.	R/W	Undefined

(6) FXXP — XP (Preservation) Bits for Each operation way

This register holds the XP (preservation) bits of extended floating-point operation exceptions for each operation way. Any attempts to write this register will not affect the value of the XP bits of the FXSR register.

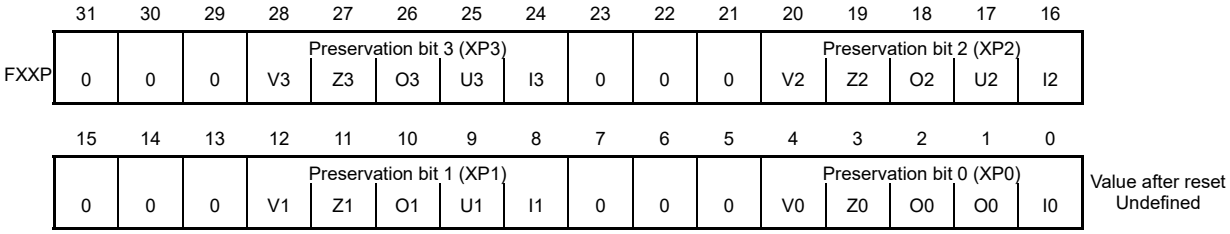


Table 3.62 FXXP Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 29	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
28 to 24	XP3 (V3,Z3,O3,U3,I3)	These bits hold the preservation bits associated with operation way 3.	R/W	Undefined
23 to 21	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
20 to 16	XP2 (V2,Z2,O2,U2,I2)	These bits hold the preservation bits associated with operation way 2.	R/W	Undefined
15 to 13	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
12 to 8	XP1 (V1,Z1,O1,U1,I1)	These bits hold the preservation bits associated with operation way 1.	R/W	Undefined
7 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	XP0 (V0,Z0,O0,U0,I0)	These bits hold the preservation bits associated with operation way 0.	R/W	Undefined

3.2.3.6 MPU Function Registers

(a) MPU Function System Registers

MPU function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.63 MPU Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	SV
SR2, 5	MPCFG	MPU configuration	SV
SR8, 5	MCA	Memory protection setting check address	SV
SR9, 5	MCS	Memory protection setting check size	SV
SR10, 5	MCC	Memory protection setting check command	SV
SR11, 5	MCR	Memory protection setting check result	SV
SR12, 5	MCI	Memory protection setting check SPID	SV
SR16, 5	MPIDX	Index of memory protection setting registers to be accessed	SV
SR17, 5	MPBK	MPU Bank Setting	SV
SR20, 5	MPLA	Protection area minimum address	SV
SR21, 5	MPUA	Protection area maximum address	SV
SR22, 5	MPAT	Protection area attribute	SV
SR24, 5	MPID0	SPID which can access protection area	SV
SR25, 5	MPID1	SPID which can access protection area	SV
SR26, 5	MPID2	SPID which can access protection area	SV
SR27, 5	MPID3	SPID which can access protection area	SV
SR28, 5	MPID4	SPID which can access protection area	SV
SR29, 5	MPID5	SPID which can access protection area	SV
SR30, 5	MPID6	SPID which can access protection area	SV
SR31, 5	MPID7	SPID which can access protection area	SV

(1) MPM — Memory Protection Operation Mode

The memory protection operation mode register is used to define the basic operation mode of the memory protection function.

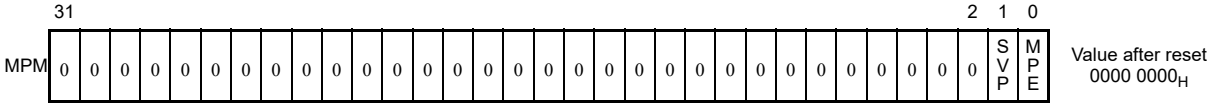


Table 3.64 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	SVP	In SV mode (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*1 0: Enable all accesses in SV mode. 1: Restrict access according to the SX, SW, and SR bits in SV mode.*2	R/W	0
0	MPE	This bit is used to specify whether to enable or disable MPU function. 0: Disable 1: Enable	R/W	0

Note 1. When the SVP bit is set to 1 by an LDSR instruction, the setting is effective from the operand access by the subsequent instruction. On the other hand, since the instruction fetch access is executed independently from the setting of the SVP bit, it is not possible to identify the timing when the new setting is applied to the instruction fetch access. In this case, make settings of the protection area in advance so that the program which sets the SVP bit itself will not be blocked.

Note 2. If the SVP bit is set to 1, the MDP or MIP exception handler cannot be executed if it is not permitted to access to the necessary memory areas. Be sure to make settings of the protection area in advance so that the necessary memory areas can be accessed by the exception handler.

(2) MPCFG — MPU Configuration

This register holds the information about the configuration of the MPU.

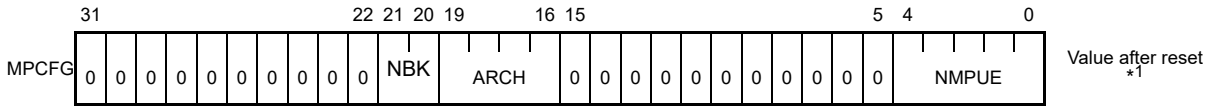


Table 3.65 MPCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 22	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
21, 20	NBK	Indicates the "number of banks - 1" of the MPU bank *2 equipped in this CPU. Since the MPU bank of this CPU is equipped with one bank, 0 is read.	R	*1
19 to 16	ARCH	These bits indicate the version of the MPU architecture specifications. A value of 2 is read for this CPU.	R	2
15 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	NMPUE	These bits indicate the number of MPU entries implemented in this CPU minus 1. A value of 31 is read since this CPU incorporates 32 MPU entries.	R	31

Note 1. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

Note 2. The number of MPU bank mounted on the MPU is equal or more than one. The MPU bank consists of MPU entries (MPAT, MPUA, MPLA) whose number is specified by NMPUE. If two banks are equipped, the number of MPU entries twice as large as the value indicated by the NMPUE are equipped. However, each MPU bank is used exclusively. Only the MPU bank indicated by the MPBK register is used for memory protection. Therefore, even if multiple banks are equipped, the maximum number of the MPU entries that can be used simultaneously for memory protection is up to the value indicated by the NMPUE.

(3) MCA — Memory Protection Setting Check Address

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

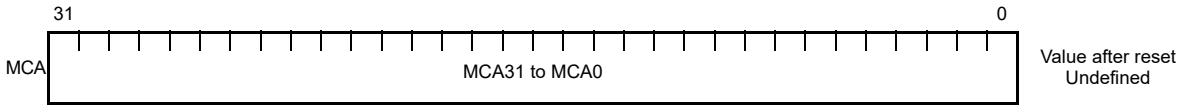


Table 3.66 MCA Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCA31 to MCA0	These bits are used to specify the starting address of the memory area which subjects to a memory protection setting check in bytes.	R/W	Undefined

(4) MCS — Memory Protection Setting Check Size

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

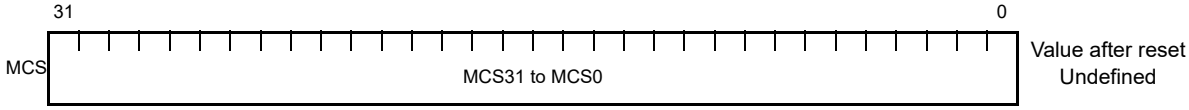


Table 3.67 MCS Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCS31 to MCS0	These bits are used to specify the size of the memory area in bytes which subjects to a memory protection setting check. Because the specified size is assumed to represent an unsigned integer, it is not possible to check an area in the direction in which the address value decreases relative to the MCA register value.	R/W	Undefined

(5) MCC — Memory Protection Setting Check Command

This command register is used to start a memory protection setting check.

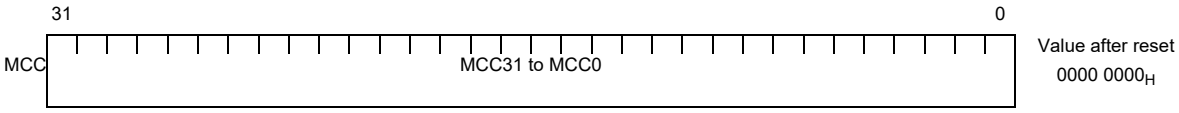


Table 3.68 MCC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCC31 to MCC0	When any value is written to the MCC register, a memory protection setting check starts. By setting up the MCA / MCS register and then writing to the MCC register, results are stored in MCR. Because the check is started by any written value, a check can be started by using r0 as the source register without using any unnecessary registers. Note that, for the check, the results are applied according to each area setting regardless of the state of the PSW.UM bit. This checking applies for memory protection setting of a MPU bank that is specified by MPBK register. When the MCC register is read, value 0000 0000_H is always returned.	R/W	0

(6) MCR — Memory Protection Setting Check Result

This register is used to store the results of a memory protection setting check.

When the MCC register is written, the value of the MPIDn registers (n = 0 to 7), the WMPIDn bits (n = 0 to 7) (for write permission) and the RMPIDn bits (n = 0 to 7) (for execution and read permission) of each MPAT register are checked to verify that the SPID specified by the MCI register matches the SPID for which the access is permitted.

However, if the WG bit or RG bit of the MPAT register is set to 1, it is assumed that the SPID specified by the MCI register matches the SPID for which write or execution and read access is permitted regardless of the SPID matching result above.

If an SPID match is found, protection attributes of the protection area which includes the address area specified by the MCA and MCS registers are stored in the corresponding attribute bits of the MCR register. If multiple protection areas include the specified address area, and if an access is permitted by any of the protection areas, 1, which indicates the access is permitted, is stored in the corresponding attribute bits of the MCR register.

If no SPID match is found or no protection area including the specified address area exists, 0, which indicates the access is not permitted, is stored in each attribute bit of the MCR register.

CAUTION

If the specified area to be checked crosses 0000 0000_H or 7FFF $FFFF_H$, it is judged as an area setting error, and the MCR.OV bit is set to 1. This means that the MCR.OV bit must be read to confirm that the result is not invalid (OV = 0) before referencing the check result of each attribute bit. When the MCR.OV bit is set to 1, all attribute bits are cleared to 0 unless the MPM.SVP bit is 0. If the MPM.SVP bit is 0, the MCR.SXE, SWE and SRE bits are set to 1, even if the MCR.OV bit is set to 1. Even when no SPID match is found, the MCR.OV bit is set to 1 if the area setting is wrong.

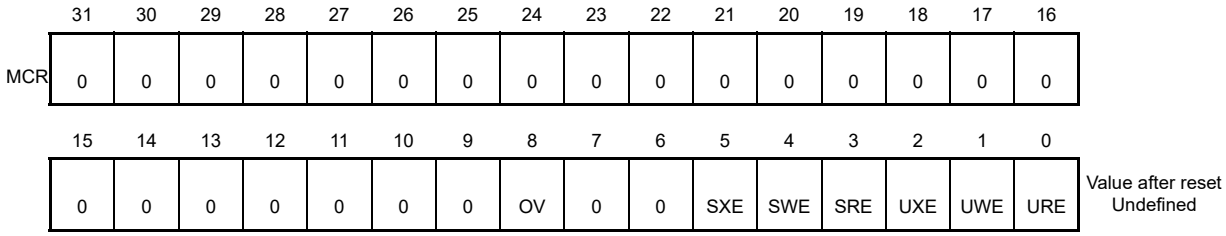


Table 3.69 MCR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	OV	If the specified area includes 0000 0000_H or 7FFF $FFFF_H$, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SXE	If the specified area is contained within one protection area and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
4	SWE	If the specified area is contained within one protection area and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
3	SRE	If the specified area is contained within one protection area and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
2	UXE	If the specified area is contained within one protection area and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
1	UWE	If the specified area is contained within one protection area and writing from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
0	URE	If the specified area is contained within one protection area and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

(7) MCI — Memory Protection Setting Check SPID

This register is used to specify the SPID for which a memory protection settings check is to be performed.

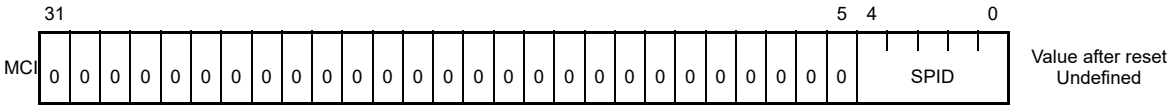


Table 3.70 MCI Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits specify the SPID for which a memory protection settings check is to be performed.	R/W	Undefined

(8) MPIDX — Index of Memory Protection Setting Registers to be Accessed

This register is used to specify the index of memory protection setting registers to be accessed.

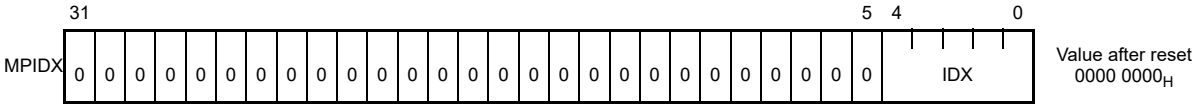


Table 3.71 MPIDX Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	IDX	These bits specify the MPU entries index of the MPLA, MPUA, and MPAT registers to be accessed by the LDSR or STSR instruction. The result of changing the value of this register is reflected in the instruction immediately following. Values from 0 to the maximum entry number indicated in the MPCFG.NMPUE can be specified for the IDX. If you specify a value exceeding the maximum entry number for the IDX, and the MPLA, MPUA, or MPAT register is accessed by the LDSR or STSR instruction, it is handled as an undefined register.	R/W	0

(9) MPBK — MPU Bank Setting

The MPBK register selects the MPU bank.

This register is not updated if the SVLOCK.SVL bit is set to 1.

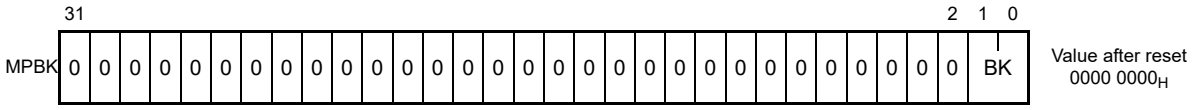


Table 3.72 MPBK Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1, 0	BK	These bits specify the MPU bank to be used for memory protection. 0: MPU bank 0 Other than above: Setting prohibited The MPU bank specified in the BK field is subject to memory protection, memory protection setting check function, the LDM.MP instruction, the STM.MP instruction. Each register of the MPU bank not specified in the BK field has no effect on the MPU function. And it is impossible to operate such registers with the LDSR and STSR instructions. It is impossible to set the value exceeding the MPCFG.NBK in the BK field. If an attempt is made to write such a value, the value is ignored and the original value is kept.	R/W	0

(10) MPLA — Protection Area Minimum Address

This register indicates the minimum address of a protection area. The value written into this register with the LDSR instruction is set as the minimum address of area n (n = 0 to MPCFG.NMPUE) specified in the MPIDX register. The value read from this register with the STSR instruction is the minimum address of area n specified in the MPIDX register. It is impossible to manipulate the minimum address of area n without using the MPIDX register and this register.

If the value more than MPCFG.NMPUE is specified in the MPIDX register, the register is handled as an undefined register.

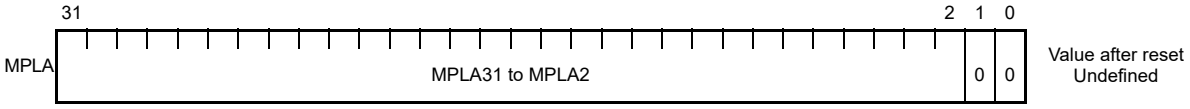


Table 3.73 MPLA Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	MPLA31 to MPLA2	These bits specify the minimum address of area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register. The specified minimum address is included in the range of area matching.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(11) MPUA — Protection Area Maximum Address

This register indicates the maximum address of a protection area. The value written into this register with the LDSR instruction is set as the maximum address of area n (n = 0 to MPCFG.NMPUE) specified in the MPIDX register. The value read from this register with the STSR instruction is the maximum address of area n specified in the MPIDX register. It is impossible to manipulate the maximum address of area n without using the MPIDX register and this register.

If the value more than MPCFG.NMPUE is specified in the MPIDX register, the register is handled as an undefined register.

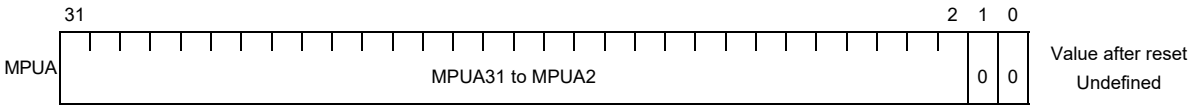


Table 3.74 MPUA Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	MPUA31 to MPUA2	These bits specify the maximum address of area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register. The specified maximum address is included in the range of area matching. Bits 1 and 0 of the maximum address are handled as 1.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(12) MPAT — Protection Area Attribute

This register indicates the attribute of a protection area. The value written into this register with the LDSR instruction is set as the protection attribute of area n (n = 0 to MPCFG.NMPUE) specified in the MPIDX register. The value read from this register with the STSR instruction is the protection attribute of area n specified by the MPIDX register. It is impossible to manipulate the attribute of area n without using the MPIDX register and this register.

If the value more than MPCFG.NMPUE is specified in the MPIDX register, the register is handled as an undefined register.

For details on how to set these bits, see **Section 3.2.5.1 (2), Protection Area Settings**.

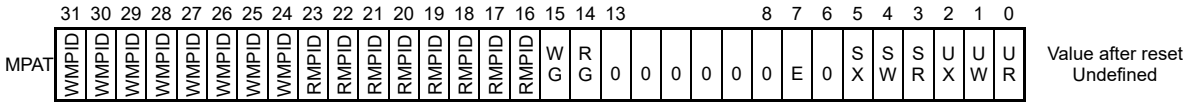


Table 3.75 MPAT Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 24	WMPID7 to 0	This bit indicates the write permission for the SPID specified in the MPIDn (n = 0 to 7) register. 0: Writing by the SPID specified by the MPIDn register is disabled. 1: Writing by the SPID specified by the MPIDn register is enabled.	R/W	Undefined
23 to 16	RMPID7 to 0	This bit indicates the execution and read permissions for the SPID specified in the MPIDn (n = 0 to 7) register. 0: Execution and reading by the SPID specified by the MPIDn register is disabled. 1: Execution and reading by the SPID specified by the MPIDn register is enabled.	R/W	Undefined
15	WG	This bit indicates the write permission for any SPID. 0: Writing is enabled by the settings of the MPIDn register and WMPIDn bit (n = 0 to 7). 1: Writing by any SPID is enabled regardless of the settings of the MPIDn register and WMPIDn bit (n = 0 to 7).	R/W	Undefined
14	RG	This bit indicates the execution and read permissions for any SPID. 0: Execution and reading is enabled by the settings of the MPIDn register and RMPIDn bit (n = 0 to 7). 1: Execution and reading by any SPID is enabled regardless of the settings of the MPIDn register and RMPIDn bit (n = 0 to 7).	R/W	Undefined
13 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	E	This bit indicates whether area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled.	R/W	0
6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5	SX	This bit indicates the execution permission in the supervisor mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register.*1 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
4	SW	This bit indicates the write permission in the supervisor mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register.*1 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
3	SR	This bit indicates the read permission in the supervisor mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register.*1, *2 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Table 3.75 MPAT Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
2	UX	This bit indicates the execution permission in the user mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register. 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
1	UW	This bit indicates the write permission in the user mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register. 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
0	UR	This bit indicates the read permission in the user mode for area n (n = 0 to MPCFG.NMPUE) specified by the MPIDX register.*2 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Note 1. If access is restricted in SV mode, the MDP or MIP exception handler might not be executed depending on the settings. Be sure to make settings of the protection area in advance so that the necessary memory areas can be accessed by the exception handler.

Note 2. For the CALLT, SWITCH, and SYSCALL instructions and interrupts in table reference method, an MDP exception occurs if reading the address where the table is stored is not permitted. Permission for the SYSCAL instruction and interrupts in table reference method are judged after the operation mode of the CPU is transitioned to the supervisor mode. Therefore, when the MPM.SVP bit is set (1), the SR bit need to be set (1) as well.

(13) MPIDn —SPID which can Access Protection Area

This register specifies the SPID which can access protection area. The accessibility for the specified SPID is determined in conjunction with the WMPIDn and RMPIDn bits (n = 0 to 7) which are specified via the MPAT register.

This CPU has 8 MPID registers.

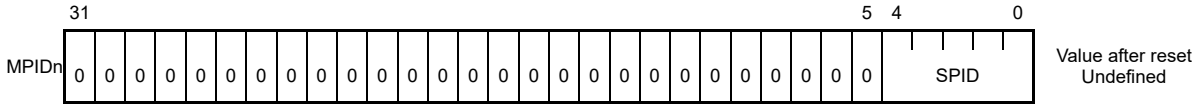


Table 3.76 MPIDn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 5	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
4 to 0	SPID	These bits specify the SPID which can access protection area.	R/W	Undefined

3.2.3.7 Cache Operation Function Registers

(a) Cache Control Function System Registers

Cache control function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.77 Cache Control Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR16, 4	ICTAGL	Instruction cache tag Lo access	SV
SR17, 4	ICTAGH	Instruction cache tag Hi access	SV
SR18, 4	ICDATL	Instruction cache data Lo access	SV
SR19, 4	ICDATH	Instruction cache data Hi access	SV
SR24, 4	ICCTRL	Instruction cache control	SV
SR26, 4	ICCFG	Instruction cache configuration	SV
SR28, 4	ICERR	Instruction cache error	SV

(1) ICTAGL — Instruction Cache Tag Lo Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the tag RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the tag RAM for the instruction cache are stored.

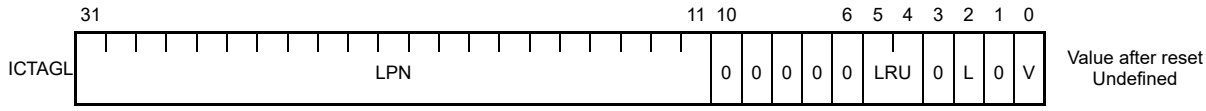


Table 3.78 ICTAGL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 11	LPN	These bits hold address bits 27 to 11. Bits 31 to 28 must always be set to 0. Bits 27 to 11 are held if the cache size is 8 Kbytes. Bits 27 to 12 are held and bit 11 is always set to 0 if the cache size is 16 Kbytes.*1 Bits 27 to 13 are held and bit 12 and 11 are always set to 0 if the cache size is 32 Kbytes.*2	R/W	Undefined
10 to 6	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
5, 4	LRU	These bits indicate LRU information of specified cache line. LRU information cannot be freely changed to any value by the CIST instruction.	R/W	Undefined
3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	L	This bit holds the lock information.*3 0: The cache line is not locked. 1: The cache line is locked.	R/W	Undefined
1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	V	This bit holds valid/invalid information of specified cache line.*3 0: The cache line is disabled. 1: The cache line is enabled.	R/W	Undefined

Note 1. Bit 11 cannot be written with any value by an LDSR instruction if the cache size is 16 Kbytes. It is always set to 0.

Note 2. Bit 12 and 11 can not be changed by an LDSR instruction if the cache size is 32Kbytes. It is always set to 0.

Note 3. If a tag RAM is set by a CIST instruction, the settings of other bits are valid only when the V bit is enabled. When the V bit is disabled, the cache line is always judged as a cache miss regardless of the value in the LPN bit. Also, when the V bit is disabled, even if the cache line is locked by the setting of the L bit, the cache line becomes the target to be replaced.

(2) ICTAGH — Instruction Cache Tag Hi Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the ECC for the tag RAM and the Data RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the ECC for the tag RAM and the Data RAM for the instruction cache are stored.

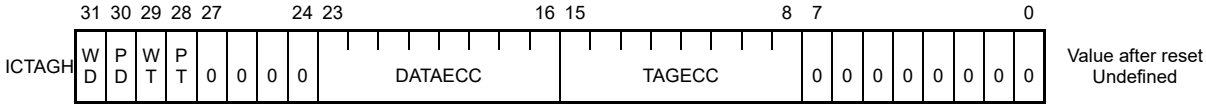


Table 3.79 ICTAGH Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	WD	This bit specifies whether the data RAM of the cache is updated during execution of the CIST instruction. 0: Data RAM of the cache is not updated. 1: Data RAM of the cache is updated.	R/W	0
30	PD	This bit specifies the data to be written to the ECC of the data RAM when the WD bit is set to 1 and the CIST instruction is executed. 0: ECC automatically generated from the write data is written to the ECC of the data RAM. 1: Values in the DATAECC field are written to the ECC of the data RAM.	R/W	0
29	WT	This bit specifies whether the tag RAM of the cache is updated during execution of the CIST instruction. When this bit is set to 1, the V bit and L bit of the cache line are also updated. 0: Tag RAM of the cache is not updated. 1: Tag RAM of the cache is updated.	R/W	0
28	PT	This bit specifies the data to be written to the ECC of the tag RAM when the WT bit is set to 1 and the CIST instruction is executed. 0: ECC automatically generated from the write data is written to the ECC of the tag RAM. 1: Values in the TAGECC field are written to the ECC of the tag RAM.	R/W	0
27 to 24	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
23 to 16	DATAECC	These bits hold ECC for data RAM.	R/W	Undefined
15 to 8	TAGECC	These bits hold ECC for tag RAM. Bit 15 is fixed to 0.	R/W	Undefined
7 to 0	—	(Reserved for future expansion. Be sure to set to 0.)	R	0

(3) ICDATL — Instruction Cache Data Lo Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the data RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the data RAM for the instruction cache are stored.

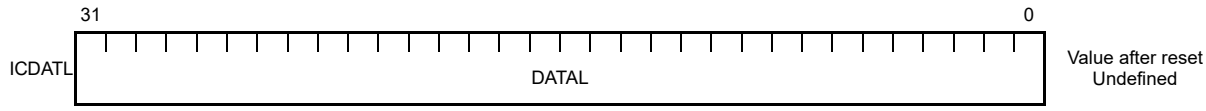


Table 3.80 ICDATL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	DATAL	These bits hold the lower-order 32 bits of the 64-bit instruction data*1 in the block in the specified cache line. The bits to be held is specified by offset of the index.*2 Offset of index = 00: Bits 31 to 0*3 Offset of index = 01: Bits 95 to 64*3 Offset of index = 10: Bits 159 to 128*3 Offset of index = 11: Bits 223 to 192*3	R/W	Undefined

Note 1. Data alignment of the instruction data in the block in the cache line is different from the data alignment of memory. For details, see **Section 3.2.5.2 (10), Configuration of Instruction Cache**.

Note 2. For details on how to specify Offset, see **Section 3.2.5.2 (6), Cache Index Specification Method**.

Note 3. For details on the bit position, see **Figure 3.31** in **Section 3.2.5.2 (10), Configuration of Instruction Cache**.

(4) ICDATH — Instruction Cache Data Hi Access

This register is used by the CIST/CILD instruction in relation to the instruction cache. During execution of the CIST instruction, values that are stored to the data RAM for the instruction cache are stored. During execution of the CILD instruction, values read from the data RAM for the instruction cache are stored.

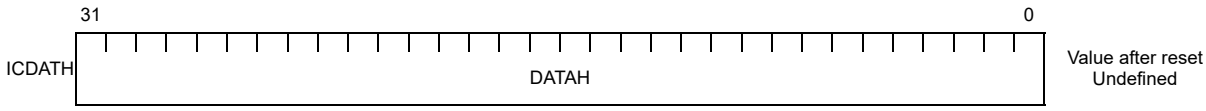


Table 3.81 ICDATH Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	DATAH	These bits hold the upper-order 32 bits of the 64-bit instruction data*1 in the block in the specified cache line. The bits to be held is specified by offset of the index.*2 Offset of index = 00: Bits 63 to 32*3 Offset of index = 01: Bits 127 to 96*3 Offset of index = 10: Bits 191 to 160*3 Offset of index = 11: Bits 255 to 224*3	R/W	Undefined

Note 1. Data alignment of the instruction data in the block in the cache line is different from the data alignment of memory. For details, see **Section 3.2.5.2 (10), Configuration of Instruction Cache**.

Note 2. For details on how to specify Offset, see **Section 3.2.5.2 (6), Cache Index Specification Method**.

Note 3. For details on the bit position, see **Figure 3.31** in **Section 3.2.5.2 (10), Configuration of Instruction Cache**.

(5) ICCTRL — Instruction Cache Control

This register is used to control the instruction cache.

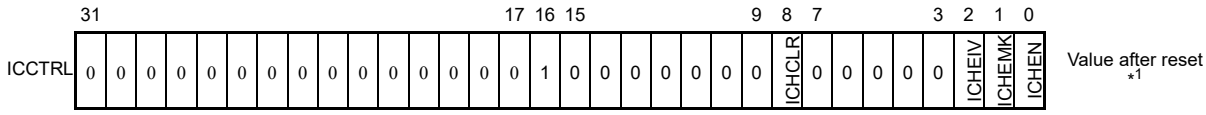


Table 3.82 ICCTRL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 17	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
16	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
15 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	ICHCLR	When this bit is set to 1, the entire instruction cache is cleared. This clears the V and L bits (to 0) and initializes the LRU information. This bit is always read as 0.	R/W	0
7 to 3	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
2	ICHEIV	When this bit is set to 1, the instruction cache is automatically set as invalid (the ICHEN bit is cleared to 0) whenever a cache error*2 occurs.	R/W	0
1	ICHEMK	When this bit is set to 1, it masks notification of cache error exceptions for the CPU after a cache error*2 has occurred.	R/W	1
0	ICHEN	This bit indicates valid/invalid status of instruction cache. 0: Instruction cache is invalid 1: Instruction cache is valid	R/W	*1

Note 1. The value after reset is determined by the hardware specifications. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

Note 2. For details on the cache error, see **Section 3.2.5.2 (10), Configuration of Instruction Cache**.

(6) ICCFG — Instruction Cache Configuration

This register indicates the instruction cache configuration.

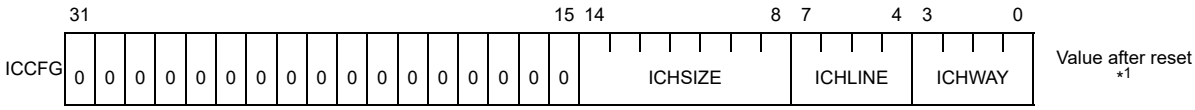


Table 3.83 ICCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 15	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
14 to 8	ICHSIZE	These bits indicate the size (in Kbytes) of the instruction cache. 000 1000: 8 Kbytes 001 0000: 16 Kbytes 010 0000: 32 Kbytes Other than above: Setting prohibited	R	*1
7 to 4	ICHLINE	These bits indicate the number of lines for each way in the instruction cache. 0010: 64 lines 0100: 128 lines 1000: 256 lines Other than above: Setting prohibited	R	*1
3 to 0	ICHWAY	These bits indicate the number of ways in the instruction cache. 0100: 4 ways Other than above: Setting prohibited	R	*1

Note 1. The value after reset is determined by the hardware specifications. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

(7) ICERR — Instruction Cache Error

This register is used to store cache error information for the instruction cache.

After the ICHERR bit is set to 1, any subsequent cache error information that is generated is not stored until this setting is explicitly cleared to 0. However, information on error status (ESAFE, ESMH, ESPBSE, ESTE1, ESTE2, ESDC, and ESDE) is accumulated.

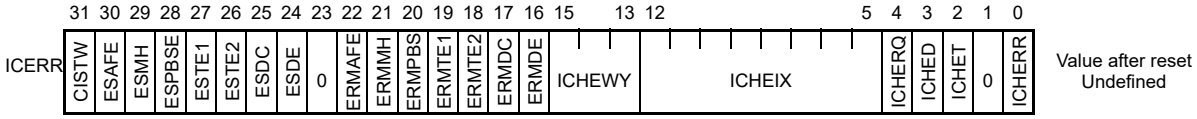


Table 3.84 ICERR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	CISTW	This bit is set to indicate that the destination way specified for a CISTI instruction was in error. Although the entry information is overwritten so that writing is completed, the V bit will be cleared the next time the cache line is read (i.e. reading will be judged to have missed the cache). However, setting of this bit is not accompanied by an exception for the CPU.	R/W	Undefined
30	ESAFE	Error status: Address feedback error	R/W	Undefined
29	ESMH	Error status: Multi hit	R/W	Undefined
28	ESPBSE	Error status: WAY error	R/W	Undefined
27	ESTE1	Error status: Tag RAM 1-bit error	R/W	Undefined
26	ESTE2	Error status: Tag RAM 2-bits error	R/W	Undefined
25	ESDC	Error status: Data RAM 1-bit error	R/W	Undefined
24	ESDE	Error status: Data RAM 2-bits error	R/W	Undefined
23	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
22	ERMAFE	Error exception notification mask: Address feedback error*3	R/W	0
21	ERMMH	Error exception notification mask: Multi hit*3	R/W	0
20	ERMPBSE	Error exception notification mask: WAY error*3	R/W	0
19	ERMTE1	Error exception notification mask: Tag RAM 1-bit error*3	R/W	0
18	ERMTE2	Error exception notification mask: Tag RAM 2-bits error*3	R/W	0
17	ERMDC	Error exception notification mask: Data RAM 1-bit error*3	R/W	0
16	ERMDE	Error exception notification mask: Data RAM 2-bits error*3	R/W	0
15 to 13	ICHEWY	These bits hold the way number where a cache error occurred. Bit 15 is always set to 0.	R/W	Undefined
12 to 5	ICHEIX	These bits hold the cache index where a cache error occurred. Bits 10 to 5 are held and bits 12 and 11 are fixed to 0 for a cache size of 8 Kbytes.*1 Bits 11 to 5 are held and bit 12 is fixed to 0 for a cache size of 16 Kbytes.*2 Bits 12 to 5 are held for cache size of 32 Kbytes.	R/W	Undefined
4	ICHERQ	When this bit is set to 1, this bit indicates that cache error exception notification is in progress. However, if cache error exception notification has been masked, the CPU is not notified even when 1 has been set to this bit.	R/W	0
3	ICHED	This bit indicates that an error has occurred in data RAM.	R/W	0
2	ICHET	This bit indicates that an error has occurred in tag RAM.	R/W	0
1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	ICHERR	This bit is set to 1 when a cache error has occurred.	R/W	0

Note 1. Bits 12 and 11 cannot be written with any value by an LDSR instruction if the cache size is 8 Kbytes. They are always set to 0.

Note 2. Bit 12 cannot be written with any value by an LDSR instruction if the cache size is 16 Kbytes. It is always set to 0.

Note 3. While the error exception notification mask bit for each error cause is cleared (to 0), if the corresponding error occurs, the ICHERR, ICHERQ, ICHED/ICHET, ICHEIX, and ICHEWY bits are updated and the corresponding error status bit is set (to 1). While the error exception notification mask bit is set (to 1), if the corresponding error occurs, only the corresponding error status bit is set (to 1).

3.2.3.8 Count Function Registers

(a) Count Function System Registers

Count function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.85 Count Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission	
			Read	Write
SR0, 11	TSCOUNTL	Timestamp count L register	UM	SV
SR1, 11	TSCOUNTH	Timestamp count H register	UM	SV
SR2, 11	TSCTRL	Timestamp count control register	SV	
SR8, 11	PMUMCTRL	Performance counter User mode control register	SV	
SR0, 14	PMCTRL0	Performance count control 0 register	SV*1	
SR1, 14	PMCTRL1	Performance count control 1 register	SV*1	
SR2, 14	PMCTRL2	Performance count control 2 register	SV*1	
SR3, 14	PMCTRL3	Performance count control 3 register	SV*1	
SR4, 14	PMCTRL4	Performance count control 4 register	SV*1	
SR5, 14	PMCTRL5	Performance count control 5 register	SV*1	
SR6, 14	PMCTRL6	Performance count control 6 register	SV*1	
SR7, 14	PMCTRL7	Performance count control 7 register	SV*1	
SR16, 14	PMCOUNT0	Performance count 0 register	SV*1	
SR17, 14	PMCOUNT1	Performance count 1 register	SV*1	
SR18, 14	PMCOUNT2	Performance count 2 register	SV*1	
SR19, 14	PMCOUNT3	Performance count 3 register	SV*1	
SR20, 14	PMCOUNT4	Performance count 4 register	SV*1	
SR21, 14	PMCOUNT5	Performance count 5 register	SV*1	
SR22, 14	PMCOUNT6	Performance count 6 register	SV*1	
SR23, 14	PMCOUNT7	Performance count 7 register	SV*1	
SR0, 15	PMSUBCND0	Performance count subcondition 0 register	SV*1	
SR1, 15	PMSUBCND1	Performance count subcondition 1 register	SV*1	
SR2, 15	PMSUBCND2	Performance count subcondition 2 register	SV*1	
SR3, 15	PMSUBCND3	Performance count subcondition 3 register	SV*1	
SR4, 15	PMSUBCND4	Performance count subcondition 4 register	SV*1	
SR5, 15	PMSUBCND5	Performance count subcondition 5 register	SV*1	
SR6, 15	PMSUBCND6	Performance count subcondition 6 register	SV*1	
SR7, 15	PMSUBCND7	Performance count subcondition 7 register	SV*1	

Note 1. Accessing in user mode becomes possible by configuring the PMUMCTRL register.

(1) TSCOUNTL — Timestamp Count L

This register, together with the timestamp count H register, implements a 64-bit counter. This register serves as the lower-order 32 bits of the 64-bit counter.

This register can be read only in user mode. A PIE exception will occur if an attempt is made to write this register in user mode. This register can be read and written in supervisor mode.

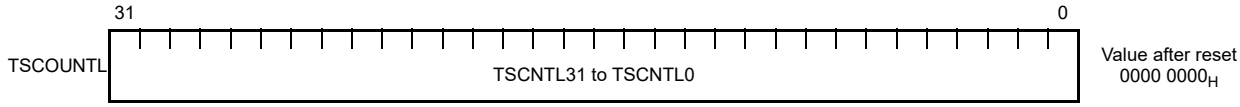


Table 3.86 TSCOUNTL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	TSCNTL31 to TSCNTL0	These bits account for the lower-order 32 bits of the 64-bit counter. If counting is enabled by the TSCTRL register, the value of this register is incremented by 1 on every clock cycle. When this register counts up from FFFF FFFF$_H$, the TSCOUNTH register is incremented by 1 as a carryover and this register wraps around to 0000 0000$_H$. This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.	R/W*1	0

Note 1. To use this register for making up an accurate 64-bit counter, see **Section 3.2.2.12, Timestamp Counter**.

(2) TSCOUNTH — Timestamp Count H

This register, together with the timestamp count L register, implements a 64-bit counter. This register serves as the higher-order 32 bits of the 64-bit counter.

This register can be read only in user mode. A PIE exception will occur if an attempt is made to write this register in user mode. This register can be read and written in supervisor mode.

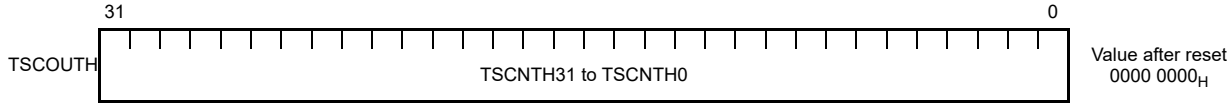


Table 3.87 TSCOUNTH Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	TSCNTH31 to TSCNTH0	These bits account for the higher-order 32 bits of the 64-bit counter. If counting is enabled by the TSCTRL register, the value of this register is incremented by 1 when the TSCOUNTL register counts up and generates a carryover. If a carryover occurs in the TSCOUNTL register when this register holds a value of FFFF FFFF$_H$, the TSCTRL.OVF bit is set to 1 and this register wraps around to 0000 0000$_H$. This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.	R/W*1	0

Note 1. To use this register for making up an accurate 64-bit counter, see **Section 3.2.2.12, Timestamp Counter**.

(3) TSCTRL — Timestamp Count Control

This register is used to control the 64-bit timestamp counter which is implemented by combining the TSCOUNTH and TSCOUNTL registers.

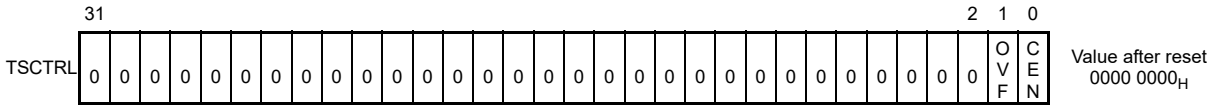


Table 3.88 TSCTRL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	OVF	This bit serves as the overflow flag. This bit is set to 1 when an overflow occurs in the TSCOUNTH register as the result of its count operation. This bit is not automatically cleared to 0. To reset the overflow state, write a 0 into this bit. Since this bit can be written with a 1, it can be an overflow condition regardless of the count operation. Although it does not affect the counter operation, care must be taken not to take that condition for an overflow.	R/W	0
0	CEN	This bit enables or disables the count operation of the 64-bit timestamp counter which is implemented by combining the TSCOUNTH and TSCOUNTL registers. 0: Disables count operation. 1: Enables count operation. If this bit is written with a 1 when it is set to 0, the counter starts counting immediately. If this bit is written with a 0 when it is set to 1, the counter stops counting immediately. If this bit is 0, the values of the TSCOUNTH and TSCOUNTL registers are preserved. There is no factor that will automatically change the value of this bit.	R/W	0

(4) PMUMCTRL — Performance Counter User Mode Control

This register specifies the accessibility to system registers of the performance measurement function in user mode. The accessibility of the system registers is specified for each of the corresponding channels of the performance measurement function.

When the access in user mode is disabled, any attempt to access one of these system registers generates a PIE exception. When the access in user mode is enabled, the system registers can be read and written.

The performance measurement function itself runs in user mode regardless of the settings of this register. Even when all channels are disabled by this register for accesses in user mode, it is possible to measure the performance in user mode if the settings are done in supervisor mode.

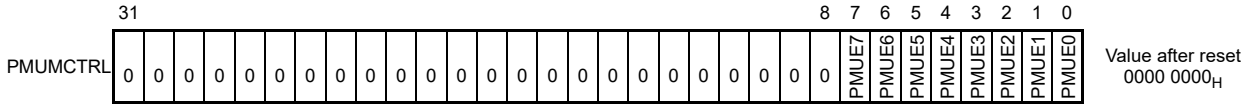


Table 3.89 PMUMCTRL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 8	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
7	PMUE7	This bit specifies the accessibility of the PMCTRL7, PMCOUNT7 and PMSUBCND7 registers in user mode. 0: Disables accesses to PMCTRL7/PMCOUNT7/PMSUBCND7 in user mode. 1: Enables accesses to PMCTRL7/PMCOUNT7/PMSUBCND7 in user mode.	R/W	0
6	PMUE6	This bit specifies the accessibility of the PMCTRL6, PMCOUNT6 and PMSUBCND6 registers in user mode. 0: Disables accesses to PMCTRL6/PMCOUNT6/PMSUBCND6 in user mode. 1: Enables accesses to PMCTRL6/PMCOUNT6/PMSUBCND6 in user mode.	R/W	0
5	PMUE5	This bit specifies the accessibility of the PMCTRL5, PMCOUNT5 and PMSUBCND5 registers in user mode. 0: Disables accesses to PMCTRL5/PMCOUNT5/PMSUBCND5 in user mode. 1: Enables accesses to PMCTRL5/PMCOUNT5/PMSUBCND5 in user mode.	R/W	0
4	PMUE4	This bit specifies the accessibility of the PMCTRL4, PMCOUNT4 and PMSUBCND4 registers in user mode. 0: Disables accesses to PMCTRL4/PMCOUNT4/PMSUBCND4 in user mode. 1: Enables accesses to PMCTRL4/PMCOUNT4/PMSUBCND4 in user mode.	R/W	0
3	PMUE3	This bit specifies the accessibility of the PMCTRL3, PMCOUNT3 and PMSUBCND3 registers in user mode. 0: Disables accesses to PMCTRL3/PMCOUNT3/PMSUBCND3 in user mode. 1: Enables accesses to PMCTRL3/PMCOUNT3/PMSUBCND3 in user mode.	R/W	0

Table 3.89 PMUMCTRL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
2	PMUE2	This bit specifies the accessibility of the PMCTRL2, PMCOUNT2 and PMSUBCND2 registers in user mode. 0: Disables accesses to PMCTRL2/PMCOUNT2/PMSUBCND2 in user mode. 1: Enables accesses to PMCTRL2/PMCOUNT2/PMSUBCND2 in user mode.	R/W	0
1	PMUE1	This bit specifies the accessibility of the PMCTRL1, PMCOUNT1 and PMSUBCND1 registers in user mode. 0: Disables accesses to PMCTRL1/PMCOUNT1/PMSUBCND1 in user mode. 1: Enables accesses to PMCTRL1/PMCOUNT1/PMSUBCND1 in user mode.	R/W	0
0	PMUE0	This bit specifies the accessibility of the PMCTRL0, PMCOUNT0 and PMSUBCND0 registers in user mode. 0: Disables accesses to PMCTRL0/PMCOUNT0/PMSUBCND0 in user mode. 1: Enables accesses to PMCTRL0/PMCOUNT0/PMSUBCND0 in user mode.	R/W	0

(5) PMCTRLn — Performance Count Control

This register controls the counting operation of the PMCOUNTn register. This CPU has 8 channels (n = 0 to 7) of performance count control registers.

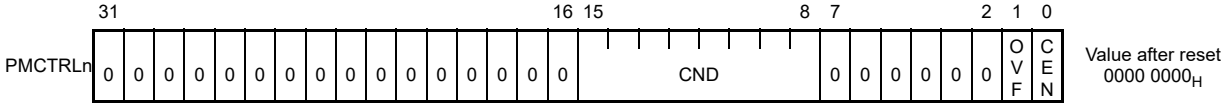


Table 3.90 PMCTRLn Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 16	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
15 to 8	CND	These bits specify the number of the event of which occurrences is to be counted. The PMCOUNTn register is not updated if the CEN bit is set (to 1) specifying an invalid event number.	R/W	0

Number	Details of Event	Speculative Operation*1
00_H	Number of all clock cycles*2	None
10_H	Number of executions of all instructions*3	None
18_H	Number of instructions that cause branch*4	None
19_H	Number of executions of conditional branch instructions (Bcond/Loop)*5	None
$1A_H$	Number of branch prediction misses of conditional branch instructions (Bcond/Loop)*5	None
20_H	Number of EIINTn acceptances*6	None
21_H	Number of FEINT acceptances	None
22_H	Number of terminating type exception acceptances*6 (including EIINTn and FEINT)	None
23_H	Number of resumable and pending type exception acceptances	None
28_H	Number of clock cycles during no interrupt is processed (period during which the ISPR register holds 0000_H*7)	None
29_H	Number of clock cycles during no interrupt is processed and interrupts are disabled (period during which the ISPR register holds 0000_H*7 and PSW.ID = 1)	None
30_H	Number of instruction fetch requests*8	Yes
31_H	Number of instruction cache hits*9	Yes
40_H	Number of stall cycles during which instructions are not issued to the instruction execution unit*10	Yes
50_H	Number of instruction fetch requests to Flash ROM*11	Yes
51_H	Number of data read requests to Flash ROM*12	Yes

Table 3.90 PMCTRLn Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
7 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	OVF	This bit serves as the overflow flag. This bit is set to 1 when an overflow occurs in the PMCOUNTn register as the result of its count operation. This bit is not automatically cleared to 0. To reset the overflow state, write a 0 into this bit. Since this bit can be written with a 1, it can be an overflow condition regardless of the count operation. Although it does not affect the counter operation, care must be taken not to take that condition for an overflow.	R/W	0
0	CEN	This bit enables or disables the count operation of the PMCOUNTn register. 0: Disables count operation. 1: Enables count operation. If this bit is written with a 1 when it is set to 0, the counter starts counting immediately. If this bit is written with a 0 when it is set to 1, the counter stops counting immediately. If this bit is 0, the values of the PMCOUNTn register is preserved. There is no factor that will automatically change the value of this bit.	R/W	0

Note 1. For the events for which speculative operation column is “Yes”, events whose execution results are cancelled are also counted.

Note 2. All clock cycles from the start of counting until the end, including the period during which no instruction is executed, are counted.

Note 3. The LDSR instruction which starts counting is not included but the LDSR instruction which stops counting is included in this number.

Note 4. Applicable instructions are JR, JMP, Bcond (only if condition is met), JARL, LOOP (only if condition is met), CALLT, DISPOSE, SWITCH, SYSCALL, CTRET, EIRET and FERET instructions. Other instructions with branch processing are not counted. Bcond and LOOP instructions are not counted if condition is not met.

Note 5. It is possible to obtain the branch prediction miss rate (hit rate) by measuring the numbers of executions of conditional branch instructions and branch prediction misses. Note that the unconditional branch (BR) of the Bcond instruction is counted in “Number of executions of conditional branch instructions” (CND = 19_H), but it is not counted in “Number of branch prediction misses of conditional branch instructions” (CND = $1A_H$).

Note 6. In EIINTn with table reference method, an exception which is detected during reading of the table or automatically saving the context to the register bank is not included in this number.

Note 7. Since this event presumes that the ISPR register is automatically updated on the acceptance of and returning from interrupt, it cannot be measured correctly if it is used with the INTCFG.ISPC bit set to 1.

Note 8. This CPU acquires the instruction code in 64-bit units. Every 64-bit acquisition is counted as a request for a single instruction fetching. Note that this is not the number of instructions. Instruction fetching by the CACHE instruction and PREF instruction is not counted.

Note 9. In this CPU, a cache hit indicates the number of requests for a single instruction fetching but not the number of instructions. Cache operation by the CACHE instruction is not counted.

Note 10. Instruction issue stall occurs when issuing of an instruction (starting execution of an instruction) is stalled if the CPU satisfies any of the following conditions.

- At execution of an STSR instruction that reads the PSW register, issue of the instruction is stalled until completion of all preceding instructions.
- At execution of a SYNCP, SYNCM, or SYNCI instruction, issue of the instruction is stalled until waiting conditions for the instruction are cleared. For details on the waiting conditions, see **Section 3.2.7.1, Synchronization Processing**.
- At execution of an LDSR instruction or STSR instruction that manipulates the cache control function system registers, issue of the instruction is stalled until completion of all preceding CACHE instructions.
- At execution of an LDSR instruction or STSR instruction that manipulates the FPU system registers, issue of the instruction is stalled until completion of all preceding FPU instructions.
- At execution of an LDSR instruction or STSR instruction that manipulates the FXU system registers, issue of the instruction is stalled until completion of all preceding FXU instructions.

- If an LDSR instruction updates the system register which guarantees reflection of its update to the subsequent instruction, issue of the instruction is stalled until the update of the system register is completed. For details, see **Section 3.2.7.3, Hazard Management after System Register Update**.
- If execution of an STSR instruction is started after execution of an LDSR instruction is started, issue of the STSR instruction is stalled until completion of the LDSR instruction. (This applies even if the system register numbers of the LDSR and STSR instructions do not match).
- When a HALT instruction is executed, issue of the subsequent instruction is stalled until conditions for releasing the halt state are generated.
- When a SNOOZE instruction is executed, issue of the subsequent instruction is stalled until conditions for releasing the snooze state are generated.
- When any of the following instructions that includes branching is executed, issue of the subsequent instruction is stalled until completion of the branching and cancellation of execution of the subsequent instruction.
  SYSCALL, CALLT, EIRET, FERET, CTRET, TRAP, FETRAP, SYNCI
- When any of the following exceptions is detected, issue of the subsequent instruction is stalled until acknowledgement of the exception is completed and cancellation of execution of the subsequent instruction.
  MIP, SYSERR (resumable), RIE, UCPOP, PIE

Note 11. Actual Flash ROM access requests are counted. If an instruction fetch request hits the instruction cache, it is not counted. Instruction fetch requests to Flash ROM by the CACHE and PREF instructions are counted. Instruction pre-fetching to Flash ROM is also counted.

Note 12. Actual Flash ROM access requests are counted. If a data read request hits the ROM data buffer, it is not counted. Pre-fetching to Flash ROM is counted. Note that the number of data read requests to Flash ROM is different from the number of Flash ROM accesses by instructions.

Note 13. Since the number of instruction fetch request (30H) and the number of instruction cache hits (31H) include the speculative instruction fetching, the number of hits divided by the number of requests may not be the accurate instruction cache hit rate of this CPU. The instruction cache hit rate of this CPU can be calculated by the number of instruction fetch requests and the number of instruction fetch requests to Flash ROM (50H). In this case, the number of instruction fetch request reduced by the number of instruction fetch requests to Flash ROM will be the number of instruction cache hits. However, even in this case, the number of instruction cache hits includes the speculative instruction fetching. Pre-fetching by the PREF instruction is counted as the number of instruction fetch requests, though accessing non-cacheable area is not counted as the number of instruction fetch requests to Flash ROM.

(6) PMCOUNTn — Performance Count

This register counts the number of occurrences of various events, which are specified by the PMCTRLn register. This CPU has 8 channels (n = 0 to 7) of performance count registers.

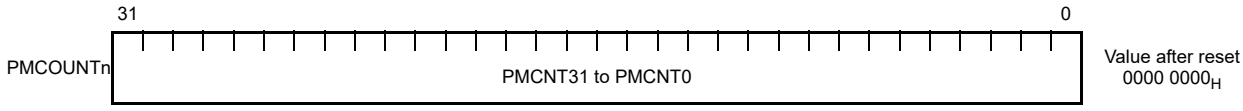


Table 3.91 PMCOUNTn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	PMCNT31 to PMCNT0	These bits form a 32-bit counter register. The counter counts the number of occurrences of the event that is specified by the PMCTRLn.CND bit. When this register counts up from FFFF FFFF$_H$, the PMCTRLn.OVF bit is set to 1 and the register wraps around to 0000 0000$_H$. This register is accessible at an arbitrary timing regardless of whether counting is enabled or disabled. In addition, it is possible to start counting at an arbitrary count by loading that value in this register before starting to count.	R/W	0

(7) PMSUBCNDn — Performance Count Sub condition

The PMSUBCNDn register specifies the subcondition of the count execution according to the setting of the PMCTRLn.CND. In the PMSUBCNDn register, 8 channels (n = 0 to 7) are equipped.

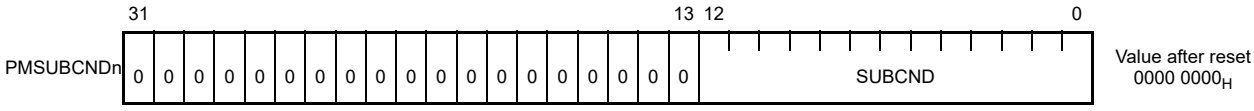


Table 3.92 PMSUBCNDn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 13	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
12 to 0	SUBCND	How the value of the SUBCND field is used depends on the setting of the PMCTRLn.CND. If the value of the PMCTRLn.CND is other than the following, the value of the SUBCND field does not affect the operation of the performance counter. **Value of the PMCTRLn.CND** / **Operation of the performance counter** 20_H / The channel number of the interrupt (EIINTn) for which the acceptance count is counted can be limited. The operation is specified by the value of bit 12. 0: The channel number to be counted is not limited. 1: The channel number to be counted is limited. If the bit 12 is set (1), only when the channel number n of the accepted interrupt (EIINTn) matches*1 the value specified by bits 11 to 0, the performance counter is counted up. 21_H / The channel number of interrupt (FEINTm) for which the number of accepted times is counted can be limited. The operation is specified by the value of bit 4. 0: The channel number to be counted is not limited. 1: The channel number to be counted is limited. If the bit 4 is set (1), only when the channel number m of the accepted interrupt (FEINTm) matches*2 the value specified by bits 3 to 0, the performance counter is counted up. The values of bits 12 to 5 do not affect the operation of the performance counter.	R/W	0

Note 1. If the total number of the channel numbers of interrupt (EIINTn) can be expressed in binary number of 11 bits or less, 0 is added to the upper side of the channel number compared with bits 11 to 0 of the SUBCND. For example, if the total number of channel numbers of interrupt (EIINTn) is 1024 (channel number is 0 to 1023), bit 12 and bit 11 of the SUBCND are set (1), bits 10 to 0 of the SUBCND are cleared (0), if it is limited to counting with the channel number 1024, the channel number always becomes inconsistent, the interrupt (EIINTn) acceptance is not counted the number of times.

Note 2. Set only the mounted channel number to this bit.

3.2.3.9 Hardware Function Registers

(a) Hardware Function System Registers

Hardware function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.93 Hardware Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR 0, 12	LSTEST0	Lock-step function self-diagnosis register 0	SV
SR1, 12	LSTEST1	Lock-step function self-diagnosis register 1	SV
SR2, 12	LSCFG	Lock-step function configuration	SV
SR5, 12	IFCR	Instruction fetch control register	SV
SR8, 12	BRPCTRL0	Branch prediction function control register	SV
SR12, 13	L1RCFG	L1RAM configuration	SV
SR24, 13	RDBCR	ROM data buffer control register	SV

(1) LSTEST0 — Lock-step Function Self-diagnosis Register 0

This register is for the self-diagnosis of the lock-step function. It can be used for the self-diagnosis of the lock-step function in combination with the LSTEST1 register. For details of the lock-step function and its availability, see **Section 44, Functional Safety**.

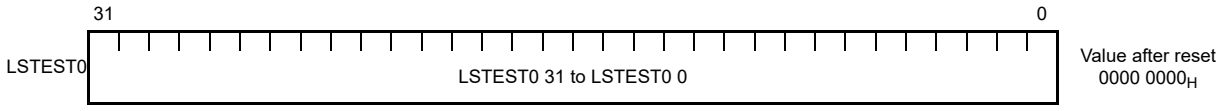


Table 3.94 LSTEST0 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	LSTEST0 31 to LSTEST0 0	[Write] Write any value to bits of the LSTEST0 register. [Read] Value in LSTEST0 is read by the master CPU of the lock-stepped CPU. Value in LSTEST1 is read by the checker CPU of the lock-stepped CPU.	R/W	0

(2) LSTEST1 — Lock-step Function Self-diagnosis Register 1

This register is for the self-diagnosis of the lock-step function. It can be used for the self-diagnosis of the lock-step function in combination with the LSTEST0 register. For details of the lock-step function and its availability, see **Section 44, Functional Safety**.

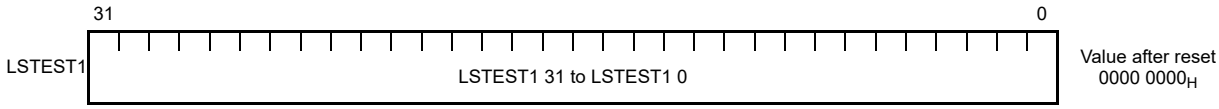


Table 3.95 LSTEST1 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	LSTEST1 31 to LSTEST1 0	[Write] Write any value to bits of the LSTEST1 register. [Read] Value in LSTEST1 is read by the master CPU of the lock-stepped CPU. Value in LSTEST0 is read by the checker CPU of the lock-stepped CPU.	R/W	0

(3) LSCFG — Configuration of Lock Step function

This register indicates the configuration of the Lock Step function.

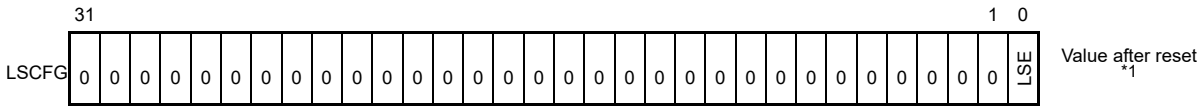


Table 3.96 LSCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	LSE	This bit indicates the availability of the Lock Step function. 0: The Lock Step function is not available. 1: The Lock Step function is available.	R	*1

Note 1. For details, see **Section 3.9.7, Product information of initial value for G4MH register**.

Note: For details of the Lock Step function, see **Section 44, Functional Safety**.

(4) IFCR — Instruction Fetch Control Register

This register controls the instruction fetching function.

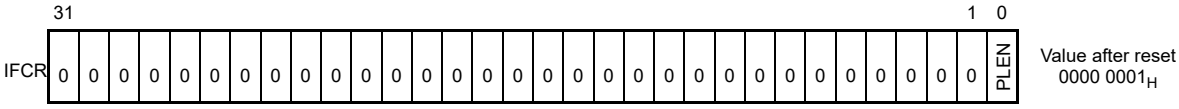


Table 3.97 IFCR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
0	PLEN	This bit specifies whether the instruction fetch preload function is enabled or disabled. 0: Disabled 1: Enabled*1	R/W*2	1

Note 1. Instruction fetch preload is executed only when the instruction cache is enabled and the fetching is to the cacheable area.

Note 2. Instruction fetch access and a system register access are independent from each other. Therefore, the settings of this register can be changed even the preload function is running. In this case, the preload function currently running normally completes with the old settings and the new settings are applied from the next operations of the preload function. If you want to suppress preloading before, during, and after changing of the register settings, we recommend execution of instructions by which the register settings are changed in the non-cacheable area.

(5) BRPCTRL0 — Branch Prediction Function Control Register

This register controls the branch prediction function.

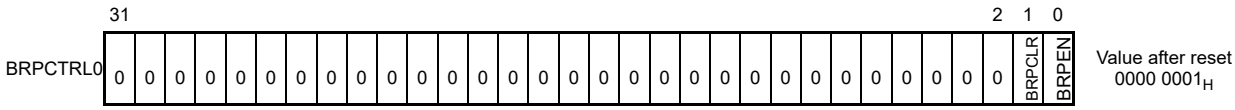


Table 3.98 BRPCTRL0 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	BRPCLR	Setting this bit to 1 clears the buffers related to branch prediction function. This bit is always read as 0.	R/W*2	0
0	BRPEN	This bit specifies whether the branch prediction function*1 is enabled or disabled. 0: Disabled 1: Enabled	R/W*2	1

Note 1. When this function is enabled, the address of the branch destination of the branch instruction is predicted based on the prediction function incorporated in the product, and instructions are fetched from the predicted address. If this function is disabled, branch prediction is not performed and instructions are fetched from the next instruction address. Branch prediction is performed only when the instruction cache is enabled and fetching is to the cacheable area.

Note 2. Branch prediction and a system register access are independent from each other. Therefore, the settings of this register can be changed even the branch prediction function is running. In this case, the branch prediction function currently running normally completes with the old settings and the new settings are applied from the next operations of the branch prediction function. If you want to suppress branch prediction before, during, and after changing of the register settings, we recommend execution of instructions by which the register settings are changed in the non-cacheable area.

(6) L1RCFG — Configuration of L1RAM

This register indicates the configuration of the L1RAM.

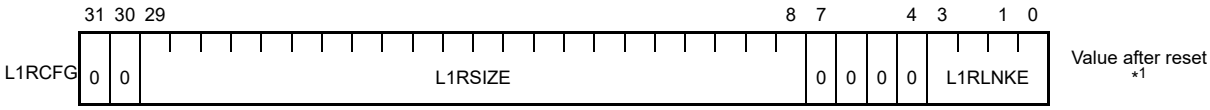


Table 3.99 L1RCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31, 30	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
29 to 8	L1RSIZE	This field indicates the L1RAM capacity (Kbytes). (For example: 000020_H: 32 Kbytes)	R	*1
7 to 4	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
3 to 0	L1RLNKE	This field indicates the number of entries of links available in L1RAM. (For example: 0010_B: 2 entries)	R	*1

Note 1. See **Section 3.9.7, Product information of initial value for G4MH register**.

(7) RDBCR — ROM Data Buffer Control Register

This register controls the ROM data buffer.

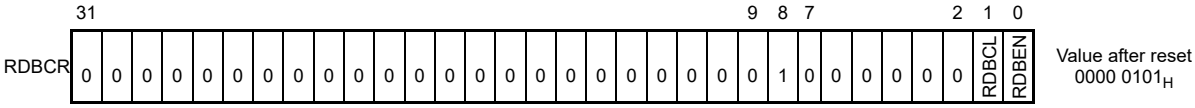


Table 3.100 RDBCR Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 9	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
8	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
7 to 2	—	(Reserved for future expansion. Be sure to set to 0.)	R	0
1	RDBCLR	Setting this bit (to 1) clears the ROM data buffer and initializes the LRU information in the ROM data buffer. If this bit is set (to 1) during load processing or when a bus request issued by the prefetch function is present, the load data is not registered in the ROM data buffer. This bit is always read as 0.	R/W*1	0
0	RDBEN	This bit specifies whether the ROM data buffer is enabled or disabled. 0: ROM data buffer is disabled. 1: ROM data buffer is enabled.	R/W*1	1

Note 1. Any change in the value of this register is processed independently of any load instructions that use the ROM data buffer. To have the change in the value of this register exactly reflected, it is recommended that this register be manipulated after all load operations are completed.

3.2.3.10 Virtualization Support Function System Registers

Virtualization support function system registers are the system registers that are used for controlling the virtualization support function. For details, see the *RH850G4MH Virtualization User's Manual: Hardware*.

Table 3.101 List of Virtualization Support Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR16, 1	HVCFG	Setting virtualization support function	SV*1

Note 1. HVCFG register is used when Virtualization support function is enable. When Virtualization support function is disable, access to this register is prohibited. For details about Virtualization support function, see the *RH850G4MH Virtualization User's Manual: Hardware*.

3.2.4 Exceptions and Interrupts

An exception is a particular event that forces branching of operation from the current program to another program.

The program at the branch destination of a given exception is called an “exception handler”.

CAUTION

This CPU handles interrupts as types of exception.

3.2.4.1 Outline of Exceptions

This section describes the elements that assign properties to exceptions, and shows how exceptions work.

(1) Exception Cause List

This CPU supports the following exceptions.

Table 3.102 Exception Cause List

Exception	Name	Source	Type*1	Saved Resource	Return/ Restoration	Exception Cause Code*2	Priority Order*3		Acknowledgment Condition (PSW)		Update (PSW)				
							Priority Level	Priority	ID	NP	UM	ID	NP	EP	EBV
RESET	Reset	Reset input	Terminating	—	—	None	1	—	x	x	0	1	0	0	0
MDP	Memory protection exception (access privilege)	Memory protection violation due to an interrupt using the table reference method	Terminating	FE	Yes	95_H	2*7	—	x	x	0	1	1	1	s
FENMI	FENMI interrupt	Interrupt input terminal	Terminating	FE	No	$E0_H$	4	1	x	x	0	1	1	0	s
FEINT	FEINT interrupt	Interrupt input terminal	Terminating	FE	Yes	$F0_H$-FF_H	4	2	x	0	0	1	1	0	s
SYSERR	System error	Error due to context saving to the register bank	Terminating	FE	No	$1C_H$	4	3	x	x	0	1	1	1	s
EIINT0-2047	User interrupt	Interrupt input terminal	Terminating	EI	Yes	1000_H-$17FF_H$*5	5	*10	0	0	0	*9	s	0	s
MIP	Memory protection exception (execution privilege)	Memory protection violation due to instruction fetching	Resumable	FE	Yes	90_H	7	1	x	x	0	1	1	1	s
SYSERR	System error	Error due to instruction fetching	Resumable	FE	No	10_H-$1F_H$*4	7	2	x	x	0	1	1	1	s
UCPOP	Coprocessor unusable exception	Execution of a coprocessor instruction/access permission violation	Resumable	FE	Yes	80_H-82_H*6	7	3	x	x	0	1	1	1	s
RIE	Reserved instruction exception	Execution of a reserved instruction	Resumable	FE	Yes	60_H	7	4	x	x	0	1	1	1	s
PIE	Privilege instruction exception	Execution of a privileged instruction/access permission violation	Resumable	FE	Yes	$A0_H$	7	5	x	x	0	1	1	1	s
SYSERR	System error	Error prior to context restoration from the register bank	Resumable	FE	No	$1D_H$	7	6	x	x	0	1	1	1	s
MAE	Misalignment exception	Misaligned access occurrence	Resumable	FE	Yes	$C0_H$	8	1	x	x	0	1	1	1	s
MDP	Memory protection exception (access privilege)	Memory protection violation due to operand access	Resumable	FE	Yes	91_H	8	2	x	x	0	1	1	1	s
FPE	FPU exception (precise)	Execution of an FPU instruction	Resumable	EI	Yes	71_H	8	3	x	x	0	1	s	1	s
FXE	FXU exception (precise)	Execution of an FXU instruction	Resumable	EI	Yes	75_H	8	4	x	x	0	1	s	1	s
SYSCALL	System call	Execution of the SYSCALL instruction	Pending	EI	Yes	8000_H-$80FF_H$	9	*8	x	x	0	1	s	1	s
FETRAP	FE level trap	Execution of the FETRAP instruction	Pending	FE	Yes	31_H-$3F_H$	9		x	x	0	1	1	1	s
TRAP0	EI level trap 0	Execution of the TRAP instruction	Pending	EI	Yes	40_H-$4F_H$	9		x	x	0	1	s	1	s
TRAP1	EI level trap 1	Execution of the TRAP instruction	Pending	EI	Yes	50_H-$5F_H$	9		x	x	0	1	s	1	s

Note: s: Retained, x: Not an acknowledgment condition

Note 1. For details, see **Section 3.2.4.1 (3), Types of Exceptions**.

Note 2. Represents lower-order 16 bits of the exception cause code. The higher-order 16 bits of the exception cause code are loaded with a detail code which is defined for each exception. The code is 0000_H unless it is specifically described in the individual functional descriptions.

Note 3. The acknowledgment priority for exceptions is checked by the priority level, and then priority. A smaller value has a higher priority. For details, see **Section 3.2.4.1 (4), Exception Acknowledgment Conditions and Priority Order**.

Note 4. For details, see **Section 3.2.4.1 (2), Overview of Exception Causes**.

Note 5. 1000_H-$17FF_H$ (channels 0 to 2047) are selected according to the channel.

Note 6. 80_H-82_H correspond to the coprocessor use permissions (CU0-CU2), respectively.

Note 7. The case in which an MDP exception occurs during the processing (table read or automatic context saving onto the register bank) which is performed after a table reference method interrupt (EIINTn) is selected as the result of priority determination. The occurrence of this type of exceptions takes precedence over that of the terminating-type exceptions except the reset. For details, see **Section 3.2.4.1 (2), Overview of Exception Causes**.

Note 8. Occurs in an exclusive manner as it is caused by the execution of an instruction. There is no priority difference within the same priority level.

Note 9. There are cases in which the PSW.ID bit is set to 0 on interrupts of table reference method in which the register bank is used. For details, see **Section 3.2.4.5 (2), Automatic Context Saving**.

Note 10. The priority of EIINT0-EIINT2047 varies depending on settings of the interrupt controller. For details, see **Section 3.2.4.1 (5), Interrupt Exception Priority and Priority Masking**.

(2) Overview of Exception Causes

The following is an overview of the exception causes handled by the CPU.

(a) RESET

RESET is generated when inputting a reset signal. For details, see **Section 3.2.8, Reset**.

(b) FENMI, FEINT, and EIINT

These are interrupts generated by interrupt signals from the interrupt controller to activate a certain program. For details about the interrupt functions, see **Section 3.2.3.3, Interrupt Function Registers** and the specifications of the interrupt controller incorporated in your product.

(c) SYSERR

This is a system error exception.

An error occurring during automatic context saving using the register bank function is notified as a terminating-type SYSERR exception. In this case, as with the case of acknowledging an interrupt, the PC of the instruction that is interrupted when the exception occurred is loaded in the FEPC and the PSW at that time in the FEPSW, respectively. An error occurring during the restoration of the context with the RESBANK instruction is notified as a resumable-type SYSERR exception. In this case, the PC of the RESBANK instruction is loaded in the FEPC and the PSW at that time in the FEPSW, respectively.

An error that occurs at an instruction fetch access is notified as a resumable-type SYSERR exception. In this case, the PC of the instruction to be fetched is loaded in the FEPC and the PSW at that time in the FEPSW, respectively.

Table 3.103 lists the exception cause codes that are loaded in the lower-order 16 bits of the FEIC register when SYSERR exceptions occur. The higher-order 16 bits of the exception cause code are padded with 0s.

Table 3.103 Lower-order 16 Bits of the Exception Cause Codes Associated with the SYSERR Exception

Exception Cause Code	Cause
11_H	A response error occurred in a bus slave at the time of instruction fetching.*1
13_H	An error within the scope of safety functions, such as an ECC error or parity error, occurred at the time of instruction fetching.*1,*2
$1C_H$	An error occurred when automatically saving context to the register bank.
$1D_H$	An error occurred at the time of context restoration from the register bank (during the execution of the RESBANK instruction).

Note 1. For details, see **Section 3.9.8, Product information of SYSERR factor**.

Note 2. When an ECC or parity error is found in fetching from the instruction cache, it is handled as a cache miss so a SYSERR exception does not occur.

All the causes that generate a SYSERR exception for this CPU are listed in **Table 3.103**. An error that is detected externally to the CPU does not generate a SYSERR exception.

(d) FPE

These are exceptions that occur when a floating-point instruction is being executed. For details, see **Section 3.2.6.1, Floating-Point Operation**.

(e) FXE

These are exceptions that occur when an extended floating-point instruction is being executed. For details, see **Section 3.2.6.2, Extended Floating-Point Operation**.

(f) MIP and MDP

These are exceptions that occur when the MPU detects a violation. Detecting an exception is performed when the address at which the instruction will access the memory is calculated. For details, see **Section 3.2.5.1, Memory Protection Unit (MPU)**.

There are cases in which an MDP exception is detected during a table read or automatic context saving onto the register bank when a table reference method interrupt (EIINTn) is selected as the result of determining the priority level of the terminating-type exception. In such a case, the execution of the instruction is interrupted and an MDP exception (terminating-type) is generated as with an ordinary interrupt. In this case, the PC of the instruction that is interrupted is saved in the FEPC and the PSW that is established before the interrupt is accepted is saved in the FEPSW, respectively. For the table reference method, see **Section 3.2.4.4, Exception Handler Address**.

(g) RIE

This is a reserved instruction exception. This exception occurs when an attempt is made to execute the opcode of an instruction other than an instruction whose operation is defined. The operation is the same as the RIE instruction. For details of reserved instruction and RIE instruction, see the *RH850G4MH User's Manual: Software*.

(h) PIE

This is a privilege instruction exception. This exception occurs when an attempt is made to execute an instruction that does not have the required privilege. For details, see **Section 3.2.2.1 (3), CPU Operating Mode and Privileges**, **Section 3.2.2.2, Instruction Execution**, and **Section 3.2.2.5(3)(a), LDSR and STSR**.

(i) UCPOP

This is an exception that occurs when an attempt is made to execute a coprocessor instruction when the coprocessor in question is not usable. For details, see **Section 3.2.2.4, Coprocessors**.

(j) MAE

This is an exception that occurs when the result of address calculation is a misaligned address. For details, see **Section 3.2.2.6 (3), Data Alignment**.

(k) TRAP, FETRAP, and SYSCALL

These are exceptions that occur according to the result of instruction execution. For details, see the *RH850G4MH User's Manual: Software*.

(3) Types of Exceptions

This CPU divides exceptions into the following three types according how they are executed.

- Terminating exceptions
- Resumable exceptions
- Pending exceptions

(a) Terminating Exceptions

In the case of an exception of this type, the exception is acknowledged by interrupting the current instruction before its operation is executed. These exceptions include interrupts, etc.

In the case of interrupts, their occurrence is not dependent upon the result of executing the current instruction, that is, generation is not related to that instruction. When an interrupt occurs, the PSW.EP bit is cleared to 0, unlike other exceptions. Consequently, termination of the exception handler routine is reported to the external interrupt controller when the return instruction is executed. Be sure to execute a return instruction from an interrupt while the PSW.EP bit is cleared to 0.

CAUTIONS

1. **The PSW.EP bit is cleared to 0 only when an interrupt (EIINTn, FEINT, or FENMI) is acknowledged. It is set to 1 when any other exception occurs.**
   If a return instruction from the exception handler routine that has been started by an interrupt is executed while the PSW.EP bit is set to 1, the resources on the external interrupt controller might not be released, causing malfunctioning.
2. **A terminating exception may be accepted during the execution of an instruction that performs multiple memory accesses. In this case, although the execution of the instruction is terminated, the result of memory accesses that have been already completed are not canceled. For example, memory is updated by the PREPARE instruction and the general-purpose registers are updated by the DISPOSE instruction. However, it is guaranteed that the PC and SP retain the original values required to re-execute the instruction. For details, see the *RH850G4MH User's Manual: Software*. The relevant instructions are listed below.**
   – **PREPARE, DISPOSE, PUSHSP, POPSP, RESBANK, STM.MP, LDM.MP**
3. **For the instructions that perform load-access and branch, a terminating exception may be accepted after the load-access is completed. However, it is guaranteed that the PC and the system registers (such as CTPC) that are updated on completion of the instruction retain the original values required to re-execute the instruction. For details, see the *RH850G4MH User's Manual: Software*. The relevant instructions are listed below.**
   – **CALLT, SYSCALL, SWITCH**

(b) Resumable Exceptions

A resumable exception is the one that occurs during the operation of an instruction and that is accepted without the completion of that instruction. It is also called a precise exception because it is accepted precisely without the completion of the instruction that was being executed and without the execution of the subsequent instructions. The occurrence of this exception suppresses the update of the general-purpose and system registers. In addition, since the return PC of the exception points to the very instruction that caused the exception, it is possible to resume the execution at the point before the exception occurred.

The return PC of a resumable exception is the PC of the instruction which caused the exception (current PC).

(c) Pending Exceptions

This is an exception acknowledged after the execution of an instruction finishes as a result of executing the instruction operation. Pending exceptions include software exceptions. Because pending exceptions occur as a result of normal instruction execution, the processing resumes with the instruction following the instruction that caused the pending exceptions when processing control is returned. The original processing can be normally continued after the exception handling.

The return PC of a pending exception is the PC of the next instruction (next PC).

(4) Exception Acknowledgment Conditions and Priority Order

This CPU acknowledges only one exception at specific timing based on the exception acknowledgment conditions and priority order. The exception to be acknowledged is determined based on the exception acknowledgment conditions and priority order, as shown in **Figure 3.16** below.

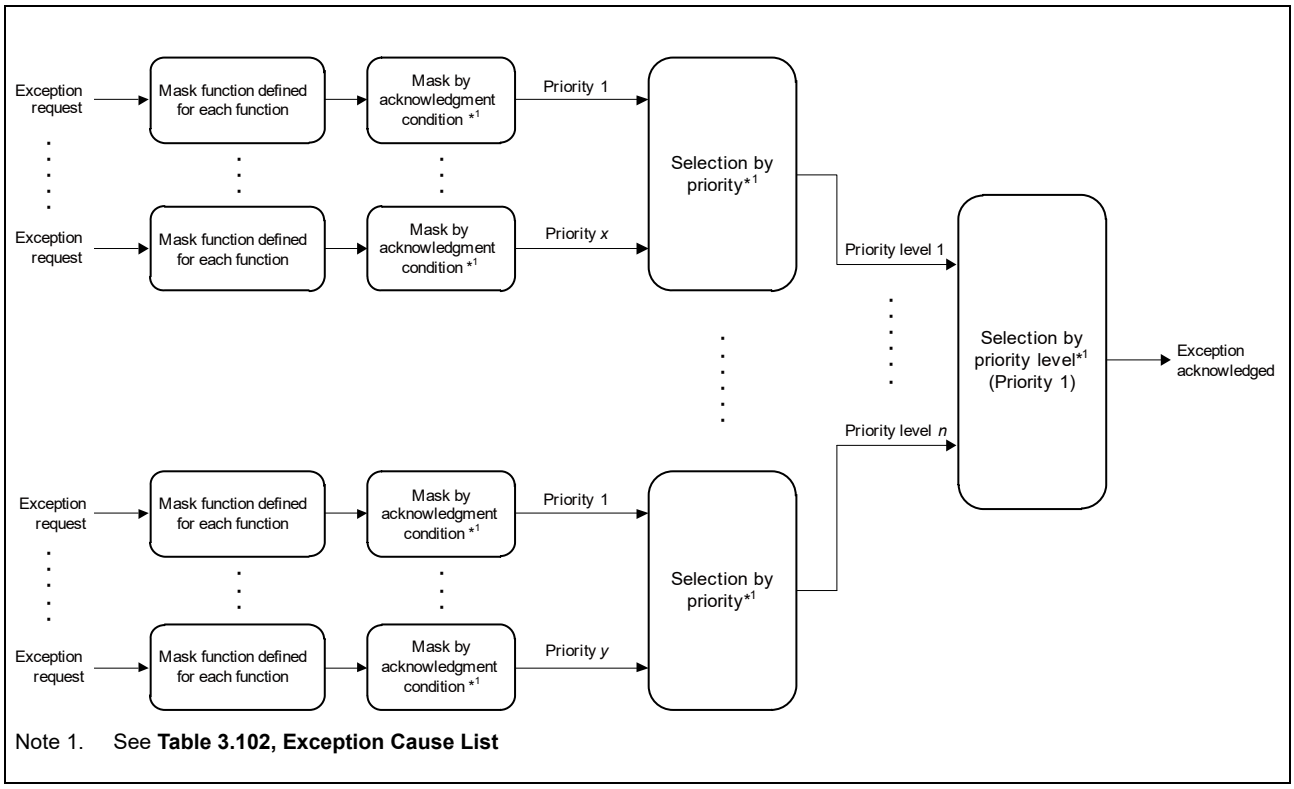


Figure 3.16 Exception Acknowledgment Conditions and Priority Order

In **Table 3.102**, an exception with “0” in the acknowledgment condition column can be acknowledged when the corresponding bit is “0”. This kind of exception is not accepted when the corresponding bit is “1”. When it changes to “0” and the acknowledgment conditions are met, acknowledgment of the exception becomes possible. If no value is specified for a bit, it is not an acknowledgment condition. If multiple bits are specified as conditions, all the conditions must be met simultaneously.

Some exceptions are provided with dedicated mask functions that are unique to their functionality. The decision on the acceptability of the exceptions based on the individually defined mask function is made before the decision on the acknowledgement (acceptance) conditions listed in **Table 3.102**. For details, see the description of the cause of the individual exceptions.

If more than two exceptions satisfy the acknowledgment conditions simultaneously, one exception is selected according to the priority order. The priority order is determined in multiple stages; priority level, and then priority. A smaller number has a higher priority.

When this CPU accepts an exception, it returns an acceptance response to the requesting module.

If a terminating-type exception is not accepted, the request for that exception is not made pending by this CPU. To keep the exception request pending for acceptance, the module such as the interrupt controller that issued the terminating-type exception request to the CPU makes the exception request pending.

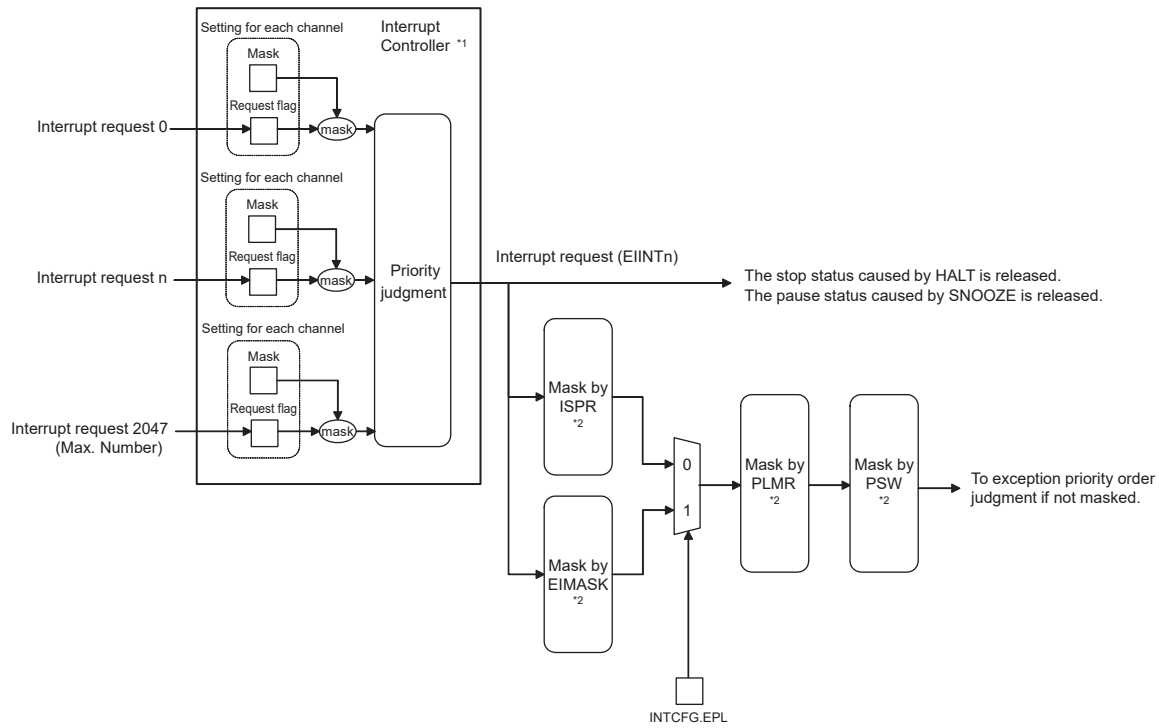
Any exceptions that occur at the same time as a reset are not accepted. For details, see **Section 3.2.4.2 (1), Special Operations**.

For details about acknowledgment conditions, priority level, and priority, see **Table 3.102**.

(5) Interrupt Exception Priority and Priority Masking

An interrupt (EIINTn) can be masked for each exception priority or interrupt priority by setting registers. This function allows the software implementation of an interrupt ceiling with a more flexible software structure and no maintenance.

Figure 3.17 shows an overview of the functions of interrupt exception priority and priority masking.



Note 1. For details about the interrupt controller, see **Section 6, Interrupts**.

Note 2. EIINTn which is masked by each masking function can be shown by each bit of IMSR or ICSR.PMEI.

Figure 3.17 Interrupt Exception Priority and Priority Masking

(a) Interrupt Priority

This CPU supports the following priority levels for interrupts (EIINTn):

When the interrupt priority level extension function is disabled (INTCFG.EPL is cleared to 0): Maximum 16 priority level

When the interrupt priority level extension function is enabled (INTCFG.EPL is set to 1): Maximum 64 priority level

For details on the procedure to set interrupt priority, see **Section 6, Interrupts**.

(i) When interrupts with priority level 16 to 63 occur in case the interrupt priority level expansion is disabled (INTCFG.EPL is cleared to 0)

The function for acknowledgment of interrupt (EIINTn) works limited. Interrupt can be masked by ISPR and PLMR. If either one of bit for ISPR is set to 1, all interrupts (EIINTn) whose priority is lower than or equal to 16 is masked. PLMR always judges acknowledgment based on the set value and interrupt priority level regardless of the setting of the interrupt priority level extension function. On the

other hand, the priority information of acknowledged interrupt (EIINTn) is not saved in ISPR and PSW.EIMASK.

As it's mentioned above, it's not recommended to use because the priority information is not handled by CPU even if the acknowledge of interrupt (EIINTn) is possible.

(ii) Constraints for interrupts with priority level 16 to 62

Regardless of whether the interrupt priority level extension function is enabled or disabled, the following constraints are valid for interrupts (EIINTn) with priority level 16 to 62.

- When the acknowledged interrupt (EIINTn) has a priority level between 16 and 62 and the exception handler address is generated by the direct vector address method, the offset address is $1F0_H$ that is the same as for interrupt with priority level 15. Therefore, the same exception handler address is used for all interrupts (EIINTn) whose priority level is lower than or equal to 15. Refer to **Section 3.2.4.4, Exception Handler Address** for more details.
- When the acknowledged interrupt (EIINTn) whose priority level between 16 and 62, the availability of register bank is specified by RBCR0.BE[15] that is the same as for interrupts whose priority level is 15. For all interrupts (EIINTn) whose priority level is lower than or equal to 15, the availability of the register bank cannot be specified by priority level. Refer to **Section 3.2.4.5 (2), Automatic Context Saving**.
- When the acknowledged interrupt (EIINTn) whose priority level is between 16 and 62, the value of PSW.ID bit is specified by RBCR1.NC[15] after being saved to the register bank, that is the same in the case of priority level 15. For all interrupts (EIINTn) whose priority level is lower than or equal to 15,the value of the PSW.ID bit after being saved to the register bank cannot be specified by priority level. Refer to **Section 3.2.4.5(2)(b), Suppressing the Update of the PSW.ID Bit**.

(iii) Constraint for interrupt with priority level 63

If the interrupt priority level extension function is disabled, the interrupt (EIINTn) with priority level 63 (lowest priority level) will always be masked by PLMR. If the interrupt priority level extension function is enabled, the interrupt (EIINTn) with priority level 63 (lowest priority level) will always be masked by PSW.EIMASK. As it's mentioned above, interrupt (EIINTn) with priority level 63 is not acknowledged regardless of whether the interrupt priority level extension function is enabled or disabled. But both stop status caused by the HALT instruction and temporary halt status caused by the SNOOZE instruction can be released by occurrence of interrupt with priority level 63.

(b) Interrupt Priority Mask

If interrupt priority level extension function is disabled (INTCFG.EPL is cleared to 0), interrupt (EIINTn) acknowledge is judged by ISPR and PLMR. If interrupt priority level extension function is enabled (INTCFG.EPL is set to 1), interrupt (EIINTn) acknowledge is judged by PSW.EIMASK and PLMR.

(i) ISPR

Interrupt acknowledge is judged by ISPR only when interrupt priority level extension function is disabled.

For the ISPR register, the bit corresponding to the priority is set to 1 when the hardware acknowledges an interrupt, and interrupts with the same or lower priority are masked. When the EIRET instruction

corresponding to the interrupt is executed, the corresponding bit of the ISPR register is cleared to 0 to clear the mask.

This automatic interrupt ceiling makes multiple interrupts servicing easy without using software control.

The function of the INTCFG register allows you to disable automatic update of the ISPR register upon acknowledgment of and return from an interrupt. To perform interrupt ceiling control by using software without using the function of the ISPR register, set the ISPC bit in the INTCFG register to 1, clear the ISPR register, and then control the ceiling value with software by using the PLMR register.

(ii) PSW.EIMASK

Interrupt acknowledge is judged by PSW.EIMASK only when interrupt priority level extension function is enabled.

PSW.EIMAK masks the interrupt (EIINTn) whose priority level is lower than or equal to the set value. When the CPU acknowledges an interrupt (EIINTn), its interrupt priority is stored. When EIRET or FERET instruction is executed, the value of EIPSW.EIMASK or FEPSW.EIMASK is stored. The settings of PSW.EIMASK can be changed by LDSR instruction.

(iii) PLMR

PLMR masks the interrupt (EIINTn) whose priority level is lower than or equal to the set value.

The PLMR register allows you to mask specific interrupt priorities with software. Use it to raise the priority level of the interrupt ceiling temporarily in a program. The mask setting specified by the ISPR register or PSW.EIMASK and the mask setting of PLMR might overlap, and an interrupt is masked if it is masked by any of them. Normally, use the PLMR register to raise the ceiling value from the ceiling value of the ISPR register or PSW.EIMASK.

Also, when you are using the PLMR register, you can check if any interrupt is masked with the PLMR register by using the ICSR register.

(6) Return and Restoration

When exception handling has been performed, it might affect the original program that was interrupted by the acknowledged exception. This effect is indicated from two perspectives: “Return” and “Restoration”.

- Return: Indicates whether or not the original program can be re-executed from where it was interrupted.
- Restoration: Indicates whether or not the processor statuses (status of processor resources such as general-purpose registers and system registers) can be restored as they were when the original program was interrupted.

An exception that cannot be returned or restored from (“No” in **Table 3.102**) might cause the return PC to be lost, making it impossible to return from the exception to the original processing by using a return instruction. An exception whose trigger cannot be selected is an unreturnable or unrestorable exception.

For an unrestorable exception, it is possible to return to the original program flow. However, because the state before the occurrence of the exception cannot be restored at that point, care must be taken in continuing subsequent program operation.

(7) Context Saving

To save the current program sequence when an exception occurs, appropriately save the following resources according to the function definitions. There are resources that are automatically saved by hardware and resources that need to be saved by software.

- Program counter (PC)
- Program status word (PSW)
- Exception cause code (EIIC, FEIC)
- Work system register (EIWR, FEWR)

The resource to use as the saving destination is determined according to the exception type. Saved resource determination is described below.

For exceptions that use the register bank function, specific resources are automatically saved. For details, see **Section 3.2.4.5, Register Bank Function**.

(a) Context Saving

Exceptions with certain acknowledgment conditions might not be acknowledged at the start of exception handling, based on the pending bits (PSW.ID and NP bits) that are automatically set when another exception is acknowledged.

To enable multiple exception processing which makes exceptions of the same level acceptable again, the return registers and certain information about the corresponding exception causes must be saved, such as to a stack. This information that must be saved is called the “context”.

In principle, it is necessary to make sure that no exceptions of the same level can occur before saving the context.

The working system registers that can be used in the work of saving contexts and those for which the values are saved to enable the handling of multiple exceptions as required are referred to as the basic context registers.

These basic context registers are provided for each exception level. For this reason, it is possible to precisely return from the current exception since its context will not be overwritten when an exception of a different level occurs before saving the current context.

Table 3.104 Basic Context Registers

Exception Level	Basic Context Registers
EI level	EIPC, EIPSW, EIIC, EIWR
FE level	FEPC, FEPSW, FEIC, FEWR

See **Section 3.2.2.3 (2), Exception Level**.

3.2.4.2 Operation when Acknowledging an Exception

Check whether each exception that is reported during instruction execution is acknowledged according to the priority. The procedure for exception-specific acknowledgment operation is shown below.

<1> Check whether the acknowledgment conditions are satisfied and whether exceptions are acknowledged according to their priority.

<2> Calculate the exception handler address according to the current PSW value*1

<3> For FE level exceptions, the following processing is performed.

- Saving the PC to FEPC
- Saving the PSW to FEPSW
- Storing the exception cause code in FEIC
- Updating the PSW*2
- Store the exception handler address calculated in (2) in the PC, and then pass control to the exception handler.

<4> For EI level exceptions, the following processing is performed.

- Saving the PC to EIPC
- Saving the PSW to EIPSW
- Storing the exception cause code in EIIC
- Updating the PSW*2
- Store the exception handler address calculated in (2) in the PC, and then pass control to the exception handler.
- When the exception is an interrupt that uses the register bank, save the context automatically*3.

Note 1. For details, see **Section 3.2.4.4, Exception Handler Address**.

Note 2. For the values to be updated, see **Table 3.102**.

Note 3. For details on register banks, see **Section 3.2.4.5, Register Bank Function**.

Figure 3.18 shows the steps <1> to <4>.

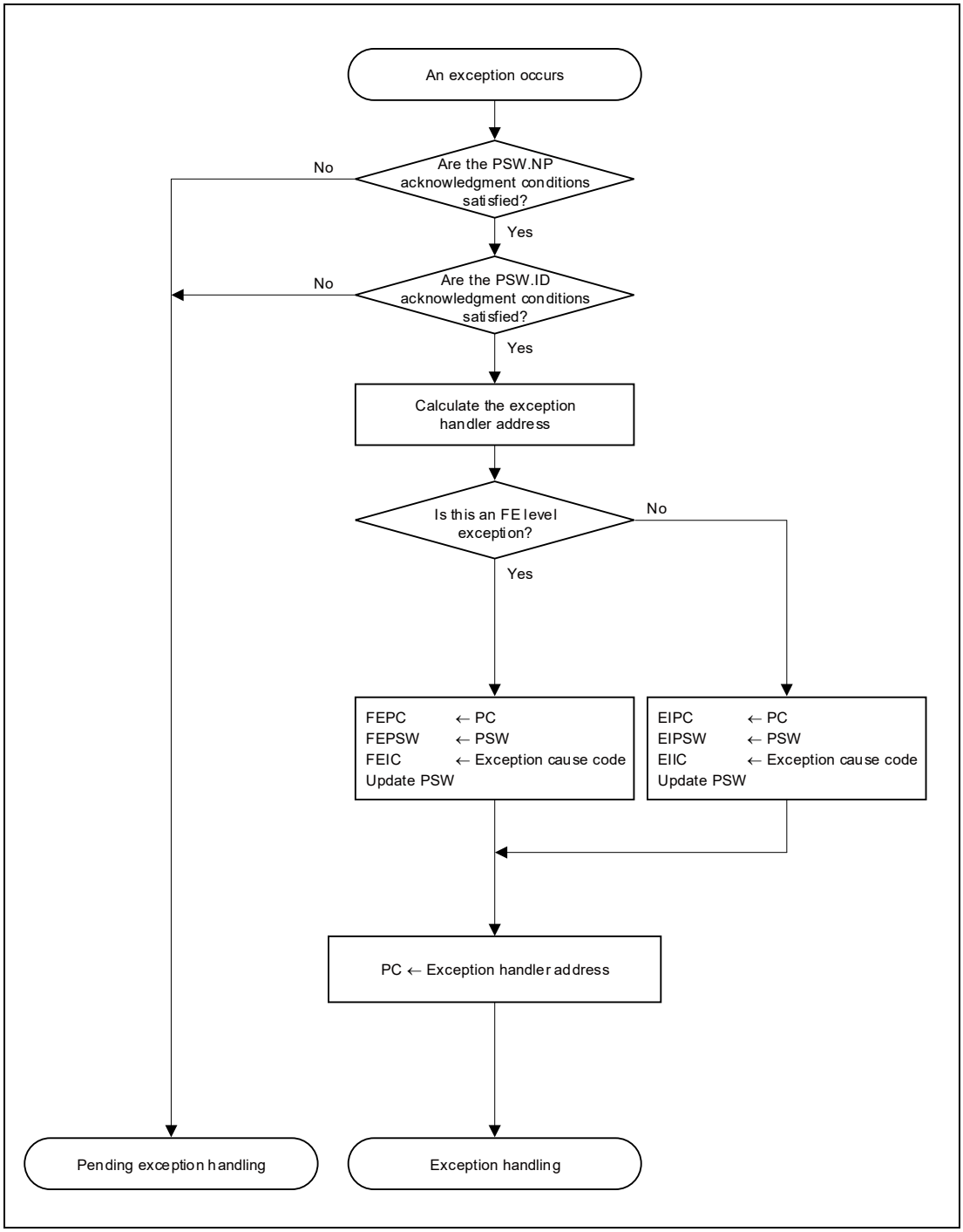


Figure 3.18 Operation When Acknowledging an Exception

(1) Special Operations

(a) EP Bit of PSW Register

If an interrupt is acknowledged, the PSW.EP bit is cleared to 0. If an exception other than an interrupt is acknowledged, the PSW.EP bit is set to 1.

The operation of the CPU when executing the EIRET and FERET instructions depends on the state of the EP bit. If the EP bit is cleared to 0, the CPU notifies the external interrupt controller of the termination of the exception processing routine. This function is necessary to properly control the request flag in the interrupt controller and other resources upon return from the interrupt. When the EP bit is cleared to 0, and the EIRET instruction is executed, the bit with the highest priority (0 is the highest) among the bits set to 1 in ISPR.ISP15 to ISPR.ISP0 is cleared to 0.

To return from an interrupt, be sure to execute the return instruction with the EP bit cleared to 0.

(b) Coprocessor Unusable Exception

For coprocessor unusable exceptions, the opcodes that cause the exception depend on the status of the CU bit of the PSW register.
When a coprocessor is not included in the product or it is not usable, if an attempt is made to execute a coprocessor instruction corresponding to the coprocessor, a coprocessor unusable exception (UCPOP) immediately occurs. If an LDSR or STSR instruction attempts to access a system register of the coprocessor, a coprocessor unusable exception (UCPOP) immediately occurs too.
For details, see **Section 3.2.2.4 (3), Coprocessor Unusable Exceptions**.

(c) Reserved Instruction Exception

If an opcode that is reserved for future function extension and for which no instruction is defined is executed, a reserved instruction exception (RIE) occurs.

The opcode that always generates a reserved instruction exception is defined as the RIE instruction.

(d) Reset

Reset is performed in the same way as exception handling, but it is not regarded as EI level exception or FE level exception. The reset operation is the same that of an exception without acknowledgment conditions, but the value of each register is changed to the value after reset. In addition, returning to the original program from the reset is not possible.

All exceptions that have occurred at the same time as CPU initialization are canceled and not acknowledged even after CPU initialization.

For details, see **Section 3.2.8, Reset**.

3.2.4.3 Return from Exception Handling

To return from exception handling, execute the return instruction (EIRET or FERET) corresponding to the relevant exception level.

When a context has been saved, such as to a stack, the context must be restored before executing the return instruction. When execution is returned from an unrestorable exception, the status before the exception occurs in the original program cannot be restored. Consequently, the execution result might differ from that when the exception does not occur.

The EIRET instruction is used to return from EI level exception handling and the FERET instruction is used to return from FE level exception handling.

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes control to the return PC address.

<1> If the PSW.EP bit is set to 0, the CPU notifies the interrupt controller of the termination of the exception routine.

<2> When the EIRET instruction is executed while PSW.EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.
When the FERET instruction is executed, the CPU does not update the ISPR register.

<3> When the EIRET instruction is executed, return PC and PSW are loaded from the EIPC and EIPSW registers.
When the FERET instruction is executed, return PC and PSW are loaded from the FEPC and FEPSW registers.

<4> Control is passed to the address indicated by the return PC that were loaded.

Figure 3.19 shows the flow for returning from exception handling using the EIRET or FERET instruction.

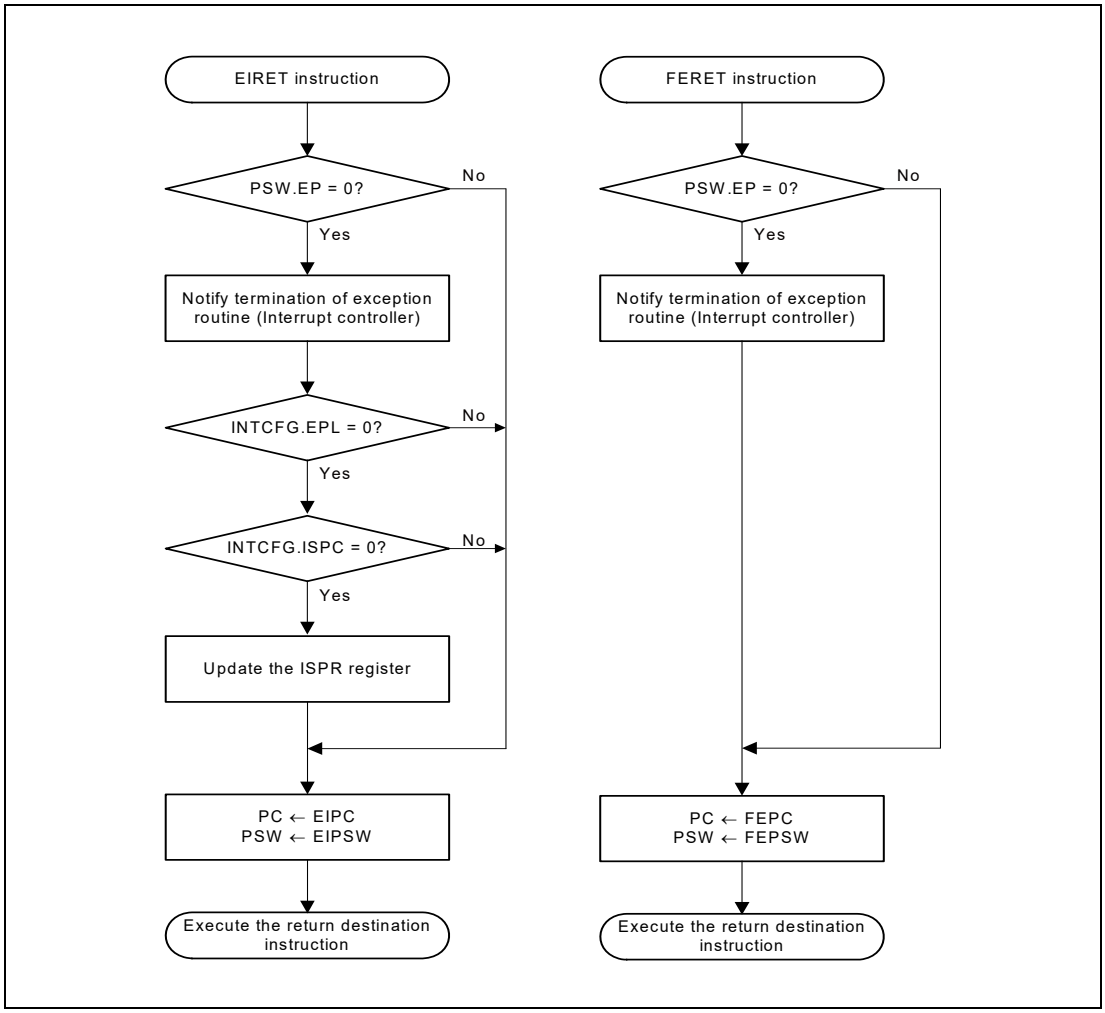


Figure 3.19 Return Instruction-Based Exception Return Flow

3.2.4.4 Exception Handler Address

For this CPU, the exception handler address used for execution during reset input, exception acknowledgment, or interrupt acknowledgment can be changed according to the settings.

(1) Resets, Exceptions, and Interrupts

The exception handler address for resets, exceptions and interrupts is determined by using the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, RBASE register, and EBASE register. For user interrupts, table reference method can be also specified. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory.

(a) Direct Vector Method

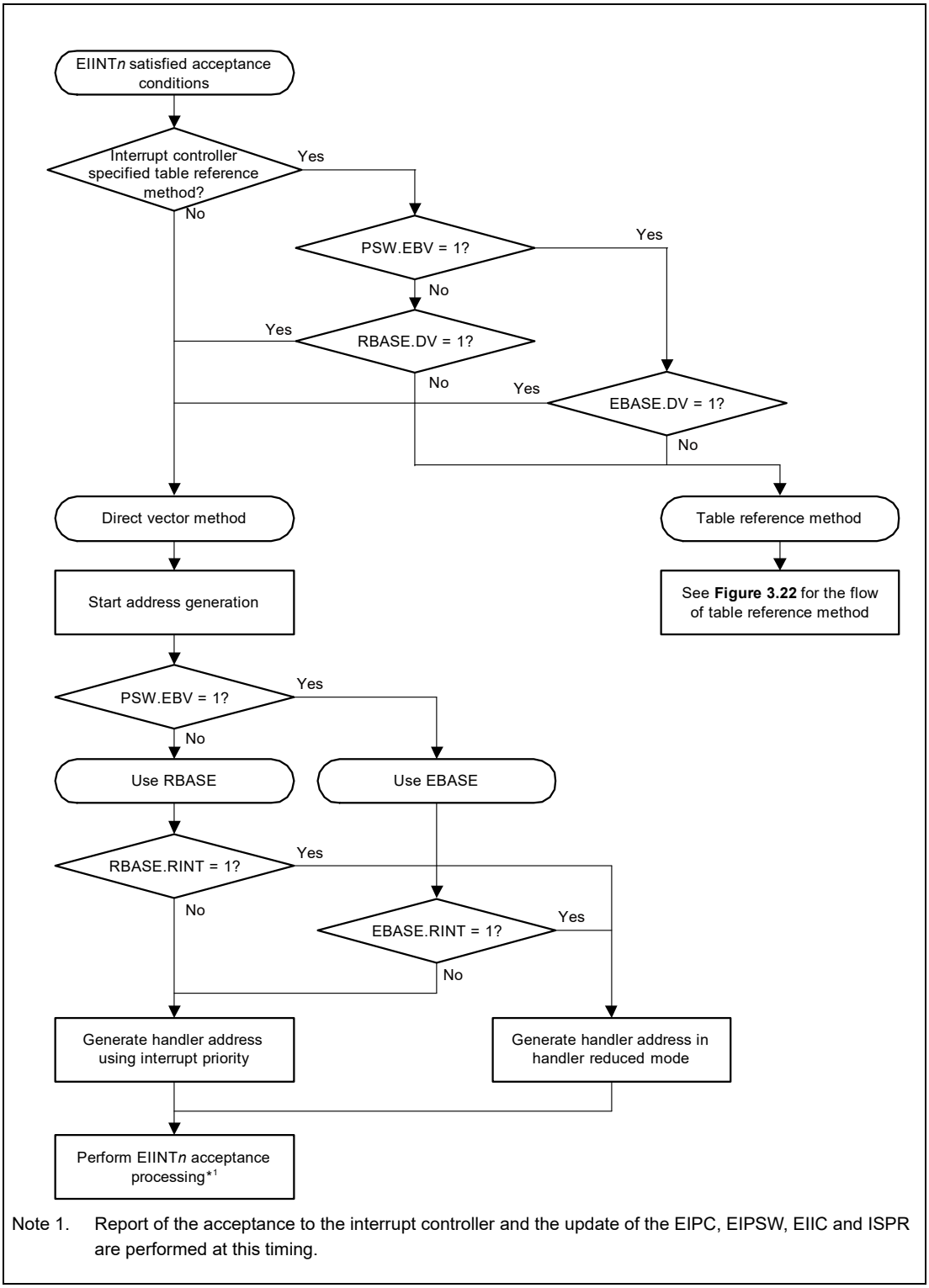
This CPU uses the result of adding the offset address shown in **Table 3.105** to the base address indicated by the RBASE or EBASE register as the exception handler address.

Whether to use the RBASE or EBASE register as the base address is selected according to the PSW.EBV bit. If the PSW.EBV bit is set to 1, the EBASE register value is used as the base address. If the bit is cleared to 0, the RBASE register value is used as the base address.

However, reset input always refers to the RBASE register.

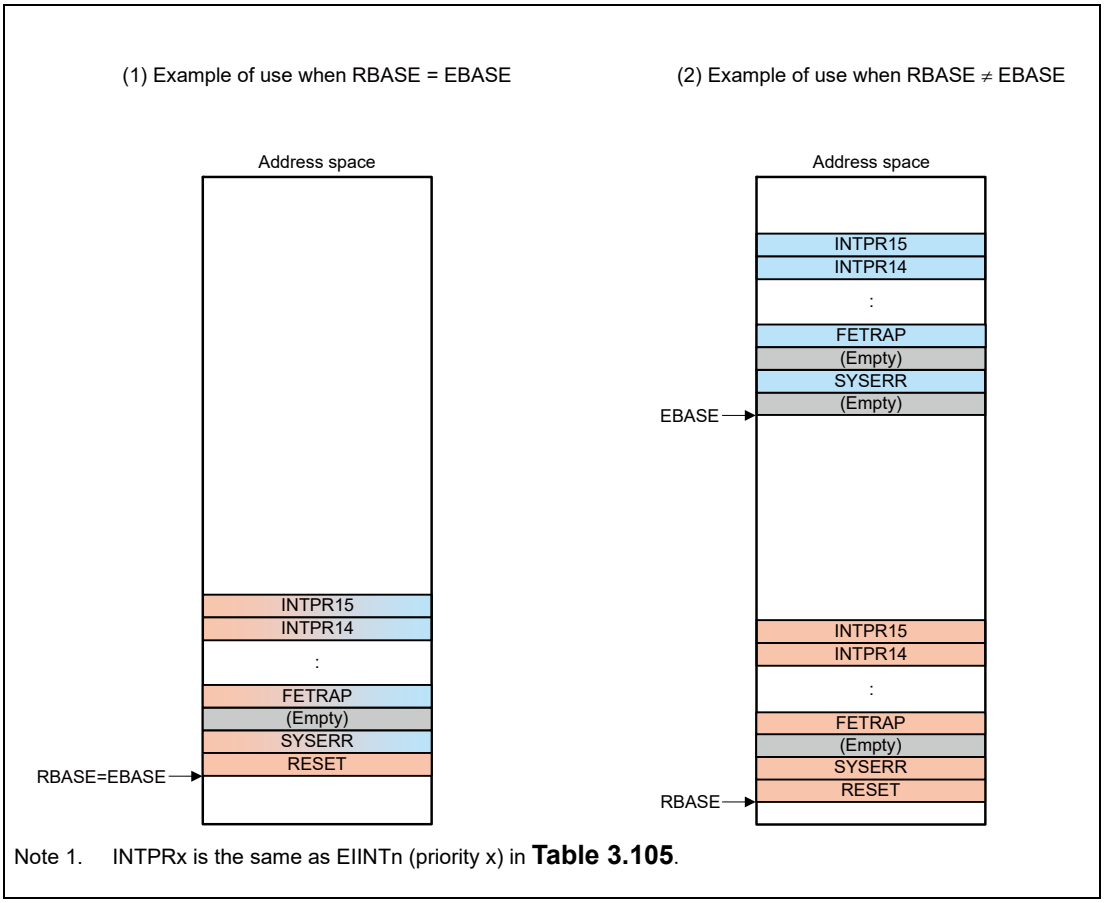
In addition, user interrupts (EIINTn) refer to the RINT bit of the selected base register, and reduce the offset address according to the value of the bit. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, all user interrupts are handled using an offset address of 100_H. If the bit is cleared to 0, the offset address is determined according to **Table 3.105**.

Figure 3.20 shows the flow of generating a handler address for the direct vector method.



Note 1. Report of the acceptance to the interrupt controller and the update of the EIPC, EIPSW, EIIC and ISPR are performed at this timing.

Figure 3.20 Flow of Generating a Handler Address of the Direct Vector Method



Note 1. INTPRx is the same as EIINTn (priority x) in **Table 3.105**.

Figure 3.21 Direct Vector Method

The table below shows how base register is selected and offset address for each exception. The value of the PSW register used to determine the exception handler address is the value after being updated due to the acknowledgment of an exception.

Table 3.105 Selection of Base Register/Offset Address

	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
	Base Register		Offset Address	
RESET	RBASE	None[*1]	000_H	000_H
SYSERR		EBASE	010_H	010_H
(R.F.U.)			020_H	020_H
FETRAP			030_H	030_H
TRAP0			040_H	040_H
TRAP1			050_H	050_H
RIE			060_H	060_H
FPE/FXE			070_H	070_H
UCPOP			080_H	080_H
MIP/MDP			090_H	090_H
PIE			$0A0_H$	$0A0_H$
(R.F.U.)			$0B0_H$	$0B0_H$
MAE			$0C0_H$	$0C0_H$
(R.F.U.)			$0D0_H$	$0D0_H$
FENMI			$0E0_H$	$0E0_H$
FEINT			$0F0_H$	$0F0_H$
EIINTn (priority 0)			100_H	100_H
EIINTn (priority 1)			110_H	
EIINTn (priority 2)			120_H	
EIINTn (priority 3)			130_H	
EIINTn (priority 4)			140_H	
EIINTn (priority 5)			150_H	
EIINTn (priority 6)			160_H	
EIINTn (priority 7)			170_H	
EIINTn (priority 8)			180_H	
EIINTn (priority 9)			190_H	
EIINTn (priority 10)			$1A0_H$	
EIINTn (priority 11)			$1B0_H$	
EIINTn (priority 12)			$1C0_H$	
EIINTn (priority 13)			$1D0_H$	
EIINTn (priority 14)			$1E0_H$	
EIINTn (priority 15 or less[*2])			$1F0_H$	

Note 1. An exception generated to update EBV to 0.

Note 2. Interrupt (EIINTn) which priority level is less than equal 16 uses the same offset address with the interrupt (EIINTn) with priority level 15. Because the same interrupt handler is used, software process is necessary for recognition of interrupt factor. By this constraint, the increasing of memory area which is used for interrupt handler by direct vector method can be avoided even if interrupt priority level extension function is enabled.

The user interrupt offset address reduction function is used to reduce the memory size required by the exception handler for specific operating modes of the system. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

(b) Table Reference Method

In the direct vector method, there is one user-interrupt exception handler for each interrupt priority, and user-interrupts with the same priority branch to the same interrupt handler, but some users might want to use code areas that differ from the start time for each interrupt handler.

This CPU defines a table reference method to accommodate to such uses.

Figure 3.22 shows the flow of generating a handler address for the table reference method.

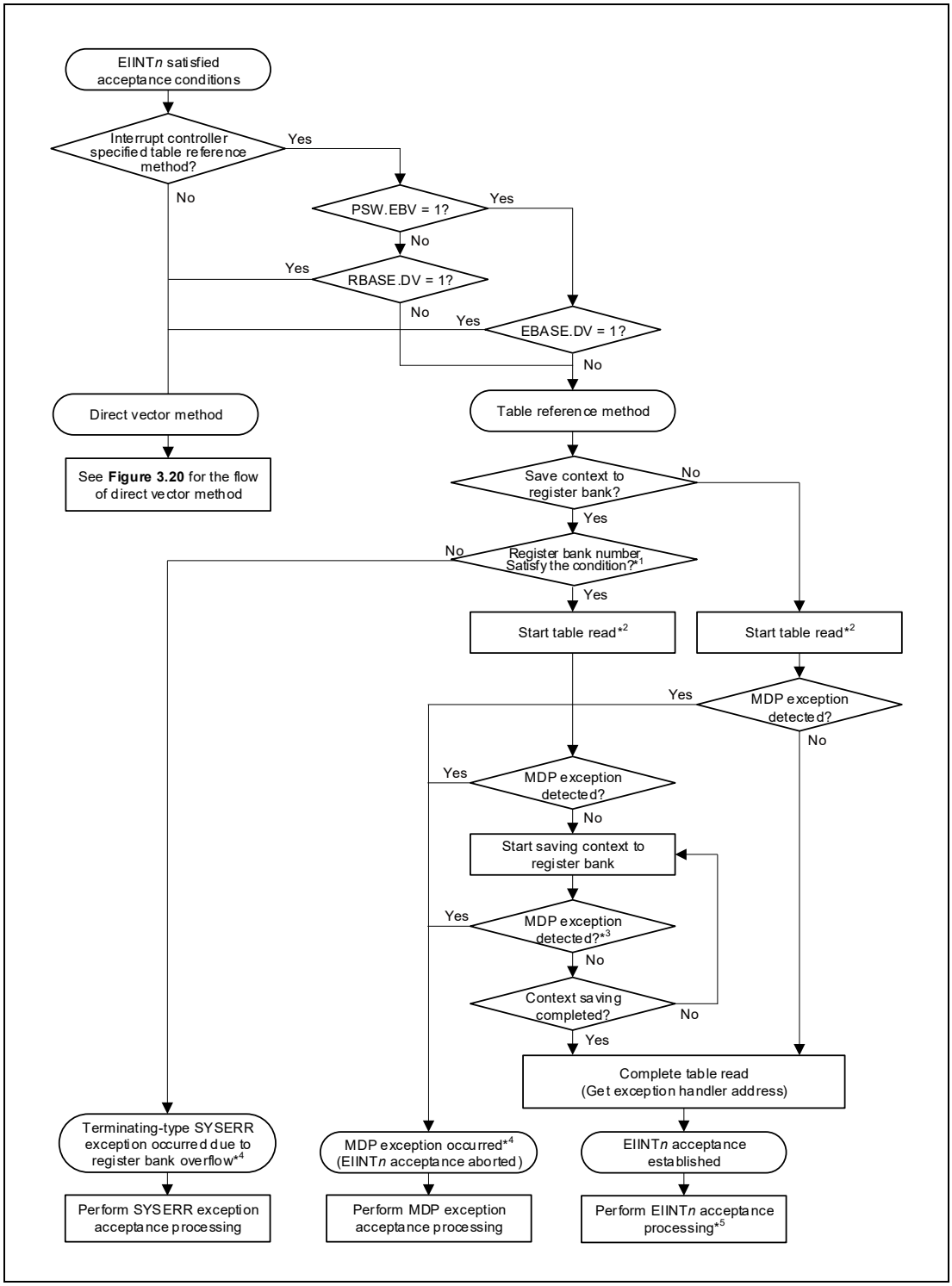


Figure 3.22 Flow of Generating a Handler Address of Table Reference Method (1/2)

Note 1. For automatic context saving onto the register bank, the operation condition judgment is necessary. For details, see **Figure 3.24**.

Note 2. This CPU updates the EIPC and EIPSW at this timing.

Note 3. Detection of an MDP exception is performed every time an individual context is saved. An MDP exception occurs when an MDP violation is detected when saving an individual context rather than after all context saving is completed.

Note 4. The PC of the instruction interrupted by the interrupt and the value of the PSW at that time are saved to the FEPC and the FEPSW respectively. These values are the same as the values saved to the EIPC and the EIPSW at the timing of Note 2. Therefore, when returning from the exception handler, it is possible to return exactly to the interrupted instruction. In addition, because no acceptance response has been reported to the interrupt controller, unless canceling the interrupt request in the exception handler, the interrupt request will remain held and interrupt acceptance processing will start again.

Note 5. Report of the acceptance to the interrupt controller and the update of the EIIC and ISPR are performed at this timing.

Figure 3.22 Flow of Generating a Handler Address of Table Reference Method (2/2)

<1> In any of the following cases, the exception handler address is determined by using the direct vector method.

- When the interrupt channel setting is not the table reference method
- When PSW.EBV = 0 and RBASE.DV = 1
- When PSW.EBV = 1 and EBASE.DV = 1

<2> In cases other than <1>, calculate the table read position.
Exception handler address read position = INTBP register + channel number × 4 bytes

<3> Read word data starting at the exception handler address read position calculated in <2>.

<4> Use the word data read in <3> as the exception handler address.

<5> The acceptance of the interrupt is established when the exception handler address has been got. In addition to the exception acceptance processing shown in **Figure 3.18**, the CPU returns an acceptance response to the interrupt controller and updates the ISPR.

CAUTIONS

1. **For details about the interrupt channel settings, see Section 6, Interrupts.**
2. **There are cases in which a memory protection exception (MDP) occurs while reading word data from the exception handler address read position depending on the memory protection settings. In such a case, the acceptance of the interrupt is temporarily cancelled. Since no acceptance response is returned to the interrupt controller, the interrupt request is held pending and is become acceptable again when execution is returned from the memory protection exception processing.**
3. **If a memory protection exception occurs during a word data read from the exception handler address read position, the PC of the instruction that is interrupted by the interrupt is saved in the FEPC. Since the acceptance of the interrupt is cancelled, the value of the PSW established when the interrupt occurred (which should have been saved in the EIPSW) is saved in the FEPSW.**

Exception handler address read positions corresponding to interrupt channels and the placement of the interrupt handler address table in memory are shown below.

Table 3.106 Exception Handler Address Read Position

Type	Exception Handler Address Read Position
EIINT interrupt channel 0	INTBP + 0 × 4
EIINT interrupt channel 1	INTBP + 1 × 4
:	:
EIINT interrupt channel 2046	INTBP + 2046 × 4
EIINT interrupt channel 2047	INTBP + 2047 × 4

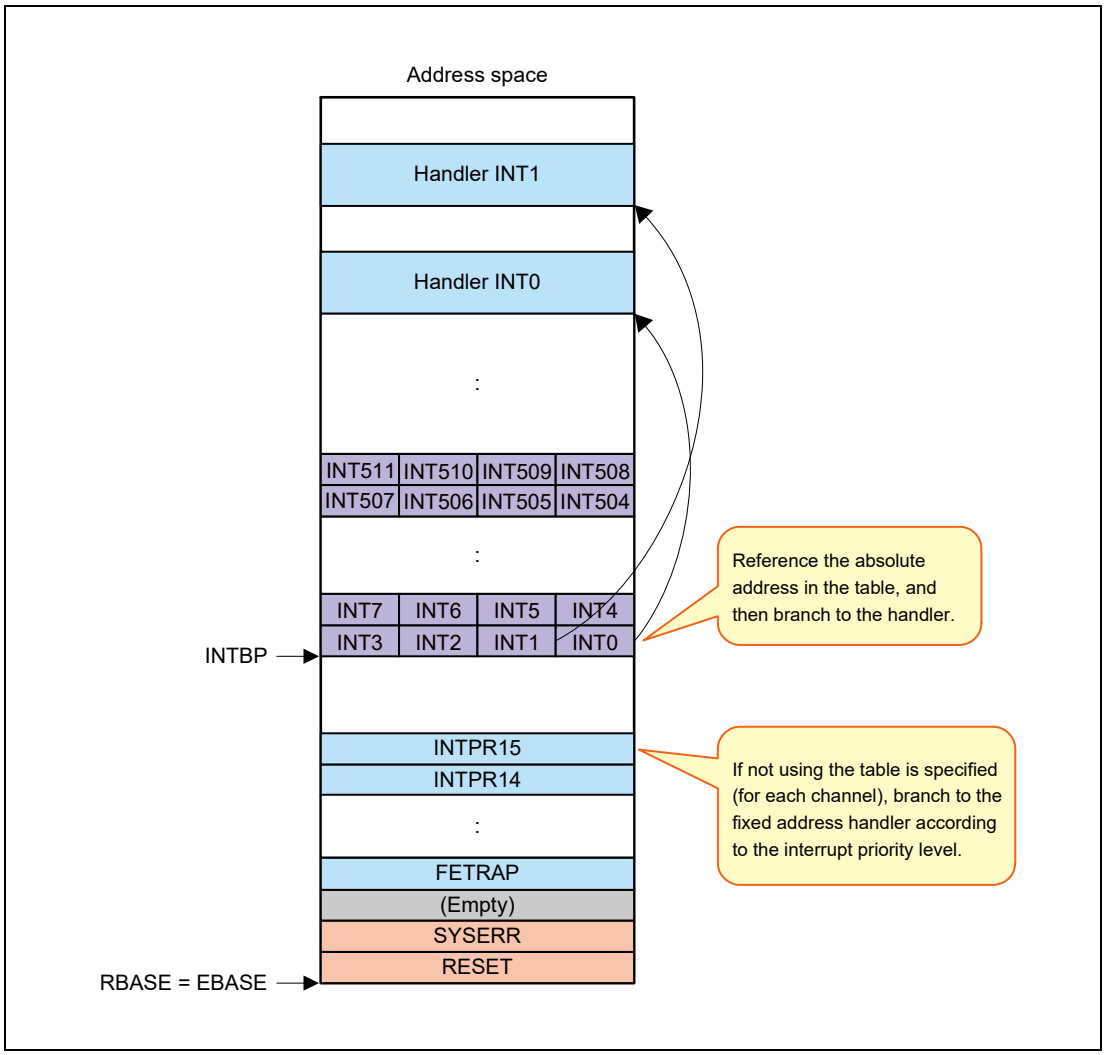


Figure 3.23 Placement of the Interrupt Handler Address Table in Memory

For details about the exception handler address selection method settings for each interrupt channel, see **Section 6, Interrupts**.

(2) System Calls

For system call exceptions, the referenced table entry is selected according to the value of the vector specified by the SYSCALL instruction and the value of the SCCFG.SIZE bit, and the exception handler address is calculated according to the contents of the table entry and the SCBP register value.

As an example, if table size n is specified by SCCFG.SIZE, the table entry is selected as shown below. Note that if the vector specified by the SYSCALL instruction (vector 8) is greater than table size n, the table entry referenced by vector n + 1 to 255 is table entry 0.

Table 3.107 System Calls

Vector	Exception Cause Code	Referenced Table Entry
0	$0000\ 8000_H$	Table entry 0
1	$0000\ 8001_H$	Table entry 1
2	$0000\ 8002_H$	Table entry 2
(Omitted)	:	:
n − 1	$0000\ 8000_H + (n - 1)_H$	Table entry n − 1
n	$0000\ 8000_H + n_H$	Table entry n
n + 1	$0000\ 8000_H + (n + 1)_H$	Table entry 0
(Omitted)	:	:
254	$0000\ 80FE_H$	Table entry 0
255	$0000\ 80FF_H$	Table entry 0

CAUTION

Because table entry 0 is selected if a vector that exceeds the table size specified by SCCFG.SIZE is specified, allocate the error processing routine at table entry 0.

3.2.4.5 Register Bank Function

This CPU provides the register bank function that automatically saves the context when it accepts an interrupts (EIINTn). Since the saving of the context proceeds in parallel with the interrupt acceptance processing, this function enables the CPU to make high-speed interrupt response.

(1) Outline of the Register Bank Function

The register bank function that this CPU provides has the following features:

- Automatically saves the context upon acceptance of an interrupt (EIINTn) meeting some conditions (see **Section 3.2.4.5 (2), Automatic Context Saving**).
- The destination of the context saving is not dedicated memory but an area of ordinary memory installed in this CPU (specified by the RBIP register).
- The context to be saved can be selected from two groups (specified by the RBCR0.MD bit).
- Supports a maximum of 64 levels of multiple interrupts (the RBNR.BN bits indicate the number of accepted interrupts).
- The context is restored from the register bank by executing the RESBANK instruction (no automatic restoration function).

(2) Automatic Context Saving

Automatic context saving is carried out when the following conditions are satisfied for an interrupt request (EIINTn) notified:

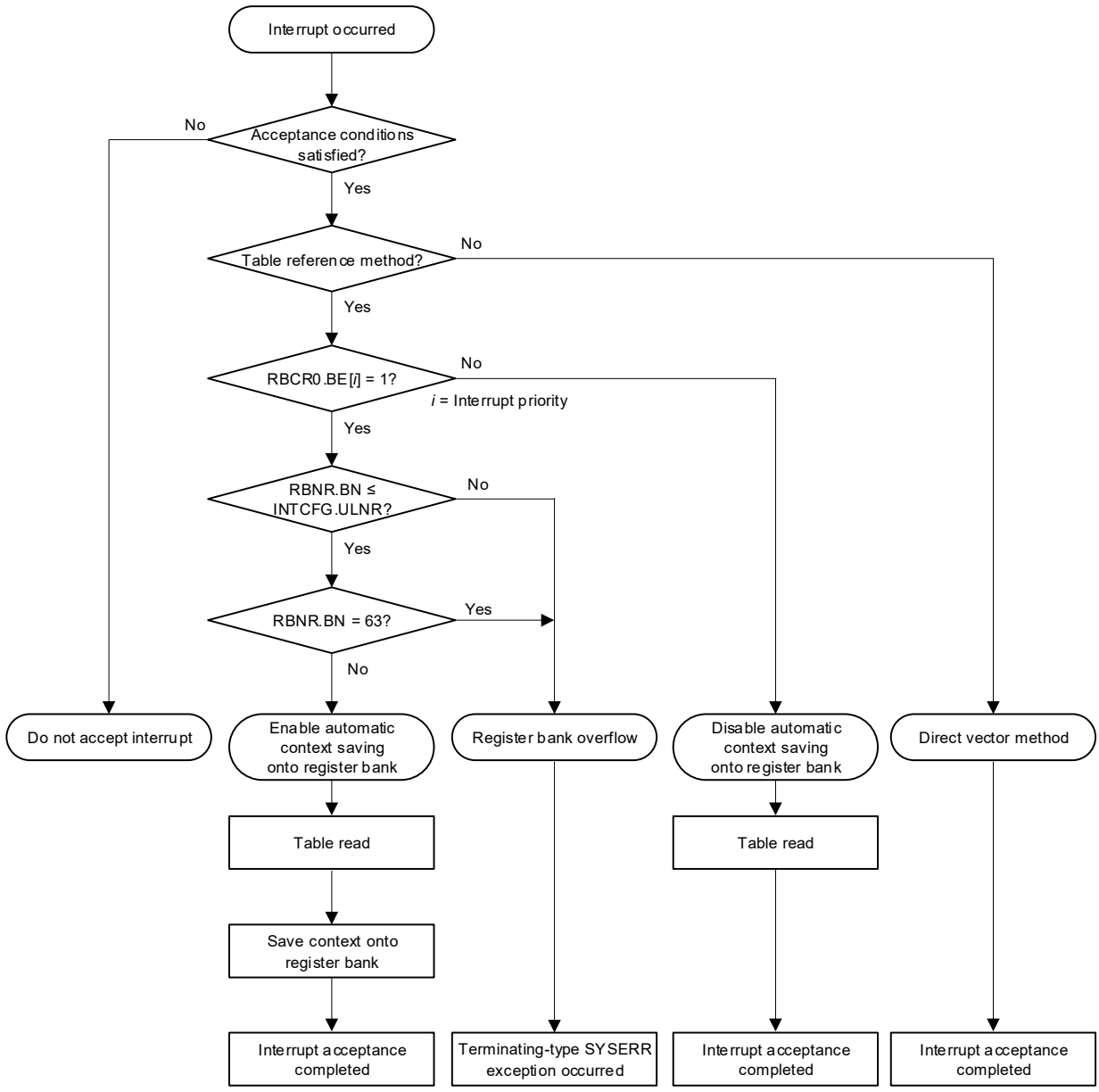
- Table reference method is specified for the interrupt request*1.
- The use of the register bank is specified by the RBCR0.BE[*i*] bit which is associated with the priority (*i*) of the interrupt request *2(the automatic context saving onto the register bank is enabled).
- The RBNR.BN bits have a value no greater than the value of INTCFG.ULNR*3 (check the number of the register banks in use).

Note 1. See **Section 6, Interrupts** for the procedure to specify the table reference method.
Note 2. Availability of context auto saving for all interrupt (EIINTn) which priority level is less than equal 16 is specified by RBCR0.BE[15].
Note 3. If the value of RBNR.BN is bigger than INTCFG.ULNR or the value of RBNR.BN is 63, the context auto saving is not done. In this case, SYSERR exception that exception factor code (lower 16 bits) is $1C_H$ occurs.

Automatic context saving is not carried out if the table reference method is not specified for the interrupt request (EIINTn) or the use of the register bank is not specified by the RBCR0.BE[*i*] bit associated with the interrupt priority (*i*).

In addition, no automatic context saving is carried out if the value of the RBNR.BN bits are greater than INTCFG.ULNR, or if the value of RBNR.BN is 63. In such a case, it is assumed that an unexpected interrupt acceptance processing is being performed and a terminating-type SYSERR exception is notified for error processing.

Figure 3.24 shows the flow of checking the conditions for automatic context saving.



Note 1. For details of operations other than the condition judgement, see **Figure 3.20** and **Figure 3.22**.

Figure 3.24 Flow of Checking the Conditions for Automatic Context Saving

(a) Automatic Context Saving

Shown below is the processing of automatic context saving onto the register bank.

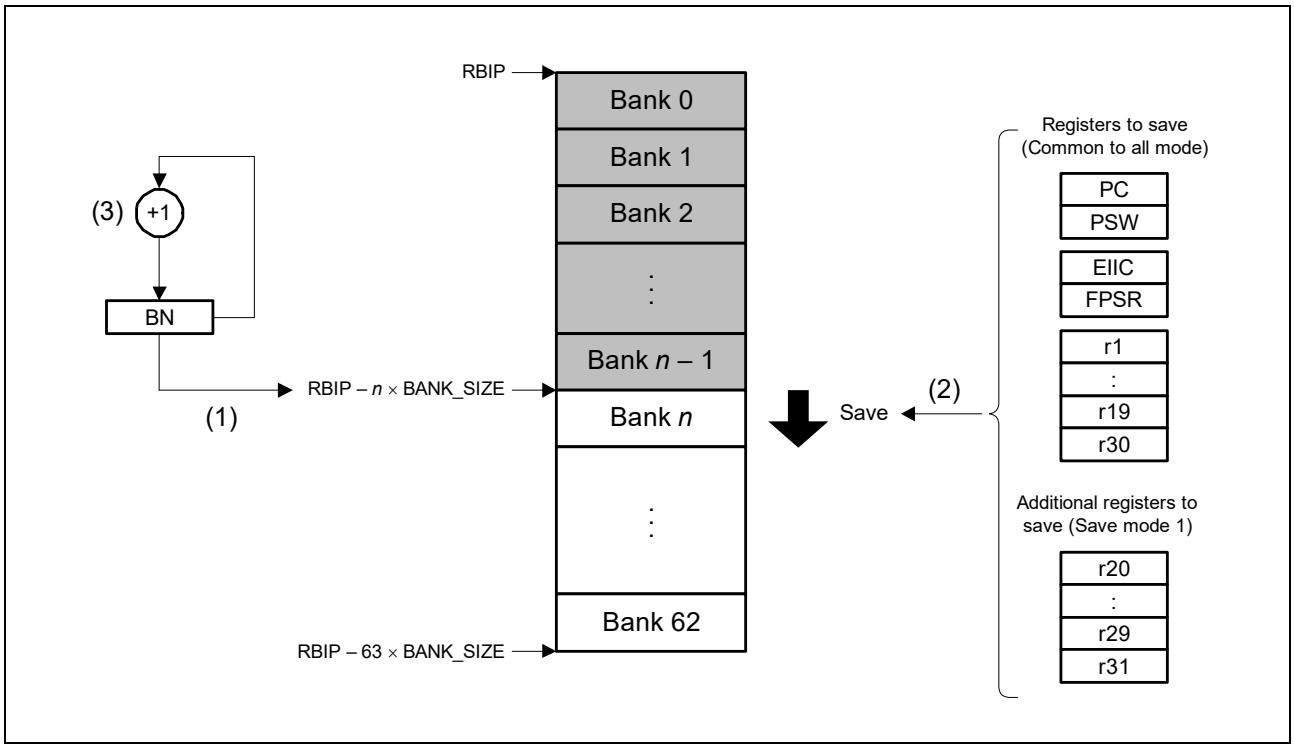


Figure 3.25 Processing of Automatic Context Saving

1. Calculate the start address of register bank n on which the context is to be saved based on the value of the register bank initial pointer (RBIP) and the value (*n*) of the BN bits of the register bank number register (RBNR) established upon acceptance of the interrupt. The size (BANK_SIZE) of the register bank differs depending on the save mode that is selected.
   Start address:
   RBIP – *n* × BANK_SIZE
   BANK_SIZE:
   Save mode 0: 60_H
   Save mode 1: 90_H
2. Save the target registers onto the register bank n specified by the RBNR.BN bits according to the save mode specified by the RBCR0.MD bit*1. **Table 3.108** shows a list of the registers to save, their addresses, and save order.
3. Increment the value of the RBNR.BN bits by 1. This terminates the register bank saving processing*2 and initiates the execution of the interrupt handler*3.

Note 1. MDP exceptions (terminating-type) might be caused by memory accesses made during the register bank saving processing. The other types of interrupts and exceptions cannot be accepted.

Note 2. The saving of the registers into the memory area specified as the register bank does not have always been completed. To ensure the completion of register saving, this CPU can employ a procedure similar to the one with the ordinary store instruction. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

Note 3. Exceptions of higher priorities can be accepted after the saving onto the register bank is finished.

Table 3.108 List of Registers to Save, Addresses, Save Order, and Restore Order

Addresses RBIP − n × BANK_SIZE	Save mode 0 (MD = 0)	Save mode 1 (MD = 1)	Save Order *1, *2	Restore Order *1, *3
−04H	PC	PC	1	24 / 35
−08H	PSW	PSW	2	23 / 34
−0CH	EIIC	EIIC	3	22 / 33
−10H	FPSR	FPSR	4	21 / 32
−14H	r1	r1	5	20 / 31
−18H	r2	r2	6	19 / 30
−1CH	r3	r3	7	18 / 29
−20H	r4	r4	8	17 / 28
−24H	r5	r5	9	16 / 27
−28H	r6	r6	10	15 / 26
−2CH	r7	r7	11	14 / 25
−30H	r8	r8	12	13 / 24
−34H	r9	r9	13	12 / 23
−38H	r10	r10	14	11 / 22
−3CH	r11	r11	15	10 / 21
−40H	r12	r12	16	9 / 20
−44H	r13	r13	17	8 / 19
−48H	r14	r14	18	7 / 18
−4CH	r15	r15	19	6 / 17
−50H	r16	r16	20	5 / 16
−54H	r17	r17	21	4 / 15
−58H	r18	r18	22	3 / 14
−5CH	r19	r19	23	2 / 13
−60H	r30	r20	24	1 / 12
−64H	*4	r21	25	11
−68H		r22	26	10
−6C H		r23	27	9
−70H		r24	28	8
−74H		r25	29	7
−78H		r26	30	6
−7CH		r27	31	5
−80H		r28	32	4
−84H		r29	33	3
−88H		r30	34	2
−8CH		r31	35	1
−90H		*5		
−94H		*6		

Note 1. Register saving and restoration start sequentially at sequence number 1.

Note 2. The target registers to be saved in save mode 0 (RBCR0.MD = 0) are the registers up to r30 with a save sequence number of 24.

Note 3. The restoration sequence numbers shown on the left side are for save mode 0 and those shown on the right side are for save mode 1.

Note 4. In save mode 0, the next bank (n + 1) starts from this address.

Note 5. In save mode 1, no register is saved in this address. This address is not used.

Note 6. In save mode 1, the next bank (n + 1) starts from this address.

(b) Suppressing the Update of the PSW.ID Bit

When an interrupt (EIINTn) is accepted in the normal state, the PSW.ID bit is set to 1 and interrupts of the same EI level cannot be accepted unless the acceptance of interrupts is enabled explicitly with the EI instruction.

For interrupts that make use of the register bank (satisfying the automatic context saving conditions), on the other hand, by clearing the RBCR1.NC[*i*] bit associated with the interrupt priority (*i*) to 0, the PSW.ID bit keeps its value 0 without being set to 1 when an interrupt is accepted. This makes it possible to accept interrupts of higher priorities without software intervention after the end of automatic context saving. Since necessary registers are automatically saved, the CPU can return to the original interrupt processing precisely.

NOTE

The PSW.ID bit of all interrupt (EIINTn) which priority level is less than equal 16 after interrupt acknowledgment is defined by RBCR1.NC[15].

(3) Context Restoration

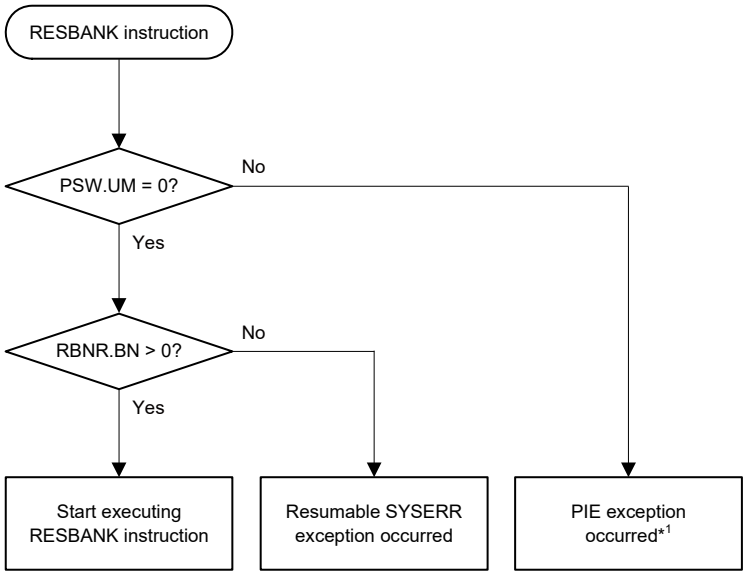
The context automatically saved onto the register bank need to be restored before returning from the interrupt processing to the original program. Although the context saving onto the register bank is automatically done, its restoration needs to be accomplished explicitly by software executing the RESBANK instruction.

(a) Conditions for Executing the RESBANK Instruction

The RESBANK instruction must be executed to restore the context from the register bank. The RESBANK instruction is a supervisor-privileged instruction. A PIE exception will occur if it is executed in user mode.

A resumable-type SYSERR exception will occur if the RESBANK instruction is executed when the value of the RBNR.BN bits are 0. A 0 in the RBNR.BN bits means that no context has automatically been saved onto the register bank. Accordingly, an invalid value will be restored if the RESBANK instruction is executed in such case.

The above-mentioned operating conditions are summarized in **Figure 3.26**.



Note 1. The process for detecting MIP and other exceptions that are not caused directly by the RESBANK instruction is omitted.

Figure 3.26 Flow of Checking the Conditions for Executing the RESBANK Instruction

(b) Context Restoration

The figure below shows the processing of the RESBANK instruction for restoring the context from the register bank.

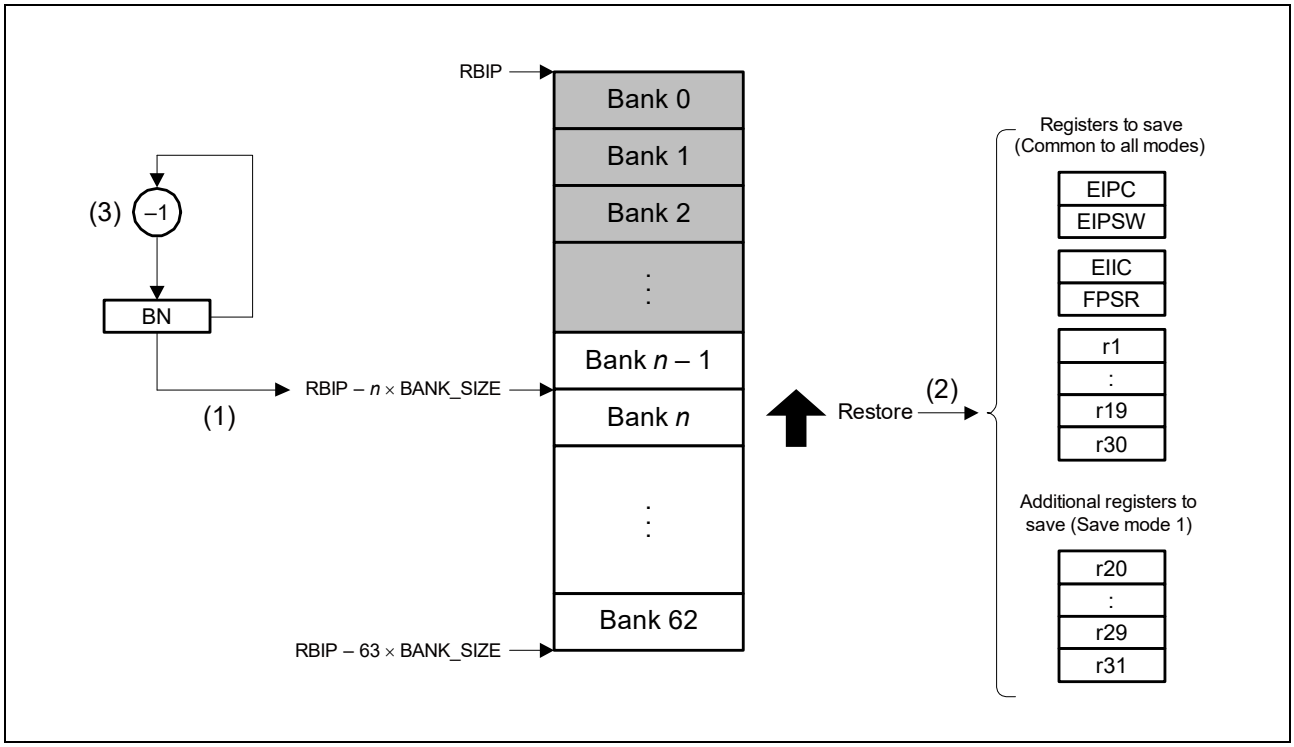


Figure 3.27 Processing of Context Restoration

1. Calculate the start address of register bank n-1 from which the context is to be restored based on the value of the register bank initial pointer (RBIP) and the value (n) of the BN bits of the register bank number register (RBNR) established upon acceptance of the interrupt. Unlike in the case of saving the context, the start address to be referenced when restoring the context is the lower-limit side of the register bank addresses. The size (BANK_SIZE) of the register bank differs depending on the save mode that is selected.
   Start address:
   RBIP – n × BANK_SIZE
   BANK_SIZE:
   Save mode 0: 60_H
   Save mode 1: 90_H
2. Restore the context from the register bank n-1 in the target registers according to the save mode specified by the RBCR0.MD bit*1. **Table 3.108** shows a list of the registers to restore, their addresses, and restore order.
3. Decrement the value of the RBNR.BN bits by 1*2. This terminates the register bank restoration processing and completes the execution of the RESBANK instruction*3.

Note 1. The CPU can accept terminating-type exceptions while restoring the context from the register bank. MDP exceptions (resumable-type) might be caused by memory accesses made during the register bank restoration processing.

Note 2. When the CPU accepts an exception before the value of RBNR.BN is updated, it stops the execution of the RESBANK instruction even if the restoration of all registers is not yet

completed. In this case, though there are some registers of which restoration has been completed, the CPU cannot know what registers have been restored. Since the return PC of the exception is the PC of this RESBANK instruction, the CPU can re-execute precisely the RESBANK instruction if none of the resources related to the RESBANK instruction are altered during the exception processing.

Note 3. After the RESBANK instruction is executed, execute the EIRET instruction to return from the interrupt handler.

3.2.4.6 List of Memory Access Exceptions

Table 3.109 shows a list of instructions and exceptions that make memory accesses and the exceptions that can be detected. Exceptions identified by the symbol ✓ can be detected during the memory accesses that are made during the processing of the listed instructions and exceptions. Exceptions identified by the symbol × are not detected.

Table 3.109 List of Memory Access Exceptions

Instruction/Exception	MAE	MDP	Instruction/Exception	MAE	MDP
SLD.B	×	✓	PREPARE	×	✓
SLD.BU	×	✓	DISPOSE	×	✓
SLD.H	✓	✓	PUSHSP	×	✓
SLD.HU	✓	✓	POPSP	×	✓
SLD.W	✓	✓	STM.MP	×	✓
SST.B	×	✓	LDM.MP	×	✓
SST.H	✓	✓	SWITCH	×	✓
SST.W	✓	✓	CALLT	×	✓
LD.B	×	✓	SYSCALL	×	✓
LD.BU	×	✓	LDV.W	✓	✓
LD.H	✓	✓	LDV.DW	✓	✓
LD.HU	✓	✓	LDV.QW	✓	✓
LD.W	✓	✓	STV.W	✓	✓
LD.DW	✓	✓	STV.DW	✓	✓
ST.B	×	✓	STV.QW	✓	✓
ST.H	✓	✓	LDVZ.H4	✓	✓
ST.W	✓	✓	STVZ.H4	✓	✓
ST.DW	✓	✓	CACHE (CHBII)	×	✓
LDL.BU	×	✓	CACHE (CIBII)	×	×
LDL.HU	✓	✓	CACHE (CFALI)	×	✓
LDL.W	✓	✓	CACHE (CISTI)	×	×
STC.B	×	✓	CACHE (CILDI)	×	×
STC.H	✓	✓	CACHE (other commands)	×	×
STC.W	✓	✓	PREF (PREFI)	×	×*1
CAXI	✓	✓	PREF (other commands)	×	×
SET1	×	✓	Table reference type EIINTn	×	✓*2
CLR1	×	✓	Register bank saving processing	×	✓*2
NOT1	×	✓	RESBANK	×	✓
TST1	×	✓			

Note 1. When a violation is detected during the execution of the PREF instruction, no MDP exception is generated but the memory access is suppressed. Read access is not performed.

Note 2. Both of table read accesses due to table reference method interrupts and write accesses for automatically saving the context onto the register bank occur independently of the execution of an instruction. This CPU handles any MDP exception that is caused by these accesses as a terminating-type exception. The exception cause code of this MDP exception differs from that of resumable-type MDP exceptions. For details see **Table 3.102**.

3.2.5 Memory Management

This CPU provides the following functions for managing memory.

- Memory protection unit (MPU)
- Instruction cache

3.2.5.1 Memory Protection Unit (MPU)

Memory protection functions are provided in an MPU (memory protection unit) to maintain a smooth system by detecting and preventing unauthorized use of system resources by unreliable programs, runaway events, etc.

(1) Features

(a) Memory Access Control

Multiple protection areas can be assigned to the address space. Consequently, unauthorized program execution or data manipulation by user programs can be detected and prevented. The upper and lower limit addresses of each area can be specified so that the address space can be used precisely and efficiently.

(b) Access Management for Each CPU Operation Mode

In this CPU, several status bits are used to control access to resources, and these bits are used in combination to perform protection that is appropriate, according to each program's level of reliability.

(c) Protection with the System Protection Identifier (SPID)

This CPU can use the system protection identifier (SPID) to check for area matching. The usage and limitations on settings of the SPID register depend on the system specifications of the product in which this CPU is mounted. Reflecting such system specifications in the control of memory access can enable the efficient management of access.

(2) Protection Area Settings

(a) Protection Area Attribute Settings

Set the respective protection areas appropriately. For details about registers, see **Section 3.2.3, Register Set**.

E bit

This sets the target protection area setup as enabled or disabled. When disabled, all settings are disabled. Make sure valid setting values have been stored for other protection area settings (MPUA, MPLA, and MPAT) at the time when this bit is set to 1.

UX, UR, and UW bits

These bits indicate the access privileges for the target protection area during user mode. A 1 in these bits indicates the presence of the corresponding access privilege.

SX, SR, and SW bits

These bits indicate the access privileges for the target protection area during supervisor mode. These bits, in addition to the E bit, are valid only when the MPM.SVP bit has been set to 1. If the MPM.SVP

bit has been cleared to 0, even when the E bit is set to 1, protection is not performed while in supervisor mode, regardless of the values of the SX, SR, and SW bits, and the entire address space becomes access-enabled.

WG and WMPIDn bits and MPIDn registers (n = 0 to 7)

These bits indicate how to reference the system protection identifier (SPID) during the check for the permission to write to a protection area.

When the WG bit is set (to 1), the access permission for all write accesses is judged only by the settings of the UW or SW bit regardless of the value of the SPID that is set.

When the WG bit is cleared (to 0), if the value of the SPID register matches any of the values specified in the MPIDn registers and if the WMPIDn bit corresponding to the matching MPIDn register is set (to 1), the access permission is judged according to the settings of the UW or SW bit. If the value of the SPID register matches none of the values specified in the MPIDn registers or if the WMPIDn bit corresponding to the matching MPIDn register is cleared (to 0), the access permission is judged to be “write-disabled” regardless of the settings of the UW or SW bit.

In addition, when the values of multiple MPIDn registers match the value of the SPID register, writing is possible if any of the corresponding WMPIDn bit is set (to 1).

RG bit, RMPIDn bit, and MPIDn registers (n = 0 to 7)

These bits indicate how to reference the system protection identifier (SPID) during the check for the permission to read from a protection area.

When the RG bit is set (to 1), the access permission for all read accesses is judged only by the settings of the UX and UR bits or the SX and SR bits regardless of the value of the SPID that is set. When the RG bit is cleared (to 0), if the value of the SPID register matches any of the values specified in the MPIDn registers and if the RMPIDn bit corresponding to the matching MPIDn register is set (to 1), the access permission is judged according to the settings of the UX and UR bits or the SX and SR bits. If the value of the SPID register matches none of the values specified in the MPIDn registers or if the RMPIDn bit corresponding to the matching MPIDn register is cleared (to 0), the access permission is judged to be “read-disabled” regardless of the settings of the UX and UR bits or the SX and SR bits.

In addition, when the values of multiple MPIDn registers match the value of the SPID register, reading is possible if any of the corresponding RMPIDn bit is set (to 1).

(3) Caution Points for Protection Area Setup

(a) Crossing Protection Area Boundaries

When the specified protection areas overlap, the access control settings for the overlapping parts are set to “allow most”.

In other words, when multiple protection areas have been specified, if access is enabled for either of the protection areas, access is judged to be enabled.

(b) Invalid Protection Area Settings

Protection area settings are invalid in the following case.

When value set to lower-limit address is larger than value set to upper-limit address

CAUTION

Addresses are handled as unsigned integers (0_H to FFFF $FFFF_H$).

For example, the following settings alone cannot make the target area accessible (MPAT is omitted):

MPLA0 = FFFF $FF80_H$, MPUA0 = 0000 $00FC_H$

This setting should be divided into the following two protection area settings.

MPLA0 = FFFF $FF80_H$, MPUA0 = FFFF $FFFC_H$

MPLA1 = 0000 0000_H, MPUA1 = 0000 $00FC_H$

MPLA0 refers to the MPLA register for the protection area 0. The same holds true for the other registers.

(c) Lower- and Upper-Limit Addresses Referenced during Protection Violation Checks

Table 3.110 lists the lower-limit addresses that are compared with MPLA and the upper-limit addresses that are compared with MPUA during the protection violation check.

Table 3.110 Lower- and Upper-limit Addresses Referenced during the Protection Violation Check

Instruction/Event	Access size	Lower-Limit Address	Upper-Limit Address
SLD, SST, LD, ST, LDL, STC, LDV, STV	Byte	Address calculated according to the addressing specified in the instruction	Same as lower-limit address
	Half word		Lower-limit address + 1
	Word		Lower-limit address + 3
	Double word*1		Lower-limit address + 7
	Quad word*1		Lower-limit address + 15
CAXI	Word	Value of reg1	Lower-limit address + 3
SET1, NOT1, CLR1, TST1	Byte	Address calculated according to the addressing specified in the instruction	Same as lower-limit address
PREPARE, DISPOSE, PUSHSP, POPSP, STM, LDM	Word*2	Address calculated from the value of the SP and the number of accesses	Lower-limit address + 3
SWITCH, CALLT	Half word	Address calculated according to the addressing specified in the instruction	Lower-limit address + 1
SYSCALL	Word	Address calculated according to the default addressing	Lower-limit address + 3
CACHE (CHBII, CFALI), PREF	Cache line size*3	reg1 value rounded by cache line size	Lower-limit address + Cache line size – 1
Table reference EIINTn	Word	Address from which the exception handler address is read when table reference method is selected	Lower-limit address + 3
Automatic context saving onto a register bank	Word*2	Address calculated from the values of the RBIP and the RBNR.BN bit	Lower-limit address + 3
RESBANK	Word*2	Address calculated from the values of the RBIP and the RBNR.BN bit	Lower-limit address + 3
Memory protection configuration check function	Value of MCS	Value of MCA	Value of MCA + MCS – 1
Instruction fetch	Instruction fetch size*4	PC rounded by the instruction fetch size	Lower-limit address + Instruction fetch size – 1

Note 1. Since the minimum protection area unit is word-size, there are cases in which the lower- or upper-limit address specified for a double or quad word access does not fall within the address range designated by a single protection area setting. If either one of the lower- and upper-limit addresses does not fall within the specified address range, a protection violation is detected and an MDP exception is generated. See **Section 3.2.5.1(3)(d), Memory Access Spanning Contiguous Protection Areas,** for the operation for two contiguous protection area settings.

Note 2. The instruction accesses contiguous multi-byte area, however, memory accesses for a single register are carried out on word-size basis. Memory protection is effected on each of these word-size accesses. For this reason, there are cases in which a data protection violation is detected when some of word-size memory access are executed and there still are memory accesses that are yet to be executed. In these cases, an MDP exception occurs during the processing of the instruction, and the instruction is aborted. Memory or registers are updated by the completed processing of the instruction.

Note 3. For the cache line size, see **Section 3.9.9, Product Information of Cache Structure**.

Note 4. For the instruction fetch size, see **Section 3.9.10, Product Information of Fetch Size**.

(d) Memory Access Spanning Contiguous Protection Areas

With respect to the memory protection against an operand access, it is necessary that the associated memory access is encompassed entirely in a single access enabled area. No accesses that span over two or more areas are allowed even when the access enabled areas are allocated in contiguous spaces. In this CPU, since the minimum protection area unit is word-size and misaligned accesses cannot be executed, memory accesses spanning over areas can occur during double word size memory accesses (LD.DW, ST.DW, LDV.DW, and STV.DW) or quad-word memory access (LDV.QW, STV.QW).

On the other hand, the memory accesses associated with the instructions PREPARE, DISPOSE, PUSHSP, POPSP, STM.MP, LDM.MP, and RESBANK and the memory accesses associated with the automatic context saving onto a register bank may span over two or more access enabled areas since they are handled as repetitions of two or more word size accesses.

With respect to the memory protection against an instruction fetch, if a fetch access spans over protection areas, the instruction codes that are fetched are given the result of judging the memory protection which is made with the word size as a delimiter. Since only a specific part of the instruction codes that are fetched is required to execute the instruction, even if an instruction fetch which spans over enabled and disabled areas is made, the instruction code in the enabled area can be executed. If the instruction code in the disabled area is used, an MIP exception will occur.

Since the instruction codes are allocated on a half word basis, for an instruction whose instruction length is the word size or longer, there are cases in which the instruction itself spans over protection areas. In such a case, the instruction may be executed if both areas are enabled. If one area is disabled, however, the instruction cannot be executed even if the other area is enabled.

Table 3.111 below shows the examples of operand accesses which span over protection areas. **Table 3.112** below shows the examples of instruction fetches which span over protection areas. In the following examples, the LD.DW, DISPOSE instructions and the instruction fetch start to access address 0000 00FC$_H$ and proceed in the direction of increasing addresses. The PREPARE instruction starts to access address 0000 0100$_H$ and proceeds in the direction of decreasing addresses. The memory protection setting check function checks the area that begins from 0000 00FC$_H$ and goes beyond the boundary of the protection area. In the examples, following two protection areas are set.

Protection area setting 0: MPLA0 = 0000 0000$_H$, MPUA0 = 0000 00FC$_H$
Protection area setting 1: MPLA1 = 0000 0100$_H$, MPUA1 = 0000 01FC$_H$

Table 3.111 Examples of Operand Memory Accesses Spanning Over Contiguous Protection Areas

Protection Setting		Operation			
Setting 0	Setting 1	LD.DW, etc.	PREPARE, etc. (save context)	DISPOSE, etc. (restore context)	Memory Protection Setting Check
Disabled	Disabled	MDP	MDP	MDP	Prohibited (0)*5
Disabled	Enabled	MDP	MDP*1	MDP*3	Prohibited (0)*5
Enabled	Disabled	MDP	MDP*2	MDP*4	Prohibited (0)*5
Enabled	Enabled	MDP	Access executed	Access executed	Prohibited (0)*5

Note 1. Accesses made to addresses in the protection area setting 1 are enabled and update the memory. When an access is made to an address in the protection area setting 0, an MDP exception occurs and the execution of the instruction is aborted. The MEA register holds the address at which the first data protection violation is detected.

Note 2. Since the address at which accessing is to start is disabled, an MDP exception occurs without the execution of any memory access.

Note 3. Since the address at which accessing is to start is disabled, an MDP exception occurs without the execution of any memory access.

Note 4. Accesses made to addresses in the protection area setting 0 are enabled and update the registers. When an access is made to an address in the protection area setting 1, an MDP exception occurs and the execution of the instruction is aborted. The MEA register holds the address at which the first data protection violation is detected.

Note 5. The memory protection setting check function sets “Prohibited (0)” to the corresponding bits of the MCR register unless the target area falls within a single protection area.

Table 3.112 Examples of Instruction Fetch Memory Accesses Spanning Over Contiguous Protection Areas

Protection Setting		Operation			
Setting 0	Setting 1	Execute Instructions Covered by Only Setting 0	Execute Instructions Covered by Only Setting 1	Execute Instructions Spanning Over Settings 0 and 1	Manipulate Instruction Cache with CACHE Instruction/Prefetching by the PREF Instruction
Disabled	Disabled	MIP	MIP	MIP	MDP*1/Disabled*2
Disabled	Enabled	MIP	Instruction executed	MIP	MDP*1/Disabled*2
Enabled	Disabled	Instruction executed	MIP	MIP	MDP*1/Disabled*2
Enabled	Enabled	Instruction executed	Instruction executed	Instruction executed	MDP*1/Disabled*2

Note 1. An MDP exception occurs when a memory protection violation is detected for the CACHE instruction which manipulates the instruction cache using the address specification method.

Note 2. An MDP exception does not occur, and prefetching is disabled when a PREF instruction leads to a violation of memory protection.

(e) Memory Access which Spans Over Address 0000 0000_H

Table 3.113 summarizes the operations that are performed when a memory access which spans over addresses FFFF $FFFF_H$ to 0000 0000_H is made with the protection setting such that all the address spaces are enabled by the following single protection area setting:

Protection area setting 0: MPLA0 = 0000 0000_H, MPUA0 = FFFF $FFFC_H$

Table 3.113 Operations of Memory Access That Span Over Address 0000 0000_H

Instruction/Event	Operation
LD.DW, ST.DW	MDP exception
PREPARE, DISPOSE, PUSHSP, POPSP, RESBANK, Automatic context saving onto a register bank	Access executed

For instructions and events other than those mentioned above, no memory access which spans over address 0000 0000_H is generated by a single instruction or event.

(4) Access Control

In this CPU, accesses are controlled appropriately according to the settings described in **Section 3.2.5.1 (2), Protection Area Settings**. In any of the cases listed below, the CPU ensures logical integrity by limiting actual access, detecting violations before instruction execution is completed, and setting up exceptions.

- When about to execute an instruction at an address outside the executable area
- When about to execute an instruction that reads from an address outside the read-accessible area
- When about to execute an instruction that writes to an address outside the write-accessible area

The specifics of access control are as follows.

- The result of the access which is judged as prohibited is not reflected in memory or I/O devices.
- The result of the access which is judged as enabled is reflected in memory or I/O devices.

CAUTIONS

1. Even when access is enabled, there might be cases where access is blocked by another function that prohibits it.

2. In some cases, access judged to be prohibited may be executed for a memory. The cases are as listed below.

- **Reading local RAM**
- **Reading of code flash memory by instruction prefetching of the instruction cache**

Since execution of the instructions that read from the local RAM or execute the prefetched instruction is inhibited by an exception, such access does not affect the execution of instructions. However, when a debugger is monitoring access to the local RAM or code flash memory, it may observe access judged to be prohibited.

(5) Violations and Exceptions

In this CPU, violations are detected during instruction fetch access or operand access according to the protection area settings, and an exception is generated.

- Execution protection violation (during instruction fetch access)
- Data protection violation (during operand access)

(a) Execution Protection Violation (MIP Exception)

This violation is detected when an instruction is executed. An execution protection violation is detected when attempting to execute an instruction that has been placed in a non-executable area.

When an execution protection violation is detected, an MIP exception always occurs.

(b) Data Protection Violation (MDP Exception)

This violation is detected during operand access. A data protection violation is detected when a memory access instruction, etc. attempts to access data from an access-prohibited area.

When a data protection violation is detected, an MDP exception always occurs.

(c) Exception Cause Code and Exception Address

When an execution protection violation or data protection violation has been detected, the exception cause code is determined as shown in **Table 3.114**. The determined exception cause code is set to the FEIC register.

The MEA register is used to store either the PC of the instruction that detected the execution protection violation or the access address used when the data protection violation occurred. The MEA register is shared by MIP and MDP exceptions since these exceptions do not occur simultaneously. Also, when a data protection violation occurs, the information of the instruction or event that caused the violation is stored in the MEI register.

Table 3.114 Exception Cause Code of Memory Protection Violation

Exception	Operation Mode When Violation Occurred	Bit Number and Bit Name										
		31-25	24	23	22	21	20	19	18	17	16	15-0
		—	MS	BL	RMW	SX	SW	SR	UX	UW	UR	—
MIP	User mode	0	0	0	0	0	0	0	1	0	0	90_H
	Supervisor mode	0	0	0	0	1	0	0	0	0	0	90_H
MDP	User mode	0	*5	*4	*3	0	0	0	0	*2	*1	91_H
	Supervisor mode	0				0	*2	*1	0	0	0	91_H*6

Note: UR: A violation is detected during a read operation in user mode (PSW.UM = 1).
UW: A violation is detected during a write operation in user mode (PSW.UM = 1).
UX: A violation is detected during instruction execution in user mode (PSW.UM = 1).
SR: A violation is detected during a read operation in supervisor mode (PSW.UM = 0).
SW: A violation is detected during a write operation in supervisor mode (PSW.UM = 0).
SX: A violation is detected during instruction execution in supervisor mode (PSW.UM = 0).
RMW: Set to 1 when the instruction causing the violation contains a read-modify-write operation (SET1, NOT1, CLR1,or CAXI).
BL: Set to 1 when the instruction causing the violation performs a block transfer (PREPARE, DISPOSE, PUSHSP, or POPSP).
MS: Set to 1 when the instruction causing the violation performs a misaligned access.

Note 1. When a read violation is caused by an instruction that includes a read operation, either the SR or UR bit is set to 1.

Note 2. When a write violation is caused by an instruction that includes a write operation, either the SW or UW bit is set to 1.

Note 3. This bit is set to 1 when a violation is caused by the SET1, NOT1, CLR1, or CAXI instruction.

Note 4. This bit is set to 1 when a violation is caused by the PREPARE, DISPOSE, PUSHSP, or POPSP instruction.

Note 5. This bit is set to 1 when the instruction causing the violation performs a misaligned access.

Note 6. These bits are loaded with 95_H for a data protection violation that is detected during a table read initiated by table reference EIINTn or automatic context saving onto a register bank. These are executed only in supervisor mode.

(6) Memory Protection Setting Check Function

For the programs, such as OS, that provide services, this CPU provides a memory protection setting check function to enable implementation of a service protection function that checks in advance whether or not the address area to be used for the requested operations is within an area that is accessible by the source that requested the service. By using this function, when verifying the validity of the user-supplied parameters for a system service, the OS can complete the verification in a shorter time than by repeating reading and checking the area settings by software.

(a) Check Details

The memory protection setting check function checks what access permission settings are made for the specified address area according to the protection area settings that are stored in the MPU.

If the address area specified by the MCA and MCS registers is contained within one protection area (specified by a pair of the MPLA and MPUA registers) and the instruction execution (instruction fetch)/write access/read access for the SPID specified by the MCI register is permitted in supervisor mode/user mode, the corresponding bit in the MCR register is set (to 1).

If the specified address area is not contained within one protection area or an access for the specified SPID is not permitted in an operation mode, the corresponding bit in the MCR register is cleared (to 0).

This check operation is realized by executing the same memory protection judgement procedure that is applied to the operand access or instruction fetch access to the specified address area and SPID. Therefore, when the checked address area is accessed by a load instruction, store instruction or instruction fetch, the access is judged as permitted or a memory protection exception occurs according to the check result.

CAUTIONS

1. **For the memory protection setting check function to judge the access as permitted, it is necessary that the specified address area is contained within one protection area. For example, when two contiguous protection areas are set as access permitted (see Section 3.2.5.1(3)(d), Memory Access Spanning Contiguous Protection Areas), the memory protection setting check result for the address area spanning over the contiguous protection areas becomes “access prohibited (0)”. Therefore, if there are such memory protection settings, there may be a difference between the actual access result by PREPARE instruction, instruction fetch, etc. and the result of memory protection setting check function.**
2. **The memory protection check function makes the same judgement as the memory protection judgement that is applied to the actual memory accesses. For example, if the MPM.MPE bit is cleared (to 0), all check results are always set to “access permitted (1)”. If the MPM.SVP bit is cleared (to 0), the bits MCR.SXE, SWE, and SRE are always set to “access permitted (1)”. Or if the MPM.MPE bit is set (to 1) (MPM.SVP is assumed to be cleared) and the MPAT.E bit of all protection area settings is cleared (to 0), the MCR.UXE, UWE, and URE bits are always set to “access prohibited (0)”.**

(b) Checking Procedure

Set the base address (lower limit) of the target address area to the MCA register, the size of the target area to the MCS register, and the system protection identifier (SPID) of the source requesting the service to the MCI register, then use the LDSR instruction (r0 specification is recommended) to access the MCC register and execute a check. The results can be read from the MCR register by the STSR instruction.

CAUTION

If the specified area to be checked crosses 0000 0000$_H$ or 7FFF FFFF$_H$, it is judged as an area setting error, and the MCR.OV bit is set to 1. This means that the MCR.OV bit must be checked before accessing the check results. Do not use the check result until it is confirmed that the result is not invalid (OV = 0).

(c) Sample Code

It is assumed that the memory protection setting check function will be used for the following operations.

```
_service_protection:
        …
        MOV         ADDRESS, r10        // Store the start address of the area to be checked to r10
        MOV         SIZE, r11           // Store the size of the area to be checked to r11
        MOV         SPID, r12           // Store the system protection identifier to r12
        DI
        LDSR        r10, sr8, 5         // Set the address to MCA
        LDSR        r11, sr9, 5         // Set the size to MCS
        LDSR        r12, sr12, 5        // Set the system protection identifier to MCI
        LDSR        r0, sr10, 5         // Start checking with MCC
        STSR        sr11, r13, 5        // Get the results from MCR
        EI
        ANDI        0x0100, r13, r0
        BNZ         _overflow           // Processing of invalid input when OV = 1
        BR          _result_check       // Otherwise, result is checked
```

(d) Method of Calculating Address Areas

The address area to be checked is treated as follows and compared with the memory protection setting.

Lower-limit address of the area: MCA setting
Upper-limit address of the area: Value calculated from MCA + (MCS – 1)

In addition, when MCS is set to 0000 0000$_H$, the value of "MCS – 1" is treated as FFFF FFFF$_H$. Moreover, if the result of calculating an upper address exceeds FFFF FFFF$_H$, it will be calculated as a 33-bit value, but the most significant bit, which represents an overflow, is discarded.

When MCA is 0000 0000$_H$ and MCS is 0000 0000$_H$, the upper-limit address is FFFF FFFF$_H$.
When MCA is FFFF FFFF$_H$ and MCS is 0000 0002$_H$, the upper-limit address is 0000 0001$_H$.

(e) Checking Results of the Memory Protection Setting Checking Function

Table 3.115 lists the results of the memory protection setting checking function for various settings.

Table 3.115 List of Checking Results of Memory Protection Setting Checking Function

MPM Setting		Check Setting (MCA, MCS, MCI)		Operation in the Various Types of Protected Area					Final Result of Checking (in the MCR)*2		
				Protection Setting			Result of Checking for One Protected Area*1				
MPE	SVP	Address	MCI, MPIDn	E	Address	xMPIDn*3	SXE, SWE, SRE	UXE, UWE, URE	OV	SXE, SWE, SRE	UXE, UWE, URE
0	—	—	—	—	—	—	—	—	0	All bits are set to 1	All bits are set to 1
1	0	Spanning 0000 0000_H	—	—	—	—	—	—	1	All bits are set to 1	All bits are set to 0
		Spanning 7FFF $FFFF_H$	—	—	—	—	—	—	1	All bits are set to 1	All bits are set to 0
		Other than above	*4	*4	*4	*4	All bits are set to 1	*4	0	All bits are set to 1	*4
	1	Spanning 0000 0000_H	—	—	—	—	—	—	1	All bits are set to 0	All bits are set to 0
		Spanning 7FFF $FFFF_H$	—	—	—	—	—	—	1	All bits are set to 0	All bits are set to 0
		Other than above	Mismatch	—	—	—	All bits are set to 0	All bits are set to 0	0	All bits are set to 0	All bits are set to 0
			Match	0	—	—	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas
				1	Not included	—	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas
					—	Disabled	All bits are set to 0	All bits are set to 0	0	Results for each of the areas	Results for each of the areas
					Included	Enabled	Results corresponding to the setting	Results corresponding to the setting	0	Results for each of the areas	Results for each of the areas

Note 1. For the MPM and other settings, this indicates the result produced by a single protection-setting area, but it cannot be directly monitored.

Note 2. The values to be stored in each bit of the MCR are the overall results for all areas covered by the given protection setting. “Results for each of the areas” will be enabled if the result of checking for even one area is enabled. However, for the value of the OV bit or all bits being set to 1 or 0, the result is as defined regardless of the result of each protection setting area.

Note 3. This indicates the result of enabling when the WG or RG bit is set (to 1), or the collective results of enabling or disabling by the individual WMPIDn and RMPIDn bits.

Note 4. When the MPM.MPE bit is set (to 1) and the MPM.SVP bit is cleared (0), access to all memory in supervisor mode is enabled. On the other hand, the judgment in user mode is different. The results obtained in accord with the settings for checking and protection are stored in the same way as the results when both the MPM.MPE and MPM.SVP bits are set (to 1).

3.2.5.2 Cache

This section describes the function to control the cache memory that is installed in this CPU.

In normal operation, the cache memory is used implicitly in instruction fetching by the CPU. However, if the setting is to be changed or the state of the cache memory is directly manipulated by the CACHE instruction, consistency with the normal operation of CPU must be maintained. In many cases, the hardware makes adjustments so that inconsistencies between the normal operation of the CPU and the manipulation of the cache do not arise. However, to make sure that the results of cache manipulation are the intended state, take care that the manipulation does not affect the implicit operation.

For a detailed description of the cache memory that is installed in the individual products, see **Section 3.9.9, Product Information of Cache Structure**.

(1) Features

A 16-KB and parity-based way select (PBS) 4-way set associative instruction cache is placed between the CPU and the code flash. The instruction cache and the code flash are connected to each other via a 256-bit dedicated bus to minimize penalties caused by a cache miss. A data buffer is also mounted between the CPU and the code flash to achieve high-speed data access. The 256-MB area from 0000 0000_H to 0FFF $FFFF_H$ in the address space is intended for the instruction cache and data buffer.

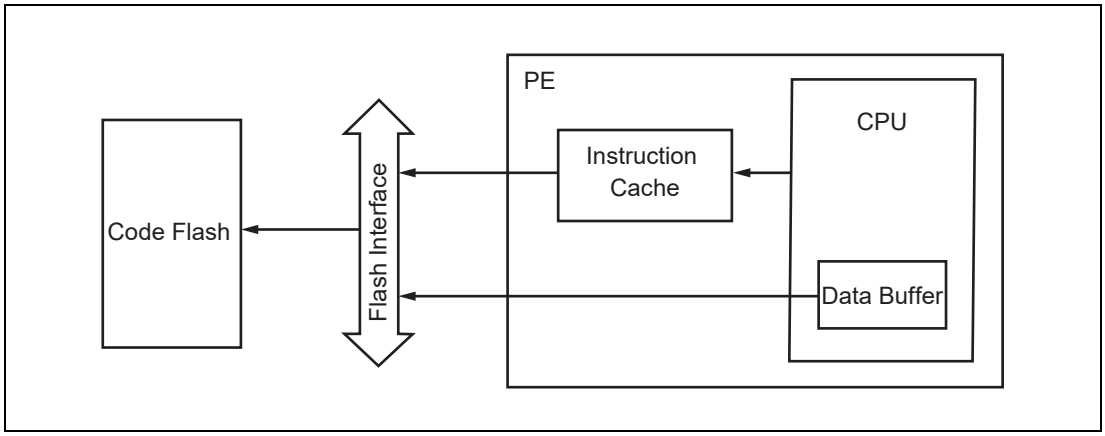


Figure 3.28 Instruction Cache and Data Buffer

(2) Cache Operation Registers

Figure 3.29 shows the system registers for cache operation. The supervisor privilege is required for the operation.

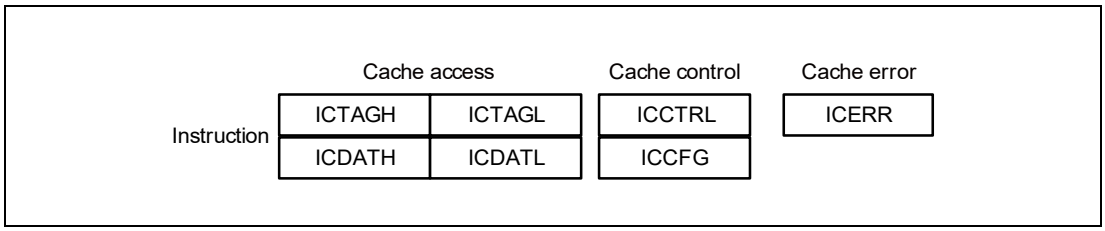


Figure 3.29 Cache Operation Registers

(3) Change Cache Use Mode

(a) Change Use Mode of Instruction Cache

The instruction cache use mode can be changed by using the ICCTRL.ICHEN bit. To enable an instruction cache, set the ICHEN bit to 1.

To disable the instruction cache, clear the ICHEN bit to 0.

Completion of the LDSR instruction that sets ICHEN might not coincide with completion of the instruction cache setting change. As in the following sample code, by executing the SYNCI instruction after the LDSR instruction, it is ensured that the change of the instruction cache setting takes effect after the SYNCI instruction.

```
LDSR r10, sr24, 4    // Change the instruction cache settings (ICCTRL)(setting value is stored in
                        r10)
SYNCI                // Wait for completion of the LDSR instruction, then refetch the subsequent
                        instructions.
```

(4) Cache Operations Using CACHE Instruction

The CACHE instruction manipulates the specified data in the cache memory.

Such data manipulation by the CACHE instruction starts after updating of the cache memory by all preceding memory access has been completed. Consequently, the result of preceding memory access is guaranteed to be the target for operations using the CACHE instruction. Additionally, a suitable synchronization operation is needed following execution of the CACHE instruction to ensure that the results are reflected in subsequent instructions.

The CACHE instruction can manipulate the cache memory even when the cache is disabled.

(a) Specification Method for Target of CACHE Instruction

There are basically two ways to specify the target for operations.

Directly specify the address to be accessed:
In this CPU, this is called the address specification method. In this case, the cache line containing the specified address is subject to operation.

Directly specify the cache memory's way number and line number:
In this CPU, this is called the index specification method. In this case, no hit judgment for the cache is performed, and the operation is performed on the specified cache index. For details about the cache index specification method, see **Section 3.2.5.2 (6), Cache Index Specification Method**.

(b) Operations Performed using the CACHE Instruction

The operations performed on the cache memory are divided into the following. For details about each operation, see the *RH850G4MH User's Manual: Software*.

Cache Hit Block Invalidate / Cache Indexed Block Invalidate (CHBI / CIBI)

This disables the specified cache line. When using the address specification method, the cache line is disabled only when there is a hit. When using the index method, specified cache line is disabled. If the specified cache line is locked, it is unlocked. This operation can be used in cases such as when the entire memory cache is initialized by software.

Disabling a cache line refers to the process of clearing the cache information corresponding to the ICTAGL.V and ICTAGL.L bits (to 0).

Cache Fetch and Lock (CFAL)

This stores the data at the specified address to the cache memory. At this time, the cache line where the data is stored is locked. This prevents the cache line from being replaced. If the target cache line has already been stored in the cache memory, it is simply locked. If the target cache line has already been stored in the cache memory and is locked, this operation does nothing.

This operation can be used to improve execution efficiency by reducing variations in instruction execution time that occur due to cache misses in the specified memory area.

Cache lines are locked only when they are enabled. Although enabling and locking of a cache line are carried out simultaneously during the execution of the CFAL instruction, no cache line is locked when only locking is configured using the CIST instruction. For details, see **Section 3.2.3.7 (1), ICTAGL — Instruction Cache Tag Lo Access**.

CAUTION

In the instruction cache supported by this CPU, if all the cache lines at the same index in the target way group are locked, the cache lines are not replaced. Care must be taken with respect to the cache lock specifications and the number of cache ways when monopolizing the cache memory efficiently using this operation. For the way group, see Section 3.2.5.2 (10), Configuration of Instruction Cache.
For details, see Section 3.9.9, Product Information of Cache Structure.

Cache Indexed Load / Cache Indexed Store (CILD / CIST)

This operation is used to directly access the cache memory. Values can be written and read, via a system register, at a position in the cache memory specified by using an index. Because cache data and cache tags can be accessed directly, this operation can be used for purposes such as software debugging.

(5) Cache Operation by the PREF Instruction

The PREF instruction is provided to realize efficient cache access by advising the CPU that an address is likely to be used in the near future. Getting the CPU to prefetch data into the cache memory before using the data can reduce the read wait time caused by a cache miss.

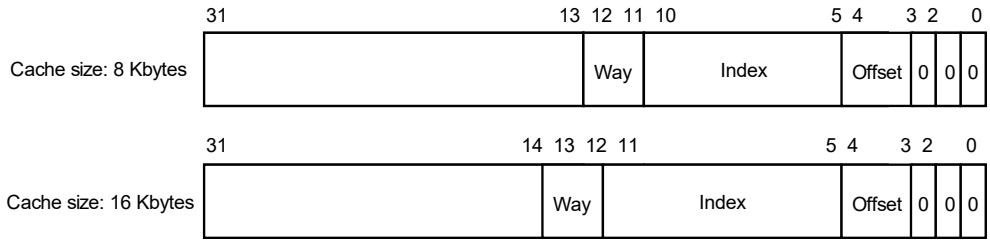
Assuming support by compilers and other tools, the PREF instruction can be executed regardless of the CPU operating mode. Execution of the PREF instruction does not cause an exception generated by the MPU, and has no effect on logical operations, just like the NOP instruction.

CAUTION

Because a data read request by the PREF instruction is rather speculative, it might not be executed depending on the system conditions. No cache fill is performed if all the cache lines at the same index in the target way group are locked. If an area outside the cacheable area is specified, the read itself is not carried out.

(6) Cache Index Specification Method

For a cache instruction that uses the index specification method, explicitly specify the cache memory subject to operation in the format shown in **Figure 3.30**, instead of specifying an address. The bit positions of each field depend on the size of the cache memory incorporated in the CPU core. Information about the incorporated cache memory and size can be read from the ICCFG register.



Note: Way: Specify the way within the cache.
Index: Specify the cache index.
Offset: Specify the offset within the cache line.

Note: Always set 0 to the bits other than the WAY, Index and Offset bits.

Figure 3.30 Cache Index Specification Method

NOTE

The Offset field indicates the byte position within the cache line. This setting is not required (i.e., ignored) in normal index specification operations. For a CILD/CIST operation, it is used to specify a position within the cache line because the ICDATH/ICDATL register is shorter than the cache line length.

(7) Execution Privilege of the CACHE/PREF Instruction

Because the CACHE instruction directly manipulates the contents of the cache memory, privileges are specified according to the type of operation. When the CACHE instruction is executed without the privilege required for the CACHE operation, a privilege instruction exception (PIE) occurs.

On the other hand, the PREF instruction provides information for speculative execution, so it can be executed in any mode.

The privileges required by the different operations performed by the CACHE instruction are shown below.

(a) Operations allowed with the user authority

Among address specification method operations, operations without a cache lock (CHBI) can be executed in any operation mode.

(b) Operations requiring the supervisor authority

Among address specification method operations, operations with a cache lock (CFAL) require the supervisor privilege.

In addition, index specification method operations also require the supervisor privilege.

(8) Memory Protection for the CACHE and PREF Instructions

The memory protection judgment for the CACHE and PREF instructions proceeds in accord with the operating mode when the instruction is executed. These instructions are handled as read access. When memory protection is enabled and the MPAT.SR or UR bit is set (to 1), the instruction can be executed according to the operating mode.

The CACHE instructions using the address specification mode are subject to the memory protection provided by the MPU. If a protection violation is detected, the cache operation is not executed and an MDP exception occurs.

On the other hand, since the CACHE instructions using the index specification method specify the position of data in the instruction cache but not the memory address, they are not subject to the memory protection. No matter what value is specified for the index, an MDP exception does not occur.

If a protection violation for a PREF instruction is detected, the prefetching is not performed and an MDP exception does not occur.

For both CACHE and PREF instructions, if operation is undefined for the specified opcode, cache operation and memory access are not performed at all, so they are not subject to the memory protection.

Table 3.116 shows the correspondence between operations and access permissions.

Table 3.116 Relationship between Cache Operations and Permissions

Instruction	Address/Index	Instruction Execution Privilege	Access Permission
CHBII	Address	UM	Read
CIBII	Index	SV	—
CFALI	Address	SV	Read
CISTI	Index	SV	—
CILDI	Index	SV	—
PREF	Address	UM	Read

(9) Example of Using the CACHE Instruction to Manipulate Cache Memory

This section gives an example of cache memory manipulation by the CACHE instruction, with setting up self-modifying code as the intention.

In the following example, an instruction code is generated in memory that is being cached in the instruction cache, and the instruction cache is disabled. Regarding whether memory that supports such cache memory manipulation is present, see **Section 3.9.9, Product Information of Cache Structure**.

```
ST.W r20, 0[r10]          //  Write the target data to memory
SYNCM                     //  Wait for the completion of memory writing
CACHE 0x00, [r10]         //  CHBII: Disable caching of the target area by the instruction cache
SYNCI                     //  Wait for the completion of instruction cache manipulation
JMP [r12]                 //  Brach to the target area and execute the updated instruction code
```

(10) Configuration of Instruction Cache

This section describes the configuration of instruction cache mounted for this CPU. For more information, see **Section 3.9.9, Product Information of Cache Structure**.

This CPU supports an instruction cache in the following configuration.

- Four ways
- 256-bit cache line size (unit of filling)
- 256-Mbyte target area for caching
- ECC protection
- Cache error notification

(a) Four ways

Since the instruction cache mounted for this CPU is configured with four ways, the capacity of each way is one-quarter of the capacity of the whole cache.

The four ways are divided into two way groups; way 0 and way 1 belong to way group A, and way 2 and way 3 belong to way group B. As for the access address, if the parity of the bits to be stored in the tag is 0, way group A is used, whereas way group B is used if the parity is 1.

Table 3.117 shows the addresses to generate parity for each cache capacity.

Table 3.117 Parity generation target address

Cache capacity	Parity generation target address
8 KB	Bit 24 to 11
16 KB	Bit 24 to 12
32 KB	Bit 24 to 13

When storing data in a new cache line, the LRU (least recently used) method determines the destination way for storage from within the given way group.

(b) Cache line size

The cache line size (unit of filling) is the amount of data that is stored in one line of the cache, and is 256 bits in the instruction cache for this CPU.

(c) Allocation of cache data

Data from memory are allocated to the cache memory in the way shown in **Figure 3.31**.

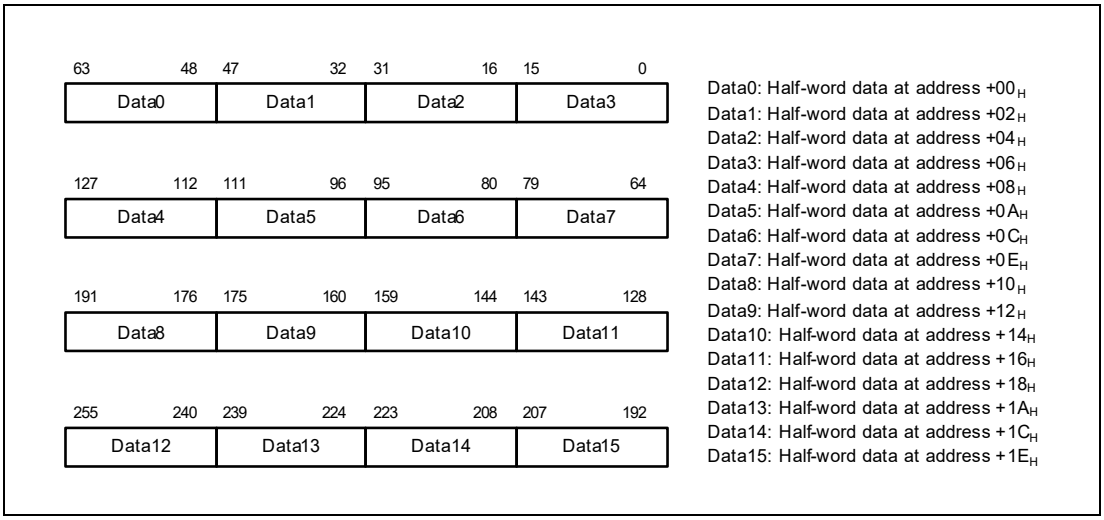


Figure 3.31 Allocation of Data to the Cache

(d) Cacheable area

The cacheable area for the instruction cache of this CPU is the 256 Mbytes from 0000 0000_H to 0FFF $FFFF_H$. Although instructions can be fetched from locations beyond the above area, they are never stored in the instruction cache in that case.

(e) ECC protection

The data RAM and the tag RAM of the instruction cache for this CPU are protected by ECCs. Upon detection of an ECC error in either area of RAM, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

For details on the ECC protection, refer to **Section 44, Functional Safety**.

(f) Cache error notification

The instruction cache for this CPU detects the following five types of errors. A cache error is detected at the time of read operation by the instruction fetching or the CACHE instruction using the address specification method. In addition to the read operation, an address-feedback error is also detected at the time of write operation by cache fill. This CPU provides external notification of these errors. For the operation in cases of these errors occurring, refer to **Section 44, Functional Safety** and **Section 3.9.12, Error Notifications**.

- ECC error of data RAM
- ECC error of tag RAM
- Multi-hit error
- Way error
- Address-feedback error

A multi-hit error is detected when access to an address hits two ways in a way group at the same time. Upon detection of a multi-hit error, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

A way error is detected when the parity in a way group does not match the parity of data stored in the tag RAM. Upon detection of a way error, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds.

An address-feedback error is detected when the address for access to the RAM does not match the feedback address. When the error is detected at the time of cache reading, the target line is disabled, and the CPU provides external notification of the occurrence of the error at the same time as re-fetching proceeds. When the error is detected at the time of cache writing, the target line is disabled, and the CPU provides external notification of the occurrence of the error.

(11) Data Buffer Function

The four-line buffer with 256 bits per line is mounted as a data buffer. 256-bit data read from the code flash is stored in the data buffer in 256-bit units. When the data buffer hit occurs, the data is read out from the data buffer, so the code flash is not accessed again.

3.2.6 Coprocessor

3.2.6.1 Floating-Point Operation

The floating-point unit (FPU) operates as the CPU coprocessor, and executes floating-point instructions.

Either single-precision (32-bit) or double-precision (64-bit) data can be used. In addition, the conversion between a floating point type and an integer type is possible.

The FPU of this CPU conforms to ANSI/IEEE standard 754-2008 (IEEE Standard for Floating-Point Arithmetic).

(1) Configuration of Floating-Point Operation Function

(a) Not implemented

If the floating-point operation function is not implemented, all the floating-point instructions cannot be used. If an attempt is made to execute such an instruction, a coprocessor unusable exception occurs. In addition, all the floating-point system registers become inaccessible, and if an attempt is made to access any of the registers using LDSR/STSR instruction, a coprocessor unusable exception occurs.

(b) Implementing only single precision

If only the floating-point operation function with single precision is implemented, only floating-point instructions classified as single precision*1 can be used. If an attempt is made to execute a floating-point instruction classified as double precision*2, a coprocessor unusable exception occurs. All the floating-point system registers supply the function described in **Section 3.2.3.4, FPU Function Registers**.

Note 1. The single-precision floating-point instruction is the instruction described as (Single) in the description of each instruction in the *RH850G4MH User's Manual: Software*.

Note 2. The double-precision floating-point instruction is the instruction described as (Double) in the description of each instruction in the *RH850G4MH User's Manual: Software*.

(c) Implementing single precision and double precision

All the floating-point instructions can be used when floating-point instructions of single precision and double precision are implemented. All the floating-point system registers supply the functions described in **Section 3.2.3.4, FPU Function Registers**.

(2) Data Types

(a) Floating-point format

The FPU supports 32-bit (single precision) and 64-bit (double precision) IEEE754 floating-point operations.

The single-precision floating-point format consists of a 24-bit signed fraction (s + f) and an 8-bit exponent (e), as shown in **Figure 3.32**.

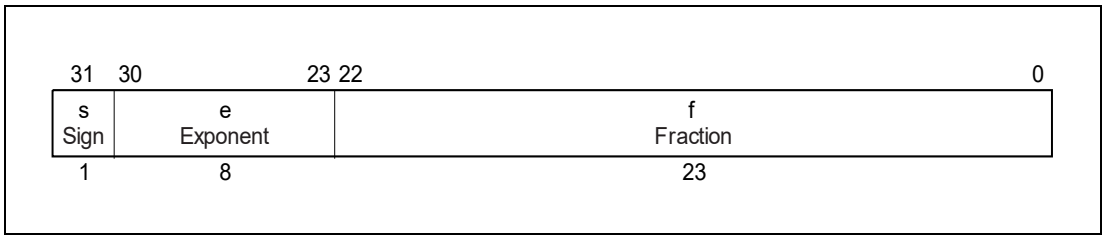


Figure 3.32 Single-Precision Floating-Point Format

The double-precision floating-point format consists of a 53-bit signed fraction (s + f) and an 11-bit exponent (e), as shown in **Figure 3.33**.

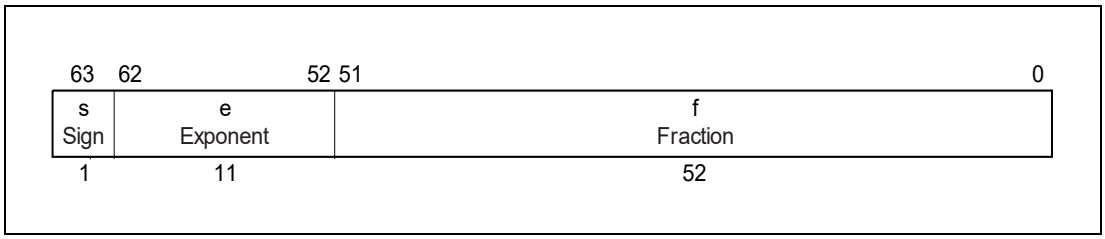


Figure 3.33 Double-precision Floating-Point Format

A numerical value in the floating-point format includes the following three areas.

- Sign bit s
- Exponent: e = E + bias value
- Fraction: $f = .b_1b_2...b_{P-1}$ (value lower than the first decimal place)

The bias value for the single-precision format is 127. For double-precision format, the bias value is 1023.

The range of the exponent value E when unbiased covers all integers from Emin to Emax, along with two reserved values, Emin –1 (±0 or subnormal number), and Emax +1 (±∞ or NaN: not-a-number). A numeric value other than 0 is represented in one format, depending on the single-precision and double-precision formats.

The numeric value (v) represented in this format can be calculated by the expression shown in **Table 3.118**.

Table 3.118 Calculation Expression of Floating-Point Value

Type	Calculation Expression	
NaN (not-a-number)	If E = Emax + 1 and f ≠ 0	then v = NaN regardless of s
±∞ (infinite number)	If E = Emax + 1 and f = 0	then v = $(-1)^s\infty$
Normalized number	If Emin ≤ E ≤ Emax	then v = $(-1)^s2^E$ (1.f)
Subnormal number	If E = Emin – 1 and f ≠ 0	then v = $(-1)^s2^{Emin}$ (0.f)
±0 (zero)	If E = Emin – 1 and f = 0	then v = $(-1)^s0$

- NaN (not-a-number)
  IEEE754 defines a floating-point value called NaN (not-a-number). Because this value is not a numerical value, it does not have any “greater than” or “less than” relationships to other values. If v is NaN in all of the floating-point formats, it might be either SignalingNaN (S-NaN) or QuietNaN (Q-NaN), depending on the value of the most significant bit of “f”. If the most significant bit of “f” is set, v is QuietNaN; if the most significant bit is cleared, it is SignalingNaN.

Table 3.119 shows the value of each parameter defined in floating-point formats.

Table 3.119 Floating-Point Formats and Parameter Values

Parameter	Format	
	Single Precision	Double Precision
Emax	+127	+1023
Emin	–126	–1022
Bias value of exponent	+127	+1023
Length of exponent (number of bits)	8	11
Integer bits	Cannot be seen	Cannot be seen
Length of fraction (number of bits)	23	52
Length of format (number of bits)	32	64

Table 3.120 shows the minimum and maximum values that can be represented in floating-point formats.

Table 3.120 Floating-Point Minimum and Maximum Values

Type	Value
Minimum value of single-precision floating point	1.40129846e – 45
Minimum value of single-precision floating point (normal)	1.17549435e – 38
Maximum value of single-precision floating point	3.40282347e + 38
Minimum value of double-precision floating point	4.9406564584124654e – 324
Minimum value of double-precision floating point (normal)	2.2250738585072014e – 308
Maximum value of double-precision floating point	1.7976931348623157e + 308

(b) Fixed-point formats

The value of a fixed point is held in the format of 2's complement. **Figure 3.34** shows a 32-bit fixed-point format and **Figure 3.35** shows a 64-bit fixed-point format. No signed bits exist in the unsigned fixed-point format, and all bits represent the integer value.

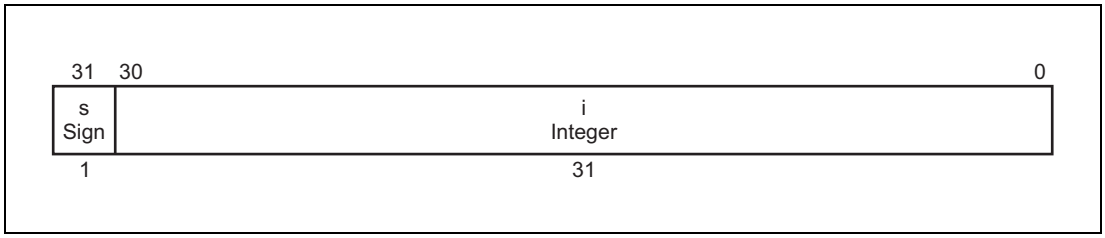


Figure 3.34 32-bit Fixed-Point Format

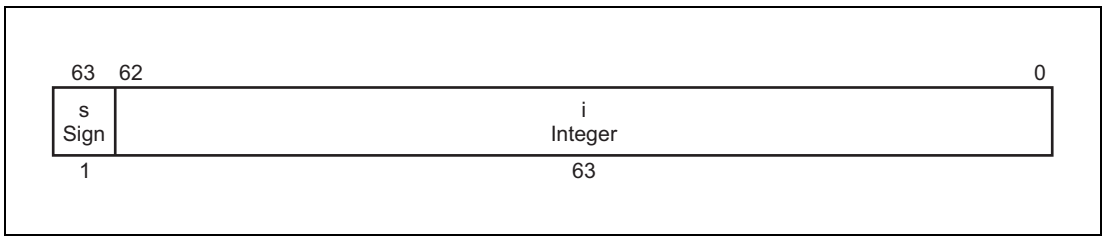


Figure 3.35 64-bit Fixed-Point Format

(c) Expanded floating-point format

This CPU supports the 16-bit (half-precision) IEEE754 floating-point format as a floating-point format for storing data. The half-precision floating-point format is used to decrease the amount of data; it is not supported for arithmetic operations. Instructions are available for converting single-precision floating-point format data into half-precision floating-point data and vice-versa. The half-precision floating-point format consists of an 11-bit signed fraction (s + f) and a 5-bit exponent (e), as shown in **Figure 3.36**.

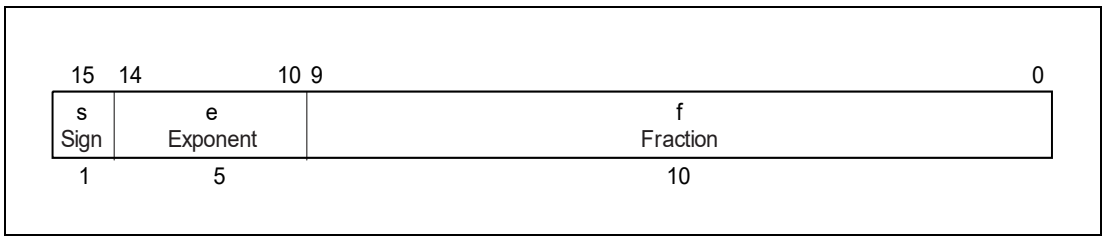


Figure 3.36 Half-Precision Floating-Point Format

Like other floating-point formats, the numeric values represented in this format can be calculated by using the expressions shown in **Table 3.118**. The values of the parameters defined by the half-precision floating-point format are shown in **Table 3.121**.

Table 3.121 Half-Precision Floating-Point Format and Parameter Values

Parameter	Half Precision
Emax	+15
Emin	–14
Bias value of exponent	+15
Length of exponent (number of bits)	5
Integer bits	Cannot be seen
Length of fraction (number of bits)	10
Length of format (number of bits)	16

Table 3.122 shows the minimum and maximum values that can be represented in the half-precision floating-point format.

Table 3.122 Half-Precision Floating-Point Minimum and Maximum Values

Type	Value
Minimum value of half-precision floating point	5.96046e – 8
Maximum value of half-precision floating point (normal)	6.10352e – 5
Maximum value of half-precision floating point	65504

(3) Register Set

For details about the register set, see **Section 3.2.3.4, FPU Function Registers**.

(4) Floating-Point Instructions

Floating-point instructions are divided into single-precision instructions (single) and double-precision instructions (double).

For details about the floating-point instructions, see the *RH850G4MH User's Manual: Software*.

(5) Floating-Point Operation Exceptions

This section describes how the FPU processes floating-point operation exceptions.

(a) Types of exceptions

When floating-point operations or processing of operation results cannot be done using the ordinary method, a floating-point operation exception occurs.

One of the following two operations is performed when a floating-point operation exception has occurred.

- When exceptions are enabled
  The cause bit is set in the floating-point configuration/status register (FPSR), and processing (by software) is passed to the exception handler routine.
- When exceptions are prohibited
  The preservation bit is set in the floating-point configuration/status register (FPSR), an appropriate value (initial value) is stored in the FPU destination register, then execution is continued.

The FPU uses cause bits, enable bits, and preservation bits (status flags) to support the following five types of IEEE754-defined exception causes.

- Inexact operation (I)
- Overflow (O)
- Underflow (U)
- Division by zero (Z)
- Invalid operation (V)

The sixth type of exception cause is unimplemented operation (E), which causes an exception when a floating-point operation cannot be executed. This exception requires processing by software. An unimplemented operation exception (E) occurs when exceptions are always enabled, rather than by using properties, enable bits, or preservation bits.

Figure 3.37 shows the FPSR register bits that are used to support exceptions.

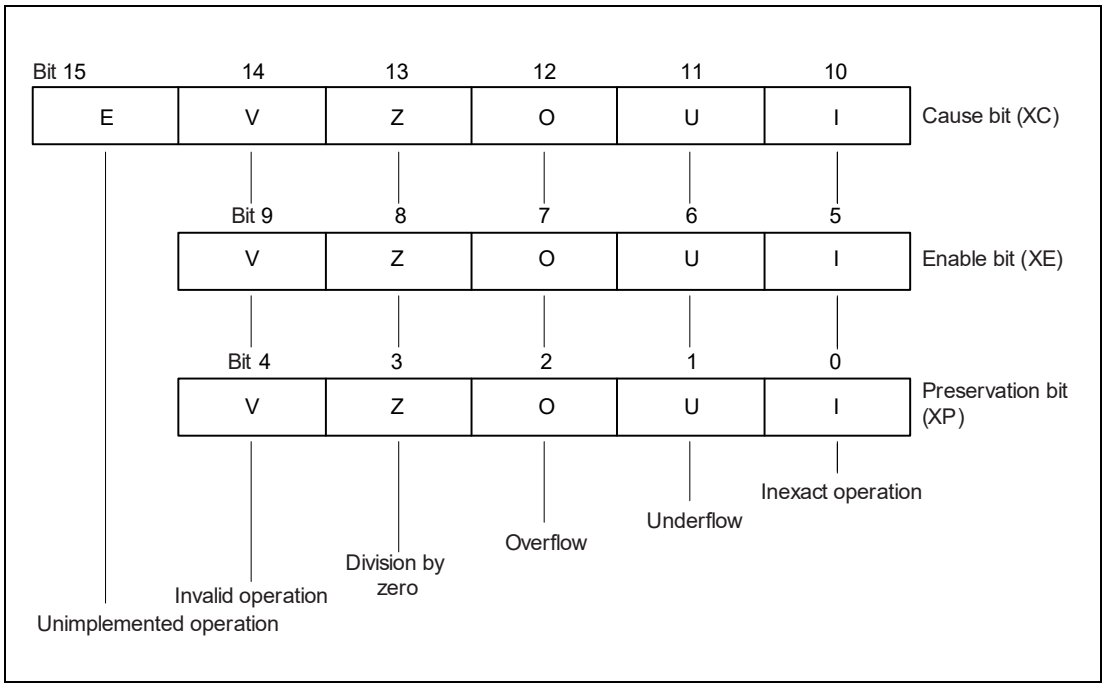


Figure 3.37 Cause, Enable, and Preservation Bits of FPSR Register

The five exceptions (V, Z, O, U, and I) defined by IEEE754 are enabled when the corresponding enable bits are set. When an exception occurs, if the corresponding enable bit has been set, the FPU sets the corresponding cause bit. If the exception can be acknowledged, processing is passed to the exception handler routine. If exceptions are prohibited, the exception corresponding preservation bit is set, and processing is not passed to the exception handler routine.

(b) Exception handling

When a floating-point operation exception occurs, the cause bits of the FPSR register indicate the cause of the floating-point operation exception.

Status flag

A corresponding preservation bit is available for each IEEE754-defined exception. The preservation bit is set when the corresponding exception is prohibited but the exception condition has been detected. The preservation bit is set or reset whenever new values are written to the FPSR register by the LDSR instruction.

If an exception is prohibited by an enable bit, predetermined processing is performed by the FPU. This processing provides an initial value as the result, rather than a floating-point operation result. This initial value is determined according to the type of exception. For an overflow exception or underflow exception, the initial value also differs depending on the current rounding mode. **Table 3.123** shows the initial values provided for each of the FPU IEEE754-defined exceptions.

Table 3.123 FPU Initial Values for IEEE754-Defined Exceptions

Area	Description	Rounding Mode	Initial Value
V	Invalid operation	—	Quiet not-a-number (Q-NaN)
Z	Division by zero	—	Correctly signed ∞
O	Overflow	RN	∞ with sign of intermediate result
		RZ	Maximum normalized number with sign of intermediate result
		RP	Negative overflow: Maximum negative normalized number Positive overflow: +∞
		RM	Positive overflow: Maximum positive normalized number Negative overflow: –∞
U	Underflow*1	RN*2	0 with sign of intermediate result
		RZ	0 with sign of intermediate result
		RP	Positive underflow: Minimum positive normalized number Negative underflow: 0
		RM	Negative underflow: Minimum negative normalized number Positive underflow: 0
I	Inexact operation	—	Rounded result

Note 1. If the FPSR.FS bit is cleared, an unimplemented operation exception (E) will occur if an underflow occurs in the rounded result; an underflow exception (U) will not occur. If the FS bit of the FPSR register is set, the flushed result is used as the initial value.

Note 2. If the rounding mode is RN and the FN bit of the FPSR register is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1 (9), Flush to Nearest.**

(6) Exception Details

The following describes the conditions under which each of the FPU exceptions occurs and the FPU responses.

(a) Inexact exception (I)

In the following cases, the FPU detects an inexact exception.

- When the precision of the rounded result is dropped
- When the rounded result overflows while overflow exceptions are prohibited
- When the rounded result underflows while underflow exceptions are prohibited
- When the operand that is a subnormal number is flushed, neither an invalid operation exception (V) nor a division by zero exception (Z) is detected, and the other operands are not Q-NaN

CAUTION

If the FS bit of the FPSR register is cleared and the operation result underflows, an unimplemented operation exception (E) occurs. In such cases, the underflow exception is not detected, so the inexact exception is not detected either.

If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and an inexact exception occurs.

If exception is not enabled

If no other exception occurs, the rounded result or the result that underflows or overflows is stored in the destination register.

(b) Invalid operation exception (V)

An invalid operation exception occurs when either or both of the operands are invalid.

Arithmetic operation with S-NaN included in operands. The conditional move instruction (CMOV), absolute value (ABS), and arithmetic negation (NEG) are not handled as arithmetic operations, but minimum value (MIN) and maximum value (MAX) are handled as arithmetic operations.

- Multiplication: $\pm 0 \times \pm\infty$ or $\pm\infty \times \pm 0$
- Fused-multiply-add: $(\pm 0 \times \pm\infty) + c$ or $(\pm\infty \times \pm 0) + c$. But only if c is not Q-NaN.
- Addition/subtraction or multiply-add operation*1:
  Addition of infinite values with different signs or subtraction of infinite values with the same sign
- Division: $\pm 0 \div \pm 0$ or $\pm\infty \div \pm\infty$
- Square root: When operand is less than 0
- Conversion to integer when source is outside of integer range.
- Comparison: With condition codes 8 to 15, if the operand is unordered (see the table "Definitions of Condition Code Bits and Their Logical Inversions" in the *RH850G4MH User's Manual: Software*).

Note 1. When the multiplication result is infinite or when adding or subtracting between infinities.

If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and an invalid operation exception occurs.

If exception is not enabled

If no other exception occurs, and the destination is a floating-point format, Q-NaN is stored in the destination register. If the destination has an integer format, see the operation result description of each instruction for the value to be stored in the destination register.

(c) Division by zero exception (Z)

A division by zero exception occurs when a divisor is 0 and a dividend is a finite number other than 0.

If exception is enabled

The contents of the destination register are not changed, contents of the source register are saved, and a division by zero exception occurs.

If exception is not enabled

If no other exception occurs, a correctly signed infinite number ($\pm\infty$) is stored in the destination register.

(d) Overflow exception (O)

An overflow exception is detected if the exponent range is infinite and if the result of the rounded floating point is greater than maximum finite number in the destination format.

If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an overflow exception occurs.

If exception is not enabled

If no other exception occurs, the initial value that is determined by the rounding mode and the sign of the intermediate result is stored in the destination register (see **Table 3.123, FPU Initial Values for IEEE754-Defined Exceptions**).

(e) Underflow exception (U)

If the operation result is -2^{Emin} to $+2^{Emin}$ (but not zero), an underflow exception is detected.

Although IEEE754 defines several methods for detecting an underflow, the same method should be used to detect underflows, regardless of the processing to be performed.

The following two methods can be used to detect an underflow for binary floating point numbers.

- The result calculated after rounding and using an infinite exponent range is not zero and is within $\pm 2^{Emin}$.
- The result calculated before rounding and using an infinite exponent range and precision is not zero and is within $\pm 2^{Emin}$.

In this CPU, an underflow is detected before rounding. Or the rounded result is one of the following, an inexact result is detected.

- When a given result differs from the result calculated when the exponent range and precision are infinite

In this CPU, the behavior when an inexact result is detected differs as follows depending on whether underflow exceptions are enabled or disabled:

If exception is enabled

When the FS bit of the FPSR register has been set, if underflow exceptions are enabled, an underflow exception (U) occurs. When the FS bit of the FPSR register has been set, if underflow exceptions are not enabled but inexact exceptions are enabled, an inexact exception (I) occurs.

If exception is not enabled

If the FS bit of the FPSR register has been set, the initial value determined according to the rounding mode and intermediate result value is stored in the destination register (see **Table 3.123, FPU Initial Values for IEEE754-Defined Exceptions**).

CAUTION

If the FS bit of the FPSR register has not been set, an unimplemented operation exception (E) occurs regardless of whether or not exceptions are enabled. Because an unimplemented operation exception (E) must occur, an underflow exception (U) does not occur.

(f) Unimplemented operation exception (E)

The E bit is set and an unimplemented operation exception (E) occurs when an abnormal operand or abnormal result that cannot be correctly processed by hardware has been detected. The operand and destination register contents do not change.

If the FS bit of the FPSR register has been set, an unimplemented operation exception (E) will not occur.

If the FS bit of the FPSR register has been cleared, an unimplemented operation exception (E) will occur under the following conditions (except for CMOVF.D, CMOVF.S, CMPF.D, CMPF.S, CVTF.HS, ABSF.D, ABSF.S, MAXF.D, MAXF.S, MINF.D, MINF.S, NEGF.D, and NEGF.S instructions).

- When the operand is a subnormal number
- When the operation result is a subnormal number, or an underflow has occurred

CAUTION

If the FS bit of the FPSR register is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.

(7) Saving and Returning Status

When a floating-point operation exception occurs, the PC and PSW are saved to the EIPC and EIPSW registers respectively, and the exception cause code is saved to the EIIC register.

The exception cause code of the floating-point operation exception is 71_H.

When an EI level exception is acknowledged while processing a floating-point operation exception, an EIPC register override occurs, which prevents the returning to the instruction that caused the floating-point operation exception to occur. When acknowledgment of EI level exceptions is required, the contents of the EIPC, EIPSW, and EIIC registers must be saved, such as to a stack.

When a floating-point instruction is used in a floating-point operation exception handler routine, the FPSR and FPEPC registers will be overridden if another floating-point operation exception occurs. In such cases, the FPSR and FPEPC registers should be saved at the start of the floating-point operation exception handler processing, and should be returned at the end of the handler processing.

The cause bits of the FPSR register hold the results from only one enabled exception. In any case, the previous results are held until the next enabled exception occurs.

(8) Flushing Subnormal Numbers

This CPU can process subnormal numbers—very small numbers that are lower than the minimum normalized number—in one of the following two ways:

- Normalize the operand or operation result and continue executing arithmetic processing
- Generate an unimplemented operation exception (E) and execute exception handling

Executing software-based exception handling will obtain a more accurate result, but the amount of time required to obtain the result will vary depending on the input value. In control systems that require a real-time performance, therefore, this is usually unacceptable. In this case, it is important to obtain the result within a certain amount time rather than focus on accuracy.

(a) Normalize the subnormal numbers and continue executing arithmetic processing

By setting the FS bit of the FPSR register to 1, this CPU can normalize the operand or operation result to a specific value and continue executing arithmetic processing if a subnormal number is input as the operand or obtained as the operation result. At this time, extremely small differences in values might not appear in the operation result.

For the operand and operation result, the values to which subnormal numbers are flushed when the FS bit is set (1) are shown in **Table 3.124** and **Table 3.125** below.

Table 3.124 Rounding Mode and Flush Value of Input Operand

Sign of Subnormal Operand	Rounding Mode and Value to Which Input Operand Is Flushed			
	RN	RZ	RP	RM
+	+0			
–	–0			

Table 3.125 Rounding Mode and Flush Value of Operation Result

Sign of Subnormal Operand Result	Rounding Mode and Value to Which Operation Result Is Flushed			
	RN[*1]	RZ	RP	RM
+	+0	+0	$+2^{Emin}$	+0
–	–0	–0	–0	-2^{Emin}

Note 1. If the rounding mode is RN and the FN bit of the FPSR register is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1 (9), Flush to Nearest**.

Whether an input operand that is a subnormal number has been flushed or not can be checked by referencing the IF bit of the FPSR register. Whether an operation result that is a subnormal number has been flushed or not can be checked by referencing the U bit of the FPSR register.

CAUTIONS

1. **In control systems that require a real-time performance, it is recommended to always set the FS bit of the FPSR register to 1.**
2. **If the FS bit of the FPSR register is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.**
3. **Whether the operation result is a subnormal number is judged by using the value before rounding.**
4. **The IF bit of the FPSR register also accumulates and indicates information about flushing instructions that have caused a floating-point operation exception.**

(b) Generate an unimplemented operation exception (E) and execute exception handling

By clearing the FS bit of the FPSR register to 0, an unimplemented operation exception (E) will occur if a subnormal number is input as the operand or obtained as the operation result. When an unimplemented operation exception occurs, software-based progressive underflow processing is performed in the floating-point operation exception handling routine, enabling a more accurate result to be obtained. In this case, however, a real-time processing performance might not be realized due to the software processing load.

(c) Instructions that can handle subnormal numbers

The following instructions can be executed without causing an unimplemented operation exception (E) even if an operand that is a subnormal number is input while the FS bit of the FPSR register is 0.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS)

(d) Instructions that are not affected by flushing subnormal numbers

For the following instructions, flushing does not occur even an operand that is a subnormal number is input while the FS bit of the FPSR register is 1.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS)

(9) Flush to Nearest

This CPU provides flush-to-nearest mode, a feature for flushing to the nearest number with higher accuracy when a flushing operation results subnormal number. Flush-to-nearest mode is enabled when the rounding mode is RN and the FN bit of the FPSR register is set (1). When this mode is used, the FPU determines the value to which to flush the subnormal number based on the number of the operation result and not just the sign. This feature has no effect in rounding modes other than RN or on the result of flushing an input operand.

Table 3.126 Rounding Mode and Value to Which Operation Result is Flushed

Value of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result Is Flushed				
	RN		RZ	RP	RM
	FN = 1	FN = 0			
$+2^{Emin-1} \leq$ Operation result $< +2^{Emin}$	$+2^{Emin}$	+0	+0	$+2^{Emin}$	+0
$+0 <$ Operation result $< +2^{Emin-1}$	+0				
$-2^{Emin-1} <$ Operation result < -0	−0	−0	−0	−0	-2^{Emin}
$-2^{Emin} <$ Operation result $\leq -2^{Emin-1}$	-2^{Emin}				

CAUTION

Whether the operation result is a subnormal number is judged by using the value before rounding.

(10) Limitation on the Floating-Point Operation Result

There is a limitation on the floating-point operation result of this CPU.

(a) Operation instruction result when the input data is NaN (not-a-number)

When the input data is NaN (not-a-number), the operation instruction result will be as shown in **Table 3.127**.

Table 3.127 Operation Result when Input data is NaN

Input Data*3	Single-precision Operation Result	Double-precision Operation Result
SNaN and QNaN mixed	7FFF FFFF$_H$	7FFFFFFF FFFFFFFF$_H$
Only SNaN	7FFF FFFF$_H$	7FFFFFFF FFFFFFFF$_H$
Only QNaN	*1	*1
SNaN and real number	7FFF FFFF$_H$	7FFFFFFF FFFFFFFF$_H$
QNaN and real number	*1, *4	*1, *4
Other than above*2	7FFF FFFF$_H$	7FFFFFFF FFFFFFFF$_H$

Note 1. If the input data contains QNaN, QNaN in the input data is selected as the operation result by the priority of reg2, reg1, and reg3.

Note 2. In this case, input data contains real numbers only, and the operation result is invalid.

Note 3. When the operand of an ABSF.S, ABSF.D, NEGF.S, or NEGF.D instruction is NaN, the result of the operation will be the value obtained by the operation on the sign bit alone.

Note 4. When one of the operands of an MAXF.S, MAXF.D, MINF.S, and MINF.D instruction is QNaN, the result of the operation will be whichever operand is a real number.

(b) Results of conversion instructions when the operand is NaN (not-a-number)

When the operand is NaN (not-a-number), the results of conversion are as shown in **Table 3.128**.

Table 3.128 Results of Conversion when the Operand is NaN (not-a-number)

Input Data	Output Format	Results of Conversion
Half-precision SNaN	Single precision	7FFF FFFF$_H$
Half-precision QNaN	Single precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the output value are 0.
Single-precision SNaN	Word (signed)	8000 0000$_H$
	Word (unsigned)	0000 0000$_H$
	Long (signed)	8000 0000_0000 0000$_H$
	Long (unsigned)	0000 0000_0000 0000$_H$
	Half precision	7FFF$_H$
	Double precision	7FFF FFFF_FFFF FFFF$_H$
Single-precision QNaN	Word (signed)	8000 0000$_H$
	Word (unsigned)	0000 0000$_H$
	Long (signed)	8000 0000_0000 0000$_H$
	Long (unsigned)	0000 0000_0000 0000$_H$
	Half precision	The QNaN value with the same sign. Bits 12 to 0 in the fractional part of the operand are truncated.
	Double precision	The QNaN value with the same sign. Bits 28 to 0 in the fractional part of the output value are 0.
Double-precision SNaN	Word (signed)	8000 0000$_H$
	Word (unsigned)	0000 0000$_H$
	Long (signed)	8000 0000_0000 0000$_H$
	Long (unsigned)	0000 0000_0000 0000$_H$
	Single precision	7FFF FFFF$_H$
Double-precision QNaN	Word (signed)	8000 0000$_H$
	Word (unsigned)	0000 0000$_H$
	Long (signed)	8000 0000_0000 0000$_H$
	Long (unsigned)	0000 0000_0000 0000$_H$
	Single precision	The QNaN value with the same sign. Bits 28 to 0 in the fractional part of the operand are truncated.

(c) When 0 comparison is performed by MAXF and MINF

When both reg1 and reg2 are either +0 or –0, the operation will be as follows.

MINF.S, MINF.D: The value of reg1 is stored in reg3.

MAXF.S, MAXF.D: The value of reg2 is stored in reg3.

(d) In the case of the precision instruction (RECIPF)

The operation results of RECIPF.S and RECIPF.D are the same as the operation results of 1/x by DIVF.S and DIVF.D.

3.2.6.2 Extended Floating-Point Operation

The extended floating-point operation unit (FXU) operates as a coprocessor of the CPU, and executes an extended floating-point operation instruction.

The execution of one extended floating-point operation instruction allows the execution of four single-precision floating-point (vector data) operations. This is an operation generally referred to as an SIMD (Single Instruction Multiple Data) operation.

Also, conversion is possible between a floating-point type and an integer type. In extended floating-point operation by this CPU, the floating-point is handled according to the ANSI/IEEE standard 754-2008 “IEEE Standard for Floating-point Arithmetic”.

The FXU consists of an extended floating-point operation block and vector registers where vector data is stored. There are 32 vector registers and the data width is 128 bits.

An extended floating-point operation instruction is executed by the extended floating-point operation block. The extended floating-point operation block performs 32-bit x 4 vector data operation on 128-bit vector registers. This is an operation method generally referred to as SIMD execution.

(1) Configuration of Extended Floating-Point Operation Function

(a) Not implemented

If the extended floating-point operation function is not implemented, any extended floating-point operation instruction cannot be used, and any attempt to execute an extended floating-point operation instruction will result in a coprocessor unusable exception. Also, any extended floating-point system register cannot be accessed, and any attempt to manipulate an extended floating-point system register by means of the LDSR/STSR instruction will result in a coprocessor unusable exception.

(b) Implemented

If the extended floating-point operation function is implemented, any extended floating-point operation instruction can be used. Also, all vector registers and extended floating-point system registers provide the functions described in **Section 3.2.3.5, FXU Function Registers.**

(2) Data Types

The FXU supports the single-precision floating-point formats specified by IEEE 754. For details of the single-precision floating-point formats, see **Section 3.2.6.1 (2), Data Types**.

The FXU does not support the double-precision floating-point formats.

(3) Register Set

For details of the register set, see **Section 3.2.3.5, FXU Function Registers**.

(4) Extended Floating-Point Operation Instructions

Extended floating-point operation instructions consist of extended floating-point vector operation instructions and extended floating-point vector manipulation instructions. For details of the extended floating-point operation instructions, see the *RH850G4MH User’s Manual: Software*.

(5) Extended Floating-Point Operation Exceptions

This section describes how the FXU handles extended floating-point operation exceptions.

(a) Types of Exceptions

When it becomes impossible to perform floating-point operation or handle operation results on any operation way during the execution of an extended floating-point operation instruction, an extended floating-point operation exception occurs.

One of the following two operations is performed when an extended floating-point operation exception has occurred.

- When exceptions are enabled
  The exception cause of each operation way is set in the FXXC register, and the logical OR of the extended floating-point operation exception that has occurred on each operation way is saved into the cause bits of the extended floating-point configuration/status register FXSR.
  Control (software control) is transferred to the exception handler routine.
- When exceptions are disabled
  The exception cause of each operation way is set in the FXXP register, and the logical OR of the extended floating-point operation exception that has occurred on each operation way is saved into the preservation bits of the extended floating-point configuration/status register FXSR.
  An appropriate value (initial value) is stored in the destination register for an instruction that has caused an exception, and execution is continued.

The FXU uses cause bits, enable bits, and preservation bits (status flags) to support the following five types of IEEE754-defined exception causes.

- Inexact operation (I)
- Overflow (O)
- Underflow (U)
- Division by zero (Z)
- Invalid operation (V)

The sixth type of exception cause is unimplemented operation (E), which causes an exception when an extended floating-point operation cannot be executed. This exception requires processing by software. An unimplemented operation exception (E) occurs when exceptions are always enabled, rather than by using properties, enable bits, or preservation bits.

Figure 3.38 shows the relationship among the enable bits and the cause bits of the FXSR register and the cause bits of the FXXC register; these bits are used to support exceptions. **Figure 3.39** shows the relationship among the enable bits and the preservation bits of the FXSR register and the preservation bits of the FXXP register.

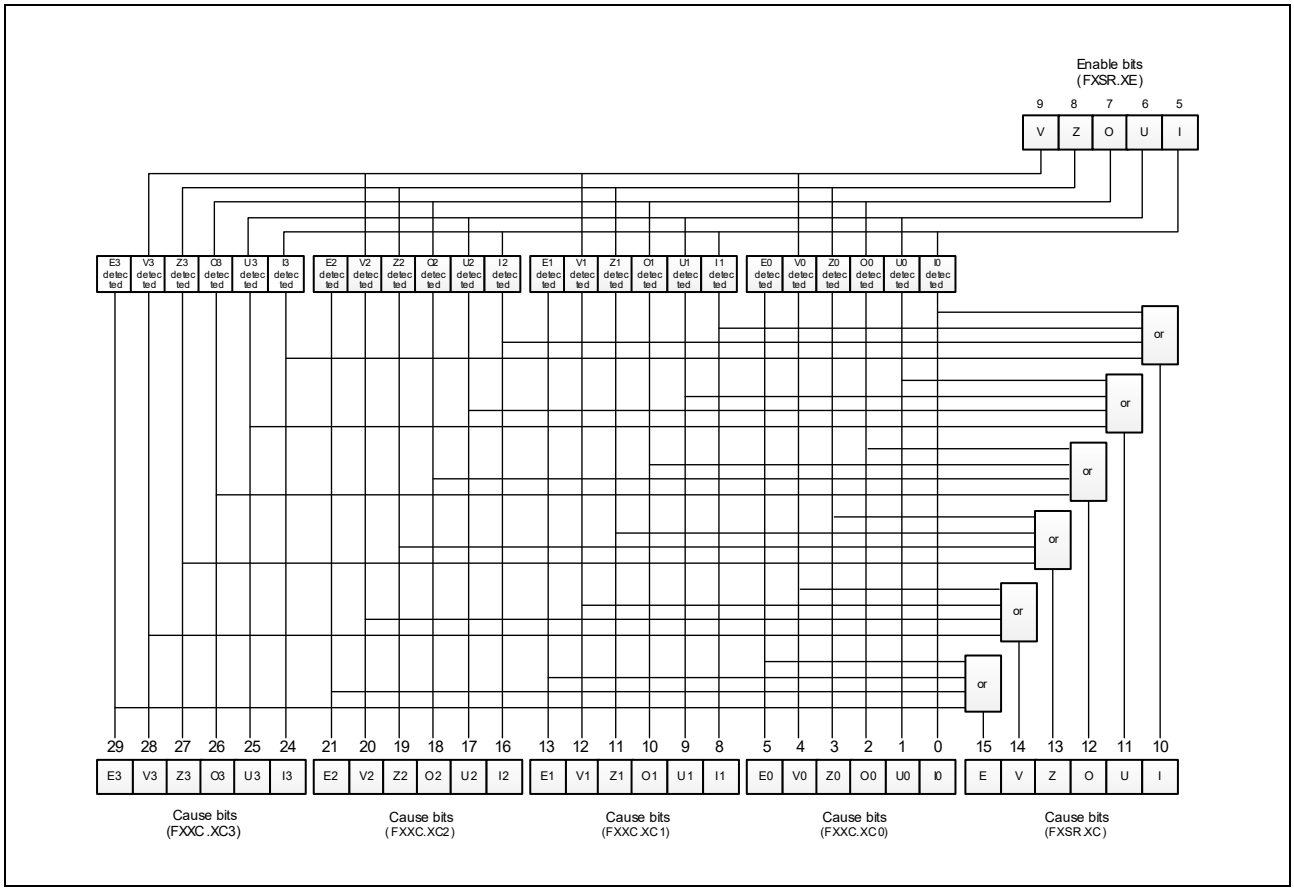


Figure 3.38 Relationship among the Cause and Enable Bits of the FXXC and FXSR Registers

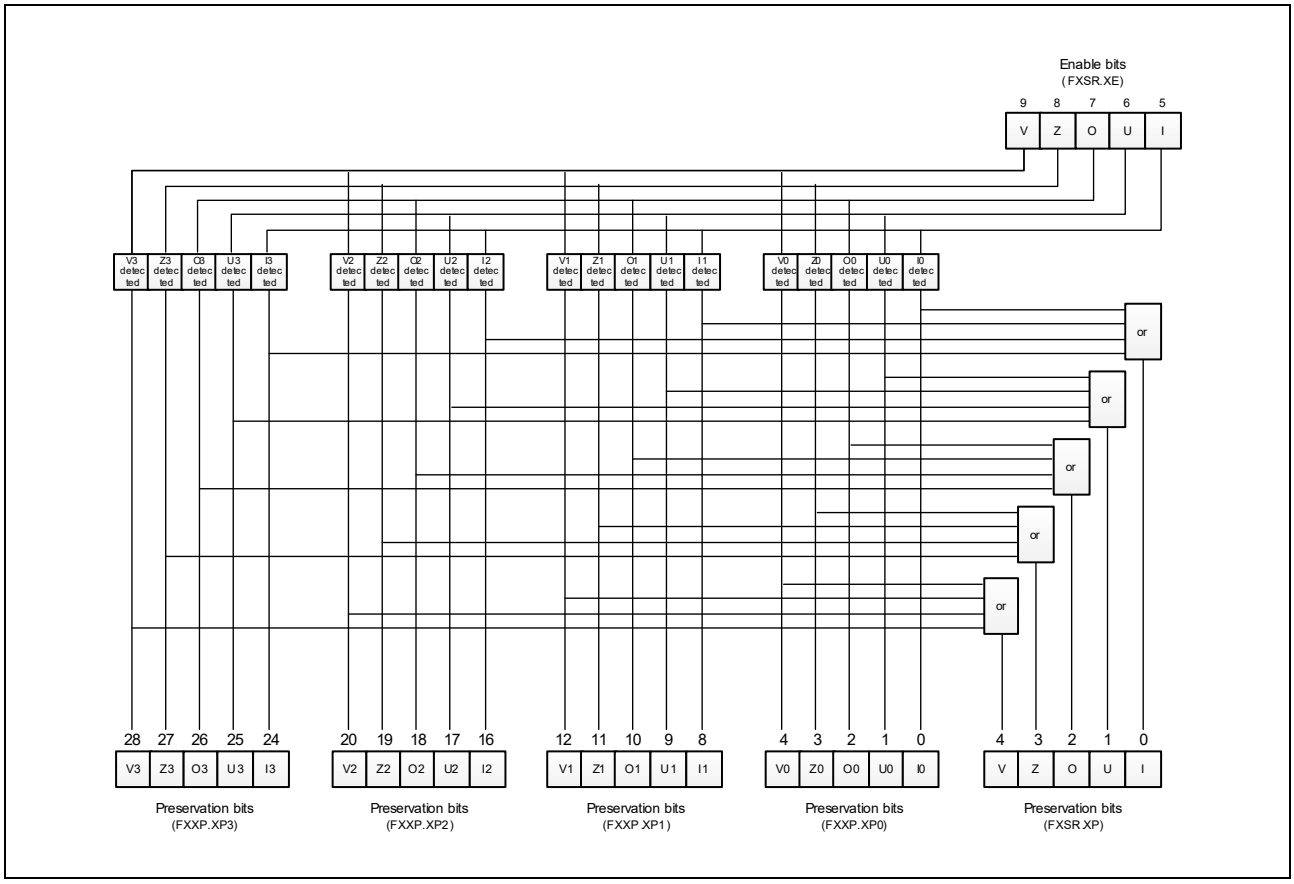


Figure 3.39 Relationship among the Preservation and Enable Bits of the FXXP and FXSR Registers

The five exceptions (V, Z, O, U, and I) defined by IEEE754 are enabled when the corresponding enable bits are set. All operation ways share the enable bits of the FXSR register, but detection of an exception is performed for each operation way. Therefore, different exceptions may be detected on multiple operation ways. In this case, one extended floating-point operation exception occurs.

If an exception has been detected on an operation way with its corresponding enable bit set, the FXU sets the cause bit of the operation way in the FXXC register. Also, it obtains the logical OR of the cause bits of all operation ways and sets the corresponding cause bits in the FXSR register. If the exception can be acknowledged, control is transferred to the exception handler routine.

If exceptions are prohibited, the preservation bit of that operation way in the FXXP register is set. Also, the logical OR of the preservation bits of all operation ways is obtained to set the corresponding preservation bits in the FXSR register. In this case, control is not transferred to the exception handler routine.

(b) Exception handling

When an extended floating-point operation exception occurs, the cause bits of the FXSR register indicate the cause of the extended floating-point operation exception.

Status flag

A corresponding preservation bit is available for each IEEE754-defined exception. The preservation bit is set when the corresponding exception is prohibited and the exception condition has been detected. The preservation bit is set or reset whenever new values are written to the FXSR register by the LDSR instruction.

If an exception is prohibited by an enable bit, predetermined processing is performed by the FXU. This processing provides an initial value as the result, rather than a floating-point operation result. This initial value is determined according to the type of exception. For an overflow exception or underflow exception, the initial value also differs depending on the current rounding mode. **Table 3.129** shows the initial values provided for each of the IEEE754-defined exceptions.

Note that initial values are stored only for operation ways where an exception has been detected. If different exceptions have been detected on multiple operation ways, the initial value corresponding to each exception is stored in the destination of each operation way.

Table 3.129 FXU Initial Values for IEEE754 Defined Exceptions

Area	Description	Rounding Mode	Initial Value
V	Invalid operation	—	Quiet not-a-number (Q-NaN)
Z	Division by zero	—	Correctly signed ∞
O	Overflow	RN	∞ with sign of intermediate result
		RZ	Maximum normalized number with sign of intermediate result
		RP	Negative overflow: Maximum negative normalized number Positive overflow: +∞
		RM	Positive overflow: Maximum positive normalized number Negative overflow: –∞
U	Underflow*1	RN*2	0 with sign of intermediate result
		RZ	0 with sign of intermediate result
		RP	Positive underflow: Minimum positive normalized number Negative underflow: 0
		RM	Negative underflow: Minimum negative normalized number Positive underflow: 0
I	Inexact operation	—	Rounded result

Note 1. If the FXSR.FS bit is cleared, an underflow in the rounded result will cause an unimplemented operation exception (E); therefore, in this case, an underflow exception (U) will not occur. If the FXSR.FS bit is set, the flushed result is used as the initial value.

Note 2. If the rounding mode is RN and FXSR.FN is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1 (9), Flush to Nearest**.

(6) Exception Details

The following describes the conditions under which each extended floating-point operation exception occurs and how the FXU handles it.

(a) Inexact operation exception (I)

If any of the following conditions is detected on any operation way, the FXU detects an inexact operation exception.

- When the precision of the rounded result is dropped
- When the rounded result overflows while overflow exceptions are prohibited
- When the rounded result underflows while underflow exceptions are prohibited
- When the operand that is a subnormal number is flushed, neither an invalid operation exception (V) nor a division by zero exception (Z) is detected, and the other operands are not Q-NaN

CAUTION

If the FXSR.FS bit is cleared and the operation result underflows, an unimplemented operation exception (E) occurs. In such cases, the underflow exception is not detected, so the inexact operation exception is not detected either.

If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an inexact operation exception occurs.

If exception is not enabled

If no other exception occurs, the rounded result or the result that underflows or overflows is stored in the destination register.

(b) Invalid operation exception (V)

If either or both of the operands are invalid on any operation way, an invalid operation exception is detected.

- Arithmetic operation with S-NaN included in operands. The conditional move instruction (CMOV), absolute value (ABS), and arithmetic negation (NEG) are not handled as arithmetic operations, but minimum value (MIN) and maximum value (MAX) are handled as arithmetic operations.
- Multiplication: $\pm 0 \times \pm\infty$ or $\pm\infty \times \pm 0$
- Multiply-add operation: $(\pm 0 \times \pm\infty) + c$ or $(\pm\infty \times \pm 0) + c$. But only if c is not Q-NaN.
- Addition/subtraction or multiply-add operation*1:
  Addition of infinite values with different signs or subtraction of infinite values with the same sign.
- Division: $\pm 0 \div \pm 0$ or $\pm\infty \div \pm\infty$
- Square root: When operand is less than 0.
- Conversion to integer when source is outside of integer range.
- Comparison: With condition codes 8 to 15, if the operand is unordered (see the table "Comparison Condition" for the CMPF.S4 instruction in *the RH850G4MH User's Manual: Software*).

Note 1. When the multiplication result is infinite and when adding or subtracting between infinities.

If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an invalid operation exception occurs.

If exception is not enabled

If no other exception occurs, and the destination is a floating-point format, Q-NaN is stored in the destination register. If the destination has an integer format, see the operation result description of each instruction for the value to be stored in the destination register.

(c) Division by zero exception (Z)

A division by zero exception is detected if, on any operation way, a divisor is 0 and a dividend is a finite number other than 0.

If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and a division by zero exception occurs.

If exception is not enabled

If no other exception occurs, a correctly signed infinite number (±∞) is stored in the destination register.

(d) Overflow exception (O)

An overflow exception is detected if, on any operation way, the exponent range is infinite and if the result of the rounded floating-point is greater than maximum finite number in the destination format.

If exception is enabled

The contents of the destination register are not changed, the contents of the source register are saved, and an overflow exception occurs.

If exception is not enabled

If no other exception occurs, the initial value that is determined by the rounding mode and the sign of the intermediate result is stored in the destination register (see **Table 3.129, FXU Initial Values for IEEE754 Defined Exceptions**).

(e) Underflow exception (U)

If, on any operation way, the operation result is -2^{Emin} to $+2^{Emin}$ (but not zero), an underflow exception is detected.

Although IEEE754 defines several methods for detecting an underflow, the same method should be used to detect underflows, regardless of the processing to be performed.

The following two methods can be used to detect an underflow for binary floating-point numbers.

- The result calculated after rounding and using an infinite exponent range is not zero and is within $\pm 2^{Emin}$

- The result calculated before rounding and using an infinite exponent range and precision is not zero and is within $\pm 2^{Emin}$.

In this CPU, an underflow is detected before rounding.

Also, if the rounded result is one of the following, an inexact result is detected.

- When a given result differs from the result calculated when the exponent range and precision are infinite

In this CPU, the behavior when an inexact result is detected differs as follows depending on whether underflow exceptions are enabled or disabled.

If exception is enabled

When the FXSR.FS bit has been set, if underflow exceptions are enabled, an underflow exception (U) occurs. When the FXSR.FS bit has been set, if underflow exceptions are not enabled but inexact operation exceptions are enabled, an inexact operation exception (I) occurs.

If exception is not enabled

If the FXSR.FS bit has been set, the initial value determined according to the rounding mode and intermediate result value is stored in the destination register (see **Table 3.129, FXU Initial Values for IEEE754 Defined Exceptions**).

CAUTION

If the FXSR.FS bit has not been set, an unimplemented operation exception (E) always occurs regardless of whether or not exceptions are enabled. Therefore, an underflow exception (U) does not occur.

(f) Unimplemented operation exception (E)

The E bit is set and an unimplemented operation exception (E) occurs when, on any operation way, an abnormal operand or abnormal result that cannot be correctly processed by hardware has been detected. The operand and destination register contents do not change.

If the FXSR.FS bit has been set, an unimplemented operation exception (E) will not occur.

If the FXSR.FS bit has been cleared, an unimplemented operation exception (E) will occur under the following conditions (except for ABSF. S4, CMOVF. W4, CMPF.S4, CVTF.HS4, MAXF.S4, MAXRF.S4, MINF.S4, MINRF.S4, and NEGF.S4 instructions).

- When the operand is a subnormal number
- When the operation result is a subnormal number, or an underflow has occurred

CAUTION

If the FXSR.FS bit is set to 1, an unimplemented operation exception (E) will not occur under any circumstances.

(7) Saving and Restoring Status

When an extended floating-point operation exception occurs, the PC and PSW are saved to the EIPC and EIPSW registers respectively, and the exception cause code is saved to the EIIC register.

The exception cause code of the extended floating-point operation exception is 75_H.

When an EI level exception is acknowledged while processing an extended floating-point operation exception, an EIPC register override occurs, which prevents the returning to the instruction that caused the extended floating-point operation exception to occur. Therefore, when acknowledgment of EI level exceptions is required, the contents of the EIPC, EIPSW, and EIIC registers must be saved in advance into a stack or the like.

When an extended floating-point operation instruction is used in an extended floating-point operation exception handler routine, the FXSR and FXXC registers will be overridden if another floating-point operation exception occurs. In such cases, the FXSR and FXXC registers should be saved at the start of the extended floating-point operation exception handler processing, and should be restored at the end of the handler processing.

The cause bits of the FXSR register hold the results from only one enabled exception. In any case, the previous results are held until the next enabled exception occurs.

(8) Flushing Subnormal Numbers

This CPU can process subnormal numbers—very small numbers that are lower than the minimum normalized number—in one of the following two ways:

- Normalize the operand or operation result and continue executing arithmetic processing
- Generate an unimplemented operation exception (E) and execute exception handling

Executing software-based exception handling will obtain a more accurate result, but the amount of time required to obtain the result will vary depending on the input value. In control systems that require a real-time performance, therefore, this is usually unacceptable. In this case, it is important to obtain the result within a certain amount of time rather than focus on accuracy.

(a) Normalize the subnormal numbers and continue executing arithmetic processing

By setting the FS bit of the FXSR register to 1, this CPU can normalize the operand or operation result to a specific value and continue executing arithmetic processing even if a subnormal number is input as the operand or obtained as the operation result. At this time, extremely small differences in values might not appear in the operation result.

For the operand and operation result, the values to which subnormal numbers are flushed when the FS bit is set to 1 are shown in **Table 3.130** and **Table 3.131** below.

Table 3.130 Rounding Mode and Flush Value of Input Operand

Sign of Subnormal Operand	Rounding Mode and Value to Which Input Operand is Flushed			
	RN	RZ	RP	RM
+	+0			
–	–0			

Table 3.131 Rounding Mode and Flush Value of Operation Result

Sign of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result is Flushed			
	RN*1	RZ	RP	RM
+	+0	+0	$+2^{Emin}$	+0
–	–0	–0	–0	-2^{Emin}

Note 1. If the rounding mode is RN and FXSR.FN is set, flushing will occur in the direction of higher accuracy. For details, see **Section 3.2.6.1(9), Flush to Nearest.**

Whether an input operand that is a subnormal number has been flushed or not can be checked by referencing the IF bit of the FXSR register. Whether an operation result that is a subnormal number has been flushed or not can be checked by referencing the U bit of the FXSR register.

CAUTIONS

1. **In control systems that require a real-time performance, it is recommended to always set the FS bit of the FXSR register to 1.**
2. **If the FS bit of the FXSR register is set to 1, an unimplemented operation exception will not occur under any circumstances.**
3. **Whether the operation result is a subnormal number is judged by using the value before rounding.**
4. **The IF bit of the FXSR register also accumulates and indicates information about flushing instructions that have caused an extended floating-point operation exception.**

(b) Generate an unimplemented operation exception (E) and execute exception handling

Clearing the FS bit of the FXSR register to 0 will cause an unimplemented operation exception (E) if a subnormal number is input as the operand or obtained as the operation result. When an unimplemented operation exception occurs, software-based progressive underflow processing is performed in the floating-point operation exception handling routine, enabling a more accurate result to be obtained. In this case, however, a real-time processing performance might not be ensured depending on the software processing load.

(c) Instructions that can handle subnormal numbers

The following instructions can be executed without causing an unimplemented operation exception (E) even if an operand that is a subnormal number is input while the FS bit of the FXSR register is 0.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS4)

(d) Instructions that are not affected by flushing subnormal numbers

For the following instructions, flushing does not occur even if an operand that is a subnormal number is input while the FS bit of the FXSR register is 1.

- Conditional move instruction (CMOV), absolute value (ABS), arithmetic negation (NEG)
- Minimum value (MIN), maximum value (MAX), compare (CMPF)
- Conversion from half-precision to single-precision (CVTF.HS4)

(9) Flush to Nearest

This CPU provides flush-to-nearest mode, a feature for flushing to the nearest number with higher accuracy when a flushing operation results subnormal number. Flush-to-nearest mode is enabled when the rounding mode is RN and FXSR.FN is set to 1. When this mode is used, the FXU determines the value to which to flush the subnormal number based on the number of the operation result and not just the sign. This feature has no effect in rounding modes other than RN or on the result of flushing an input operand.

Table 3.132 Rounding Mode and Value to Which Operation Result is Flushed

Value of Subnormal Operation Result	Rounding Mode and Value to Which Operation Result is Flushed				
	RN		RZ	RP	RM
	FN = 1	FN = 0			
$+2^{Emin-1} \leq$ Operation result $< +2^{Emin}$	$+2^{Emin}$	+0	+0	$+2^{Emin}$	+0
$+0 <$ Operation result $< +2^{Emin-1}$	+0				
$-2^{Emin-1} <$ Operation result < -0	−0	−0	−0	−0	-2^{Emin}
$-2^{Emin} <$ Operation result $\leq -2^{Emin-1}$	-2^{Emin}				

CAUTION

Whether the operation result is a subnormal number is judged by using the value before rounding.

(10) Limitation on the Extended Floating-Point Operation Result

There is a limitation on the extended floating-point operation result of this CPU.

(a) Operation instruction result when the input data is NaN (not-a-number)

When the input data is NaN (not-a-number), the operation instruction result will be as shown in **Table 3.133**.

Table 3.133 Operation Result when Input data is NaN

Input Data*3	Operation Result
SNaN and QNaN mixed	$7FFF\ FFFF_H$
Only SNaN	$7FFF\ FFFF_H$
Only QNaN	*1,*4
SNaN and real number	$7FFF\ FFFF_H$
QNaN and real number	*1,*4
Other than above*2	$7FFF\ FFFF_H$

Note 1. If the input data contains QNaN, QNaN in the input data is selected as the operation result by the priority of wreg2, wreg1, and wreg3.

Note 2. In this case, input data contains real numbers only, and the operation result is invalid.

Note 3. When the operand of an ABSF.S4 or NEGF.S4 instruction is NaN, the result of the operation will be the value obtained by the operation on the sign bit alone.

Note 4. When one of the operands of an MAXF.S4, MAXRF.S4, MINF.S4, and MINRF.S4 instruction is QNaN, the result of the operation will be whichever operand is a real number.

(b) Results of conversion instructions when the input data is NaN (not-a-number)

When the input data is NaN (not-a-number), the results of conversion are as shown in **Table 3.134**.

Table 3.134 Results of Conversion when the Input Data is NaN (not-a-number)

Input Data	Output Format	Results of Conversion
Half-precision SNaN	Single precision	7FFF $FFFF_H$
Half-precision QNaN	Single precision	The QNaN value with the same sign. Bits 12 to 0 in the mantissa part of the output value are 0.
Single-precision SNaN	Word (signed)	8000 0000_H
	Word (unsigned)	0000 0000_H
	Half precision	$7FFF_H$
Single-precision QNaN	Word (signed)	8000 0000_H
	Word (unsigned)	0000 0000_H
	Half precision	The QNaN value with the same sign. Bits 12 to 0 in the mantissa part of the input data are truncated.

(c) When 0 comparison is performed by MAXF and MINF

When wreg1 and wreg2 elements are either +0 or –0, the operation will be as follows.

MINF.S4: The values of each element of wreg1 are stored in each element of wreg3.

MAXF.S4: The values of each element of wreg2 are stored in each element of wreg3.

MINRF.S4: The values of even elements of wreg1 and wreg2 are stored in each element of wreg3.

MAXRF.S4: The values of odd elements of wreg1 and wreg2 are stored in each element of wreg3.

(d) In the case of the precision instruction (RECIPF)

The operation results of RECIPF.S4 are the same as the operation results of 1/x by DIVF.S4.

3.2.7 Hazard Control

This CPU may require hazard management in software in order to accurately refer to the operation result of memory and system registers in subsequent instructions.

- Special instruction for synchronous processing
- Execution completion guarantee of store instruction
- Hazard control after updating system register

3.2.7.1 Synchronization Processing

In order to increase the processing performance, this CPU executes subsequent instructions without waiting for completion of the processing of preceding instructions, as long as there is no dependence relationship between a preceding instruction and a subsequent instruction. For example, when a store instruction is to update the value of a control register of a peripheral device which operates more slowly than the CPU, even when updating of the register is not completed while the bus system is still handling processing to store the value, the execution of a subsequent instruction will proceed. For this reason, if a subsequent instruction should certainly be made to wait for completion of processing of a preceding instruction by software, procedures for synchronization processing are required.

Moreover, when a single CPU is executing multiple load and store instructions, the results of execution are guaranteed to be in the order of the instructions of the program. However, in order to guarantee the order of data accesses among multiple bus masters outside a mutual exclusion control section, synchronization processing should be performed before the subsequent data access. For details of synchronization processing, refer to **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** and **Section 3.9.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation**

This CPU provides four instructions dedicated for synchronization processing. In addition, since a wait for the result of preceding load processing depends on general-purpose registers and handled automatically by hardware, no synchronization processing is required by software.

(1) SYNCP

The SYNCP instruction is a special instruction for pipeline synchronization processing, which makes pipelines wait for the result of executing a preceding instruction is reflected before starting to execute a subsequent instruction. Though the SYNCP instruction causes a wait for the results of executing all preceding load processing (up to storing of the loaded values in general-purpose registers), it does not cause waiting for the results of executing store processing (updating of the destination memory locations for storage and of memory-mapped control registers). For how to be sure that the results of executing store processing are reflected for subsequent instructions, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(2) SYNCM

The SYNCM instruction is a special instruction for memory-access synchronization processing, and causes a wait for the results of executing all preceding load processing (up to storage of the loaded values in general-purpose registers) and for the results of executing all preceding store processing (updating of the destination memory locations for storage and memory-mapped control registers). However, for those bus systems and peripheral devices for which storage is speculatively completed

(resulting in delays in the completion of writing to the memory and control registers), the SYNCM instruction does not guarantee completion of the execution of store instructions. For how to be sure that the results of executing store processing are reflected for subsequent instructions in such cases, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(3) SYNCI

The SYNCI instruction is a special instruction for the synchronization of instruction fetching. It causes subsequent instructions that have already been fetched into the pipeline but not executed to be discarded, and restarts instruction fetching from the earliest of the discarded instructions to have been fetched. This instruction is used when the result of a preceding instruction must be reflected in processing by subsequently fetched instructions. Though the SYNCI instruction causes a wait for the results of executing all preceding load processing (up to storing of the loaded values in general-purpose registers), it does not cause waiting for the results of executing store processing (updating of the destination memory locations for storage and of memory-mapped control registers). If a cache instruction has been issued, this instruction also causes a wait for the completion of the instruction cache operation.

The SYNCI instruction should be executed after changes to the CPU operating mode or reflection of changes to the memory protection settings if this will be required in subsequent instruction fetching. For details, see **Section 3.2.7.3 (1), Updating the Settings Related to Instruction Fetching**.

When reflecting the results of executing store instructions in subsequent instruction fetching is required, due to changes to self-modifying code or switching between areas of the code-flash memory, execute a dummy read before the SYNCI instruction. This causes a wait for the results of executing store instructions, resulting in these being reflected in subsequent instruction fetching. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

(4) SYNCE

The SYNCE instruction of this CPU does not perform processing related to synchronization.

(5) Synchronization Processing summary

The effects of the four synchronization processing instructions are summarized in **Table 3.135**.

Table 3.135 Effects of Synchronization Processing Instructions

Synchronization Processing Instruction	Guaranteed Synchronization Processing				
	Instruction Fetching		Instruction Execution		
	Subsequent Instructions	Cache Operations*1	Load Instructions	Store Instructions	Instructions for Other Operations
SYNCP	—	—	Execution completed*3	—	Execution completed
SYNCM	—	—	Execution completed*3	Execution completed*4	Execution completed
SYNCI	Re-fetching*2	Execution completed	Execution completed*3	—	Execution completed
SYNCE	—	—	—	—	—

Note 1. This indicates the execution of cache instructions. The LDSR instruction to update the function registers for cache operations is included among the instructions for other operations.

Note 2. This CPU speculatively fetches instructions, so even if the next instruction from the SYNCI instruction has already been fetched, it is fetched again.

Note 3. This indicates the processing up to the storing of loaded data in general-purpose registers

Note 4. This indicates the processing up to updating of the destination memory locations and memory-mapped control registers for storage. However, updating may not be guaranteed. This depends on the specification of the destination for writing. For details, see **Section 3.2.7.2, Guaranteeing the Completion of Store Instruction**.

3.2.7.2 Guaranteeing the Completion of Store Instruction

Instructions for storage may appear to be completed faster than those for loading given the same address. However, the data for storage may not actually have been stored when the instruction has been completed. For example, in the synchronization of multiple cores or the manipulation of peripheral devices, processing related to the stored values should only proceed after storage in the target locations is confirmed. This should be checked by software after the execution of instructions for storage.

This CPU provides two approaches to waiting for the progress of storage at different stages.

(1) Using the SYNCM Instruction to Wait for the Completion of Storage

Executing the SYNCM instruction causes operations to wait until the results of preceding store instructions will be correctly read by subsequent reading. This guarantees confirmed execution when an instruction stored in the memory is to be executed or data are to be transferred to the other CPU core or by DMA via the memory.

In the following example, after waiting for the execution of the two store instructions that proceed the SYNCM instruction, the processing for which waiting was required proceeds in “Operations” below. Processing to branch to an address where an instruction code is stored, for notification of the other CPU, and so on, falls into the category of “Operations”.

```
ST.W      r21, 0[r1]          //    Store data 1
ST.W      r22, 4[r1]          //    Store data 2
SYNCM                         //    Wait for completion of the store instructions
Operation                     //    Processing after waiting for completion of the store instructions
```

However, the SYNCM instruction does not guarantee that the results of preceding store instructions will have already been reflected at the destinations. For example, when storing a value in a register of a peripheral device, the register of the peripheral device may not have been updated when waiting due to the SYNCM instruction has finished, and the resulting may be operation that is not as desired. In such cases, use dummy reading to cause the wait. This is described in **Section 3.2.7.2 (2), Using Dummy Reading to Wait for the Completion of Storage**.

In addition, since the SYNCM instruction requires hardware operations to wait in response to its execution, it is only effective in access to bus slaves where the CPU can recognize the progress of processing for storage. If a bus slave is capable of arbitrating requests from multiple bus masters, or is connected to the bus via a bus bridge, the CPU is unable to recognize the progress of storage, so using the SYNCM instruction to cause the wait may not be effective.

The bus slaves (address groups) that support using the SYNCM instruction to wait for the completion of storage in this CPU are listed in **Table 3.136**.

Table 3.136 Bus Slaves that Support Using the SYNCM Instruction to Wait for the Completion of Storage in This CPU

Bus Slaves	Description
L1RAM (given CPU)	Waiting for the updating of memory is also possible.*1
L1RAM (other CPU)	A configuration of two CPU cores can be assumed for this device. For configurations with three or more cores, refer to **Section 3.9.1.3, Product information of SYNCM**.
L2RAM	A configuration of two CPU cores with a single shared L2RAM can be assumed for this device. For configurations with three or more cores or having multiple L2RAMs, refer to **Section 3.9.1.3, Product information of SYNCM**.
INTC1	This is the dedicated interrupt controller for connection with the CPU.
Others	Depends on the specifications of the connected bus or slave. For details, refer to **Section 3.9.1.3, Product information of SYNCM**.

Note 1. For details, **Section 3.2.7.2 (2), Using Dummy Reading to Wait for the Completion of Storage**.

The instructions for which the SYNCM instruction causes a wait are all preceding instructions for storing and loading. Multiple instructions for storing and loading running at the same time raises the possibility of the wait being longer than is necessary.

One SYNCM instruction per store instruction is not required. Following consecutive storage in a specific bus slave in units of memory type, peripheral device, and so on, a single SYNCM instruction after the last store instruction leads to a wait for the completion of all preceding store instructions. In the above example, one SYNCM instruction is executed after the execution of two store instructions.

(2) Using Dummy Reading to Wait for the Completion of Storage

Executing a load instruction for the same address immediately after a store instruction can also be used to wait for the completion of a store instruction. This approach is unlike using the SYNCM instruction to cause waiting, since it involves actually waiting for the result of the store instruction to be reflected. On the other hand, this approach may require more cycles than using the SYNCM instruction.

Specifying r0 as the destination for storage of the value loaded by the dummy read enables discarding of the loaded values when they are unnecessary, without destroying the other register values. The following is an example of code for this.

```
ST.W       r21, 0[r1]          //    Store data 1
ST.W       r22, 4[r1]          //    Store data 2
LD.W       4[r1], r0           //    Load data from the target for access (dummy read).
SYNCP                          //    Wait for loading to be completed.
Operation                      //    Processing after waiting for completion of the store instructions
```

Unlike using the SYNCM instruction, this approach combines functions of the CPUs to cause the wait. Since it does not require dedicated hardware, it may be used with many bus slaves.

On the other hand, if a cache for temporarily storing data is present in the bus system along the path to the destination, a value to be stored will be read by a subsequent load instruction even if it has not yet reached its destination, so this method cannot guarantee the completion of store instructions in such cases. Also, depending on the specifications of peripheral devices, a predetermined time after a change to a setting by a store instruction may have to elapse before the desired value can be read. In addition to this approach, such cases require processing to wait for the predetermined time.

Though this CPU does not include such cache functions or peripheral functions with limits on the time to elapse before reading, the situation depends on the product. For details, refer to **Section 3.9.1.4, Product information of Waiting for the Completion of Storage**.

As with using the SYNCM instruction, in using dummy reading, following consecutive storage in a particular bus slave, only a single load instruction following the final store instruction is effective in waiting for the execution of all preceding store instructions to be complete.

Table 3.137 lists the bus slaves for which using dummy reading to wait is effective.

Table 3.137 Bus Slaves for which Using Dummy Reading to Wait is Effective

Bus Slaves	Description
L1RAM (other CPU)	
L2RAM	
INTC1	
Others	Depends on the specifications of the connected bus or slave. For details, refer to **Section 3.9.1.4, Product information of Waiting for the Completion of Storage**.

In the case of access to the L1RAM of a given CPU, for the sake of the efficiency of processing, execution of the load instruction may be completed before the memory is actually updated. On the other hand, it is possible to execute a SYNCM instruction to cause the wait for completion of the updating of memory. Therefore, use the SYNCM instruction to cause the wait for the completion of instructions for storage in the L1RAM of a given CPU.

When reflecting the results of executing store instructions in subsequent instruction fetching is required, due to changes to self-modifying code or switching between areas of the code-flash memory, execute a SYNCI instruction rather than a SYNCP instruction after the dummy read. This causes a wait for the results of executing store instructions, resulting in these being reflected in subsequent instruction fetching. The following shows an example of code for this.

```
ST.W        r21, 0[r1]              //    A change relating to instruction fetching
LD.W        0[r1], r0               //    Load data from the target for access (dummy read).
SYNCI                               //    Synchronization of the wait for the execution of loading and instruction
                                            fetching
Operation                           //    Processing after the change to instruction fetching operations
```

3.2.7.3 Hazard Management after System Register Update

If an LDSR instruction is executed to update the setting of a system register before an STSR instruction is executed to read the system register or before a CALLT instruction or the like, which uses the system register, is executed, then the new setting must be reflected in order to perform the desired operation.

This CPU guarantees that if an LDSR instruction is used to update system registers shown in **Table 3.138**, the new register setting will be applied when the subsequent instruction is executed. However, it does not guarantee that the new setting will be applied in instruction fetching. In this case, synchronization process is required. Also, the execution of an EI or DI instruction is treated in the same way as the update of PSW by an LDSR instruction; that is, it is guaranteed that the values updated by an EI or DI instruction is applied to the subsequent instruction.

Table 3.138 System Registers that Guarantee Reflection of their Updates by LDSR Instruction to the Subsequent Instruction

selID*1	System Register*2							
0	EIPC	EIPSW	FEPC	FEPSW	PSW	FPSR	FPEPC	FPST
	FPCC	FPCFG	EIIC	FEIC	CTPC	CTPSW	CTBP	SNZCFG
	EIWR	FEWR						
1	SPID	SPIDLIST	RBASE	EBASE	INTBP*3	MCTL	PID	SVLOCK
	SCCFG	SCBP						
2	PEID	BMID	MEA	MEI	ISPR*3	ICSR	INTCFG*3	PLMR*3
	RBCR0*4	RBCR1*4	RBNR*4	RBIP*4	IMSR			
5	MPM	MPCFG	MCA	MCS	MCC	MCR	MCI	MPIDX*5
10	FXSR*6	FXST*6	FXINFO*6	FXCFG*6	FXXC*6	FXXP*6		
12	LSCFG							
13	RDBCR	L1RCFG						

Note 1. In the representation of LDSR and STSR instructions, selection ID are represented as selID.

Note 2. This table includes a register whose value cannot be updated and a register that has a bit whose value cannot be updated.

Note 3. If these registers, which control the acceptance of interrupts, are updated, interrupts will be accepted with the new register settings if interrupt requests are present at the time of executing the subsequent instruction.

Note 4. If these registers, which control the register bank function, are updated, interrupts will be accepted with the new register settings if interrupt requests are present at the time of executing the subsequent instruction. However, when the RESBANK instruction is executed after updating the RBCR0.MD, a synchronization process is required.

Note 5. If an LDSR instruction is executed to update the MPLA, MPUA, or MPAT register immediately after another LDSR instruction is executed to update the MPIDX register, it is guaranteed that the new setting of MPIDX is reflected in the subsequent instructions. In order to read the MPLA, MPUA, or MPAT register by using a STSR instruction after the update of the MPIDX register, a synchronization process is required.

Note 6. FXU system register is not supported in RH850/U2A.

If the settings related to instruction fetch or a system register that is not shown in **Table 3.138** is updated, the new register setting can be reflected by performing any of the following synchronization processes immediately after the LDSR instruction. The appropriate synchronization process should be decided according to details of the new register setting and the subsequent operation.

For SYNCI or SYNCP instruction, that are used for synchronization process, see **Section 3.2.7.1, Synchronization Processing**.

(1) Updating the Settings Related to Instruction Fetching

To effectively supply instruction codes read from the memory to the instruction execution unit, operations of instruction fetching are independent from those of the instruction execution unit and are speculative. This means that the progress of reading the subsequent instruction code by instruction fetching is not known on completion of the system register update. In other words, the result from which instruction fetching is applied to the new setting of the system register is not known. Therefore, an explicit synchronization process is required for updates of the system registers which are related to instruction fetching. This process is also necessary for updates of the system registers listed in **Table 3.138** in order to ensure that the updates are reflected in instruction fetching.

The system registers related to instruction fetching are listed below:

CPU operating mode:	PSW.UM, PSW.CU1, PSW.CU0
Access protection:	SPID
Memory protection:	MPU function registers
Instruction cache:	Cache operation function registers
Instruction fetch:	Instruction fetch control registers

If the settings of the system registers listed above are updated, by executing the SYNCI, EIRET, or FERET instruction, the new settings are surely reflected when the subsequent instruction is fetched. Here, these instructions are called “Instruction fetch synchronization instructions”. For usage of the EIRET and FERET instructions, see **Section 3.2.7.3 (7), Use of EIRET and FERET Instructions in Synchronization Process**.

For fetching the instruction fetch synchronization instructions themselves, whether the updates of the system registers are reflected or not cannot be decided. In these cases, the following cares must be taken.

(a) Updating PSW.UM

When an LDSR instruction is used to update PSW, especially by setting PSW.UM (to 1), which in response enables memory protection function, be sure that fetching of the SYNCI instruction, which is used for synchronizing the instruction fetch, does not cause a memory protection violation. Also note that setting PSW.UM disables execution of some instructions and updating of some system registers.

On the other hand, when an EIRET or FERET instruction is executed, the value in EIPSW or FEPSW is transferred to PSW as the operation of the instruction. This means that fetching of the instruction takes place before updating of PSW.UM by the instruction itself. For these reasons, updating of PSW.UM by an EIRET or FERET instruction is recommended.

(b) Updating PSW.CU1 and PSW.CU0

After PSW.CU1 or PSW.CU0 is updated, if a FXU or FPU instruction is executed without synchronization process, a coprocessor unusable exception (UCPOP) may not be properly detected. To avoid this, execute a SYNCI instruction for synchronization after updating these bits. Unlike updating of PSW.UM, updating of PSW.CU1 or PSW.CU0 and detection of a coprocessor unusable exception can be surely synchronized by a SYNCI instruction.

For updating PSW.CU1 or PSW.CU0 and PSW.UM at the same time, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

(c) Updating SPID

SPID is referenced by memory protection function and the bus system outside the CPU. If an ID without access permission to the memory resources or peripheral devices is set to SPID, instruction fetching does not take place correctly. Be sure to set an appropriate ID to SPID. To ensure that the bus system references any updates on SPID, it might be the case that the instruction cache need to be disabled once. For details, see **Section 44, Functional Safety**.

(d) Updating MPU Function Registers

Updating the settings of memory protection by manipulating the MPU function registers is detailed in **Section 3.2.7.3 (2), Updating the Memory Protection Settings of MPU**.

For reflecting the updated settings in fetching of the subsequent instruction, execute either of the SYNCI, EIRET, and FERET instructions. Note, however, that whether the instruction fetch synchronization instruction itself is fetched with the new protection settings or not is not known. In this case, the following operations are required.

i) Enable memory protection in user mode only and update the settings in supervisor mode.

ii) Update the settings of memory protection in the area where execution of instructions are allowed.

iii) Surely set execution of instructions for the memory area including those for instruction fetch synchronization instruction to be enabled

For updating PSW.UM with operation i) above, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

(e) Updating Cache Operation Function Registers

Updating these registers by an instruction does not require certain synchronization process. However, if you want to make sure that the disabled state of the instruction cache is reflected in fetching of the subsequent instruction, execute an instruction fetch synchronization instruction after the update of the cache operation function registers.

(f) Updating Instruction Fetch Control Registers

Updating these registers by an instruction does not require certain synchronization process.

(2) Updating the Memory Protection Settings of MPU

The system registers of MPU that are shown in **Table 3.138** guarantee that their latest settings are reflected in the subsequent instruction, but this does not apply to other system registers (MPIDX, MPLA, MPUA, MPAT, and MPIDn (n = 0 to 7) that configure the settings of protection areas.

In order to read settings updated by an LDSR instruction by using the subsequent STSR instruction, the synchronization process described in **Section 3.2.7.3 (4), Updating Register Bank Function-Related System Registers** is required.

For reflecting the updated memory protection settings in the instruction fetch, see **Section 3.2.7.3 (1), Updating the Settings Related to Instruction Fetching**.

For reflecting the updated memory protection setting in the instruction that makes operand accesses, execute a SYNCI or SYNCP instruction after the update. This ensures that the new protection setting is reflected in the operand accesses by the subsequent instructions.

The EIRET and FERET instructions can be used instead of the SYNCI and SYNCP instructions. Especially, for the case of updating memory protection settings related to instruction fetch as well as making operand accesses, use of an EIRET or FERET instruction is recommended.

(3) Updating Interrupt-Related System Registers

As shown in **Table 3.138**, when the system registers related to interrupts (ISPR, INTCFG, PLMR) are updated, it is guaranteed that the updates are reflected when the subsequent instruction is executed.

However, when updating the setting of INTCFG to stop automatic update of ISPR, for example, if an interrupt is generated after INTCFG is updated and before an LDSR instruction that updates ISPR is executed, only the update in INTCFG is reflected in the interrupt handling. As the update of ISPR is not reflected, the desired interrupt handling cannot be performed.

To avoid this, when updating the system registers related to interrupts, it is recommended to disable interrupts by setting PSW.ID to 1.

We also recommend that updating of PSW.EBV and EBASE, which specify vector address of an interrupt handler, are executed while interrupts are disabled. For details, see **Section 3.2.7.3 (8), Updating PSW.EBV and EBASE**.

(4) Updating Register Bank Function-Related System Registers

As shown in **Table 3.138**, when the system registers related to the register bank function (RBCR0, RBCR1, RBNR, RBIP) are updated, it is guaranteed that the updates are reflected when the subsequent instruction is executed. For example, if an interrupt is generated after an LDSR instruction that updates RBIP is executed, the automatic context saving to the register bank is executed using the updated value of RBIP.

However, when updating the settings of RBCR0 and RBCR1, for example, if an interrupt is generated after RBCR0 is updated and before an LDSR instruction that updates RBCR1 is executed, only the update in RBCR0 is reflected in the automatic context saving to the register bank. As the update of RBCR1 is not reflected, the desired processing cannot be performed.

To avoid this, when updating the system registers related to the register bank function, it is recommended to disable interrupts by setting PSW.ID to 1.

For RBCR0 only, synchronization process by the SYNCI instruction is necessary to surely reflect the new setting of the register to the subsequent RESBANK instruction. The following shows an example of code for this.

```
DI                              //  Disable interrupt
LDSR       r20, 15, 2           //  Update RBCR0
SYNCI                           //  Synchronization of instruction fetching
RESBANK                         //  Execute RESBANK using updated RBCR0
```

(5) Reading a System Register by Using an STSR Instruction

After executing an LDSR instruction to update a system register, execute a SYNCP instruction before executing an STSR instruction for read operation. Since the SYNCP instruction waits until the execution of the preceding LDSR instruction is completed, it is ensured that the new register setting is read.

For the system registers shown in **Table 3.138**, the new register settings are read without synchronization process by SYNCP instruction.

(6) Referencing a System Register by the Subsequent Instruction

After executing an LDSR instruction to update a system register, execute a SYNCP instruction before executing an instruction to refer to the system register. Since the SYNCP instruction waits until the execution of the preceding LDSR instruction is completed, it is ensured that the new register setting is reflected.

(7) Use of EIRET and FERET Instructions in Synchronization Process

The EIRET and FERET instructions are defined as return instructions from the exception handler of their corresponding exception levels. These instructions can set any PC and PSW at the same time and therefore usable as instructions for synchronization process in this CPU.

To use an EIRET instruction for synchronization process, set appropriate values to EIPC and EIPSW and execute the instruction. To use an FERET instruction for synchronization process, set appropriate values to FEPC and FEPSW and execute the instruction.

Set values to EIPC and FEPC that is the PCs after these instructions are executed. Set values to EIPSW and FEPSW that is the PSW after these instructions are executed.

Note that these instructions are supervisor privileged instructions. That means, the state of PSW.UM can be changed only from 0 (clear) to 1 (set), supervisor mode to user mode.

For details on the EIRET and FERET instructions, see the *RH850G4MH User's Manual: Software*.

(8) Updating PSW.EBV and EBASE

EBASE is a register that indicates the vector address of the exception handler. This is enabled when PSW.EBV is set (to 1). The recommended updating procedure is as follows.

1. Set PSW.ID to 1
2. Clear PSW.EBV to 0 (updating at the same time with PSW.ID is possible)
3. Update EBASE
4. Set PSW.EBV to 1
5. Clear PSW.ID to 0 (updating at the same time with PSW.EBV is possible)

For updating PSW.UM in the procedure above as well, use of an EIRET or FERET instruction described in **Section 3.2.7.3(1)(a), Updating PSW.UM** is recommended.

3.2.8 Reset

3.2.8.1 Status of Registers After Reset

If a reset signal is input by a method defined by the hardware specifications, the program registers and system registers are placed in the status shown by the value after reset of each register in **Section 3.2.3, Register Set**, and program execution is started. Set the contents of each register to an appropriate value in the program.

The CPU executes a reset to start execution of a program from the reset address specified by **Section 3.2.4.4, Exception Handler Address**.

Note that because the PSW.ID bit is set to 1 immediately after a reset, conditional EI level exceptions will not be acknowledged. To acknowledge conditional EI level exceptions, clear the PSW.ID bit to 0.

3.2.9 Ensuring Coherency after Code Flash Programming

The CPU has an efficient instruction cache and data buffer for the code flash area.

Therefore, after using self-programming to program the code flash memory, clear the instruction cache and data buffer to ensure coherency. The instruction cache and data buffer can be cleared by using the ICCTRL register and the RDBCR register, respectively. The entire CPU's instruction cache and data buffer can also be cleared at the same time by using the TM_CC register. For details on the ICCTRL and RDBCR register, see **Section 3.2.3.7, Cache Operation Function Registers** and **Section 3.2.3.9, Hardware Function Registers**. For details on the TM_CC register, see **Section 3.2.9.1, Registers**.

3.2.9.1 Registers

(1) Register Base Address

The GCFU base address is listed in the following table. The GCFU register addresses are given as offsets from the base addresses.

Table 3.139 Register Base Address

Base Address Name	Base Address	Bus Group
<GCFU_base>	FFFB 1400_H	P-Bus Group 0

(2) Clock Supply

Clock supply by and to GCFU is listed in the following table.

Table 3.140 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
GCFU	PCLK	CLK_HBUS

(3) List of Registers

Table 3.141 List of Registers

Module Name	Register Name	Symbol	Address	Access
GCFU	Cache Clear Operation Register	TM_CC	<GCFU_base> + 08_H	32

(4) TM_CC — Cache Clear Operation Register

The TM_CC register is used to issue requests for clearing of the unit (instruction cache or data buffer) that holds data from the code flash.

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ICCMK	DCCMK	—	ICCLR	DCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Table 3.142 TM_CC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ICCMK	Instruction Cache Clear Mask Writing 1 to this bit masks cache clear requests upon updating of the memory mapping enable register GCFU.TM_ME and CFU.TM_ME. Cache clear requests by writing 1 to the GCFU.TM_CC.ICCLR bit are not masked. 0: Do not mask cache clear requests 1: Mask cache clear requests
3	DCCMK	Data Cache (Buffer) Clear Mask Writing 1 to this bit masks cache clear requests upon updating of the memory mapping enable register GCFU.TM_ME and CFU.TM_ME. Cache clear requests by writing 1 to the GCFU.TM_CC.DCCLR bit are not masked. 0: Do not mask cache clear requests 1: Mask cache clear requests
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ICCLR	Instruction Cache Clear Request Writing 1 to this bit issues an instruction cache clear request. Writing 0 is ignored. The read value is 1 when this bit is read before completion of the cache clear request, and 0 when it is read after completion of the cache clear request. **Note:** If 1 has been written to this bit and then 1 is written again before completion of the cache clear request, the second write is ignored. (The cache clear request is not executed by the second write.) If 1 has been written to this bit and then 0 is written before completion of the cache clear request, the cache clear request is not canceled. To execute cache clear requests continuously, be sure to check that execution of the preceding cache clear request has been completed (by reading this bit and checking that the readout value is 0) before executing the next cache clear request. 0: Indicates that a cache clear request is not currently being processed, or processing of a cache clear request has been completed (when read). 1: Issues a cache clear request (when written). Indicates that a cache clear request is currently being processed (when read).

Table 3.142 TM_CC Register Contents (2/2)

Bit Position	Bit Name	Function
0	DCCLR	Data Buffer Clear Request Writing 1 to this bit issues a data buffer clear request. Writing 0 is ignored. The read value is 1 when this bit is read before completion of the data buffer clear request, and 0 when it is read after completion of the data buffer clear request. **Note:** If 1 has been written to this bit and then 1 is written again before completion of the data buffer clear request, the second write is ignored. (The data buffer clear request is not executed by the second write.) If 1 has been written to this bit and then 0 is written before completion of the data buffer clear request, the data buffer clear request is not canceled. To execute data buffer clear requests continuously, be sure to check that execution of the preceding data buffer clear request has been completed (by reading this bit and checking that the readout value is 0) before executing the next data buffer clear request. 0: Indicates that a data buffer clear request is not currently being processed, or processing of a data buffer clear request has been completed (when read). 1: Issues a data buffer clear request (when written). Indicates that a data buffer clear request is currently being processed (when read).

3.3 Inter-CPU Overview

This subsection contains a Barrier-Synchronization (BARR), Inter-Processor Interrupt (IPIR) and Time Protection Timer (TPTM). The first part of this subsection describes this product's specific properties, such as the number of units, register base addresses, etc. The remainder of the subsection shows the functions and registers.

3.3.1 Features

3.3.1.1 Number of Units

Table 3.143 Number of Units

Product Name		RH850/U2A-EVA	RH850/U2A16	RH850/U2A8	RH850/U2A6
BARR	Number of units	1	1	1	1
	Name of unit	BARR			
IPIR	Number of units	1	1	1	1
	Number of channels for each CPU	4	4	4	4
	Name of unit	IPIR			
TPTM	Number of units	1	1	1	1
	Number of channels for each CPU	2 interval timers 1 free-run timer 2 Up timers	2 interval timers 1 free-run timer 2 Up timers	2 interval timers 1 free-run timer 2 Up timers	2 interval timers 1 free-run timer 2 Up timers
	Name of unit	TPTM			

3.3.1.2 Register Base Address

The base address of BARR, IPIR and TPTM are listed in the following table. BARR, IPIR and TPTM register addresses are given as offsets from these base addresses.

Table 3.144 Register Base Addresses

Base Address Name	Base Address	Bus Group
<BARR_base>	FFFB 8000$_H$	I-Bus Group1
<IPIR_base>	FFFB 9000$_H$	I-Bus Group0
<TPTM_base>	FFFB B000$_H$	I-Bus Group2

3.3.1.3 Clock Supplies

The clock supplies are listed in the following table .

Table 3.145 Clock Supplies

Unit Name	Unit Clock Name	Supply Clock Name
BARR	cpu_clk	CLK_CPU
IPIR	cpu_clk	CLK_CPU
TPTM	cpu_clk	CLK_CPU

3.3.1.4 Interrupt Requests

The interrupt requests are listed in the following table.

Table 3.146 Interrupt Requests (1/2)

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTIPIR0	ipir_int_ch0	IPIR CH0 interrupt	EIINT0	—	—
INTIPIR1	ipir_int_ch1	IPIR CH1 interrupt	EIINT1	—	—
INTIPIR2	ipir_int_ch2	IPIR CH2 interrupt	EIINT2	—	—
INTIPIR3	ipir_int_ch3	IPIR CH3 interrupt	EIINT3	—	—
FEINT or INTTPTM*1	TPTM_IRQ	TPTM interval interrupt	FEINT or EIINT31*1	—	—
INTTPTMU00	INTTPTMU00	TPTM up timer interrupt for PE0 with comparison value 0	EIINT209	—	Group3-26
INTTPTMU01	INTTPTMU01	TPTM up timer interrupt for PE0 with comparison value 1	EIINT210	—	Group3-27
INTTPTMU02	INTTPTMU02	TPTM up timer interrupt for PE0 with comparison value 2	EIINT211	—	—
INTTPTMU03	INTTPTMU03	TPTM up timer interrupt for PE0 with comparison value 3	EIINT212	—	—
INTTPTMU10	INTTPTMU10	TPTM up timer interrupt for PE1 with comparison value 0	EIINT213	—	Group3-28
INTTPTMU11	INTTPTMU11	TPTM up timer interrupt for PE1 with comparison value 1	EIINT214	—	Group3-29
INTTPTMU12	INTTPTMU12	TPTM up timer interrupt for PE1 with comparison value 2	EIINT215	—	—
INTTPTMU13	INTTPTMU13	TPTM up timer interrupt for PE1 with comparison value 3	EIINT216	—	—
INTTPTMU20	INTTPTMU20	TPTM up timer interrupt for PE2 with comparison value 0	EIINT217	—	Group3-30
INTTPTMU21	INTTPTMU21	TPTM up timer interrupt for PE2 with comparison value 1	EIINT218	—	Group3-31

Table 3.146 Interrupt Requests (2/2)

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTTPTMU22	INTTPTMU22	TPTM up timer interrupt for PE2 with comparison value 2	EIINT219	—	—
INTTPTMU23	INTTPTMU23	TPTM up timer interrupt for PE2 with comparison value 3	EIINT220	—	—
INTTPTMU30	INTTPTMU30	TPTM up timer interrupt for PE3 with comparison value 0	EIINT221	—	Group3-32
INTTPTMU31	INTTPTMU31	TPTM up timer interrupt for PE3 with comparison value 1	EIINT222	—	Group3-33
INTTPTMU32	INTTPTMU32	TPTM up timer interrupt for PE3 with comparison value 2	EIINT223	—	—
INTTPTMU33	INTTPTMU33	TPTM up timer interrupt for PE3 with comparison value 3	EIINT224	—	—

Note 1. This interrupt can be selected for use as FEINT or EIINT31. See **Section 6.3.15, TPTMSEL — TPTM Interrupt FE EI Select Register**.

3.3.1.5 Reset Sources

The reset sources are shown below. These modules are initialized by the following reset sources.

Table 3.147 Reset Sources

Unit Name	Register Name	Reset Condition					
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset
BARR	All registers	√	√	√	√	—	—
IPIR	All registers	√	√	√	√	—	—
TPTM	All registers	√	√	√	√	—	—

3.3.1.6 External Input/Output Signals

This module has no external input/output signals.

3.3.2 Processor Element Identifier

The PEID, the ID number of a processor element, can be read from the PEID register. Which CPU core performs a specific program can be known by referring to the PEID. The following shows the PEIDs of this product.

For the PEID register, see **3.2.3.2, Basic System Registers**.

CPU Core	PEID
CPU0 (PE0)	000_B
CPU1 (PE1)	001_B
CPU2 (PE2)	010_B
CPU3 (PE3)	011_B

3.4 Inter-Processor Interrupt

3.4.1 Inter-Processor Interrupt Overview

Inter Processor Interrupt Register (IPIR) is a function that controls fast interrupt requests between PEs. Use of IPIR achieves faster processing of inter-PE interrupts than setting a request flag for the INTC2 interrupt channel by using software.

IPIR has the following features.

- Support of inter-PE interrupt function of 4 channels
- Level detection of interrupts. Edge detection is not supported.
- Accessible from all clusters and all PEs
- Identification of interrupt request source PE is possible.
- Unintended inter-PE interrupts can be prevented by masking interrupt requests.
- SET1, CLR1, and NOT1 can be executed as atomic operation instructions to IPIR.
- Support of address EDC function
- Support of data ECC function
- Support of guard function to prevent unauthorized access.
  - Read-modify-write operations using the LD instruction and the ST instruction are also possible, but not as atomic operations.

3.4.2 Inter-Processor Interrupts Registers

3.4.2.1 List of Registers

Table 3.148 List of Registers

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
IPIR	IPIRn*2 Inter-PE interrupt enable self-register*1	IPInENS	<IPIR_base> + 000_H + 020_H × n	8	IPIGPROT0_Rn	—
	IPIRn Inter-PE interrupt flag self-register*1	IPInFLGS	<IPIR_base> + 004_H + 020_H × n	8	IPIGPROT0_Rn	—
	IPIRn Inter-PE interrupt clear self-register*1	IPInFCLRS	<IPIR_base> + 008_H + 020_H × n	8	IPIGPROT0_Rn	—
	IPIRn Inter-PE interrupt request self-register*1	IPInREQS	<IPIR_base> + 010_H + 020_H × n	8	IPIGPROT0_Tn	—
	IPIRn Inter-PE interrupt request clear self-register*1	IPInRCLRS	<IPIR_base> + 014_H + 020_H × n	8	IPIGPROT0_Tn	—
	IPIRn Inter-PE interrupt enable register m*3	IPInENm	<IPIR_base> + 800_H + 020_H × n + 100_H × m	8	IPIGPROT0_4	—
	IPIRn Inter-PE interrupt flag register m*3	IPInFLGm	<IPIR_base> + 804_H + 020_H × n + 100_H × m	8	IPIGPROT0_4	—
	IPIRn Inter-PE interrupt clear register m*3	IPInFCLRm	<IPIR_base> + 808_H + 020_H × n + 100_H × m	8	IPIGPROT0_4	—
	IPIRn Inter-PE interrupt request register m*3	IPInREQm	<IPIR_base> + 810_H + 020_H × n + 100_H × m	8	IPIGPROT0_4	—
	IPIRn Inter-PE interrupt request clear register m*3	IPInRCLRm	<IPIR_base> + 814_H + 020_H × n + 100_H × m	8	IPIGPROT0_4	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses the IPI0REQS register, PE1 can also access the IPI0REQ1 register.

Note 2. n = 0 to 3. n is the channel number of IPIR.

Note 3. m = 0 to 3. m is the PEID.

3.4.2.2 Self Region

The self region contains the following five types of registers.

- IPInENS —IPIRn Inter-PE interrupt enable self-register
- IPInFLGS —IPIRn Inter-PE interrupt flag self-register
- IPInFCLRS —IPIRn Inter-PE interrupt clear self-register
- IPInREQS — IPIRn Inter-PE interrupt request self-register
- IPInRCLRS —IPIRn Inter-PE interrupt request clear self-register

The self-registers are virtual registers that do not physically exist. Access from each PE to self-registers is routed to the actual register corresponding to the access source PE. **Table 3.149** lists the access source PEs and routing destination registers. For the functions of each register bit, refer to the specifications of the routing destination register. When masters except PEx access the self registers, the register returns 0, write access is ignored and error response will be notified.

It is basically assumed that the IPIR registers are accessed from PEs via self region when PEs use the IPIR function. This allows PEs to use the same code because it is not necessary to specify different register addresses for each PE.

It is also possible to directly access registers by specifying the address of the actual register without using the self-registers. In this case, the purpose is assumed to be to check the status of registers for another PE, or to reference the individual registers by a debugging tool.

Table 3.149 Self Region Register Routing List

Self Register	Access Source PE			
	PE0	PE1	PE2	PE3
IPInENS	IPInEN0	IPInEN1	IPInEN2	IPInEN3
IPInFLGS	IPInFLG0	IPInFLG1	IPInFLG2	IPInFLG3
IPInFCLRS	IPInFCLR0	IPInFCLR1	IPInFCLR2	IPInFCLR3
IPInREQS	IPInREQ0	IPInREQ1	IPInREQ2	IPInREQ3
IPInRCLRS	IPInRCLR0	IPInRCLR1	IPInRCLR2	IPInRCLR3

3.4.2.3 IPInENm — IPIRn Inter-PE Interrupt Enable Register m

This register sets the transmitting PE allowed to issue inter-PE interrupt requests to PEm.

This register is used to enable inter-PE interrupt requests of PE (PEx) by the receiving PE (PEm) itself.

Access: This register can be read or written in 8-bit units.

Address: <IPIR_base> + 800_H + 020_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EN[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.150 IPInENm Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	EN[3:0]	Inter-PE Interrupt Enable. Write 1 to the x-th bit to enable the issuance of inter-PE interrupt requests from PEx to PEm. Write 0 to the x-th bit to disable the issuance of inter-PE interrupt requests from PEx to PEm. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.4.2.4 IPInFLGm — IPIRn Inter-PE Interrupt Flag Register

This register indicates the transmitting PE that issued an inter-PE interrupt request to PEm.

This register is used to distinguish the requesting PE by PEm when PEm has received an inter-PE interrupt request.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <IPIR_base> + 804_H + 020_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FLG[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 3.151 IPInFLGm Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	FLG[3:0]	Inter-PE Interrupt Request Flag. This register indicates the status of Inter-PE Interrupt request from other PEs. The x-th bit is automatically updated according to the IPInREQx[m] change while the value of the IPInENm[x] bit is 1. The x-th bit is not set if IPInENm[x] is 0, even if IPInREQx[m] is set. The x-th bit is also cleared when the IPInFCLRm[x] bit is set. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.4.2.5 IPInFCLRm — IPIRn Inter-PE Interrupt Clear Register m

This register clears inter-PE interrupt requests to PEm.

This register is used to clear the request flag (IPInFLGm) and the request (IPInREQx) after receiving the Inter-PE interrupt request from another PE.

Access: This register is write-only register that can be written in 8-bit units.

Address: <IPIR_base> + 808_H + 020_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FCLR[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 3.152 IPInFCLRm Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	FCLR[3:0]	Inter-PE Interrupt Request Flag Clear IPInFLGm[x] and IPInREQx[m] can be cleared by writing 1 to this bit. Writing 0 is ignored. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.4.2.6 IPInREQm — IPIRn Inter-PE Interrupt Request Register m

This register controls Inter-PE interrupt request from PEm to other PEs.

Access: This register can be read or written in 8-bit units.

Address: <IPIR_base> + 810_H + 020_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	REQ[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.153 IPInREQm Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	REQ[3:0]	When 1 is written to the x-th bit while the value of the IPInENx[m] bit is 1: The value of the x-th bit becomes 1. The high level is output to the inter-PE interrupt request for PEx, and IPInFLGx[m] is automatically set to 1. When 1 is written to the x-th bit while the value of the IPInENx[m] bit is 0: The value of the x-th bit becomes 1. There are no other operations. When 0 is written to the x-th bit: Writing 0 is ignored. When read: The register value is read out. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.4.2.7 IPInRCLRm — IPIRn Inter-PE Interrupt Request Clear Register m

This register clears inter-PE interrupt requests for other PEs by PEm.

This register is assumed to be used to clear inter-PE interrupt requests by the transmitting PE.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <IPIR_base> + 814_H + 020_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RCLR[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 3.154 IPInRCLRm Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	RCLR[3:0]	Inter-PE Interrupt Request Clear Writing 1 to the x-th bit clears the IPInREQm[x] bit. If the value of the IPInENx[m] bit is 1, the IPInFLGx[m] bit is also cleared at the same time. The transmitting PE allowed to issue inter-PE interrupt requests by the IPInENm register can also cancel an inter-PE interrupt request by using this bit, but the receiving PE may in some cases accept the inter-PE interrupt request while the cancellation processing takes place. Writing 0 is ignored. This bit always returns 0 when read. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.4.3 Inter-Processor Interrupt Function

3.4.3.1 Initial Setting

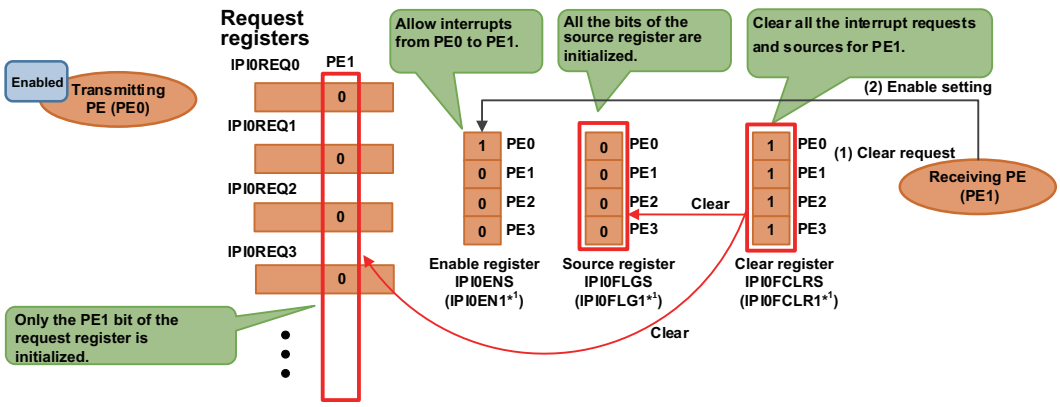
Initial setting of authorized PEs by the IPInENm register must be made before IPIR is used. An example of initial setting is given below.

(1) Initial Setting by Receiving PE

Figure 3.40 shows an example of the initial setting of IPIR channel 0 by PE1, which is a receiving PE.

First, if IPIR has already been used and the values of bits IPI0REQ0 to 3[1] and the value of the IPI0FLG1 register have changed from the initial values, PE1 writes $0F_H$ to the IPI0FCLRS (= IPI0FCLR1) register to clear bits IPI0REQ0 to 3[1] and the IPI0FLG1 register. If the values of bits IPI0REQ0 to 3[1] and the IPI0FLG1 register are the initial values, for example after hardware reset, this clearing operation is not required.

Next, PE1 writes 01_H to the IPI0ENS (= IPI0EN1) register to accept interrupt requests from PE0 to PE1. Like for the clear register, upon completion of setting of the enable register, PE1 can accept interrupt requests from PE0.



Note 1. Register actually accessed from PE1.

Figure 3.40 Example of Initial Setting of IPIR Channel 0 by Receiving PE (PE1)

(2) Initial Setting by Control PE

An example of initial setting from a PE other than a receiving PE is shown below. In this section, the PE that performs the initial setting is called the control PE for the sake of convenience. **Figure 3.41** shows an example of the initial setting of IPIR channel 0 by PE3, which is the control PE. In this figure, bits PE0 to 3 of each register are omitted.

First, if IPIR has already been used and the values of bits IPI0REQ0 to 3[1] and the value of the IPI0FLG1 register have changed from the initial values, PE3 writes $0F_H$ to the IPI0FCLR1 register to clear bits IPI0REQ0 to 3[1] and the IPI0FLG1 register. If the values of bits IPI0REQ0 to 3[1] and the IPI0FLG1 register are the initial values, for example after hardware reset, this clearing operation is not required.

Next, PE3 writes 01_H to the IPI0EN1 register to accept interrupt requests from PE0 to PE1. Like for the clear register, upon completion of setting of the enable register, PE1 can accept interrupt requests from PE0.

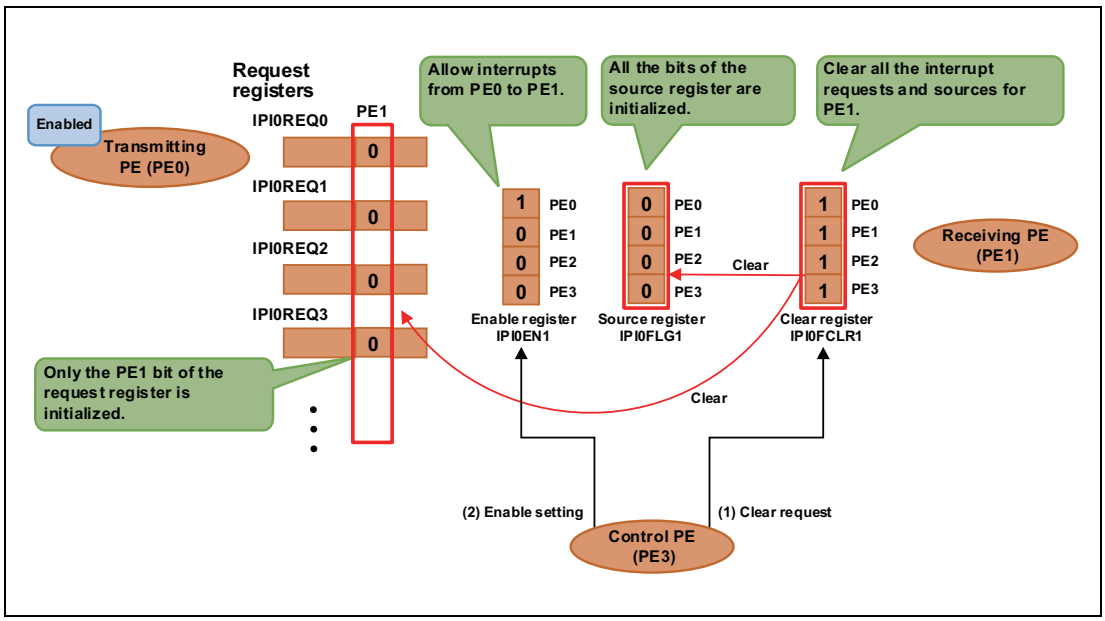


Figure 3.41 Example of Initial Configuration of IPIR Channel 0 by Control PE (PE3)

3.4.3.2 Inter-PE Interrupt Request

Figure 3.42 shows an example of the operation when PE0 sends an inter-PE interrupt request to PE1 using IPIR channel 0. In this figure, bits PE0 to 3 of each register are omitted. In this example, the initial setting is set to IPI0EN1 in advance to enable inter-PE interrupt requests from PE0. For an operation example of the initial setting, see **Section 3.4.3.1, Initial Setting**. The following describes an example of accessing the registers through the self-register.

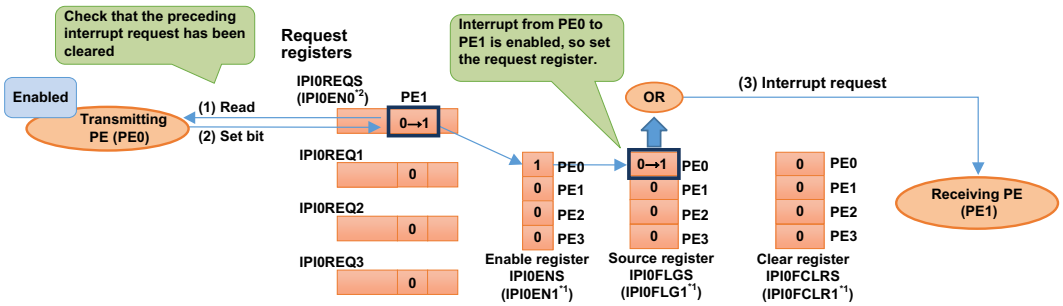
First, PE0, which is the transmitting PE, reads the IPI0REQS (= IPI0REQ0) register and checks that the value of the IPI0REQS[1] (= IPI0REQ0[1]) bit is 0_B. If the value of the IPI0REQS[1] (= IPI0REQ0[1]) bit is 1_B, this means that the previous inter-PE interrupt request from PE0 to PE1 has not been accepted by PE1, and thus a new inter-PE interrupt request cannot be issued. If it is confirmed that the value of the IPI0REQS[1] (= IPI0REQ0[1]) bit is 0_B and thus an inter-PE interrupt request from PE0 to PE1 is enabled, PE0 sets the IPI0REQS[1] (= IPI0REQ0[1]) bit to 1_B to issue an inter-PE interrupt request from PE1. Because inter-PE interrupts from PE0 to PE1 are enabled by the IPI0EN1 register, when PE0 sets the IPI0REQS[1] (= IPI0REQ0[1]) bit to 1_B, the IPI0FLG1[1] bit is automatically set to 1_B, and an inter-PE interrupt request signal is output to PE1 from IPIR.

Figure 3.43 shows an example of the operation when PE1 receives an inter-PE interrupt request from PE0 using IPIR channel 0. When PE1, which is the receiving PE, receives an inter-PE interrupt request signal, it reads the IPI0FLGS (= IPI0FLG1) register, and because the value of the IPI0FLGS[0] (= IPI0FLG1[0]) bit is 1B, it recognizes that the inter-PE interrupt request has been sent from PE0. But if the request side clears the request, the value of it is 0B. After verifying the source of the inter-PE interrupt, PE1 sets the IPI0FCLRS[0] (= IPI0FCLR1[0]) bit to 1B, and the processing transitions to the inter-PE interrupt processing. The IPI0REQS[1] (= IPI0REQ0[1]) bit and the IPI0FLGS (= IPI0FLG1) bit are automatically cleared when the IPI0FCLRS[0] (= IPI0FCLR1[0]) bit is set to 1B by PE1.

Figure 3.44 shows the operation flow from initial setting of IPIR by the receiving PE (PE1), to sending of an inter-PE interrupt request by the transmitting PE (PE0), and completion of reception of that inter-PE interrupt request by the receiving PE (PE1).

Note that if a new inter-PE interrupt request is generated to the same receiving PE before the source register is cleared by the receiving PE with the clear register, the bit corresponding to the new transmitting PE in the source register will be set to 1B, but the inter-PE interrupt request signal being retained high, the second and subsequent inter-PE interrupt requests will not be output. Therefore, the processing when multiple sources occur when the receiving PE has received an inter-PE interrupt request must be controlled by software.

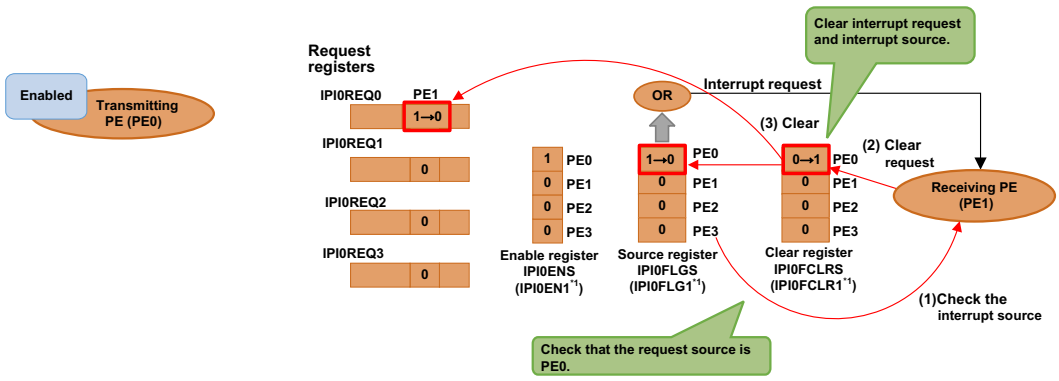
The transmitting PE can detect whether the preceding inter-PE interrupt request was successfully received by the receiving PE, by checking the value of the request register. To monitor the request register with a polling loop, it is recommended to curb the bus load by executing the snooze instruction within the polling loop in order to avoid the bus system occupation for extended periods of time.



Note 1. Register actually accessed from PE1.

Note 2. Register actually accessed from PE0.

Figure 3.42 Example of Inter-PE Interrupt Transmission Operation



Note 1. Register actually accessed from PE1.

Figure 3.43 Example of Inter-PE Interrupt Reception Operation

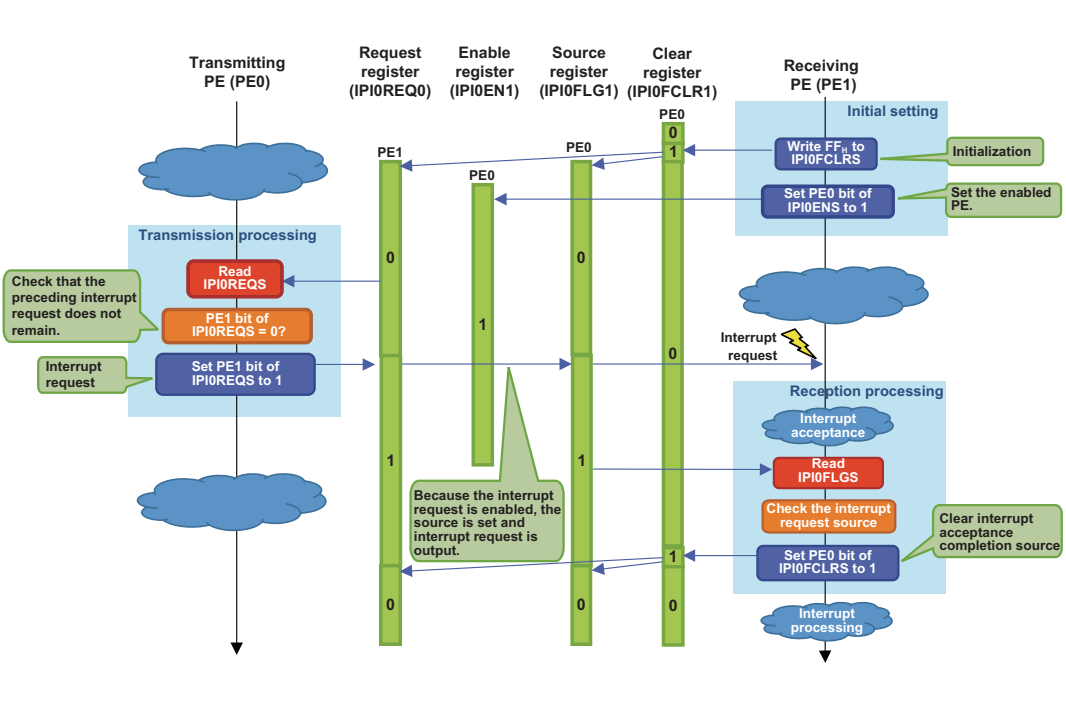


Figure 3.44 Operation Flow of Inter-PE Interrupt

(1) Sample Code

(a) Transmission Processing

Figure 3.45 shows a sample flowchart of the inter-PE interrupt request transmission processing using IPIR channel 0.

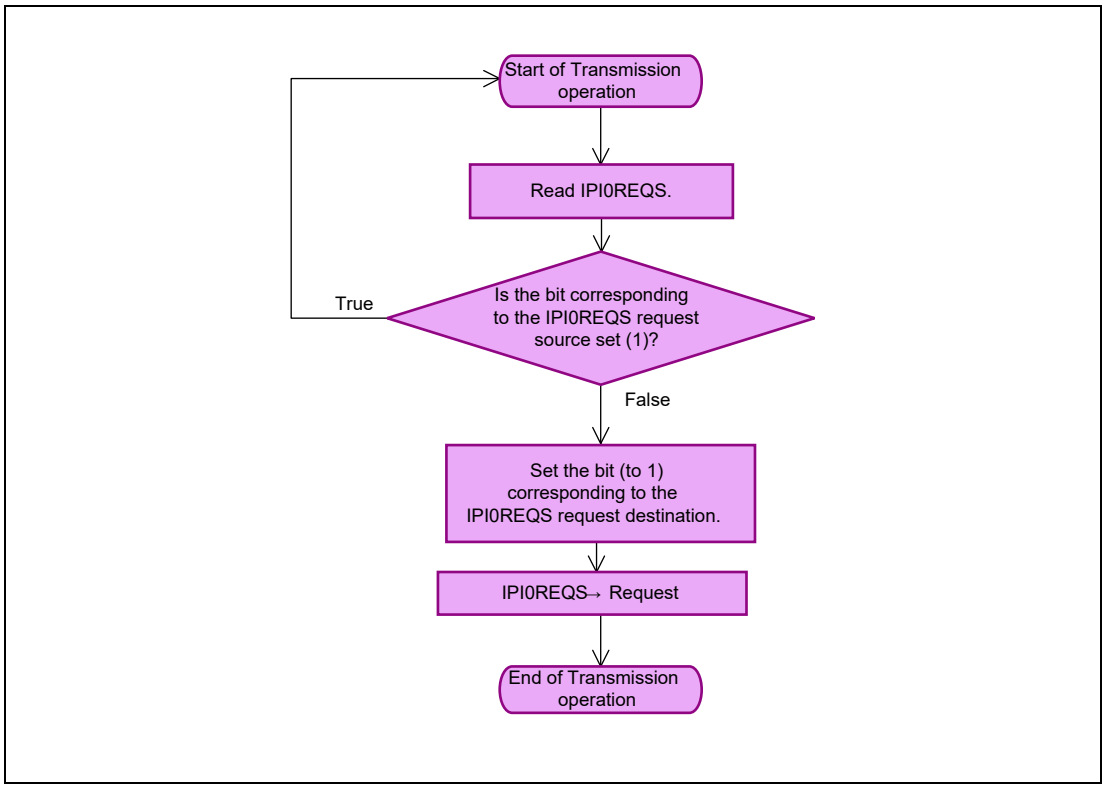


Figure 3.45 Flowchart of Inter-PE Interrupt Request Transmission Processing

(b) Reception Processing

Figure 3.46 shows a sample flowchart of the inter-PE interrupt request reception processing using IPIR channel 0. To prevent source flag loss, the source flag & source clear register configuration is used.

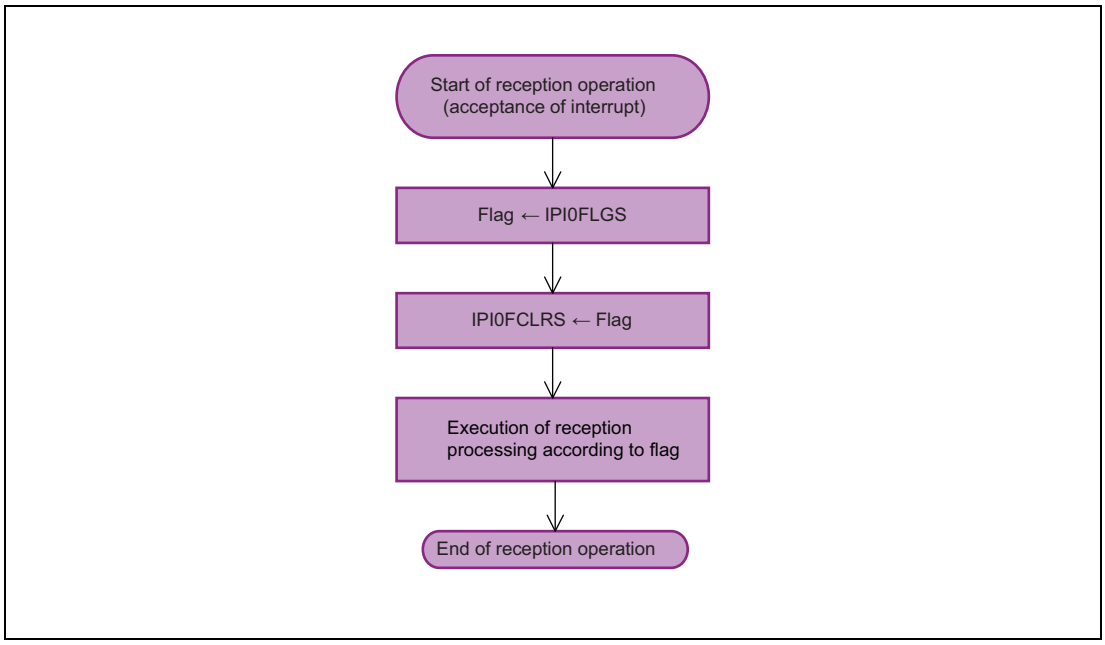


Figure 3.46 Flowchart of Inter-PE Interrupt Request Transmission Processing

3.4.3.3 Request Mask Function

Inter-PE interrupt requests disabled by the IPInENm register setting are ignored. **Figure 3.47** shows an example of the operation when an inter-PE interrupt is requested from a disabled PE. In this figure, bits PE0 to 3 of each register are omitted. In this example, the initial setting is set to IPI0EN1 in advance to prohibit inter-PE interrupt requests from PE2 to PE1. For an operation example of the initial setting, see **Section 3.4.3.1, Initial Setting**.

Even if PE2 sets IPI0REQS[1] (= IPI0REQ2[1]) to 1B to request PE1 to issue an inter-PE interrupt, inter-PE interrupts from PE2 to PE1 are prohibited, so the IPInFLG1[2] bit remains 0B and no interrupt request signal is output.

If the transmitting PE disabled by the IPInENm register setting unintentionally writes 1B to the IPI0REQm register, the IPI0REQm register can be cleared by using the IPInRCLRm register. For details, see **Section 3.4.3.5, Inter-PE Interrupt Request Clear Function**.

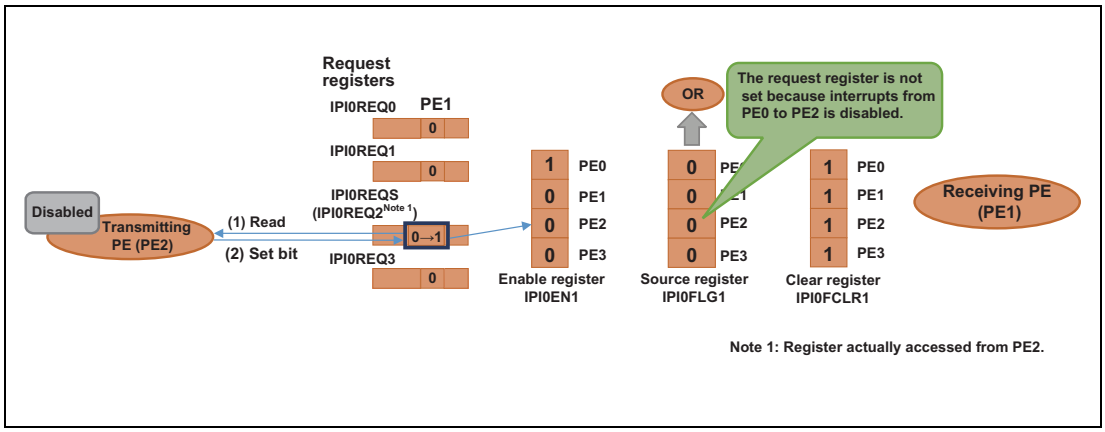


Figure 3.47 Request Mask Operation Example

3.4.3.4 Inter-PE Interrupt Requests of Multiple Systems

If the combination of the transmitting PE and receiving PE differs, multiple inter-PE interrupt requests can be output simultaneously on one channel of IPIR. For example it is possible to output inter-PE interrupt request A (PE0 to PE1) and inter-PE interrupt request B (PE1 to PE0) on channel 1 of IPIR simultaneously. Each register operates independently when the combination of the transmitting PE and receiving PE differs, so multiple inter-PE interrupt requests can be output simultaneously.

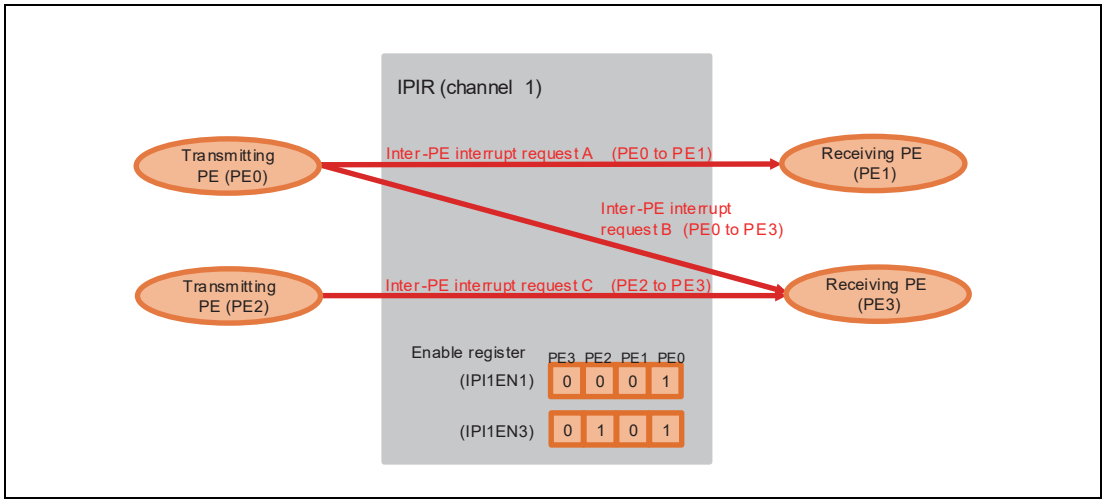


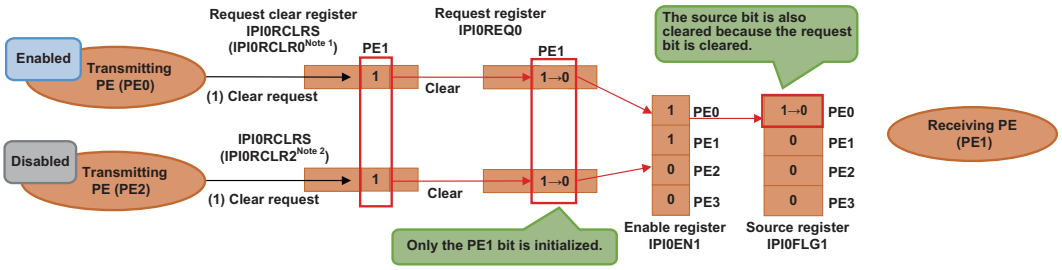
Figure 3.48 Example of Simultaneous Output of Inter-Processor Interrupt Requests

3.4.3.5 Inter-PE Interrupt Request Clear Function

The IPInREQm[x] bit can be cleared by setting the IPInRCLRm[x] bit to 1B, allowing you to cancel inter-PE interrupt requests from the transmitting PE. If inter-PE interrupt requests from the transmitting PE are enabled by the IPInENx[m] bit, the IPInFLGx[m] bit is also cleared at the same time as the IPInREQm[x] bit is cleared. However, the receiving PE may in some cases accept the inter-PE interrupt while the cancellation processing takes place.

Figure 3.49 shows an example of the operation to clear inter-PE interrupt requests from PE0 and PE2 to PE1. When PE0 writes 1B to the IPI0RCLRS[1] (= IPI0RCLR0[1]) bit, the IPI0REQ0[1] bit is cleared. Moreover, because inter-PE interrupt requests from PE0 to PE1 are enabled by the IPI0EN1 register setting, IPI0FLG1[0] is also cleared.

When PE2 writes 1B to the IPI0RCLRS[1] (= IPI0RCLR2[1]) bit, the IPI0REQ2[1] bit is cleared. Here, inter-PE interrupt requests from PE2 to PE1 are disabled by the IPI0EN1 register setting, so the IPI0RCLR2 register does not affect the value of the IPI0FLG1 register.



Note 1. Register actually accessed from PE0.

Note 2. Register actually accessed from PE2.

Figure 3.49 Example of Inter-PE Interrupt Request Clear Function Operation

3.4.4 Usage When Virtualization Enabled

3.4.4.1 Overview

In this section, VM means "Host or Guest".

IPIR is designed to support interruption from PE to PE. More exactly, it is from a source PE to INTC1 of destination PE. Bits of IPIR registers are prepared for each PE, not for each VM. When it is used for interruption from a VM to another, 2 more settings are required.

Destination VM is specified using EIBD register of INTC1. Source VM configuration is indirectly realized by guard function assigning different SPIDs to VMs. Then, registers for a PE of the channel can be written by just one VM of the PE. Read access from non-source VMs is not harmful in normal use. IPIR cannot tell source VM to destination VM. If it is necessary to know who sent an interrupt, send GPID in some method or avoid time-dependent channel assignment.

It is also possible to use IPIR for intra-PE interrupt, interrupt from a VM to another in the same PE.

3.4.4.2 Setting Example

Assumptions:

Considering 6 masters shown in **Figure 3.50**, **Table 3.155**.

Three interrupts are using channel n, from Guest 0 of PE0 (SPID = 8) to Guest 1 of PE0 (SPID = 16), from Guest 0 of PE1 (SPID = 9) to Guest 1 of PE0, from Guest 0 of PE0 to Guest 1 of PE1 (SPID =1 7).

Host of PE0 (SPID = 0) and Host of PE1 (SPID = 1) are error handling PEs.

Table 3.156 shows each register can be accessed by which VMs when guard registers are set as shown in **Table 3.155**. Access control is done using PEID and SPID. This is visualized in **Figure 3.51**. In the figure, a black arrow from a Master (green box) indicates write access. If it reached a register (blue box), the register can be written by the master. If not, the access is blocked by PEID matching (yellow box) or by SPID matching (red box).

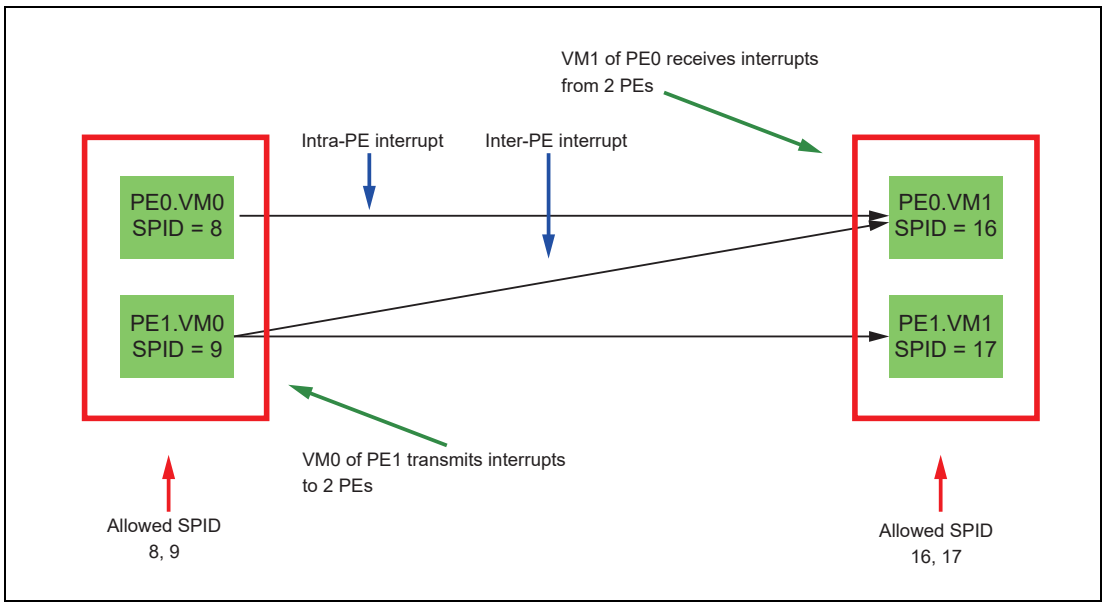


Figure 3.50 Example of Inter-VM Interrupt

Table 3.155 Setting Example of IPIR Registers

Register	Value	Description
IPInEN0	03_H	PE0 receives interrupts from PE0 and PE1.
IPInEN1	02_H	PE1 receives interrupts from PE1.
IPIGPROT0_Rn	$0000\ 0101_H$	Read access: Allowed Write access: Allowed if specified in IPIGPROT1_n
IPIGPROT1_Rn	$0003\ 0000_H$	Allowed SPID: 16, 17
IPIGPROT0_Tn	$0000\ 0101_H$	Read access: Allowed Write access: Allowed if specified in IPIGPROT1_n
IPIGPROT1_Tn	$0000\ 0300_H$	Allowed SPID: 8, 9
IPIGPROT0_4	$0000\ 0101_H$	Read access: Allowed Write access: Allowed if specified in IPIGPROT1_4
IPIGPROT1_4	$0000\ 0003_H$	Allowed SPID: 0, 1
EIBDn (INTC1 of PE0)	00008100_H	Channel n is bound to Guest 1
EIBDn (INTC1 of PE1)	00008101_H	Channel n is bound to Guest 1

Table 3.156 Summary of Access Control

		PE0			PE1		
		Host	Guest 0	Guest 1	Host	Guest 0	Guest 1
Register	Access	SPID = 0	SPID = 8	SPID = 16	SPID = 1	SPID = 9	SPID = 17
IPInEN0, IPInFLG0, IPInFCLR0	Via Self Registers	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)	Not allowed (PEID)
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)
IPInEN1, IPInFLG1, IPInFCLR1	Via Self Registers	Not allowed (SPID)	Not allowed (SPID)	Not allowed (PEID)	Not allowed (SPID)	Not allowed (SPID)	Allowed
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)
IPInREQ0, IPInRCLR0	Via Self Registers	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)	Not allowed (PEID)	Not allowed (SPID)
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)
IPInREQ1, IPInRCLR1	Via Self Registers	Not allowed (SPID)	Not allowed (PEID)	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)

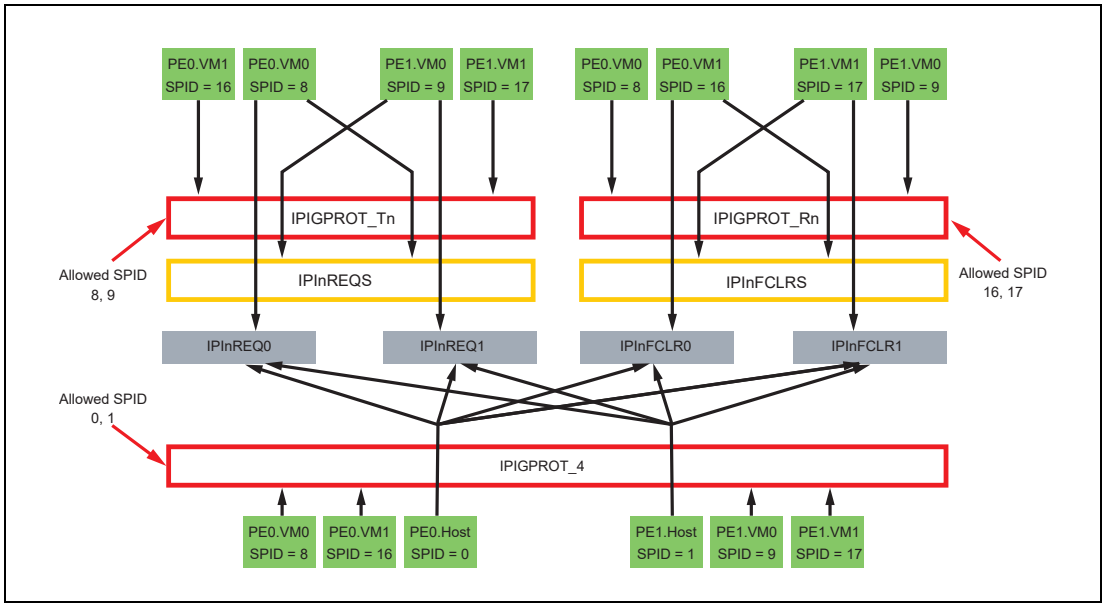


Figure 3.51 Example of per-VM Write Access Control

3.5 Mutual Exclusion

This CPU provides instructions that enable shared resources to be controlled mutually exclusively from multiple programs when the system is operating in a multi- processing environment.

CAUTION

Embedded CPUs in a single-processor configuration use a programming model in which data coherence is maintained by disabling the acknowledgment of maskable interrupts. This is a very easy and sure method of maintaining data coherence, but naturally in a multi-processor, multiple programs might be executing and attempting to use the data at the same time. In this case it is not possible to maintain data coherence simply by disabling maskable interrupt acknowledgment.

3.5.1 Mutual Exclusion Overview

The local RAM and cluster RAM are available as resources for exclusive control. As atomic operation instructions, the instructions of LDL/STC, CAXI, SET1, CLR1, and NOT1 can be performed for the local RAM and cluster RAM. These registers are also accessible by the LD and ST instructions but are not regarded as atomic operation.

3.5.2 Mutual Exclusion Function

3.5.2.1 Shared Data that does not Require Mutual Exclusion Processing

This CPU maintains data access coherence for the following types of data access even in a multi-processor environment.

- Access in which the data is aligned to the size that matches the data type (aligned access)
  - LD, ST, SLD, SST, LDL, STC, and bit manipulation instructions (TST1)
- Access by using a bit manipulation instruction (SET1, CLR1, or NOT1) (read-modify-write)
- Access by using the CAXI instruction (read-modify-write)

NOTE

Whether an instruction is atomically executed or not depends on the data format. Refer to **Section 3.2.2.6, Data Types**.

With some exceptions, mutual exclusion is achieved by using these types of data access. In other words, it is guaranteed that while one CPU is executing the instructions that perform the above data accesses, another CPU is not accessing the data. This is known as an instruction being executed atomically or an instruction guaranteeing the atomicity.

Note that the atomic execution of an instruction means that a data access bus transaction completes with no disruption; it does not necessarily mean that a series of transactions has been completed.

CAUTION

The extent to which coherency is guaranteed might be limited, depending on the hardware specifications. For example, for some memories, coherency might not be guaranteed even if aligned access is used. For details, see Section 3.5.1, Mutual Exclusion Overview and Section 3.9.3, Accesses to Registers by Bit-Manipulation Instructions.

3.5.2.2 Operation of the LDL and STC Instructions

The LDL and STC instructions can be used to obtain atomic read-modify-write operations for accurate processing in the updating of memory by multicore systems. The LDL and STC instructions operate as follows. For the operation of the LDL and STC instructions, refer to the *RH850G4MH User's Manual: Software*.

(1) Link

Only one link (LLbit) can be created per CPU. The link includes information on the address for which it was created, and control is applied according to whether an STC instruction succeeds or fails at this address and whether the link is lost. The link also includes the data size information when it the link is created and therefore any STC instruction which has a data size different from that of the LDL instruction that created a link always fails, and the link is deleted.

(2) Link generation

Each CPU is capable of generating a link to the local RAM and the cluster RAM.

Executing the LDL instruction on the target RAM for the operation leads to the link address being registered, the link flag being set, and a link being generated in response to reading by the instruction.

(a) The local RAM for the given processor

(b) The cluster RAM

Each CPU is capable of generating a link to either (a) or (b).

(3) Success in storing

After a link has been generated, storing will only proceed in response to executing an STC instruction corresponding to the generated link.

(4) Failure in storing

If a link is lost, storing does not proceed even when an STC instruction for the corresponding address is processed. Storing also does not proceed when an STC instruction that does not correspond to the link is processed.

(5) Condition for successful storing

If the following condition is met, the STC instruction is judged to be for the address corresponding to the link.

– The address and size for the LDL instruction that generated the link matches that for the STC instruction.

(6) Loss of the link

A link is lost when certain event or address conditions are satisfied. **Table 3.157** shows the link loss conditions. A link is lost if any of the conditions shown in this table satisfied.

Table 3.157 Link Loss Conditions

Event Condition	Remark
Store operation to a 32-byte-aligned address range that includes the address of the existing link*1	The following operations by both the CPU for which the link was generated and another bus master (e.g. CPU) correspond. • ST, SST, STC and STV instructions • SET1, NOT1, CLR1, and CAXI instructions • PREPARE, PUSHSP and STM.MP instructions
Execution of STC instruction for a location corresponding to the existing link within its own CPU	The corresponding link (for (1) or (2) above) will be lost whether the result of the instruction was success or fail.
Execution of LDL instruction for a location corresponding to the existing link within its own CPU	The link generated by a CPU in response to its preceding LDL instruction will be lost, and the link in response to the following LDL instruction is generated.
Execution of CLL instruction	
Exception acknowledgment	
Execution of EIRET instruction	
Execution of FERET instruction	
Bus access error of LDL	

Note 1. In the local RAM, if the store instruction except STC/CAXI instruction is executed, the Link is not always lost. Therefore use the program flow which does not need that these instruction cause the Link Loss. For example, in the sample code (7), after reading the Lock variable using the LDL instruction, by executing the STC instruction only if nobody has the Lock, the program flow does not need the Link Loss caused by the store instruction for the Lock Release.

(7) Sample Code

The sample code of a spinlock executed by using the LDL.W and STC.W instructions is shown below.

Lock Acquisition

```
                MOV             lock_adr, r20
Lock:           LDL.W           [r20], r21
                CMP             r0, r21
                BNZ             Lock_wait
                MOV             1, r21
                STC.W           r21, [r20]
                CMP             r0, r21
                BNZ             Lock_success
Lock_wait:      SNOOZE
                BR              Lock
Lock_success:
```

Lock Release

```
                ST.W            r0, 0[r20]
```

3.5.2.3 Performing Mutual Exclusion by Using the SET1 Instruction

The SET1 instruction can be used to perform mutual exclusion over multiple data arrays. By executing the SET1 instruction on the same bit in the memory and then checking the PSW.Z flag, which indicates the execution result, it can be determined whether lock acquisition succeeded or failed.

CAUTIONS

1. **Depending on the hardware specifications, the system performance might drop if exclusive control is executed frequently by using the SET1 instruction, because this causes the bus to be occupied for a long time. It is therefore recommended to execute exclusive control by using the LDL/STC instructions as much as possible.**
2. **When performing mutual exclusion by using the SET1 instruction, to prevent the problem of excessive bus occupancy described in Caution 1 above, execute the SNOOZE instruction before attempting to acquire a lock again after lock acquisition has failed, and adjust the lock acquisition loop execution interval.**

(1) Sample Code

The sample code of a spinlock executed by using the SET1 instruction is shown below.

Lock Acquisition

```
                MOV             lock_adr, r20
Lock:           SET1            0, 0[r20]
                BZ              Lock_success
                SNOOZE
                BR              Lock
Lock_success:
```

Lock Release

```
                CLR1            0, 0[r20]
```

3.5.2.4 Performing Mutual Exclusion by Using the CAXI Instruction

The CAXI instruction can be used to perform mutual exclusion over multiple data arrays. By executing the CAXI instruction on the same word in the memory and then checking the destination register, it can be determined whether lock acquisition succeeded or failed.

CAUTIONS

1. **Depending on the hardware specifications, the system performance might drop if exclusive control is executed frequently by using the CAXI instruction, because this causes the bus to be occupied for a long time. It is therefore recommended to execute exclusive control by using the LDL/STC instructions as much as possible.**
2. **When performing mutual exclusion by using the CAXI instruction, to prevent the problem of excessive bus occupancy described in Caution 1 above, execute the SNOOZE instruction before attempting to acquire a lock again after lock acquisition has failed, and adjust the lock acquisition loop execution interval.**

(1) Sample Code

The sample code of a spinlock executed by using the CAXI instruction is shown below.

Lock Acquisition

```
                MOV             lock_adr, r20
Lock:           MOV             1, r21
                CAXI            [r20], r0, r21
                BZ              Lock_success
                SNOOZE
                BR              Lock
Lock_success:
```

Lock Release

```
                ST.W            r0, 0[r20]
```

3.6 Barrier-Synchronization

3.6.1 Barrier-Synchronization Overview

When parallel processing is performed using a multicore processor, it may be necessary to make the cores wait until the data required for the next processing is ready, at which point the processing can proceed to the next processing. Typically, control that sets up a barrier that prevents the next process from starting before all the applicable cores complete the predetermined processing is called barrier synchronization. It is possible to implement barrier synchronization just using software, but this may degrade system processing performance due to the concentration of barrier arrival notifications from the various cores and memory access from the various cores for verification purposes.

Barrier synchronization registers enable easy barrier synchronization by detecting whether all the applicable cores have completed the required processing through the reception of completion notifications from the various PEs, while also providing a barrier synchronization support function that automatically clears completion notifications from the various PEs in preparation for the next barrier synchronization. This function makes it possible to reduce the number of memory accesses for barrier synchronization, thereby minimizing the negative impact of barrier synchronization on processing performance. This function can also be used for barrier synchronization between PEs forming a cluster. The barrier synchronization register has the following features.

- The 16-ch barrier synchronization registers are provided.
- Barrier synchronization can be implemented using the same code for all the cores.
- Accessible from all clusters and all PEs within the system
- Support of address EDC function
- Support of data ECC function
- Support of guard function to prevent unauthorized access.

3.6.2 Barrier-Synchronization Registers

3.6.2.1 List of Registers

Table 3.158 List of Registers

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
BARR	Barrier-Synchronization n*2 initialization register	BRnINIT	<BARR_base> + 000_H + 010_H × n	8	BRGPROT0_n	—
	Barrier-Synchronization n*2 barrier participating PE setting register	BRnEN	<BARR_base> + 004_H + 010_H × n	8	BRGPROT0_n	—
	Barrier-Synchronization n*2 barrier check register self*1	BRnCHKS	<BARR_base> + 100_H + 010_H × n	8	BRGPROT0_n	—
	Barrier-Synchronization n*2 barrier Synchronization completion register self*1	BRnSYNCS	<BARR_base> + 104_H + 010_H × n	8	BRGPROT0_n	—
	Barrier-Synchronization n*2 barrier check register m*3	BRnCHKm	<BARR_base> + 800_H + 010_H × n + 100 H × m	8	BRGPROT0_16	—
	Barrier-Synchronization n*2 barrier synchronization completion register m*3	BRnSYNCm	<BARR_base> + 804_H + 010_H × n + 100_H × m	8	BRGPROT0_16	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses BR0CHKS register, PE1 can access BR0CHK1 register.

Note 2. n = 0 to 15. n is the Barrier-Synchronization channel number.

Note 3. m = 0 to 3. m is the value of PEID.

3.6.2.2 Self Region

The self region contains the following two types of registers.

- BRnCHKS —Barrier-Synchronization n - barrier check self-register
- BRnSYNCS— Barrier-Synchronization n – barrier synchronization completion self-register

The self registers are virtual registers that do not physically exist. Access from each PE to self registers is routed to the actual register corresponding to the access source PE. **Table 3.159** lists the access source PEs and routing destination registers. For the functions of each register bit, refer to the specifications of the routing destination register. When masters except PEx access the self registers, the register returns 0, write access is ignored and error response will be notified.

It is basically assumed that the barrier synchronization registers are accessed from PEs via self region when PEs use the barrier synchronization function. This allows PEs to use same code because it is not necessary to specify different register address for each PE.

It is also possible to directly access registers by specifying the address of the actual register without using the self register. In this case, the purpose is assumed to check the status of registers for another PE, or referencing of individual registers by a debugging tool.

Table 3.159 Self Region Register Routing List

Self Register	Access Source PE			
	PE0	PE1	PE2	PE3
BRnCHKS	BRnCHK0	BRnCHK1	BRnCHK2	BRnCHK3
BRnSYNCS	BRnSYNC0	BRnSYNC1	BRnSYNC2	BRnSYNC3

3.6.2.3 BRnINIT — Barrier-Synchronization n Initialization Register

This register initializes the Barrier-Synchronization channel n (n = 0 to 15). This register clears the BRnCHKm register and BRnSYNCm register of channel n.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <BARR_base> + 000_H + 010_H × n

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRINIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 3.160 BRnINIT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	BRINIT	Barrier-Synchronization n initialize Writing 1 to this bit clears registers BRnCHK0 to 3 and registers BRnSYNC0 to 3 of the same channel. This bit always returns 0 when read.

3.6.2.4 BRnEN — Barrier-Synchronization n Barrier Participating PE setting Register

This register enables Barrier-Synchronization Participation in channel n for PEs.

Access: This register can be read or written in 8-bit units.

Address: <BARR_base> + 004_H + 010_H × n

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	BREN[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.161 BRnEN Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	BREN[3:0]	Barrier-Synchronization Participation enable for channel n Writing 1 to the x-th bit enables the barrier synchronization support function for PEx. Writing 0 to the x-th bit disables the barrier synchronization support function for PEx. Bit 0: PE0 Bit 1: PE1 : Bit 3: PE3

3.6.2.5 BRnCHKm — Barrier-Synchronization n Barrier Check Register m

This register is used for PEm to notify that PEm has arrived at the barrier of channel n. The PE that has arrived at the barrier is to write the BRCHK bit of this register.

Access: This register can be read or written in 8-bit units.

Address: <BARR_base> + 800_H + 010_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRCHK
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 3.162 BRnCHKm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BRCHK	Barrier Check Bit of channel n This bit can be updated by both software and hardware. Software: When the BRnCHKm register is written, the value of this bit becoming 1 regardless of the writing value. This bit can be set even if the BRnEN.BRENm bit is 0, but it does not affect the barrier -synchronization of channel n. Hardware: This bit is cleared by the hardware when the barrier-synchronization is established at the condition that all the BRnCHKm.BRCHK bits of participating PEs, which are enabled by the BRnEN register, are set. At the same time, the BRnSYNCm.BRSYNC bits of all participating PEs are set. This bit is cleared by writing 1 to the BRnINIT.BRINIT bit.

NOTE

If all bits of the BRnEN register are 0 (no PE participates the barrier synchronization), BRCHK bit cannot be set.

3.6.2.6 BRnSYNCm — Barrier-Synchronization n Barrier Synchronization Completion Register m

This register indicates that all the PEs that participate in the barrier of channel n have arrived at the barrier of channel n.

Access: This register can be read or written in 8-bit units.

Address: <BARR_base> + 804_H + 010_H × n + 100_H × m

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BRSYNC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 3.163 BRnSYNCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BRSYNC	Barrier synchronization completion This bit can be updated by both software and hardware. Software: When the BRnSYNCm register is written, this bit is updated with the write data. Hardware: If the BRnEN.BRENm bit is 1, this bit is automatically set when the barrier-synchronization is established at the condition that all the BRnCHKm.BRCHK bits of participating PEs, which are enabled by the BRnEN register, are set. If the BRnEN.BRENm bit is 0, this bit is not updated by the hardware. This bit is cleared by writing 1 to the BRnINIT.BRINIT bit.

3.6.3 Barrier-Synchronization Function

3.6.3.1 Initial Setting

Before the barrier synchronization register is used, initial settings must be made in the BRnEN register. **Figure 3.52** shows an example of the initial setting of synchronization register channel 0 by PE0.

First, to clear the setting of the previous operation, the BR0EN register is cleared by write operation. After clearing of the BR0EN register, write 07_H to the BR0EN register to participate PE0, PE1 and PE2 in barrier synchronization. Lastly, PE0 writes 1_B to the BR0INIT register to clear registers BR0CHK0 to 3 and registers BR0SYNC0 to 3. Upon completion of setting of the participating PE setting register and initialization, barrier synchronization using the barrier synchronization register is possible.

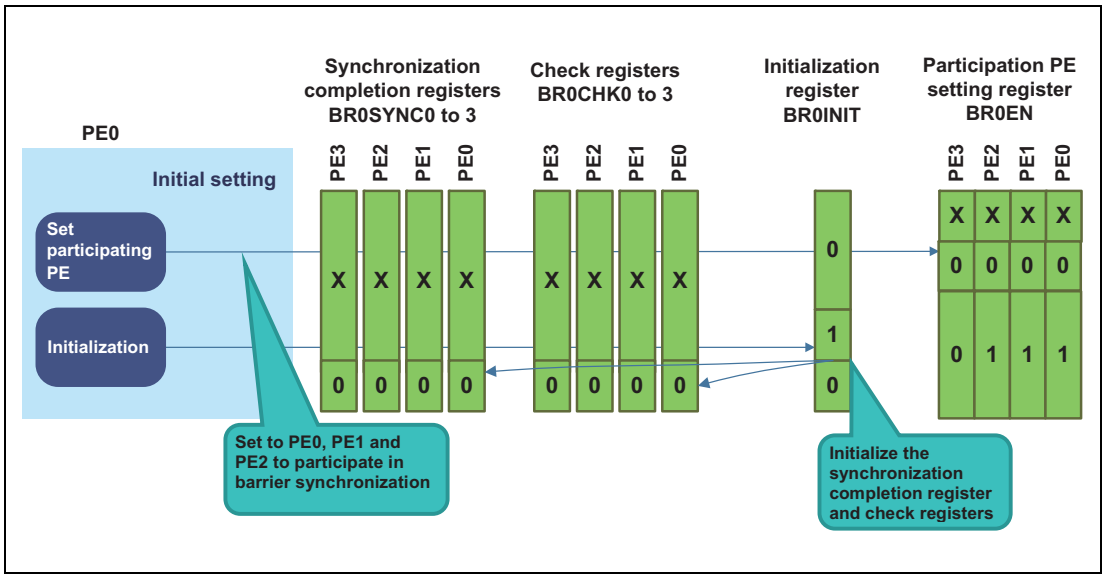
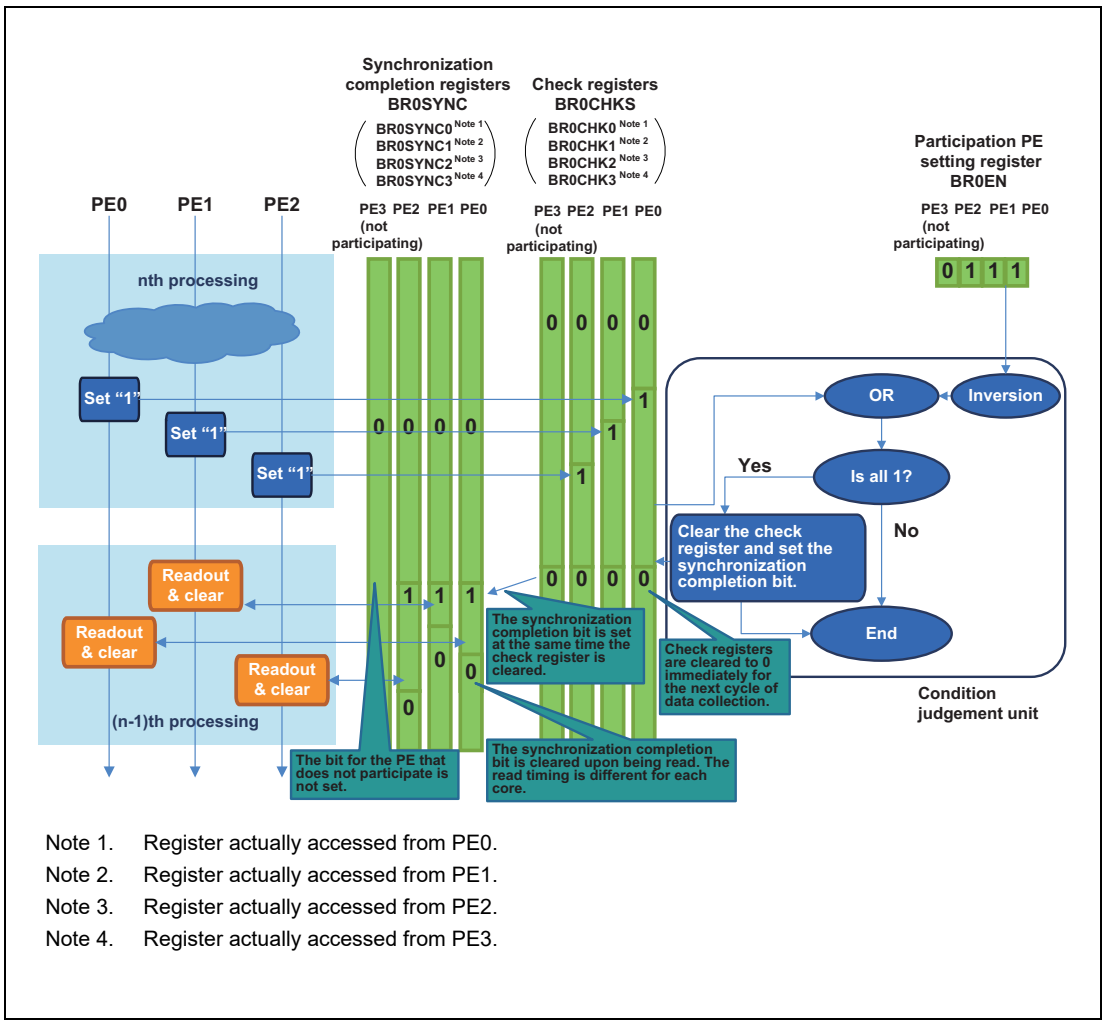


Figure 3.52 Example of Initial Setting of Barrier Synchronization Register

3.6.3.2 Barrier Synchronization

Figure 3.53 shows an example of the operation executing barrier synchronization of PE0, PE1, and PE2 using barrier synchronization register channel 0. In this example, PE0, PE1, and PE2 are set by the BR0EN register in advance to participate in barrier synchronization by the initial setting processing described in **Section 3.6.3.1, Initial Setting**.

When PE0, PE1, and PE2, which are set to participate in barrier synchronization by the BR0EN register, arrive at the barrier, they each write data to the BR0CHKS register, setting it to 1B. Note that the BR0CHKS register is set to 1B when it is written to, regardless of the write value. Each of the PEs, upon arriving at the barrier, polls the BR0SYNCS register and waits for all the other PEs participating in barrier synchronization to arrive at the barrier. When all the PEs participating in barrier synchronization arrive at the barrier and the values of registers BR0CHK0 to 2 have all become 1B, registers BR0CHK0 to 3 are automatically cleared, and 1B is automatically set to registers BR0SYNC0 to 2 (= BR0SYNCS). When a PE that arrives at the barrier detects that the BR0SYNCS register has been set to 1B, it considers that all the PEs participating in barrier synchronization have arrived at the barrier, and it clears the respective BR0SYNCS registers to 0B to allow transition to the next processing. When the values of registers BR0CHK0 to 2 have all become 1B, registers BR0CHK0 to 3 are automatically cleared, so each PE is able to move on to the next cycle as soon as the BR0SYNCS register is cleared.



Note 1. Register actually accessed from PE0.
Note 2. Register actually accessed from PE1.
Note 3. Register actually accessed from PE2.
Note 4. Register actually accessed from PE3.

Figure 3.53 Example of Operation of Barrier Synchronization Using Barrier Synchronization Register

3.6.4 Usage When Virtualization Enabled

3.6.4.1 Overview

Barrier-Sync is designed to support synchronization of VMs running at the same time. Virtualization is realized by time-sharing in this architecture. Therefore, Barrier-Sync cannot be used to synchronize multiple VMs in the same PE.

Bits of Barrier-Sync registers are prepared for each PE, not for each VM. If a PE is allowed to participate in a channel of synchronization, Barrier-Sync cannot distinguish a participating VM from non-participating VMs. To avoid this, use guard function and assign different SPIDs to VMs. Then, registers for a PE of the channel can be written by just one VM. Read access from non-participating VMs is not harmful in normal use.

3.6.4.2 Setting Example

Assumptions:

Considering 6 masters shown in **Table 3.164**, **Table 3.165** and **Figure 3.54**.

Synchronizing Guest 0 of PE0 (SPID=8) and Guest 1 of PE1 (SPID=17) using channel n.

Host of PE0 (SPID=0) and Host of PE1 (SPID=1) are error handling PEs

Table 3.165 shows each register can be accessed by which VMs when guard registers are set as shown **Table 3.164**. Access control is done using PEID and SPID. This is visualized in **Figure 3.54**. In the figure, a black arrow from a Master indicates write access. If it reached a register (BRnINIT, BRnCHK0, BRnCHK1), the register can be written by the master. If not, the access is blocked by PEID matching (BRnCHKS) or by SPID matching (BRGPROT_16, BRGPROT_n).

Table 3.164 Setting Example of Barrier-Sync Registers

Register	Value	Description
BRnEN	03_H	PE0 and PE1 are participating in channel n.
BRRGPROT0_n	$0000\ 0101_H$	Read access: Allowed Write access: Allowed if specified in BRRGPROT1_n
BRRGPROT1_n	$0002\ 0103_H$	Allowed SPID: 0, 1, 8, 17
BRRGPROT0_16	$0000\ 0101_H$	Read access: Allowed Write access: Allowed if specified in BRRGPROT1_16
BRRGPROT1_16	$0000\ 0003_H$	Allowed SPID: 0, 1

Table 3.165 Summary of Access Control

Register	Access	PE0			PE1		
		Host	Guest 0	Guest 1	Host	Guest 0	Guest 1
		SPID = 0	SPID = 8	SPID = 16	SPID = 1	SPID = 9	SPID = 17
BRnINIT, BRnEN	Direct	Allowed	Allowed	Not allowed (SPID)	Allowed	Not allowed (SPID)	Allowed
BRnCHK0, BRnSYNC0	Via Self Registers	Allowed	Allowed	Not allowed (SPID)	Not allowed (PEID)	Not allowed (PEID)	Not allowed (PEID)
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)
BRnCHK1, BRnSYNC1	Via Self Registers	Not allowed (PEID)	Not allowed (PEID)	Not allowed (PEID)	Allowed	Not allowed (SPID)	Allowed
	Direct	Allowed	Not allowed (SPID)	Not allowed (SPID)	Allowed	Not allowed (SPID)	Not allowed (SPID)

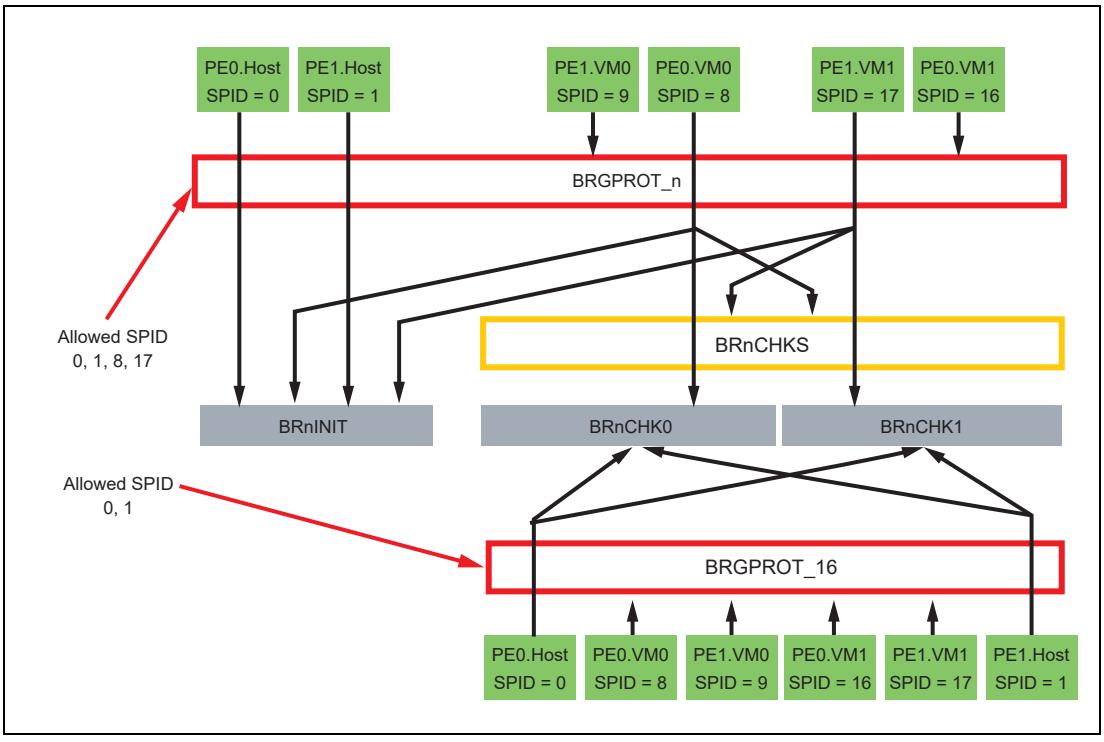


Figure 3.54 Example of per-VM Write Access Control

3.7 TPTM

3.7.1 TPTM Overview

Time Protection Timer (TPTM) is sets of CPU dedicated timers used to achieve high performance and functionality of timing protection. One timer set is assigned to each CPU respectively. One timer set is composed of 2 Interval timers, 1 Free-run timer and 2 Up timers.

Each timer set of TPTM has the following features.

- Interval timer × 2ch (down counter)
- Free-run timer × 1ch (up counter)
- Up timer × 2ch (up counter)
- Start, Stop and Restart of counter for the interval timers, the free-run timer and the up timers.
- Divided counter of the timers can be configured for the interval timers, the free-run timer, up timer respectively.
- Simultaneous count control for the interval timers belonging to the same PE.
- Simultaneous count control for the up timers belonging to the same PE using registers prepared for each PE.
- Globally simultaneous count control registers are prepared. A PE can control all up counters including ones owned by other PEs.
- Underflow interrupt for the interval timers.
- Comparison value matching interrupts for the up timers.
- Address EDC for the access of registers.
- Data ECC for the access of registers.
- Support of guard function to prevent unauthorized access.
- Each timer set has 3 control signals to stop timers while in debug mode. One is for interval timers and the free-run timer, another is for up timer 0, the other is for up timer 1.

3.7.2 TPTM Registers

3.7.2.1 List of Registers

Table 3.166 List of Registers (1/3)

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
TPTM self*1	Counter Start Register of Interval Timer Self	TPTMSIRUN	<TPTM_base> + 000_H	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Interval Timer Self	TPTMSIRRUN	<TPTM_base> + 004_H	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Interval Timer Self	TPTMSISTP	<TPTM_base> + 008_H	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Interval Timer Self	TPTMSISTR	<TPTM_base> + $00C_H$	8, 16, 32	TPTGPROT0_n	—
	Interrupt Enable Register of Interval Timer Self	TPTMSIIEN	<TPTM_base> + 010_H	8, 16, 32	TPTGPROT0_n	—
	Underflow Status Register of Interval Timer Self	TPTMSIUSTR	<TPTM_base> + 014_H	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Interval Timer Self	TPTMSIDIV	<TPTM_base> + 018_H	8, 16, 32	TPTGPROT0_n	—
	Counter Start Register of Free-run Timer Self	TPTMSFRUN	<TPTM_base> + 020_H	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Free-run Timer Self	TPTMSFRRUN	<TPTM_base> + 024_H	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Free-run Timer Self	TPTMSFSTP	<TPTM_base> + 028_H	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Free-run Timer Self	TPTMSFSTR	<TPTM_base> + $02C_H$	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Free-run Timer Self	TPTMSFDIV	<TPTM_base> + 030_H	8, 16, 32	TPTGPROT0_n	—
	Counter Start Register of Up Timer Self	TPTMSURUN	<TPTM_base> + 040_H	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Up Timer Self	TPTMSURRUN	<TPTM_base> + 044_H	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Up Timer Self	TPTMSUSTP	<TPTM_base> + 048_H	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Up Timer Self	TPTMSUSTR	<TPTM_base> + $04C_H$	8, 16, 32	TPTGPROT0_n	—
	Interrupt Enable Register of Up Timer Self	TPTMSUIEN	<TPTM_base> + 050_H	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Up Timer Self	TPTMSUDIV	<TPTM_base> + 058_H	8, 16, 32	TPTGPROT0_n	—
	Trigger Select Register of Up Timer self	TPTMSUTRG	<TPTM_base> + $05C_H$	8, 16, 32	TPTGPROT0_n	—
	Counter Channel 0 Register of Interval Timer Self	TPTMSICNT0	<TPTM_base> + 080_H	32	TPTGPROT0_n	—
	Load Channel 0 Register of Interval Timer Self	TPTMSILD0	<TPTM_base> + 084_H	32	TPTGPROT0_n	—
	Counter Channel 1 Register of Interval Timer Self	TPTMSICNT1	<TPTM_base> + 088_H	32	TPTGPROT0_n	—
	Load Channel 1 Register of Interval Timer Self	TPTMSILD1	<TPTM_base> + $08C_H$	32	TPTGPROT0_n	—
	Counter Register of Free-run Timer Self	TPTMSFCNT	<TPTM_base> + $0A0_H$	32	TPTGPROT0_n	—
	Counter Channel 0 Register of Up Timer Self	TPTMSUCNT0	<TPTM_base> + $0C0_H$	32	TPTGPROT0_n	—
	Compare Value 0 Register of Up Timer 0 Self	TPTMSUCMP00	<TPTM_base> + $0C4_H$	32	TPTGPROT0_n	—
	Compare Value 1 Register of Up Timer 0 Self	TPTMSUCMP01	<TPTM_base> + $0C8_H$	32	TPTGPROT0_n	—
	Compare Value 2 Register of Up Timer 0 Self	TPTMSUCMP02	<TPTM_base> + $0CC_H$	32	TPTGPROT0_n	—
	Compare Value 3 Register of Up Timer 0 Self	TPTMSUCMP03	<TPTM_base> + $0D0_H$	32	TPTGPROT0_n	—
	Counter Channel 1 Register of Up Timer Self	TPTMSUCNT1	<TPTM_base> + $0E0_H$	32	TPTGPROT0_n	—
	Compare Value 0 Register of Up Timer 1 Self	TPTMSUCMP10	<TPTM_base> + $0E4_H$	32	TPTGPROT0_n	—
	Compare Value 1 Register of Up Timer 1 Self	TPTMSUCMP11	<TPTM_base> + $0E8_H$	32	TPTGPROT0_n	—
	Compare Value 2 Register of Up Timer 1 Self	TPTMSUCMP12	<TPTM_base> + $0EC_H$	32	TPTGPROT0_n	—
	Compare Value 3 Register of Up Timer 1 Self	TPTMSUCMP13	<TPTM_base> + $0F0_H$	32	TPTGPROT0_n	—

Table 3.166 List of Registers (2/3)

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
TPTM PEn*2	Counter Start Register of Interval Timer for PEn	TPTMnIRUN	<TPTM_base> + 100_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Interval Timer for PEn	TPTMnIRRUN	<TPTM_base> + 104_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Interval Timer for PEn	TPTMnISTP	<TPTM_base> + 108_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Interval Timer for PEn	TPTMnISTR	<TPTM_base> + $10C_H$ + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Interrupt Enable Register of Interval Timer for PEn	TPTMnIIEN	<TPTM_base> + 110_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Underflow Status Register of Interval Timer for PEn	TPTMnIUSTR	<TPTM_base> + 114_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Interval Timer for PEn	TPTMnIDIV	<TPTM_base> + 118_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Start Register of Free-run Timer for PEn	TPTMnFRUN	<TPTM_base> + 120_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Free-run Timer for PEn	TPTMnFRRUN	<TPTM_base> + 124_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Free-run Timer for PEn	TPTMnFSTP	<TPTM_base> + 128_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Free-run Timer for PEn	TPTMnFSTR	<TPTM_base> + $12C_H$ + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Free-run Timer for PEn	TPTMnFDIV	<TPTM_base> + 130_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Start Register of Up Timer for PEn	TPTMnURUN	<TPTM_base> + 140_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Restart Register of Up Timer for PEn	TPTMnURRUN	<TPTM_base> + 144_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Stop Register of Up Timer for PEn	TPTMnUSTP	<TPTM_base> + 148_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Status Register of Up Timer for PEn	TPTMnUSTR	<TPTM_base> + $14C_H$ + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Interrupt Enable Register of Up Timer for PEn	TPTMnUIEN	<TPTM_base> + 150_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Divider Register of Up Timer for PEn	TPTMnUDIV	<TPTM_base> + 158_H + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Trigger Select Register of Up Timer for PEn	TPTMnUTRG	<TPTM_base> + $15C_H$ + (100_H × n)	8, 16, 32	TPTGPROT0_n	—
	Counter Channel 0 Register of Interval Timer for PEn	TPTMnICNT0	<TPTM_base> + 180_H + (100_H × n)	32	TPTGPROT0_n	—
	Load Channel 0 Register of Interval Timer for PEn	TPTMnILD0	<TPTM_base> + 184_H + (100_H × n)	32	TPTGPROT0_n	—
	Counter Channel 1 Register of Interval Timer for PEn	TPTMnICNT1	<TPTM_base> + 188_H + (100_H × n)	32	TPTGPROT0_n	—
	Load Channel 1 Register of Interval Timer for PEn	TPTMnILD1	<TPTM_base> + $18C_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Counter Register of Free-run Timer for PEn	TPTMnFCNT	<TPTM_base> + $1A0_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Counter Channel 0 Register of Up Timer PEn	TPTMnUCNT0	<TPTM_base> + $1C0_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare Value 0 Register of Up Timer 0 PEn	TPTMnUCMP00	<TPTM_base> + $1C4_H$ + (100_H × n)	32	TPTGPROT0_n	—

Table 3.166 List of Registers (3/3)

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
TPTM PEn*2	Compare Value 1 Register of Up Timer 0 PEn	TPTMnUCMP01	<TPTM_base> + $1C8_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare Value 2 Register of Up Timer 0 PEn	TPTMnUCMP02	<TPTM_base> + $1CC_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare Value 3 Register of Up Timer 0 PEn	TPTMnUCMP03	<TPTM_base> + $1D0_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Counter Channel 1 Register of Up Timer PEn	TPTMnUCNT1	<TPTM_base> + $1E0_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare value 0 Register of Up Timer 1 PEn	TPTMnUCMP10	<TPTM_base> + $1E4_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare value 1 Register of Up Timer 1 PEn	TPTMnUCMP11	<TPTM_base> + $1E8_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare value 2 Register of Up Timer 1 PEn	TPTMnUCMP12	<TPTM_base> + $1EC_H$ + (100_H × n)	32	TPTGPROT0_n	—
	Compare value 3 Register of Up Timer 1 PEn	TPTMnUCMP13	<TPTM_base> + $1F0_H$ + (100_H × n)	32	TPTGPROT0_n	—
TPTM GLOBAL	Simultaneous Counter Start Register of Up Timer 0	TPTMG0URUN	<TPTM_base> + 940_H	8, 16, 32	TPTGPROT0_8	—
	Simultaneous Counter Restart Register of Up Timer 0	TPTMG0URRUN	<TPTM_base> + 944_H	8, 16, 32	TPTGPROT0_8	—
	Simultaneous Counter Stop Register of Up Timer 0	TPTMG0USTP	<TPTM_base> + 948_H	8, 16, 32	TPTGPROT0_8	—
	Simultaneous Counter Start Register of Up Timer 1	TPTMG1URUN	<TPTM_base> + $A40_H$	8, 16, 32	TPTGPROT0_9	—
	Simultaneous Counter Restart Register of Up Timer 1	TPTMG1URRUN	<TPTM_base> + $A44_H$	8, 16, 32	TPTGPROT0_9	—
	Simultaneous Counter Stop Register of Up Timer 1	TPTMG1USTP	<TPTM_base> + $A48_H$	8, 16, 32	TPTGPROT0_9	—

Note 1. When PE accesses the self register, PE can access the corresponding register for each PE. For example, when PE1 accesses TPTMSIRUN register, PE1 can access the TPTM1IRUN register.

Note 2. n = 0 to 3. n is the PEID number.

3.7.2.2 Self Region

The self region contains the following registers. The self region can be accessed by only particular CPU.

The self-registers are virtual registers that do not physically exist. Access from each PE to self-registers is routed to the actual register corresponding to the access source PE. **Table 3.167** lists the access source PEs and routing destination registers. For the functions of each register bit, refer to the specifications of the routing destination register. When masters except PEx access the self registers, the register returns 0, write access is ignored and error response will be notified.

It is basically assumed that the TPTM registers are accessed from PEs via self region. This allows PEs to use same code because it is not necessary to specify different register address for each PE.

It is also possible to directly access registers by specifying the address of the actual register without using the self register. In this case, the purpose is assumed to check the status of registers for another PE, or referencing of individual registers by a debugging tool.

- TPTMSIRUN— Counter Start Register of Interval Timer
- TPTMSIRRUN —Counter Restart Register of Interval Timer
- TPTMSISTP— Counter Stop Register of Interval Timer
- TPTMSISTR— Counter Status Register of Interval Timer
- TPTMSIIEN— Interrupt Enable Register of Interval Timer
- TPTMSIUSTR—Underflow Status Register of Interval Timer
- TPTMSIDIV— Divider Register of Interval Timer
- TPTMSFRUN— Counter Start Register of Free-run Timer
- TPTMSFRRUN —Counter Restart Register of Free-run Timer
- TPTMSFSTP— Counter Stop Register of Free-run Timer
- TPTMSFSTR— Counter Status Register of Interval Timer
- TPTMSFDIV— Divider Register of Free-run Timer
- TPTMSURUN—Counter Start Register of Up Timer
- TPTMSURRUN—Counter Restart Register of Up Timer
- TPTMSUSTP—Counter Stop Register of Up Timer
- TPTMSUSTR—Counter Status Register of Up Timer
- TPTMSUIEN—Interrupt Enable Register of Up Timer
- TPTMSUDIV—Divider Register of Up Timer
- TPTMSUTRG—Trigger Select Register of Up Timer
- TPTMSICNT0— Counter Channel 0 Register of Interval Timer
- TPTMSILD0— Load Channel 0 Register of Interval Timer
- TPTMSICNT1— Counter Channel 1 Register of Interval Timer
- TPTMSILD1— Load Channel 1 Register of Interval Timer
- TPTMSFCNT— Counter Register of Free-run Timer

- TPTMSUCNTm— Counter Channel m Register of Up Timer (m = 0 to 1)
- TPTMSUCMPmi— Compare Value i of 0 Register of Up Timer m (m = 0 to 1, i = 0 to 3)

Table 3.167 Self Region Register Routing List

Self Register	Access Source PE			
	PE0	PE1	PE2	PE3
TPTMSIRUN	TPTM0IRUN	TPTM1IRUN	TPTM2IRUN	TPTM3IRUN
TPTMSIRRUN	TPTM0IRRUN	TPTM1IRRUN	TPTM2IRRUN	TPTM3IRRUN
TPTMSISTP	TPTM0ISTP	TPTM1ISTP	TPTM2ISTP	TPTM3ISTP
TPTMSISTR	TPTM0ISTR	TPTM1ISTR	TPTM2ISTR	TPTM3ISTR
TPTMSIIEN	TPTM0IIEN	TPTM1IIEN	TPTM2IIEN	TPTM3IIEN
TPTMSIUSTR	TPTM0IUSTR	TPTM1IUSTR	TPTM2IUSTR	TPTM3IUSTR
TPTMSIDIV	TPTM0IDIV	TPTM1IDIV	TPTM2IDIV	TPTM3IDIV
TPTMSFRUN	TPTM0FRUN	TPTM1FRUN	TPTM2FRUN	TPTM3FRUN
TPTMSFRRUN	TPTM0FRRUN	TPTM1FRRUN	TPTM2FRRUN	TPTM3FRRUN
TPTMSFSTP	TPTM0FSTP	TPTM1FSTP	TPTM2FSTP	TPTM3FSTP
TPTMSFSTR	TPTM0FSTR	TPTM1FSTR	TPTM2FSTR	TPTM3FSTR
TPTMSFDIV	TPTM0FDIV	TPTM1FDIV	TPTM2FDIV	TPTM3FDIV
TPTMSURUN	TPTM0URUN	TPTM1URUN	TPTM2URUN	TPTM3URUN
TPTMSURRUN	TPTM0URRUN	TPTM1URRUN	TPTM2URRUN	TPTM3URRUN
TPTMSUSTP	TPTM0USTP	TPTM1USTP	TPTM2USTP	TPTM3USTP
TPTMSUSTR	TPTM0USTR	TPTM1USTR	TPTM2USTR	TPTM3USTR
TPTMSUIEN	TPTM0UIEN	TPTM1UIEN	TPTM2UIEN	TPTM3UIEN
TPTMSUDIV	TPTM0UDIV	TPTM1UDIV	TPTM2UDIV	TPTM3UDIV
TPTMSUTRG	TPTM0UTRG	TPTM1UTRG	TPTM2UTRG	TPTM3UTRG
TPTMSICNT0	TPTM0ICNT0	TPTM1ICNT0	TPTM2ICNT0	TPTM3ICNT0
TPTMSILD0	TPTM0ILD0	TPTM1ILD0	TPTM2ILD0	TPTM3ILD0
TPTMSICNT1	TPTM0ICNT1	TPTM1ICNT1	TPTM2ICNT1	TPTM3ICNT1
TPTMSILD1	TPTM0ILD1	TPTM1ILD1	TPTM2ILD1	TPTM3ILD1
TPTMSFCNT	TPTM0FCNT	TPTM1FCNT	TPTM2FCNT	TPTM3FCNT
TPTMSUCNT0	TPTM0UCNT0	TPTM1UCNT0	TPTM2UCNT0	TPTM3UCNT0
TPTMSUCMP0i	TPTM0UCMP0i (i = 0 to 3)	TPTM1UCMP0i (i = 0 to 3)	TPTM2UCMP0i (i = 0 to 3)	TPTM3UCMP0i (i = 0 to 3)
TPTMSUCNT1	TPTM0UCNT1	TPTM1UCNT1	TPTM2UCNT1	TPTM3UCNT1
TPTMSUCMP1i	TPTM0UCMP1i (i = 0 to 3)	TPTM1UCMP1i (i = 0 to 3)	TPTM2UCMP1i (i = 0 to 3)	TPTM3UCMP1i (i = 0 to 3)

3.7.2.3 TPTMnIRUN — Counter Start Register of Interval Timer for PEn

This register controls start of interval timer for each channel.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 100_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRUN1	IRUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.168 TPTMnIRUN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	IRUN1	Start command bit of interval timer channel 1 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD1 is loaded in TPTMnICNT1. This bit is always read as 0.
0	IRUN0	Start command bit of interval timer channel 0 0: No action 1: Start counter When writing 1 to the bit, the value of TPTMnILD0 is loaded in TPTMnICNT0. This bit is always read as 0.

3.7.2.4 TPTMnIRRUN — Counter Restart Register of Interval Timer for PEn

This register controls restart of interval timer for each channel.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 104_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRRUN1	IRRUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.169 TPTMnIRRUN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	IRRUN1	Restart command bit of interval timer channel 1 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT1. This bit is always read as 0.
0	IRRUN0	Restart command bit of interval timer channel 0 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnICNT0. This bit is always read as 0.

3.7.2.5 TPTMnISTP — Counter Stop Register of Interval Timer for PEn

This register controls stop of interval timer for each channel

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 108_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISTP1	ISTP0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.170 TPTMnISTP Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	ISTP1	Stop command bit of interval timer channel 1 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT1. This bit is always read as 0.
0	ISTP0	Stop command bit of interval timer channel 0 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnICNT0. This bit is always read as 0.

3.7.2.6 TPTMnISTR — Counter Status Register of Interval Timer for PEn

This register indicates the interval timer status for each channel.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + $10C_H$ + 100_H × n

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISTR1	ISTR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.171 TPTMnISTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	ISTR1	Status bit of interval timer channel 1 0: Counter is stopped. 1: Counter is running.
0	ISTR0	Status bit of interval timer channel 0 0: Counter is stopped. 1: Counter is running.

3.7.2.7 TPTMnIIEN — Interrupt Enable Register of Interval Timer for PEn

This register controls enabling/disabling of interrupt request by TPTM_IRQ[n].

When underflow occurs in interval timer channel m in TPTMn, TPTMnIUSTR.IUSTRm bit is set whether IIENm bit is set or not. If IIENm bit is set, TPTM_IRQ[n] is asserted.

Note that TPTM_IRQ[n] will be asserted immediately after IIENm bit is set if TPTMnIUSTR.IUSTRm holds 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 110_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIEN1	IIEN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.172 TPTMnIIEN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IIEN1	Interrupt enable bit of interval timer channel 1 0: Disable. 1: Enable.
0	IIEN0	Interrupt enable bit of interval timer channel 0 0: Disable. 1: Enable.

3.7.2.8 TPTMnIUSTR — Underflow Status Register of Interval Timer for PEn

This register indicates the status of underflow occurrence of interval timers in TPTMn.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 114_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IUSTR1	IUSTR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.173 TPTMnIUSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IUSTR1	Underflow flag bit of interval timer channel 1 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR1 is cleared. Writing 1 to the bit is ignored.
0	IUSTR0	Underflow flag bit of interval timer channel 0 0: Underflow did not occur. 1: Underflow occurred. When writing 0 to the bit, IUSTR0 is cleared. Writing 1 to the bit is ignored.

NOTE

To clear a single bit, it is recommended to use CLR1 instruction.

3.7.2.9 TPTMnIDIV — Divider Register of Interval Timer for PEn

This register specifies dividing ratio of interval timer counting clock for each channel.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 118_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	IDIV[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.174 TPTMnIDIV Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	IDIV[7:0]	Clock dividing setting of interval timer for all channels Counter clock ratio is specified by the following expression. count_clock = cpu_clk / (IDIV[7:0] + 1)

CAUTIONS

1. **To avoid unintended timer behavior, the TPTMnIDIV register should be written while all channels of TPTMnICNTm are in stop state.**
2. **It is prohibited to access this register by bit-manipulation instruction.**

3.7.2.10 TPTMnFRUN — Counter Start Register of Free-run Timer for PEn

This register controls start of free-run timer.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 120_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.175 TPTMnFRUN Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	FRUN	Start command bit of free-run timer 0: No action 1: Start counter When writing 1 to the bit during the counter stopping, the value of 0000 0000_H is loaded in TPTMnFCNT and the counter is started. When writing 1 to the bit during the counter working, the value of 0000 0000_H is loaded in TPTMnFCNT and the counter is re-started. This bit is always read as 0.

3.7.2.11 TPTMnFRRUN — Counter Restart Register of Free-run Timer for PEn

This register controls restart of free-run timer.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 124_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRRUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.176 TPTMnFRRUN Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	FRRUN	Restart command bit of free-run timer 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnFCNT. This bit is always read as 0.

3.7.2.12 TPTMnFSTP — Counter Stop Register of Free-run Timer for PEn

This register controls stop of free-run timer.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 128_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.177 TPTMnFSTP Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	FSTP	Stop command bit of free-run timer 0: No action 1: Stop counter When writing 1 to the bit, the counter is stopped at the value of TPTMnFCNT. This bit is always read as 0.

3.7.2.13 TPTMnFSTR — Counter Status Register of Free-run Timer for PEn

This register indicates the free-run timer status.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + $12C_H$ + 100_H × n

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FSTR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.178 TPTMnFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	FSTR	Status bit of Free- run timer 0: Counter is stopped. 1: Counter is running.

3.7.2.14 TPTMnFDIV — Divider Register of Free-run Timer for PEn

This register specifies dividing ratio of free-run timer counting clock.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 130_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FDIV[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.179 TPTMnFDIV Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	FDIV[7:0]	Clock dividing setting of free-run timer Counter clock ratio is specified by the following expression. count_clock = cpu_clk / (FDIV[7:0] + 1)

CAUTIONS

1. **To avoid unintended timer behavior, the TPTMnFDIV register should be written while TPTMnFCNT is in stop state.**
2. **It is prohibited to access this register by bit-manipulation instruction.**

3.7.2.15 TPTMnURUN — Counter Start Register of Up Timer for PEn

This register controls start of up timer for each channel.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 140_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URUN1	URUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.180 TPTMnURUN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	URUN1	Start command bit of up timer channel 1 0: No action 1: Start counter When writing 1 to the bit, the value of 0000 0000_H is loaded onto TPTMnUCNT1. This bit is always read as 0.
0	URUN0	Start command bit of up timer channel 0 0: No action 1: Start counter When writing 1 to the bit, the value of 0000 0000_H is loaded onto TPTMnUCNT0. This bit is always read as 0.

3.7.2.16 TPTMnURRUN — Counter Restart Register of Up Timer for PEn

This register controls restart of up timer for each channel.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 144_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URRUN1	URRUN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.181 TPTMnURRUN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	URRUN1	Restart command bit of up timer channel 1 0: No action 1: Restart counter When writing 1 to the bit, the counter is restarted from current value of TPTMnUCNT1. This bit is always read as 0.
0	URRUN0	Restart command bit of up timer channel 0 0: No action 1: Start counter When writing 1 to the bit, the counter is restarted from current value of TPTMnUCNT0. This bit is always read as 0.

3.7.2.17 TPTMnUSTP — Counter Stop Register of Up Timer for PEn

This register controls stop of up timer for each channel.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 148_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USTP1	USTP0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.182 TPTMnUSTP Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	USTP1	Stop command bit of up timer channel 1 0: No action 1: Stop counter When writing 1 to the bit, the counter is stop at the value of TPTMnUCNT1. This bit is always read as 0.
0	USTP0	Stop command bit of up timer channel 0 0: No action 1: Stop counter When writing 1 to the bit, the counter is stop at the value of TPTMnUCNT0. This bit is always read as 0.

3.7.2.18 TPTMnUSTR — Counter Status Register of Up Timer for PEn

This register indicates the up timer status for each channel.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + $14C_H$ + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USTR1	USTR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.183 TPTMnUSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	USTR1	Status bit of up timer channel 1 0: Counter is stopped (initial). 1: Counter is running.
0	USTR0	Status bit of up timer channel 0 0: Counter is stopped (initial). 1: Counter is running.

3.7.2.19 TPTMnUIEN — Interrupt Enable Register of Up Timer for PEn

This register controls enabling/disabling of interrupt requests, INTTPTMUni.

When TPTMnUIEN.UmIENi is set, INTTPTMUni is asserted if TPTMnUCNTm and TPTMnUCNTmi match. There is no status register for INTTPTMUni because they are pulse interrupt (m = 0 to 1, i = 0 to 3).

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 150_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	U1IEN3	U1IEN2	U1IEN1	U1IEN0	—	—	—	—	U0IEN3	U0IEN2	U0IEN1	U0IEN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 3.184 TPTMnUIEN Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	U1IEN3	Interrupt enable bit of up timer channel 1, compare value 3 0: Disabled (initial). 1: Enabled.
10	U1IEN2	Interrupt enable bit of up timer channel 1, compare value 2 0: Disabled (initial). 1: Enabled.
9	U1IEN1	Interrupt enable bit of up timer channel 1, compare value 1 0: Disabled (initial). 1: Enabled.
8	U1IEN0	Interrupt enable bit of up timer channel 1, compare value 0 0: Disabled (initial). 1: Enabled.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	U0IEN3	Interrupt enable bit of up timer channel 0, compare value 3 0: Disabled (initial). 1: Enabled.
2	U0IEN2	Interrupt enable bit of up timer channel 0, compare value 2 0: Disabled (initial). 1: Enabled.
1	U0IEN1	Interrupt enable bit of up timer channel 0, compare value 1 0: Disabled (initial). 1: Enabled.
0	U0IEN0	Interrupt enable bit of up timer channel 0, compare value 0 0: Disabled (initial). 1: Enabled.

3.7.2.20 TPTMnUDIV — Divider Register of Up Timer for PEn

This register specifies dividing ratio of up timer counting clock for each channel.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 158_H + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UDIV[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.185 TPTMnUDIV Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	UDIV[7:0]	Clock dividing setting of up timer for all channels Counter clock ratio is specified by the following expression. count_clock = cpu_clk / (UDIV[7:0] + 1)

CAUTIONS

1. To avoid unintended timer behavior, the TPTMnUDIV register should be written while all channels of TPTMnUCNTm are in stop state.
2. It is prohibited to access this register by bit-manipulation instruction.

3.7.2.21 TPTMnUTRG — Trigger Select Register of Up Timer for PEn

This register specifies the trigger of up timer for each channel.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + $15C_H$ + 100_H × n

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GTRGEN1	—	—	—	—	—	—	GTRGCH1	GTRGEN0	—	—	—	—	—	—	GTRGCH0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W

Table 3.186 TPTMnUTRG Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	GTRGEN1	Up timer 1 global trigger enable 0: Up counter 1 is controlled by TPTMnURUN, TPTMnURRUN, TPTMnUSTP 1: Up counter 1 is controlled by simultaneous control registers (TPTMGgURUN, TPTMGgURRUN, TPTMGgUSTP)
14 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GTRGCH1	This bit is configured to specify the registers by which up counter 1 is controlled. When TPTMnUTRG.GTRGEN1 = 1 and TPTMnUTRG.GTRGCH1 = g (g = 0 to 1), up counter 1 is controlled by TPTMGgURUN, TPTMGgURRUN, TPTMGgUSTP.
7	GTRGEN0	Up timer 0 global trigger enable 0: Up counter 0 is controlled by TPTMnURUN, TPTMnURRUN, TPTMnUSTP 1: Up counter 1 is controlled by simultaneous control registers (TPTMGgURUN, TPTMGgURRUN, TPTMGgUSTP)
6 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	GTRGCH0	This bit is configured to specify the registers by which up counter 0 is controlled. When TPTMnUTRG.GTRGEN0 = 1 and TPTMnUTRG.GTRGCH0 = g (g = 0 to 1), up counter 0 is controlled by TPTMGgURUN, TPTMGgURRUN, TPTMGgUSTP.

3.7.2.22 TPTMnICNTm — Counter Channel m Register of Interval Timer for PEn

This register is the counter itself of interval timer channel m (m = 0 to 1).

Access: This register can be read or written in 32-bit units.

Address: <TPTM_base> + 180_H + 100_H × n + 8_H × m

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ICNT[31:16]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICNT[15:0]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.187 TPTMnICNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	ICNT[31:0]	Counter value of interval timer channel m for PEn When writing 1_B to TPTMnIRUN.IRUNm, the value of TPTMnILDm is loaded in this register and this register starts to count down from this value. Also, when an underflow of this register occurs, the value of TPTMnILDm is reloaded in this register and this register continues to count down from this value.

CAUTION

To avoid unintended timer behavior, the TPTMnICNTm register should be written while in stop state.

3.7.2.23 TPTMnILDm — Load Channel m Register of Interval Timer for PEn

This register specifies the loading value for interval timer channel m (m = 0 to 1).

Access: This register can be read or written in 32-bit units.

Address: <TPTM_base> + 184_H + 100_H × n + 8_H × m

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ILD[31:16]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ILD[15:0]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.188 TPTMnILDm Register Contents

Bit Position	Bit Name	Function
31 to 0	ILD[31:0]	Load data value of interval timer channel m for PEn At the timing of writing 1 to TPTMnIRUNm, the value of this register is loaded in TPTMnICNTm. Also, when an underflow of TPTMnICNTm occurs, the value of this register is reloaded in TPTMnICNTm.

CAUTION

To avoid unintended timer behavior, the TPTMnILDm register should be written while TPTMnCNTm of corresponding channel is in stop state.

3.7.2.24 TPTMnFCNT — Counter Register of Free-run Timer for PEn

This register is the counter itself of free-run timer.

Access: This register can be read or written in 32-bit units.

Address: <TPTM_base> + $1A0_H$ + 100_H × n

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FCNT[31:16]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCNT[15:0]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.189 TPTMnFCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	FCNT[31:0]	Counter value of free-run timer for PEn When writing 1_B to TPTMnFRUN.FRUN, the value of 0000_0000_H is loaded in this register and this register starts to count up from this value. Also, when an overflow of this register occurs, the value of 0000_0000_H is reloaded in this register and this register continues to count up from this value.

CAUTION

To avoid unintended timer behavior, the TPTMnFCNT register should be written while in stop state.

3.7.2.25 TPTMnUCNTm — Counter Register of Up Timer for PEn

This register is the counter itself of up timer channel m (m = 0 to 1).

Access: This register can be read or written in 32-bit units.

Address: <TPTM_base> + $1C0_H$ + 100_H × n + 20_H x m

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UCNT[31:16]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UCNT[15:0]															
Value after reset	Undefined															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.190 TPTMnUCNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	UCNT[31:0]	Counter value of up timer for PEn When writing 1_B to TPTMnURUN.URUNm, the value of 0000 0000_H is loaded in this register and this register starts to count up from this value. Also, when an overflow of this register occurs, the value of 0000 0000_H is reloaded in this register and this register continues to count up from this value.

CAUTION

To avoid unintended timer behavior, the TPTMnUCNTm register should be written while in stop state.

3.7.2.26 TPTMnUCMPmi — Compare Value i Register of Up Timer m for PEn

This register specifies the i-th value compared with the value of up timer channel m (m = 0 to 1, i = 0 to 3).

Access: This register can be read or written in 32-bit units.

Address: <TPTM_base> + $1C4_H$ + 100_H × n + 20_H x m + 4_H x i

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.191 TPTMnUCMPmi Register Contents

Bit Position	Bit Name	Function
31 to 0	UCMP[31:0]	Value compared with up counter If TPTMnUIEN.UmIENi = 1 and TPTMnUCNTm is equal to TPTMnUCMPmi, INTTPTMni is asserted.

CAUTION

To avoid unintended timer behavior, the TPTMnUCMPmi register should be written while in stop state.

When the TPTMnUCMPmi register is updated while the up timer is running, it must be avoided that TPTMnUCNTm exceeds TPTMnUCMPmi during the write process to TPTMnUCMPmi as the interrupt (INTTPTMUni) of the up timer might be lost.

3.7.2.27 TPTMGgURUN — Counter Start Register of Up Timer for Global Control Channel g

This register starts Up timers.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 940_H + 100_H x g (g = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.192 TPTMGgURUN Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	URUN	Start command bit of up timers 0: No action 1: Start up timer m of PEn if TPTMnUTRG.GTRGENm is 1 and TPTMnUTRG.GTRGCHm = g. If not, no action. When writing 1 to the bit, the value of 0000 0000_H is loaded onto TPTMnUCNTm. This bit is always read as 0.

3.7.2.28 TPTMGgURRUN — Counter Restart Register of Up Timer for Global Control Channel g

This register restarts Up timers.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 944_H + 100_H x g (g = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URRUN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.193 TPTMGgURRUN Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	URRUN	Restart command bit of up timers 0: No action 1: Restart up timer m of PEn if TPTMnUTRG.GTRGENm is 1 and TPTMnUTRG.GTRGCHm = g. If not, no action. When writing 1 to the bit, the counter is restarted from current value of TPTMnUCNTm. This bit is always read as 0.

3.7.2.29 TPTMGgUSTP — Counter Stop Register of Up Timer for Global Control Channel g

This register stops Up timers.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <TPTM_base> + 948_H + 100_H x g (g = 0 to 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	USTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.194 TPTMGgUSTP Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	USTP	Stop command bit of up timers 0: No action 1: Stop up timer n of PEn if TPTMnUTRG.GTRGENm is 1 and TPTMnUTRG.GTRGCHm = g. If not, no action. When writing 1 to the bit, the counter is stop at the value of TPTMnUCNTm. This bit is always read as 0.

3.7.3 TPTM Function

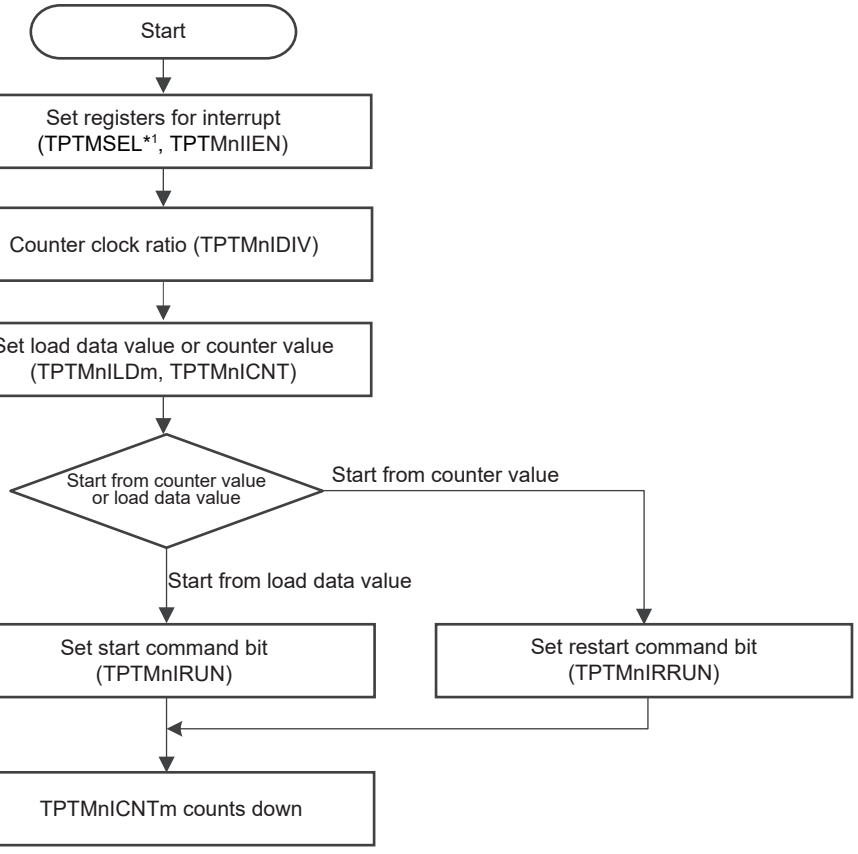
CAUTION

If interruption is used as EI level instead of FE level, set the TPTMSEL register before TPTM starts. It is prohibited to change the setting value of the TPTMSEL register after TPTM start.

3.7.3.1 Interval Timer Operation

(1) Normal Operation

The normal operation flow is shown in **Figure 3.55**.



Note 1. see **Section 6.3.15, TPTMSEL — TPTM Interrupt FE EI Select Register**.

Figure 3.55 Flow of Normal Operation

It shows the timing chart of normal operation of interval timer in case that counter clock ratio is a 1/2 period of CPU_CLK and load data value is 0000 0004_H in **Figure 3.56**.

TPTMnICNTm starts countdown at the next cycle of setting TPTMnIRUN.IRUNm bit.

If TPTMnIRUN.IRUNm bit is set while the counter is running, the TPTMnICNTm value is updated by the TPTMnILDm value and continues countdown.

When the TPTMnICNTm turns zero, TPTMnUSTR.USTRm is set and, if TPTMnIIEN.IIENm bit is set, TPTM_IRQ[n] is asserted.

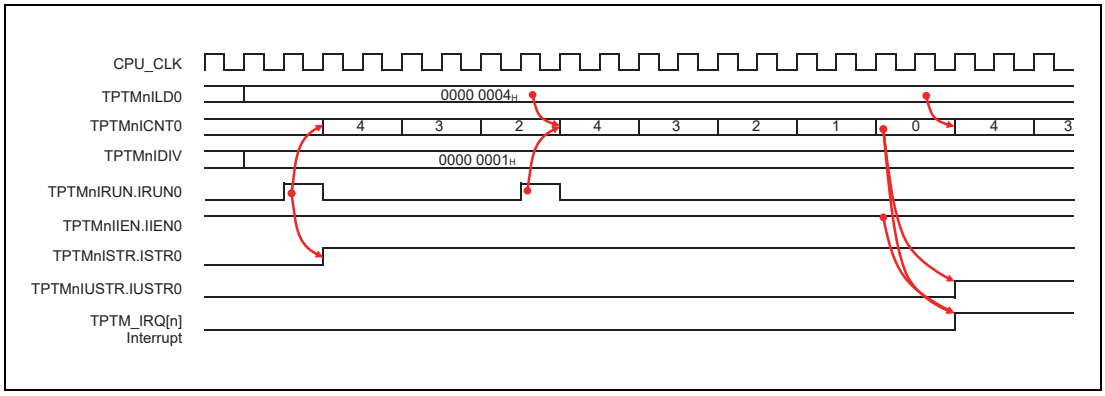


Figure 3.56 Normal Operation (TPTMnIDIV = 0000 0001$_H$, TPTMnILD = 0000 0004$_H$)

(2) Operation during a Restart

The flow of operation during a restart is shown in **Figure 3.57**.

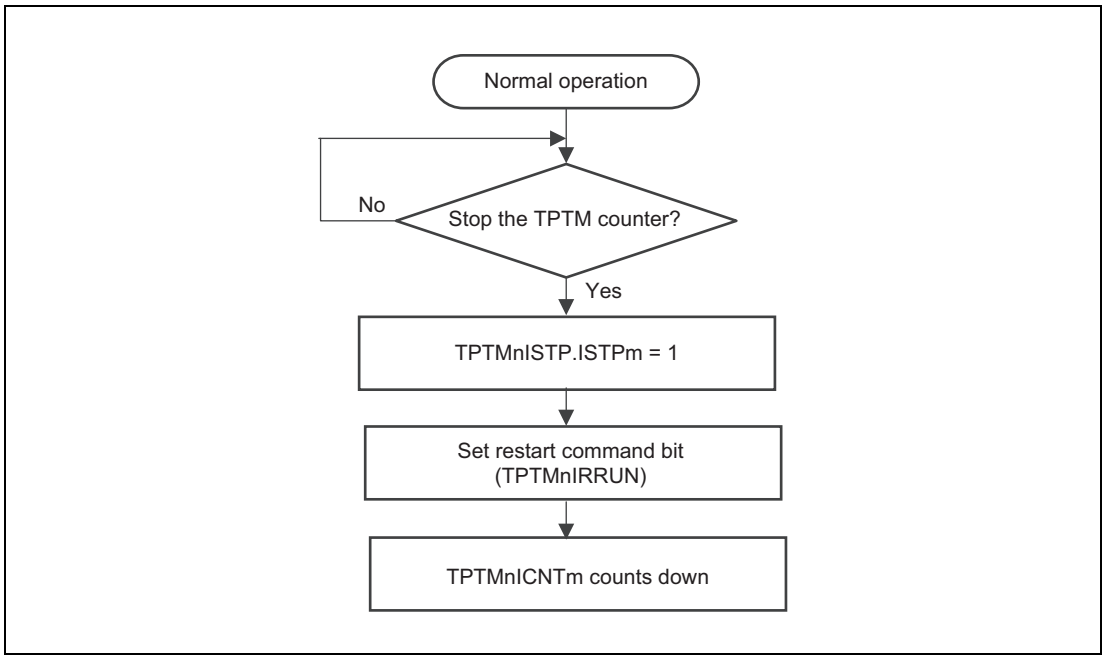


Figure 3.57 Flow of Operation during a Restart

TPTMnICNTm starts countdown at the next cycle after setting TPTMnIRUN.

TPTMnICNTm stops countdown at the next cycle after setting TPTMnISTP during countdown.

TPTMnICNTm restarts countdown from the current counter value at the next cycle after setting TPTMnIRRUN during countdown stop.

The timing chart of interval timer operation at the time of restart when the counter clock ratio is 1/2 period of CPU_CLK and load data value is 0000 0004$_H$ is shown in **Figure 3.58**.

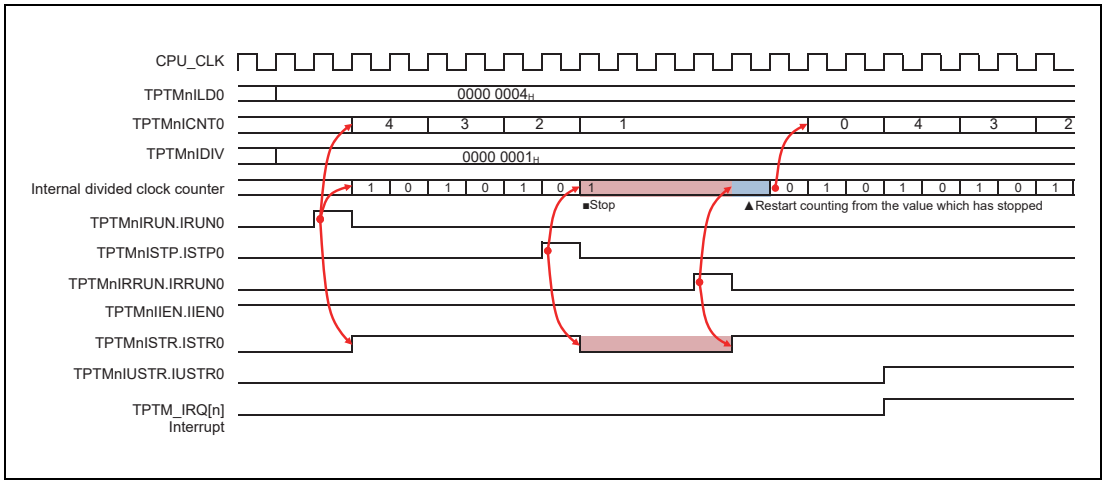


Figure 3.58 Operation during a Restart (TPTMnIDIV = 0000 0001$_H$, TPTMnILD = 0000 0004$_H$)

(3) Operation during an Underflow

If the interval counter is still running even when the TPTMnCNTm is 0000 0000$_H$, counter under flow will occur and TPTMnIUSTR.IUSTRm bit will be set at the next cycle.

When TPTMnIIEN.IIENm is 1$_B$, an interrupt is enabled, TPTMnIUSTR.IUSTRm is set and TPTM_IRQ is asserted at the next cycle after TPTMnCNTm becomes 0000 0000$_H$.

When CPU has finished interrupt processing of TPTM, it needs to clear TPTMnIUSTR.IUSTRm.

Cycles of TPTMn Interrupt

The cycles of TPTMn Interrupt are as follows.

- TPTMn Interrupt generation cycle = TPTMnCNTm × Counter-Clock cycle*

  *Counter-Clock cycle = CPU_CLK × (TPTMnIDIV+1)

When TPTMnIIEN.IIENm is 0, an interrupt is disabled, TPTMnIUSTR.IUSTRm is set but TPTM_IRQ is not asserted at the next cycle after TPTMnCNTm becomes 0000 0000$_H$. In this case, TPTM_IRQ is asserted if TPTMnIIEN.IIENm is set before TPTMnIUSTR.IUSTRm is cleared.

3.7.3.2 Free-run Timer Operation

Free-run timer operation is same as interval timer basically, except following features. The explanation for common feature is omitted.

- Counter direction is up-counter. TPTMnFCNT value is incremented by every TPTMnFDIV period.
- No load value register. TPTMnFCNT always starts 0000 0000$_H$ by setting TPTMnFRUN.FRUN bit.
- No interrupt assertion for the counter overflow.
- Number of channels: 1 channel

3.7.3.3 Up Timer Operation

Up timer operation is same as interval timer basically, except following features. The explanation for common feature is omitted.

- Counter direction is up-counter. TPTMnUCNTm value is incremented by every TPTMnUDIV period.
- No load value register. TPTMnUCNT always starts 0000 0000_H by setting TPTMnURUN.URUNm bit.
- Edge interrupt asserted based on counter value comparison. INTTPTMUni is asserted either when TPTMnUCNT0 = TPTMnUCMP0i or when TPTMnUCNT1 = TPTMnUCMP1i.
- Number of channels: 2 channels

3.8 Inter-VM Communication (IVC)

Data communication between Virtual Machines (VMs) is accomplished through shared memory blocks. This hardware supports function to establish a single owner for this shared memory block without Virtual Machine Monitor (VMM)*1 management at any time. This function is called Inter-VM communication (IVC).

Note 1. Virtual Machine Monitor (VMM) or hypervisor means Software that manages virtual machines.

3.8.1 Features

IVC supports function below

- Access control is based on CRAM guard function (CRG).
- Only two VMs can exchange data with the shared memory block.
- Only either VM is permitted to write the memory block. The write accessible VM is called “owner”. Write access permission is called “ownership”.
- Two VMs can exchange the ownership to write the shared memory block.
- The current owner can block the ownership request by another VM. In this case, it is possible that current owner can change the ownership.

3.8.2 Registers

IVC function is based on CRAM guard function. In this section, Registers related to IVC function are described. Refer to **Section 44, Functional Safety** to see about the detail of CRAM guard function register.

3.8.2.1 Register Base Address

Table 3.195 Register Base Address

Base Address Name	Base Address	Peripheral Group
<GUARD_CRAMCRG0_base>	FFC6 D000$_H$	P-Bus Group 0
<GUARD_CRAMCRG1_base>*1	FFC6 D200$_H$	P-Bus Group 0
<GUARD_CRAMCRG2_base>*2	FFC6 D400$_H$	P-Bus Group 0
<GUARD_CRAMCRG3_base>	FFC6 D600$_H$	P-Bus Group 0
<GUARD_CRAMCRG01_base>	FFC6 E800$_H$	P-Bus Group 0
<GUARD_CRAMCRG11_base>*1	FFC6 EA00$_H$	P-Bus Group 0
<GUARD_CRAMCRG21_base>*2	FFC6 EC00$_H$	P-Bus Group 0
<GUARD_CRAMCRG31_base>	FFC6 EE00$_H$	P-Bus Group 0

Note 1. This base address is not available in RH850/U2A8 and RH850/U2A6.
Note 2. This base address is not available in RH850/U2A6.

3.8.2.2 List of Register

Table 3.196 List of Registers

Module Name	Register Name	Register Symbol	Address	Access Size	Access Protection	
					IBG	Other
CRAMCRGn *5	Channel protection control register	CRGPROTm (m = 0 to 7)	<GUARD_CRAMCRGn_base> + 10_H + m * 20_H	8, 16, 32	*1	CRGKCPROT
	Inter-VM Communication Access Permission Register	CRGIVCSPIDm (m = 0 to 7)	<GUARD_CRAMCRGn_base> + 20_H + m * 20_H	8, 16, 32	*1	*2
CRAMCRGn1 *5	Inter-VM Communication Ownership Request Register	CRGIVCREQm (m = 0 to 7)	<GUARD_CRAMCRGn1_base> + 00_H + m * 20_H	32	—	*3
	Inter-VM Communication Ownership Request Lock Register	CRGIVCLOCKm (m = 0 to 7)	<GUARD_CRAMCRGn1_base> + 04_H + m * 20_H	32	—	*4
	Inter-VM Communication Current Ownership Register	CRGIVCOWNRm (m = 0 to 7)	<GUARD_CRAMCRGn1_base> + 08_H + m * 20_H	32	—	—

Note 1. n=0: PBG00#3
n=1: PBG00#4
n=2: PBG00#5
n=3: PBG00#6

Note 2. It is possible to write to this register when CRGKCPROT.KCE=1 and CRGPROTm.OW=0.

Note 3. Write operation that matches to conditions below is valid when CRGSPIDm.OW=1
When CRGIVCLOCKm.LK=0, master with SPID that is assigned to CRGIVCSPIDm register
When CRGIVCLOCKm.LK=1, master that has the ownership of IVC function

Note 4. It is possible to write to this register when either condition below matches CRGKCPROT.KCE=1.
Write access from the current owner VM, when CRGPROTm.OW=1.

Note 5. n is the cluster number.
RH850/U2A-EVA, RH850/U2A16: n = 0, 1, 2, 3
RH850/U2A8: n = 0, 2, 3
RH850/U2A6: n = 0, 3

3.8.2.3 CRGPROTm — CRAM Guard Control Register

Specifies the access to be protected for CRAM. Access prohibited by any of the identifiers is protected as unauthorized access. This register is used to specify the settings for those other than SPID. This register also is used to enable or disable the function of IVC.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base> + 10_H + m × 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 3.197 CRGPROTm Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	OW	Enable Inter-VM Communication function. It enable to control CRGSPIDm register setting, even if CRGKCPROT.KCE is set to 0. 0: Disable Inter-VM Communication function. It is possible to write to CRGSPIDm register when CRGKCPROT.KCE is set to 1. 1: Enable Inter-VM Communication function. Only VMs with an SPID set in CRGIVCSPIDm register can control CRGSPIDm register setting by access to CRGIVCREQm and CRGIVCLOCKm registers, even if CRGKCPROT.KCE is set to 0.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting. For more details, see **Section 44.5.5, CRG**
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master. For more details, see **Section 44.5.5, CRG**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode. For more details, see **Section 44.5.5, CRG**
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable. For more details, see **Section 44.5.5, CRG**
0	RG	Read Global Enable. For more details, see **Section 44.5.5, CRG**

3.8.2.4 CRGIVCSPIDm — Inter-VM Communication Access Permission Register

Defines which VMs are used for IVC function. SPID values, which are assigned to VMs for IVC function, are set to this register. VMs that are set to this register can control CRGSPIDm register setting by access to CRGIVCREQm and CRGIVCLOCKm registers. When KCE bit in CRGKCPROT register is 1 and OW bit in CRGPROTm register is 0, it is possible to write to this register. It is mandatory to set only 2bits, that are SPID bits of VM to control IVC function.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base>+ 20_H + m × 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IVCSPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IVCSPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.198 CRGIVCSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	IVCSPID	Setting of CRGSPIDm register access permission This register is a list of bits representing SPID value to be used for IVC. Each bit number shows SPID value assigned to VM. When CRGPROTm.OW is set to 1 and IVCSPID[n] is set to 1, VM that has SPID=n is permitted the write access to CRGIVCREQm and CRGIVCLOCKm registers for Inter VM communication. This register is writeable only when CPGKCPROT.KCE=1 and CRGPROTm.OW=0. [n]=0: Write access from VM with SPID=n to CRGSPIDREQm and CRGIVCLOCKm registers is not permitted. (n=0-31) [n]=1: Write access from VM with SPID=n to CRGSPIDREQm and CRGIVCLOCKm registers is permitted. (n=0-31)

Note 1. It is mandatory to set only two bits to control Inter-VM Communication.

3.8.2.5 CRGIVCREQm — Inter-VM Communication Ownership Request Register

Sets the trigger to issue the request of the ownership, when Inter-VM Communication is valid. Only access from VMs that are set to CRGIVCSPIDm register is accepted.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <GUARD_CRAMCRGn1_base> + 00_H + m × 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IVCREQ[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IVCREQ[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 3.199 CRGIVCREQm Register Contents

Bit Position	Bit Name	Function
31 to 0	IVCREQ	Issue the ownership request from VM. This register is write only register. Read data is always 0. When VM assigned to IVC function writes any data to this register, CRGSPIDm register can be updated from the current SPID value to another SPID value to be used for IVC function. Ownership request is issued when CRGPROTm.LK = 0. - Write access from either VM, when CRGIVCLOCKm.LK = 0. - Write access from the current owner. It does not depend on CRGIVCLOCKm.LK value.

3.8.2.6 CRGIVCLOCKm — Inter-VM Communication Ownership Request Lock Register

Locks the request from non-owner VM. This register is writable when KCE bit in CRGKCPROT register is 1 or the write access from the owner when CRGPROTm.OW is set to 1. This register is automatically cleared, when the current owner writes to CRGIVCREQm register.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_CRAMCRGn1_base> + 04_H + m × 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.200 CRGIVCLOCKm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LK	Block the ownership request from VM that does not have the ownership. It is possible to write to this bit, when either condition below matches. - CRGKCPROTm.KCE = 1 - CRGPROTm.OW = 1 and the access from the owner VM It is possible to clear this register automatically, when the owner writes to CRGIVCREQm register to release the ownership. 0: Ownership request is accepted. 1: Ownership request is NOT accepted.

3.8.2.7 CRGIVCOWNRm — Inter-VM Communication Current Ownership Register

It is possible to read the status of the current owner. The value of this register is same with the value of CRGSPIDm register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_CRAMCRGn1_base> + 08_H + m × 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IVCOWN[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IVCOWN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.201 CRGIVCOWNRm Register Contents

Bit Position	Bit Name	Function
31 to 0	IVCOWN	Current owner SPID for Inter-VM Communication This register is read-only register, and the value is same with that of CRGSPIDm register.

3.8.3 Function

Each VM is identified by SPID value. Inter-VM Communication controls shared memory (CRAM) access with SPID setting of CRAM guard function.

CRAM guard operation can be changed to support Inter-VM Communication, when OW bit in CRGPROTm is set to 1.

GRGIVCSPIDm register defines SPID values of VMs to be used for Inter-VM Communication. This register must be set to 1 for only 2 bits exactly because Inter-VM Communication supports only data exchange with 1-to-1 VMs. Therefore, it is necessary to be set by the trusted software like VMM.

It is necessary for CRGSPIDm register to set either SPID value to be used for IVC function.

Two VMs assigned to IVC function can control the condition of CRGSPIDm register with CRGIVCREQm and CRGIVCLOCKm registers without setting KCE bit in CRGKCPROT register to 1. By CRGIVCREQm register write access from VM for IVC function, CRGSPIDm value can be updated from the current SPID to another SPID. Setting 1 to LK bit in CRGIVCLOCKm, CRGSPIDm cannot be updated by request from non-owner VM.

3.8.3.1 Example of Operation

This section is described about the example of IVC operation.

- Condition
  - SPID value of VMs for IVC: VM0 (SPID=1) and VM1 (SPID=7)
  - VM0 has the ownership
- Necessary initial register setting to enable IVC function
  - CRGSPIDm.SPID[1] = 1 (CRGSPIDm = 00000002_H)
  - CRGIVCSPIDm[1] and CRGIVCSPIDm[7] are set to 1. (CRGIVCSPIDm=00000082_H)
  - CRGPROTm.OW = 1
  - Other CRAM guard setting (Address range, Guard function enable and so on)
- Case 1: VM0 writes to CRGIVCREQm register.
  - CRGSPIDm register value changes from 00000002_H to 00000080_H. (Ownership is changed)
- Case 2: VM1 writes to CRGIVCREQm register.
  - CRGSPIDm register value changes from 00000002_H to 00000080_H. (Ownership is changed)
- Case 3: VM0 writes to CRGIVCREQm register after VM0 set 1 to CRGIVCLOCKm register.
  - CRGSPIDm register value changes from 00000002_H to 00000080_H (Ownership is changed)
  - CRGIVCLOCKm register is automatically cleared.
- Case 4: VM1 writes to CRGIVCREQm register after VM0 set 1 to CRGIVCLOCKm register.
  - CRGSPIDm register value keeps 00000002_H. (Ownership is kept)
  - CRGIVCLOCKm register keeps 00000001_H.
- Case5: VM1 write to CRGIVCLOCKm register
  - CRGIVCLOCKm register keeps 00000000_H. Because VM1 is not an owner.

3.8.4 Software flow of IVC

Software flow is described for IVC.

3.8.4.1 Initial setting for IVC

Initial setting must be executed by the trusted software.

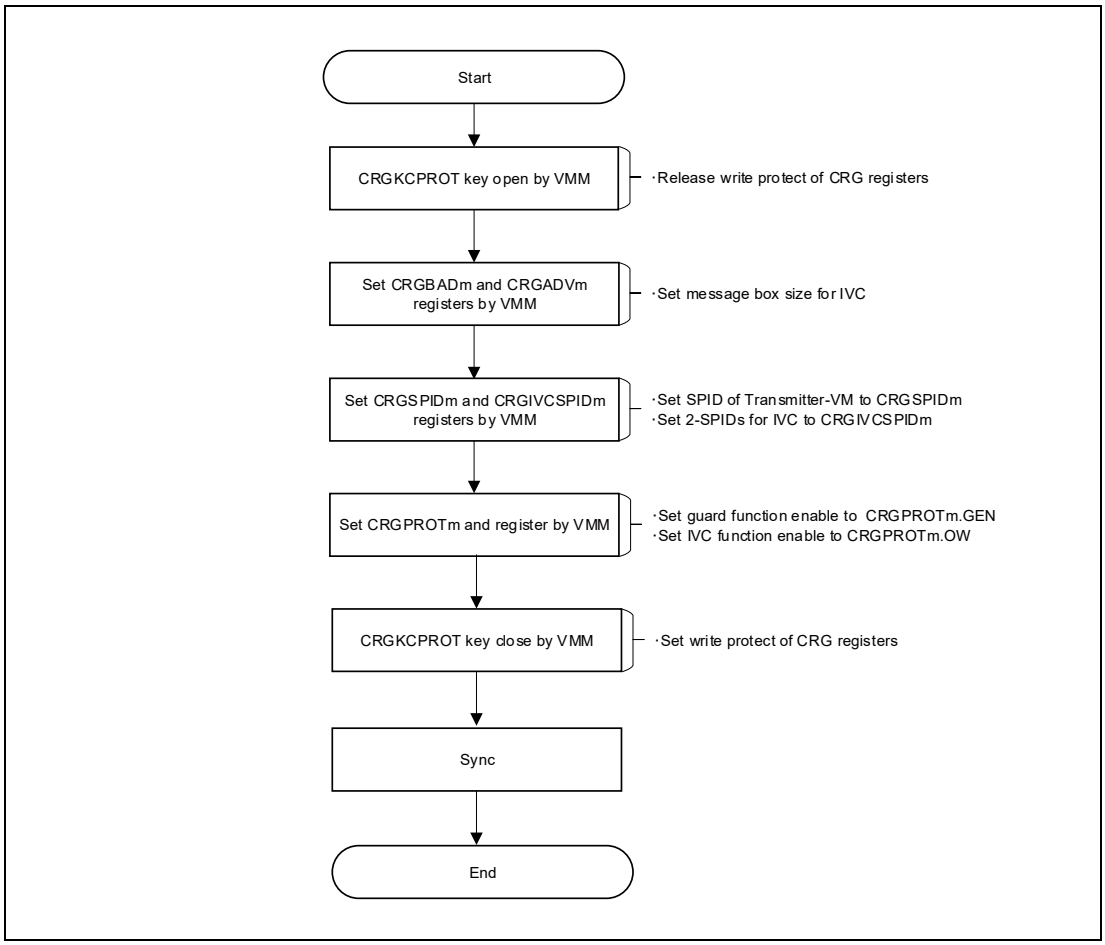


Figure 3.59 Initial Setting Flow for IVC

3.8.4.2 Data Transfer and Pass the Ownership

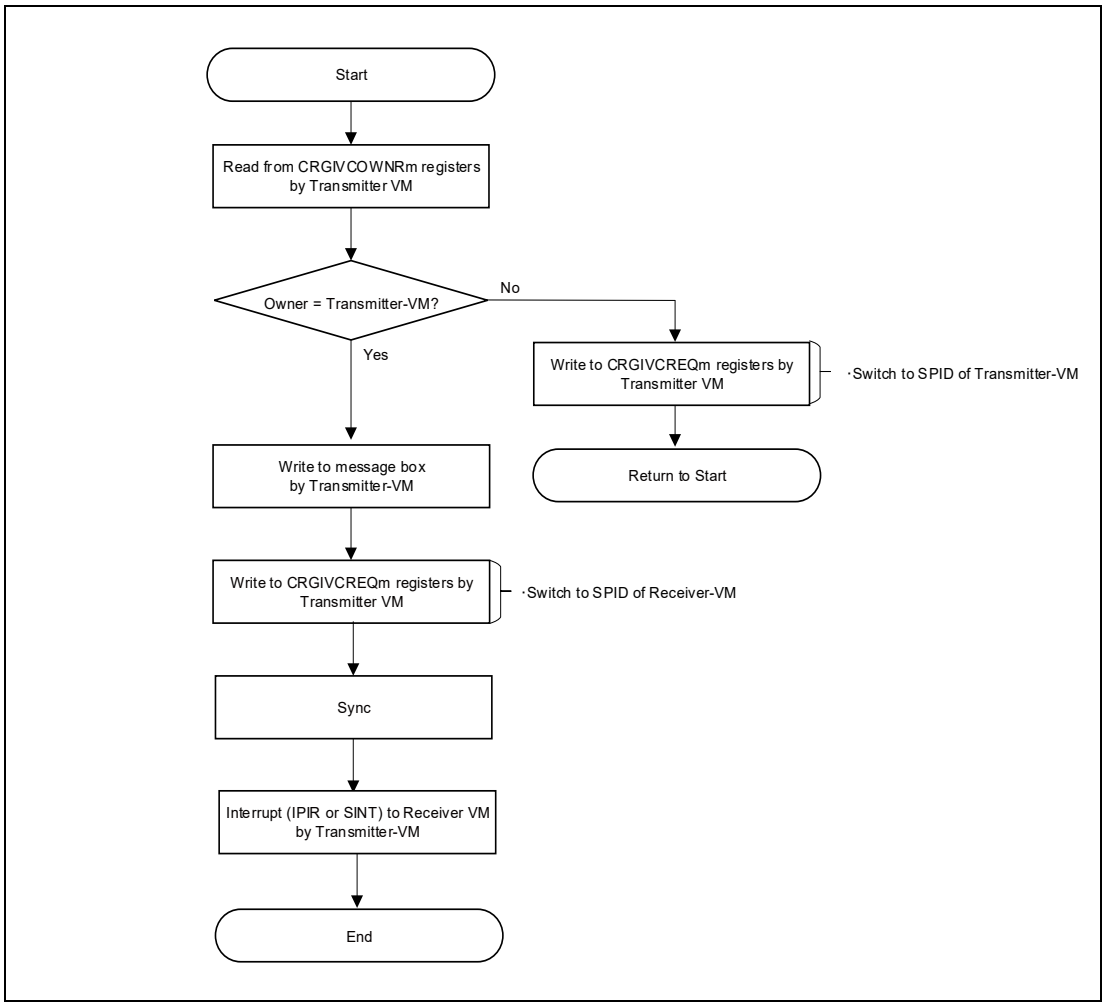


Figure 3.60 Data Transfer and Pass the Ownership Flow

3.8.4.3 Data Received

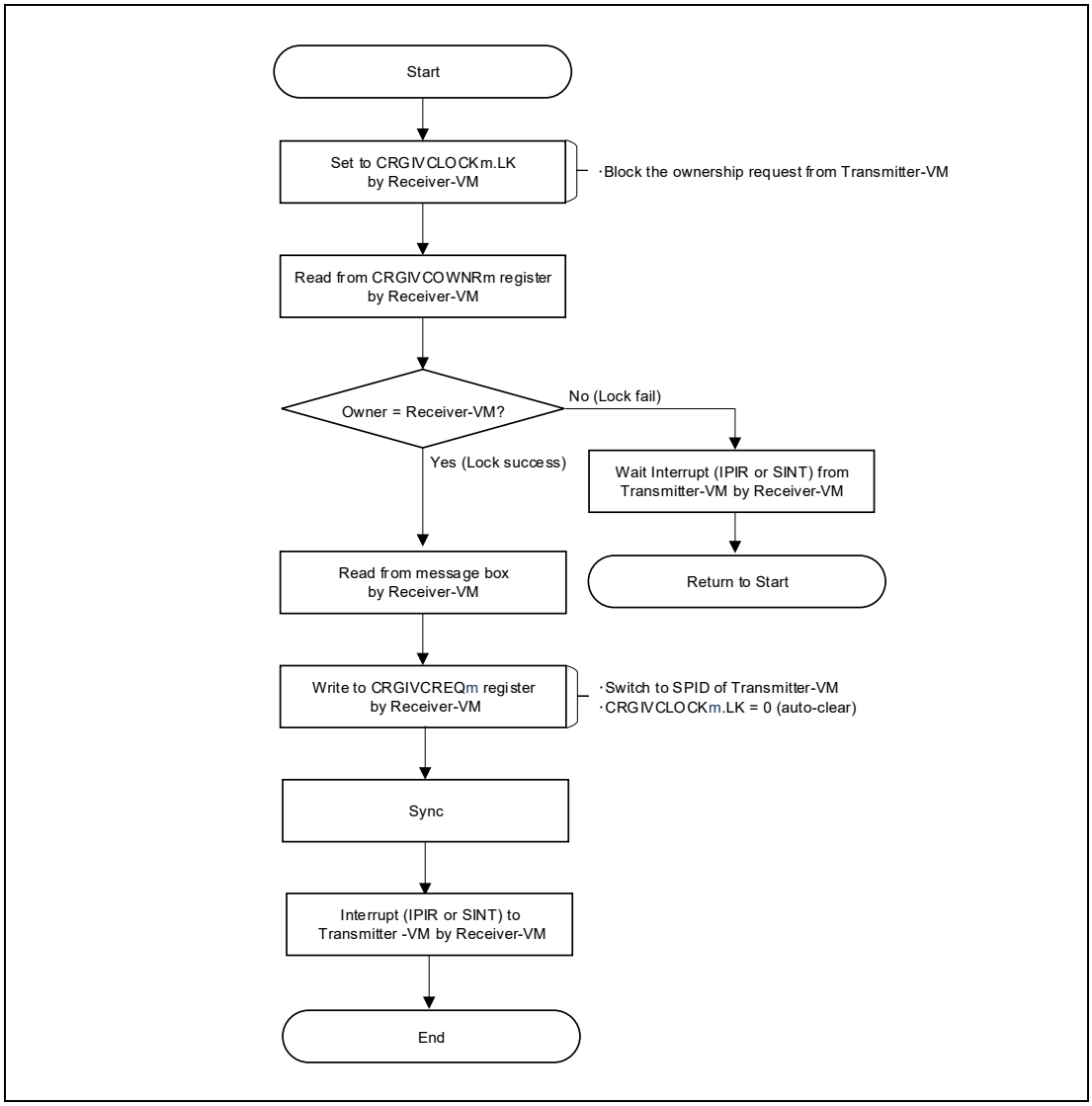


Figure 3.61 Data Received Flow

3.9 Usage Notes

3.9.1 Synchronization of Store Instruction Completion and Subsequent Instruction Generation

When a control register is updated by a store instruction, there is a time lag from execution of the store instruction by the CPU to actual updating of the control register. Therefore, appropriate processing for synchronization is required to ensure that control registers reflect updated values before execution of subsequent instructions. The processing for synchronization is shown below.

For details on the procedure regarding updating of the system registers by the LDSR instruction and synchronization with subsequent instructions, refer to **Section 3.2.7.3, Hazard Management after System Register Update**.

3.9.1.1 When Updated Results in the Control Registers or Memory are Reflected in the Implementation of a Subsequent Instruction:

Example 1: An interrupt may be enabled by implementation of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits. Proceed as follows in this case.

(1) Store instruction to update a control register (ST.W, etc.)

(2) Dummy reading of the above-mentioned control register (LD.W, etc.)*1

(3) SYNCP

(4) Subsequent instruction (EI, etc.)

Example 2: Implement the same processing even when the access required after waiting to secure the updating of a given control register (register A) is to another control register (register B). This includes the following cases: the interlinked operation of different peripheral modules and when releasing the interrupt mask in INTC after making peripheral module settings.

However, this processing is unnecessary if control registers A and B are in the same P-Bus Group.

For the correspondence between the P-Bus Groups and the peripheral modules/registers, see each section.

(1) Store instruction to update control register A (ST.W, etc.)

(2) Dummy reading of the above-mentioned control register (LD.W, etc.)*1

(3) SYNCP

(4) Store/Load instruction to access control register B (ST.W, LD.W, etc.)

The same processing is also required when access to control registers and memory within the scope of protection starts after waiting for the completion of settings for safety functions such as memory protection, ECC checking, and so on.

Moreover, the same processing is also required when guarantee the completion of store accesses to memory.

Note 1. Dummy reading of any register of the same P-Bus Group can be used instead.

3.9.1.2 When the Updated Results in the Control Registers and Memories are Reflected in the Instruction Fetch of a Subsequent Instruction:

1. If you wish to write an instruction to the RAM and then branch to the RAM to execute the written instruction, do this as follows.
   (1) Store instruction to update a memory (ST.W, etc.)
   (2) Dummy reading of the above-mentioned memory (LD.W, etc.)
   (3) SYNCI
   (4) Subsequent instruction (branch instruction, etc.)
2. When branching to a target memory after waiting for the completion of updating the control registers for memory protection and ECC, do this as follows.
   (1) Store instruction to update a control register (ST.W, etc.)
   (2) Dummy reading of the control register (LD.W, etc.)
   (3) SYNCI
   (4) Subsequent instruction (branch instruction, etc.)

3.9.1.3 Product information of SYNCM

The following table shows the correspondence between the module names given in **Table 3.136, Bus Slaves that Support Using the SYNCM Instruction to Wait for the Completion of Storage in This CPU** and the module names in this product information document.

Table 3.202 Module Name Correspondence in this Product Information Document

Module name in *Table 3.136*	Module name of this product information document
L1RAM (given CPU)	Local RAM (own core)
L1RAM (other CPU)	Local RAM (other core)
L2RAM	Cluster 0/1/2/3 RAM
INTC1	INTC1
Others	There is no module corresponding to "Others" in this product

3.9.1.4 Product information of Waiting for the Completion of Storage

Table 3.203 Bus Slaves for which Using Dummy Reading to Wait is Effective

Bus Slaves	Description
L1RAM (given CPU)	Local RAM (own core)
L1RAM (other CPU)	Local RAM (other core)
L2RAM	Cluster 0/1/2/3 RAM
INTC1	INTC1
Others	All peripheral module

3.9.2 Synchronization of Load Instruction Completion and Subsequent Instruction Generation

When a control register and memory is accessed by a load instruction, the order of data accesses does not guarantee the same order of program. Therefore, appropriate processing for synchronization is required to ensure the completion of a load instruction for the subsequent data access instructions. However, the synchronization processing is not necessary for accesses to some target data space. For details on target data space, refer to **Section 4.2.2, Data Space Accessible by CPUs**. The processing for load synchronization is shown below.

Example) It is necessary to wait for the completion of a load access to an address A before an access to other address B, proceed as follows in this case.

(1) Load instruction to refer Address A (LD.W, etc.*1)

(2) SYNCP

(3) Store/Load instruction to access Address B (ST.W, LD.W, etc.*1)

Note 1. Load instruction: LD.*, SLD.*, LDV.*, TST1, SET1, CLR1, NOT1, POPSP, RESBANK, DISPOSE, SWITCH, CALLT, SYSCALL, CAXI, LDL
Store instruction: ST.*, SST.*, STV.*, SET1, CLR1, NOT1, PUSHSP, PREPARE, CAXI, STC

3.9.3 Accesses to Registers by Bit-Manipulation Instructions

Processing of a bit-manipulation instruction takes the form of atomic reading, modification, and writing of an eight-bit unit. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. However, take care in the cases of registers that contain multiple flag bits, since the read-modify-write cycle may also clear flags other than those which were for clearing.

Write access to H-Bus group registers by using bit-manipulation instructions is not atomic. Access by other masters may interrupt the read-modify-write processing of these instructions.

3.9.4 Ensuring Coherency after Code Flash Programming

For PE0 and PE1, see **Section 3.2.9, Ensuring Coherency after Code Flash Programming**.

For ICUMHA, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

3.9.5 Overwriting Context when Acknowledging Multiple Exceptions

Exceptions may be acknowledged regardless of the states of the ID or NP bits of the PSW register. This depends on the type of exception. When multiple exceptions occur, the contents of the system registers which hold the context information are overwritten. Regarding the conditions for acknowledging exceptions from each source and the possibility of return and recovery, see **Section 3.2.4, Exceptions and Interrupts**.

3.9.6 Usage Notes on Prefetching

CPU executes speculative instruction fetching from locations after the current value of the program counter to maintain the throughput of instruction fetches. Reading from memory due to such prefetching may proceed even from locations to which instruction codes have not been assigned (Note 1. in **Figure 3.62**). Note the following and keep in mind that the CPU does not execute values read in such cases.

The following notes apply to instruction fetching from memory in general.

- Occurrence of ECC errors due to values in memory being undefined

This prefetching may lead to an ECC error in case of reading from the code flash memory after it has been erased or from the Local RAM or Cluster RAM before initialization. When instruction codes are assigned to memory, initialize said area with values as desired (Note 1. in **Figure 3.62**).

- Detection of illegal access by the Slave Guard (CRG, PEG, etc.)

The Slave Guard may detect such prefetching as illegal access. To prevent prefetching being detected as an illegal access, do not allow any region of overlap area with said areas (Note 1. in **Figure 3.62**) and areas to which access is prohibited by the Slave guard. Prefetch from an reading from an area protected by the MPU does not cause a memory protection exception.

- Access to Access Prohibited Area

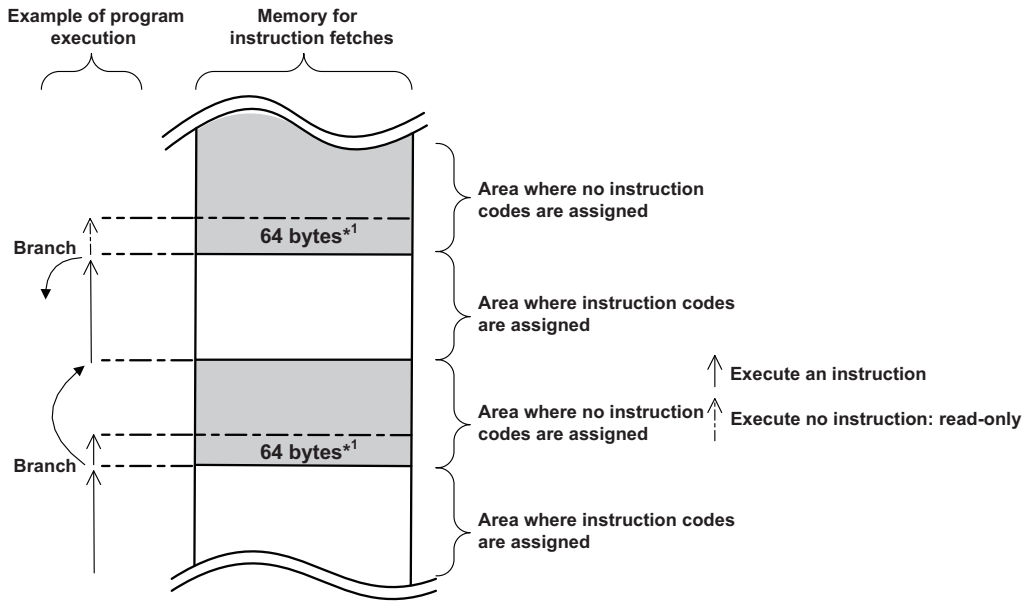
Prefetch can also cause access to unimplemented area.

This caution needs to be taken care also for RAM instruction fetch.

Assign instruction codes to memory without allowing any overlap between said area (Note 1. in **Figure 3.62**) and an access-prohibited area.

- Speculative read operation by Prefetch

Depending on the decode result of prefetch instruction, the CPU may issue speculative read request to Local RAM (own core), Local RAM (other core), Cluster RAM or Code Flash. The lockstep error described in **Section 44.4.5, Usage Note** may be occurred by speculative read request. If it is unnecessary as a result of branch instruction, the data is discarded.



Note 1. Area from final address of instruction code to final address + 64 bytes. Even if instruction codes are not placed in this area, requests for prefetching may lead to reading from memory at the time of branching

Figure 3.62 Areas that Require Attention Regarding Prefetching

3.9.7 Product information of initial value for G4MH register

The G4MH register's initial values of the product dependent are shown below.

Table 3.204 The initial value for G4MH register

Register No	Symbol	Initial value
SR0,1	SPID, HMSPID*7	*1
SR1,1	SPIDLIST, HMSPIDLIST*7	FFFF FFFF$_H$
SR2,1	RBASE	*2
SR6,1	PID	0784 0700$_H$*3
SR16,1	HVCFG	*4
SR0,2	PEID, HMPEID*7	*5
SR1,2	BMID	*6
SR24,4	ICCTRL	0001 0003$_H$
SR26,4	ICCFG	0000 1044$_H$
SR16,9	GMSPID*7	*1
SR17,9	GMSPIDLIST*7	FFFF FFFF$_H$
SR30,9	GMPEID*7	*5
SR12,13	L1RCFG	*11
SR2,12	LSCFG	*8
SR2,5	MPCFG	0002 001F$_H$*9
SR0,3	DBGEN*7	0000 01FF$_H$
SR28,3	DBCFG*10	0000 000B$_H$

Note 1. The initial values of SPID/GMSPID register are listed below.
CPU0: 0000 0000$_H$
CPU1: 0000 0001$_H$
CPU2: 0000 0002$_H$
CPU3: 0000 0003$_H$

Note 2. See **Section 4, Address Space** about RBASE register's bit to 31 from 9. Initial value of RBASE.DV bit is 0 and Initial value of RBASE.RINT bit is 0.

Note 3. Bit 23 relates to Virtualization features. See the *RH850G4MH Virtualization User's Manual: Hardware*.

Note 4. The initial value of HVCFG register is defined by Option Byte. See **Section 51, Flash Memory** for details.

Note 5. The initial values of PEID/GMPEID register are listed below.
CPU0: 0000 0000$_H$
CPU1: 0000 0001$_H$
CPU2: 0000 0002$_H$
CPU3: 0000 0003$_H$

Note 6. The initial values of BMID register are listed below.
CPU0: 0000 0000$_H$
CPU1: 0000 0001$_H$
CPU2: 0000 0002$_H$
CPU3: 0000 0003$_H$

Note 7. These registers relate to Virtualization features. For details, see the *RH850G4MH Virtualization User's Manual: Hardware*.

Note 8. The initial value of LSCFG are listed below.
CPU0: 0000 0001$_H$
CPU1, CPU2, CPU3: is defined by Option Byte. See **Section 51, Flash Memory** for details.

Note 9. MPCFG.NBK = 0 means that there is only one MPU bank mounted in this product.

Note 10. This register relates to Debug features. For details, see the *RH850/U2A-EVA Group User's Manual: Emulation*.

Note 11. The initial values of L1RCFG register are listed below.
U2A-EVA, U2A16 and U2A6: 0000 4004$_H$
U2A8: 0000 4002$_H$

3.9.8 Product information of SYSERR factor

The product-dependent SYSERR causes are shown below. Refer to **Section 3.2.4, Exceptions and Interrupts**.

Table 3.205 Production information of SYSERR factor

Exception Cause Code*1	Cause
11_H	When any of the following errors occurs in instruction fetch • ADDRESS ECC error in Local RAM, Cluster RAM and Code Flash • Guard error in PEG, CRG and security guard.*2 • Address Feedback error in LRAM and Cluster RAM • An unimplemented area is fetch accessed
13_H	When any of the following errors occurs in instruction fetch • Uncorrectable DATA ECC error in Local RAM, Cluster RAM and Code Flash (including cases where 1-bit correction is prohibited) • Address parity error in Code Flash.

Note 1. Refer to **Section 3.2.4, Exceptions and Interrupts** about another exception cause code (excluding 11_H and 13_H)

Note 2. Guard error caused by security setting. For details of Security guard, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

NOTE

If system register SR16, 13 is accessed by write operation, SYSERR exception whose codes are different from the code in **Table 3.103, Lower-order 16 Bits of the Exception Cause Codes Associated with the SYSERR Exception** may occur. Therefore, write access to SR16,13 is prohibited.

When an error occurs in instruction fetch at the time of cache miss or cache invalid, following ECM error sources may not be notified and following error information registers may not be updated.

< ECM error sources >

No.80: Code Flash - Address parity error

No.81: Code Flash - Data ECC 2-bit error

No.82: Code Flash - Data ECC 1-bit error

No.83: Code Flash - ECC overflow error

< Error information registers>

ECMmESSTR2 CF_SERSTR

CF_DERSTR CF_OVFSTR

CF_SERINF CF_nSEADR

CF_00DEADR

However, by executing the following sequence in the SYSERR exception handler, the above notification and the update can be performed. In that case, note that the access information will be recorded as load since it is not fetch access.

- Clear(0) RDBCR.RDBEN
- Read memory area from FEPC to FEPC+7 using LD.H instruction

3.9.9 Product Information of Cache Structure

The product-dependent cache structure are shown below.

Table 3.206 Product information of Cache Structure

Cache structure	RH850/U2A-EVA	RH850/U2A16	RH850/U2A8	RH850/U2A6
Total Size	16 KB	16 KB	16 KB	16 KB
Number of Way	4 ways	4 ways	4 ways	4 ways
Number of Line	128 lines	128 lines	128 lines	128 lines

3.9.10 Product Information of Fetch Size

These product's CPU(RH850G4MH2) fetch size is 64-Bit.

3.9.11 Register Initialization

The CPU in this product has general purpose registers and system registers which value after reset are undefined. In case of lockstep core, initial value of the master core and checker core after reset may be different. Therefore, these registers must be initialized at first. The registers which need the initialization are shown in **Table 3.207**.

For Usage of Lockstep Error, see **Section 44, Functional Safety**.

Table 3.207 Registers must be Initialized

Type	Registers
General purpose register	r1 to r31
Basic system register	EIPC, FEPC, CTPC, EIWR, FEWR, EBASE, INTBP, MEA, MEI, RBIP
FPU system register *1	FPSR, FPEPC, FPST, FPCC
MPU function register	MCA, MCS, MCR, MPLA*3, MPUA*3, MPAT*3, MPIDn, MCI
Cache operation function register	ICTAGL, ICTAGH, ICDATL, ICDATH, ICERR
Virtualization support function system register*2	HVSB
Guest Context Register*2	GMEIPC, GMFEPC, GMEBASE, GMINTBP, GMEIWR, GMFEWR, GMMEA, GMMEI.

Note 1. When FPU is valid, the undefined bits of these registers must be initialized to 0.

Note 2. When the virtualization support function is valid, the undefined bits of these registers must be initialized to 0. For details of those registers, see the *RH850G4MH Virtualization User's Manual: Hardware.*

Note 3. The registers of all the 32 MPU entries should be initialized. Set the index register MPIDX from 0 to 31, and initialize the corresponding MPLA, MPUA, MPAT registers.

3.9.12 Error Notifications

The error notifications of this module are listed in the following table.

Table 3.208 Error Notifications

Error Notification	Description	ECM error Number	Error Response to bus master
Multi-hit error (PE0)	Instruction cache Multi-hit error	—	√
Way error (PE0)	Instruction cache Way error	—	√
Multi-hit error (PE1)	Instruction cache Multi-hit error	—	√
Way error (PE1)	Instruction cache Way error	—	√
Multi-hit error (PE2)	Instruction cache Multi-hit error	—	√
Way error (PE2)	Instruction cache Way error	—	√
Multi-hit error (PE3)	Instruction cache Multi-hit error	—	√
Way error (PE3)	Instruction cache Way error	—	√
Data Access Error (PE0)*1	PE0 data access error	255	—
Data Access Error (PE1)*1	PE1 data access error	287	—
Data Access Error (PE2)*1	PE2 data access error	319	—
Data Access Error (PE3)*1	PE3 data access error	351	—

Note 1. This error is generated due to the error response caused by some error.
For details of the error response caused by some error, see the column for Error Response to bus master in Error Notifications Table shown in each section.

3.9.13 Target for LDL, STC and CAXI Instructions

LDL, STC and CAXI instructions target only local RAM and cluster RAM.

Section 4 Address Space

4.1 Overview

Table 4.1 and **Table 4.2** show the address space for each product.

For further details of Code Flash and Data Flash area, see **Section 51, Flash Memory**.

For details of ECC Test Area, see the *RH850/U2Ax Safety Application Note*.

Table 4.1 Address Space (Single Map Mode) (1/3)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available					Access from		
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6	CPU	DMA modules	H-Bus modules
0000 0000$_H$	002F FFFF$_H$	Code flash (User Area 0) (Bank A)	√	√	√	√	√			
0030 0000$_H$	003F FFFF$_H$		√	√	√	√	—			
0040 0000$_H$	006F FFFF$_H$	Code flash (User Area 1) (Bank B)	√	√	√	√	√			
0070 0000$_H$	007F FFFF$_H$		√	√	√	√	—			
0080 0000$_H$	00BF FFFF$_H$	Code flash (User Area 2) (Bank C)	√	—	√	—	—			
00C0 0000$_H$	00FF FFFF$_H$	Code flash (User Area 3) (Bank D)	√	—	√	—	—			
0100 0000$_H$	03FF FFFF$_H$	Reserved	—	—	—	—	—			
0400 0000$_H$	07FF FFFF$_H$	Code flash (Non-overlay area)*1	√	√	√	√	√			
0800 0000$_H$	0800 FFFF$_H$	Code flash (User Boot Area 0) (Bank A)	√	√	√	√	√			
0801 0000$_H$	0802 FFFF$_H$	Reserved	—	—	—	—	—			
0803 0000$_H$	0803 7FFF$_H$	Code flash (Product Info Area 0) (Bank A)	√	√	√	√	√			
0803 8000$_H$	0804 FFFF$_H$	Reserved	—	—	—	—	—			
0805 0000$_H$	0805 FFFF$_H$	Code flash (ECC Test Area of Bank A)	√	√	√	√	√			
0806 0000$_H$	083F FFFF$_H$	Reserved	—	—	—	—	—			
0840 0000$_H$	0840 FFFF$_H$	Code flash (User Boot Area 1) (Bank B)	√	√	√	√	√			
0841 0000$_H$	0842 FFFF$_H$	Reserved	—	—	—	—	—			
0843 0000$_H$	0843 7FFF$_H$	Code flash (Product Info Area 1) (Bank B)	√	√	√	√	√			
0843 8000$_H$	0844 FFFF$_H$	Reserved	—	—	—	—	—			
0845 0000$_H$	0845 FFFF$_H$	Code flash (ECC Test Area of Bank B)	√	√	√	√	√			
0846 0000$_H$	0884 FFFF$_H$	Reserved	—	—	—	—	—			
0885 0000$_H$	0885 FFFF$_H$	Code flash (ECC Test Area of Bank C)	√	—	√	—	—			
0886 0000$_H$	08C4 FFFF$_H$	Reserved	—	—	—	—	—			
08C5 0000$_H$	08C5 FFFF$_H$	Code flash (ECC Test Area of Bank D)	√	—	√	—	—			
08C6 0000$_H$	0BFE FFFF$_H$	Reserved	—	—	—	—	—			
0BFF 0000$_H$	0BFF FFFF$_H$	Code flash (ECC Test Area of Global Area)	√	√	√	√	√			
0C00 0000$_H$	0FFF FFFF$_H$	Code flash (Blank Check Area)*2	√	√	√	√	√			
1000 0000$_H$	3FFF FFFF$_H$	H-Bus area	√	√	√	√	√			*3
4000 0000$_H$	F900 FFFF$_H$	Reserved	—	—	—	—	—			
F901 0000$_H$	F901 FFFF$_H$	Debug (CPU0)	√	√	√	√	√			
F902 0000$_H$	F902 FFFF$_H$	Debug (CPU1)	√	√	√	√	√			
F903 0000$_H$	F903 FFFF$_H$	Debug (CPU2)	√	—	√	—	—			
F904 0000$_H$	F904 FFFF$_H$	Debug (CPU3)	√	—	√	—	—			
F905 0000$_H$	F9FF FFFF$_H$	Reserved	—	—	—	—	—			
FA00 0000$_H$	FA01 7FFF$_H$	Instrumentation RAM	√	√	—	—	—			
FA01 8000$_H$	FA0F FFFF$_H$	Reserved	—	—	—	—	—			
FA10 0000$_H$	FA10 3FFF$_H$	Trace filter RAM (Cluster RAM) (Cluster 0)	—	—	—	—	√			
FA10 4000$_H$	FA10 7FFF$_H$	Reserved	—	—	—	—	—			
FA10 8000$_H$	FA10 BFFF$_H$	Trace filter RAM (Cluster RAM) (Cluster 1)	—	—	—	—	—			
FA10 C000$_H$	FA11 FFFF$_H$	Reserved	—	—	—	—	—			

Table 4.1 Address Space (Single Map Mode) (2/3)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available					Access from		
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6	CPU	DMA modules	H-Bus modules
FA12 0000$_H$	FA12 FFFF$_H$	Trace filter RAM (Cluster RAM) (Cluster 2)	—	—	—	—	—			
FA13 0000$_H$	FA13 FFFF$_H$	Reserved	—	—	—	—	—			
FA14 0000$_H$	FA14 0FFF$_H$	Trace filter RAM (Cluster RAM) (Cluster 3)	—	—	—	—	√			
FA14 1000$_H$	FA14 1FFF$_H$		—	—	—	—	—			
FA14 2000$_H$	FA1A FFFF$_H$	Reserved	—	—	—	—	—			
FA1B 0000$_H$	FA1B 07FF$_H$	Trace filter RAM (Local RAM) (CPU3)	—	—	—	—	—			
FA1B 0800$_H$	FA1B FFFF$_H$	Reserved	—	—	—	—	—			
FA1C 0000$_H$	FA1C 07FF$_H$	Trace filter RAM (Local RAM) (CPU2)	—	—	—	—	—			
FA1C 0800$_H$	FA1C FFFF$_H$	Reserved	—	—	—	—	—			
FA1D 0000$_H$	FA1D 07FF$_H$	Trace filter RAM (Local RAM) (CPU1)	—	—	—	—	√			
FA1D 0800$_H$	FA1D FFFF$_H$	Reserved	—	—	—	—	—			
FA1E 0000$_H$	FA1E 07FF$_H$	Trace filter RAM (Local RAM) (CPU0)	—	—	—	—	√			
FA1E 0800$_H$	FAFF FFFF$_H$	Reserved	—	—	—	—	—			
FB00 0000$_H$	FB00 1FFF$_H$	Cluster0 ERAM	√	√	—	—	√			
FB00 2000$_H$	FB01 FFFF$_H$		√	√	—	—	—			
FB02 0000$_H$	FB02 1FFF$_H$		√	√	—	—	√			
FB02 2000$_H$	FB03 FFFF$_H$		√	√	—	—	—			
FB04 0000$_H$	FB04 1FFF$_H$		√	√	—	—	√			
FB04 2000$_H$	FB05 FFFF$_H$		√	√	—	—	—			
FB06 0000$_H$	FB06 1FFF$_H$		√	√	—	—	√			
FB06 2000$_H$	FB0F FFFF$_H$		√	√	—	—	—			
FB10 0000$_H$	FB1F FFFF$_H$	Cluster1 ERAM	√	—	—	—	—			
FB20 0000$_H$	FB3F FFFF$_H$	Reserved	—	—	—	—	—			
FB40 0000$_H$	FB5F FFFF$_H$	Global Emulation RAM	√	√	—	—	—			
FB60 0000$_H$	FBFF FFFF$_H$	Reserved	—	—	—	—	—			
FC00 0000$_H$	FD5F FFFF$_H$	Reserved	—	—	—	—	—			
FD60 0000$_H$	FD60 FFFF$_H$	Local RAM (CPU3)	√	—	√	—	—			
FD61 0000$_H$	FD7F FFFF$_H$	Reserved	—	—	—	—	—			
FD80 0000$_H$	FD80 FFFF$_H$	Local RAM (CPU2)	√	—	√	—	—			
FD81 0000$_H$	FD9F FFFF$_H$	Reserved	—	—	—	—	—			
FDA0 0000$_H$	FDA0 FFFF$_H$	Local RAM (CPU1)	√	√	√	√	√			
FDA1 0000$_H$	FDBF FFFF$_H$	Reserved	—	—	—	—	—			
FDC0 0000$_H$	FDC0 FFFF$_H$	Local RAM (CPU0)	√	√	√	√	√			
FDC1 0000$_H$	FDDF FFFF$_H$	Reserved	—	—	—	—	—			
FDE0 0000$_H$	FDE0 FFFF$_H$	Local RAM (self)	√	√	√	√	√			
FDE1 0000$_H$	FDFF FFFF$_H$	Reserved	—	—	—	—	—			
FE00 0000$_H$	FE07 FFFF$_H$	Cluster RAM (Cluster0)	√	√	√	√	√			
FE08 0000$_H$	FE0F FFFF$_H$	Reserved	—	—	—	—	—			
FE10 0000$_H$	FE17 FFFF$_H$	Cluster RAM (Cluster1)	√	—	√	—	—			
FE18 0000$_H$	FE3F FFFF$_H$	Reserved	—	—	—	—	—			
FE40 0000$_H$	FE4F FFFF$_H$	Cluster RAM (Cluster2)	√	√	√	√	—			
FE50 0000$_H$	FE5F FFFF$_H$		√	—	√	—	—			
FE60 0000$_H$	FE7F FFFF$_H$	Reserved	—	—	—	—	—			
FE80 0000$_H$	FE81 FFFF$_H$	Cluster RAM (Cluster3) (Retention RAM)	√	√	√	√	√			
FE82 0000$_H$	FE83 FFFF$_H$		√	—	√	—	—			
FE84 0000$_H$	FEFF FFFF$_H$	Reserved	—	—	—	—	—			

Table 4.1 Address Space (Single Map Mode) (3/3)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available					Access from		
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6	CPU	DMA modules	H-Bus modules
FF00 0000$_H$	FFFB 7FFF$_H$	P-Bus area	√	√	√	√	√			
(FF20 0000$_H$	FF28 FFFF$_H$	Data Flash (Data Area))								
(FF32 0000$_H$	FF37 4FFF$_H$	Data Flash (Hardware Property Area))								
(FF40 0000$_H$	FF5F FFFF$_H$	Data Flash (Blank Check Area))								
FFFB 8000$_H$	FFFB FFFF$_H$	I-Bus area	√	√	√	√	√			
FFFC 0000$_H$	FFFC 3FFF$_H$	CPU peripheral (self)	√	√	√	√	√			
FFFC 4000$_H$	FFFC 7FFF$_H$	CPU peripheral (CPU0)	√	√	√	√	√			
FFFC 8000$_H$	FFFC BFFF$_H$	CPU peripheral (CPU1)	√	√	√	√	√			
FFFC C000$_H$	FFFC FFFF$_H$	CPU peripheral (CPU2)	√	—	√	—	—			
FFFD 0000$_H$	FFFD 3FFF$_H$	CPU peripheral (CPU3)	√	—	√	—	—			
FFFD 4000$_H$	FFFF FFFF$_H$	Reserved	—	—	—	—	—			

Note 1. Non-overlay mirror area of 0000 0000$_H$ - 03FF FFFF$_H$

Note 2. Blank Check Area for 0000 0000$_H$ - 03FF FFFF$_H$ or 0800 0000$_H$ - 0BFF FFFF$_H$

Note 3. Only RHSIF can access to H-Bus slave modules. For other H-Bus master, this area poses as reserved area. [For U2A-EVA, U2A16 and U2A8 Only]

: Fetch and data access available

: Data access available

: Access prohibited

Table 4.2 Address Space (Double Map Mode)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available					Access from		
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6	CPU	DMA modules	H-Bus modules
0000 0000$_H$	002F FFFF$_H$	Code flash (User Area Valid Area) (Bank A or B)	√	√	√	√	√			
0030 0000$_H$	003F FFFF$_H$		√	√	√	√	—			
0040 0000$_H$	007F FFFF$_H$	Code flash (User Area Valid Area) (Bank C or D)	√	—	√	—	—			
0080 0000$_H$	01FF FFFF$_H$	Reserved	—	—	—	—	—			
0200 0000$_H$	022F FFFF$_H$	Code flash (User Area Invalid Area) (Bank B or A)	√	√	√	√	√			
0230 0000$_H$	023F FFFF$_H$		√	√	√	√	—			
0240 0000$_H$	027F FFFF$_H$	Code flash (User Area Invalid Area) (Bank D or C)	√	—	√	—	—			
0280 0000$_H$	03FF FFFF$_H$	Reserved	—	—	—	—	—			
0400 0000$_H$	07FF FFFF$_H$	Code flash (Non-overlay area)*1	√	√	√	√	√			
0800 0000$_H$	0800 FFFF$_H$	Code flash (User Boot Area Valid Area) (Bank A or B)	√	√	√	√	√			
0801 0000$_H$	0802 FFFF$_H$	Reserved	—	—	—	—	—			
0803 0000$_H$	0803 7FFF$_H$	Code flash (Product Info Area Valid Area) (Bank A or B)	√	√	√	√	√			
0803 8000$_H$	0804 FFFF$_H$	Reserved	—	—	—	—	—			
0805 0000$_H$	0805 FFFF$_H$	Code flash (ECC Test Area of Bank A or B)	√	√	√	√	√			
0806 0000$_H$	0844 FFFF$_H$	Reserved	—	—	—	—	—			
0845 0000$_H$	0845 FFFF$_H$	Code flash (ECC test Area of Bank C or D)	√	—	√	—	—			
0846 0000$_H$	09FF FFFF$_H$	Reserved	—	—	—	—	—			
0A00 0000$_H$	0A00 FFFF$_H$	Code flash (User Boot Area Invalid Area) (Bank B or A)	√	√	√	√	√			
0A01 0000$_H$	0A02 FFFF$_H$	Reserved	—	—	—	—	—			
0A03 0000$_H$	0A03 7FFF$_H$	Code flash (Product Info Area Invalid Area) (Bank B or A)	√	√	√	√	√			
0A03 8000$_H$	0A04 FFFF$_H$	Reserved	—	—	—	—	—			
0A05 0000$_H$	0A05 FFFF$_H$	Code flash (ECC Test Area of Bank B or A)	√	√	√	√	√			
0A06 0000$_H$	0A44 FFFF$_H$	Reserved	—	—	—	—	—			
0A45 0000$_H$	0A45 FFFF$_H$	Code flash (ECC Test Area of Bank D or C)	√	—	√	—	—			
0A46 0000$_H$	0BFE FFFF$_H$	Reserved	—	—	—	—	—			
0BFF 0000$_H$	0BFF FFFF$_H$	Code flash (ECC Test Area of Global Area)	√	√	√	√	√			
0C00 0000$_H$	0FFF FFFF$_H$	Code flash (Blank Check Area)*2	√	√	√	√	√			
1000 0000$_H$	FFFF FFFF$_H$	This address area does not change even in double map mode. See **Table 4.1**.								

Note 1. Non-overlay mirror area of 0000 0000$_H$ - 03FF FFFF$_H$

Note 2. Blank Check Area for 0000 0000$_H$ - 03FF FFFF$_H$ or 0800 0000$_H$ - 0BFF FFFF$_H$

: Fetch and data access available

: Data access available

: Access prohibited

4.2 Address Space Viewed from Each Bus Master

4.2.1 Space in which instructions can be fetched

1. Instructions of CPUs can be fetched from the code flash, local RAM and Cluster RAM.
2. The reset vector (RBASE initial value) of CPUs:
   - In User Boot Mode with CPU0, its head address is 0800 0000_H.
   - In User Boot Mode with CPUn (n:1-3), its head address is same as in Normal Operation Mode.
   - In Normal Operation Mode with CPUn (n:0-3), its head address can be set within Code Flash (User Area) by Reset Vector PEn (OPT_RBASEn).
     If Code Flash is in Double Map Mode, it can be selected within Valid area.
     (The size of User Area depends on the products. See **Section 51.3.1, Mapping of Code Flash Memory** for each variant. See also **Section 51.12, Configuration Setting Area (Option Bytes, Reset Vector)** for setting of OPT_RBASEn.)

NOTE

For assignment of instruction codes, see **Section 3.9.6, Usage Notes on Prefetching**.

4.2.2 Data Space Accessible by CPUs

See **Section 4.1, Overview** for the accessible spaces from CPUs.

NOTE

The data accesses inside the following target data space are operated in the same order as the program.

- Code Flash
- H-Bus area Group 0, 1, 2, 3
- Debug (CPU0, CPU1, CPU2, CPU3)
- Instrumentation RAM
- Cluster ERAM (Cluster 0, 1)
- Global Emulation RAM
- Cluster RAM (Cluster 0, 1, 2, 3)
- P-Bus area Group 0, 1, 2H/2L, 3, 4, 5, 6H/6L, 7, 8, 9
- I-Bus area Group 0, 1, 2
- CPU peripheral area (self, CPU0, CPU1, CPU2, CPU3)

The load accesses inside the following target data space may be operated before the completion of the preceding access (load or store). In order to guarantee the order of accesses, synchronization processing by software for the preceding access is necessary.

- Local RAM (self, CPU0, CPU1, CPU2, CPU3)

For the data accesses between different target spaces, the order of data accesses is not guaranteed as the program order.

In order to guarantee the order of accesses, synchronization processing by software for the preceding access is necessary.

For details of synchronization processing, refer to **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** and **Section 3.9.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation**.

4.2.3 Data Space Accessible by DMA modules

See **Section 4.1, Overview** for the accessible spaces from DMA modules.

4.2.4 Data Space Accessible by H-Bus modules

See **Section 4.1, Overview** for the accessible spaces from H-Bus modules.

NOTE

Only RHSIF can access to H-Bus slave modules. For other H-Bus master, this area poses as reserved area. [For U2A-EVA, U2A16 and U2A8 Only]

4.3 Error notification for an access to unmapped area

When any master accesses to “unmapped area” in which the significant resource is not assigned, error response via bus to the access master is signaled. See **Table 4.3** for error response to bus master. Regarding of the reaction for an error response of each area, see **Section 4.3.1** to **Section 4.3.9**.

Table 4.3 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
Unmapped Area Error	Unmapped area access error	—	√

4.3.1 Unmapped Code Flash area access error

Error response will be notified if accessed to the area represented as "Not available" in **Section 4.1, Overview**.

Table 4.4 and **Table 4.5** describe the area that notify error response for access.

For details of Code Flash and Data Flash area, see **Section 51, Flash Memory**.

Table 4.4 Error Notifications of Code Flash area (Single Map Mode) (1/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
0000 0000$_H$	002F FFFF$_H$	Code flash (User Area 0) (Bank A)	—	—	—	—	—
0030 0000$_H$	003F FFFF$_H$		—	—	—	—	ER
0040 0000$_H$	006F FFFF$_H$	Code flash (User Area 1) (Bank B)	—	—	—	—	—
0070 0000$_H$	007F FFFF$_H$		—	—	—	—	ER
0080 0000$_H$	00BF FFFF$_H$	Code flash (User Area 2) (Bank C)	—	ER	—	ER	ER
00C0 0000$_H$	00FF FFFF$_H$	Code flash (User Area 3) (Bank D)	—	ER	—	ER	ER
0100 0000$_H$	03FF FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0400 0000$_H$	07FF FFFF$_H$	Code flash (Non-overlay area)	*2	*2	*2	*2	*2
0800 0000$_H$	0800 FFFF$_H$	Code flash (User Boot Area 0) (Bank A)	—	—	—	—	—
0801 0000$_H$	0802 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0803 0000$_H$	0803 7FFF$_H$	Code flash (Product Info Area 0) (Bank A)	—	—	—	—	—
0803 8000$_H$	0804 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0805 0000$_H$	0805 FFFF$_H$	Code flash (ECC Test Area of Bank A)	*3	*3	*3	*3	*3
0806 0000$_H$	082F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0830 0000$_H$	083F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER
0840 0000$_H$	0840 FFFF$_H$	Code flash (User Boot Area 1) (Bank B)	—	—	—	—	—
0841 0000$_H$	0842 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0843 0000$_H$	0843 7FFF$_H$	Code flash (Product Info Area 1) (Bank B)	—	—	—	—	—
0843 8000$_H$	0844 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0845 0000$_H$	0845 FFFF$_H$	Code flash (ECC Test Area of Bank B)	*3	*3	*3	*3	*3
0846 0000$_H$	086F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0870 0000$_H$	087F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER
0880 0000$_H$	0884 FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER

Table 4.4 Error Notifications of Code Flash area (Single Map Mode) (2/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
0885 0000$_H$	0885 FFFF$_H$	Code flash (ECC Test Area of Bank C)	*3	ER	*3	ER	ER
0886 0000$_H$	08C4 FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER
08C5 0000$_H$	08C5 FFFF$_H$	Code flash (ECC Test Area of Bank D)	*3	ER	*3	ER	ER
08C6 0000$_H$	08FF FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER
0900 0000$_H$	0BFE FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0BFF 0000$_H$	0BFF FFFF$_H$	Code flash (ECC Test Area of Global Area)	*3	*3	*3	*3	*3
0C00 0000$_H$	0FFF FFFF$_H$	Code flash (Blank Check Area)	*4	*4	*4	*4	*4

Note 1. The following errors also occur.
* ECC 2-bit error
* Address Parity Error depending on the address

Note 2. Same as 0000 0000$_H$ to 03FF FFFF$_H$

Note 3. For details of ECC Test Area, see the *RH850/U2Ax Safety Application Note*.

Note 4. See **Section 51, Flash Memory**.

Table 4.5 Error Notifications of Code Flash area (Double Map Mode) (1/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
0000 0000$_H$	002F FFFF$_H$	Code flash (User Area Valid Area) (Bank A or B)	—	—	—	—	—
0030 0000$_H$	003F FFFF$_H$		—	—	—	—	ER
0040 0000$_H$	007F FFFF$_H$	Code flash (User Area Valid Area) (Bank C or D)	—	ER	—	ER	ER
0080 0000$_H$	01FF FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0200 0000$_H$	022F FFFF$_H$	Code flash (User Area Invalid Area) (Bank B or A)	—	—	—	—	—
0230 0000$_H$	023F FFFF$_H$		—	—	—	—	ER
0240 0000$_H$	027F FFFF$_H$	Code flash (User Area Invalid Area) (Bank D or C)	—	ER	—	ER	ER
0280 0000$_H$	03FF FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0400 0000$_H$	07FF FFFF$_H$	Code flash (Non-overlay area)	*2	*2	*2	*2	*2
0800 0000$_H$	0800 FFFF$_H$	Code flash (User Boot Area Valid Area) (Bank A or B)	—	—	—	—	—
0801 0000$_H$	0802 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0803 0000$_H$	0803 7FFF$_H$	Code flash (Product Info Area Valid Area) (Bank A or B)	—	—	—	—	—
0803 8000$_H$	0804 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0805 0000$_H$	0805 FFFF$_H$	Code flash (ECC Test Area of Bank A or B)	*3	*3	*3	*3	*3
0806 0000$_H$	082F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0830 0000$_H$	083F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER
0840 0000$_H$	0844 FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER
0845 0000$_H$	0845 FFFF$_H$	Code flash (ECC test Area of Bank C or D)	*3	ER	*3	ER	ER
0846 0000$_H$	087F FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER

Table 4.5 Error Notifications of Code Flash area (Double Map Mode) (2/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
0880 0000$_H$	09FF FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0A00 0000$_H$	0A00 FFFF$_H$	Code flash (User Boot Area Invalid Area) (Bank B or A)	—	—	—	—	—
0A01 0000$_H$	0A02 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0A03 0000$_H$	0A03 7FFF$_H$	Code flash (Product Info Area Invalid Area) (Bank B or A)	—	—	—	—	—
0A03 8000$_H$	0A04 FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0A05 0000$_H$	0A05 FFFF$_H$	Code flash (ECC Test Area of Bank B or A)	*3	*3	*3	*3	*3
0A06 0000$_H$	0A2F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER*1
0A30 0000$_H$	0A3F FFFF$_H$	Reserved	ER*1	ER*1	ER*1	ER*1	ER
0A40 0000$_H$	0A44 FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER
0A45 0000$_H$	0A45 FFFF$_H$	Code flash (ECC Test Area of Bank D or C)	*3	ER	*3	ER	ER
0A46 0000$_H$	0A7F FFFF$_H$	Reserved	ER*1	ER	ER*1	ER	ER
0A80 0000$_H$	0BFE FFFF$_H$	Reserved	ER	ER	ER	ER	ER
0BFF 0000$_H$	0BFF FFFF$_H$	Code flash (ECC Test Area of Global Area)	*3	*3	*3	*3	*3
0C00 0000$_H$	0FFF FFFF$_H$	Code flash (Blank Check Area)	*4	*4	*4	*4	*4
1000 0000$_H$	FFFF FFFF$_H$	This address area does not change even in double map mode. See **Table 4.1**.					

Note 1. The following errors also occur.
* ECC 2-bit error
* Address Parity Error depending on the address

Note 2. Same as 0000 0000$_H$ to 03FF FFFF$_H$

Note 3. For details of ECC Test Area, see the *RH850/U2Ax Safety Application Note*.

Note 4. See **Section 51, Flash Memory**.

4.3.2 Unmapped Cluster RAM area access error

Error response will be notified if accessed to the area represented as "Not available" in **Section 4.1, Overview**.

Table 4.6 Error Notifications of Cluster RAM area

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
FE00 0000$_H$	FE07 FFFF$_H$	Cluster RAM (Cluster0)	—	—	—	—	—
FE08 0000$_H$	FE0F FFFF$_H$	Reserved	ER	ER	ER	ER	ER
FE10 0000$_H$	FE17 FFFF$_H$	Cluster RAM (Cluster1)	—	ER	—	ER	ER
FE18 0000$_H$	FE3F FFFF$_H$	Reserved	ER	ER	ER	ER	ER
FE40 0000$_H$	FE4F FFFF$_H$	Cluster RAM (Cluster2)	—	—	—	—	ER
FE50 0000$_H$	FE5F FFFF$_H$		—	ER*1	—	ER*1	ER
FE60 0000$_H$	FE7F FFFF$_H$	Reserved	ER	ER	ER	ER	ER
FE80 0000$_H$	FE81 FFFF$_H$	Cluster RAM (Cluster3) (Retention RAM)	—	—	—	—	—
FE82 0000$_H$	FE83 FFFF$_H$		—	ER*1	—	ER*1	ER*1
FE84 0000$_H$	FEFF FFFF$_H$	Reserved	ER	ER	ER	ER	ER

Note 1. CRG (Cluster RAM Guard) error will be occurred even without guard setting when access to this area.

4.3.3 Unmapped Local RAM area access error

Error response will be notified if accessed to the area represented as "Not available" in **Section 4.1, Overview**.

Error response will be notified if accessed to Local RAM (self) by DMA/DTS or H-Bus.

Availability of error notification depends on PEx_DISABLE setting in OPBT3, if accessed to Local RAM (CPUx). (x:1-3)

4.3.4 Unmapped CPU peripheral area access error

Error response will be notified if accessed to the area represented as "Not available" in **Section 4.1, Overview**.

Error response will be notified if accessed to CPU peripheral (self) by DMA/DTS or H-Bus.

Availability of error notification depends on PEx_DISABLE setting in OPBT3, if accessed to CPU peripheral area (CPUx). (x:1-3)

4.3.5 Details of P-Bus area

Error response will be notified if accessed to the area represented as "Unmapped area" and "Not available of other areas".

Table 4.7 List of P-Bus area access (1/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FF00 0000$_H$	FF00 00FF$_H$	RHSIF0_L1	√	√	√	√	√	√	√	—	—	—	—	#9
FF00 0100$_H$	FF04 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0000$_H$	FF05 001F$_H$	E7FR00	√	√	√	√	√	√	√	√	√	√	√	#9
FF05 0020$_H$	FF05 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0200$_H$	FF05 021F$_H$	E7FR01	√	√	√	√	√	√	√	√	√	√	√	#9
FF05 0220$_H$	FF05 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0400$_H$	FF05 041F$_H$	E7FR02	√	√	√	√	√	√	√	√	√	√	√	#9
FF05 0420$_H$	FF05 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0600$_H$	FF05 061F$_H$	E7FR10	√	√	√	√	√	√	√	—	—	—	—	#9
FF05 0620$_H$	FF05 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0800$_H$	FF05 081F$_H$	E7FR11	√	√	√	√	√	√	√	—	—	—	—	#9
FF05 0820$_H$	FF05 09FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF05 0A00$_H$	FF05 0A1F$_H$	E7FR12	√	√	√	√	√	√	√	—	—	—	—	#9
FF05 0A20$_H$	FF08 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF09 0000$_H$	FF09 003F$_H$	INTIF	√	√	√	√	√	√	√	√	√	√	√	#9
FF09 0040$_H$	FF09 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF09 0200$_H$	FF09 0203$_H$	INTIF (TPTMSEL)	√	√	√	√	√	√	√	√	√	√	√	#9
FF09 0204$_H$	FF09 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF09 0400$_H$	FF09 047F$_H$	DMATRGSEL	√	√	√	√	√	√	√	√	√	√	√	#9
FF09 0480$_H$	FF09 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF09 0600$_H$	FF09 063F$_H$	DMATRGSEL (DTSSELm)	√	√	√	√	√	√	√	√	√	√	√	#9
FF09 0640$_H$	FF09 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 0000$_H$	FF0A 007F$_H$	ECCCNT_A_V2A9	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 0080$_H$	FF0A 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 0200$_H$	FF0A 027F$_H$	ECCCNT_D_V2A9W	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 0280$_H$	FF0A 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 0400$_H$	FF0A 04FF$_H$	BECCCAP_V2A9	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 0500$_H$	FF0A 10FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 1100$_H$	FF0A 1103$_H$	PB9ECC	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 1104$_H$	FF0A 12FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 1300$_H$	FF0A 137F$_H$	PBG90	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 1380$_H$	FF0A 13FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 1400$_H$	FF0A 141F$_H$	PBGERRSLV90	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 1420$_H$	FF0A 19FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 1A00$_H$	FF0A 1A1F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 1A20$_H$	FF0A 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 2000$_H$	FF0A 27FF$_H$	ETNB0	√	√	√	√	√	√	√	√	√	—*1	√	#9
FF0A 2800$_H$	FF0A 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 3000$_H$	FF0A 301F$_H$	E7ME00	√	√	√	√	√	√	√	√	√	—*1	√	#9
FF0A 3020$_H$	FF0A 31FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 3200$_H$	FF0A 321F$_H$	E7ME01	√	√	√	√	√	√	√	√	√	—*1	√	#9
FF0A 3220$_H$	FF0A 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 4000$_H$	FF0A 47FF$_H$	ETNB1	√	—	√	√	√	√	√	—	—	—	—	#9

Table 4.7 List of P-Bus area access (2/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FF0A 4800$_H$	FF0A 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 5000$_H$	FF0A 50FF$_H$	ETNB1 (SGMII Interface Related Registers)	√	—	√	√	√	√	√	—	—	—	—	#9
FF0A 5100$_H$	FF0A 53FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 5400$_H$	FF0A 541F$_H$	E7GE00	√	—	√	√	√	√	√	—	—	—	—	#9
FF0A 5420$_H$	FF0A 55FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 5600$_H$	FF0A 561F$_H$	E7GE01	√	—	√	√	√	√	√	—	—	—	—	#9
FF0A 5620$_H$	FF0A 57FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 5800$_H$	FF0A 580F$_H$	ETNB1 (SGMII Interface Related Registers)	√	—	√	√	√	√	√	—	—	—	—	#9
FF0A 5810$_H$	FF0A 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0A 8000$_H$	FF0A 83FF$_H$	SPIDCTL	√	√	√	√	√	√	√	√	√	√	√	#9
FF0A 8400$_H$	FF0A FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0B 0000$_H$	FF0B 01FF$_H$	EPC	√	√	√	√	√	√	√	√	√	√	√	#9
FF0B 0200$_H$	FF0B FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 0000$_H$	FF0C 01FF$_H$	HB92MECC	√	√	√	√	√	√	√	√	√	√	√	#9
FF0C 0200$_H$	FF0C 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 0400$_H$	FF0C 05FF$_H$	HB92SECC	√	√	√	√	√	√	√	√	√	√	√	#9
FF0C 0600$_H$	FF0C 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 0800$_H$	FF0C 09FF$_H$	HB93MECC	√	√	√	√	√	√	√	—	—	—	—	#9
FF0C 0A00$_H$	FF0C 0BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 0C00$_H$	FF0C 0DFF$_H$	HB93SECC	√	√	√	√	√	√	√	—	—	—	—	#9
FF0C 0E00$_H$	FF0C 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 1000$_H$	FF0C 11FF$_H$	HB95MECC	√	√	√	√	√	√	√	√	√	—*1	√	#9
FF0C 1200$_H$	FF0C 13FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 1400$_H$	FF0C 15FF$_H$	HB94MECC	√	—	√	√	√	√	√	—	—	—	—	#9
FF0C 1600$_H$	FF0C 17FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 1800$_H$	FF0C 19FF$_H$	HB91MECC	√	√	√	√	√	√	√	—	—	—	—	#9
FF0C 1A00$_H$	FF0C 1BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 1C00$_H$	FF0C 1DFF$_H$	HB91SECC	√	√	√	√	√	√	√	—	—	—	—	#9
FF0C 1E00$_H$	FF0C 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0C 2000$_H$	FF0C 21FF$_H$	HB96SECC	√	√	√	√	√	√	√	√	√	—*1	—*1	#9
FF0C 2200$_H$	FF0C FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 0000$_H$	FF0D 000F$_H$	HBG92	√	√	√	√	√	√	√	√	√	√	√	#9
FF0D 0010$_H$	FF0D 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 0200$_H$	FF0D 020F$_H$	HBG93	√	√	√	√	√	√	√	—	—	—	—	#9
FF0D 0210$_H$	FF0D 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 0400$_H$	FF0D 040F$_H$	HBG96	√	√	√	√	√	√	√	√	√	—*1	—*1	#9
FF0D 0410$_H$	FF0D 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 0600$_H$	FF0D 060F$_H$	HBG91	√	√	√	√	√	√	√	—	—	—	—	#9
FF0D 0610$_H$	FF0D 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1000$_H$	FF0D 101F$_H$	HBGERRSLV92	√	√	√	√	√	√	√	√	√	√	√	#9
FF0D 1020$_H$	FF0D 11FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1200$_H$	FF0D 121F$_H$	HBGERRSLV93	√	√	√	√	√	√	√	—	—	—	—	#9
FF0D 1220$_H$	FF0D 13FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1400$_H$	FF0D 141F$_H$	HBGERRSLV96	√	√	√	√	√	√	√	√	√	—*1	—*1	#9
FF0D 1420$_H$	FF0D 15FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1600$_H$	FF0D 161F$_H$	HBGERRSLV91	√	√	√	√	√	√	√	—	—	—	—	#9

Table 4.7 List of P-Bus area access (3/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FF0D 1620$_H$	FF0D 17FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1800$_H$	FF0D 181F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#9
FF0D 1820$_H$	FF0D 19FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF0D 1A00$_H$	FF0D 1A1F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#9
FF0D 1A20$_H$	FF1E FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#9
FF1F 0000$_H$	FF1F FFFF$_H$	ICUMHA	√	√	√	√	√	√	√	√	√	√	√	#9
FF20 0000$_H$	FF5F FFFF$_H$	Data Flash	√	√	√	√	√	√	√	√	√	√	√	#1
FF60 0000$_H$	FF6F 33D7$_H$	GTM0	√	√	√	√	√	√	√	√	√	√	√	#6H
FF6F 33D8$_H$	FF70 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2000$_H$	FF70 201F$_H$	E7GT00	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2020$_H$	FF70 21FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2200$_H$	FF70 221F$_H$	E7GT01	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2220$_H$	FF70 23FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2400$_H$	FF70 241F$_H$	E7GT10	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2420$_H$	FF70 25FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2600$_H$	FF70 261F$_H$	E7GT11	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2620$_H$	FF70 27FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2800$_H$	FF70 281F$_H$	E7GT20	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2820$_H$	FF70 29FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2A00$_H$	FF70 2A1F$_H$	E7GT21	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2A20$_H$	FF70 2BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2C00$_H$	FF70 2C1F$_H$	E7GT30	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2C20$_H$	FF70 2DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 2E00$_H$	FF70 2E1F$_H$	E7GT31	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 2E20$_H$	FF70 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF70 6000$_H$	FF70 6007$_H$	GTM0_1	√	√	√	√	√	√	√	√	√	√	√	#6H
FF70 6008$_H$	FF74 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 0000$_H$	FF75 007F$_H$	ECCCNT_A_V2A6	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 0080$_H$	FF75 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 0200$_H$	FF75 027F$_H$	ECCCNT_D_V2A6W	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 0280$_H$	FF75 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 0400$_H$	FF75 04FF$_H$	BECCCAP_V2A6	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 0500$_H$	FF75 08FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 0900$_H$	FF75 0903$_H$	PB6ECC	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 0904$_H$	FF75 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 1000$_H$	FF75 107F$_H$	PBG6H0	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 1080$_H$	FF75 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 2000$_H$	FF75 201F$_H$	PBGERRSLV6H0	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 2020$_H$	FF75 21FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF75 2200$_H$	FF75 221F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#6H
FF75 2220$_H$	FF7F FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6H
FF80 0000$_H$	FF97 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2H
FF98 0000$_H$	FF98 FFFF$_H$	Reset Controller, Power Supply Voltage Monitor, Clock Controller, Clock Monitor, Standby Controller	√	√	√	√	√	√	√	√	√	√	√	#2L
FF99 0000$_H$	FF99 6FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF99 7000$_H$	FF99 707F$_H$	RTCA0	√	√	√	√	√	√	√	√	√	√	√	#2L

Table 4.7 List of P-Bus area access (4/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FF99 7080$_H$	FF9A 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 2000$_H$	FF9A 2FFF$_H$	ADCJ2	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FF9A 3000$_H$	FF9A 37FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3800$_H$	FF9A 381F$_H$	ADCJ2_SELB	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FF9A 3820$_H$	FF9A 39FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3A00$_H$	FF9A 3A0F$_H$	FENC	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 3A10$_H$	FF9A 3AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3B00$_H$	FF9A 3B0F$_H$	FEINC_PE0	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 3B10$_H$	FF9A 3BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3C00$_H$	FF9A 3C0F$_H$	FEINC_PE1	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 3C10$_H$	FF9A 3CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3D00$_H$	FF9A 3D0F$_H$	FEINC_PE2	√	—	√	√	√	—	—	—	—	—	—	#2L
FF9A 3D10$_H$	FF9A 3DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 3E00$_H$	FF9A 3E0F$_H$	FEINC_PE3	√	—	√	√	√	—	—	—	—	—	—	#2L
FF9A 3E10$_H$	FF9A 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 4000$_H$	FF9A 407F$_H$	LPS0	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FF9A 4080$_H$	FF9A 41FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 4200$_H$	FF9A 421F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 4220$_H$	FF9A 43FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 4400$_H$	FF9A 441F$_H$	BIST	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 4420$_H$	FF9A 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 5000$_H$	FF9A 501F$_H$	WDTBA	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 5020$_H$	FF9A 51FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 5200$_H$	FF9A 520F$_H$	WDTBAOPBMON	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 5210$_H$	FF9A 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FF9A 6000$_H$	FF9A 60FF$_H$	BIST	√	√	√	√	√	√	√	√	√	√	√	#2L
FF9A 6100$_H$	FF9F FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFA0 0000$_H$	FFA0 001F$_H$	FLMD	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA0 0020$_H$	FFA0 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFA0 8000$_H$	FFA0 81FF$_H$	IDCTRL	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA0 8200$_H$	FFA0 83FF$_H$	IDCTRL	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA0 8400$_H$	FFA0 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFA1 0000$_H$	FFA1 1FFF$_H$	FACI0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA1 2000$_H$	FFA1 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFA1 4000$_H$	FFA1 5FFF$_H$	FACI1	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA1 6000$_H$	FFA1 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFA1 8000$_H$	FFA1 9FFF$_H$	FACI2	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA1 A000$_H$	FFA1 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFA2 0000$_H$	FFA2 FFFF$_H$	FACI0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA3 0000$_H$	FFA3 FFFF$_H$	FACI1	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA4 0000$_H$	FFA4 FFFF$_H$	FACI2	√	√	√	√	√	√	√	√	√	√	√	#6L
FFA5 0000$_H$	FFBE FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFBF 0000$_H$	FFBF 003F$_H$	OSTM0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0040$_H$	FFBF 00FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0100$_H$	FFBF 013F$_H$	OSTM1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0140$_H$	FFBF 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0200$_H$	FFBF 023F$_H$	OSTM2	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0240$_H$	FFBF 02FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0300$_H$	FFBF 033F$_H$	OSTM3	√	√	√	√	√	√	√	√	√	√	√	#5

Table 4.7 List of P-Bus area access (5/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFBF 0340$_H$	FFBF 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0400$_H$	FFBF 043F$_H$	OSTM4	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0440$_H$	FFBF 04FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0500$_H$	FFBF 053F$_H$	OSTM5	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0540$_H$	FFBF 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0600$_H$	FFBF 063F$_H$	OSTM6	√	—	√	√	√	—	—	—	—	—	—	#5
FFBF 0640$_H$	FFBF 06FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0700$_H$	FFBF 073F$_H$	OSTM7	√	—	√	√	√	—	—	—	—	—	—	#5
FFBF 0740$_H$	FFBF 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0800$_H$	FFBF 083F$_H$	OSTM8	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0840$_H$	FFBF 08BF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 08C0$_H$	FFBF 08C3$_H$	IC0CKSEL8	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 08C4$_H$	FFBF 08FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 0900$_H$	FFBF 093F$_H$	OSTM9	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 0940$_H$	FFBF 09BF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 09C0$_H$	FFBF 09C3$_H$	IC0CKSEL9	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 09C4$_H$	FFBF 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 1000$_H$	FFBF 101F$_H$	WDTB0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 1020$_H$	FFBF 10FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 1100$_H$	FFBF 111F$_H$	WDTB1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 1120$_H$	FFBF 11FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 1200$_H$	FFBF 121F$_H$	WDTB2	√	—	√	√	√	—	—	—	—	—	—	#5
FFBF 1220$_H$	FFBF 12FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 1300$_H$	FFBF 131F$_H$	WDTB3	√	—	√	√	√	—	—	—	—	—	—	#5
FFBF 1320$_H$	FFBF 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 2000$_H$	FFBF 201F$_H$	SWDTA0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 2020$_H$	FFBF 21FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 2200$_H$	FFBF 2203$_H$	SWDTA_SELB0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 2204$_H$	FFBF 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 3000$_H$	FFBF 307F$_H$	ENCA0	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#5
FFBF 3080$_H$	FFBF 30FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 3100$_H$	FFBF 317F$_H$	ENCA1	√	√	√	√	√	√	√	√	√	—*1	√	#5
FFBF 3180$_H$	FFBF 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 4000$_H$	FFBF 43FF$_H$	TAUD0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 4400$_H$	FFBF 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 5000$_H$	FFBF 53FF$_H$	TAUD1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 5400$_H$	FFBF 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 6000$_H$	FFBF 63FF$_H$	TAUD2	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 6400$_H$	FFBF 67FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 6800$_H$	FFBF 6803$_H$	SELB_TAUD2I	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 6804$_H$	FFBF 68FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 6900$_H$	FFBF 693F$_H$	DNFA (TAUD2)	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 6940$_H$	FFBF 6FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 7000$_H$	FFBF 70FF$_H$	TAUJ0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 7100$_H$	FFBF 71FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 7200$_H$	FFBF 72FF$_H$	TAUJ1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 7300$_H$	FFBF 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 8000$_H$	FFBF 83FF$_H$	TSG30	√	√	√	√	√	√	√	√	√	—*1	—*1	#5
FFBF 8400$_H$	FFBF 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5

Table 4.7 List of P-Bus area access (6/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFBF 8800$_H$	FFBF 8BFF$_H$	TSG31	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 8C00$_H$	FFBF 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 9000$_H$	FFBF 903F$_H$	TAPA0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 9040$_H$	FFBF 90FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 9100$_H$	FFBF 913F$_H$	TAPA1	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#5
FFBF 9140$_H$	FFBF 91FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 9200$_H$	FFBF 923F$_H$	TAPA2	√	√	√	√	√	√	√	√	√	—*1	—*1	#5
FFBF 9240$_H$	FFBF 92FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF 9300$_H$	FFBF 933F$_H$	TAPA3	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF 9340$_H$	FFBF 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF A000$_H$	FFBF A3FF$_H$	TPBA0	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF A400$_H$	FFBF A7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF A800$_H$	FFBF ABFF$_H$	TPBA1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF AC00$_H$	FFBF AEFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF AF00$_H$	FFBF AFFF$_H$	PIC1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF B000$_H$	FFBF B1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF B200$_H$	FFBF B203$_H$	PIC1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF B204$_H$	FFBF B3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF B400$_H$	FFBF B403$_H$	PIC1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF B404$_H$	FFBF B5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF B600$_H$	FFBF B603$_H$	PIC1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF B604$_H$	FFBF BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF C000$_H$	FFBF C7FF$_H$	PIC20	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF C800$_H$	FFBF C9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF CA00$_H$	FFBF CA03$_H$	PIC20	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF CA04$_H$	FFBF CFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF D000$_H$	FFBF D7FF$_H$	PIC21	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF D800$_H$	FFBF DFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF E000$_H$	FFBF E0FF$_H$	PIC22	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF E100$_H$	FFBF EFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF F000$_H$	FFBF F0FF$_H$	AURORA	√	√	—	—	—	—	—	—	—	—	—	#5
FFBF F100$_H$	FFBF F3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF F400$_H$	FFBF F4FF$_H$	KCRC1	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF F500$_H$	FFBF F6FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF F700$_H$	FFBF F7FF$_H$	KCRC3	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF F800$_H$	FFBF F9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF FA00$_H$	FFBF FAFF$_H$	KCRC5	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF FB00$_H$	FFBF FCFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFBF FD00$_H$	FFBF FDFF$_H$	KCRC7	√	√	√	√	√	√	√	√	√	√	√	#5
FFBF FE00$_H$	FFBF FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC0 0000$_H$	FFC0 000F$_H$	EINT	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC0 0010$_H$	FFC0 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC0 0200$_H$	FFC0 02FF$_H$	OTS0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC0 0300$_H$	FFC4 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC4 8000$_H$	FFC4 807F$_H$	ECCCNT_CFP_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 8080$_H$	FFC4 80FF$_H$	ECCCNT_CFP_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 8100$_H$	FFC4 817F$_H$	ECCCNT_CFP_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (7/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC4 8180$_H$	FFC4 81FF$_H$	ECCCNT_CFP_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 8200$_H$	FFC4 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 8800$_H$	FFC4 887F$_H$	ECCCNT_CFCCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 8880$_H$	FFC4 88FF$_H$	ECCCNT_CFCCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 8900$_H$	FFC4 89FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 8A00$_H$	FFC4 8A7F$_H$	ECCCNT_CFS	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 8A80$_H$	FFC4 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 9000$_H$	FFC4 907F$_H$	ECCCNT_IT_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9080$_H$	FFC4 90FF$_H$	ECCCNT_IT_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9100$_H$	FFC4 917F$_H$	ECCCNT_IT_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9180$_H$	FFC4 91FF$_H$	ECCCNT_IT_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9200$_H$	FFC4 93FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 9400$_H$	FFC4 947F$_H$	ECCCNT_ID_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9480$_H$	FFC4 94FF$_H$	ECCCNT_ID_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9500$_H$	FFC4 957F$_H$	ECCCNT_ID_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9580$_H$	FFC4 95FF$_H$	ECCCNT_ID_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9600$_H$	FFC4 97FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 9800$_H$	FFC4 987F$_H$	ECCCNT_LR_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9880$_H$	FFC4 98FF$_H$	ECCCNT_LR_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9900$_H$	FFC4 997F$_H$	ECCCNT_LR_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9980$_H$	FFC4 99FF$_H$	ECCCNT_LR_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9A00$_H$	FFC4 9BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 9C00$_H$	FFC4 9C7F$_H$	ECCCNT_CRCCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9C80$_H$	FFC4 9CFF$_H$	ECCCNT_CRCCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 9D00$_H$	FFC4 9D7F$_H$	ECCCNT_CRCCL2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC4 9D80$_H$	FFC4 9DFF$_H$	ECCCNT_CRCCL3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9E00$_H$	FFC4 9E7F$_H$	ECCCNT_CRA	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 9E80$_H$	FFC4 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 A000$_H$	FFC4 A07F$_H$	ECCCNT_DTS	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 A080$_H$	FFC4 A3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 A400$_H$	FFC4 A47F$_H$	ECCCNT_DMDE0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 A480$_H$	FFC4 A4FF$_H$	ECCCNT_DMDA0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 A500$_H$	FFC4 A57F$_H$	ECCCNT_DMDE1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 A580$_H$	FFC4 A5FF$_H$	ECCCNT_DMDA1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 A600$_H$	FFC4 AFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 B000$_H$	FFC4 B07F$_H$	ECCCNT_A_CCIB0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 B080$_H$	FFC4 B0FF$_H$	ECCCNT_A_CCIB1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 B100$_H$	FFC4 B1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 B200$_H$	FFC4 B27F$_H$	ECCCNT_A_CCIB0CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 B280$_H$	FFC4 B2FF$_H$	ECCCNT_A_CCIB1CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 B300$_H$	FFC4 BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (8/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC4 C000$_H$	FFC4 C07F$_H$	ECCCNT_SA_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C080$_H$	FFC4 C0FF$_H$	ECCCNT_A_GCFU0ICL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C100$_H$	FFC4 C17F$_H$	ECCCNT_A_GCFU0DCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C180$_H$	FFC4 C1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 C200$_H$	FFC4 C27F$_H$	ECCCNT_SA_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C280$_H$	FFC4 C2FF$_H$	ECCCNT_A_GCFU1ICL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C300$_H$	FFC4 C37F$_H$	ECCCNT_A_GCFU1DCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 C380$_H$	FFC4 C3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 C400$_H$	FFC4 C47F$_H$	ECCCNT_SA_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C480$_H$	FFC4 C4FF$_H$	ECCCNT_A_GCFU2ICL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C500$_H$	FFC4 C57F$_H$	ECCCNT_A_GCFU2DCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C580$_H$	FFC4 C5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 C600$_H$	FFC4 C67F$_H$	ECCCNT_SA_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C680$_H$	FFC4 C6FF$_H$	ECCCNT_A_GCFU3ICL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C700$_H$	FFC4 C77F$_H$	ECCCNT_A_GCFU3DCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 C780$_H$	FFC4 CFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 D000$_H$	FFC4 D07F$_H$	ECCCNT_A_X2VCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D080$_H$	FFC4 D0FF$_H$	ECCCNT_A_X2VCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 D100$_H$	FFC4 D1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 D200$_H$	FFC4 D27F$_H$	ECCCNT_A_BARR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D280$_H$	FFC4 D2FF$_H$	ECCCNT_A_IPIR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D300$_H$	FFC4 D37F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 D380$_H$	FFC4 D3FF$_H$	ECCCNT_A_TPTM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D400$_H$	FFC4 D47F$_H$	ECCCNT_A_CRAM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D480$_H$	FFC4 D4FF$_H$	ECCCNT_A_SG0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D500$_H$	FFC4 D57F$_H$	ECCCNT_A_SX2PV	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D580$_H$	FFC4 D5FF$_H$	ECCCNT_A_SX2FX	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D600$_H$	FFC4 D67F$_H$	ECCCNT_A_FX2SX	√	√	√	√	√	√	√	—	—	—	—	#0
FFC4 D680$_H$	FFC4 D6FF$_H$	ECCCNT_A_GCFUF	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D700$_H$	FFC4 D77F$_H$	ECCCNT_A_SX2MB	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 D780$_H$	FFC4 DFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 E000$_H$	FFC4 E07F$_H$	ECCCNT_D_V2XWCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E080$_H$	FFC4 E0FF$_H$	ECCCNT_D_V2XRCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E100$_H$	FFC4 E17F$_H$	ECCCNT_D_V2XWCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 E180$_H$	FFC4 E1FF$_H$	ECCCNT_D_V2XRCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 E200$_H$	FFC4 E3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (9/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC4 E400$_H$	FFC4 E47F$_H$	ECCCNT_D_PV2APBW	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E480$_H$	FFC4 E4FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 E500$_H$	FFC4 E57F$_H$	ECCCNT_D_CRAM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E580$_H$	FFC4 E5FF$_H$	ECCCNT_D_EMU	√	√	√	√	√	√	√	—	—	—	—	#0
FFC4 E600$_H$	FFC4 E67F$_H$	ECCCNT_D_DMDE0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E680$_H$	FFC4 E6FF$_H$	ECCCNT_D_DMDE1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 E700$_H$	FFC4 EFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 F000$_H$	FFC4 F07F$_H$	ECCCNT_SD_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F080$_H$	FFC4 F0FF$_H$	ECCCNT_MD_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F100$_H$	FFC4 F17F$_H$	ECCCNT_SD_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F180$_H$	FFC4 F1FF$_H$	ECCCNT_MD_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F200$_H$	FFC4 F27F$_H$	ECCCNT_SD_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 F280$_H$	FFC4 F2FF$_H$	ECCCNT_MD_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 F300$_H$	FFC4 F37F$_H$	ECCCNT_SD_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 F380$_H$	FFC4 F3FF$_H$	ECCCNT_MD_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC4 F400$_H$	FFC4 F7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 F800$_H$	FFC4 F87F$_H$	ECCCNT_D_BARR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F880$_H$	FFC4 F8FF$_H$	ECCCNT_D_IPIR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 F900$_H$	FFC4 F97F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 F980$_H$	FFC4 F9FF$_H$	ECCCNT_D_TPTM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 FA00$_H$	FFC4 FA7F$_H$	ECCCNT_D_DTS	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 FA80$_H$	FFC4 FAFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC4 FB00$_H$	FFC4 FB7F$_H$	ECCCNT_D_DMA0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 FB80$_H$	FFC4 FBFF$_H$	ECCCNT_D_DMA1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC4 FC00$_H$	FFC4 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 0000$_H$	FFC5 00FF$_H$	MECCCAP_LR_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 0100$_H$	FFC5 01FF$_H$	MECCCAP_LR_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 0200$_H$	FFC5 02FF$_H$	MECCCAP_LR_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC5 0300$_H$	FFC5 03FF$_H$	MECCCAP_LR_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC5 0400$_H$	FFC5 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 0800$_H$	FFC5 08FF$_H$	MECCCAP_LRA	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 0900$_H$	FFC5 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 1000$_H$	FFC5 10FF$_H$	MECCCAP_IT_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 1100$_H$	FFC5 11FF$_H$	MECCCAP_IT_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 1200$_H$	FFC5 12FF$_H$	MECCCAP_IT_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (10/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC5 1300$_H$	FFC5 13FF$_H$	MECCCAP_IT_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC5 1400$_H$	FFC5 17FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 1800$_H$	FFC5 18FF$_H$	MECCCAP_ID_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 1900$_H$	FFC5 19FF$_H$	MECCCAP_ID_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 1A00$_H$	FFC5 1AFF$_H$	MECCCAP_ID_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC5 1B00$_H$	FFC5 1BFF$_H$	MECCCAP_ID_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC5 1C00$_H$	FFC5 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 2000$_H$	FFC5 20FF$_H$	MECCCAP_CRAM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2100$_H$	FFC5 23FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 2400$_H$	FFC5 24FF$_H$	MECCCAP_CFL	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2500$_H$	FFC5 25FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 2600$_H$	FFC5 26FF$_H$	MECCCAP_DTS	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2700$_H$	FFC5 27FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 2800$_H$	FFC5 28FF$_H$	MECCCAP_DMDE0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2900$_H$	FFC5 29FF$_H$	MECCCAP_DMDA0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2A00$_H$	FFC5 2AFF$_H$	MECCCAP_DMDE1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2B00$_H$	FFC5 2BFF$_H$	MECCCAP_DMDA1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 2C00$_H$	FFC5 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 4000$_H$	FFC5 40FF$_H$	BECCCAP_LRAM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4100$_H$	FFC5 41FF$_H$	BECCCAP_CRAM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4200$_H$	FFC5 42FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 4300$_H$	FFC5 43FF$_H$	BECCCAP_CFL	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4400$_H$	FFC5 44FF$_H$	BECCCAP_PERI	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4500$_H$	FFC5 45FF$_H$	BECCCAP_DMDT	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4600$_H$	FFC5 47FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 4800$_H$	FFC5 48FF$_H$	BECCCAP_EMU	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 4900$_H$	FFC5 81FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 8200$_H$	FFC5 820F$_H$	DTS_COMPC	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 8210$_H$	FFC5 83FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC5 8400$_H$	FFC5 843F$_H$	ERRGEN	√	√	√	√	√	√	√	√	√	√	√	#0
FFC5 8440$_H$	FFC5 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 0000$_H$	FFC6 001F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 0020$_H$	FFC6 2BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 2C00$_H$	FFC6 2C3F$_H$	ECCDF	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 2C40$_H$	FFC6 2F5F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 2F60$_H$	FFC6 2F63$_H$	PB1ECC	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 2F64$_H$	FFC6 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3000$_H$	FFC6 307F$_H$	PBG10	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3080$_H$	FFC6 30FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3100$_H$	FFC6 311F$_H$	PBGERRSLV10	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3120$_H$	FFC6 37FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3800$_H$	FFC6 387F$_H$	ECCCNT_A_V2A1	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3880$_H$	FFC6 38FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3900$_H$	FFC6 397F$_H$	ECCCNT_D_V2A1W	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3980$_H$	FFC6 39FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1

Table 4.7 List of P-Bus area access (11/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC6 3A00$_H$	FFC6 3AFF$_H$	BECCCAP_V2A1	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3B00$_H$	FFC6 3BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3C00$_H$	FFC6 3C1F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3C20$_H$	FFC6 3CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3D00$_H$	FFC6 3D3F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3D40$_H$	FFC6 3DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 3E00$_H$	FFC6 3E1F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#1
FFC6 3E20$_H$	FFC6 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#1
FFC6 4000$_H$	FFC6 7FFF$_H$	GUARD_INTC2	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 8000$_H$	FFC6 83FF$_H$	GUARD_DTS	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 8400$_H$	FFC6 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 9000$_H$	FFC6 93FF$_H$	GUARD_DMAC0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 9400$_H$	FFC6 97FF$_H$	GUARD_DMAC1	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 9800$_H$	FFC6 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 A000$_H$	FFC6 A1FF$_H$	GUARD_BARR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 A200$_H$	FFC6 A3FF$_H$	GUARD_IPIR	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 A400$_H$	FFC6 A5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 A600$_H$	FFC6 A7FF$_H$	GUARD_TPTM	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 A800$_H$	FFC6 AFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 B000$_H$	FFC6 B01F$_H$	PBGERRSLV00	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 B020$_H$	FFC6 B07F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 B080$_H$	FFC6 B0FF$_H$	PBG00	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 B100$_H$	FFC6 B17F$_H$	PBG01	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 B180$_H$	FFC6 BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 C000$_H$	FFC6 C0FF$_H$	GUARD_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C100$_H$	FFC6 C1FF$_H$	GUARD_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C200$_H$	FFC6 C2FF$_H$	GUARD_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C300$_H$	FFC6 C3FF$_H$	GUARD_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C400$_H$	FFC6 C7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 C800$_H$	FFC6 C81F$_H$	PEGCAP_M_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C820$_H$	FFC6 C83F$_H$	PEGCAP_S_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C840$_H$	FFC6 C85F$_H$	PEGCAP_M_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C860$_H$	FFC6 C87F$_H$	PEGCAP_S_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 C880$_H$	FFC6 C89F$_H$	PEGCAP_M_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C8A0$_H$	FFC6 C8BF$_H$	PEGCAP_S_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C8C0$_H$	FFC6 C8DF$_H$	PEGCAP_M_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C8E0$_H$	FFC6 C8FF$_H$	PEGCAP_S_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 C900$_H$	FFC6 CFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 D000$_H$	FFC6 D103$_H$	GUARD_CRAMCRG0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 D104$_H$	FFC6 D1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 D200$_H$	FFC6 D303$_H$	GUARD_CRAMCRG1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 D304$_H$	FFC6 D3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 D400$_H$	FFC6 D503$_H$	GUARD_CRAMCRG2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 D504$_H$	FFC6 D5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 D600$_H$	FFC6 D703$_H$	GUARD_CRAMCRG3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 D704$_H$	FFC6 D7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 D800$_H$	FFC6 D81F$_H$	CRGCAP_PE0CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 D820$_H$	FFC6 D83F$_H$	CRGCAP_PE1CL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 D840$_H$	FFC6 D85F$_H$	CRGCAP_PE2CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (12/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC6 D860$_H$	FFC6 D87F$_H$	CRGCAP_PE3CL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 D880$_H$	FFC6 D9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 DA00$_H$	FFC6 DA1F$_H$	CRGCAP_CRAMHCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DA20$_H$	FFC6 DA3F$_H$	CRGCAP_CRAMLCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DA40$_H$	FFC6 DA5F$_H$	CRGCAP_SX2MBHCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DA60$_H$	FFC6 DA7F$_H$	CRGCAP_SX2MBLCL0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DA80$_H$	FFC6 DA9F$_H$	CRGCAP_CRAMHCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 DAA0$_H$	FFC6 DABF$_H$	CRGCAP_CRAMLCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 DAC0$_H$	FFC6 DADF$_H$	CRGCAP_SX2MBHCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 DAE0$_H$	FFC6 DAFF$_H$	CRGCAP_SX2MBLCL1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 DB00$_H$	FFC6 DB1F$_H$	CRGCAP_CRAMHCL2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 DB20$_H$	FFC6 DB3F$_H$	CRGCAP_CRAMLCL2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 DB40$_H$	FFC6 DB5F$_H$	CRGCAP_SX2MBHCL2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 DB60$_H$	FFC6 DB7F$_H$	CRGCAP_SX2MBLCL2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 DB80$_H$	FFC6 DB9F$_H$	CRGCAP_CRAMHCL3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DBA0$_H$	FFC6 DBBF$_H$	CRGCAP_CRAMLCL3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DBC0$_H$	FFC6 DBDF$_H$	CRGCAP_SX2MBHCL3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DBE0$_H$	FFC6 DBFF$_H$	CRGCAP_SX2MBLCL3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 DC00$_H$	FFC6 DFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 E000$_H$	FFC6 E06F$_H$	GUARD_CRAMCSG0	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 E070$_H$	FFC6 E1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 E200$_H$	FFC6 E26F$_H$	GUARD_CRAMCSG1	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 E270$_H$	FFC6 E3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 E400$_H$	FFC6 E46F$_H$	GUARD_CRAMCSG2	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 E470$_H$	FFC6 E5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 E600$_H$	FFC6 E66F$_H$	GUARD_CRAMCSG3	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 E670$_H$	FFC6 E7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 E800$_H$	FFC6 E8EB$_H$	GUARD_CRAMCRG01	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 E8EC$_H$	FFC6 E9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 EA00$_H$	FFC6 EAEB$_H$	GUARD_CRAMCRG11	√	—*1	√	√	√	—	—	—	—	—	—	#0
FFC6 EAEC$_H$	FFC6 EBFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 EC00$_H$	FFC6 ECEB$_H$	GUARD_CRAMCRG21	√	√	√	√	√	√	√	—	—	—	—	#0
FFC6 ECEC$_H$	FFC6 EDFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0

Table 4.7 List of P-Bus area access (13/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC6 EE00$_H$	FFC6 EEEB$_H$	GUARD_CRAMCRG31	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 EEEC$_H$	FFC6 EFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 F000$_H$	FFC6 F03F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 F040$_H$	FFC6 F7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 F800$_H$	FFC6 F8FF$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 F900$_H$	FFC6 F9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC6 FA00$_H$	FFC6 FA1F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#0
FFC6 FA20$_H$	FFC6 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFC7 0000$_H$	FFC7 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2000$_H$	FFC7 207F$_H$	ECCCNT_A_V2A3	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2080$_H$	FFC7 20FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2100$_H$	FFC7 217F$_H$	ECCCNT_D_V2A3W	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2180$_H$	FFC7 21FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2200$_H$	FFC7 22FF$_H$	BECCCAP_V2A3	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2300$_H$	FFC7 28FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2900$_H$	FFC7 2903$_H$	PB3ECC	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2904$_H$	FFC7 2AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2B00$_H$	FFC7 2B7F$_H$	PBG30	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2B80$_H$	FFC7 2BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2C00$_H$	FFC7 2C7F$_H$	PBG31	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2C80$_H$	FFC7 2CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2D00$_H$	FFC7 2D7F$_H$	PBG32	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2D80$_H$	FFC7 2DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 2E00$_H$	FFC7 2E7F$_H$	PBG33	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 2E80$_H$	FFC7 31FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 3200$_H$	FFC7 321F$_H$	PBGERRSLV30	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 3220$_H$	FFC7 33FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 3400$_H$	FFC7 341F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#3
FFC7 3420$_H$	FFC7 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFC7 4000$_H$	FFC7 407F$_H$	ECCCNT_A_V2A4	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 4080$_H$	FFC7 40FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 4100$_H$	FFC7 417F$_H$	ECCCNT_D_V2A4W	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 4180$_H$	FFC7 41FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 4200$_H$	FFC7 42FF$_H$	BECCCAP_V2A4	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 4300$_H$	FFC7 50FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5100$_H$	FFC7 5103$_H$	PB4ECC	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5104$_H$	FFC7 52FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5300$_H$	FFC7 537F$_H$	PBG40	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5380$_H$	FFC7 53FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5400$_H$	FFC7 541F$_H$	PBGERRSLV40	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5420$_H$	FFC7 55FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5600$_H$	FFC7 561F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5620$_H$	FFC7 56FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5700$_H$	FFC7 571F$_H$	E7MS00	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5720$_H$	FFC7 573F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5740$_H$	FFC7 574F$_H$	MSPI0INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5750$_H$	FFC7 576F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5770$_H$	FFC7 577F$_H$	MSPI0INTIF	√	√	√	√	√	√	√	√	√	√	√	#4

Table 4.7 List of P-Bus area access (14/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC7 5780$_H$	FFC7 579F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 57A0$_H$	FFC7 57AF$_H$	MSPI0INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 57B0$_H$	FFC7 57CF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 57D0$_H$	FFC7 57DF$_H$	MSPI0INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 57E0$_H$	FFC7 57FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5800$_H$	FFC7 581F$_H$	E7MS02	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5820$_H$	FFC7 583F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5840$_H$	FFC7 584F$_H$	MSPI2INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5850$_H$	FFC7 586F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5870$_H$	FFC7 587F$_H$	MSPI2INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5880$_H$	FFC7 589F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 58A0$_H$	FFC7 58AF$_H$	MSPI2INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 58B0$_H$	FFC7 58CF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 58D0$_H$	FFC7 58DF$_H$	MSPI2INTIF	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 58E0$_H$	FFC7 58FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5900$_H$	FFC7 591F$_H$	E7MS04	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFC7 5920$_H$	FFC7 593F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5940$_H$	FFC7 594F$_H$	MSPI4INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFC7 5950$_H$	FFC7 596F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5970$_H$	FFC7 597F$_H$	MSPI4INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFC7 5980$_H$	FFC7 599F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 59A0$_H$	FFC7 59AF$_H$	MSPI4INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFC7 59B0$_H$	FFC7 59CF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 59D0$_H$	FFC7 59DF$_H$	MSPI4INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFC7 59E0$_H$	FFC7 59FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5A00$_H$	FFC7 5A1F$_H$	E7MS06	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5A20$_H$	FFC7 5A3F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5A40$_H$	FFC7 5A4F$_H$	MSPI6INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5A50$_H$	FFC7 5A6F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5A70$_H$	FFC7 5A7F$_H$	MSPI6INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5A80$_H$	FFC7 5A9F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5AA0$_H$	FFC7 5AAF$_H$	MSPI6INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5AB0$_H$	FFC7 5ACF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5AD0$_H$	FFC7 5ADF$_H$	MSPI6INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5AE0$_H$	FFC7 5AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5B00$_H$	FFC7 5B1F$_H$	E7MS08	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5B20$_H$	FFC7 5B3F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5B40$_H$	FFC7 5B4F$_H$	MSPI8INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5B50$_H$	FFC7 5B6F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5B70$_H$	FFC7 5B7F$_H$	MSPI8INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5B80$_H$	FFC7 5B9F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5BA0$_H$	FFC7 5BAF$_H$	MSPI8INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5BB0$_H$	FFC7 5BCF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5BD0$_H$	FFC7 5BDF$_H$	MSPI8INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFC7 5BE0$_H$	FFC7 5CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 5D00$_H$	FFC7 5D3F$_H$	MSPITG	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 5D40$_H$	FFC7 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#4
FFC7 6000$_H$	FFC7 7FFF$_H$	MSPI0	√	√	√	√	√	√	√	√	√	√	√	#4
FFC7 8000$_H$	FFC7 80FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5

Table 4.7 List of P-Bus area access (15/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC7 8100$_H$	FFC7 817F$_H$	LTSC0	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 8180$_H$	FFC7 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 9000$_H$	FFC7 907F$_H$	ECCCNT_A_V2A5	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 9080$_H$	FFC7 90FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 9100$_H$	FFC7 917F$_H$	ECCCNT_D_V2A5W	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 9180$_H$	FFC7 91FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 9200$_H$	FFC7 92FF$_H$	BECCCAP_V2A5	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 9300$_H$	FFC7 A0FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 A100$_H$	FFC7 A103$_H$	PB5ECC	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 A104$_H$	FFC7 A2FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 A300$_H$	FFC7 A37F$_H$	PBG50	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 A380$_H$	FFC7 A3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 A400$_H$	FFC7 A47F$_H$	PBG51	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 A480$_H$	FFC7 A4FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 A500$_H$	FFC7 A57F$_H$	PBG52	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 A580$_H$	FFC7 A5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 A600$_H$	FFC7 A67F$_H$	PBG53	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 A680$_H$	FFC7 AFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 B000$_H$	FFC7 B01F$_H$	PBGERRSLV50	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 B020$_H$	FFC7 B3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 B400$_H$	FFC7 B41F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 B420$_H$	FFC7 BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C000$_H$	FFC7 C03F$_H$	RLIN31	√	√	√	√	√	√	√	√	√	—*1	—*1	#5
FFC7 C040$_H$	FFC7 C07F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C080$_H$	FFC7 C0BF$_H$	RLIN33	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C0C0$_H$	FFC7 C0FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C100$_H$	FFC7 C13F$_H$	RLIN35	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C140$_H$	FFC7 C17F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C180$_H$	FFC7 C1BF$_H$	RLIN37	√	√	√	√	√	√	√	√	√	—*1	√	#5
FFC7 C1C0$_H$	FFC7 C1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C200$_H$	FFC7 C23F$_H$	RLIN39	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C240$_H$	FFC7 C27F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C280$_H$	FFC7 C2BF$_H$	RLIN311	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C2C0$_H$	FFC7 C2FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C300$_H$	FFC7 C33F$_H$	RLIN313	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C340$_H$	FFC7 C37F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C380$_H$	FFC7 C3BF$_H$	RLIN315	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C3C0$_H$	FFC7 C3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C400$_H$	FFC7 C43F$_H$	RLIN317	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C440$_H$	FFC7 C47F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C480$_H$	FFC7 C4BF$_H$	RLIN319	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C4C0$_H$	FFC7 C4FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C500$_H$	FFC7 C53F$_H$	RLIN321	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C540$_H$	FFC7 C57F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C580$_H$	FFC7 C5BF$_H$	RLIN323	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 C5C0$_H$	FFC7 C7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C800$_H$	FFC7 C81F$_H$	E7MS01	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C820$_H$	FFC7 C83F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C840$_H$	FFC7 C84F$_H$	MSPI1INTIF	√	√	√	√	√	√	√	√	√	√	√	#5

Table 4.7 List of P-Bus area access (16/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC7 C850$_H$	FFC7 C86F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C870$_H$	FFC7 C87F$_H$	MSPI1INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C880$_H$	FFC7 C89F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C8A0$_H$	FFC7 C8AF$_H$	MSPI1INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C8B0$_H$	FFC7 C8CF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 C8D0$_H$	FFC7 C8DF$_H$	MSPI1INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 C8E0$_H$	FFC7 C9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CA00$_H$	FFC7 CA1F$_H$	E7MS03	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 CA20$_H$	FFC7 CA3F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CA40$_H$	FFC7 CA4F$_H$	MSPI3INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 CA50$_H$	FFC7 CA6F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CA70$_H$	FFC7 CA7F$_H$	MSPI3INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 CA80$_H$	FFC7 CA9F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CAA0$_H$	FFC7 CAAF$_H$	MSPI3INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 CAB0$_H$	FFC7 CACF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CAD0$_H$	FFC7 CADF$_H$	MSPI3INTIF	√	√	√	√	√	√	√	√	√	√	√	#5
FFC7 CAE0$_H$	FFC7 CBFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CC00$_H$	FFC7 CC1F$_H$	E7MS05	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFC7 CC20$_H$	FFC7 CC3F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CC40$_H$	FFC7 CC4F$_H$	MSPI5INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFC7 CC50$_H$	FFC7 CC6F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CC70$_H$	FFC7 CC7F$_H$	MSPI5INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFC7 CC80$_H$	FFC7 CC9F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CCA0$_H$	FFC7 CCAF$_H$	MSPI5INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFC7 CCB0$_H$	FFC7 CCCF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CCD0$_H$	FFC7 CCDF$_H$	MSPI5INTIF	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFC7 CCE0$_H$	FFC7 CDFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CE00$_H$	FFC7 CE1F$_H$	E7MS07	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 CE20$_H$	FFC7 CE3F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CE40$_H$	FFC7 CE4F$_H$	MSPI7INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 CE50$_H$	FFC7 CE6F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CE70$_H$	FFC7 CE7F$_H$	MSPI7INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 CE80$_H$	FFC7 CE9F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CEA0$_H$	FFC7 CEAF$_H$	MSPI7INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 CEB0$_H$	FFC7 CECF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 CED0$_H$	FFC7 CEDF$_H$	MSPI7INTIF	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFC7 CEE0$_H$	FFC7 CFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 D000$_H$	FFC7 D01F$_H$	E7MS09	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFC7 D020$_H$	FFC7 D03F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 D040$_H$	FFC7 D04F$_H$	MSPI9INTIF	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFC7 D050$_H$	FFC7 D06F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 D070$_H$	FFC7 D07F$_H$	MSPI9INTIF	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFC7 D080$_H$	FFC7 D09F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 D0A0$_H$	FFC7 D0AF$_H$	MSPI9INTIF	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFC7 D0B0$_H$	FFC7 D0CF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 D0D0$_H$	FFC7 D0DF$_H$	MSPI9INTIF	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFC7 D0E0$_H$	FFC7 DFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFC7 E000$_H$	FFC7 FFFF$_H$	MSPI1	√	√	√	√	√	√	√	√	√	√	√	#5
FFC8 0000$_H$	FFC8 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L

Table 4.7 List of P-Bus area access (17/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFC8 1000$_H$	FFC8 107F$_H$	PBG6L0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC8 1080$_H$	FFC8 11FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC8 1200$_H$	FFC8 127F$_H$	PBG6L1	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC8 1280$_H$	FFC8 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC8 3000$_H$	FFC8 301F$_H$	PBGERRSLV6L0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC8 3020$_H$	FFC8 303F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC8 3040$_H$	FFC8 305F$_H$	PBGERRSLV6L1	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC8 3060$_H$	FFC8 31FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFC8 3200$_H$	FFC8 321F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#6L
FFC8 3220$_H$	FFC9 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFCA 0000$_H$	FFCA 0FFF$_H$	ADCJ0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFCA 1000$_H$	FFCA 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFCA 2000$_H$	FFCA 20FF$_H$	AVSEG	√	√	√	√	√	√	√	√	√	√	√	#6L
FFCA 2100$_H$	FFCB FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFCC 0000$_H$	FFCC 1FFF$_H$	MSPI3	√	√	√	√	√	√	√	√	√	√	√	#5
FFCC 2000$_H$	FFCC 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFCC 4000$_H$	FFCC 5FFF$_H$	MSPI5	√	√	√	√	√	√	√	√	√	√	—*1	#5
FFCC 6000$_H$	FFCC 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFCC 8000$_H$	FFCC 9FFF$_H$	MSPI7	√	—	√	√	—*1	√	—*1	—	—	—	—	#5
FFCC A000$_H$	FFCC BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFCC C000$_H$	FFCC DFFF$_H$	MSPI9	√	—	√	—*1	—*1	—	—	—	—	—	—	#5
FFCC E000$_H$	FFCC FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#5
FFCD 0000$_H$	FFCD 01FF$_H$	SCDS	√	√	√	√	√	√	√	√	√	√	√	#6L
FFCD 0200$_H$	FFCD 03FF$_H$	SCDS	√	√	√	√	√	√	√	√	√	√	√	#6L
FFCD 0400$_H$	FFCD 05FF$_H$	SCDS	√	√	√	√	√	√	√	√	√	√	√	#6L
FFCD 0600$_H$	FFCF FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFD0 0000$_H$	FFD1 FFFF$_H$	RSCFD1	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 0000$_H$	FFD2 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 2000$_H$	FFD2 201F$_H$	E7RC11	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 2020$_H$	FFD2 203F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 2040$_H$	FFD2 205F$_H$	E7RC12	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 2060$_H$	FFD2 207F$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 2080$_H$	FFD2 209F$_H$	E7RC13	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 20A0$_H$	FFD2 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8000$_H$	FFD2 803F$_H$	RLIN30	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 8040$_H$	FFD2 81FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8200$_H$	FFD2 823F$_H$	RLIN32	√	√	√	√	√	√	√	√	√	—*1	√	#3
FFD2 8240$_H$	FFD2 83FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8400$_H$	FFD2 843F$_H$	RLIN34	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 8440$_H$	FFD2 85FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8600$_H$	FFD2 863F$_H$	RLIN36	√	√	√	√	√	√	√	√	√	—*1	—*1	#3
FFD2 8640$_H$	FFD2 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8800$_H$	FFD2 883F$_H$	RLIN38	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 8840$_H$	FFD2 89FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8A00$_H$	FFD2 8A3F$_H$	RLIN310	√	√	√	√	√	√	√	√	√	√	√	#3
FFD2 8A40$_H$	FFD2 8BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8C00$_H$	FFD2 8C3F$_H$	RLIN312	√	—	√	√	—*1	√	—*1	—	—	—	—	#3
FFD2 8C40$_H$	FFD2 8DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD2 8E00$_H$	FFD2 8E3F$_H$	RLIN314	√	—	√	√	—*1	√	—*1	—	—	—	—	#3

Table 4.7 List of P-Bus area access (18/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available												Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)		
FFD2 8E40$_H$	FFD2 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD2 9000$_H$	FFD2 903F$_H$	RLIN316	√	—	√	√	—*1	√	—*1	—	—	—	—		#3
FFD2 9040$_H$	FFD2 91FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD2 9200$_H$	FFD2 923F$_H$	RLIN318	√	—	√	√	—*1	√	—*1	—	—	—	—		#3
FFD2 9240$_H$	FFD2 93FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD2 9400$_H$	FFD2 943F$_H$	RLIN320	√	—	√	√	—*1	√	—*1	—	—	—	—		#3
FFD2 9440$_H$	FFD2 95FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD2 9600$_H$	FFD2 963F$_H$	RLIN322	√	—	√	√	—*1	√	—*1	—	—	—	—		#3
FFD2 9640$_H$	FFD3 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 1000$_H$	FFD3 101F$_H$	SCI30	√	√	√	√	√	√	√	—	—	—	—		#3
FFD3 1020$_H$	FFD3 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 2000$_H$	FFD3 201F$_H$	SCI31	√	√	√	√	√	√	√	—	—	—	—		#3
FFD3 2020$_H$	FFD3 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3000$_H$	FFD3 301F$_H$	SCI32	√	√	√	√	√	√	√	—	—	—	—		#3
FFD3 3020$_H$	FFD3 35FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3600$_H$	FFD3 367F$_H$	RSENT0	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3680$_H$	FFD3 36FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3700$_H$	FFD3 377F$_H$	RSENT1	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3780$_H$	FFD3 37FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3800$_H$	FFD3 387F$_H$	RSENT2	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3880$_H$	FFD3 38FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3900$_H$	FFD3 397F$_H$	RSENT3	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3980$_H$	FFD3 39FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3A00$_H$	FFD3 3A7F$_H$	RSENT4	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3A80$_H$	FFD3 3AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3B00$_H$	FFD3 3B7F$_H$	RSENT5	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3B80$_H$	FFD3 3BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3C00$_H$	FFD3 3C7F$_H$	RSENT6	√	√	√	√	√	√	√	√	√	√	—*1		#3
FFD3 3C80$_H$	FFD3 3CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3D00$_H$	FFD3 3D7F$_H$	RSENT7	√	√	√	√	√	√	√	√	√	√	—*1		#3
FFD3 3D80$_H$	FFD3 3DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 3E00$_H$	FFD3 3E03$_H$	RSENTTSSEL	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 3E04$_H$	FFD3 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 4000$_H$	FFD3 40FF$_H$	KCRC0	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 4100$_H$	FFD3 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 5000$_H$	FFD3 50FF$_H$	KCRC2	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 5100$_H$	FFD3 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 6000$_H$	FFD3 60FF$_H$	KCRC4	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 6100$_H$	FFD3 6FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 7000$_H$	FFD3 70FF$_H$	KCRC6	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 7100$_H$	FFD3 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 8000$_H$	FFD3 87FF$_H$	ECM	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 8800$_H$	FFD3 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 9000$_H$	FFD3 90FF$_H$	ECMM	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 9100$_H$	FFD3 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD3 A000$_H$	FFD3 A0FF$_H$	ECMC	√	√	√	√	√	√	√	√	√	√	√		#3
FFD3 A100$_H$	FFD3 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3
FFD4 0000$_H$	FFD4 07FF$_H$	CXP10	√	√	√	√	√	√	√	—	—	—	—		#3
FFD4 0800$_H$	FFD4 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—		#3

Table 4.7 List of P-Bus area access (19/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFD4 1000$_H$	FFD4 17FF$_H$	CXP11	√	√	√	√	√	√	√	—	—	—	—	#3
FFD4 1800$_H$	FFD4 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD4 2000$_H$	FFD4 27FF$_H$	CXP12	√	√	√	√	√	√	√	—	—	—	—	#3
FFD4 2800$_H$	FFD4 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD4 3000$_H$	FFD4 37FF$_H$	CXP13	√	√	√	√	√	√	√	—	—	—	—	#3
FFD4 3800$_H$	FFD4 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 0000$_H$	FFD5 07FF$_H$	PSI5S0	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#3
FFD5 0800$_H$	FFD5 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 1000$_H$	FFD5 17FF$_H$	PSI5S1	√	√	√	√	√	√	√	√	√	√	√	#3
FFD5 1800$_H$	FFD5 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 2000$_H$	FFD5 21FF$_H$	PSI50	√	√	√	√	√	√	√	√	√	√	√	#3
FFD5 2200$_H$	FFD5 23FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 2400$_H$	FFD5 25FF$_H$	PSI51	√	√	√	√	√	√	√	√	√	√	—*1	#3
FFD5 2600$_H$	FFD5 27FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 2800$_H$	FFD5 29FF$_H$	PSI52	√	√	√	√	√	√	√	√	√	√	√	#3
FFD5 2A00$_H$	FFD5 2BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 2C00$_H$	FFD5 2DFF$_H$	PSI53	√	√	√	√	√	√	√	√	√	√	√	#3
FFD5 2E00$_H$	FFD5 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 3000$_H$	FFD5 3003$_H$	PSI5_SELB	√	√	√	√	√	√	√	√	√	√	√	#3
FFD5 3004$_H$	FFD5 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 5000$_H$	FFD5 50FF$_H$	MMCA0	√	√	√	√	√	√	√	√	√	—*1	—*1	#3
FFD5 5100$_H$	FFD5 51FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 5200$_H$	FFD5 523F$_H$	E7MM00	√	√	√	√	√	√	√	√	√	—*1	—*1	#3
FFD5 5240$_H$	FFD5 52FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD5 5300$_H$	FFD5 533F$_H$	E7MM01	√	√	√	√	√	√	√	√	√	—*1	—*1	#3
FFD5 5340$_H$	FFD7 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD8 0000$_H$	FFD8 1FFF$_H$	MSPI2	√	√	√	√	√	√	√	√	√	√	√	#4
FFD8 2000$_H$	FFD8 3FFF$_H$	MSPI4	√	√	√	√	√	√	√	√	√	√	—*1	#4
FFD8 4000$_H$	FFD8 5FFF$_H$	MSPI6	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFD8 6000$_H$	FFD8 7FFF$_H$	MSPI8	√	—	√	√	—*1	√	—*1	—	—	—	—	#4
FFD8 8000$_H$	FFD8 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#3
FFD9 0000$_H$	FFD9 7FFF$_H$	PORT	√	√	√	√	√	√	√	√	√	√	√	#2L
FFD9 8000$_H$	FFD9 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDA 0000$_H$	FFDA 7FFF$_H$	PORT	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDA 8000$_H$	FFDD FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0000$_H$	FFDE 007F$_H$	ECCCNT_A_V2A2	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0080$_H$	FFDE 00FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0100$_H$	FFDE 017F$_H$	ECCCNT_D_V2A2W	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0180$_H$	FFDE 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0200$_H$	FFDE 02FF$_H$	BECCCAP_V2A2	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0300$_H$	FFDE 08FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0900$_H$	FFDE 0903$_H$	PB2ECC	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0904$_H$	FFDE 0AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0B00$_H$	FFDE 0B7F$_H$	PBG20	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0B80$_H$	FFDE 0BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0C00$_H$	FFDE 0C7F$_H$	PBG21	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0C80$_H$	FFDE 0CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 0D00$_H$	FFDE 0D7F$_H$	PBG22	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 0D80$_H$	FFDE 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L

Table 4.7 List of P-Bus area access (20/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFDE 1000$_H$	FFDE 101F$_H$	PBGERRSLV20	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 1020$_H$	FFDE 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFDE 2000$_H$	FFDE 201F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#2L
FFDE 2020$_H$	FFE7 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFE8 0000$_H$	FFE8 00FF$_H$	TAUJ2	√	√	√	√	√	√	√	√	√	√	√	#2H
FFE8 0100$_H$	FFE8 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2H
FFE8 0800$_H$	FFE8 0803$_H$	SELB_TAUJ2I	√	√	√	√	√	√	√	√	√	√	√	#2H
FFE8 0804$_H$	FFE8 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2H
FFE8 1000$_H$	FFE8 10FF$_H$	TAUJ3	√	√	√	√	√	√	√	√	√	√	√	#2H
FFE8 1100$_H$	FFE8 17FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2H
FFE8 1800$_H$	FFE8 1803$_H$	SELB_TAUJ3I	√	√	√	√	√	√	√	√	√	√	√	#2H
FFE8 1804$_H$	FFEA FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2H
FFEB 0000$_H$	FFEC FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0000$_H$	FFED 000F$_H$	DNFA (RLIN3)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0010$_H$	FFED 00FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0100$_H$	FFED 010F$_H$	DNFA (RSCAN)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0110$_H$	FFED 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0200$_H$	FFED 020F$_H$	DNFA (FLXA)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0210$_H$	FFED 02FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0300$_H$	FFED 030F$_H$	DNFA (RSENT)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0310$_H$	FFED 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0400$_H$	FFED 040F$_H$	DNFA (PSI5)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0410$_H$	FFED 04FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0500$_H$	FFED 050F$_H$	DNFA (TAPA)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0510$_H$	FFED 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0600$_H$	FFED 060F$_H$	DNFA (ADCJ0)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0610$_H$	FFED 06FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0700$_H$	FFED 070F$_H$	DNFA (ADCJ1)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0710$_H$	FFED 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0800$_H$	FFED 080F$_H$	DNFA (ADCJ2)	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FFED 0810$_H$	FFED 08FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0900$_H$	FFED 090F$_H$	DNFA (ENCA)	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FFED 0910$_H$	FFED 09FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0A00$_H$	FFED 0A0F$_H$	DNFA (TAUD0)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0A10$_H$	FFED 0AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0B00$_H$	FFED 0B0F$_H$	DNFA (TAUD1)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0B10$_H$	FFED 0BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0C00$_H$	FFED 0C0F$_H$	DNFA (TSG3)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0C10$_H$	FFED 0CFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0D00$_H$	FFED 0D0F$_H$	DNFA (ECM)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0D10$_H$	FFED 0DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0E00$_H$	FFED 0E0F$_H$	DNFA (TAUJ0)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0E10$_H$	FFED 0EFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 0F00$_H$	FFED 0F0F$_H$	DNFA (TAUJ1)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 0F10$_H$	FFED 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 1000$_H$	FFED 100F$_H$	DNFA (TAUJ2)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 1010$_H$	FFED 10FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 1100$_H$	FFED 110F$_H$	DNFA (TAUJ3)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 1110$_H$	FFED 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L

Table 4.7 List of P-Bus area access (21/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFED 4000$_H$	FFED 401F$_H$	FCLA (RLIN3)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4020$_H$	FFED 40FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4100$_H$	FFED 411F$_H$	FCLA (RSCAN)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4120$_H$	FFED 41FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4200$_H$	FFED 421F$_H$	FCLA (FLX)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4220$_H$	FFED 42FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4300$_H$	FFED 431F$_H$	FCLA (RSENT)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4320$_H$	FFED 43FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4400$_H$	FFED 441F$_H$	FCLA (PSI5)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4420$_H$	FFED 44FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4500$_H$	FFED 451F$_H$	FCLA (TAPA)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4520$_H$	FFED 45FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4600$_H$	FFED 461F$_H$	FCLA (NMI)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4620$_H$	FFED 46FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4700$_H$	FFED 471F$_H$	FCLA (INTP0)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4720$_H$	FFED 47FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4800$_H$	FFED 481F$_H$	FCLA (INTP1)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4820$_H$	FFED 48FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4900$_H$	FFED 491F$_H$	FCLA (INTP2)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4920$_H$	FFED 49FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4A00$_H$	FFED 4A1F$_H$	FCLA (INTP3)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4A20$_H$	FFED 4AFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4B00$_H$	FFED 4B1F$_H$	FCLA (INTP4)	√	√	√	√	√	√	√	√	√	√	√	#2L
FFED 4B20$_H$	FFED 4BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFED 4C00$_H$	FFED 4C1F$_H$	FCLA (ADCJ2)	√	√	√	√	√	√	√	√	√	—*1	√	#2L
FFED 4C20$_H$	FFED FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#2L
FFEE 0000$_H$	FFEF EFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFEF F000$_H$	FFEF F01F$_H$	PWBA0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFEF F020$_H$	FFEF FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0000$_H$	FFF0 003F$_H$	PWGC0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 0040$_H$	FFF0 01FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0200$_H$	FFF0 023F$_H$	PWGC1	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0240$_H$	FFF0 03FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0400$_H$	FFF0 043F$_H$	PWGC2	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 0440$_H$	FFF0 05FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0600$_H$	FFF0 063F$_H$	PWGC3	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0640$_H$	FFF0 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0800$_H$	FFF0 083F$_H$	PWGC4	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0840$_H$	FFF0 09FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0A00$_H$	FFF0 0A3F$_H$	PWGC5	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0A40$_H$	FFF0 0BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0C00$_H$	FFF0 0C3F$_H$	PWGC6	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0C40$_H$	FFF0 0DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 0E00$_H$	FFF0 0E3F$_H$	PWGC7	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 0E40$_H$	FFF0 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1000$_H$	FFF0 103F$_H$	PWGC8	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 1040$_H$	FFF0 11FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1200$_H$	FFF0 123F$_H$	PWGC9	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 1240$_H$	FFF0 13FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L

Table 4.7 List of P-Bus area access (22/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF0 1400$_H$	FFF0 143F$_H$	PWGC10	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 1440$_H$	FFF0 15FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1600$_H$	FFF0 163F$_H$	PWGC11	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 1640$_H$	FFF0 17FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1800$_H$	FFF0 183F$_H$	PWGC12	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 1840$_H$	FFF0 19FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1A00$_H$	FFF0 1A3F$_H$	PWGC13	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 1A40$_H$	FFF0 1BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1C00$_H$	FFF0 1C3F$_H$	PWGC14	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 1C40$_H$	FFF0 1DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 1E00$_H$	FFF0 1E3F$_H$	PWGC15	√	√	√	√	√	√	√	√	—*1	—*1	√	#6L
FFF0 1E40$_H$	FFF0 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2000$_H$	FFF0 203F$_H$	PWGC16	√	√	√	√	√	√	√	√	—*1	—*1	√	#6L
FFF0 2040$_H$	FFF0 21FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2200$_H$	FFF0 223F$_H$	PWGC17	√	√	√	√	√	√	√	√	—*1	—*1	√	#6L
FFF0 2240$_H$	FFF0 23FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2400$_H$	FFF0 243F$_H$	PWGC18	√	√	√	√	√	√	√	√	—*1	—*1	√	#6L
FFF0 2440$_H$	FFF0 25FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2600$_H$	FFF0 263F$_H$	PWGC19	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 2640$_H$	FFF0 27FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2800$_H$	FFF0 283F$_H$	PWGC20	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 2840$_H$	FFF0 29FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2A00$_H$	FFF0 2A3F$_H$	PWGC21	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 2A40$_H$	FFF0 2BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2C00$_H$	FFF0 2C3F$_H$	PWGC22	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 2C40$_H$	FFF0 2DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 2E00$_H$	FFF0 2E3F$_H$	PWGC23	√	√	√	√	√	√	√	√	—*1	—*1	√	#6L
FFF0 2E40$_H$	FFF0 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3000$_H$	FFF0 303F$_H$	PWGC24	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 3040$_H$	FFF0 31FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3200$_H$	FFF0 323F$_H$	PWGC25	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 3240$_H$	FFF0 33FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3400$_H$	FFF0 343F$_H$	PWGC26	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 3440$_H$	FFF0 35FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3600$_H$	FFF0 363F$_H$	PWGC27	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 3640$_H$	FFF0 37FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3800$_H$	FFF0 383F$_H$	PWGC28	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 3840$_H$	FFF0 39FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3A00$_H$	FFF0 3A3F$_H$	PWGC29	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 3A40$_H$	FFF0 3BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3C00$_H$	FFF0 3C3F$_H$	PWGC30	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 3C40$_H$	FFF0 3DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 3E00$_H$	FFF0 3E3F$_H$	PWGC31	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 3E40$_H$	FFF0 3FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4000$_H$	FFF0 403F$_H$	PWGC32	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 4040$_H$	FFF0 41FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4200$_H$	FFF0 423F$_H$	PWGC33	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 4240$_H$	FFF0 43FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4400$_H$	FFF0 443F$_H$	PWGC34	√	√	√	√	√	√	√	√	√	√	√	#6L

Table 4.7 List of P-Bus area access (23/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF0 4440$_H$	FFF0 45FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4600$_H$	FFF0 463F$_H$	PWGC35	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 4640$_H$	FFF0 47FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4800$_H$	FFF0 483F$_H$	PWGC36	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 4840$_H$	FFF0 49FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4A00$_H$	FFF0 4A3F$_H$	PWGC37	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 4A40$_H$	FFF0 4BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4C00$_H$	FFF0 4C3F$_H$	PWGC38	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 4C40$_H$	FFF0 4DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 4E00$_H$	FFF0 4E3F$_H$	PWGC39	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 4E40$_H$	FFF0 4FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5000$_H$	FFF0 503F$_H$	PWGC40	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 5040$_H$	FFF0 51FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5200$_H$	FFF0 523F$_H$	PWGC41	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 5240$_H$	FFF0 53FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5400$_H$	FFF0 543F$_H$	PWGC42	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 5440$_H$	FFF0 55FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5600$_H$	FFF0 563F$_H$	PWGC43	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 5640$_H$	FFF0 57FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5800$_H$	FFF0 583F$_H$	PWGC44	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 5840$_H$	FFF0 59FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5A00$_H$	FFF0 5A3F$_H$	PWGC45	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 5A40$_H$	FFF0 5BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5C00$_H$	FFF0 5C3F$_H$	PWGC46	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 5C40$_H$	FFF0 5DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 5E00$_H$	FFF0 5E3F$_H$	PWGC47	√	√	√	√	√	√	√	√	√	—*1	—*1	#6L
FFF0 5E40$_H$	FFF0 5FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6000$_H$	FFF0 603F$_H$	PWGC48	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 6040$_H$	FFF0 61FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6200$_H$	FFF0 623F$_H$	PWGC49	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 6240$_H$	FFF0 63FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6400$_H$	FFF0 643F$_H$	PWGC50	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 6440$_H$	FFF0 65FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6600$_H$	FFF0 663F$_H$	PWGC51	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 6640$_H$	FFF0 67FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6800$_H$	FFF0 683F$_H$	PWGC52	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 6840$_H$	FFF0 69FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6A00$_H$	FFF0 6A3F$_H$	PWGC53	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 6A40$_H$	FFF0 6BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6C00$_H$	FFF0 6C3F$_H$	PWGC54	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 6C40$_H$	FFF0 6DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 6E00$_H$	FFF0 6E3F$_H$	PWGC55	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 6E40$_H$	FFF0 6FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7000$_H$	FFF0 703F$_H$	PWGC56	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7040$_H$	FFF0 71FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7200$_H$	FFF0 723F$_H$	PWGC57	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7240$_H$	FFF0 73FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7400$_H$	FFF0 743F$_H$	PWGC58	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7440$_H$	FFF0 75FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L

Table 4.7 List of P-Bus area access (24/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF0 7600$_H$	FFF0 763F$_H$	PWGC59	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7640$_H$	FFF0 77FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7800$_H$	FFF0 783F$_H$	PWGC60	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7840$_H$	FFF0 79FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7A00$_H$	FFF0 7A3F$_H$	PWGC61	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7A40$_H$	FFF0 7BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7C00$_H$	FFF0 7C3F$_H$	PWGC62	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7C40$_H$	FFF0 7DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 7E00$_H$	FFF0 7E3F$_H$	PWGC63	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 7E40$_H$	FFF0 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8000$_H$	FFF0 803F$_H$	PWGC64	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 8040$_H$	FFF0 81FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8200$_H$	FFF0 823F$_H$	PWGC65	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 8240$_H$	FFF0 83FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8400$_H$	FFF0 843F$_H$	PWGC66	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 8440$_H$	FFF0 85FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8600$_H$	FFF0 863F$_H$	PWGC67	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 8640$_H$	FFF0 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8800$_H$	FFF0 883F$_H$	PWGC68	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 8840$_H$	FFF0 89FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8A00$_H$	FFF0 8A3F$_H$	PWGC69	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 8A40$_H$	FFF0 8BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8C00$_H$	FFF0 8C3F$_H$	PWGC70	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 8C40$_H$	FFF0 8DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 8E00$_H$	FFF0 8E3F$_H$	PWGC71	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 8E40$_H$	FFF0 8FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9000$_H$	FFF0 903F$_H$	PWGC72	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 9040$_H$	FFF0 91FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9200$_H$	FFF0 923F$_H$	PWGC73	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 9240$_H$	FFF0 93FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9400$_H$	FFF0 943F$_H$	PWGC74	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 9440$_H$	FFF0 95FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9600$_H$	FFF0 963F$_H$	PWGC75	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 9640$_H$	FFF0 97FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9800$_H$	FFF0 983F$_H$	PWGC76	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 9840$_H$	FFF0 99FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9A00$_H$	FFF0 9A3F$_H$	PWGC77	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 9A40$_H$	FFF0 9BFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9C00$_H$	FFF0 9C3F$_H$	PWGC78	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 9C40$_H$	FFF0 9DFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 9E00$_H$	FFF0 9E3F$_H$	PWGC79	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 9E40$_H$	FFF0 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 A000$_H$	FFF0 A03F$_H$	PWGC80	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 A040$_H$	FFF0 A1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 A200$_H$	FFF0 A23F$_H$	PWGC81	√	√	√	√	√	√	√	√	√	—*1	√	#6L
FFF0 A240$_H$	FFF0 A3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 A400$_H$	FFF0 A43F$_H$	PWGC82	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 A440$_H$	FFF0 A5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 A600$_H$	FFF0 A63F$_H$	PWGC83	√	√	√	√	√	√	√	√	√	√	√	#6L

Table 4.7 List of P-Bus area access (25/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF0 A640$_H$	FFF0 A7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 A800$_H$	FFF0 A83F$_H$	PWGC84	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 A840$_H$	FFF0 A9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 AA00$_H$	FFF0 AA3F$_H$	PWGC85	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 AA40$_H$	FFF0 ABFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 AC00$_H$	FFF0 AC3F$_H$	PWGC86	√	√	√	√	√	√	√	√	√	√	—*1	#6L
FFF0 AC40$_H$	FFF0 ADFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 AE00$_H$	FFF0 AE3F$_H$	PWGC87	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 AE40$_H$	FFF0 AFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 B000$_H$	FFF0 B03F$_H$	PWGC88	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 B040$_H$	FFF0 B1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 B200$_H$	FFF0 B23F$_H$	PWGC89	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 B240$_H$	FFF0 B3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 B400$_H$	FFF0 B43F$_H$	PWGC90	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 B440$_H$	FFF0 B5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 B600$_H$	FFF0 B63F$_H$	PWGC91	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 B640$_H$	FFF0 B7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 B800$_H$	FFF0 B83F$_H$	PWGC92	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 B840$_H$	FFF0 B9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 BA00$_H$	FFF0 BA3F$_H$	PWGC93	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 BA40$_H$	FFF0 BBFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 BC00$_H$	FFF0 BC3F$_H$	PWGC94	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 BC40$_H$	FFF0 BDFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 BE00$_H$	FFF0 BE3F$_H$	PWGC95	√	√	√	√	√	√	√	√	—*1	—*1	—*1	#6L
FFF0 BE40$_H$	FFF0 BFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 C000$_H$	FFF0 C03F$_H$	SLPWG	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 C040$_H$	FFF0 C1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 C200$_H$	FFF0 C20F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 C210$_H$	FFF0 C3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 C400$_H$	FFF0 C40F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 C410$_H$	FFF0 C5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 C600$_H$	FFF0 C60F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 C610$_H$	FFF0 C7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 C800$_H$	FFF0 C80F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 C810$_H$	FFF0 C9FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 CA00$_H$	FFF0 CA0F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 CA10$_H$	FFF0 CBFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 CC00$_H$	FFF0 CC0F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 CC10$_H$	FFF0 CDFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 CE00$_H$	FFF0 CE0F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 CE10$_H$	FFF0 CFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 D000$_H$	FFF0 D00F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 D010$_H$	FFF0 D1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 D200$_H$	FFF0 D20F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 D210$_H$	FFF0 D3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 D400$_H$	FFF0 D40F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 D410$_H$	FFF0 D5FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 D600$_H$	FFF0 D60F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 D610$_H$	FFF0 D7FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L

Table 4.7 List of P-Bus area access (26/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF0 D800$_H$	FFF0 D80F$_H$	PWGCINTF	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 D810$_H$	FFF0 DFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF0 E000$_H$	FFF0 E7FF$_H$	PWSD0	√	√	√	√	√	√	√	√	√	√	√	#6L
FFF0 E800$_H$	FFF1 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF2 0000$_H$	FFF2 0FFF$_H$	ADCJ1	√	√	√	√	√	√	√	√	√	√	√	#7
FFF2 1000$_H$	FFF2 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF2 2000$_H$	FFF2 207F$_H$	RIIC0	√	√	√	√	√	√	√	√	√	√	√	#7
FFF2 2080$_H$	FFF2 2FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF2 3000$_H$	FFF2 307F$_H$	RIIC1	√	√	√	√	√	√	√	√	√	√	—*1	#7
FFF2 3080$_H$	FFF2 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF2 8000$_H$	FFF2 807F$_H$	ECCCNT_A_V2A8	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 8080$_H$	FFF2 83FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 8400$_H$	FFF2 847F$_H$	ECCCNT_D_V2A8W	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 8480$_H$	FFF2 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 8800$_H$	FFF2 88FF$_H$	BECCCAP_V2A8	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 8900$_H$	FFF2 90FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 9100$_H$	FFF2 9103$_H$	PB8ECC	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 9104$_H$	FFF2 92FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 9300$_H$	FFF2 937F$_H$	PBG80	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 9380$_H$	FFF2 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 A000$_H$	FFF2 A01F$_H$	PBGERRSLV80	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 A020$_H$	FFF2 A1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 A200$_H$	FFF2 A21F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 A220$_H$	FFF2 AFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 B000$_H$	FFF2 B01F$_H$	E7RC01	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 B020$_H$	FFF2 B1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 B200$_H$	FFF2 B21F$_H$	E7RC02	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 B220$_H$	FFF2 B3FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF2 B400$_H$	FFF2 B41F$_H$	E7RC03	√	√	√	√	√	√	√	√	√	√	√	#8
FFF2 B420$_H$	FFF2 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#8
FFF3 0000$_H$	FFF4 7FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 8000$_H$	FFF4 807F$_H$	ECCCNT_A_V2A7	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 8080$_H$	FFF4 83FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 8400$_H$	FFF4 847F$_H$	ECCCNT_D_V2A7W	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 8480$_H$	FFF4 87FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 8800$_H$	FFF4 88FF$_H$	BECCCAP_V2A7	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 8900$_H$	FFF4 90FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 9100$_H$	FFF4 9103$_H$	PB7ECC	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 9104$_H$	FFF4 93FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 9400$_H$	FFF4 947F$_H$	PBG70	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 9480$_H$	FFF4 9FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 A000$_H$	FFF4 A01F$_H$	PBGERRSLV70	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 A020$_H$	FFF4 A1FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF4 A200$_H$	FFF4 A21F$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#7
FFF4 A220$_H$	FFF4 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#7
FFF5 0000$_H$	FFF6 FFFF$_H$	RSCFD0	√	√	√	√	√	√	√	√	√	√	√	#8
FFF7 0000$_H$	FFF7 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#6L
FFF8 0000$_H$	FFF8 7FFF$_H$	INTC2	√	√	√	√	√	√	√	√	√	√	√	#0
FFF8 8000$_H$	FFF8 BFFF$_H$	DTSCNT	√	√	√	√	√	√	√	√	√	√	√	#0

Table 4.7 List of P-Bus area access (27/27)

Start Address	End Address	Area	Availability of Area √: Available, —: Not available											Peripheral Group No.
			U2A-EVA (516 pins) U2A16 mode	U2A-EVA (516 pins) U2A8 mode	U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292 pins)	U2A6 (176 pins)	U2A6 (156 pins)	U2A6 (144 pins)	
FFF8 C000$_H$	FFF8 FFFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFF9 0000$_H$	FFF9 7FFF$_H$	SDMAC0	√	√	√	√	√	√	√	√	√	√	√	#0
FFF9 8000$_H$	FFF9 FFFF$_H$	SDMAC1	√	√	√	√	√	√	√	√	√	√	√	#0
FFFA 0000$_H$	FFFB 07FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 0800$_H$	FFFB 080F$_H$	CCIBH0	√	√	√	√	√	√	√	√	√	√	√	#0
FFFB 0810$_H$	FFFB 081F$_H$	System Reserved	√	√	√	√	√	—	—	—	—	—	—	#0
FFFB 0820$_H$	FFFB 0FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 1000$_H$	FFFB 10FF$_H$	CFU0	√	√	—*2	—*2	—*2	—*2	—*2	—*2	—*2	—*2	—*2	#0
FFFB 1100$_H$	FFFB 11FF$_H$	CFU1	√	√	—*2	—*2	—*2	—	—	—	—	—	—	#0
FFFB 1200$_H$	FFFB 13FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 1400$_H$	FFFB 17FF$_H$	GCFU	√	√	√	√	√	√	√	√	√	√	√	#0
FFFB 1800$_H$	FFFB 1FFF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 2000$_H$	FFFB 200F$_H$	BOOTCTRL	√	√	√	√	√	√	√	√	√	√	√	#0
FFFB 2010$_H$	FFFB 27FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 2800$_H$	FFFB 280F$_H$	ECCKCPROT	√	√	√	√	√	√	√	√	√	√	√	#0
FFFB 2810$_H$	FFFB 77FF$_H$	Unmapped area	—	—	—	—	—	—	—	—	—	—	—	#0
FFFB 7800$_H$	FFFB 7FFF$_H$	System Reserved	√	√	√	√	√	√	√	√	√	√	√	#0

Note 1. The access does not output an error response.

Note 2. PBG(P-Bus Guard) error will be occurred according to setting of PBG00#1 when access to this area.

4.3.6 Details of H-Bus area

Error response will be notified if accessed to reserved area.

Table 4.8 List of H-Bus area access

Start Address	End Address	Area	Availability of Area √: Available, —: Not available					
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6 (292 pins and 176 pins)	U2A6 (156 pins and 144 pins)
1000 0000_H	1000 $0FFF_H$	RHSIF0_L2	√	√	√	√	—	—
1000 1000_H	1001 $FFFF_H$	Reserved	—	—	—	—	—	—
1002 0000_H	1002 $0FFF_H$	FLXA0	√	√	√	√	√	√
1002 1000_H	1002 $1FFF_H$	FLXA1	√	√	√	√	—	—
1002 2000_H	1003 $FFFF_H$	Reserved	—	—	—	—	—	—
1004 0000_H	1004 $0FFF_H$	SFMA0	√	√	√	√	√	—*2
1004 1000_H	37FF $FFFF_H$	Reserved	—	—	—	—	—	—
3800 0000_H	3BFF $FFFF_H$	External Serial Flash Memory (SFMA)	√	√	√	√	√	—*2
3C00 0000_H	3FFF $FFFF_H$	Reserved	—	—	—	—	—	—

Note 1. Only RHSIF can access to H-Bus slave modules. For other H-Bus master, this area poses as reserved area. [For U2A-EVA, U2A16 and U2A8 Only]

Note 2. The access does not output an error response.

4.3.7 Details of I-Bus area

Error response will be notified if accessed to reserved area.

When masters except PEx access the self registers of BARR, IPIR and TPTM, error response will be notified.

Table 4.9 List of I-Bus area access

Start Address	End Address	Area	Availability of Area √: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
FFFB 8000_H	FFFB $8FFF_H$	BARR	√	√	√	√	√
FFFB 9000_H	FFFB $9FFF_H$	IPIR	√	√	√	√	√
FFFB $A000_H$	FFFB $AFFF_H$	Reserved	—	—	—	—	—
FFFB $B000_H$	FFFB $BFFF_H$	TPTM	√	√	√	√	√
FFFB $C000_H$	FFFB $FFFF_H$	Reserved	—	—	—	—	—

4.3.8 Debug area

Error response will be notified if accessed to F901 0000_H - F904 $FFFF_H$, FA00 0000_H - FA0F $FFFF_H$ by H-Bus.

Availability of error notification depends on PE_DISABLE setting in OPBT3 if accessed to F902 0000_H - F902 $FFFF_H$.

[For U2A-EVA(U2A16 mode)/U2A16]
Availability of error notification depends on PE_DISABLE setting in OPBT3 if accessed to F903 0000_H - F904 $FFFF_H$.

[For U2A-EVA(U2A16 mode)/U2A-EVA(U2A8 mode)]
Error response will be notified if accessed to FA10 0000_H - FB1F $FFFF_H$, FB40 0000_H - FB5F $FFFF_H$ by H-Bus.

[For U2A6]
Error response will be notified if accessed to FA10 0000_H - FB00 $1FFF_H$, FB02 0000_H - FB02 $1FFF_H$, FB04 0000_H - FB04 $1FFF_H$, FB06 0000_H - FB06 $1FFF_H$ by H-Bus.

Table 4.10 Error Notifications of Debug area (1/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
F900 0000_H	F900 $FFFF_H$	Reserved	ER	ER	ER	ER	ER
F901 0000_H	F901 $FFFF_H$	Debug (CPU0)	—	—	—	—	—
F902 0000_H	F902 $FFFF_H$	Debug (CPU1)	—	—	—	—	—
F903 0000_H	F903 $FFFF_H$	Debug (CPU2)	—	ER	—	ER	ER
F904 0000_H	F904 $FFFF_H$	Debug (CPU3)	—	ER	—	ER	ER
F905 0000_H	F9FF $FFFF_H$	Reserved	ER	ER	ER	ER	ER
FA00 0000_H	FA01 $7FFF_H$	Instrumentation RAM	—	—	—	—	ER
FA01 8000_H	FA0F $FFFF_H$	Reserved	—	—	—	—	ER
FA10 0000_H	FA10 $3FFF_H$	Trace filter RAM (Cluster RAM) (Cluster 0)	—	—	ER	ER	—
FA10 4000_H	FA10 $7FFF_H$	Reserved	—	—	ER	ER	—
FA10 8000_H	FA10 $BFFF_H$	Trace filter RAM (Cluster RAM) (Cluster 1)	—	—	ER	ER	—
FA10 $C000_H$	FA11 $FFFF_H$	Reserved	—	—	ER	ER	—
FA12 0000_H	FA12 $FFFF_H$	Trace filter RAM (Cluster RAM) (Cluster 2)	—	—	ER	ER	—
FA13 0000_H	FA13 $FFFF_H$	Reserved	—	—	ER	ER	—
FA14 0000_H	FA14 $1FFF_H$	Trace filter RAM (Cluster RAM) (Cluster 3)	—	—	ER	ER	—
FA14 2000_H	FA1A $FFFF_H$	Reserved	—	—	ER	ER	—
FA1B 0000_H	FA1B $07FF_H$	Trace filter RAM (Local RAM) (CPU3)	—	—	ER	ER	—
FA1B 0800_H	FA1B $FFFF_H$	Reserved	—	—	ER	ER	—
FA1C 0000_H	FA1C $07FF_H$	Trace filter RAM (Local RAM) (CPU2)	—	—	ER	ER	—
FA1C 0800_H	FA1C $FFFF_H$	Reserved	—	—	ER	ER	—
FA1D 0000_H	FA1D $07FF_H$	Trace filter RAM (Local RAM) (CPU1)	—	—	ER	ER	—
FA1D 0800_H	FA1D $FFFF_H$	Reserved	—	—	ER	ER	—
FA1E 0000_H	FA1E $07FF_H$	Trace filter RAM (Local RAM) (CPU0)	—	—	ER	ER	—
FA1E 0800_H	FAFF $FFFF_H$	Reserved	—	—	ER	ER	—

Table 4.10 Error Notifications of Debug area (2/2)

Start Address	End Address	Area	Error Response ER: Available, —: Not available				
			U2A-EVA (U2A16 mode)	U2A-EVA (U2A8 mode)	U2A16	U2A8	U2A6
FB00 0000$_H$	FB00 1FFF$_H$	Cluster0 ERAM	—	—	ER	ER	—
FB00 2000$_H$	FB01 FFFF$_H$		—	—	ER	ER	ER
FB02 0000$_H$	FB02 1FFF$_H$		—	—	ER	ER	—
FB02 2000$_H$	FB03 FFFF$_H$		—	—	ER	ER	ER
FB04 0000$_H$	FB04 1FFF$_H$		—	—	ER	ER	—
FB04 2000$_H$	FB05 FFFF$_H$		—	—	ER	ER	ER
FB06 0000$_H$	FB06 1FFF$_H$		—	—	ER	ER	—
FB06 2000$_H$	FB0F FFFF$_H$		—	—	ER	ER	ER
FB10 0000$_H$	FB1F FFFF$_H$	Cluster1 ERAM	—	—	ER	ER	ER
FB20 0000$_H$	FB3F FFFF$_H$	Reserved	ER	ER	ER	ER	ER
FB40 0000$_H$	FB5F FFFF$_H$	Global Emulation RAM	—	—	ER	ER	ER
FB60 0000$_H$	FBFF FFFF$_H$	Reserved	ER	ER	ER	ER	ER

4.3.9 Other area

Error response will be notified if accessed to $4000\ 0000_H$ - $F8FF\ FFFF_H$.

Section 5 Operating Modes

5.1 Features

This MCU has multiple operating modes, which can be selected with the five pins (FLMD0, FLMD1, FLMD2, MODE0 and $\overline{\text{TRST}}$) and the setting of STMSEL1, STMSEL0 in option byte. For details of STMSEL1, STMSEL0, see **Section 51, Flash Memory**. **Table 5.1** shows the list of the operating modes.

The values of FLMD0, FLMD1, FLMD2, MODE0 and $\overline{\text{TRST}}$ are latched at the rising edge of $\overline{\text{RESET}}$ to determine the operating mode, and part of the values of these pins should be kept during $\overline{\text{RESET}}$ = 1 except that the value of FLMD0 can be changed for selecting Flash Programming Interface in the Serial Programming Mode, for details of which pin should be kept refer to the **Table 5.1**. These pins are continually being monitored during operation to detect mode errors.

Table 5.1 Mode List

	Pins					OPBT					
	FLMD0	FLMD1	FLMD2	MODE0	$\overline{\text{TRST}}$	STMSEL1	STMSEL0	Operating Mode	Startup Area	Types of Debug Interface*1	Flash Programming Interface
User Mode	0	x	x	x	0	0	0	Normal Operating Mode	User Area*2	Nexus/LPD	—
	0	x	x	x	0	0	1	User Boot Mode 0	User Boot Area*3	Nexus/LPD	—
	0	x	x	x	0	1	x	Serial Programming Mode 0	Boot firmware	—	CSI
	0	x	x	x	1*5	x	0	Normal Operating Mode	User Area*2	Nexus/LPD	—
	0	x	x	x	1*5	x	1	User Boot Mode 0	User Boot Area*3	Nexus/LPD	—
	1	0	x	x	x	x	x	Serial Programming Mode 1	Boot firmware	—	2-wire UART/CSI*4
	1	1	0	x	x*5	x	x	User Boot Mode 1	User Boot Area*3	Nexus/LPD	—
	1	1	1	0	x	x	x	Boundary Scan Mode	—	BSCAN	—

Note 1. For the correspondence between the pin function and pin state in each debug interface. See **Section 2, Pin Functions**.

Note 2. Reset vector address can be set by option bytes. See **Section 51, Flash Memory**.

Note 3. The reset vector address of CPU0 is fixed to 0800 0000_H (head address in User Boot Area). The reset vector address of CPU 1 to 3 can be set by option bytes.

Note 4. For details of Flash Programming Interface is shown in **Section 51, Flash Memory**. CSI is the “Clocked Serial Interface”.

Note 5. CPU does not run after releasing the reset depending on option byte (CPUBTMSK_EN). BIST is not executed.

NOTE

To change operating mode shown below, restart from Power On Reset.

In the case of External Reset only, Power Control is not initialized and MCU does not restart properly.

- Serial Programming Mode 0 to User Boot Mode 1 and Boundary Scan Mode.
- Serial Programming Mode 1 to other than Serial Programming Mode 0/1.
- Other modes to Serial Programming Mode 0/1.

In addition to above mode change, restart from Power On Reset in the case of the following case as well.

- After first External Reset is released with $\overline{\text{TRST}}$ is 1 after power up, External Reset is asserted again with $\overline{\text{TRST}}$ is 0.

5.1.1 Normal Operation Mode

This mode is the basic mode for execution of application software. The On-chip debug functions can be used in this mode.

After reset release, instruction fetch is carried out from the user area. For reset vector of each CPU, see **Section 51, Flash Memory**.

The CPU will not autonomously run when $\overline{\text{TRST}}$ = 1. A debugger must be used to have the CPU run.

5.1.2 User Boot Mode 0/1

This mode is the same as Normal Operation Mode except the reset vector address of CPU0.

After reset release, instruction fetch of CPU0 is carried out from the user boot area.

Transition to chip standby modes is not supported in User Boot Mode 1.

5.1.3 Serial Programming Mode 0/1

A dedicated flash memory programmer can be used to handle the flash memory. After reset release, the MCU boots up from the On-chip boot program and starts connection in the specified protocol for flash programming. For details, see **Section 51, Flash Memory**.

CPUn (n! = 0) is disabled regardless of the option byte in this mode.

5.1.4 Boundary Scan Mode

This mode allows boundary scan tests compliant with IEEE Standard 1149.1.

CPU is disabled regardless of the option byte in this mode. For details, See **Section 53, Boundary Scan**.

5.2 Input Pins

Table 5.2 shows the pin information for mode setting. All mode terminals are taken in Mode Register. FLMD[2:1] and MODE0 can be used as GPIO.

Table 5.2 Pin information for mode setting

Pin Name	I/O	Function
FLMD0	Input	Operating mode select pin
FLMD1	Input	Operating mode select pin
FLMD2	Input	Operating mode select pin
MODE0	Input	Operating mode select pin
$\overline{\text{TRST}}$	Input	The shared of Nexus/JTAG Reset pin and operating mode select pin

5.3 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA / DTS requests.

The error notifications to ECM of this module are listed on the following table.

Table 5.3 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to Bus Master
Mode error during Normal Operation Mode/User Boot Mode	Unintended activation of Production Test Mode	24	—
Mode error during User Boot Mode	Unintended activation of Normal Operation Mode	25	—
Mode error during Normal Operation Mode	Unintended deactivation of Normal Operation Mode	26	—
Mode error during Normal Operation Mode/User Boot Mode	Unintended activation of Serial Programming Mode	27	—
Mode error during Normal Operation Mode	Unintended activation of User Boot Mode	28	—
Mode error during User Boot Mode	Unintended deactivation of User Boot Mode	29	—
Mode error during any mode	Mode check error	30	—
Unintended Debug Enable detection (PE0)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	227	—
Unintended Debug Enable detection (PE1)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	259	—
Unintended Debug Enable detection (PE2)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	291	—
Unintended Debug Enable detection (PE3)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	323	—

5.4 Register Description

5.4.1 List of Registers

The following table lists the operating modes registers.

Table 5.4 List of Registers

Register Name	Register Symbol	Address	Access Width	Value after reset	Access Protection	
					PBG	Other
Mode Register	MODE	FF98 4000_H	32	000X $00xx_H$	PBG20 #2	—

5.4.2 Reset of Registers

Register reset condition is shown in **Table 5.5**.

Table 5.5 Register Reset Condition

Register Symbol	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
MODE	—	√*1	—	—	—	—	—

Note 1. External Reset only

5.4.3 MODE — Mode Register

This register indicates the value of the mode pins and mode select option bytes latched when External Reset is released.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 4000$_H$

Value after reset: 000x 00xx$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STMSEL1	STMSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	MODE0	FLMD2	FLMD1	FLMD0	TRST
Value after reset	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.6 MODE Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	STMSEL1	STMSEL1 This bit indicates the value of the Option byte STMSEL1 latched 0: Low-level 1: High-level
16	STMSEL0	STMSEL0 This bit indicates the value of the Option byte STMSEL0 latched 0: Low-level 1: High-level
15 to 5	Reserved	When read, the value after reset is returned.
4	MODE0	MODE0 This bit indicates the value of the Mode Pin MODE0 latched 0: Low-level 1: High-level
3	FLMD2	FLMD2 This bit indicates the value of the Mode Pin FLMD2 latched 0: Low-level 1: High-level
2	FLMD1	FLMD1 This bit indicates the value of the Mode Pin FLMD1 latched 0: Low-level 1: High-level
1	FLMD0	FLMD0 This bit indicates the value of the Mode Pin FLMD0 latched 0: Low-level 1: High-level
0	TRST	TRST This bit indicates the value of the Mode Pin $\overline{\text{TRST}}$ latched 0: Low-level 1: High-level

5.5 Mode Error

This MCU has the function to notify the ECM of errors when it detects an unintended activation of a different mode or unintended de-activation of the currently selected mode. The Mode pins, STMSEL0, and STMSEL1 bits are monitored to detect unintended changes. If these values change when they are not expected to change, then a mode check error will occur.
The unintended modes are as follows. For details, see **Section 45, Error Control Module (ECM)**.
Detectable unintended modes are shown in **Table 5.7**.

Table 5.7 The List of Detectable Mode Errors

User Intended Mode	Detectable Mode Error
Normal Operation Mode	Unintended deactivation of Normal Operation Mode.
	Unintended activation of User Boot Mode
	Unintended activation of Serial Programming Mode
	Unintended activation of Production Test Mode
User Boot Mode*1	Unintended deactivation of User Boot Mode.
	Unintended activation of Normal Operation Mode.
	Unintended activation of Serial Programming Mode
	Unintended activation of Production Test Mode
Other than On chip debug mode	Unintended Debug Enabled detection
Any mode	The detection of differences between latched value and coming level from Flash with STMSEL0, STMSEL1. It is called mode check error.

Note 1. User Boot Mode 1 is not included.

NOTE

Unintended activation or de-activation mode error can be detected only when the set of the external terminals and the set of STMSEL1, STMSEL0 are correspond with Normal Operation Mode or User Boot Mode 0 under the condition of $\overline{\text{TRST}}$ = 0.

Although it is possible to change FLMD0 after $\overline{\text{RESET}}$ is released, the mode error is also detected when FLMD0 is changed from “1” to “0”. (e.g. If FLMD0 is changed to 0 after $\overline{\text{RESET}}$ release with setting of Serial Programming Mode 1 and STMSEL1 = 0, “Unintended activation of Serial Programming Mode” is detected.)

Section 6 Interrupts

This section describes the interrupt units.

6.1 Features of Interrupt Units

6.1.1 Units and Channels

This microcontroller has the following number of interrupt units.

Table 6.1 Number of Units (INTC1)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units per PE	1	1	1	1	1	1	1	1	1	1
Unit Name	INTC1									

Table 6.2 Number of Units (INTC2)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Unit Name	INTC2									

Table 6.3 Number of Units (INTIF)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Unit Name	INTIF									

Table 6.4 Number of Units (EINT)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Unit Name	EINT									

Table 6.5 Number of Units (FENC)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Unit Name	FENC									

Table 6.6 Number of Units (FEINC)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units per PE	1	1	1	1	1	1	1	1	1	1
Unit Name	FEINC									

6.1.2 Register Base Address

The register base addresses of the interrupt units are listed in **Table 6.7**.

The address of each register in this section is specified as an offset from one of these base addresses.

Table 6.7 Register Base Address

Base Address Name	SELF/PE	Base Address	Bus Group
<INTC1_SELF_base>	INTC1 of SELF	FFFC 0000$_H$	—
<INTC1_PE0_base>	INTC1 of PE0	FFFC 4000$_H$	CPU Peripheral 0
<INTC1_PE1_base>	INTC1 of PE1	FFFC 8000$_H$	CPU Peripheral 1
<INTC1_PE2_base>	INTC1 of PE2	FFFC C000$_H$	CPU Peripheral 2
<INTC1_PE3_base>	INTC1 of PE3	FFFD 0000$_H$	CPU Peripheral 3
<INTC2_base>	—	FFF8 0000$_H$	P-Bus Group 0
<INTIF_base>	—	FF09 0000$_H$	P-Bus Group 9
<EINT_base>	—	FFC0 0000$_H$	P-Bus Group 6L
<FENC_base>	—	FF9A 3A00$_H$	P-Bus Group 2L
<FEINC_PE0_base>	FEINC of PE0	FF9A 3B00$_H$	P-Bus Group 2L
<FEINC_PE1_base>	FEINC of PE1	FF9A 3C00$_H$	P-Bus Group 2L
<FEINC_PE2_base>	FEINC of PE2	FF9A 3D00$_H$	P-Bus Group 2L
<FEINC_PE3_base>	FEINC of PE3	FF9A 3E00$_H$	P-Bus Group 2L

6.1.3 Clock Supply

Clock supplies for each interrupt unit are shown in the following table.

Table 6.8 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
INTC1	clk	CLK_CPU
INTC2	pe_clk	CLK_CPU
	bus_clk	CLK_HBUS
INTIF	pclk	CLK_HBUS
EINT	clk_lsb	CLK_LSB
FENC	pclk	CLKA_LPS
FEINC	pclk	CLKA_LPS

6.1.4 Reset Sources

The reset sources for initialization of the interrupt units are shown in **Table 6.9**.

Table 6.9 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
INTC1	All registers*1	√	√	√	√	√	—	—
INTC2	All registers*1	√	√	√	√	√	—	—
INTIF	All registers*1	√	√	√	√	√	—	—
EINT	All registers*1	√	√	√	√	√	—	—
FENC	All registers*1	√	√	√	√	—	—	—
FEINC	All registers*1	√	√	√	√	—	—	—

Note 1. See **Table 6.13**.

6.1.5 External Input/Output Signals

External input/output signals of the interrupt units are listed below.

Table 6.10 External Input/Output Signals (1/2)

Unit Signal Name	Direction	Outline	Alternative Port Pin Signal
INTP0	Input	External pin INTP0	JP0_0, P0_0, P2_6, P6_13, P4_0, P10_7, P20_14
INTP1	Input	External pin INTP1	JP0_1, P0_1, P4_10, P6_7, P10_10, P10_12
INTP2	Input	External pin INTP2	JP0_2, P0_2, P3_3, P3_8, P4_13
INTP3	Input	External pin INTP3	JP0_3, P0_3, P2_5, P4_5, P6_5
INTP4	Input	External pin INTP4	P0_4, P2_0, P2_2, P4_7, P17_2
INTP5	Input	External pin INTP5	P0_5, P2_4, P2_7, P2_10, P4_12
INTP6	Input	External pin INTP6	P0_6, P2_6, P2_12, P4_15, P20_5
INTP7	Input	External pin INTP7	P0_7, P2_8, P2_14, P3_5, P20_8
INTP8	Input	External pin INTP8	P0_1, P0_8, P1_1, P3_9, P10_5, P22_3
INTP9	Input	External pin INTP9	P0_4, P0_9, P1_3, P3_10, P24_5
INTP10	Input	External pin INTP10	P0_5, P0_10, P1_5, P3_11, P20_14, P24_7
INTP11	Input	External pin INTP11	P0_7, P0_11, P1_7, P3_12, P24_9
INTP12	Input	External pin INTP12	P0_10, P0_12, P1_9, P3_13, P24_11
INTP13	Input	External pin INTP13	P0_11, P1_0, P1_11, P3_14, P24_13
INTP14	Input	External pin INTP14	P1_1, P1_13, P3_15, P8_1, P17_4
INTP15	Input	External pin INTP15	P1_2, P1_15, P4_2, P8_3, P17_6
INTP16	Input	External pin INTP16	P1_3, P2_9, P3_2, P4_9, P10_5, P10_8, P20_2, P20_14, P22_0
INTP17	Input	External pin INTP17	P1_4, P2_4, P3_3, P4_4, P24_4
INTP18	Input	External pin INTP18	P1_5, P2_1, P3_2, P3_4, P21_4, P21_7
INTP19	Input	External pin INTP19	P1_6, P2_11, P3_5, P4_15, P6_0

Table 6.10 External Input/Output Signals (2/2)

Unit Signal Name	Direction	Outline	Alternative Port Pin Signal
INTP20	Input	External pin INTP20	P1_7, P3_7, P4_11, P6_2, P6_15
INTP21	Input	External pin INTP21	P1_8, P2_3, P3_8, P4_6, P6_3
INTP22	Input	External pin INTP22	P1_9, P2_6, P4_0, P6_0, P6_4
INTP23	Input	External pin INTP23	P1_10, P2_13, P3_4, P4_1, P6_6
INTP24	Input	External pin INTP24	P1_11, P4_1, P5_6, P6_8
INTP25	Input	External pin INTP25	P1_12, P4_15, P5_2, P6_2, P6_11
INTP26	Input	External pin INTP26	P1_13, P5_3, P6_4, P6_12, P6_14
INTP27	Input	External pin INTP27	P1_14, P3_7, P5_6, P6_9, P6_14
INTP28	Input	External pin INTP28	P0_0, P1_0, P1_15, P4_4
INTP29	Input	External pin INTP29	P0_2, P1_2, P4_5, P8_0
INTP30	Input	External pin INTP30	P0_3, P1_4, P4_6, P8_1
INTP31	Input	External pin INTP31	P0_6, P1_6, P4_7, P8_2
INTP32	Input	External pin INTP32	P0_8, P1_8, P4_8, P8_3
INTP33	Input	External pin INTP33	P0_9, P1_10, P4_9, P8_4
INTP34	Input	External pin INTP34	P1_12, P4_10, P8_0, P8_5
INTP35	Input	External pin INTP35	P1_14, P4_11, P8_2, P8_6
INTP36	Input	External pin INTP36	P3_0, P5_4, P8_4, P8_7
INTP37	Input	External pin INTP37	P3_9, P6_9, P8_6, P8_8
INTP38	Input	External pin INTP38	P3_0, P3_11, P8_8, P8_9
INTP39	Input	External pin INTP39	P3_1, P3_13, P6_11, P8_10
FEINT	Input	External pin NMI	JP0_5, P4_7

Note: For details of Alternative Port Pin Signal, see Appendix “E02_01_List_of_Pin_Assignment.xlsx” in **Section 2, Pin Functions**.

6.1.6 Edge Detection Configuration

The external interrupts INTPm and NMI can be configured to generate an interrupt request upon a rising or falling edge or upon both edges of the external pin.

For details, refer to **Section 2.7, Noise Filter & Edge/Level Detector** and **Section 2.7.3.2, FCLACTLm_<name> — Filter Control Register**.

6.2 Overview

6.2.1 Outline

Interrupt Controller (INTC) determines priority of interrupt sources and controls interrupt requests to the CPU. INTC has a register to set priority for each interrupt source. Interrupt requests are processed according to the priority configured via the register.

INTC features

- Interrupt sources
  - Non-Maskable interrupt (FENMI): 1 channel for each CPU.
  - FE level interrupt (FEINT): 1 channel for each CPU.
  - Low latency EI level interrupt (maskable) (EIINT0 to 31):
    • 32 channels for each CPU.
    • Inter-processor interrupts: 4 channels shared on channel 0-3.
    • Broadcast interrupts: 4 channels shared on channel 4-7.
  - Low speed interrupts (maskable) (EIINT32 to 767):
    • 736 channels shared by all CPUs.
- Interrupt priority levels
  - Up to 64 priority levels can be configured by interrupt control registers.
- Interrupt detection method
  - The detection method of FEINT and EIINT interrupts are individually set to edge detection or level detection.
- Register banks.
  - For details about register banks, see **Section 3.2.4.5, Register Bank Function**.

INTC consists of INTC1 and INTC2.

- INTC1
  - All CPUs have their own interrupt controllers. Each CPU accesses the INTC1 that corresponds to the CPU. INTC1 controls low-latency interrupts and has the following functions:
    • Priority setting
    • Interrupt mask setting
    • Interrupt binding: Destination mode of the CPU (Host/Guest, GPID)
- INTC2
  - INTC2 is a common interrupt controller that all the CPUs share. INTC2 controls low-speed interrupts and has the following functions:
    • Priority setting.
    • Interrupt mask setting.
    • Interrupt binding: Destination PE, Destination mode of the CPU (Host/Guest, GPID).
    • Broadcast interrupt setting.

Others

- INTIF

  - Peripheral interrupt / TPTM interrupt control function.
- EINT
  - Software interrupt control function
- FENC
  - FENMI control function
- FEINC
  - FEINT control function

6.2.2 Functional Overview

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions. Exceptions are described in detail in **Section 3.2.4, Exceptions and Interrupts**.

The following three exceptions are called interrupts, and are described in this section:

- FE level Non-Maskable Interrupt (FENMI) An FENMI interrupt is acknowledged even if another FE level interrupt, FEINT, has been generated.
  - An FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
  - Return from an FENMI interrupt is not possible and recovery is disabled when multiple interrupts occur.
- FE level maskable interrupt (FEINT)
  An FEINT interrupt can be acknowledged only if an FE level non-maskable interrupt, FENMI, has not been generated.
  - An FEINT can be acknowledged if the CPU system register PSW.NP = 0.
  - Return is enabled and recovery is enabled.
- EI level maskable interrupt (EIINT)
  An EIINT interrupt can be acknowledged if an FE level interrupt, FENMI or FEINT, has not been generated.
  - An EIINT can be acknowledged only if the CPU system registers PSW.NP = 0 and PSW.ID = 0, and when the following two conditions are satisfied:
    - The interrupt priority level is higher than the priority level of the acknowledgment status, which is set to the CPU system register ISPR.
    - The interrupt priority level is higher than the priority level, which is set to the CPU system register PLMR.
  - Return is enabled and recovery is enabled.
  - Interrupt masking can be specified for each interrupt channel.
  - One of 64 interrupt priority levels can be specified for each interrupt channel.

For details about the PSW register, see **Section 3.2.3.2, (5) PSW — Program Status Word**.

6.2.3 Interrupt Sources

6.2.3.1 FE Level Non-Maskable Interrupts

For details about the priority order, return PC, status registers and return instructions, see **3.2.4, Exceptions and Interrupts**.

Table 6.11 Non-Maskable Interrupt Requests

Interrupt	Interrupt Request		Unit	Priority Order	Detection Type	Exception Cause Code	Offset Address
Symbol	Name	Cause					
FENMI	NMI	Secure WDT overflow interrupt	SWDT	*1	Edge	$0E0_H$	$0E0_H$

Note 1. See **Table 3.102, Exception Cause List**.

6.2.3.2 FE Level Maskable Interrupts

For details about the priority order, return PC, status registers and return instructions, see **Section 3.2.4, Exceptions and Interrupts**.

Table 6.12 FE Level Interrupt Requests

Interrupt	Interrupt Request	Unit	Priority Order	Detection Type	Exception Cause Code	Offset Address
Symbol	Cause					
PE0						
FEINT	NMI pin	Port	*1	Level	$F0_H$	$F0_H$
	Interrupt from Time Protection Timer for PE0*2	Time Protection Timer				
	FE level interrupt from the ECM	ECM				
PE1						
FEINT	NMI pin	Port	*1	Level	$F0_H$	$F0_H$
	Interrupt from Time Protection Timer for PE1*2	Time Protection Timer				
	FE level interrupt from the ECM	ECM				
PE2						
FEINT	NMI pin	Port	*1	Level	$F0_H$	$F0_H$
	Interrupt from Time Protection Timer for PE2*2	Time Protection Timer				
	FE level interrupt from the ECM	ECM				
PE3						
FEINT	NMI pin	Port	*1	Level	$F0_H$	$F0_H$
	Interrupt from Time Protection Timer for PE3*2	Time Protection Timer				
	FE level interrupt from the ECM	ECM				

Note 1. See **Table 3.102, Exception Cause List**.

Note 2. This interrupt can be used as an EIINT. See **Section 6.3.15, TPTMSEL — TPTM Interrupt FE EI Select Register**.

6.2.3.3 EI Level Maskable Interrupts

For details about the priority order, return PC and return instructions, see **Section 3.2.4, Exceptions and Interrupts**.

For details about the control registers, see **Section 6.3.2, EIC0 to EIC767 — EI Level Interrupt Control Register 0 to 767**.

For details, refer to Appendix file “Interrupt_table.xlsx”.

6.2.4 Block Diagram

The block diagram of the interrupt units is shown in **Figure 6.1**.

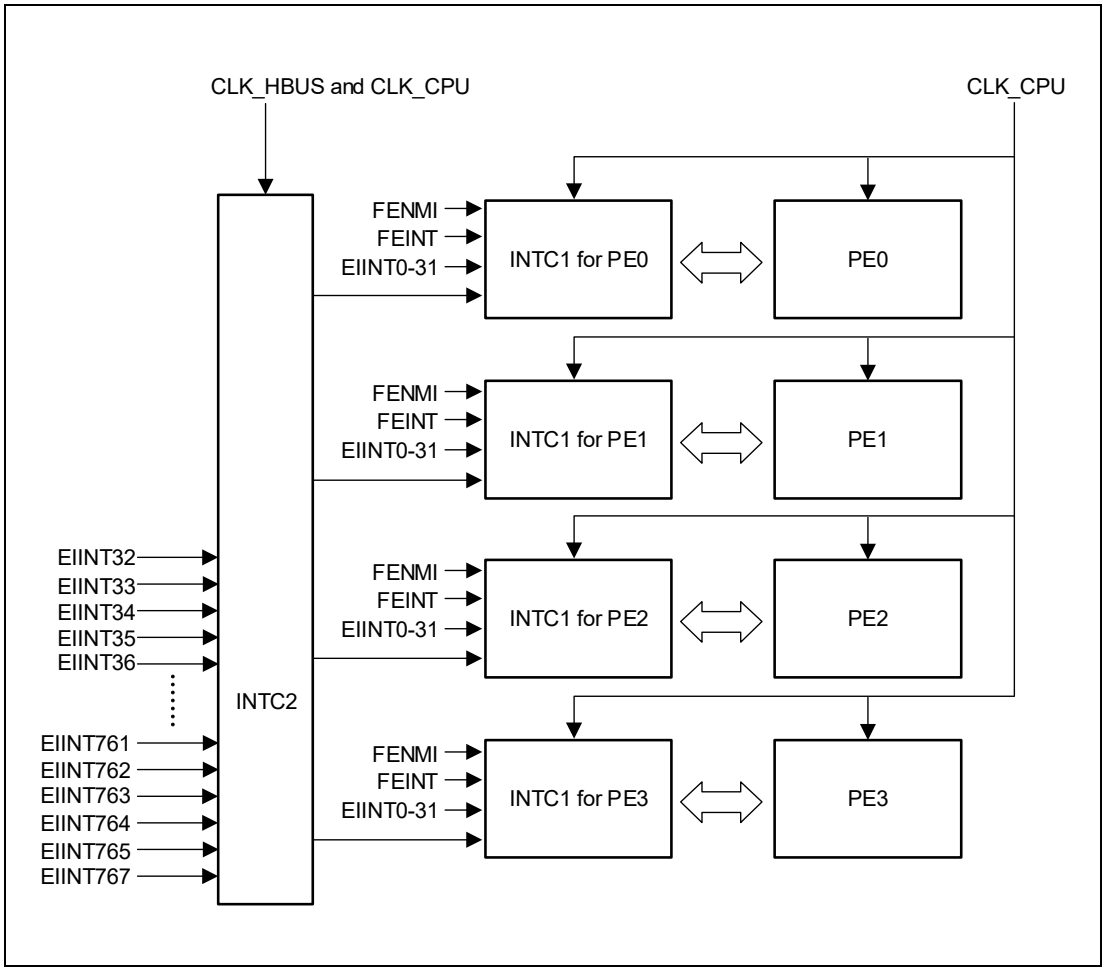


Figure 6.1 Block Diagram of Interrupt Units

6.3 Registers

6.3.1 List of Registers

Table 6.13 List of Register

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					PEG/INTC2G/PBG	Other
INTC1	EI level Interrupt Control Register*1	EICn (n = 0 to 31)	<INTC1_PEx_base*2> + 0000_H + 02_H × n	8, 16	*3	—
	EI level Interrupt Mask Register*1	IMR0	<INTC1_PEx_base*2> + $00F0_H$	8, 16, 32	*3	—
	EI level Interrupt Bind Register*1	EIBDn (n = 0 to 31)	<INTC1_PEx_base*2> + 0100_H + 04_H × n	8, 16, 32	*3	—
	FE level Interrupt Bind Register	FIBD	<INTC1_PEx_base*2> + $01C0_H$	8, 16, 32	*3	—
	Extended EI level Interrupt Control Register	EEICn (n = 0 to 31)	<INTC1_PEx_base*2> + 0200_H + 04_H × n	8, 16, 32	*3	—
	BGEIINT Priority Level Setting Register	EIBG	<INTC1_PEx_base*2> + 0280_H	8, 16, 32	*3	—
	BGFEINT Channel Mask Setting Register	FIBG	<INTC1_PEx_base*2> + $02C0_H$	8, 16, 32	*3	—
	INTC1 Virtualization Configuration Register	IHVCFG	<INTC1_PEx_base*2> + $02F0_H$	8, 16, 32	*3	—
INTC2	EI level Interrupt Control Register*1	EICn (n = 32 to 767)	<INTC2_base> + 0000_H + 02_H × n	8, 16	INTC2GPROT_n	—
	EI Level Interrupt Mask Register*1	IMRn (n = 1 to 23)	<INTC2_base> + 1000_H + 04_H × n	8, 16, 32	INTC2GPROT_IMR	—
	INTC2 BGEIINT Priority Level Setting Register	I2EIBGm (m = 0 to 3)	<INTC2_base> + $1FE0_H$ + 04_H × m	8, 16, 32	INTC2GPROT_GR	—
	EI level Interrupt Bind Register*1	EIBDn (n = 32 to 767)	<INTC2_base> + 2000_H + 04_H × n	8, 16, 32	INTC2GPROT_GR	—
	Extended EI level Interrupt Control Register	EEICn (n = 32 to 767)	<INTC2_base> + 4000_H + 04_H × n	8, 16, 32	INTC2GPROT_n	—
INTIF	Peripheral Interrupt Status Register n	PINTn (n = 0 to 7)	<INTIF_base> + 00_H + 04_H × n	32	PBG90#4	—
	Peripheral Interrupt Status Clear Register n	PINTCLRn (n = 0 to 7)	<INTIF_base> + 20_H + 04_H × n	32	PBG90#4	—
	TPTM Interrupt FE EI Select Register	TPTMSEL	<INTIF_base> + 200_H	8, 16, 32	PBG90#4	—
EINT	Software Interrupt Register 0	SINTR0	<EINT_base>	8	PBG6L0#3	—
	Software Interrupt Register 1	SINTR1	<EINT_base> + 4_H	8	PBG6L0#3	—
	Software Interrupt Register 2	SINTR2	<EINT_base> + 8_H	8	PBG6L0#3	—
	Software Interrupt Register 3	SINTR3	<EINT_base> + C_H	8	PBG6L0#3	—
FENC	FENMI Status Register	FENMIF	<FENC_base>	32	PBG20#8	—
	FENMI Status Clear Register	FENMIC	<FENC_base> + 8_H	32	PBG20#8	—
FEINC	FEINT Status Register	FEINTF	<FEINC_PEx_base>	32	PBG20#8	—
	FEINT Event Mask Register	FEINTMSK	<FEINC_PEx_base> + 4_H	32	PBG20#8	—
	FEINT Status Clear Register	FEINTC	<FEINC_PEx_base> + 8_H	32	PBG20#8	—

Note 1. It is recommended to execute a DI instruction to avoid incorrect detection of interrupts when the settings of these registers are changed.

Note 2. PEx: PE0, PE1, PE2, PE3, SELF (x = 0 to 3)

Note 3. x = 0: GUARD_PE0CL0
x = 1: GUARD_PE1CL0
x = 2: GUARD_PE2CL1
x = 3: GUARD_PE3CL1

See **Section 44.5.4.3, List of Registers** for details of PEG.

The **Table 6.14** shows Access Authority of INTC1 registers.

Table 6.14 Access Authority of INTC1 registers

Symbol	Access Authority		Description
	IHVE = 1	IHVE = 0	
EICn (n = 0 to 31)	SV	UM	When IHVCFG.IHVE = 1 and the CPU is in Guest mode*1, EICn can be accessed only by the Guest*1 to which channel n is bound.
IMR0	SV	UM	When IHVCFG.IHVE = 1 and the CPU is in Guest mode*1, Bit n can be written by the Guest*1 to which channel n is bound.
EIBDn (n = 0 to 31)	HV*1	UM	When IHVCFG.IHVE = 0, some bits are treated as Reserved bit.
FIBD	HV*1	UM	When IHVCFG.IHVE = 0, some bits are treated as Reserved bit.
EEICn (n = 0 to 31)	SV	UM	When IHVCFG.IHVE = 1 and the CPU is in Guest mode*1, EEICn can be accessed only by the Guest*1 to which channel n is bound.
EIBG	HV*1	UM	When IHVCFG.IHVE = 0, this register is treated as Reserved register.
FIBG	HV*1	UM	When IHVCFG.IHVE = 0, this register is treated as Reserved register.
IHVCFG	HV*1	UM	—

Note: HV: Hypervisor
SV: Supervisor
UM: User mode

Note 1. For details of HV (Hypervisor), Guest mode, Guest, see the *RH850G4MH Virtualization User's Manual: Hardware*.

6.3.2 EIC0 to EIC767 — EI Level Interrupt Control Register 0 to 767

These registers define control of EI level interrupts. There is one register for each EI interrupt source. For a complete list of interrupt sources, see the Interrupt table. All bits except EIP[5:4] of EEICn are shared with EICn (n is from 0 to 767, similarly thereafter). Use EEICn register if channel n is set as non-broadcast channel (EIBDn.CST = 0) and channel n is bound to a PE in 64-priority mode (EIBDn.PEID=m and INTCFG.EPL of PEm = 1). EEICn and EICn must be used exclusively.

CAUTION

If the EIRFn bit is set to 0 immediately after a peripheral module generates the corresponding interrupt request upon detection of an edge (before an interrupt is acknowledged by the CPU), the request may be lost. If EIOVn bit is set to 0 immediately after a peripheral module generates a second interrupt request, the overflow flag (EIOVn bit) may be lost. If the EIRFn bit is set to 1 immediately after an interrupt is acknowledged by the CPU, a new interrupt request will be generated. This also applies to bit manipulation instructions (set1, clr1, not1) described in Section 3.9.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation. Executing a bit-manipulation instruction to a byte has no effect on the other byte. EIRFn or EIOVn can be unintendedly updated by bit manipulations to the other bits of the byte.

Access: This register can be read or written in 16-bit or 8-bit units.

Address: EIC0 to EIC31: <INTC1_PEx_base> + 0000_H + 02_H × n (n = 0 to 31, x = 0 to 3)
EIC32 to EIC767: <INTC2_base> + 0000_H + 02_H × n (n = 32 to 767)

Value after reset: $008F_H$ (edge detection), $808F_H$ (level detection)*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	EIOVn	—	EIPn[3:0]			
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R	R/W*	R/W	R/W	R/W

Note 1. Interrupt detection by edge: 0_B
Interrupt detection by level: 1_B

Table 6.15 EIC0 to EIC767 Register Contents (1/2)

Bit Position	Bit Name	Function
15	EICTn	This bit indicates the type of interrupt detection. This bit is read only. 0: Detection by edge 1: Detection by level When writing in 8-bit or 16-bit units, write the value after a reset.
14, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	EIRFn	Interrupt Request Flag Operation varies depending on the interrupt input interface. 0: No interrupt request (Initial value) 1: Interrupt request present • Edge detection This flag is automatically cleared to 0 when an interrupt request from its own channel is acknowledged by the CPU core. • Level detection This bit cannot be set or cleared by the software. This is a read-only bit.

Table 6.15 EIC0 to EIC767 Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	EIMKn	Interrupt Mask If this bit is set to 1, interrupt requests set by the interrupt request flag (EIRFn) are masked to inhibit interrupt requests from the channel to the CPU core. The presence of unprocessed interrupts is not reported and the PMEI bit in ICSR is not set for the channel for which this bit is set to 1. Even when interrupt processing is disabled by the setting of this bit, the input of an interrupt signal is not masked and the interrupt request flag is set. The state of this bit is also reflected in the Interrupt Mask Register (IMR). When the interrupt request from the channel is masked with EIMKn = 1, the EIRFn still reflects the interrupt request for the channel and can be polled in software. When the EIMKn bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the EIMKn bit is also reflected in the corresponding IMRm register. 0: Interrupt processing is enabled. 1: Interrupt processing is disabled. (Initial value)
6	EITBn	Interrupt Vector Method Select 0: Direct vector method 1: Table reference method
5	EIOVn	Interrupt Overflow 0: No interrupt overflow 1: EIINTn rose in edge-detection mode (mode_Eict[n] = 0) when EICn.EIRF = 1. This bit is independent of write access to EICn.EIRF. If EICn.EIRF=1, EICn.EIOVn=0, and 1 is written to EICn.EIRF by 8-bit access, EICn.EIOV is NOT set. EICn.EIOV bit can be unintendedly cleared even by bit manipulation instruction to the other bits in EICn[7:0]. To avoid this, use EEICn.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	EIPn	These bits specify 16 interrupt priority levels (0: highest priority, 15: lowest priority). If two or more EI level interrupt requests are generated simultaneously, the source with a higher priority specified by these bits is selected and is sent to the CPU core. If the priorities specified by these bits are equal, the source with the lower default priority channel number is selected.

Note: n = 0 to 767
The values for the given channel numbers listed as reserved in the Interrupt table must not be set to anything other than their values after a reset.

NOTE

When a channel n is defined as broadcast interrupt (EIBDn.CST=1), EIMKn and EIRFn bits of the EICn register of the channel must be set to 0 after an initial configuration of the channel. In the period when the EIINTn interrupt is enabled, it is prohibited to mask (EIMKn=1) an interrupt processing of the channel. When it is necessary to mask a broadcast interrupt, EIC4 to EIC7 registers in INTC1 of each PE can be used to mask the corresponding broadcast interrupt.

6.3.3 IMR0 to IMR23 — EI Level Interrupt Mask Register 0 to 23

These registers are an aggregation of the EIMK bits from the EIC registers. Setting of the EIMK bit in the EIC register is reflected in this register. Also the setting of a bit in this register is reflected in the EIMK bit of the corresponding EIC register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: IMR0: <INTC1_PEx_base> + $00F0_H$ (x = 0 to 3)
IMRn (n = 1 to 23): <INTC2_base> + 1000_H + 04_H × n

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EIMK (n × 32 + 31)	EIMK (n × 32 + 30)	EIMK (n × 32 + 29)	EIMK (n × 32 + 28)	EIMK (n × 32 + 27)	EIMK (n × 32 + 26)	EIMK (n × 32 + 25)	EIMK (n × 32 + 24)	EIMK (n × 32 + 23)	EIMK (n × 32 + 22)	EIMK (n × 32 + 21)	EIMK (n × 32 + 20)	EIMK (n × 32 + 19)	EIMK (n × 32 + 18)	EIMK (n × 32 + 17)	EIMK (n × 32 + 16)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EIMK (n × 32 + 15)	EIMK (n × 32 + 14)	EIMK (n × 32 + 13)	EIMK (n × 32 + 12)	EIMK (n × 32 + 11)	EIMK (n × 32 + 10)	EIMK (n × 32 + 9)	EIMK (n × 32 + 8)	EIMK (n × 32 + 7)	EIMK (n × 32 + 6)	EIMK (n × 32 + 5)	EIMK (n × 32 + 4)	EIMK (n × 32 + 3)	EIMK (n × 32 + 2)	EIMK (n × 32 + 1)	EIMK (n × 32 + 0)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.16 IMR0 to IMR23 Register Contents

Bit Position	Bit Name	Function
31 to 0	EIMK (n × 32 + 31) to EIMK (n × 32 + 0)	These are interrupt mask bits for EI level maskable interrupt (INT) channels 0 to 767. 0: Interrupt processing enabled 1: Interrupt processing disabled

Note: n = 0 to 23
The value of EIMK bits listed as reserved for the given channel numbers in the Interrupt table must be set to 1.

NOTES

1. When writing to the EIMK bit by writing IMR1 to 23 registers, only access from the PE set in the EIBDn.PEID (n =32 to 767) register in the corresponding channel is possible. The EIMK bit is not updated when writing from a different PE.
2. When a channel is used as broadcast interrupt (EIBDn.CST=1), EIMK bit corresponding to the channel must be set to 0.
3. When virtualization is enabled, IMR1-23 write access control is the following table.

PEID	CPU mode	EIBD.GM	EIBD.GPID	Write control
Unmatch	-	-	-	Disable
Match	Host	-	-	Enable
Match	Guest	0	-	Disable
Match	Guest	1	Unmatch	Disable
Match	Guest	1	Match	Enable

6.3.4 EIBD0 to EIBD31 — EI Level Interrupt Bind Register 0 to 31

These registers are provided for each EI level interrupt source to define binding between interrupt sources and PEs. For details on each of the interrupt sources, see the Interrupt table.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC1_PEx_base> + 0100_H + 04_H × n (n = 0 to 31, x = 0 to 3)

Value after reset: 0000 0000_H (PE0)
0000 0001_H (PE1)
0000 0002_H (PE2)
0000 0003_H (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GM	—	—	—	—	GPID[2:0]			—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 6.17 EIBD0 to EIBD31 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	GM	0: Channel is bound to Host partition 1: Channel is bound to Guest partition This bit can be written only if IHVCFG.IHVE = 1.
14 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	GPID[2:0]	These bits specify the destination partition when EIBDn.GM = 1. These bits can be written only if IHVCFG.IHVE = 1.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PEID[2:0]	These bits specify the interrupt bind (request) destination. These bits are fixed in EIBD0 to EIBD31 and cannot be modified. 000: Interrupt is bound to PE0. 001: Interrupt is bound to PE1. 010: Interrupt is bound to PE2. 011: Interrupt is bound to PE3.

Note: The values for the given channel numbers listed as reserved in the Interrupt table must not be set to anything other than their values after a reset.

NOTE

Changing the corresponding EIBDn register during the processing of an EIINT request is prohibited.

Note: n = 0 to 31

6.3.5 EIBD32 to EIBD767 — EI Level Interrupt Bind Register 32 to 767

These registers are provided for each EI level interrupt source to define a binding between interrupt sources 32 to 767 and CPU cores. For a list of interrupt sources, see the Interrupt table.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC2_base> + 2000_H + 04_H × n (n = 32 to 767)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CST	—	—	—	—	—	BCP[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GM	—	—	—	—	GPID[2:0]			—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 6.18 EIBD32 to EIBD767 Register Contents

Bit Position	Bit Name	Function
31	CST	Broadcast interrupt enable. If this bit is set, channel n is used as broadcast channel. This bit can be enabled only if EICn.EICT = 0 (Edge detection).
30 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 24	BCP[1:0]	When the CST bit is 1, these bits specify a broadcast interrupt port number setting. When the CST bit is 0, these bits do not specify anything.
23 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	GM	0: Channel is bound to Host partition 1: Channel is bound to Guest partition This bit can be written only if IHVCFG.IHVE = 1.
14 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	GPID[2:0]	These bits specify the destination partition when EIBDn.GM = 1. These bits can be written only if IHVCFG.IHVE = 1.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PEID[2:0]	These bits specify the interrupt bind (request) destination. 000: Interrupt is bound to PE0. 001: Interrupt is bound to PE1. 010: Interrupt is bound to PE2. 011: Interrupt is bound to PE3. Other than above: Setting prohibited To execute an interrupt of the corresponding source, be sure to set one of the above values.

Note: The values for the given channel numbers listed as reserved in the Interrupt table must not be set to anything other than their values after a reset.

NOTE

Changing the corresponding EIBDn (n = 32 to 767) register during the processing of an EIINT request is prohibited.

6.3.6 FIBD — FE Level Interrupt Bind Register

These registers are provided for each FE level interrupt source to make correspondence between each source and PE.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC1_PEx_base> + $01C0_H$ (x = 0 to 3)

Value after reset: $0000\ 0000_H$ (PE0)
$0000\ 0001_H$ (PE1)
$0000\ 0002_H$ (PE2)
$0000\ 0003_H$ (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GM	—	—	—	—	GPID[2:0]			—	—	—	—	—	PEID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*	*
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 6.19 FIBD Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	GM	0: Channel is bound to Host. 1: Channel is bound to a Guest. This bit can be written only if IHVCFG.IHVE = 1.
14 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	GPID[2:0]	These bits specify the destination partition when EIBDn.GM = 1. These bits can be written only if IHVCFG.IHVE = 1.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PEID[2:0]	These bits specify an interrupt bind (request) destination. m: Interrupts are bound to PEm (m = 0 to 3). These bits of PEm is fixed to m.

6.3.7 EEIC0 to EEIC767 — Extended EI Level Interrupt Control Register 0 to 767

This register defines control of EI level interrupts. There is one register for each EI interrupt source. All bits except EIP[5:4] of EEICn are shared with EICn. Use EEICn register if channel n is set as non-broadcast channel (EIBDn.CST = 0) and channel n is bound to a PE in 64-priority mode (EIBDn.PEID=m and INTCFG.EPL of PEm =is 1) in 64-priority mode. EEICn and EICn must be used exclusively.

CAUTION

If EIRFn bit is set to 0 immediately after a peripheral module generates the corresponding interrupt request upon detection of an edge (before an interrupt is accepted by the CPU), the request may be lost. If EIOVn bit is set to 0 immediately after a peripheral module generates a second interrupt request, the overflow flag (EIOVn bit) may be lost.

If EIRFn bit is set to 1 immediately after an interrupt is accepted by the CPU, a new interrupt request will be generated.

This also applies to bit manipulation instructions (set1, clr1, not1) described in Section 3.9.2, Synchronization of Load Instruction Completion and Subsequent Instruction Generation.

Executing a bit-manipulation instruction to a byte has no effect on the other byte. EIRFn or EIOVn can be unintendedly updated by bit manipulations to the other bits of the byte.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: EEICn (n = 0 to 31): <INTC1_PEx_base> + 0200_H + 04_H × n (x = 0 to 3)
EEICn (n = 32 to 767): <INTC2_base> + 4000_H + 04_H × n

Value after reset: Value after reset: 0080 $000F_H$ (edge detection), 8080 $000F_H$ (level detection)*1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	—	—	—	—	—	—
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EIOVn	—	—	—	—	—	—	—	—	—	EIPn[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Interrupt detection by edge: 0_B
Interrupt detection by level: 1_B

Table 6.20 EEIC0 to EEIC767 Register Contents

Bit Position	Bit Name	Function
31	EICTn	This bit indicates the type of an interrupt detection. This bit is read only. 0: Detection with an edge 1: Detection of the level Written value to this bit must be the value after reset.
30, 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	EIRFn	Interrupt Request Flag Operation varies with the interrupt input interface. 0: No interrupt request (Initial value) 1: Interrupt request present • Edge detection This flag is automatically cleared to 0 when an interrupt request of the channel is accepted by the CPU core. • Level detection This bit cannot be set or cleared by the software. This is a read-only bit.
27 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	EIMKn	Interrupt Mask If this bit is set to 1, interrupt requests set in the interrupt request flag (EIRFn) are masked to inhibit interrupt requests from the channel to the CPU core. Notification of presence of unprocessed interrupts is not made and the PMEI bit in ICSR is not set from channels for which this bit is set to 1. Even when the interrupt request from the channel is masked with EIMKn = 1, the EIRFn still reflects the interrupt request for the channel and can be polled in software. When the EIMKn bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the EIMKn bit is also reflected in the corresponding IMRm register. 0: Interrupt processing is enabled. 1: Interrupt processing is disabled. (Initial value)
22	EITBn	Interrupt Vector Method Select 0: Direct branching method based on priority 1: Table referencing method
21 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	EIOVn	Interrupt Overflow 0: No interrupt overflow 1: EIINTn rose in edge-detection mode (mode_Eict[n] = 0) when EEICn.EIRF = 1. This bit is independent of write access to EEICn.EIRF. If EEICn.EIRF=1, EEICn.EIOVn=0, and 1 is written to EEICn.EIRF by 8-bit access, EEICn.EIOV is NOT set.
14 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	EIP5n to EIP0n	These bits specify 64 interrupt priority levels (0: highest priority, 63: lowest priority). If two or more EI level interrupt requests are generated simultaneously, a source with higher priority specified by these bits is selected and is sent to the CPU core. If the priority specified by these bits is equal, a source of less-number channel is selected as fixed priority. Note that the interrupt (EIINTn) with priority level 63 is not acknowledged in any case. For details, see **Section 3.2.4.1, (5) Interrupt Exception Priority and Priority Masking**. Even in 16-priority mode, EIP[5:4] is used for interrupt arbitration. EEICn.EIP[5:4] must be 0 if the channel is bound to a PE running in 16-priority mode.

Note: n = 0 to 767
The values of bits listed as reserved for the given channel numbers in the interrupt table must be retained after a reset.

6.3.8 IHVCFG — INTC1 Virtualization Configuration Register

This register is prepared for each INTC1 to enable configuration of virtualization-related bits of INTC1 registers. This register does not affect INTC1s in other PEs.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC1_PEx_base> + $02F0_H$ (x = 0 to 3)

Value after reset: $0000\ 000x_H$ (PE0)
$0000\ 000x_H$ (PE1)
$0000\ 000x_H$ (PE2)
$0000\ 000x_H$ (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IHVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Note 1. Initial value is defined by Option Byte. See **Section 51.12, Configuration Setting Area (Option Bytes, Reset Vector)**.

Table 6.21 IHVCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IHVE*2	Configuration of virtualization-related bits of INTC1 registers is enabled if this bit is set. Refer to the description of each register to know which bits are under control of this bit. It is mandatory that the value of this is bit is the same as the value of HVCFG.HVE bit even though the values can be configured independently.

Note 2. When the value of this bit needs to be changed from the initial value, the change must be made before the settings of other functions are changed, at the beginning of the CPU initialization process after releasing reset. Do not change the value of this bit during user program operation.

6.3.9 EIBG — BGEIINT Priority Level Setting Register

This register is used to configure priority threshold of BGEIINT used in request arbitration.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC1_PEx_base> + 0280_H (x = 0 to 3)

Value after reset: $0000\ 0000_H$ (PE0)
$0000\ 0000_H$ (PE1)
$0000\ 0000_H$ (PE2)
$0000\ 0000_H$ (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BGPR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.22 EIBG Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	BGPR[5:0]	These bits are priority threshold of BGEIINT used in request arbitration. EEICn.EIP < EIBG.BGPR: request to background Guest is allowed. EEICn.EIP ≥ EIBG.BGPR: request to background Guest is NOT allowed. (the less-number priority is the higher priority) These bits can be written only if IHVCFG.IHVE = 1. Set the same value to both EIBG.BGPR of PEx and I2EIBGm.BGPR.

6.3.10 I2EIBG0 to I2EIBG3 — INTC2 BGEIINT Priority Level Setting Register

This register is used to configure priority threshold of BGEIINT used in request arbitration.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC2_base> + $1FE0_H + 04_H \times m$ (m = 0 to 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	BGPR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.23 I2EIBG0 to I2EIBG3 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	BGPR[5:0]	These bits are priority threshold of BGEIINT to PEm used in request arbitration if EIBDn.PEID = m. EEICn.EIP < I2EIBGm.BGPR: request to background Guest is allowed. EEICn.EIP ≥ I2EIBGm.BGPR: request to background Guest is NOT allowed. (the less-number priority is the higher priority) Set the same value to both EIBG.BGPR of PEx and I2EIBGm.BGPR.

6.3.11 FIBG — BGFEINT Channel Mask Setting Register

This register is used to enable each BGFEINT.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <INTC1_PEx_base> + $02C0_H$ (x = 0 to 3)

Value after reset: $0000\ 0000_H$ (PE0)
$0000\ 0000_H$ (PE1)
$0000\ 0000_H$ (PE2)
$0000\ 0000_H$ (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BGE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 6.24 FIBG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BGE	0: FEINT to background Guest is NOT allowed. 1: FEINT to background Guest is allowed. This bit can be written only if IHVCFG.IHVE = 1.

6.3.12 SINTR0 to SINTR3 — Software Interrupt Register

SINTR0 to SINTR3 are 8-bit registers used to control software interrupts 0 to 3 (SINT0 to SINT3).

Writing 01_H to these registers increments the value of the counter; writing 00_H decrements it. When the value of the counter for any of these registers is 1 or more, the corresponding interrupt from among software interrupts 0 to 3 (SINT0 to SINT3) is generated. When reading the value from any of these registers, the read value is the current value of the counter.

Access: This register can be read or written in 8-bit units.

Address: <EINT_base> + 4 * n (n = 0 to 3)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	SINTCn[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.25 SINTR0 to SINTR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	SINTCn[7:0]	Software Interrupt Request This bit generates a software interrupt. [Reading operation] The number of SINTn interrupt request counts is read out. [Writing operation] Writing 01_H: Increments the counter.*1 Writing 00_H: Decrements the counter.*2

Note 1. When 01_H is written to the register while the value of the counter is FF_H, the counter is not incremented and its value remains FF_H.

Note 2. When 00_H is written to the register while the value of the counter is 00_H, the counter is not decremented and its value remains 00_H.

6.3.13 PINTn + x — Peripheral Interrupt Status Register

All 128 DTS interrupt flags are divided between 8 registers PINT0 to PINT7. Each register contains 32 interrupt flags. Only one bit in each register PINT0 to PINT7 can be set. If multiple DTS channels within the same PINTn register request an interrupt, only the bit of the lowest DTS channel in that register is set.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <INTIF_base> + 00_H + 04_H × (n + x)

Value after reset: 0000 0000_H

● PINT n + x (n = 0 to 3, x = 0)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTDTS[31 + 32 * n]	INTDTS[30 + 32 * n]	INTDTS[29 + 32 * n]	INTDTS[28 + 32 * n]	INTDTS[27 + 32 * n]	INTDTS[26 + 32 * n]	INTDTS[25 + 32 * n]	INTDTS[24 + 32 * n]	INTDTS[23 + 32 * n]	INTDTS[22 + 32 * n]	INTDTS[21 + 32 * n]	INTDTS[20 + 32 * n]	INTDTS[19 + 32 * n]	INTDTS[28 + 32 * n]	INTDTS[17 + 32 * n]	INTDTS[16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTDTS[15 + 32 * n]	INTDTS[14 + 32 * n]	INTDTS[13 + 32 * n]	INTDTS[12 + 32 * n]	INTDTS[11 + 32 * n]	INTDTS[10 + 32 * n]	INTDTS[9 + 32 * n]	INTDTS[8 + 32 * n]	INTDTS[7 + 32 * n]	INTDTS[6 + 32 * n]	INTDTS[5 + 32 * n]	INTDTS[4 + 32 * n]	INTDTS[3 + 32 * n]	INTDTS[2 + 32 * n]	INTDTS[1 + 32 * n]	INTDTS[0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.26 PINTn + x Register Contents (n = 0 to 3, x = 0)

Bit Position	Bit Name	Function
31 to 0	INTDTS	DTS ch (0 + 32 × n) to (31 + 32 × n) transfer completion interrupt status 0: No transfer complete interrupt request present 1: Transfer complete interrupt request present

● PINT n + x (n = 0 to 3, x = 4)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCDTS [31 + 32 * n]	INTCDTS [30 + 32 * n]	INTCDTS [29 + 32 * n]	INTCDTS [28 + 32 * n]	INTCDTS [27 + 32 * n]	INTCDTS [26 + 32 * n]	INTCDTS [25 + 32 * n]	INTCDTS [24 + 32 * n]	INTCDTS [23 + 32 * n]	INTCDTS [22 + 32 * n]	INTCDTS [21 + 32 * n]	INTCDTS [20 + 32 * n]	INTCDTS [19 + 32 * n]	INTCDTS [28 + 32 * n]	INTCDTS [17 + 32 * n]	INTCDTS [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCDTS [15 + 32 * n]	INTCDTS [14 + 32 * n]	INTCDTS [13 + 32 * n]	INTCDTS [12 + 32 * n]	INTCDTS [11 + 32 * n]	INTCDTS [10 + 32 * n]	INTCDTS [9 + 32 * n]	INTCDTS [8 + 32 * n]	INTCDTS [7 + 32 * n]	INTCDTS [6 + 32 * n]	INTCDTS [5 + 32 * n]	INTCDTS [4 + 32 * n]	INTCDTS [3 + 32 * n]	INTCDTS [2 + 32 * n]	INTCDTS [1 + 32 * n]	INTCDTS [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.27 PINTn + x Register Contents (n = 0 to 3, x = 4)

Bit Position	Bit Name	Function
31 to 0	INTCDTS	DTS ch (0 + 32 × n) to (31 + 32 × n) transfer count match interrupt status 0: No transfer count match interrupt request present 1: Transfer count match Interrupt request present

6.3.14 PINTCLRn + x — Peripheral Interrupt Status Clear Register

Writing a value from the interrupt read register (PINTn) to the corresponding interrupt clear register (PINTCLRn) will clear that interrupt.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <INTIF_base> + 20_H + 04_H × (n + x)

Value after reset: 0000 0000_H

- **PINTCLR n + x (n = 0 to 3, x = 0)**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCLR [31 + 32 * n]	INTCLR [30 + 32 * n]	INTCLR [29 + 32 * n]	INTCLR [28 + 32 * n]	INTCLR [27 + 32 * n]	INTCLR [26 + 32 * n]	INTCLR [25 + 32 * n]	INTCLR [24 + 32 * n]	INTCLR [23 + 32 * n]	INTCLR [22 + 32 * n]	INTCLR [21 + 32 * n]	INTCLR [20 + 32 * n]	INTCLR [19 + 32 * n]	INTCLR [28 + 32 * n]	INTCLR [17 + 32 * n]	INTCLR [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCLR [15 + 32 * n]	INTCLR [14 + 32 * n]	INTCLR [13 + 32 * n]	INTCLR [12 + 32 * n]	INTCLR [11 + 32 * n]	INTCLR [10 + 32 * n]	INTCLR [9 + 32 * n]	INTCLR [8 + 32 * n]	INTCLR [7 + 32 * n]	INTCLR [6 + 32 * n]	INTCLR [5 + 32 * n]	INTCLR [4 + 32 * n]	INTCLR [3 + 32 * n]	INTCLR [2 + 32 * n]	INTCLR [1 + 32 * n]	INTCLR [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 6.28 PINTCLRn + x Register Contents (n = 0 to 3, x = 0)

Bit Position	Bit Name	Function
31 to 0	INTCLR	These bits clear the DTS ch (0 + 32 × n) to (31 + 32 × n) transfer completion interrupt status. These bits should be written with the value corresponding to the PINT(n + x) register inside an interrupt handler routine.

- **PINTCLR n + x (n = 0 to 3, x = 4)**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	INTCTCLR [31 + 32 * n]	INTCTCLR [30 + 32 * n]	INTCTCLR [29 + 32 * n]	INTCTCLR [28 + 32 * n]	INTCTCLR [27 + 32 * n]	INTCTCLR [26 + 32 * n]	INTCTCLR [25 + 32 * n]	INTCTCLR [24 + 32 * n]	INTCTCLR [23 + 32 * n]	INTCTCLR [22 + 32 * n]	INTCTCLR [21 + 32 * n]	INTCTCLR [20 + 32 * n]	INTCTCLR [19 + 32 * n]	INTCTCLR [28 + 32 * n]	INTCTCLR [17 + 32 * n]	INTCTCLR [16 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTCTCLR [15 + 32 * n]	INTCTCLR [14 + 32 * n]	INTCTCLR [13 + 32 * n]	INTCTCLR [12 + 32 * n]	INTCTCLR [11 + 32 * n]	INTCTCLR [10 + 32 * n]	INTCTCLR [9 + 32 * n]	INTCTCLR [8 + 32 * n]	INTCTCLR [7 + 32 * n]	INTCTCLR [6 + 32 * n]	INTCTCLR [5 + 32 * n]	INTCTCLR [4 + 32 * n]	INTCTCLR [3 + 32 * n]	INTCTCLR [2 + 32 * n]	INTCTCLR [1 + 32 * n]	INTCTCLR [0 + 32 * n]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 6.29 PINTCLRn + x Register Contents (n = 0 to 3, x = 4)

Bit Position	Bit Name	Function
31 to 0	INTCTCLR	These bits clear the DTS ch (0 + 32 × n) to (31 + 32 × n) transfer count match interrupt status. These bits should be written with the value corresponding to the PINT(n + x) register inside an interrupt handler routine.

6.3.15 TPTMSEL — TPTM Interrupt FE EI Select Register

This register selects how TPTM interrupts are mapped in each CPU core.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units

Address: <INTIF_base> + 200_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	TPTMSEL3	TPTMSEL2	TPTMSEL1	TPTMSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 6.30 TPTMSEL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	TPTMSEL3	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE3. 0: The TPTM interrupt of PE3 is connected to FEINT. 1: The TPTM interrupt of PE3 is connected to EIINT
2	TPTMSEL2	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE2. 0: The TPTM interrupt of PE2 is connected to FEINT. 1: The TPTM interrupt of PE2 is connected to EIINT
1	TPTMSEL1	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE1. 0: The TPTM interrupt of PE1 is connected to FEINT. 1: The TPTM interrupt of PE1 is connected to EIINT
0	TPTMSEL0	This bit selects whether the FEINT or the EIINT is the TPTM interrupt of PE0. 0: The TPTM interrupt of PE0 is connected to FEINT. 1: The TPTM interrupt of PE0 is connected to EIINT

NOTE

In U2A-EVA, TPTM underflow interrupt (EIINT31) can not be used. Therefore, setting other than default value ($0000\ 0000_H$) to TPTMSEL register is prohibited in U2A-EVA.

6.3.16 FENMIF — FENMI Status Register

This register contains information about which interrupt has generated the FE level non-maskable interrupt (FENMI).

Access: This register is a read-only register that can be read in 32-bit units.

Address: <FENC_base>

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWDTNMIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.31 FENMIF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWDTNMIF	Secure WDT overflow interrupt flag 0: No Secure WDT overflow interrupt occurred. 1: Secure WDT overflow interrupt occurred.

6.3.17 FENMIC — FENMIC Status Clear Register

This register clears the FE level non-maskable interrupt flag.

Access: This register is a write-only register that can be written in 32-bit units

Address: <FENC_base> + 8_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWDTNMIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 6.32 FENMIC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWDTNMIC	Secure WDT overflow interrupt flag clear 0: — 1: Secure WDT overflow interrupt flag is cleared.

6.3.18 FEINTF — FEINT Status Register

This register contains information about which interrupt has generated the FE level maskable interrupt (FEINT).

Access: This register is a read-only register that can be read in 32-bit units.

Address: <FEINC_PEx_base> (x = 0 to 3)

Value after reset: 0000 0000_H (PE0)
0000 0000_H (PE1)
0000 0000_H (PE2)
0000 0000_H (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ECM PEx FEINTF	TPTM PEx FEINTF	NMI FEINTF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 6.33 FEINTF Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	ECMPExFEINTF	ECM PEx FEINT occurrence 0: No ECM PEx FEINT occurred. 1: ECM PEx FEINT occurred.
1	TPTMPExFEINTF	TPTM PEx FEINT occurrence 0: No TPTM PEx FEINT occurred. 1: TPTM PEx FEINT occurred.
0	NMIFEINTF	NMI pin FEINT occurrence 0: No NMI pin FEINT occurred. 1: NMI pin FEINT occurred.

6.3.19 FEINTMSK — FEINT Event Mask Register

This register masks the FE level maskable interrupt (FEINT).

Access: This register can be read or written in 32-bit units.

Address: <FEINC_PEx_base> + 4_H (x = 0 to 3)

Value after reset: 0000 0007_H (PE0)
0000 0007_H (PE1)
0000 0007_H (PE2)
0000 0007_H (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ECM PEx FEINT MSK	TPTM PEx FEINT MSK	NMI FEINT MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 6.34 FEINTMSK Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECMPExFEINTMSK	ECM PEx FEINT mask 0: Not masked ECM PEx FEINT. 1: Masked ECM PEx FEINT.
1	TPTMPExFEINTMSK	TPTM PEx FEINT mask 0: Not masked TPTM PEx FEINT. 1: Masked TPTM PEx FEINT.
0	NMIFEINTMSK	NMI pin FEINT mask 0: Not masked NMI pin FEINT. 1: Masked NMI pin FEINT.

6.3.20 FEINTC — FEINT Status Clear Register

This register clears the FE level maskable interrupt flags.

Access: This register is a write-only register that can be written in 32-bit units

Address: <FEINC_PEx_base> + 8_H (x = 0 to 3)

Value after reset: 0000 0000_H (PE0)
0000 0000_H (PE1)
0000 0000_H (PE2)
0000 0000_H (PE3)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ECM FEINTC	TPTM FEINTC	NMI FEINTC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Table 6.35 FEINTC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	ECMFEINTC	ECM FEINT flag clear 0: — 1: NMI pin FEINT flag is cleared.
1	TPTMFEINTC	TPTM FEINT flag clear 0: — 1: NMI pin FEINT flag is cleared.
0	NMIFEINTC	NMI pin FEINT flag clear 0: — 1: NMI pin FEINT flag is cleared.

6.4 Interrupt Operation

6.4.1 Level Interrupts

For the operation of level interrupts, see **Section 6.5.1, Level Interrupt Processing Flow**.

6.4.2 Inter-Processor Interrupts

For the operation of Inter-Processor Interrupts, see **Section 3.4, Inter-Processor Interrupt**.

6.4.3 Broadcast Interrupts

INTC2 has a function for broadcast interrupts. With this function, the EIINTn request input is transferred to EIINT4-7 of each CPU's INTC1 as a broadcast notification 0-3 with no priority judgement.

The interrupt detection type of the EIINTn must be edge detection if the broadcast function is used. The broadcast function cannot be activated (EIBDn.CST cannot be set) if the interrupt detection type of the EIINTn is configured as level detection.

When EIBDn.CST is set, EIINTn is used as the broadcast interrupt source.

EIBDn.BCP[1:0] determines which port of the broadcast interrupt (EIINT4-7) is triggered. It is prohibited to assign more than one interrupt channel to the same broadcast interrupt (EIINT4-7).

For the operation of broadcast interrupts, see **Section 6.3.5, EIBD32 to EIBD767 — EI Level Interrupt Bind Register 32 to 767** and **Section 6.5.4, Broadcast Interrupt Processing Flow**.

6.4.4 Software Interrupts

For the operation of software interrupts, see **Section 6.3.12, SINTR0 to SINTR3 — Software Interrupt Register** and **Section 6.4.6, Priority Level Handling.**

6.4.5 DTS Interrupt Merge Function

Up to 128 transfer end interrupts and up to 128 transfer count match interrupts are aggregated into one type of interrupt in units of 32 interrupts. 128 bits from all DTS channels are combined into 8 registers PINT0 to PINT7. If multiple DTS channels within the same status register (PINT0 to PINT7) request an interrupt, only the bit of the lowest DTS channel in that register is set. For additional information see **Section 6.3.13, PINTn + x — Peripheral Interrupt Status Register** and **Section 6.3.14, PINTCLRn + x — Peripheral Interrupt Status Clear Register**.

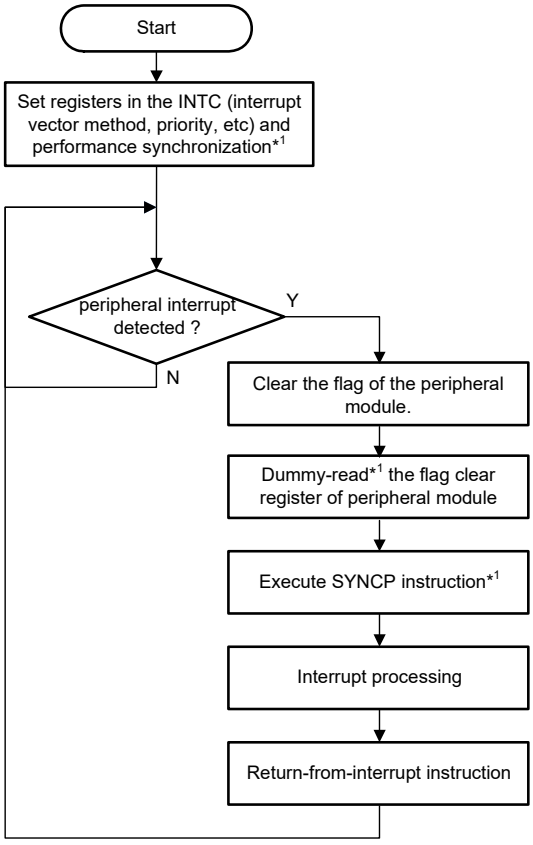
6.4.6 Priority Level Handling

There is no priority control for FENMI and FEINT because it has one channel.

EI-level maskable interrupt priority is determined within EIINT0-31 and EIINT32-767 in each PE. Priority is determined by 64 interrupt priority levels specified for each interrupt channel (EEICn.EIPn[5:0], EICn.EIPn[3:0]). In channels with same priority level, the channel with the lowest number is given priority over other channels.

6.5 Interrupt Processing Flow

6.5.1 Level Interrupt Processing Flow

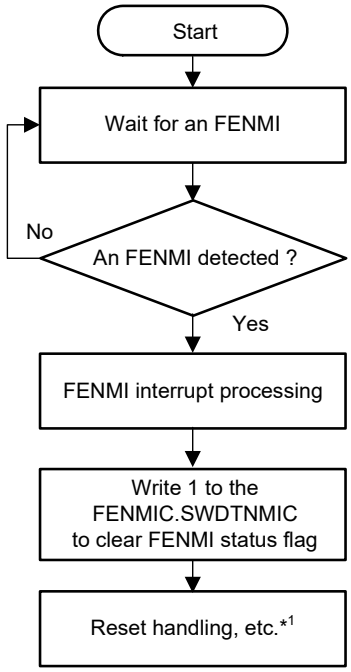


Note 1. See **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** for details about synchronization after setting registers.

Figure 6.2 Example of the Level Interrupt Processing Flow

6.5.2 FENMI Processing Flow

- After an FENMI has been detected from notification of secure WDT overflow, an interrupt request is sent to the INTC.
- FENMI interrupts are acknowledged as the highest priority even when another FE level interrupt has been generated. FENMI interrupts cannot be masked by PSW.NP in the CPU system register. Recovery or return from FENMI interrupt is not possible.

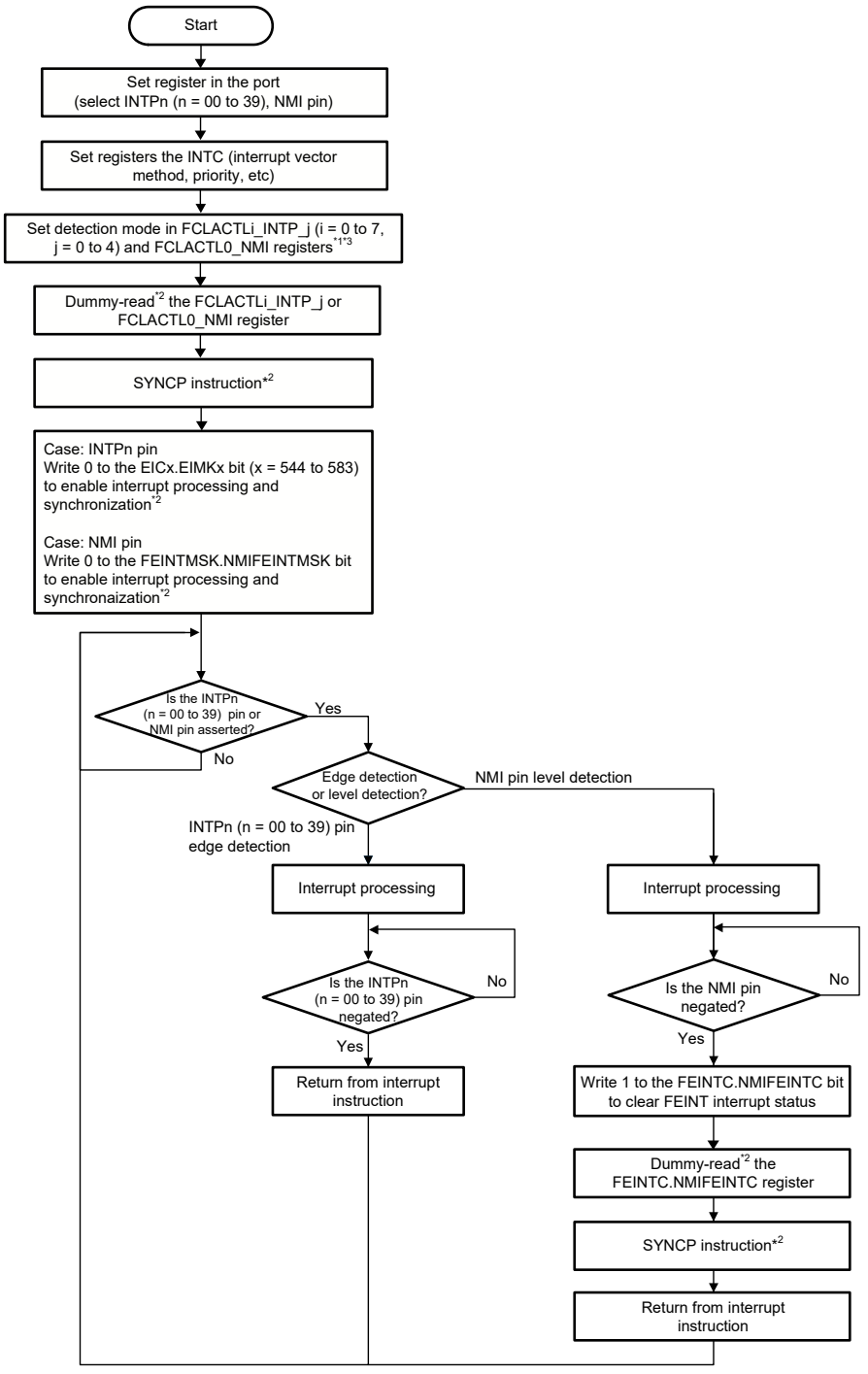


Note 1. The system registers (FEPC and FEPSW) are overwritten upon FENMI acknowledgment. For branching to user processing after FENMI acknowledgment, set the FEPC and FEPSW registers according to the needs of the application and execute an FERET instruction.

Figure 6.3 Example of the FENMI Processing Flow

6.5.3 External Interrupt Processing Flow

- Select the INTPn (n = 00 to 39) and FEINT(NMI) detection method (edge detection or level detection) by setting FCLACTLi_INTP_j (i = 0 to 7, j = 0 to 4) and FCLACTL0_NMI registers.
- After detection of INTPn (n = 00 to 39) and FEINT(NMI), an interrupt request is issued to the INTC.
- If level detection is selected, confirm that the INTPn (n = 00 to 39) and NMI pin is negated before returning from the interrupt service routine by an instruction.
- If edge detection is selected, clear the interrupt request before returning from an interrupt service routine by an instruction.



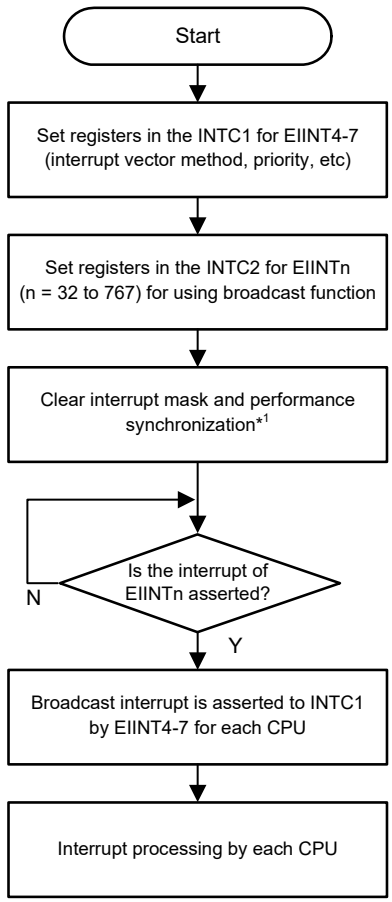
Note 1. See **Section 2.7, Noise Filter & Edge/Level Detector** and **Section 2.7.3.2, FCLACTLm_<name> — Filter Control Register**.

Note 2. See **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** for details about synchronization after setting registers.

Note 3. In order to avoid an unintended interrupt occurrence, use the following configuration sequence.
1. Configure the port registers.
2. Wait for the period of pulse rejection.
3. Configure the FCLACTLi_INPT_j and FCLACTL0_NMI registers.

Figure 6.4 Example of the External Interrupt Processing Flow

6.5.4 Broadcast Interrupt Processing Flow



Note 1. See **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** for details about synchronization after setting registers.

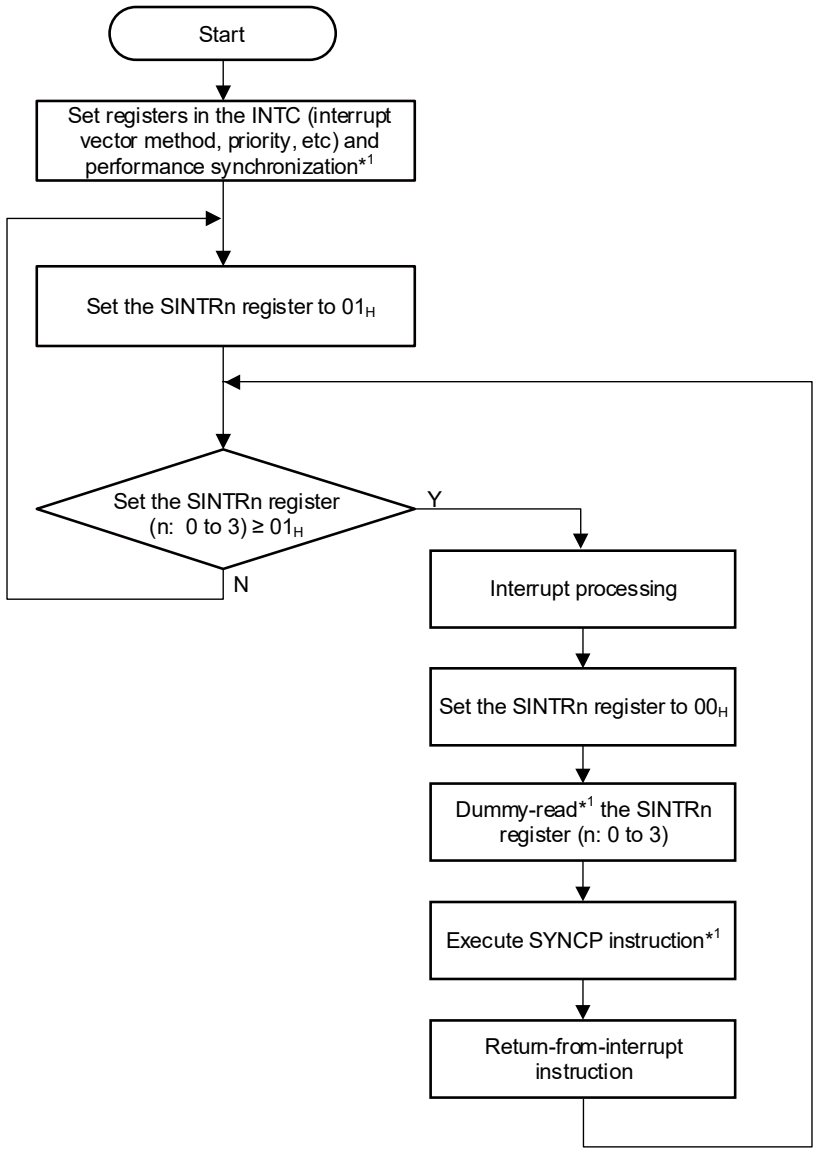
Figure 6.5 Example for the Broadcast Interrupt

NOTE

When using the broadcast function, the EIINTn (n = 32 to 767) interrupts are not generated individually upon the occurrence of each interrupt factor. Interrupts are generated in EIINT4-7 as broadcast interrupts.

6.5.5 Software Interrupt Processing Flow

- Software interrupt requests are controlled by writing 00_H or 01_H to the counter registers (SINTR0 to SINTR3).
- Writing 00_H leads to the counter's value being decremented by 1.
- Writing 01_H leads to the counter's value being incremented by 1.
- If the incremented counter value is 1 or above, an interrupt request for the INTC is generated.
- Before returning from interrupt service routine, decrement the SINTRn counter by 1. If SINTRn is 00_H after issuing the instruction to return from the interrupt, wait for the writing of 01_H to SINTRn.

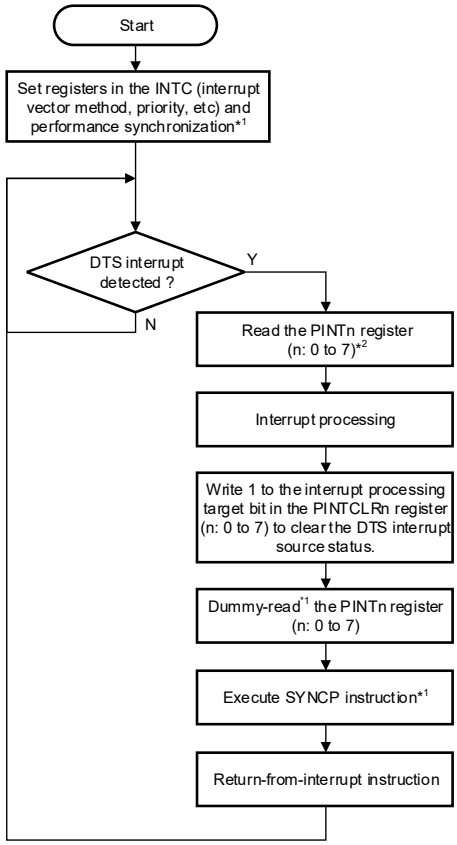


Note 1. See **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** for details about synchronization after setting registers.

Figure 6.6 Example of the Software Interrupt Processing Flow

6.5.6 DTS Interrupt Processing Flow

- When only one interrupt request is generated out of the bundled 32 interrupt sources
  - The bit corresponding to the interrupt request in the PINTn register is set to 1 and an interrupt request is output.
  - On completion of interrupt processing, write 1 to the interrupt clear register (PINTCLRn) to clear the interrupt request before issuing the return from interrupt instruction, then wait for the next one.
- When multiple interrupt sources are generated out of the bundled 32 interrupt sources
  - Only the bit of the DTS channel with the lowest number is set in the PINTn register and the request is generated.
  - On completion of interrupt processing (before returning from the interrupt service routine), write 1 to the corresponding bit in the interrupt clear register (PINTCLRn) to clear the interrupt request.
  - After clearing the interrupt request bit of the first DTS channel, the bit corresponding to the next lowest DTS channel requesting an interrupt is set and the interrupt request is output.
  - These steps are repeated until all interrupt sources bundled into the same 32-bits register are cleared.



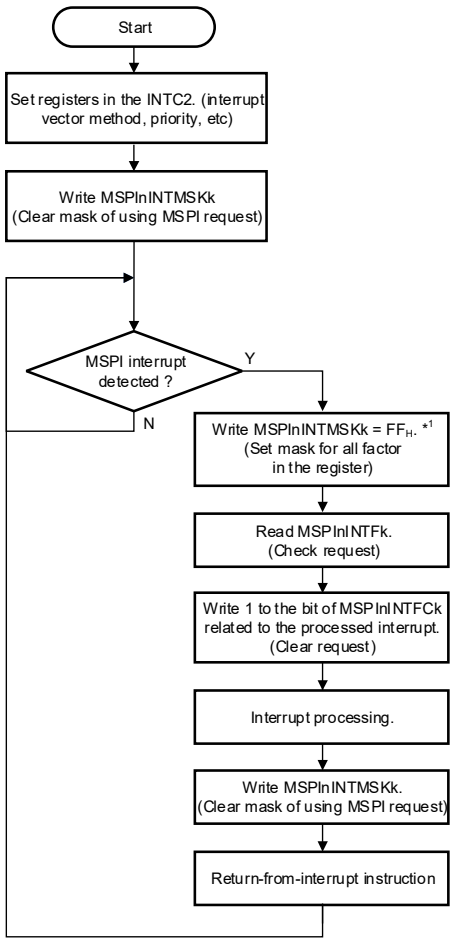
Note 1. See **Section 3.9.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation** for details about synchronization after setting registers.

Note 2. Use this data to clear the DTS interrupt status. It is necessary to set the data from the first read of the PINTn register after a DTS interrupt is detected.

Figure 6.7 Example of the DTS Interrupt Processing Flow

6.5.7 MSPI Interrupt Processing Flow

- Interrupt requests generated by MSPIn are combined into one request (Grouped Interrupt) and connected to one channel of INTC2.
  Among the Grouped Interrupt, the interrupt that has occurred can be confirmed in the MSPIn Interrupt Factor Register(MSPInINTFk).
  See **Section 19.1.4, Interrupt Requests and Error Notifications** for details.
- Some interrupt requests of MSPIn are independently connected to INTC2.
  For these interrupt requests, independent interrupt and Grouped Interrupts should be used exclusively.
- Even if multiple interrupts occur in MSPIn, there may be only one interrupt request to the CPU.
  When using INTMSPInTX, INTMSPInRX, INTMSPInFE, and INTMSPInERR, follow the flowchart in **Figure 6.8** for interrupt processing.



Note 1. Set only MSPInINTMSKk corresponding to the detected interrupt (INTMSPInTX, INTMSPInRX, INTMSPInFE, INTMSPInERR).

Figure 6.8 Example of the MSPI Interrupt Processing Flow

6.6 Interrupt Response Times

Table 6.36 Interrupt Response Times (Min)

Target	Interrupt Request Source		Number of Cycles for Processing			
	INTC Connection	Operating Clock	Synchronization	INTC2	INTC1	In PE0/PE1/PE2/PE3
PE0/PE1 PE2/PE3	Directly input to INTC1	CLK_CPU	1 × Iϕ	—	2 × Iϕ <1 × Iϕ>	See the description under “In PE0/PE1/PE2/PE3” below.
		CLK_LSB	2 × Pϕ + 1 × Sϕ			
		CLK_LSB (Peripheral Group 2L)	1 × Lϕ + 2 × Pϕ + 1 × Sϕ			
		CLK_HSB	—			
		CLK_HBUS	1 × Pϕ + 1 × Sϕ			
		CLK_UHSB	—			
	Input via INTC2	CLK_CPU	—	3 × Pϕ + 2 × Iϕ <2 × Pϕ + 2 × Iϕ>	—	
		CLK_LSB	2 × Pϕ			
		CLK_LSB (Peripheral Group 2L)	1 × Lϕ + 2 × Pϕ			
		CLK_HSB	1 × Hϕ + 2 × Pϕ			
		CLK_HSB (Peripheral Group 2H)	1 × Uϕ + 3 × Pϕ			
		CLK_HBUS	1 × Pϕ			
		CLK_UHSB	1 × Uϕ + 3 × Pϕ			

Note: The numbers in < > indicate the numbers of cycles in the case of level detection.

Note: Iϕ: CPU clock (CLK_CPU)
Sϕ: SBUS clock (CLK_SBUS)
Pϕ: HBUS clock (CLK_HBUS)
Hϕ: Peripheral high speed clock (CLK_HSB)
Lϕ: Peripheral low speed clock (CLK_LSB)
Uϕ: Peripheral ultra high speed clock (CLK_UHSB)

Table 6.37 Response Cycles in CPU Operation (Min)

Interrupt Vector Method	Cache HIT/ MISS	Virtualization support function*1	Register bank	In PE0 to PE3 (400MHz, 320MHz)	In PE0 to PE3 (240MHz)
Direct Vector	Cache Hit	Not use	—	5 × Iϕ	5 × Iϕ
		Host mode	—	5 × Iϕ	5 × Iϕ
		Guest mode	—	5 × Iϕ	5 × Iϕ
	Cache Miss	Not use	—	12 × Iϕ	11 × Iϕ
		Host mode	—	12 × Iϕ	11 × Iϕ
		Guest mode	—	12 × Iϕ	11 × Iϕ
Table Reference (The vector code is stored in Code Flash)	Cache Hit	Not use	Not use	15 × Iϕ	14 × Iϕ
			Save mode 0	20 × Iϕ	19 × Iϕ
			Save mode 1	26 × Iϕ	25 × Iϕ
		Host mode	—	15 × Iϕ	14 × Iϕ
		Guest mode	—	18 × Iϕ	17 × Iϕ
	Cache Miss	Not use	Not use	24 × Iϕ	22 × Iϕ
			Save mode 0	24 × Iϕ	22 × Iϕ
			Save mode 1	26 × Iϕ	25 × Iϕ
		Host mode	—	24 × Iϕ	22 × Iϕ
		Guest mode	—	27 × Iϕ	25 × Iϕ

Note: Iϕ: CPU clock (CLK_CPU)

Note 1. For details of virtualization support function, see the *RH850G4MH Virtualization User's Manual: Hardware*.

Table 6.38 Response Cycles in CPU Operation (BGEIINT)

Cache HIT/MISS	In PE0 to PE3 (400MHz, 320MHz)	In PE0 to PE3 (240MHz)
Cache Hit	5 × Iϕ	5 × Iϕ
Cache Miss	12 × Iϕ	11 × Iϕ

Note: Iϕ: CPU clock (CLK_CPU)

Note: For details of BGEIINT, see the *RH850G4MH Virtualization User's Manual: Hardware*.

Table 6.39 Response Cycles in CPU Operation (GMEIINT)

Interrupt Vector Method	Cache HIT/MISS	Register bank	In PE0 to PE3 (400MHz, 320MHz)	In PE0 to PE3 (240MHz)
Direct Vector	Cache HIit	—	5 × Iϕ	5 × Iϕ
	Cache Miss	—	12 × Iϕ	11 × Iϕ
Table Reference (The vector code is stored in Code Flash)	Cache Hit	Not use	15 × Iϕ	14 × Iϕ
		Save mode 0	20 × Iϕ	19 × Iϕ
		Save mode 1	26 × Iϕ	25 × Iϕ
	Cache Miss	Not use	24 × Iϕ	22 × Iϕ
		Save mode 0	24× Iϕ	22 × Iϕ
		Save mode 1	26 × Iϕ	25 × Iϕ

Note: Iϕ: CPU clock (CLK_CPU)

Note: For details of GMEIINT, see the *RH850G4MH Virtualization User's Manual: Hardware*.

Section 7 sDMA Controller (sDMAC)

This section contains a generic description of the sDMA Controller (sDMAC).

The first part of this section describes this product specific properties, such as the number of units and register base addresses.

The remainder of the section describes the functions and registers of the sDMAC.

7.1 Features sDMAC for RH850/U2A-EVA

7.1.1 Number of Units

This microcontroller has the following number of sDMAC units.

Table 7.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2	2	2	2	2	2	2	2	2	2
Number of Channels	16	16	16	16	16	16	16	16	16	16
Name	SDMACj (j = 0, 1)									

Table 7.2 Index

Index	Description
j	The individual units are identified by the index “j” (j = 0 to 1).
n	The individual channels are identified by the index “n” (n = 0 to 15).
m	The individual trigger group selection registers are identified by the index “m” (m = 0 to 15).

7.1.2 Register Base Addresses

sDMAC register addresses are given as offsets from the base addresses.

sDMAC base addresses are listed in the following table.

Table 7.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<SDMAC0_base>	FFF9 0000$_H$	P-Bus Group 0
<SDMAC1_base>	FFF9 8000$_H$	P-Bus Group 0
<DMATRGSEL_base>	FF09 0400$_H$	P-Bus Group 9

7.1.3 Clock Supply

The sDMAC clock supplies are shown in the following table.

Table 7.4 sDMAC Clock Supply

Unit Name	Clock Name in the Unit	Supply Clock Name	Description
SDMACj	SCLK	CLK_HBUS	System interconnect clock
	PCLK	CLK_HBUS	Register access clock
DMATRGSEL	PCLK	CLK_HBUS	Register access clock

7.1.4 Interrupts Requests and Error Notifications

sDMAC interrupt requests are listed in the following table.

Table 7.5 Interrupt Requests (1/2)

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number
INTSDMAC0CH0	SDMAC0.intreq_ch[0]	sDMAC0 channel 0 transfer end interrupt sDMAC0 channel 0 descriptor step end interrupt sDMAC0 channel 0 address error interrupt	47
INTSDMAC0CH1	SDMAC0.intreq_ch[1]	sDMAC0 channel 1 transfer end interrupt sDMAC0 channel 1 descriptor step end interrupt sDMAC0 channel 1 address error interrupt	48
INTSDMAC0CH2	SDMAC0.intreq_ch[2]	sDMAC0 channel 2 transfer end interrupt sDMAC0 channel 2 descriptor step end interrupt sDMAC0 channel 2 address error interrupt	49
INTSDMAC0CH3	SDMAC0.intreq_ch[3]	sDMAC0 channel 3 transfer end interrupt sDMAC0 channel 3 descriptor step end interrupt sDMAC0 channel 3 address error interrupt	50
INTSDMAC0CH4	SDMAC0.intreq_ch[4]	sDMAC0 channel 4 transfer end interrupt sDMAC0 channel 4 descriptor step end interrupt sDMAC0 channel 4 address error interrupt	51
INTSDMAC0CH5	SDMAC0.intreq_ch[5]	sDMAC0 channel 5 transfer end interrupt sDMAC0 channel 5 descriptor step end interrupt sDMAC0 channel 5 address error interrupt	52
INTSDMAC0CH6	SDMAC0.intreq_ch[6]	sDMAC0 channel 6 transfer end interrupt sDMAC0 channel 6 descriptor step end interrupt sDMAC0 channel 6 address error interrupt	53
INTSDMAC0CH7	SDMAC0.intreq_ch[7]	sDMAC0 channel 7 transfer end interrupt sDMAC0 channel 7 descriptor step end interrupt sDMAC0 channel 7 address error interrupt	54
INTSDMAC0CH8	SDMAC0.intreq_ch[8]	sDMAC0 channel 8 transfer end interrupt sDMAC0 channel 8 descriptor step end interrupt sDMAC0 channel 8 address error interrupt	55
INTSDMAC0CH9	SDMAC0.intreq_ch[9]	sDMAC0 channel 9 transfer end interrupt sDMAC0 channel 9 descriptor step end interrupt sDMAC0 channel 9 address error interrupt	56
INTSDMAC0CH10	SDMAC0.intreq_ch[10]	sDMAC0 channel 10 transfer end interrupt sDMAC0 channel 10 descriptor step end interrupt sDMAC0 channel 10 address error interrupt	57
INTSDMAC0CH11	SDMAC0.intreq_ch[11]	sDMAC0 channel 11 transfer end interrupt sDMAC0 channel 11 descriptor step end interrupt sDMAC0 channel 11 address error interrupt	58
INTSDMAC0CH12	SDMAC0.intreq_ch[12]	sDMAC0 channel 12 transfer end interrupt sDMAC0 channel 12 descriptor step end interrupt sDMAC0 channel 12 address error interrupt	59
INTSDMAC0CH13	SDMAC0.intreq_ch[13]	sDMAC0 channel 13 transfer end interrupt sDMAC0 channel 13 descriptor step end interrupt sDMAC0 channel 13 address error interrupt	60
INTSDMAC0CH14	SDMAC0.intreq_ch[14]	sDMAC0 channel 14 transfer end interrupt sDMAC0 channel 14 descriptor step end interrupt sDMAC0 channel 14 address error interrupt	61
INTSDMAC0CH15	SDMAC0.intreq_ch[15]	sDMAC0 channel 15 transfer end interrupt sDMAC0 channel 15 descriptor step end interrupt sDMAC0 channel 15 address error interrupt	62
INTSDMACERR	SDMAC0.intreq_aerr SDMAC1.intreq_aerr	sDMAC0 address error interrupt sDMAC1 address error interrupt	29

Table 7.5 Interrupt Requests (2/2)

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number
INTSDMAC1CH0	SDMAC1.intreq_ch[0]	sDMAC1 channel 0 transfer end interrupt sDMAC1 channel 0 descriptor step end interrupt sDMAC1 channel 0 address error interrupt	63
INTSDMAC1CH1	SDMAC1.intreq_ch[1]	sDMAC1 channel 1 transfer end interrupt sDMAC1 channel 1 descriptor step end interrupt sDMAC1 channel 1 address error interrupt	64
INTSDMAC1CH2	SDMAC1.intreq_ch[2]	sDMAC1 channel 2 transfer end interrupt sDMAC1 channel 2 descriptor step end interrupt sDMAC1 channel 2 address error interrupt	65
INTSDMAC1CH3	SDMAC1.intreq_ch[3]	sDMAC1 channel 3 transfer end interrupt sDMAC1 channel 3 descriptor step end interrupt sDMAC1 channel 3 address error interrupt	66
INTSDMAC1CH4	SDMAC1.intreq_ch[4]	sDMAC1 channel 4 transfer end interrupt sDMAC1 channel 4 descriptor step end interrupt sDMAC1 channel 4 address error interrupt	67
INTSDMAC1CH5	SDMAC1.intreq_ch[5]	sDMAC1 channel 5 transfer end interrupt sDMAC1 channel 5 descriptor step end interrupt sDMAC1 channel 5 address error interrupt	68
INTSDMAC1CH6	SDMAC1.intreq_ch[6]	sDMAC1 channel 6 transfer end interrupt sDMAC1 channel 6 descriptor step end interrupt sDMAC1 channel 6 address error interrupt	69
INTSDMAC1CH7	SDMAC1.intreq_ch[7]	sDMAC1 channel 7 transfer end interrupt sDMAC1 channel 7 descriptor step end interrupt sDMAC1 channel 7 address error interrupt	70
INTSDMAC1CH8	SDMAC1.intreq_ch[8]	sDMAC1 channel 8 transfer end interrupt sDMAC1 channel 8 descriptor step end interrupt sDMAC1 channel 8 address error interrupt	71
INTSDMAC1CH9	SDMAC1.intreq_ch[9]	sDMAC1 channel 9 transfer end interrupt sDMAC1 channel 9 descriptor step end interrupt sDMAC1 channel 9 address error interrupt	72
INTSDMAC1CH10	SDMAC1.intreq_ch[10]	sDMAC1 channel 10 transfer end interrupt sDMAC1 channel 10 descriptor step end interrupt sDMAC1 channel 10 address error interrupt	73
INTSDMAC1CH11	SDMAC1.intreq_ch[11]	sDMAC1 channel 11 transfer end interrupt sDMAC1 channel 11 descriptor step end interrupt sDMAC1 channel 11 address error interrupt	74
INTSDMAC1CH12	SDMAC1.intreq_ch[12]	sDMAC1 channel 12 transfer end interrupt sDMAC1 channel 12 descriptor step end interrupt sDMAC1 channel 12 address error interrupt	75
INTSDMAC1CH13	SDMAC1.intreq_ch[13]	sDMAC1 channel 13 transfer end interrupt sDMAC1 channel 13 descriptor step end interrupt sDMAC1 channel 13 address error interrupt	76
INTSDMAC1CH14	SDMAC1.intreq_ch[14]	sDMAC1 channel 14 transfer end interrupt sDMAC1 channel 14 descriptor step end interrupt sDMAC1 channel 14 address error interrupt	77
NTSDMAC1CH15	SDMAC1.intreq_ch[15]	sDMAC1 channel 15 transfer end interrupt sDMAC1 channel 15 descriptor step end interrupt sDMAC1 channel 15 address error interrupt	78

The error notifications of this module are listed in the following table.

Table 7.6 Error Notifications

Error Notification	Description	ECM error Number	Error Response to bus master
DTS/sDMAC Transfer Error	DMA address error*1 (by INTSDMACERR interrupt)	182	—

Note: This error is generated due to the error response caused by some error.
For details of the error response caused by some error, see the column for Error Response to bus master in Error Notifications Table shown in each section.

Note 1. For details about causes of address errors, see **Section 7.5, (1) Address Errors**.

7.1.5 sDMAC Transfer Requests

The sDMAC transfer request table lists all sDMAC transfer request factors that can be selected by the DMAjRS_n registers and groups that can be selected by the DMACSELj_m registers. For details of List of DMA Trigger Sources, refer to Appendix file "sDMAC_Transfer_request_Table.xlsx"

7.1.6 Reset Sources

sDMAC reset sources are listed in the following table. The sDMAC is initialized by these reset sources.

Table 7.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
SDMACj	All registers	√	√	√	√	√	—	—
DMATRGSEL	All registers	√	√	√	√	√	—	—

7.2 Overview

The sDMAC can be used in place of the CPU to perform high-speed data transfer between on-chip memories.

7.2.1 Functional Overview

The sDMAC has the following features.

- 16 channels
- 4-Gbyte physical address space
  - Source and destination addresses are 32 bits wide
- Physical bus data width: 64 bits
- Transfer data length: 1 byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, 32 bytes, 64 bytes
- The maximum transfer size: 4,294,967,295 bytes (2^{32}-1 bytes)
- Parallel reads and writes (fly-by)
- Address mode: Dual address mode
- Transfer requests:
  Two request modes are available. Requests can be triggered by an auto request (a transfer request always occurs) or a peripheral hardware request.
- Bus mode:

  Each channel can be set to either normal speed mode or slow speed mode.
- Mode for arbitration between transfer channels

  The arbitration mode can be set to either fixed priority mode or round-robin mode.
- Interrupt requests:

  CPU interrupts from the sDMAC are triggered by three factors: termination of the descriptor step, termination of a data transfer and occurrence of an address error.
- Functions of the descriptor:

  A descriptor contains register settings (source address register, destination address register, transfer size register, transfer mode register and scatter gather registers) and can be chained to the next descriptor. The descriptor memory size is 4 KB. This descriptor memory is shared by all channels.
- Scatter-gather transfer:

  Inner and outer loops for transfers with inner and outer address increments are supported. Scattering at the destination interface and gathering at the source interface can be enabled independently and in parallel.
  - Inner and outer address increments are 32 bits wide
- The Data RAM and The Descriptor RAM are protected by ECC function. For details, see **Section 44, Functional Safety**.
- sDMAC has a data RAM as a pending transfer data buffer.
  The data RAM size is 2 KB and a DMA channel can use 64 bits × 16 lines for its own transactions.

7.2.2 Block Diagram

The following is a block diagram of sDMAC.

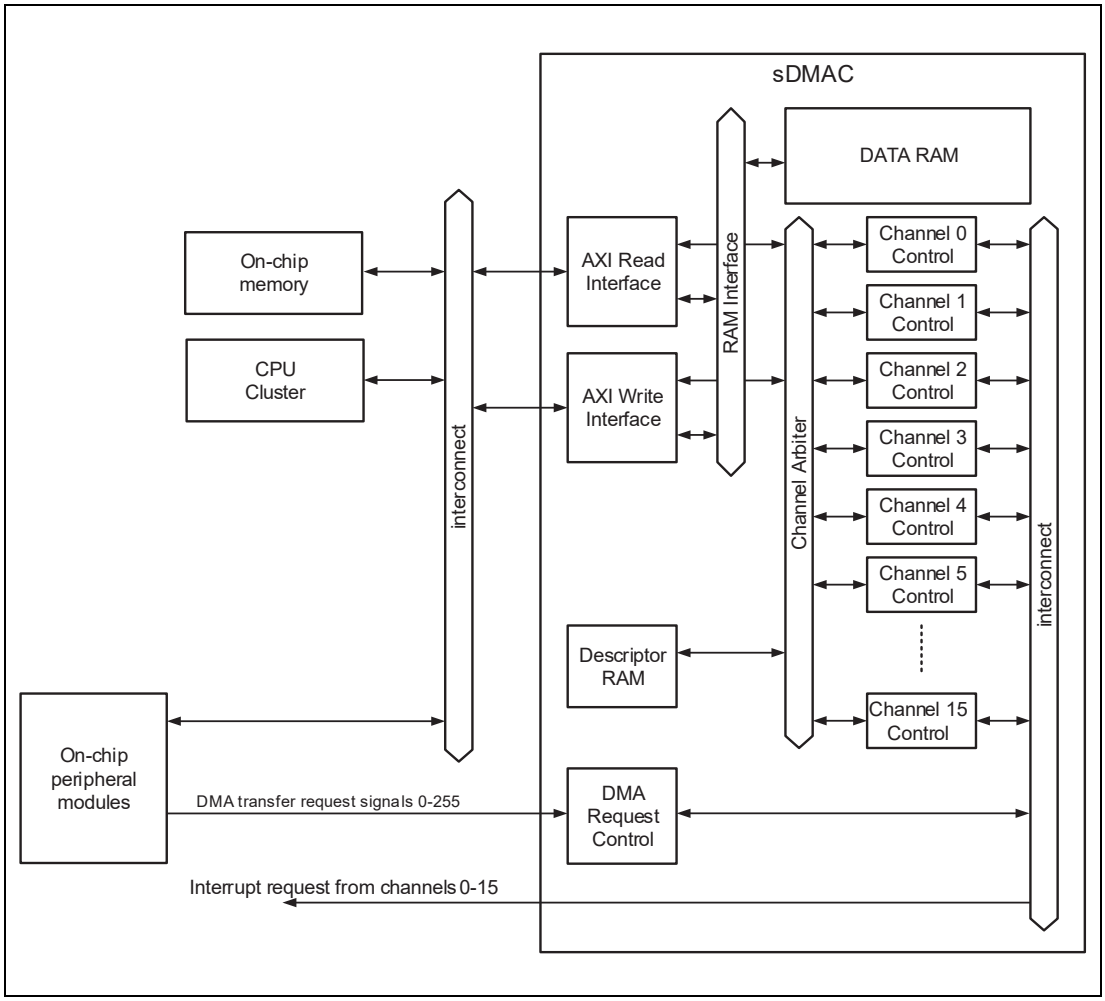


Figure 7.1 Block Diagram of sDMAC

The following block diagram shows sDMAC transfer request source selection by the DMACSELj_m registers.

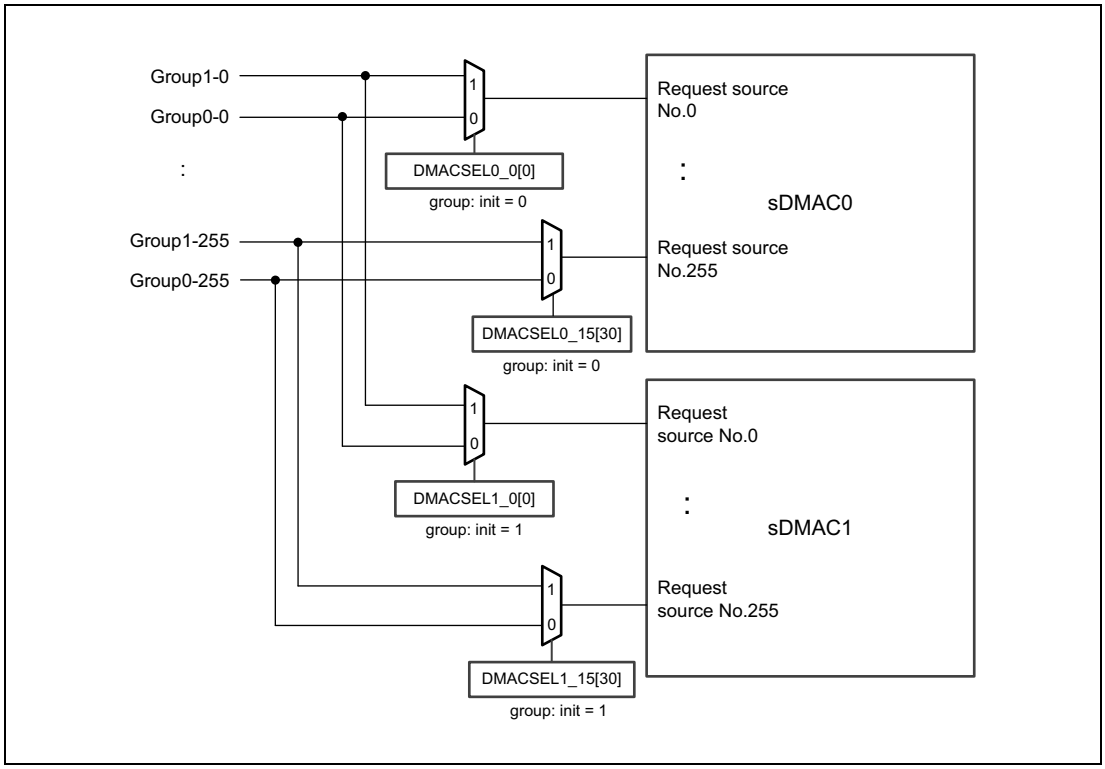


Figure 7.2 Block Diagram of sDMAC Transfer Request Selection

7.3 Registers of sDMAC

7.3.1 List of Registers

sDMAC registers are listed in the following table.

For details about <SDMACj_base>, see **Section 7.1.2, Register Base Addresses**.

Table 7.8 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					DMAj/PBG	Other
Global Registers						
SDMACj	DMA address error interrupt status register	DMAjESTA	<SDMACj_base> + 0010_H	32	DMAGPROT_GR	—
	DMA channel interrupt status register	DMAjISTA	<SDMACj_base> + 0020_H	32	DMAGPROT_GR	—
	DMA channel request priority register	DMAjCHPRI	<SDMACj_base> + 0040_H	32	DMAGPROT_GR	—
	DMA operation register	DMAjOR	<SDMACj_base> + 0060_H	16	DMAGPROT_GR	—
	DMA channel reset register	DMAjCHRST	<SDMACj_base> + 0080_H	32	DMAGPROT_GR	—
	DMA channel master setting register	DMAjCM_n	<SDMACj_base> + (n × 4_H) + 100_H	32	DMAGPROT_GR	—
Channel Registers						
SDMACj	DMA source address register	DMAjSAR_n	<SDMACj_base> + (n × 80_H) + 2000_H	32	DMAGPROT_n	—
	DMA destination address register	DMAjDAR_n	<SDMACj_base> + (n × 80_H) + 2004_H	32	DMAGPROT_n	—
	DMA transfer size register	DMAjTSR_n	<SDMACj_base> + (n × 80_H) + 2008_H	32	DMAGPROT_n	—
	DMA transfer size register B	DMAjTSRB_n	<SDMACj_base> + (n × 80_H) + $200C_H$	32	DMAGPROT_n	—
	DMA transfer mode register	DMAjTMR_n	<SDMACj_base> + (n × 80_H) + 2010_H	32	DMAGPROT_n	—
	DMA channel control register	DMAjCHCR_n	<SDMACj_base> + (n × 80_H) + 2014_H	16	DMAGPROT_n	—
	DMA channel suspend register	DMAjCHSTP_n	<SDMACj_base> + (n × 80_H) + 2016_H	16	DMAGPROT_n	—
	DMA channel status register	DMAjCHSTA_n	<SDMACj_base> + (n × 80_H) + 2018_H	32	DMAGPROT_n	—
	DMA channel flag clear register	DMAjCHFCR_n	<SDMACj_base> + (n × 80_H) + $201C_H$	32	DMAGPROT_n	—
	DMA gather inner address increment register	DMAjGIAI_n	<SDMACj_base> + (n × 80_H) + 2020_H	32	DMAGPROT_n	—
	DMA gather outer address increment register	DMAjGOAI_n	<SDMACj_base> + (n × 80_H) + 2024_H	32	DMAGPROT_n	—
	DMA scatter inner address increment register	DMAjSIAI_n	<SDMACj_base> + (n × 80_H) + 2028_H	32	DMAGPROT_n	—
	DMA scatter outer address increment register	DMAjSOAI_n	<SDMACj_base> + (n × 80_H) + $202C_H$	32	DMAGPROT_n	—
	DMA scatter gather status register	DMAjSGST_n	<SDMACj_base> + (n × 80_H) + 2038_H	32	DMAGPROT_n	—
	DMA scatter gather control register	DMAjSGCR_n	<SDMACj_base> + (n × 80_H) + $203C_H$	32	DMAGPROT_n	—

Table 7.8 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					DMAj/PBG	Other
SDMACj	DMA resource select register	DMAjRS_n	<SDMACj_base> + (n × 80_H) + 2040_H	32	DMAGPROT_n	—
	DMA buffer control registers	DMAjBUFCR_n	<SDMACj_base> + (n × 80_H) + 2048_H	32	DMAGPROT_n	—
	DMA descriptor pointer register	DMAjDPPTR_n	<SDMACj_base> + (n × 80_H) + 2050_H	32	DMAGPROT_n	—
	DMA descriptor control register	DMAjDPCR_n	<SDMACj_base> + (n × 80_H) + 2054_H	32	DMAGPROT_n	—
Descriptor memory						
SDMACj	Descriptor memory	Descriptor RAM	<SDMACj_base> + 4000_H to <SDMACj_base> + $4FFC_H$	32	DMAGPROT_DP	—
Transfer trigger selection						
DMATRGSEL	sDMACj transfer request group selection register m	DMACSELj_m	<DMATRGSEL_base> + (j × 40_H) + (m × 4_H)	8, 16, 32	PBG90#4	—

7.3.2 DMAjESTA — DMA Address Error Interrupt Status Register

DMAjESTA is a 32-bit read-only register that indicates the address error interrupt (INTSDMACERR) status of each channel. Causes of address errors are shown in **Section 7.5, (1) Address Errors**. Specification of the error notification routes is shown in **Section 7.4.2, DMA Interrupts**.

Access: DMAjESTA register is a read-only register that can be read in 32-bit units

Address: <SDMACj_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.9 DMAjESTA Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned
15 to 0	AEN[15:0]	Channel address error interrupt INTSDMACERR status These bits reflect the CAE bit status of the channels whose CAEE bit is set to 1. AEN[n]: 0: Channel n, address error does not occur 1: Channel n, address error occurs and generate address error interrupt INTSDMACERR

7.3.3 DMAjISTA — DMA Channel Interrupt Status Register

DMAjISTA is a 32-bit read-only register that indicates the interrupt signal status of each channel. Interrupt routes are shown in **Section 7.4.2, DMA Interrupts**.

Access: DMAjISTA register is a read-only register that can be read in 32-bit units

Address: <SDMACj_base> + 0020_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.10 DMAjISTA Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	INT[15:0]	Channel interrupt status INT[n]: 0: Channel n interrupt does not occur 1: Channel n Interrupt occurs

7.3.4 DMAjCHPRI — DMA Channel Request Priority Register

DMAjCHPRI is a 32-bit read/write register that specifies the channel request priority count timing.

See **Section 7.4.3, (3) Channel Request Priority** for details.

Access: DMAjCHPRI register can be read or written in 32-bit units

Address: <SDMACj_base> + 0040_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRICNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.11 DMAjCHPRI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PRICNT[15:0]	Channel Request Priority Count value This bit is used to control the timing of the channel request priority function.

7.3.5 DMAjOR — DMA Operation Register

DMAjOR is a 16-bit read/write register that controls master enable and specifies the priority level of all DMA channels.

Access: DMAjOR register can be read or written in 16-bit units

Address: <SDMACj_base> + 0060_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PR[1:0]		—	—	—	—	—	—	—	DME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 7.12 DMAjOR Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	PR[1:0]	Priority Mode Select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: CH0 > CH1 > ... > CH14 > CH15 11: Round-robin mode Other than above: Setting prohibited
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DME	DMA Master Enable Enables or disables DMA transfers on all channels. If the DME bit and the DE bit in DMAjCHCR_n are set to 1, DMA transfer is enabled. At this time, the TE bits in DMAjCHSTA_n and the CAE bits in DMAjCHSTA_n must be all 0. If this bit is cleared during transfer, transfers on all channels are terminated. 0: Disable DMA transfers on all channels 1: Enable DMA transfers on all channels

7.3.6 DMAjCHRST — DMA Channel Reset Register

DMAjCHRST register is a 32-bit write-only register that initializes each channel.

When this register is set, the specified channel state is initialized completely and the following registers are initialized.

DMAjSAR_n, DMAjDAR_n, DMAjTSR_n, DMAjTSRB_n, DMAjTMR_n, DMAjCHCR_n, DMAjCHSTP_n, DMAjCHSTA_n, DMAjGIAI_n, DMAjGOAI_n, DMAjSIAI_n, DMAjSOAI_n, DMAjSGCR_n, DMAjRS_n, DMAjBUFCR_n, DMAjDPPTR_n, DMAjDPCR_n

Access: DMAjCHRST register is a write-only register that can be written in 32-bit units

Address: <SDMACj_base> + 0080_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 7.13 DMAjCHRST Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15 to 0	CLR[15:0]	All registers in the channel can be cleared by writing to these bits. CLR[0] 0: Ignored 1: Channel 0 registers are cleared CLR[1] 0: Ignored 1: Channel 1 registers are cleared … CLR[15] 0: Ignored 1: Channel 15 registers are cleared Before writing to this register, confirm that channel BUSY is 0. These bits are always read as 0.

7.3.7 DMAjCM_n — DMA Channel Master Setting Register n

DMAjCM_n is a 32-bit read/write register that configures the DMA master information.

Access: DMAjCM_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 4_H) + 100_H

Value after reset: DMA0CM_n: 0000 $1C00_H$
DMA1CM_n: 0000 $1B00_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SPID[4:0]					—	—	—	—	—	—	—	UM
Value after reset	0	0	0	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Note 1. This value is $1C_H$ (sDMAC0) or $1B_H$ (sDMAC1).

Table 7.14 DMAjCM_n Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	SPID[4:0]	Channel master SPID setting Specifies the SPID information of the master assigned to the channel. Value after reset of DMA0CM_n: $1C_H$ Value after reset of DMA1CM_n: $1B_H$
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	UM	Channel master UM setting Specifies the UM information of the master assigned to the channel. 0: Supervisor mode 1: User mode

7.3.8 DMAjSAR_n — DMA Source Address Register n

DMAjSAR_n is a 32-bit read/write register that specifies the source address of DMA transfer. During DMA transfer, this register indicates the next source address.

The source address register must be specified based on a transaction size boundary which is set to STS bit in DMAjTMR_n register regardless of the source address mode which is set to SM bit in DMAjTMR_n register.

Access: DMAjSAR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SAR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.9 DMAjDAR_n — DMA Destination Address Register n

DMAjDAR_n is a 32-bit read/write register that specifies the destination address of DMA transfer. During DMA transfer, this register indicates the next destination address.

When the selected destination address mode is “increment” (DM bit in DMAjTMR_n register is 01_B), the destination address can be specified on a byte boundary. Otherwise the destination address must be specified on the transaction size boundary (DTS bit in DMAjTMR_n register).

Access: DMAjDAR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2004_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.10 DMAjTSR_n — DMA Transfer Size Register n

DMAjTSR_n is a 32-bit read/write register that specifies the DMA transfer size. The total size of DMA transfer is 1 byte when the setting is 0000 0001_H and 4,294,967,295 bytes (2^{32}-1 bytes) when FFFF $FFFF_H$ is set. When the setting is 0000 0000_H, transfer does not occur, and TE is set immediately. During DMA transfer, DMAjTSR_n indicates the remaining transfer size.

In DMA transfer, the size of the read transfer and write transfer may differ. This is because the DMAC uses flyby DMA transfer via the data RAM. DMAjTSR_n indicates the read transfer size.

Access: DMAjTSR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.11 DMAjTSRB_n — DMA Transfer Size Register B n

DMAjTSRB_n is a 32-bit read-only register that indicates the DMA transfer size. During DMA transfer, this register indicates the remaining transfer size (in bytes). When DMAjTSR_n is written, the same value is set in this register.

In DMA transfer, the size of the read transfer and write transfer may differ. This is because the DMAC uses flyby DMA transfer via data RAM. DMAjTSRB_n indicates the write transfer size. DMAjTSR_n value is also set to DMAjTSRB_n register when DMAjTSR_n value is updated by Descriptor function.

Access: DMAjTSRB_n register is a read-only register that can be read in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + $200C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSRW[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSRW[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

7.3.12 DMAjTMR_n — DMA Transfer Mode Register n

DMAjTMR_n is a 32-bit read/write register that specifies the DMA transfer mode.

Access: DMAjTMR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2010_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SLM[3:0]				PRI[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TRS	DM[1:0]		SM[1:0]		DTS[3:0]				STS[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.15 DMAjTMR_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	SLM[3:0]	DMA Transfer Slow Speed mode Specifies the number of clock cycles (bus clock) between two DMA transfers. DMA transfer is executed once every number of clock cycles specified by SLM. 0000: Normal mode 1000: 256 clock cycles 1001: 512 clock cycles 1010: 1024 clock cycles : 1111: 32768 clock cycles Other than above: Setting prohibited
19 to 16	PRI[3:0]	Channel Request Priority Setting These bits set the priority of channel requests. Refer to **Section 7.4.3 (3), Channel Request Priority** about detail. PRI[3]: 0: Channel request priority is disabled. 1: Channel request priority is enabled. PRI[2:0] 111: Highest priority : 000: Lowest priority
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TRS	Transfer request source Specifies which transfer request source will be sent to the DMAC. Before changing transfer request sources, the DMA enable bit (DE) should be set to 0. 0: Auto request 1: Hardware request

Table 7.15 DMAjTMR_n Register Contents (2/2)

Bit Position	Bit Name	Function
11, 10	DM[1:0]	Destination address mode Specify whether the DMA destination address is incremented or fixed, and the amount of increment. 00: Fixed destination address 01: Destination address is incremented based on destination transaction size (DTS) +1 in 1-byte unit transfers +2 in 2-byte unit transfers +4 in 4-byte unit transfers +8 in 8-byte unit transfers +16 in 16-byte unit transfers +32 in 32-byte unit transfers +64 in 64-byte unit transfers Other than above: Setting prohibited
9, 8	SM[1:0]	Source Address Mode Specify whether the DMA source address is incremented or fixed, and the amount of increment. 00: Fixed source address 01: Source address is incremented based on source transaction size (STS) +1 in 1-byte unit transfers +2 in 2-byte unit transfers +4 in 4-byte unit transfers +8 in 8-byte unit transfers +16 in 16-byte unit transfers +32 in 32-byte unit transfers +64 in 64-byte unit transfers Other than above: Setting prohibited
7 to 4	DTS[3:0]	DMA destination transaction size 0000: 1-byte unit transfer 0001: 2-byte unit transfer 0010: 4-byte unit transfer 0011: 8-byte unit transfer 0100: 16-byte unit transfer 0101: 32-byte unit transfer 0110: 64-byte unit transfer Other than above: Setting prohibited
3 to 0	STS[3:0]	DMA source transaction size 0000: 1-byte unit transfer 0001: 2-byte unit transfer 0010: 4-byte unit transfer 0011: 8-byte unit transfer 0100: 16-byte unit transfer 0101: 32-byte unit transfer 0110: 64-byte unit transfer Other than above: Setting prohibited

7.3.13 DMAjCHCR_n — DMA Channel Control Register n

DMAjCHCR_n is a 16-bit read/write register that controls DMA transfer.

Access: DMAjCHCR_n register can be read or written in 16-bit units

Address: <SDMACj_base> + (n × 80_H) + 2014_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DPE	DPB	—	—	—	CAEE	CAIE	DSIE	IE	DE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 7.16 DMAjCHCR_n Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	DPE	Descriptor enable bit Specifies whether the descriptor is enabled. 0: Disabled Descriptor operation is disabled. 1: Enabled Descriptor operation is enabled. When the DMAjDPPTR_n.CF bit is set to 1, after reading the register setting from the descriptor memory, DMA transfer is continued. Clearing the DPE bit by setting the DMAjCHFCR_n.DPEC bit to 1 terminates the descriptor chain operation.
8	DPB	Descriptor start bit Specifies the configuration to be loaded at the beginning of the descriptor. After reading the descriptor, this bit is cleared. While DE = 1, DPB must not be set to 1. 0: Start the DMA transfer via a register setting. 1: Start DMA transfer after the channel configuration is copied from the descriptor memory. The configurations to copy from the Descriptor memory are specified in the DMAjDPCR_n register.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CAEE	Channel address error interrupt INTSDMACERR enable Specifies whether a channel can generate an interrupt request upon the occurrence of an address error. When the CAEE bit and the DMAjCHSTA_n.CAE bit are set to 1, a channel (sDMAC_CH0 to sDMAC_CH15) will generate an interrupt request (INTSDMACERR) upon the occurrence of an address error. For details, see **Section 7.4.2, DMA Interrupts**. 0: Interrupt request disabled 1: Interrupt request enabled

Table 7.16 DMAjCHCR_n Register Contents (2/2)

Bit Position	Bit Name	Function
3	CAIE	Channel address error interrupt INTSDMACjCHn enable Specifies whether a channel can generate an interrupt request upon occurrence of an address error. When the CAIE bit and DMAjCHSTA_n.CAE bit are set to 1, a channel will generate an interrupt request (INTSDMACjCHn) upon the occurrence of an address error. For details, see **Section 7.4.2, DMA Interrupts**. 0: Interrupt request disabled 1: Interrupt request enabled
2	DSIE	Descriptor step end interrupt master enable Specifies whether a channel can generate an interrupt at the end of a descriptor step. When the DMAjDPPTR_n.DIE bit and DSIE bit are set to 1, a channel will generate an interrupt request (INTSDMACjCHn) upon completion of a descriptor step. For details, see **Section 7.4.2, DMA Interrupts**. 0: Interrupt request is disabled 1: Interrupt request is enabled
1	IE	Transfer end interrupt enable Specifies whether or not a channel can generate an interrupt at the end of DMA transfer. Setting this bit to 1 will allow a channel to generate an interrupt request (INTSDMACjCHn) when the DMAjCHSTA_n.TE bit is set to 1. For details, see **Section 7.4.2, DMA Interrupts**. 0: Interrupt request is disabled 1: Interrupt request is enabled
0	DE	DMA Enable Enables or disables DMA transfer for the corresponding channel. Clearing the DE bit to 0 by setting the DMAjCHFCR_n.DEC bit to 1 aborts DMA transfer. The DE bit is also cleared automatically when an ongoing DMA transfer is finished or an ECC error of the descriptor RAM occurs. See **Section 7.5, Usage Notes** for details about DMA transfer abort. 0: DMA transfer disabled 1: DMA transfer enabled

7.3.14 DMAjCHSTP_n — DMA Channel Suspend Register

DMAjCHSTP_n is a 16-bit read/write register that controls the suspend condition for DMA transfer.

Access: DMAjCHSTP_n register can be read or written in 16-bit units

Address: <SDMACj_base> + (n × 80_H) + 2016_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 7.17 DMAjCHSTP_n Register Contents

Bit Position	Bit Name	Function
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STP	DMA transmission suspend When this bit is set to 1, packet requests suspend immediately. When this bit is set to 0 after transmission has been suspended, the transmission is resumed. 0: No operation 1: DMA transfer suspend

7.3.15 DMAjCHSTA_n — DMA Channel Status Register n

DMAjCHSTA_n register is a 32-bit read-only register that shows the DMA transfer status.

Access: DMAjCHSTA_n register is a read-only register that can be read in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2018_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVF	DRQ	—	—	—	—	—	—	—	—	CAE	DSE	TE	BUSY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.18 DMAjCHSTA_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	OVF	Hardware transfer request overflow flag Indicates that a hardware trigger was asserted before a pending hardware transfer request has been acknowledged. To clear the OVF bit, write 1 to the DMAjCHFCR_n.OVFC bit. 0: No hardware transfer request overflow 1: Hardware transfer request overflow occurred
12	DRQ	Hardware request status If this bit is set, it means that a hardware transfer request exists or is retained. • This bit shows whether a hardware transfer request is retained. When the transfer specified by DMAjRS_n.TL bits is completed, this bit is automatically cleared. This bit can be cleared by writing 1 to the DMAjCHFCR_n.DRQC bit. This bit changes regardless of the value of the DMAjTMR_n.TRS bit when a hardware DMA transfer request is generated. 0: There is no hardware transfer request 1: There is a hardware transfer request
11 to 4	Reserved	When read, the value after reset is returned.
3	CAE	Address error flag Indicates that an address error interrupt occurred in the DMA transfer. The following conditions can set this bit: The transfer source or transfer destination is an invalid space. A GUARD error occurred upon access to a source or a destination address. Channel register setting is prohibited. See **Section 7.5, (1) Address Errors** for details about address errors. If this bit is set, DMA transfer is not enabled for the channel even if the DE bit is set to 1. To clear the CAE bit, write 1 to the DMAjCHFCR_n.CAEC bit. 0: Address error interrupt did not occur in the DMA transfer 1: Address error interrupt occurred in the DMA transfer

Table 7.18 DMAjCHSTA_n Register Contents (2/2)

Bit Position	Bit Name	Function
2	DSE	Descriptor step end flag When the DMAjCHCR_n.DSIE bit is set to 1 and the DMAjDPPTR_n.DIE bit is set to 1, the DSE bit is set to 1 at the end of DMA transfer. To clear the DSE bit, write 1 to the DMAjCHFCR_n.DSEC bit. 0: DMA transfer is in progress or has finished 1: The descriptor step has finished
1	TE	Transfer end flag The TE bit is set to 1 when data transfer ends (DMAjTSR_n becomes 0). When the descriptor is enabled, the TE bit is set to 1 at the end of all descriptor steps. The TE bit is not set to 1 in the following cases: - DMA transfer ends due to a DMA address error before DMAjTSR_n becomes 0. - DMA transfer is terminated by clearing the DE bit and DME bit in DMAjOR. To clear the TE bit, write 1 to the DMAjCHFCR_n.TEC bit. If the DMAjCHCR_n.DE bit is set to 1 while TE bit is set to 1, DMA transfer will not be enabled. 0: DMA transfer is in progress or DMA transfer has finished 1: DMA transfer ended at the specified count (TSR = 0)
0	BUSY	Channel Busy flag This bit indicates that the channel is busy. When transfer is started, the BUSY bit is set to 1. After the DMAjCHCR_n.DE bit is cleared or the DMAjCHSTP_n.STP bit is set, the BUSY bit is cleared after receiving the response of all packets. 0: Channel status is idle 1: Channel status is busy

7.3.16 DMAjCHFCR_n — DMA Channel Flag Clear Register n

DMAjCHFCR_n is a 32-bit write-only register that clears the DMA channel status flags.

Access: DMAjCHFCR_n register is a write-only register that can be written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + $201C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OVFC	DRQC	—	—	DPEC	—	—	—	—	—	CAEC	DSEC	TEC	DEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	R	R	W	R	R	R	R	R	W	W	W	W

Table 7.19 DMAjCHFCR_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When writing, write the value after reset.
13	OVFC	Hardware transfer request overflow flag clear This bit is always read as 0. When the OVFC bit is set to 1, the DMAjCHSTA_n.OVF bit is cleared. 0: No operation 1: DMAjCHSTA_n.OVF bit is cleared.
12	DRQC	Hardware transfer request clear This bit is always read as 0. When the DRQC bit is set to 1, the DMAjCHSTA_n.DRQ bit is cleared. 0: No operation 1: DMAjCHSTA_n.DRQ bit is cleared.
11 to 10	Reserved	When writing, write the value after reset.
9	DPEC	Descriptor enable clear This bit is always read as 0. When the DPEC bit is set to 1, the DMAjCHCR_n.DPE bit is cleared. The DMAjCHCR_n.DPE bit is the enable condition of the descriptor operation. To abort DMA transfer at the end of a descriptor step, clear the DPE bit by using this bit. 0: No operation 1: DMAjCHCR_n.DPE bit is cleared
8 to 4	Reserved	When writing, write the value after reset.
3	CAEC	Address error flag clear This bit is always read as 0. When the CAEC bit is set to 1, the DMAjCHSTA_n.CAE bit is cleared. The DMAjCHSTA_n.CAE bit is the stop condition of DMA transfer. To end DMA transfer, clear the DMAjCHCR_n.DE bit before clearing the DMAjCHSTA_n.CAE bit. 0: No operation 1: DMAjCHSTA_n.CAE bit is cleared
2	DSEC	Descriptor step end flag clear This bit is always read as 0. When the DSEC bit is set to 1, the DMAjCHSTA_n.DSE bit is cleared. 0: No operation 1: DMAjCHSTA_n.DSE bit is cleared

Table 7.19 DMAjCHFCR_n Register Contents (2/2)

Bit Position	Bit Name	Function
1	TEC	Transfer end flag clear This bit is always read as 0. When the TEC bit is set to 1, the DMAjCHSTA_n.TE bit is cleared. The DMAjCHSTA_n.TE bit is the stop condition of DMA transfer. To end DMA transfer, clear the DMAjCHCR_n.DE bit before clearing the DMAjCHSTA_n.TE bit. 0: No operation 1: DMAjCHSTA_n.TE bit is cleared
0	DEC	DMA enable clear This bit is always read as 0. When the DEC bit is set to 1, the DMAjCHCR_n.DE bit is cleared. The DMAjCHCR_n.DE bit is the enable condition of DMA transfer. To abort DMA transfer, clear the DE bit by using this bit. See **Section 7.5, Usage Notes** for details about DMA transfer abort. 0: No operation 1: DMAjCHCR_n.DE bit is cleared

7.3.17 DMAjGIAI_n — DMA Gather Inner Address Increment Register n

DMAjGIAI_n is a 32-bit read/write register that specifies the source address increment for the inner loop of a gather transfer.

Access: DMAjGIAI_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2020_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GIAI[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GIAI[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

This register must be set to a multiple of DMA source transaction size which is set to STS bit in the DMAjTMR_n register.

7.3.18 DMAjGOAI_n — DMA Gather Outer Address Increment Register n

DMAjGOAI_n is a 32-bit read/write register that specifies the source address increment for the outer loop of a gather transfer.

Access: DMAjGOAI_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2024_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GOAI[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GOAI[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

This register must be set to a multiple of DMA source transaction size which is set to STS bit in the DMAjTMR_n register.

7.3.19 DMAjSIAI_n — DMA Scatter Inner Address Increment Register n

DMAjSIAI_n is a 32-bit read/write register that specifies the destination address increment for the inner loop of a scatter transfer.

Access: DMAjSIAI_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2028_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SIAI[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIAI[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.20 DMAjSOAI_n — DMA Scatter Outer Address Increment Register n

DMAjSOAI_n is a 32-bit read/write register that specifies the destination address increment for the outer loop of a scatter transfer.

Access: DMAjSOAI_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + $202C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOAI[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOAI[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.3.21 DMAjSGST_n — DMA Scatter Gather Status Register n

DMAjSGST_n is a 32-bit read only register that indicates the count of inner scatter and gather loops. The detailed operation of scatter/gather is explained in **Section 7.4.5, Scatter Gather Transfer**.

Access: DMAjSGST_n register is a read-only register that can be read in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2038_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SICNT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	GICNT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.20 DMAjSGST_n Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 16	SICNT[13:0]	Indicates the current repeat count of the scatter inner DMA loop.
15, 14	Reserved	When read, the value after reset is returned.
13 to 0	GICNT[13:0]	Indicates the current repeat count of the gather inner DMA loop.

7.3.22 DMAjSGCR_n — DMA Scatter Gather Control Register n

DMAjSGCR_n is a 32-bit read/write register that controls scattering on the source side and gathering on the destination side. The operation of scatter/gather is explained in detail in **Section 7.4.5, Scatter Gather Transfer**.

Access: DMAjSGCR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + ($n \times 80_H$) + $203C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEN	ZF	SIRPT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GEN	—	GIRPT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.21 DMAjSGCR_n Register Contents

Bit Position	Bit Name	Function
31	SEN	Scatter enable on source side 0: Scattering is disabled 1: Scattering is enabled
30	ZF	Zero fill 0: Zero fill is disabled 1: Zero fill is enabled
29 to 16	SIRPT[13:0]	Repeat count of the inner DMA loop if scattering is enabled 0: Zero repeats 1: One repeat : 16383: 16383 repeats
15	GEN	Gather enable on destination side 0: Gathering is disabled 1: Gathering is enabled
14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	GIRPT[13:0]	Repeat count of the inner DMA loop if gathering is enabled 0: Zero repeats 1: One repeat : 16383: 16383 repeats

7.3.23 DMAjRS_n — DMA Resource Select Register n

DMAjRS_n is a 32-bit read/write register that specifies the source and control settings for hardware transfer requests. The operation of hardware transfer requests is explained in detail in **Section 7.4.1, DMA Transfer Requests**.

Access: DMAjRS_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2040_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TL[2:0]			FPT	PLE	DRQI	—	RS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.22 DMAjRS_n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	TC[15:0]	Transfer count per hardware request Specify the number of DMA transfers in hardware request mode. This bit is valid only when the DMAjRS_n.TL bit is set to 000 or 001. In this case, setting 0 is prohibited.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 12	TL[2:0]	Transfer limit per hardware request Specifies the limit of transfers generated by a hardware request. 000: Transaction size indicated by DMAjTMR_n.STS * DMAjRS_n.TC (Setting prohibited when PLE = 1) 001: Transaction size indicated by DMAjTMR_n.DTS * DMAjRS_n.TC (Setting prohibited when PLE = 0) 010: Transfer size indicated by DMAjTSR_n.TSR 011: Until the DSE flag is asserted. (Transfer also ends when the TE flag is asserted.) 100: Until the TE flag is asserted Other than above: Setting prohibited
11	FPT	First pre-load trigger Specifies the trigger to start the first pre-load transfer. If transfer data is not prepared before DE is set, this bit should be set to 1. When the pre-load function is disabled (PLE = 0), this setting is ignored. 0: First pre-load start when DE is set to 1 1: First pre-load start when hardware request is asserted
10	PLE	Pre-load enable When the pre-load function is enabled, the DMAC loads transfer data from the source memory before a hardware request is asserted. 0: Disable 1: Enable

Table 7.22 DMAjRS_n Register Contents (2/2)

Bit Position	Bit Name	Function
9	DRQI	DMA request initialization when descriptor settings are loaded Specifies whether to initialize the DRQ bits when this register setting is loaded from the descriptor memory. When DRQI is set to 1, the remaining DRQ bits are cleared. The DRQ bits are also cleared when a different source (RS) is loaded. For details, see **Section 7.4.6 (4) (h), Example 8: Hardware Request Operation with Different Request Source Settings**. 0: DRQ initialize disabled 1: DRQ initialize enabled
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	RS[7:0]	DMA request source See the DMA request resources of on-chip peripheral modules.

7.3.24 DMAjBUFCR_n — DMA Buffer Control Register n

DMAjBUFCR_n is a 32-bit read/write register that controls the upper limit of the data RAM buffer. This register specifies the upper limit of the RAM buffer available to a channel for data prefetching. This setting should not be changed unless it is necessary to limit the amount of prefetched data.

Access: DMAjBUFCR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2048_H

Value after reset: 0000 0080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ULB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.23 DMAjBUFCR_n Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ULB[7:0]	Upper limit of buffer This register controls the upper limit of a channel's data buffer. A power-of-two setting is recommended. Setting ULB to less than the transaction size (DMAjTMR_n.STS, DMAjTMR_n.DTS) is prohibited. The maximum value is 128 (bytes).

7.3.25 DMAjDPPTR_n — DMA Descriptor Pointer Register n

DMAjDPPTR_n is a 32-bit read/write register that specifies the address of the next register setting in the descriptor memory. The operation of descriptors is explained in detail in **Section 7.4.6, Descriptors**.

Access: DMAjDPPTR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + (n × 80_H) + 2050_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PTR[11:2]										DIE	CF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.24 DMAjDPPTR_n Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	PTR[11:2]	Address pointer of Descriptor These bits indicate the address of the next register setting in the descriptor memory.
1	DIE	Descriptor interrupt enable When the DMAjCHCR_n.DSIE bit and this bit are set to 1 at the end of the current DMA transfer, the DSE bit in DMAjCHSTA_n set to 1 and the descriptor step end interrupt is issued. 0: Interrupt request is disabled 1: Interrupt request is enabled
0	CF	Continuation flag of descriptor 0: Disable Transfer is not continued. The TE bit in the DMAjCHCR_n register is set to 1 at the termination of the current DMA transfer. 1: Enable When DMA transfer with the current register settings is terminated and DMAjCHCR_n.DPE is set to 1, transfer continues to read the register settings from the DPTR address of the descriptor memory.

7.3.26 DMAjDPCR_n — DMA Descriptor Control Register n

DMAjDPCR_n is a 32-bit read/write register that specifies which channel settings will be updated with the contents of the descriptor memory. The operation of descriptors is explained in details in **Section 7.4.6, Descriptors**.

Access: DMAjDPCR_n register can be read or written in 32-bit units

Address: <SDMACj_base> + ($n \times 80_H$) + 2054_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	UPF[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.25 DMAjDPCR_n Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	UPF[10:0]	Update flag of Descriptor These bits specify the registers to be updated by a descriptor operation. The DMAjDPPTR_n register is always updated. UPF[0]: Enable/disable updating the DMAjSAR_n register UPF[1]: Enable/disable updating the DMAjDAR_n register UPF[2]: Enable/disable updating the DMAjTSR_n register UPF[3]: Enable/disable updating the DMAjTMR_n register UPF[4]: Enable/disable updating the DMAjGIAI_n register UPF[5]: Enable/disable updating the DMAjGOAI_n register UPF[6]: Enable/disable updating the DMAjSIAI_n register UPF[7]: Enable/disable updating the DMAjSOAI_n register UPF[8]: Enable/disable updating the DMAjSGCR_n register UPF[9]: Enable/disable updating the DMAjRS_n register UPF[10]: Enable/disable updating the DMAjBUFCR_n register

7.3.27 Descriptor RAM — Descriptor Memory

See **Section 7.4.6, Descriptors** for details about the descriptor memory.

7.3.28 DMACSELj_m — sDMACj Transfer Request Group Selection Register m (m = 0 to 15)

DMACSELj_m register specifies the sDMACj transfer request group for each channel.

Access: DMACSELj_m register can be read or written in 8-, 16-, or 32-bit units

Address: DMACSEL0_m: <DMATRGSEL_base> + (m × 4_H)
DMACSEL1_m: <DMATRGSEL_base> + (m × 4_H) + 40_H

Value after reset: DMACSEL0_m: 0000 0000_H
DMACSEL1_m: 5555 5555_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SEL15	—	SEL14	—	SEL13	—	SEL12	—	SEL11	—	SEL10	—	SEL9	—	SEL8
Value after reset	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SEL7	—	SEL6	—	SEL5	—	SEL4	—	SEL3	—	SEL2	—	SEL1	—	SEL0
Value after reset	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1	0	*1
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Note 1. This value is 0_B (sDMAC0), 1_B (sDMAC1).

Table 7.26 DMACSELj_m Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	SEL15	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 15
29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	SEL14	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 14
27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	SEL13	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 13
25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	SEL12	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 12
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	SEL11	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 11
21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	SEL10	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 10
19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	SEL9	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 9

Table 7.26 DMACSELj_m Register Contents (2/2)

Bit Position	Bit Name	Function
17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	SEL8	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 8
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	SEL7	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 7
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SEL6	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 6
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	SEL5	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 5
9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SEL4	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 4
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SEL3	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 3
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SEL2	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 2
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	SEL1	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 1
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEL0	Selects the DMA transfer request group for sDMACj transfer request signal 16 x m + 0

The following table shows the DMACSELj_0 to DMACSELj_15 register contents.

Bit Name	Function
SELn	0_B: Select DMA transfer request group 0 1_B: Select DMA transfer request group 1

7.4 Operation

7.4.1 DMA Transfer Requests

The DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request source (TRS) bit in the DMA transfer mode register (DMAjTMR_n) determines whether a hardware DMA transfer request or a software DMA transfer request is used. How to set up DMA transfer requests is described in **Section 7.7, Setting up DMA Transfer**.

(1) Auto-Request Mode

When the TRS bit in the DMAjTMR_n register is set to 0, a software DMA transfer request can be generated. When the DE bit in DMAjCHCR_n and the DME bit in DMAjOR are set to 1 for the target channel, the transfer begins so long as the CAE bit in DMAjCHSTA_n is 0.

(2) Hardware-Request Mode

When the TRS bit in the DMAjTMR_n register is set to 1, if DMA transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0), a transfer is performed upon the input of a transfer request signal.

In the case of a hardware DMA transfer request for the DMAC, one out of 256 hardware DMA transfer sources is selected and assigned to each channel of the DMAC in the DMA request control module. This assignment is configured by the resource select (RS) bit in the DMAjRS_n register.

Examples of using hardware request mode are shown below.

(a) Example 1: Hardware Request Operation to Transfer Data from a Peripheral Module

Figure 7.3 and **Figure 7.4** show examples of a peripheral module requesting sDMAC transfer from peripheral memory to on-chip memory. When the hardware transfer request triggers a load data transfer (DMAjRS_n.PLE = 0), the DMAjRS_n.TL bit determines the load data transfer limit per hardware transfer request. Store data transfers are executed in auto request mode in this case.

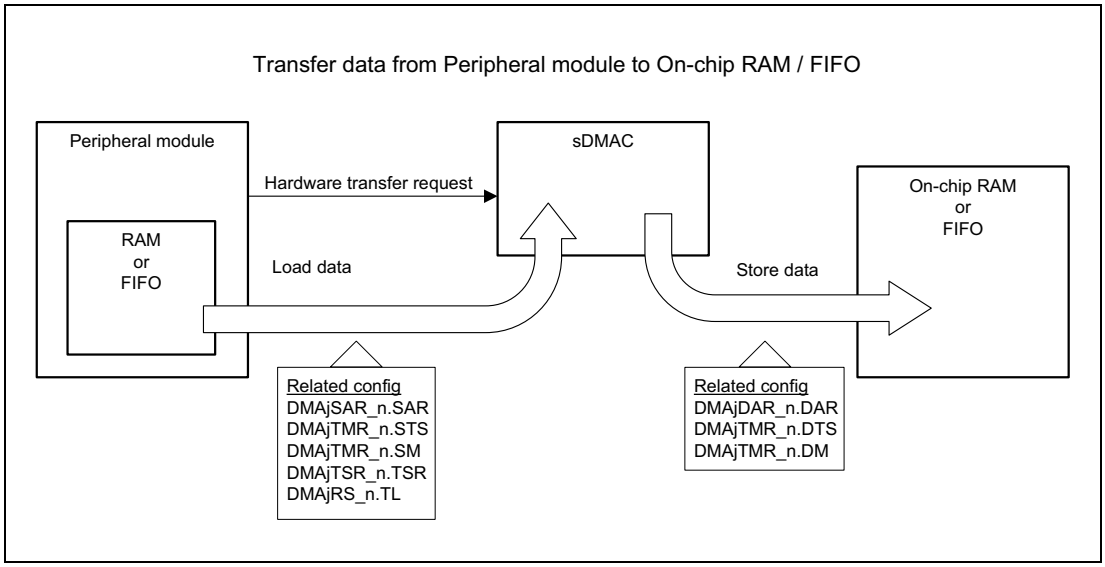


Figure 7.3 Transferring Data from Peripheral Module in Hardware-Request Mode

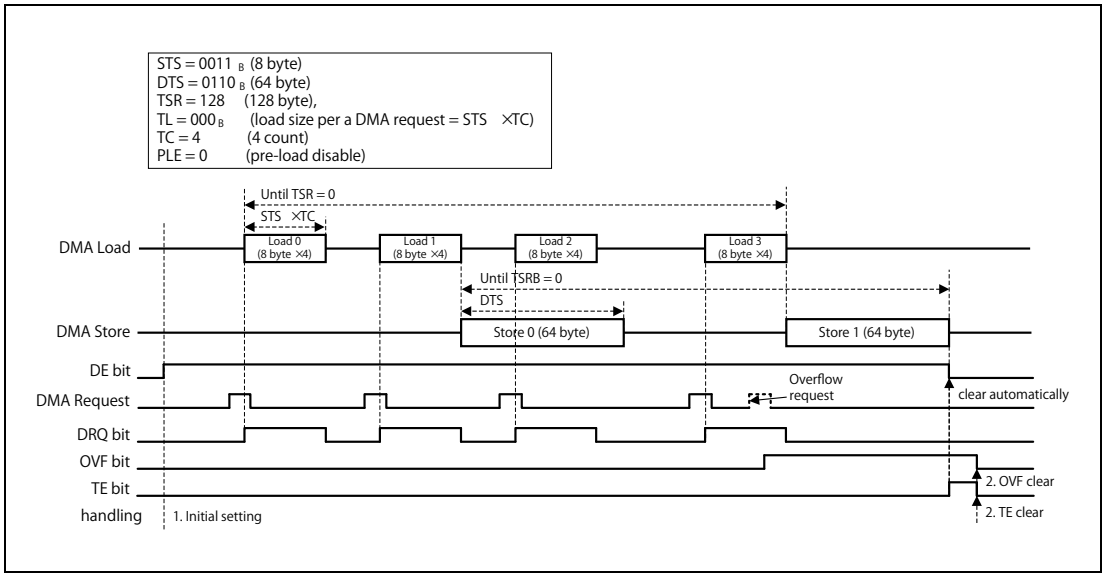


Figure 7.4 Timing Chart for Transferring Data from Peripheral Module

(b) Example 2: Hardware Request Operation to Transfer Data to a Peripheral Module

Figure 7.5, **Figure 7.6**, **Figure 7.7**, and **Figure 7.8** show examples of a peripheral module requesting sDMAC transfer from on-chip memory to peripheral module memory. When the pre-load function is enabled (DMAjRS_n.PLE = 1), a hardware transfer request triggers a store data transfer. The DMAjRS_n.TL bit determines the store data transfer limit per hardware transfer request. Load data transfers are executed in auto request mode in this case.

In case the pre-load transfer is triggered before the load data is prepared in the source memory, the pre-load limit can be controlled by setting the ULB bit in DMAjBUFCR_n register as shown in **Figure 7.8**.

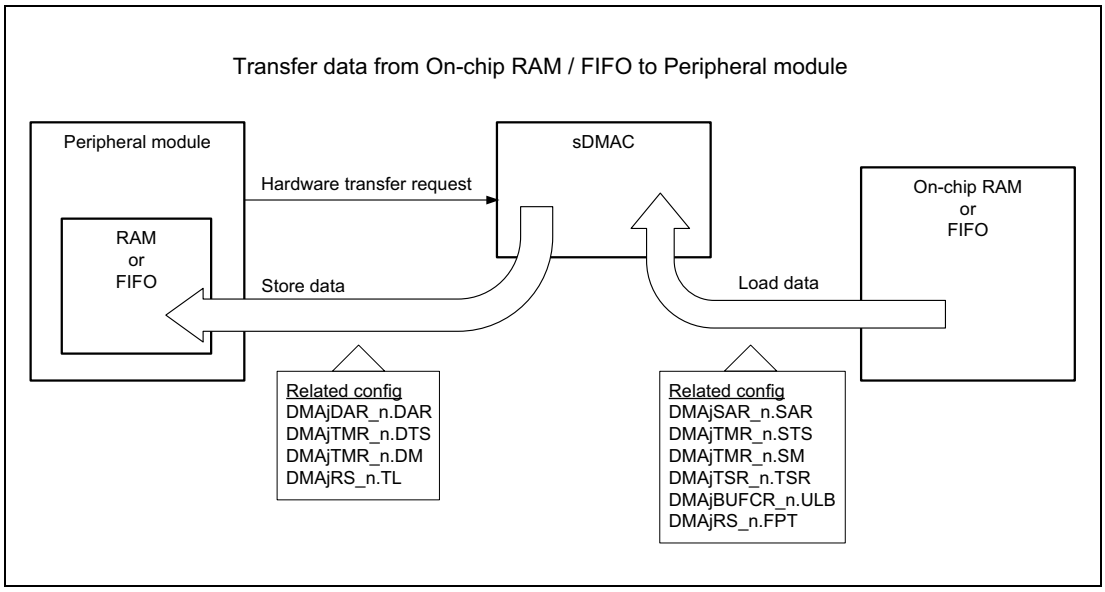
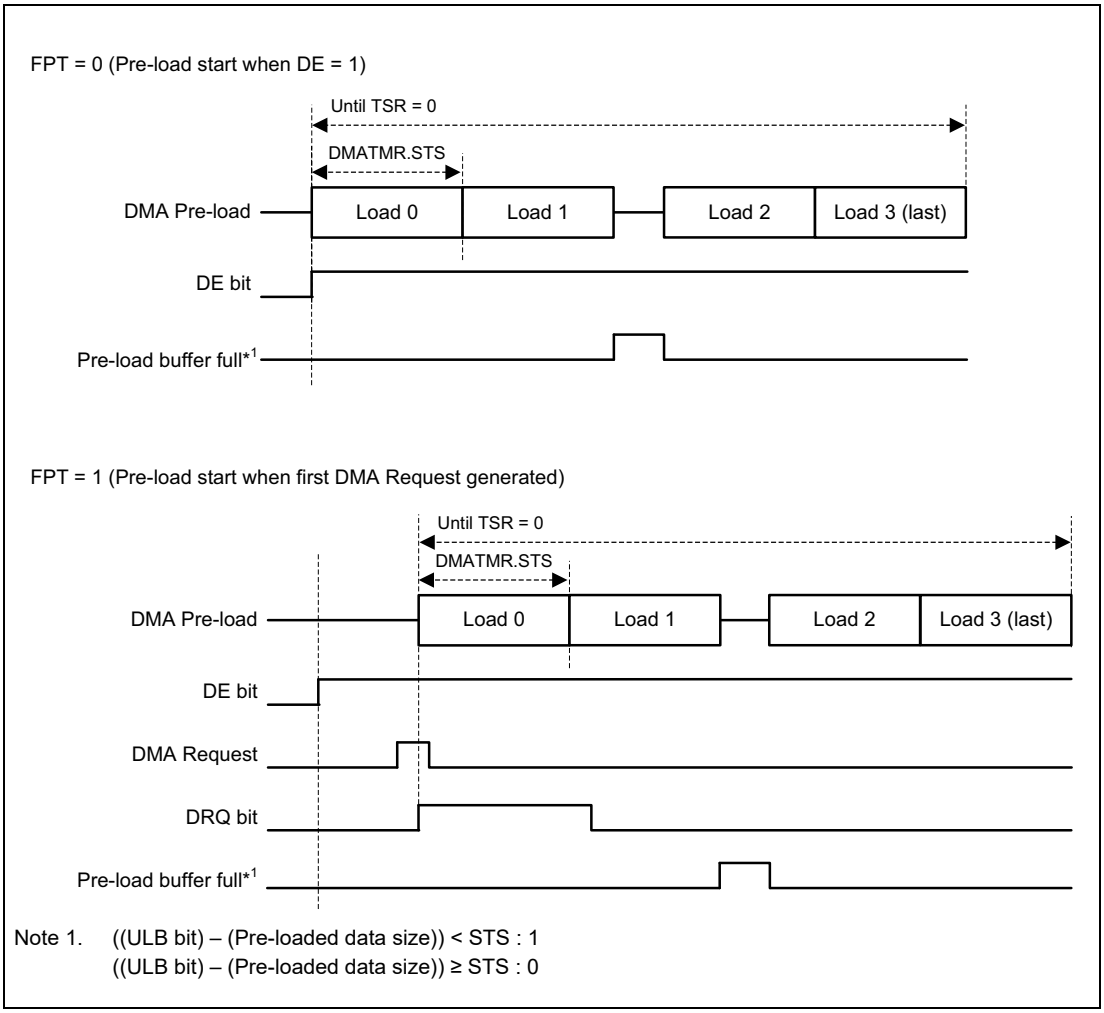


Figure 7.5 Transferring Data to a Peripheral Module in Hardware Request Mode



Note 1. ((ULB bit) – (Pre-loaded data size)) < STS : 1
((ULB bit) – (Pre-loaded data size)) ≥ STS : 0

Figure 7.6 Timing Chart for Pre-Load Function in Hardware Request Mode

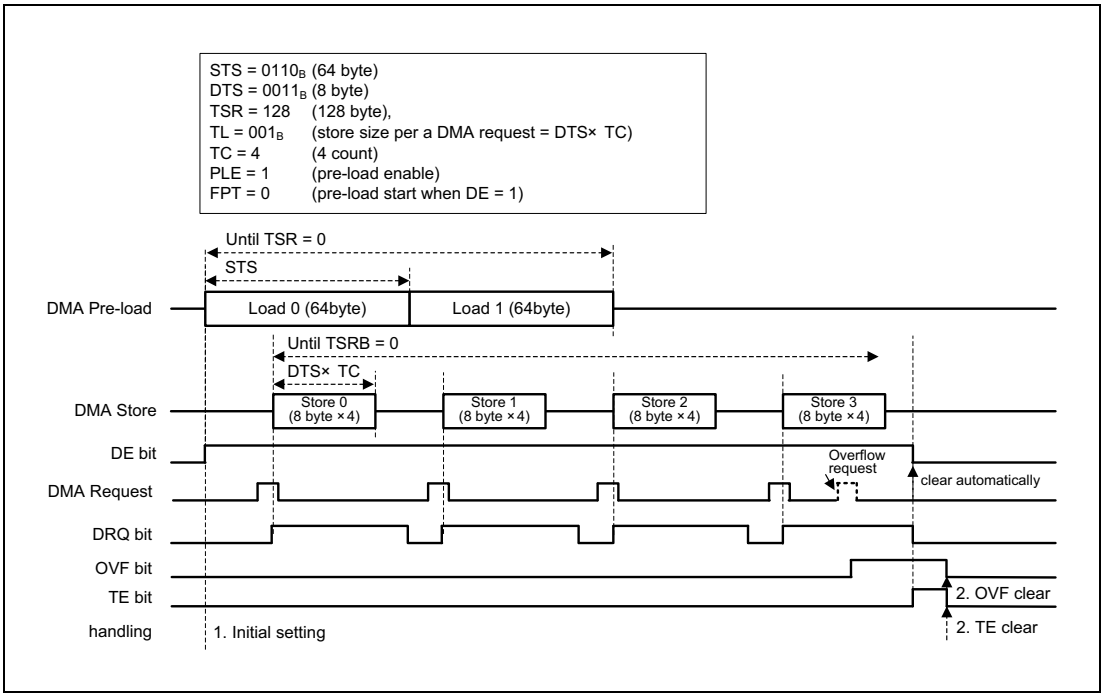


Figure 7.7 Timing Chart of Transferring Data to Peripheral Module

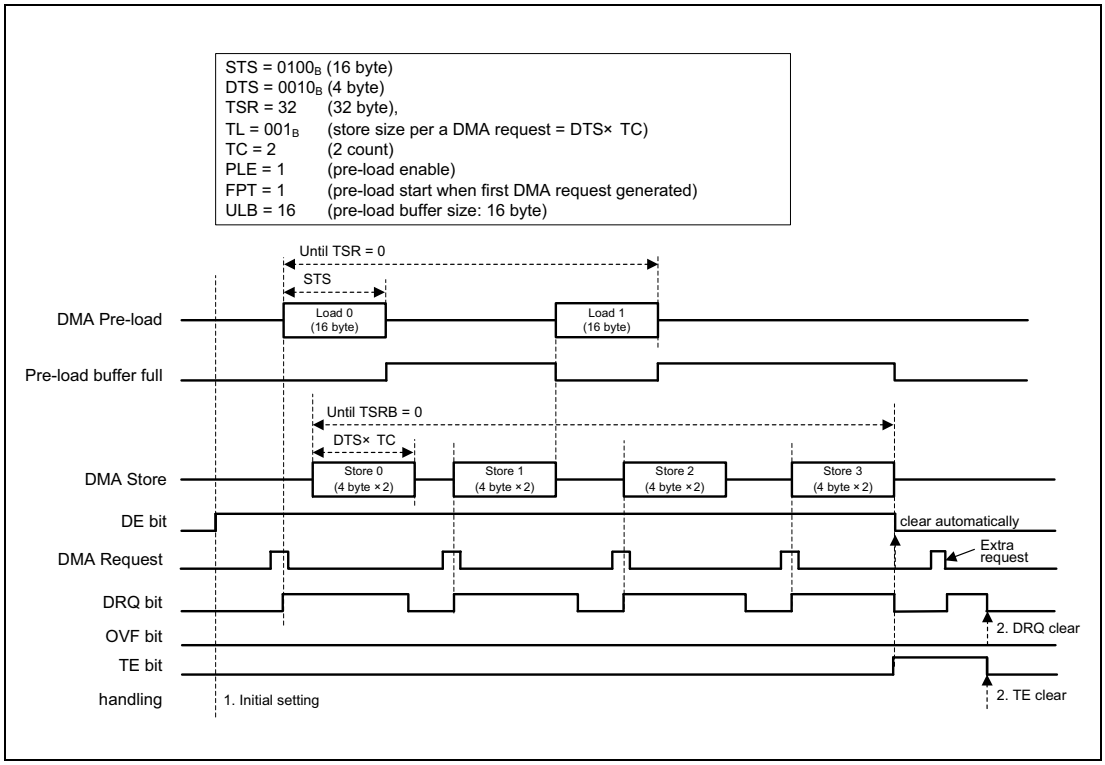


Figure 7.8 Timing Chart of Transferring Data with Buffer Control Operation

7.4.2 DMA Interrupts

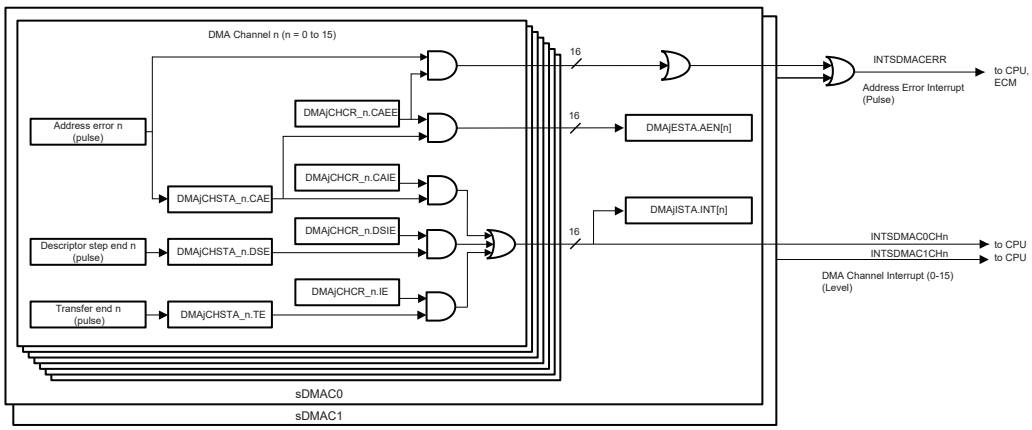
The routing of sDMAC interrupts is shown in **Figure 7.9**. sDMAC interrupts can be masked by the DMA channel registers.

A channel interrupt request is raised by sDMAC if any of these status bits is set (OR function):

- Transfer end (TE),
- Descriptor step end (DSE) ,
- Channel address error (CAE).

The address error is also notified to CPU by another path and interrupt INTSDMACERR can be generated.

The status of channel interrupt requests and address errors is indicated in global registers DMAjISTA and DMAjESTA.



Note: For details about causes of address errors, see **Section 7.5, (1) Address Errors**.

Figure 7.9 Interrupt Routing in sDMAC

7.4.3 Channel Priority

When sDMAC receives transfer requests from multiple channels, it transfers data according to a predetermined priority. One of two modes (fixed priority mode or round-robin mode) can be selected (DMAjOR.PR[1:0]).

(1) Fixed Priority Mode

In this mode, the priority levels among the channels remain fixed.

CH0 > CH1 > ... >CH14 > CH15

(2) Round-Robin Mode

In round-robin mode, each time data of one transfer unit (1 to 64 byte units) is transferred on one channel, the priority is rotated. The channel on which the transfer has just finished rotates to the bottom of the priority. The priority of round-robin mode is CH0 > CH1 > ... >CH14 > CH15 immediately after reset.

(3) Channel Request Priority

To increase the priority of the specified channel, use the PRI bits in the DMA transfer mode register (DMAjTMR_n). This function is given priority over the PR bits in the DMA operation register (DMAjOR). If a channel request is not accepted, the channel priority counter increases by the count set by the PRICNT bits in the DMA channel request priority register (DMAjCHPRI). The PRICNT bits in the DMA channel request priority register (DMAjCHPRI) are counted up by SCLK. When this priority counter reaches the maximum value (= 7), the highest priority request is accepted. However, if more than one channel has high priority, the requests are arbitrated according to the setting of the PR bits in DMAjOR.

Figure 7.10 shows the channel request priorities.

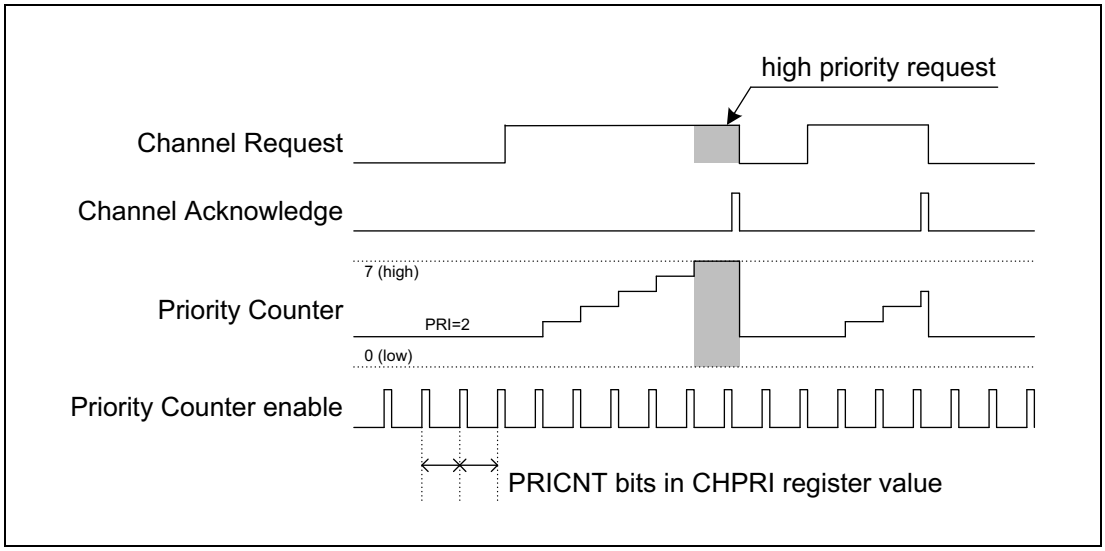


Figure 7.10 Channel Request Priorities

7.4.4 Slow Speed Mode

In slow speed mode, there is a configurable delay between consecutive DMA transfers. The delay (number of clock cycles) is specified by the SLM bit in the DMAjTMR_n register. Slow speed mode can be selected independently for each DMA channel. When one DMA channel operating in slow speed mode is waiting between DMA transfers, another DMA channel operating in slow speed mode can perform DMA transfer.

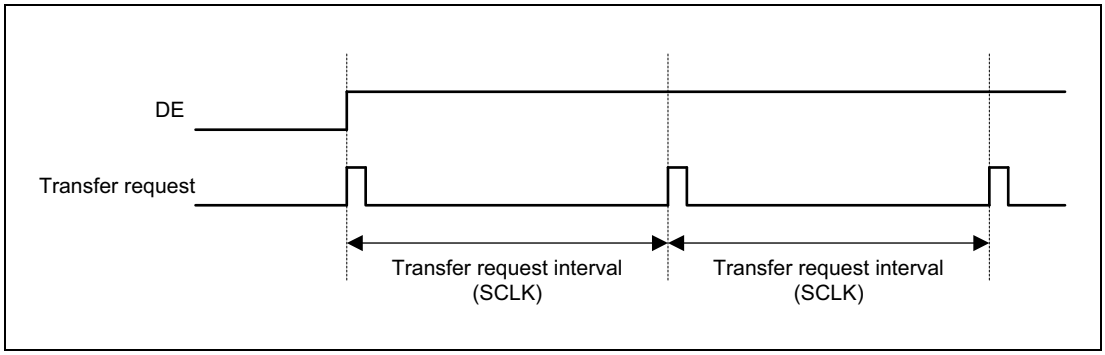


Figure 7.11 Slow Speed Mode

7.4.5 Scatter Gather Transfer

The DMAC supports gathering on the source side and scattering on the destination side.

The gather flow is shown in **Figure 7.12**. It is enabled if the GEN bit in DMAjSGCR_n for the given DMA channel is set to 1. The gather flow supports an inner and an outer loop. The number of iterations of the inner loop is specified by the GIRPT bit in DMAjSGCR_n register. The total of inner and outer loop transfers equals the number of TSR bytes. So the number of iterations of the outer loop is indirectly defined by the TSR register. The inner loop always transfers a data unit the size of an STS byte from the source to the data RAM and the SAR register is updated in the same way as transfers with gathering disabled. Additionally, the source address (SAR bits in DMAjSAR_n register) is increased by the inner address increment (GIAI bits in DMAjGIAI_n register). This procedure is repeated until the value of TSR bits in DMAjTSR_n or the value of GICNT bits in DMAjSGST_n register reaches zero, which concludes the inner loop. Before the inner loop is repeated, the source address (SAR bits in DMAjSAR_n register) is increased by the outer address increment (GOAI bits in DMAjGOAI_n register) and the GICNT bits in DMAjSGST_n register are restored with the value of GIRPT bits in DMAjSGCR_n register. The outer loop is repeated until the value of TSR bits in DMAjTSR_n register reaches zero. **Figure 7.14** shows an example with gathering and scattering enabled. If a transaction error occurs while the gather flow is active, the gather flow restarts at the beginning after the error condition is cleared.

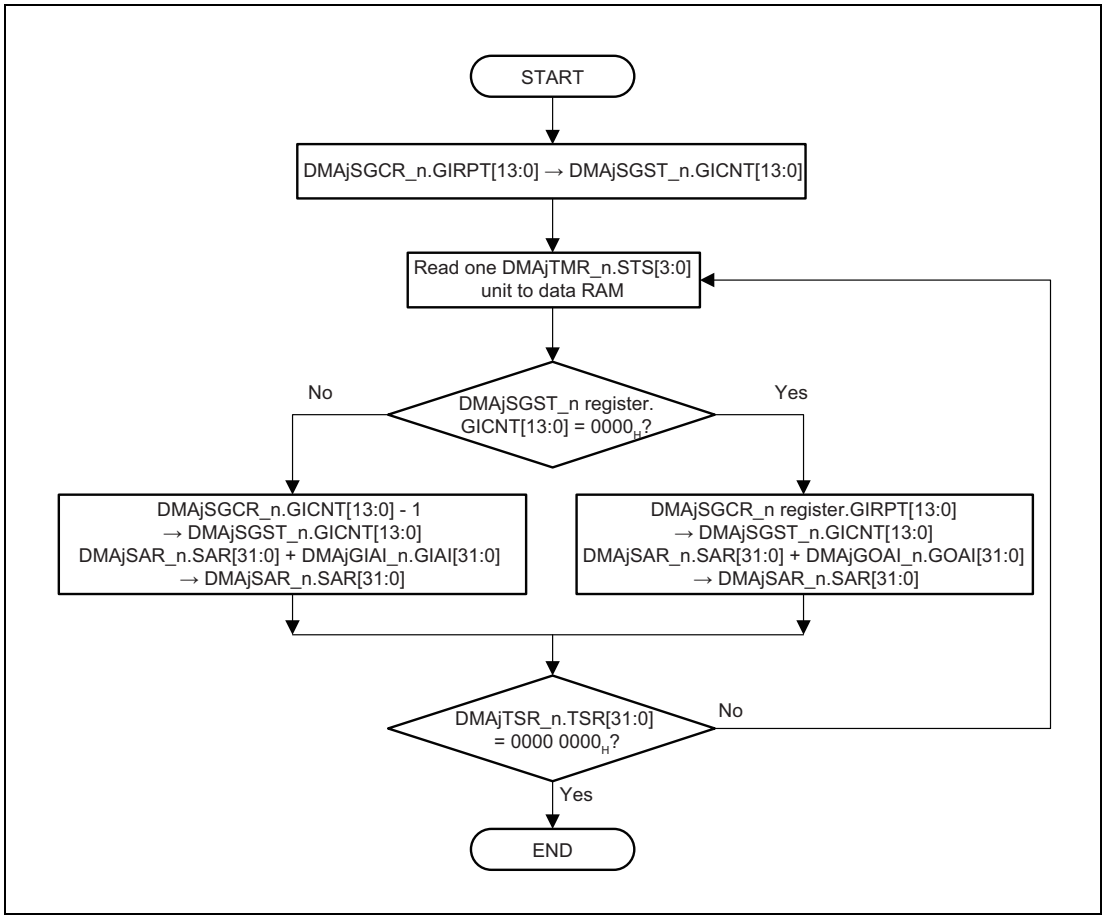


Figure 7.12 Gather Flow at Source Side If GEN = 1

The scatter flow is shown in **Figure 7.13**. It is enabled if the SEN bit in DMAjSGCR_n for the given DMA channel is set to 1. The scatter flow also supports an inner and outer loop. Additionally, zero filling is supported in the inner loop. The number of iterations of the inner loop is specified by the SIRPT register. The total of inner and outer loop transfers equals the number of TSRB bytes. So the number of iterations of the outer loop is indirectly defined by the TSRB register. The inner loop always transfers one data unit of size specified by DTS bits in DMAjTMR_n from the data RAM to the destination address and the DAR bits in DMAjDAR_n register is updated in the same way as for transfers with scattering disabled. If zero filling is disabled by setting ZF bit in DMAjSGCR_n = 0, the destination address (DAR bits in DMAjDAR_n register) is additionally increased by the inner address increment (SIAI bits in DMAjSIAI_n register), resulting in the destination address for the next data unit (the value of DAR bits in DMAjDAR_n register plus the value of SIAI bits in DMAjSIAI_n register). If zero filling is enabled by setting ZF bit in DMAjSGCR_n register = 1, value zero with data units specified by SIAI bits in DMAjSIAI_n register are written to the destination address and the DAR bits in DMAjDAR_n register is updated. This procedure also uses internal register ZFTSR to calculate the remaining zero fill transfer size. This procedure is repeated until the value of TSRW bits in DMAjTSRB_n register or the value of SICNT bits in DMAjSGST_n register reaches zero, which concludes the inner loop. Before the inner loop is repeated, the destination address (DAR bits in DMAjDAR_n register) is increased by the outer address increment (SOAI bits in DMAjSOAI_n register) if zero fill is disabled and the SICNT bits in DMAjSGST_n register is restored with the value of SIRPT bits in DMAjSGCR_n register. The outer loop is repeated until the value of TSRW bits in DMAjTSRB_n reaches zero. **Figure 7.14** shows an example with gathering and scattering enabled. If

a transaction error occurs while the scatter flow is active, the scatter flow restarts at the beginning after the error condition is cleared.

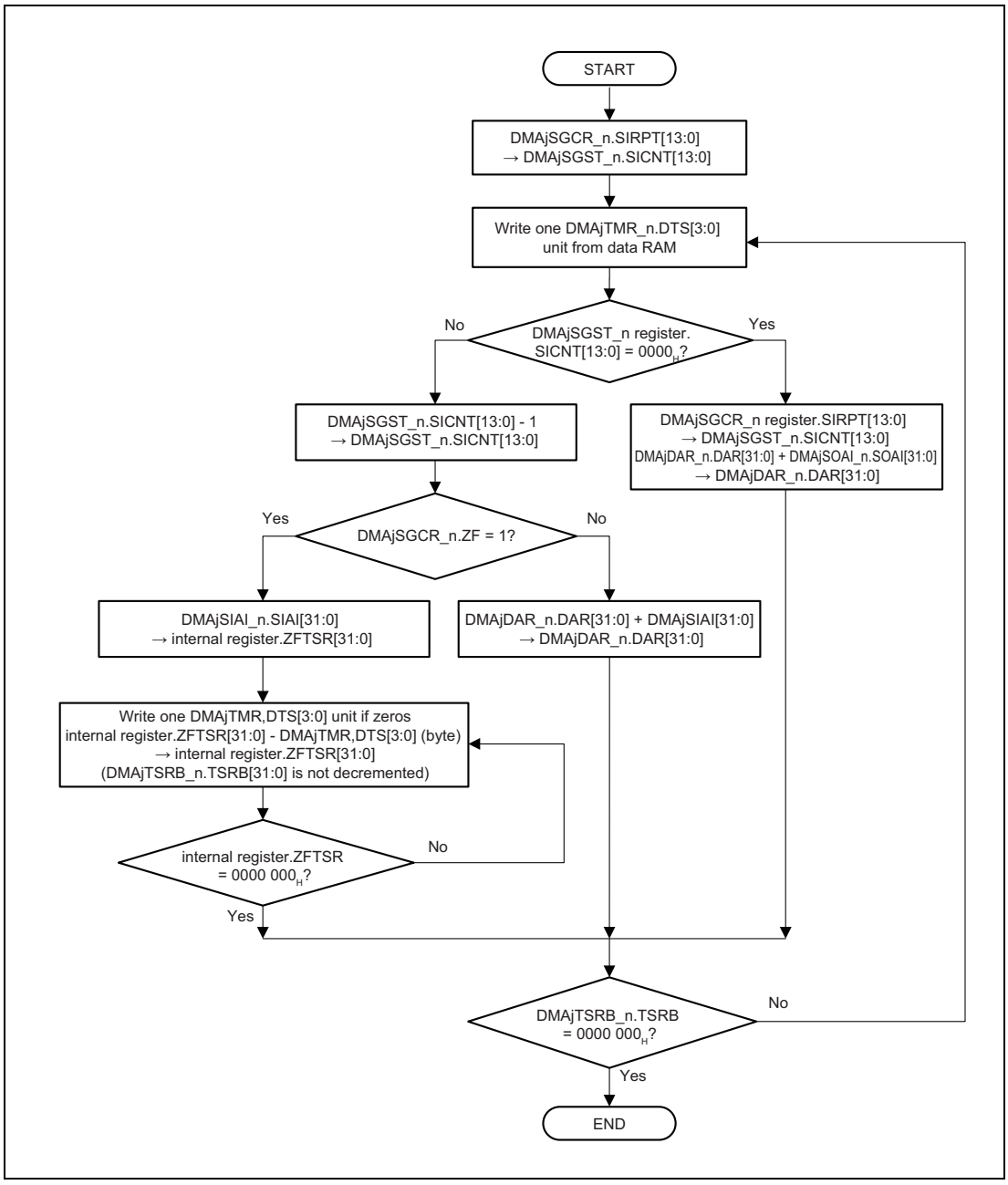


Figure 7.13 Scatter Flow at Destination Side If SEN = 1

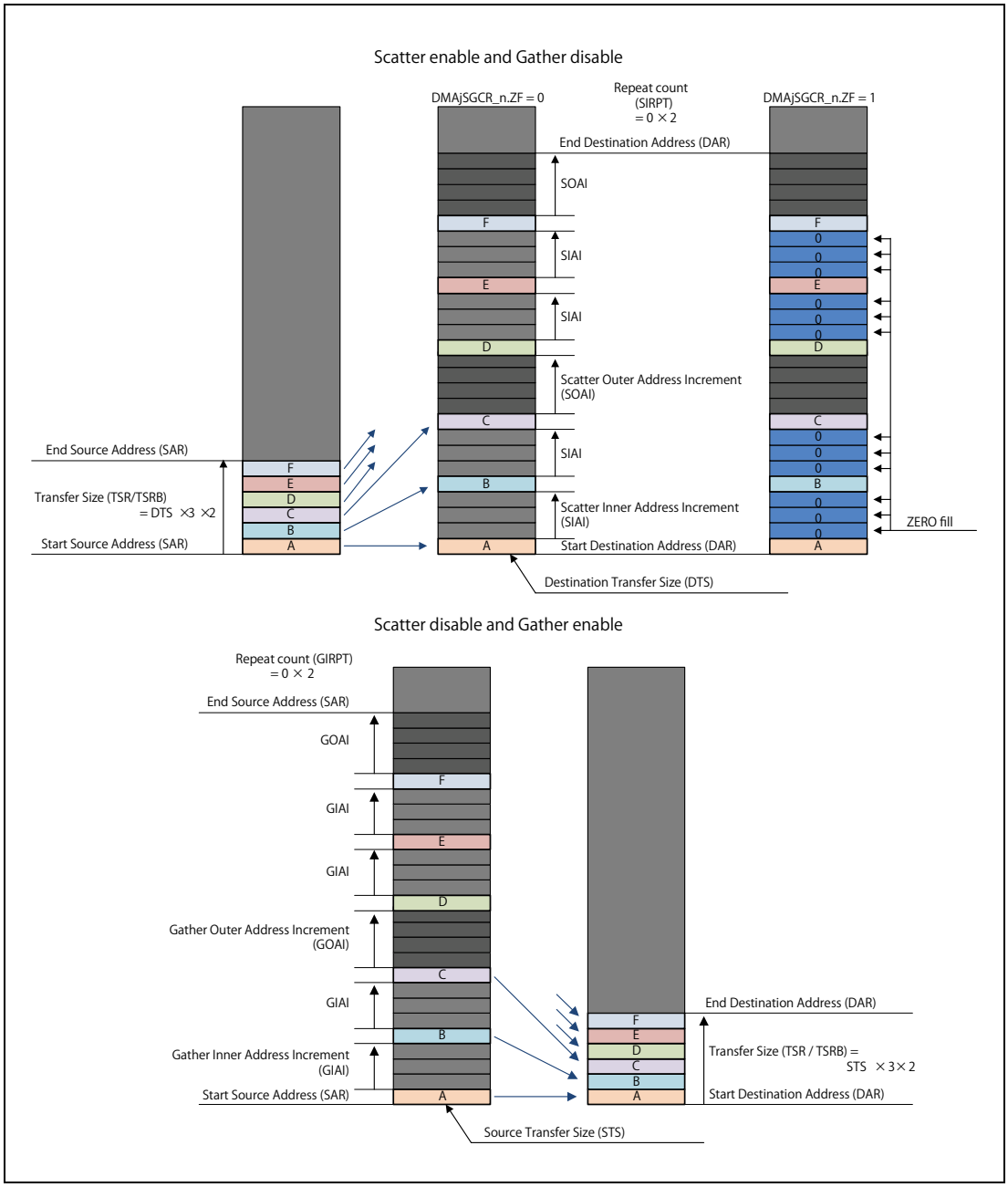


Figure 7.14 Examples That Show the Operation of Scatter and Gather Flows

The scatter gather features can be used to realize re-sorting. Transposing a column vector from a source memory address to a row vector at the destination memory address is realized by using gathering at the source side. The inner loop (GIRPT bits in DMAjSGCR_n register) is set to the number of samples in the vector. The data unit size (STS bits or DTS bits in DMAjTMR_n register) is set to the size of one vector element. The inner address offset (GIAI bits in DMAjGIAI_n register) is set to k*STS, where k is the width of the source matrix in the number of elements and STS is the value of STS bits in DMAjTMR_n register. The same principle can be used to transpose a row vector to a column vector. Here, scattering is used instead of gathering. Gathering and scattering can also be used to read every m-th source address and write it to every n-th destination address. Since gathering and scattering can be enabled independently, any combination is possible. At the destination side, zero filling can be used to set the words between the write addresses to zero.

Finally, the outer loop feature for both gathering and scattering allows the realization of inner and outer DMA loops.

7.4.6 Descriptors

A descriptor contains all the necessary register bits that describe a single transfer task on a channel. A channel can execute multiple descriptors in a chain by reloading the register settings from the descriptor memory. Descriptor operations are enabled by setting DPE bit to 1_B in DMAjCHCR_n. When the DMA transfer defined by the current register setting is finished, the next descriptor is loaded from the descriptor memory. The address of the descriptor to load is defined by the DMAjDPPTR_n.PTR bits. The new descriptor also updates the DMAjDPPTR_n register. This enables an arbitrary chain of descriptors.

Updating of the register list by descriptor operations is shown in **Table 7.27**.

The bit format in the descriptor memory matches the bit format in the corresponding DMA channel registers.

The DMAjSAR_n, DMAjDAR_n and DMAjTSR_n registers are updated during DMA transfer. If the DMAjSAR_n and DMAjDAR_n registers are not updated, the next DMA transfer is started from the final value of the previous DMA transfer. The DMAjTSR_n register is set to 0 after DMA transfer, so DMAjTSR_n register must be updated.

Table 7.27 Register List of Channels to Be Updated

No	Name	Abbreviation	Bit	DMADPCR.UPF bit
1	DMA source address register	DMAjSAR_n	All bits	UPF[0]
2	DMA destination address register	DMAjDAR_n	All bits	UPF[1]
3	DMA transfer size register	DMAjTSR_n	All bits	UPF[2]
4	DMA transfer mode register	DMAjTMR_n	All bits	UPF[3]
5	DMA gather inner address increment register	DMAjGIAI_n	All bits	UPF[4]
6	DMA gather outer address increment register	DMAjGOAI_n	All bits	UPF[5]
7	DMA scatter inner address increment register	DMAjSIAI_n	All bits	UPF[6]
8	DMA scatter outer address increment register	DMAjSOAI_n	All bits	UPF[7]
9	DMA scatter gather control register	DMAjSGCR_n	All bits	UPF[8]
10	DMA resource select register	DMAjRS_n	All bits	UPF[9]
11	DMA buffer control register	DMAjBUFCR_n	All bits	UPF[10]

Set as follows for each channel that should use the descriptor function:

- Specify the register to be updated in the UPF bits of the DMAjDPCR_n register.
- Set the DMA transfer setting in the descriptor memory.
- If DMA transfer should start when a configuration is copied from the descriptor memory to DMA channel registers
  - set the DMAjDPPTR_n.PTR bits to the first transfer configuration in the descriptor memory
  - set DMAjCHCR_n.DPB to 1

- If DMA transfer should start according to a register setting and continue with the descriptor function.
  - configure the channel registers
  - set the DMAjDPPTR_n.PTR bits to the address of the next transfer configuration in the descriptor memory
  - set DMAjDPPTR_n.CF to 1
  - set DMAjCHCR_n.DPB to 0

The descriptor memory is shared by all DMA channels. Therefore is it important to avoid overlaps of areas used by different DMA channels in common descriptor memory.

(1) Configuration of Descriptor Memory

A DMA transfer configuration can start from any address in the descriptor memory. Register configurations that are not updated should not be written.

Examples of configurations in descriptor memory are shown in the figure below.

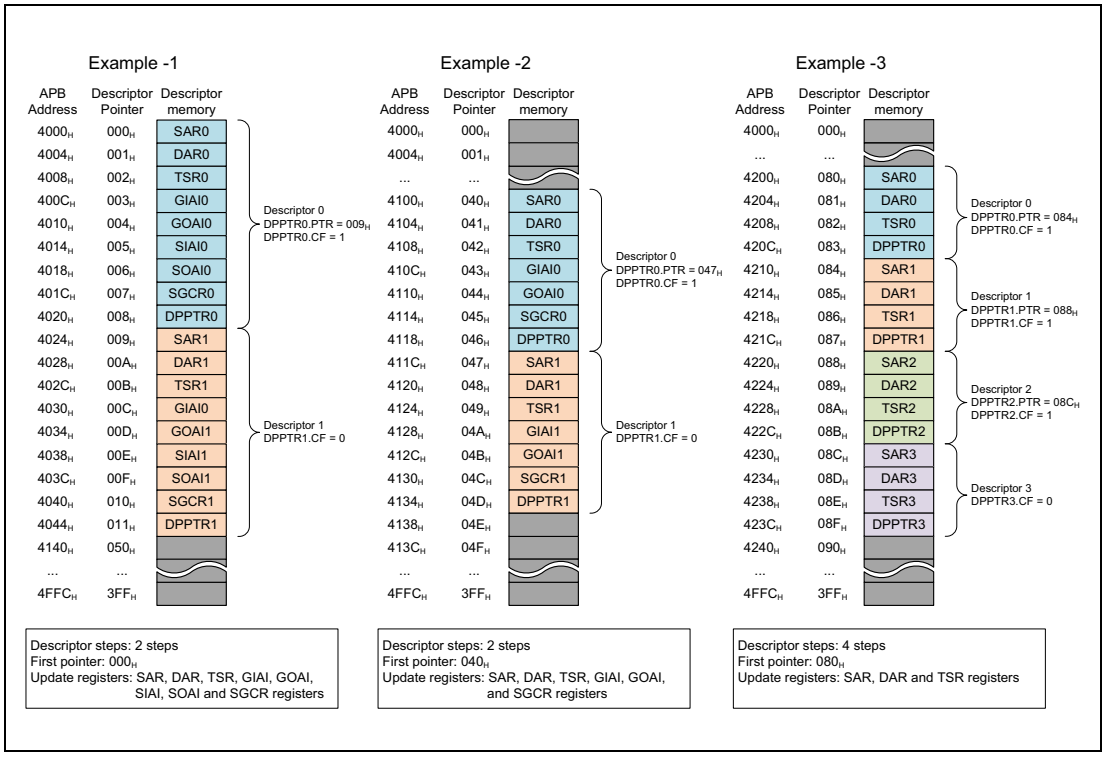


Figure 7.15 Built-in Descriptor Memory Configuration

(2) Flow of Setting Updating by Descriptor Operation

Updating a configuration by using the descriptor function is specified by the UPF bits in the DMAjDPCR_n register. If a register is updated by a descriptor operation, the PTR bit in DMAjDPPTR_n is incremented. Finally the DMAjDPPTR_n register is updated. Therefore, the DMAjDPPTR_n register setting must be always written into the descriptor memory.

This flow is automatically processed by hardware.

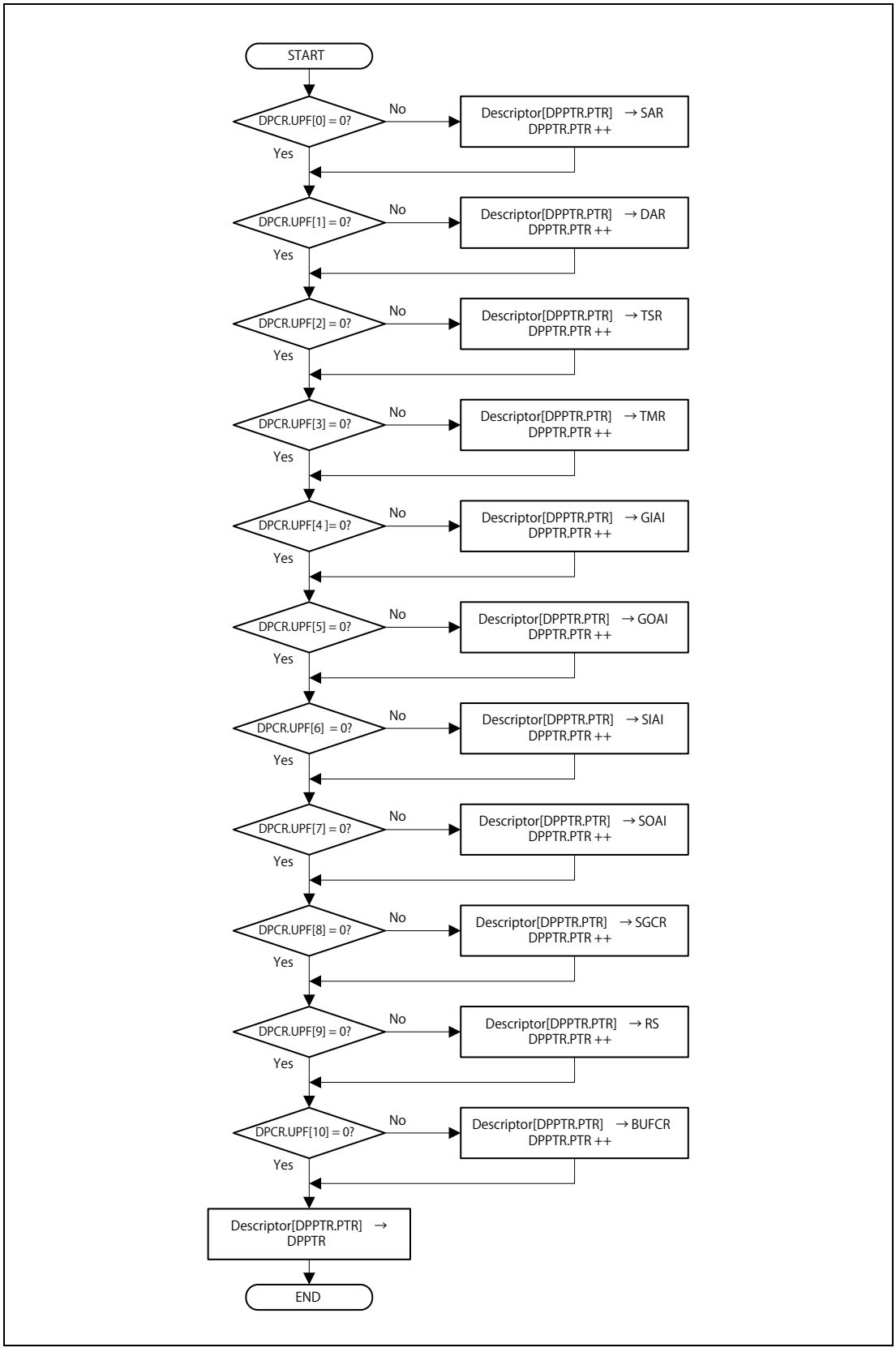


Figure 7.16 Flow of Setting Updating by Descriptor Operation

(3) Descriptor Operational Flow

Figure 7.17 shows the descriptor operational flow.

When the first transfer setting is written in the descriptor memory, the DPB bit in the DMAjCHCR_n register is set to 1. When a DSE interrupt is issued after a DMA transfer step is completed, the DSIE bit in the DMAjCHCR_n register and the DIE bit in DMAjDPPTR_n in the descriptor memory are set to 1. A DSE interrupt can be issued at the optional timing by using this setting. If the DSE bit in the DMAjCHSTA_n register is already set when the DMA transfer step completes, the next transfer setting is not read from the descriptor memory. There is a possibility that the next setting is not written in the descriptor memory because the first DSE interrupt has not been processed. The descriptor can be continued by clearing the first DSE interrupt. When the DPE bit in DMAjCHCR_n register is set to 0 or the CF bit in the DMAjDPPTR_n register in the descriptor memory is set to 0 after a DMA transfer step is complete, a TE interrupt is issued, and the descriptor ends.

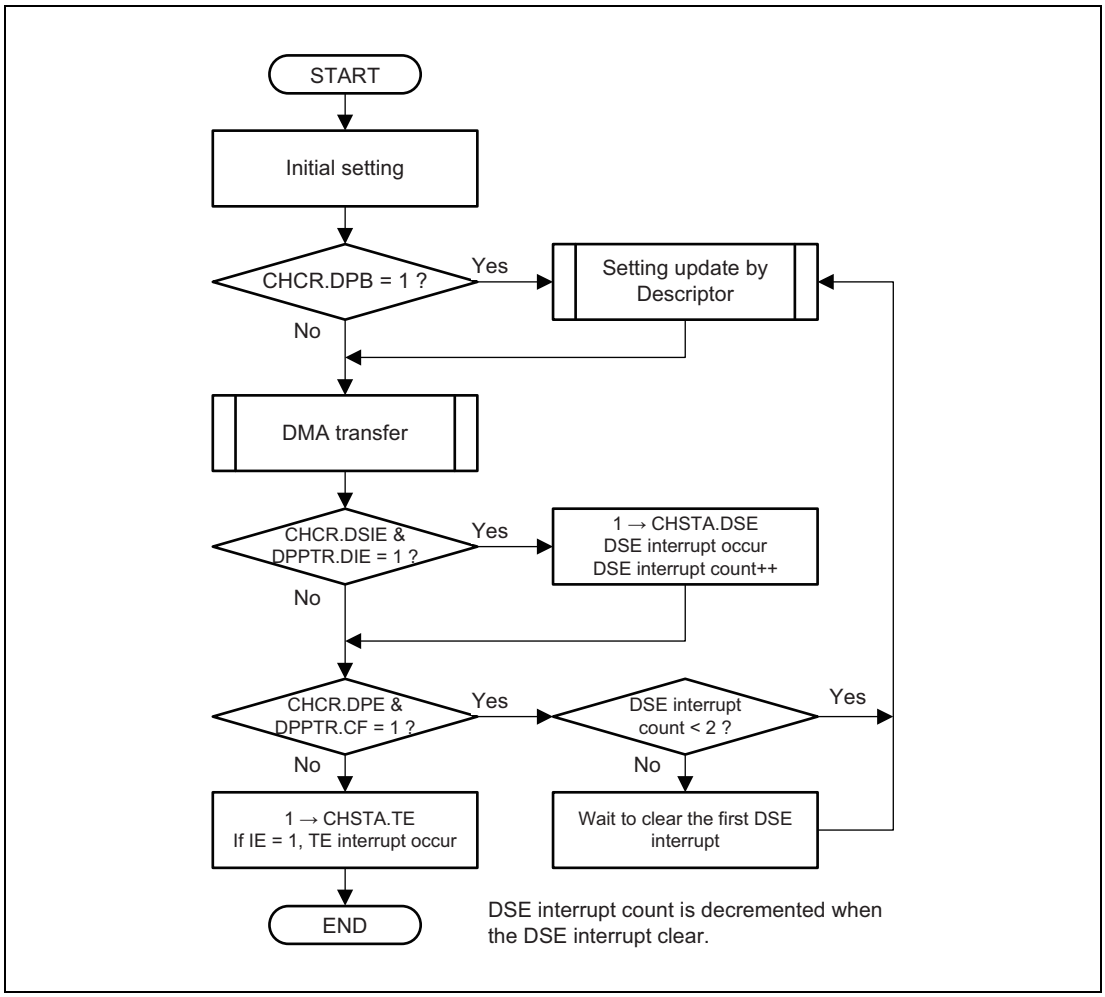


Figure 7.17 Descriptor Operational Flow

(4) Example of descriptor usage

Figure 7.18 to **Figure 7.25** show examples of using descriptors operations.

(a) Example 1: Normal Descriptor Operations

The DMAC transfers descriptors in the listed order and generates an interrupt after all of the transfers are complete.

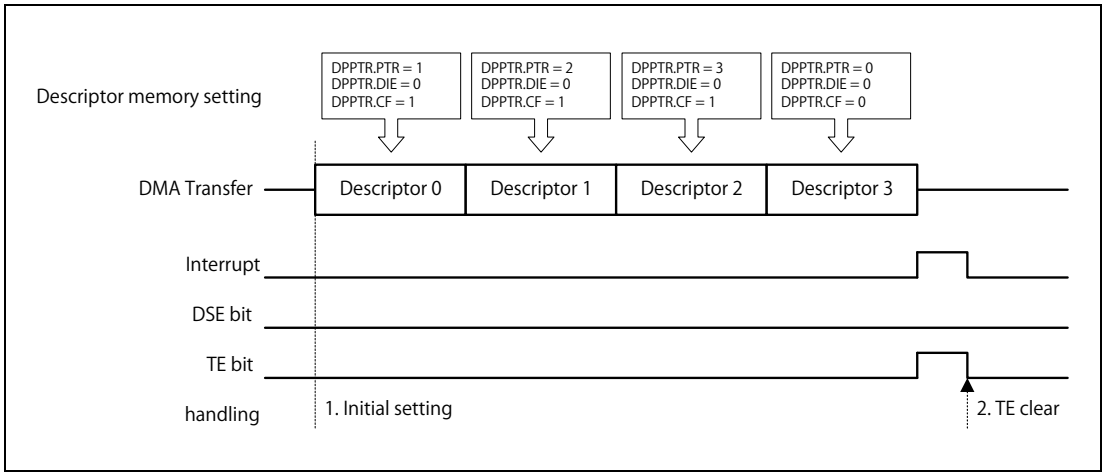


Figure 7.18 Normal Descriptor Operations (Example 1)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 0, PTR = Descriptor 2 pointer Descriptor[2].DPPTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer Descriptor[3].DPPTR CF = 0, DIE = 0, PTR = Any value The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt disable (CHCR.DSIE = 0) To start the transfer after reading descriptor memory (CHCR.DPB = 1)
2	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer (CHCR.DE = 0) Clear the transfer end flag (CHFCR.TEC = 1)

(b) Example 2: Specific Order Operations

The DMAC transfers descriptors in the specified order and generates an interrupt after all of the transfers are complete.

Descriptor 0 → Descriptor 1 → Descriptor 3 → Descriptor 2

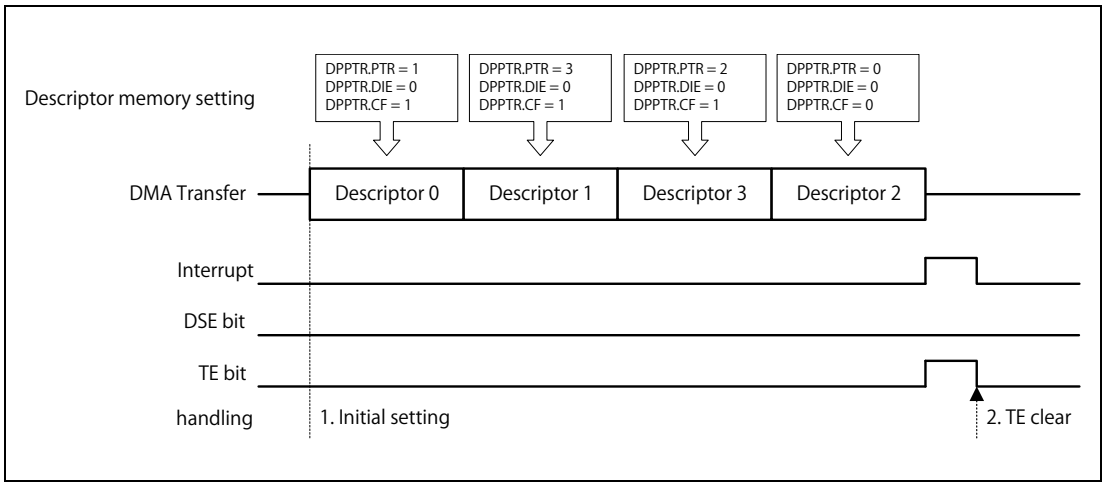


Figure 7.19 Specific Order Operation (Example 2)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer Descriptor[2].DPPTR CF = 0, DIE = 0, PTR = Any value Descriptor[3].DPPTR CF = 1, DIE = 0, PTR = Descriptor 2 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)
2	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(c) Example 3: Descriptor Step End Interrupt

The DMAC transfers the descriptors in order, and generates an interrupt after the specified descriptor is complete.

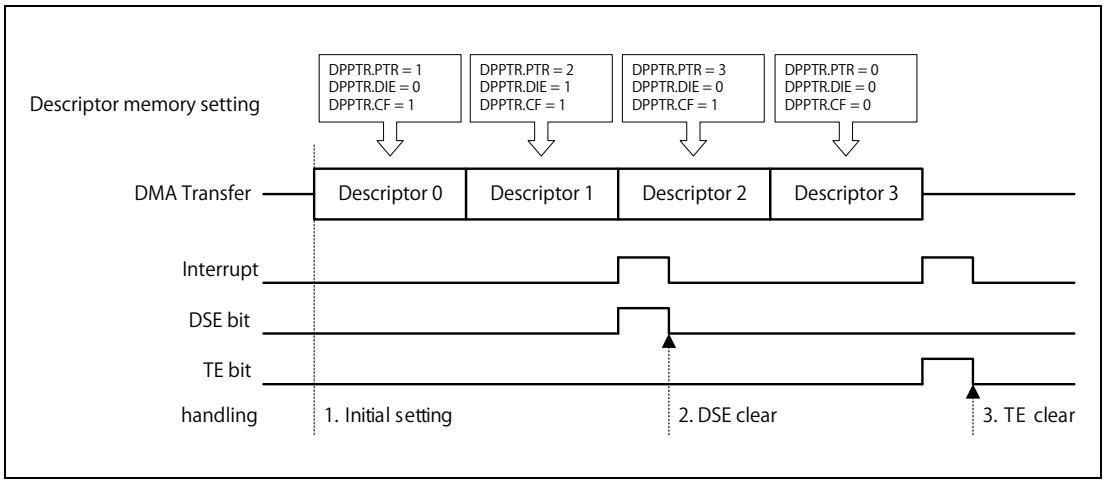


Figure 7.20 Descriptor Step End Interrupt (Example 3)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer Descriptor[2].DPPTR CF = 1, DIE = 0, PTR = Descriptor 3 pointer Descriptor[3].DPPTR CF = 0, DIE = 0, PTR = Any value The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)
2	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
3	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer (CHCR.DE = 0) Clear the transfer end flag (CHFCR.TEC = 1)

(d) Example 4: Double Buffer Operation

Example of double buffer operation using DSE interrupt.

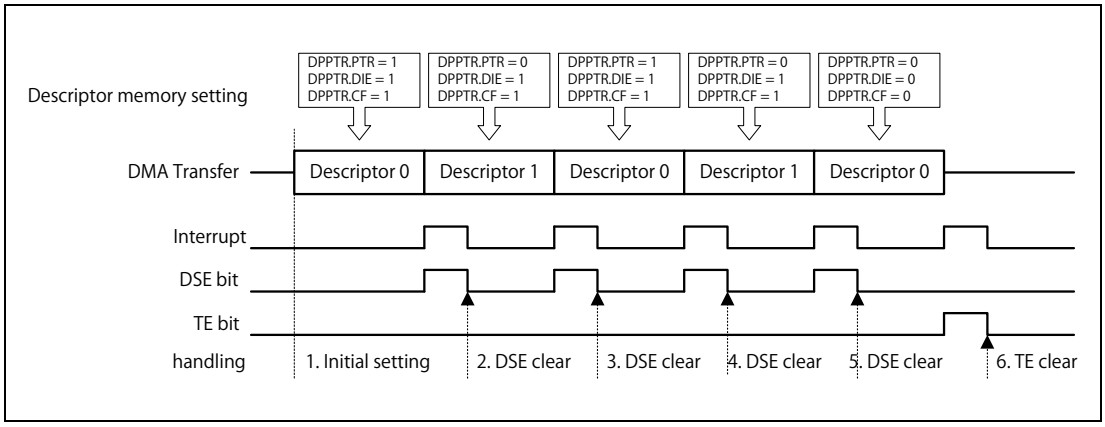


Figure 7.21 Double Buffer Operation (Example 4)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptor 0 to 1 is written in Descriptor Memory Descriptor[0].DPPTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)
2, 3	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
4	DSE clear	Check the interrupt status (read CHSTA) Clear continuation flag. (Descriptor[0].DPPTR.CF = 0) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
5	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
6	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(e) Example 5: Double Buffer Operation with Interrupt Processing Delay

When the next DSE interrupt occurs without clearing the DSE interrupt and without reading the next setting from descriptor memory, transmission is suspended. When the DSE interrupt is cleared after writing the next setting to the descriptor memory, transmission resumes.

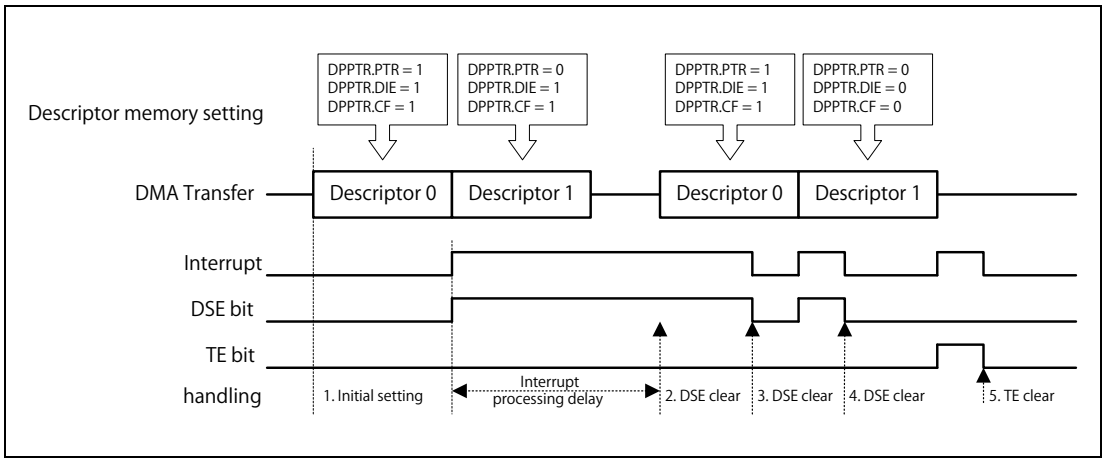


Figure 7.22 Double Buffer Operation with Interrupt Processing Delay (Example 5)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptor 0 to 1 is written in Descriptor Memory Descriptor[0].DPPTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 1) DSE interrupt enable (CHCR.DSIE = 1) After reading descriptor memory (CHCR.DPB = 1)
2	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
3	DSE clear	Check the interrupt status (read CHSTA) Clear continuation flag. (Descriptor[1].DPPTR.CF = 0) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
4	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
5	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(f) Example 6: Repeat Operation

The continuation flag remains set and DMA transfer is repeated.

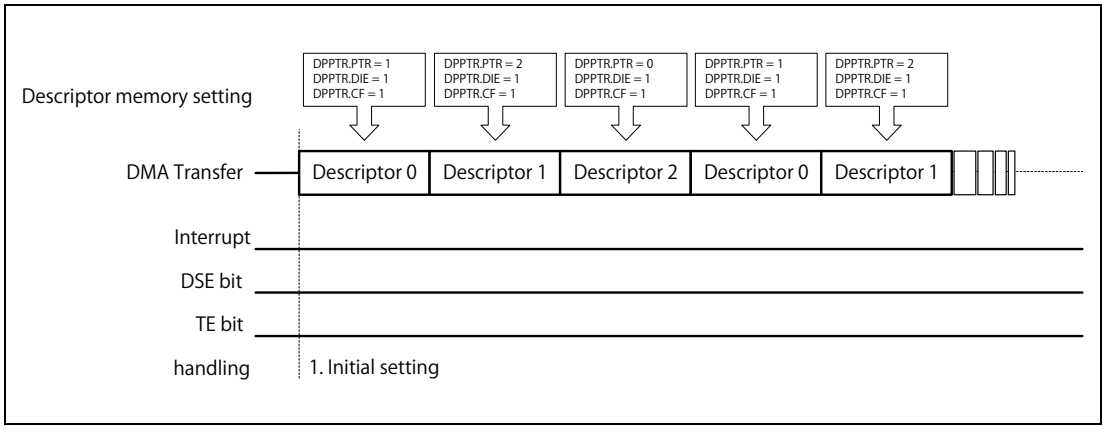


Figure 7.23 Repeat Operation (Example 6)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 2 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 1, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer Descriptor[2].DPPTR CF = 1, DIE = 1, PTR = Descriptor 0 pointer The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 0) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)

(g) Example 7: Hardware Request Operation

When hardware transfer request mode is set, the descriptor operation is executed by a hardware request.

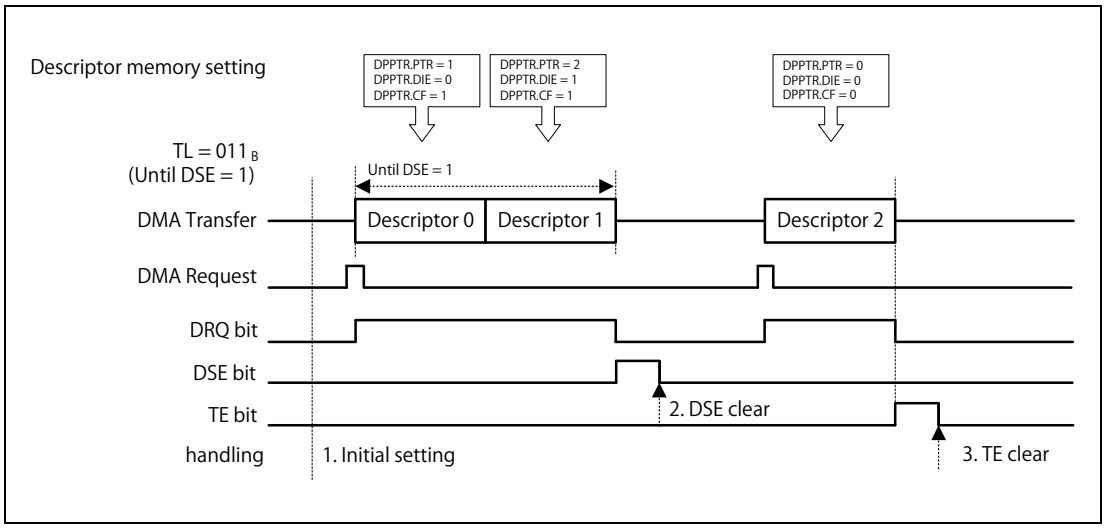


Figure 7.24 Hardware Request Operation (Example 7)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 1, DIE = 1, PTR = Descriptor 2 pointer Descriptor[2].DPPTR CF = 0, DIE = 0, PTR = Any value The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Setting hardware transfer request in each Descriptor n (n = 0~2) Hardware request mode (Descriptor[n].DMAjTMR_n.TRS = 1) Transfer limit by DSE is asserted (Descriptor[n].DMAjRS_n.TL = 3'b011) Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 0) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)
2	DSE clear	Check the interrupt status (read CHSTA) Clear Descriptor step end flag. (CHFCR.DSEC = 1)
3	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(h) Example 8: Hardware Request Operation with Different Request Source Settings

If a hardware request source setting is different in chained descriptors, the generated DMA request is automatically cleared when the descriptor memory setting is loaded.

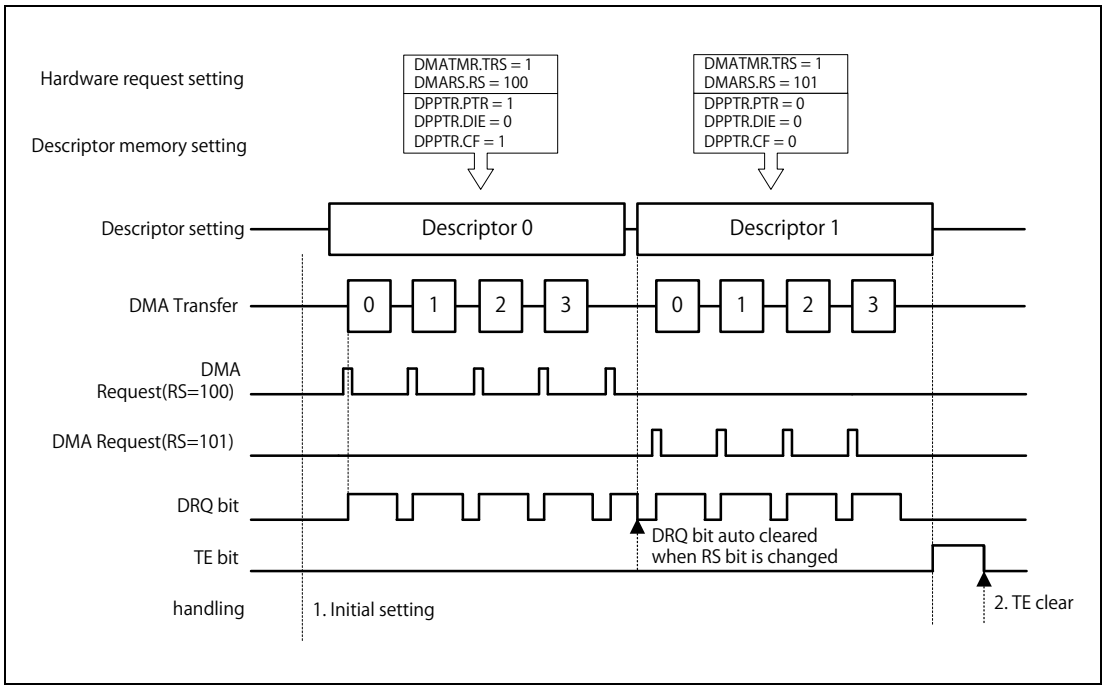


Figure 7.25 Hardware Request Operation with Different Request Source Settings (Example 8)

No	Handling	Detail
1	Initial setting	Specify the register to be updated. Write to the DPCR register. Descriptors 0 to 3 are written to the descriptor memory. Descriptor[0].DPPTR CF = 1, DIE = 0, PTR = Descriptor 1 pointer Descriptor[1].DPPTR CF = 0, DIE = 0, PTR = Any value The first descriptor pointer is written to the DPPTR register. DPPTR.PTR = 0 Setting hardware transfer request in each Descriptor Descriptor[0].DMAjTMR_n TRS = 1 Descriptor[0].DMAjRS_n RS = 100 (hardware request resource channel) Descriptor[1].DMAjTMR_n TRS = 1 Descriptor[1].DMAjRS_n RS = 101 (hardware request resource channel) Start the descriptor operation (write to CHCR) Descriptor enable (CHCR.DPE = 1) TE interrupt enable (CHCR.IE = 0) DSE interrupt disable (CHCR.DSIE = 0) After reading descriptor memory (CHCR.DPB = 1)
3	TE clear	Check the interrupt status (read CHSTA) Disable DMA transfer. (CHCR.DE = 0) Clear the transfer end flag. (CHFCR.TEC = 1)

(5) Stopping a Descriptor Operation

Three methods can be used to stop a descriptor operation.

Method 1: Descriptor updating is stopped after completion of the specific DMA transfer.
The specific CF bit in DMAjDPPTR_n in the descriptor memory is set to 0.
The TE interrupt is issued after the end of the specified DMA transfer.

Method 2: Descriptor updating is stopped after the current DMA transfer is complete.
The DPE bit in the DMAjCHCR_n register is set to 0 by writing 1 to the DPEC bit in the DMAjCHFCR_n register. The TE interrupt is issued after the end of the current DMA transfer even if the CF bit is set to 1.
It is not recommended to use the CF bit of the DMAjDPPTR_n register to stop DMA transfer. This bit is updated by the descriptor function.

Method 3: Immediate stop without completing the current DMA transfer.
See **Section 7.5, (2) DMA Transfer Abort** and **Section 7.5, (3) Suspend and Resume DMA Transfer**.

7.4.7 Transfer Flow

Write the transfer condition settings to the DMA source address register (DMAjSAR_n), DMA destination address register (DMAjDAR_n), DMA transfer size register (DMAjTSR_n), DMA transfer mode register (DMAjTMR_n), DMA Channel control register (DMAjCHCR_n), DMA resource select register (DMAjRS_n) and DMA operation register (DMAjOR). The DMAC transfers data in following order.

(1) Auto Request Mode

- A transfer request occurs and the DMAC confirms whether transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0). Transfer starts automatically.
- If a transfer request is generated, one transfer unit (specified by STS [3:0] or DTS[3:0]) will be transferred. DMAjTSR_n will be decremented by the transfer unit every time DMA transfer completes.
- When the specified number of transfers are complete (the value of DMAjTSR_n and DMAjTSRB_n is 0), transfer ends normally. At this time if the IE bit of DMAjCHSTA_n is set to 1, a TE interrupt will be issued to the CPU.
- Transfer is terminated if an address error occurs in DMA transfer. In addition, transfer is terminated even if the DE bit of DMAjCHCR_n or the DME bit of DMAjOR is set to 0.

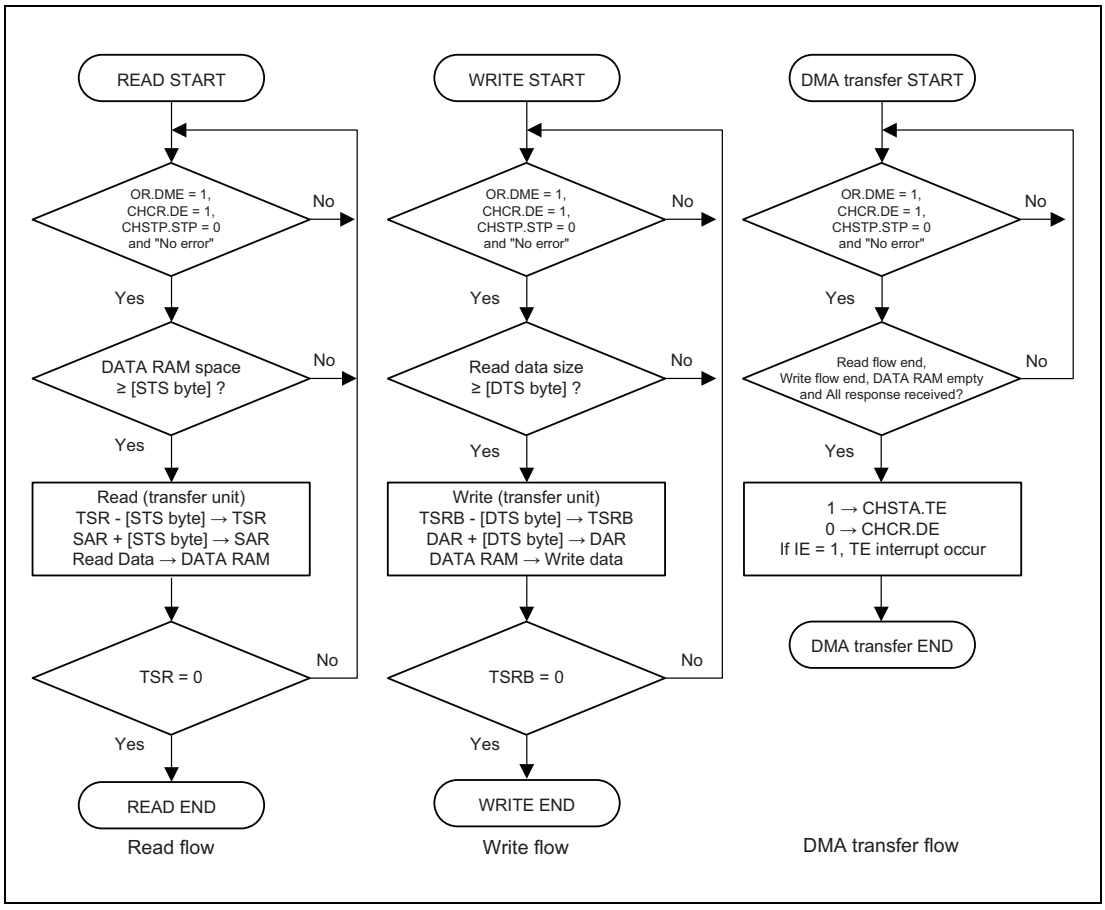


Figure 7.26 Transfer Flow in Auto Request Mode

(2) Hardware Request Mode

- **Figure 7.27** shows the flow in hardware request mode.
- A hardware request can be received when the DMAC confirms whether transfer is enabled (DE = 1, DME = 1, TE = 0, CAE = 0).
- If PLE is set to 1 and FPT is set to 0, pre-loading data from the source memory to the data RAM of sDMAC starts automatically. DMAjTSR_n will be decremented by the transfer unit every time pre-load transfer completes. If FPT is set to 1, pre-loading occurs after the first hardware transfer request is generated and the internal status register (FPS) is set to 0.
- When a hardware transfer request is generated, one transfer unit (specified by TL[2:0]) will be transferred. DMAjTSRB_n will be decremented by the transfer unit every time transfer completes.
- When the specified number of transfers are complete (the value of DMAjTSR_n and DMAjTSRB_n is 0), transfer ends normally. At this time if the IE bit of DMAjCHSTA_n is set to 1, a TE interrupt will be issued to the CPU.
- Transfer is terminated if an address error occurs in DMA transfer. In addition, transfer is terminated even if the DE bit of DMAjCHCR_n or the DME bit of DMAjOR is set to 0.

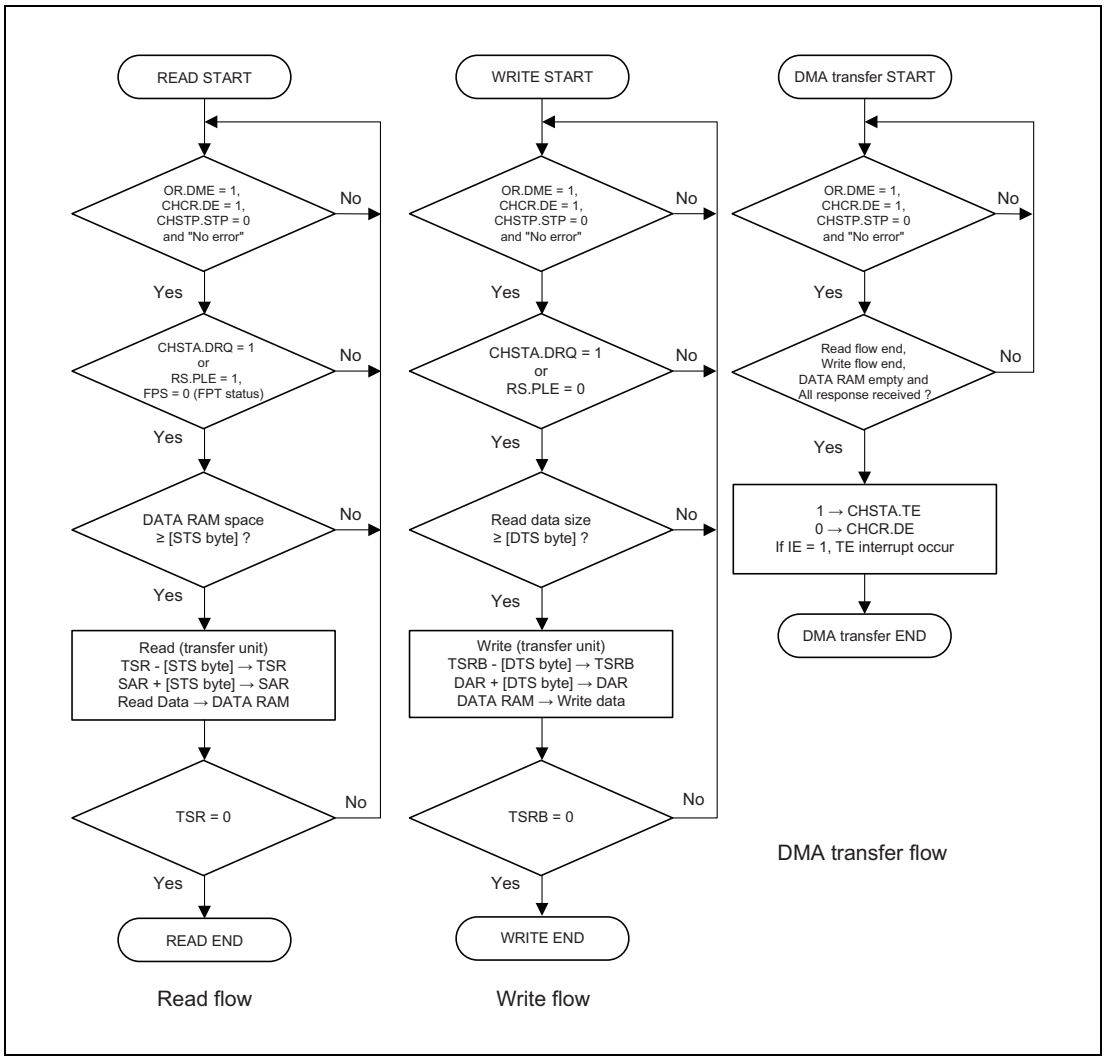


Figure 7.27 Transfer Flow in Hardware Request Mode

7.4.8 Performance

The DMA controller operates on a CLK_HBUS. Read and write transaction requests are issued in parallel on both system interconnect read and write interfaces to enable flyby transfers. Transaction requests are issued without waiting for a reply. So multiple outstanding requests can be in flight on the bus. This is used to hide latencies on the path from the DMA to the slave. The DMA supports 16 outstanding transactions per system interconnect interface (16 reads and 16 writes).

All channels operate in parallel. Arbitration on the read/write interfaces is performed as defined in the operation register DMAjOR. The bus data width is 64 bits (8 bytes). If the unit transfer size (STS/DTS) is larger than 8 bytes, the transaction requests multiple replies in sequence (also called “burst”). The maximum burst size is 8, resulting in a maximum transfer unit size of 64 bytes.

The maximum number of outstanding requests in each DMA channel changes depending on the unit transfer size (STS/DTS). Each DMA channel can use 128 bytes in data RAM as outstanding transfer data buffer. If the unit transfer size (STS/DTS) in a DMA channel is 16 bytes or more, the number of outstanding requests in the DMA channel is 8 or less and the maximum number of outstanding request is 16 or less. Each DMA channel can issue a new transaction request every three clock cycles at best. The frequency at which each DMA channel issues a new transaction request changes depending on the number of outstanding requests in the DMA channel or the number of channels using DMA transfer. If the unit transfer size (STS/DTS) in a DMA channel is 64 bytes, the number of outstanding requests in the DMA channel is 2. In this case, if DMA transfer is operated with one DMA channel and a transaction request is issued twice, a subsequent read transaction request is not issued before previous read and write transaction requests are processed, and the DMA channel cannot issue a new transaction request every three clock cycles.

To utilize the read and write data channels every cycle, two approaches can be used:

- Use three DMA channels. Each channel transfers one third of the required data. The DMAC issues a new request transaction every cycle on the read and write interfaces in parallel.
- Use one DMA channel with a STS/DTS value of 8 bytes. The read/write transaction requests are issued every three cycles.

Both approaches fully utilize the read and write data channels if the data RAM is sufficiently utilized.

7.5 Usage Notes

Pay attention to the following notes when using the DMAC.

(1) Address Errors

When a DMA address error occurs, set the register of the erroneous channel again after DMA transfer abort shown in **Section 7.5, (2) DMA Transfer Abort** and then start transfer. Address errors are caused by the factors indicated in **Table 7.28**.

Table 7.28 Address Error Factors

Factor	Description
DMA transfer error	The transfer source or transfer destination is invalid space.
	The transfer source or transfer destination is a GUARD error.
Prohibited setting error	DMA source transaction size setting is prohibited. (DMATMR.STS > 0110 : 64 bytes)
	DMA destination transaction size setting is prohibited. (DMATMR.DTS > 0110 : 64 bytes)
	DMA transfer count is zero in hardware request mode (DMARS.TC = 0 when DMARS.TL = 000, 001)

(2) DMA Transfer Abort

To abort DMA transfer, clear the DE bit in the DMA channel control register

(DMAjCHCR_n) by using the DEC bit in the DMA channel flag clear register (DMAjCHFCR_n) to disable DMA transfer. After that, the BUSY bit is cleared to 0. There is a possibility that a TE/DSE interrupt may occur to asynchronous timing after aborting transfer, so it is necessary to assume the following case and take appropriate measures:

- After DMA transfer is aborted, the TE/DSE bit is set to 1 and the BUSY bit is cleared. If the IE/DSIE bit is set to 1 at that time, the DMA interrupt is asserted.

Figure 7.28 shows a processing example when DMA transfer is aborted.

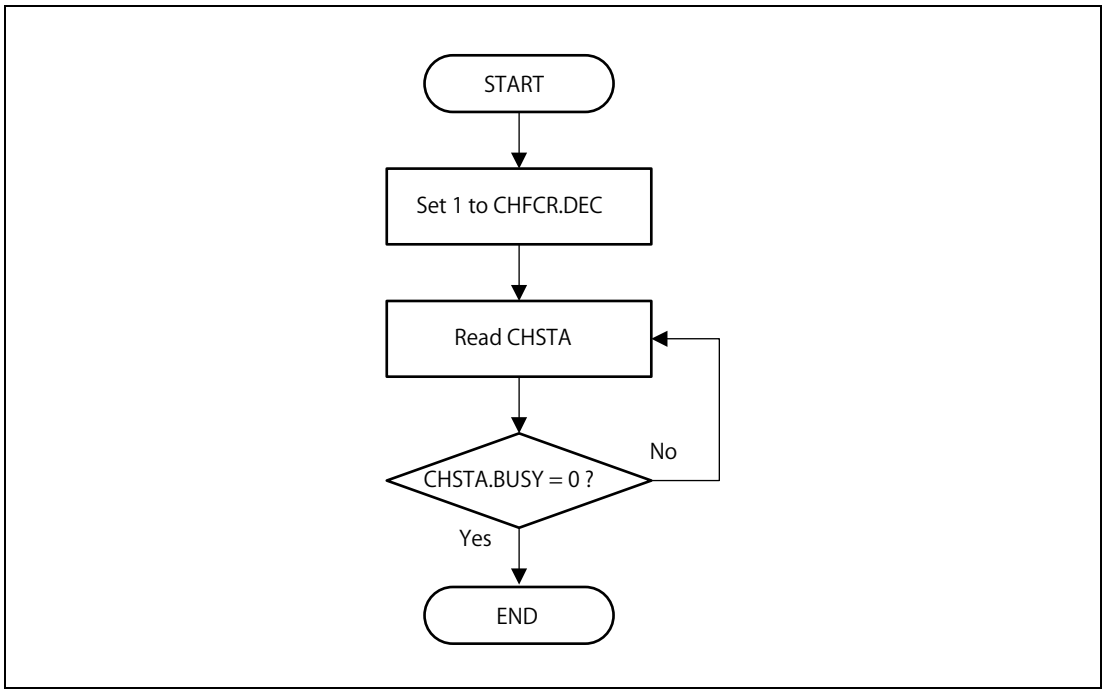


Figure 7.28 Example of DMA Transfer Abort Processing

(3) Suspend and Resume DMA Transfer

To suspend DMA transfer temporarily, write 1 to the STP bit in the DMA channel suspend register (DMAjCHSTP_n). After receiving the entire response packet, the BUSY bit is cleared to 0. To resume, set the STP bit to 0.

Figure 7.29 shows a processing example when suspending and resuming DMA transfer.

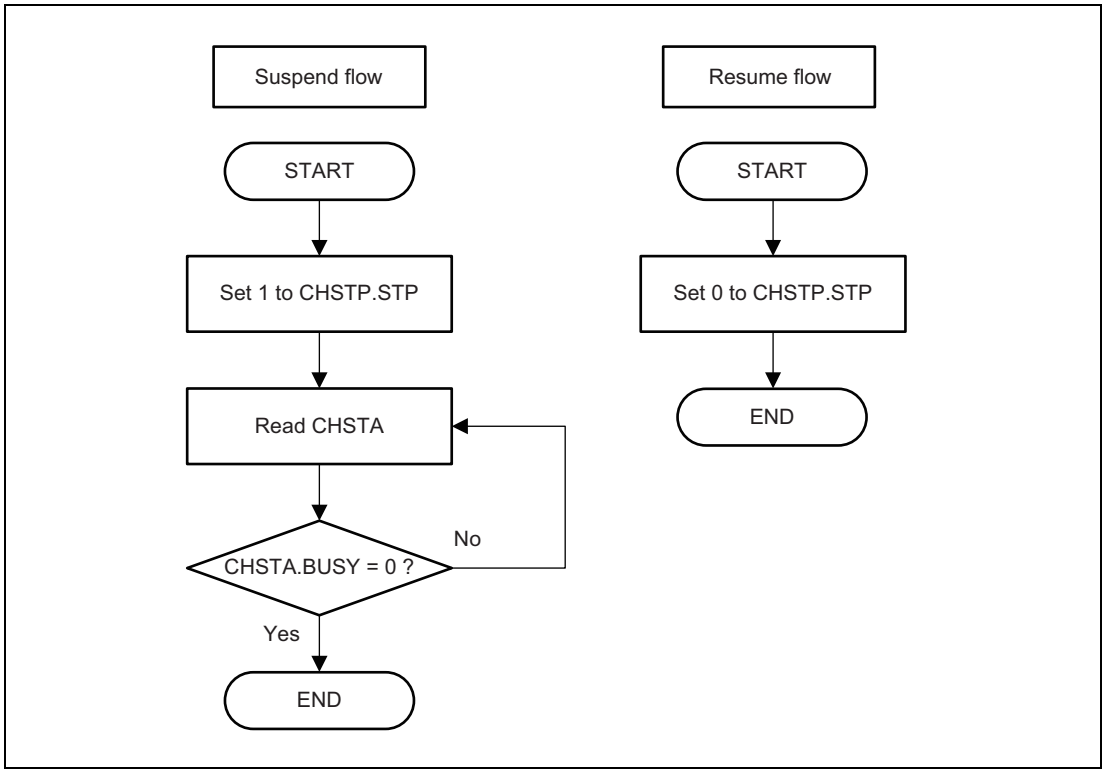


Figure 7.29 Example of DMA Transfer Suspend and Resume Processing

(4) Restriction and Impact on the Transfer Unit Size on Addresses and Offsets

The general recommendation is to set the destination address (DAR) and the scatter address increments (SIAI, SOAI) to a multiple of the related transfer unit size (DTS). If this recommendation is not followed, the byte transfers are used instead. This reduces the performance significantly.

In case of using the Gather Transfer function, the DMA Source Address Register (DMAjSAR_n), the DMA Gather Inner Address Increment Register (DMAjGIAI_n) and the DMA Gather Outer Address Increment Register (DMAjGOAI_n) must be set to multiples of DMA source transaction size which is set to STS bit in the DMAjTMR_n register.

7.6 Reliability Function

7.6.1 Overview

In this product, the DMAC is a resource used by multiple masters (CPU0 to CPU3). In order for the DMAC to support a multi-core configuration, the following reliability functions are offered.

- Register access protection function

  Refer to **Section 44, Functional Safety**.

- Master information inheritance function

7.6.2 Master Information Inheritance Function

In this product, DMA access inherits master information is set by DMAjCM_n register.

The master information that is output from the DMAC is as shown in **Table 7.29** for access guard.

Table 7.29 Master Information That Is Output from DMAC

Meaning	Value that is output from DMAC
UM	Same as the UM bit value in the channel master setting register
Secure	Non secure
SPID	Same as the SPID bit value in the channel master setting register

7.7 Setting up DMA Transfer

7.7.1 Overview of Setting up DMA Transfer

Table 7.30 Overview of Setting up DMA Transfer

No.	Description	Register		Necessity of the Setting
1	Transfer request group selection setting	DMACSELj_m	DMA transfer request group selection register m	Mandatory (if using hardware transfer request mode)
2	Overall DMA operation setting	DMAjCM_n	DMA channel master setting register	Mandatory
3	Channel setting	DMAjSAR_n	DMA source address register	Mandatory
4		DMAjDAR_n	DMA destination address register	Mandatory
5		DMAjTSR_n	DMA transfer size register	Mandatory
6		DMAjTMR_n	DMA transfer mode register	Mandatory
7		DMAjRS_n	DMA resource select register	Mandatory (if using hardware transfer request mode)
8	Status clear	DMAjCHFCR_n	DMA channel flag clear register	Mandatory
9	Channel operation enable	DMAjCHCR_n	DMA channel control register	Mandatory

NOTE

If a transfer request is used to change the transfer request group, it is necessary to set up the registers from the beginning without configuring an overall DMA operation setting.

7.7.2 Setting up the Transfer Request Group Selection

Set up the transfer request group selection before using the sDMAC. The following registers must be set up to configure a transfer request group.

- sDMACj Transfer Request Group Selection Register m (DMACSELj_m) (j = 0 to 1, m = 0 to 15)
  This register configures the transfer request group selection

7.7.3 Setting up the Overall DMA Operation

Set up the overall DMA operation before you start using DMA.

The following register must be set up to configure the overall DMA operation.

DMAC channel master setting register (DMAjCM_n)

This register configures the channel assignment. (For details, see **Section 7.6, Reliability Function**.)

If the DMAC channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

7.7.4 Setting up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC channel.

To configure the DMA channel setting, the master allowed to access each channel needs to set up the channel registers.

Follow the procedure below to set up the DMAC channel setting and use the DMAC.

(1) Disabling the DMAC channel operation

If the channel operation enable bit (DE) in the DMA channel control register (DMAjCHCR_n) is set, channel setting up registers must not to be set up. Wait for the previous DMA transfer to finish or be aborted, and then disable the DMAC channel operation.

(2) Setting up the transfer information

When setting up the transfer information for the DMAC, the following registers need to be set up.

- DMA source address register (DMAjSAR_n)
- DMA destination address register (DMAjDAR_n)
- DMA transfer size register (DMAjTSR_n)
- DMA transfer mode register (DMAjTMR_n)

(3) Setting up the DMA transfer request

While setting the transfer information, you need to set the DMA transfer request select (TRS) bit in the DMA transfer mode register (DMAjTMR_n) to define whether a hardware or software DMA transfer request is used.
The hardware and software DMA transfer request cannot be used for the same channel at the same time. If the hardware DMA transfer request is used, select one source to be used as the hardware DMA transfer request out of the 256 available hardware DMA transfer sources using the resource selection (RS) bit in the DMA resource select register (DMAjRS_n).
The DMA channel may retain a hardware DMA transfer request that came before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (DMACHSTA_n.DRQ) and overflow flag (DMACHSTA_n.OVF) by using the DMA channel flag clear register (DMAjCHFCR_n) before enabling channel operation.
If the software DMA transfer request is used, disable the hardware DMA transfer request select (TRS) bit in the DMA transfer mode register (DMAjTMR_n).

(4) Clearing the transfer status

The DMA channel status register (DMAjCHSTA_n) may retain the result of the previous DMA transfer. Clear the flags in the DMA transfer status register using the DMA channel flag clear register (DMAjCHFCR_n).

(5) Enabling the DMAC channel operation

Set the DMA channel operation enable bit (DE) in the DMA channel control register (DMAjCHCR_n) to enable the channel operation. After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and start DMA transfer.

Section 8 DTS Controller

This section provides a generic description of the Data Transfer Service (DTS) controller. The first part of this section contains information about product properties such as number of units and register base addresses. The second part of this section describes functions and registers of DTS.

8.1 Features of DTS

8.1.1 Number of Units and Channels

This product has the following number of DTS units.

Table 8.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Number of Channels	128	128	128	128	128	128	128	128	128	128
Name	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT	DTSCNT

Table 8.2 Definition of Indexes

Index	Description
nnn	The channel number is identified by the index “nnn”. (nnn = 0 to 127)
m	The individual trigger group selection registers are identified by the index “m” (m = 0 to 15).

8.1.2 Register Base Addresses

The DTS base addresses are listed in **Table 8.3**.

The DTS register addresses are specified as offsets from the base addresses.

Table 8.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<DTS_base>	FFF8 8000_H	P-Bus Group 0
<DMATRGSEL_base>	FF09 0400_H	P-Bus Group 9

8.1.3 Clock Supplies

The DTS clock supplies are shown in the following table.

Table 8.4 Clock Supplies

Unit Name	Clock for the Unit	Supply Clock Name
DTSCNT	clk	CLK_HBUS
DMATRGSEL	PCLK	CLK_HBUS

8.1.4 Interrupt Requests and Error Notifications

The DTS interrupt requests are listed in the following table.

Table 8.5 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Description	Interrupt Number
INTDTS31TO0	INTDTS[31:0]	DTS ch31-0 transfer end	39
INTDTS63TO32	INTDTS[63:32]	DTS ch63-32 transfer end	40
INTDTS95TO64	INTDTS[95:64]	DTS ch95-64 transfer end	41
INTDTS127TO96	INTDTS[127:96]	DTS ch127-96 transfer end	42
INTDTSCT31TO0	INTDTSCT[31:0]	DTS ch31-0 transfer count match	43
INTDTSCT63TO32	INTDTSCT[63:32]	DTS ch63-32 transfer count match	44
INTDTSCT95TO64	INTDTSCT[95:64]	DTS ch95-64 transfer count match	45
INTDTSCT127TO96	INTDTSCT[127:96]	DTS ch127-96 transfer count match	46
INTDTSERR	DTSERR	DTS transfer error	30

The error notifications of this module are listed in the following table.

Table 8.6 Error Notifications

Error Notification	Description	ECM error Number	Error Response to bus master
DTS/sDMAC Transfer Error	DTS transfer error	182	—

Note: This error is generated due to the error response caused by some error.
For details of the error response caused by some error, see the column for Error Response to bus master in Error Notifications Table shown in each section.

8.1.5 DTS Transfer Requests

The DTS transfer request table lists all DTS transfer request factors for DTS. The DTS transfer request group for DTS can be selected by the DTSSELm register. For details of List of DTS Trigger Sources, refer to Appendix file “DTS_Transfer_request_Table.xlsx”.

8.1.6 Reset Sources

DTS reset sources are listed in the following table. DTS is initialized by the following reset sources.

Table 8.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
DTSCNT	All registers	√	√	√	√	√	—	—
DTSSEL	All registers	√	√	√	√	√	—	—

8.1.7 External Input/Output Signals

This module has no signals mapped to external input/output pins.

8.2 Overview

8.2.1 Functional Overview

The Data Transfer Service (DTS) is a DMA controller used to access data without going through the CPU.

DTS stores transfer information in a dedicated RAM (DTSRAM).

Table 8.8 shows the configuration of DTS.

Table 8.8 Configuration of DTS

Module Name	Number of Units	Number of Channels
DTSCNT	1	128 ch

Functions of DTSFSL are arbitration of incoming the DTS requests, triggering the DTS controller and monitoring ongoing DTS transfers. DTSFSL can handle 128 DTS transfer sources.

The address space that can be used for DTS transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DTS, refer to **Section 4, Address Space**.

The DTS is connected to System Bus directly. If a conflict occurs with another bus master in System Bus, the transfer is arbitrated by round-robin arbitration.

8.2.2 Definition of Terms

Table 8.9 shows the terms used in this section.

Table 8.9 List of Term Definitions

Term	Description
DTS transfer	A term for data transfer carried out by the DTS controller
DTS cycle	A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DTS cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle.
Hardware DTS transfer source	A trigger for a DTS transfer request issued by an internal peripheral device
Hardware DTS transfer request	A DTS transfer request issued by a hardware DTS transfer source
Software DTS transfer request	A DTS transfer request issued by writing to a register via software
DTS transfer request	A trigger to start DTS transfer with DTS
Transfer information (TI)	The information required for DTS transfer, including the source address, destination address, transfer data size, and transfer count. The transfer information for DTS is referred to as TI.
DTSRAM	RAM used by DTS to store the transfer information
Single transfer	A DTS transfer consisting of one DTS cycle started by one DTS transfer request
Block transfer 1	A DTS transfer consisting of the number of DTS cycles specified by the transfer count in the transfer information, started by one DTS transfer request
Block transfer 2	A DTS transfer consisting of the number of DTS cycles specified by the address reload count in the transfer information, started by one DTS transfer request
Block transfer	A general term for both block transfer 1 and block transfer 2
Last transfer	The DTS cycle carried out when the transfer count in the transfer information is 1
Address reload transfer	The DTS cycle carried out when the address reload count in the transfer information is 1 if reload function 2 is used
Suspend	An action of pausing a DTS transfer during block transfer. The DTS transfers can be resumed after suspension.
Resume	An action of resuming a suspended DTS transfer
Transfer abort	An action of aborting a DTS transfer before it is finished. Aborted DTS transfers cannot be resumed.

8.3 Registers

8.3.1 List of Registers

Table 8.10 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					DTS/PBG	Other
DTSCNT	DTS control register 1	DTSCTL1	<DTS_base> + 0010_H	32	DTSGPROT_GR	—
	DTS control register 2	DTSCTL2	<DTS_base> + 0014_H	32	DTSGPROT_GR	—
	DTS status register	DTSSTS	<DTS_base> + 0018_H	32	DTSGPROT_GR	—
	DTS error register	DTSER	<DTS_base> + 0024_H	32	DTSGPROT_GR	—
	DTS channel priority setting 0	DTSPR0	<DTS_base> + 0060_H	32	DTSGPROT_GR	—
	DTS channel priority setting 1	DTSPR1	<DTS_base> + 0064_H	32	DTSGPROT_GR	—
	DTS channel priority setting 2	DTSPR2	<DTS_base> + 0068_H	32	DTSGPROT_GR	—
	DTS channel priority setting 3	DTSPR3	<DTS_base> + $006C_H$	32	DTSGPROT_GR	—
	DTS channel priority setting 4	DTSPR4	<DTS_base> + 0070_H	32	DTSGPROT_GR	—
	DTS channel priority setting 5	DTSPR5	<DTS_base> + 0074_H	32	DTSGPROT_GR	—
	DTS channel priority setting 6	DTSPR6	<DTS_base> + 0078_H	32	DTSGPROT_GR	—
	DTS channel priority setting 7	DTSPR7	<DTS_base> + $007C_H$	32	DTSGPROT_GR	—
	DTS channel nnn channel master setting*1	DTSnnnCM	<DTS_base> + “0200_H + 4_H × [DTS channel number] (0200_H to $03FC_H$)	32	DTSGPROT_GR	—
	DTS source address	DTSAnnn	<DTS_base> + 1000_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS destination address	DTDAnnn	<DTS_base> + 1004_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS transfer count	DTTCnnn	<DTS_base> + 1008_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS transfer control	DTTCTnnn	<DTS_base> + $100C_H$ + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS reload source address	DTRSAnnn	<DTS_base> + 1010_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS reload destination address	DTRDAnnn	<DTS_base> + 1014_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS reload transfer count	DTRTCnnn	<DTS_base> + 1018_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTS transfer count compare	DTTCCnnn	<DTS_base> + $101C_H$ + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTSFSL operation setting	DTFSLnnn	<DTS_base> + 1020_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTSFSL transfer status	DTFSTnnn	<DTS_base> + 1024_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTSFSL transfer request set	DTFSSnnn	<DTS_base> + 1028_H + 40_H × [channel number]	32	DTSGPROT_nnn	—
	DTSFSL transfer status clear	DTFSCnnn	<DTS_base> + $102C_H$ + 40_H × [channel number]	32	DTSGPROT_nnn	—
DMATRGSEL	DTS Transfer Request group selection	DTSSELm	<DMATRGSEL_base> + 200_H + (m × 4_H)	8, 16, 32	PBG90#4	—

Note 1. The [channel number] in the offset address is a number from 0 to 127. The “nnn” in the register symbols is a 3-digit number from 000 to 127.

Note 2. m index

8.3.2 Description of Global Registers

8.3.2.1 DTSCTL1 — DTS Control Register 1

DTS Control Register 1 (DTSCTL1) is a 32-bit read/write register that controls the DTS global suspension setting.

Access: DTSCTL1 register can be read or written in 32-bit units

Address: <DTS_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTSUST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.11 DTSCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DTSUST	DTS suspend This bit shows whether DTS transfers are suspended. If this bit is set to 1, a DTS suspend is requested. 0: DTS is not suspended 1: DTS is suspended or DTS suspension is requested

8.3.2.2 DTSCTL2 — DTS Control Register 2

DTS Control Register 2 (DTSCTL2) is a 32-bit write only register that controls DTS transfer abort.

Access: DTSCTL2 register is a write-only register that can be written in 32-bit units

Address: <DTS_base> + 0014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTSTIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 8.12 DTSCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DTSTIT	DTS transfer abort request Setting this bit to 1 while DTS transfer is suspended aborts the suspended DTS transfer. Writing to this bit while DTS transfer is in progress is ignored. When the suspended DTS transfer is aborted, the DTSSTS.DTSACT bit is cleared. This bit is write only. When read, 0 is returned.

8.3.2.3 DTSSTS — DTS Status Register

The DTS Status Register (DTSSTS) is a 32-bit read only register that indicates the DTS status.

Access: DTSSTS register is a read-only register that can be read in 32-bit units

Address: <DTS_base> + 0018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTSCYC	DTSACH[6:0]							DTSACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.13 DTSSTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8	DTSCYC	DTS cycle execution state This bit shows whether a DTS cycle is in progress in DTS. 0: DTS cycle is not in progress. 1: DTS cycle is in progress.
7 to 1	DTSACH[6:0]	DTS transfer channel If there is a channel in DTS executing a DTS transfer, the channel number is shown. If there is no channel in DTS executing a DTS transfer, the channel number of the last DTS transfer is shown.
0	DTSACT	DTS transfer status This bit shows whether there is a channel in DTS executing a DTS transfer. 0: There is no channel in DTS executing DTS transfer. 1: There is a channel in DTS executing DTS transfer. If DTS is in the suspended state while there is a channel executing DTS transfer, this bit remains 1. If a DTS transfer abort request is made using the DTSCTL2.DTSTIT bit, the suspended DTS transfer is aborted, and this bit is cleared to 0. When a DTS transfer error occurs and the DTS transfer is aborted, this bit is cleared.

8.3.2.4 DTSER — DTS Error Register

The DTS Error Register (DTSER) is 32-bit read only register that indicates the error status of DTS.

Access: DTSER is a read-only register that can be read in 32-bit units

Address: <DTS_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DTSERCH[6:0]							—	—	—	—	—	—	DTSERWR	DTSER
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.14 DTSER Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14 to 8	DTSERCH[6:0]	DTS error channel These bits show the DTS channel number of the smallest channel number that has a DTS transfer error. These bits are read-only and cannot be cleared.
7 to 2	Reserved	When read, the value after reset is returned.
1	DTSERWR	DTS transfer error cycle This bit indicates that a DTS transfer error occurred on the channel indicated by DTSERCH[6:0] in the read cycle or the write cycle. These bits are read-only and cannot be cleared. 0: A DTS transfer error occurred in the read cycle. 1: A DTS transfer error occurred in the write cycle.
0	DTSER	DTS transfer error flag This bit shows whether a DTS transfer error is detected in DTS. 0: A DTS transfer error was not detected in all DTS channels. 1: A DTS transfer error was detected in one of the DTS channels.

8.3.2.5 DTSPRn — DTS Channel Priority Setting Register (n = 0 to 7)

The DTS Channel Priority Setting Register (DTSPRn) is a 32-bit read/write register that defines the priorities for each DTS channel. The priorities are configured from 0 to 3, with 0 being the highest.

- **DTSPR0**

Access: DTSPR0 register can be read or written in 32-bit units

Address: <DTS_base> + 0060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS15PR[1:0]		DTS14PR[1:0]		DTS13PR[1:0]		DTS12PR[1:0]		DTS11PR[1:0]		DTS10PR[1:0]		DTS9PR[1:0]		DTS8PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS7PR[1:0]		DTS6PR[1:0]		DTS5PR[1:0]		DTS4PR[1:0]		DTS3PR[1:0]		DTS2PR[1:0]		DTS1PR[1:0]		DTS0PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.15 DTSPR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=0 to 15)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR1**

Access: DTSPR1 register can be read or written in 32-bit units

Address: <DTS_base> + 0064_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS31PR[1:0]		DTS30PR[1:0]		DTS29PR[1:0]		DTS28PR[1:0]		DTS27PR[1:0]		DTS26PR[1:0]		DTS25PR[1:0]		DTS24PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS23PR[1:0]		DTS22PR[1:0]		DTS21PR[1:0]		DTS20PR[1:0]		DTS19PR[1:0]		DTS18PR[1:0]		DTS17PR[1:0]		DTS16PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.16 DTSPR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=16 to 31)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR2**

Access: DTSPR2 register can be read or written in 32-bit units

Address: <DTS_base> + 0068_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS47PR[1:0]		DTS46PR[1:0]		DTS45PR[1:0]		DTS44PR[1:0]		DTS43PR[1:0]		DTS42PR[1:0]		DTS41PR[1:0]		DTS40PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS39PR[1:0]		DTS38PR[1:0]		DTS37PR[1:0]		DTS36PR[1:0]		DTS35PR[1:0]		DTS34PR[1:0]		DTS33PR[1:0]		DTS32PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.17 DTSPR2 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=32 to 47)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest

- **DTSPR3**

Access: DTSPR3 register can be read or written in 32-bit units

Address: <DTS_base> + $006C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS63PR[1:0]		DTS62PR[1:0]		DTS61PR[1:0]		DTS60PR[1:0]		DTS59PR[1:0]		DTS58PR[1:0]		DTS57PR[1:0]		DTS56PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS55PR[1:0]		DTS54PR[1:0]		DTS53PR[1:0]		DTS52PR[1:0]		DTS51PR[1:0]		DTS50PR[1:0]		DTS49PR[1:0]		DTS48PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.18 DTSPR3 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=48 to 63)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR4**

Access: DTSPR4 register can be read or written in 32-bit units

Address: <DTS_base> + 0070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS79PR[1:0]		DTS78PR[1:0]		DTS77PR[1:0]		DTS76PR[1:0]		DTS75PR[1:0]		DTS74PR[1:0]		DTS73PR[1:0]		DTS72PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS71PR[1:0]		DTS70PR[1:0]		DTS69PR[1:0]		DTS68PR[1:0]		DTS67PR[1:0]		DTS66PR[1:0]		DTS65PR[1:0]		DTS64PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.19 DTSPR4 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=64 to 79)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR5**

Access: DTSPR5 register can be read or written in 32-bit units

Address: <DTS_base> + 0074_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS95PR[1:0]		DTS94PR[1:0]		DTS93PR[1:0]		DTS92PR[1:0]		DTS91PR[1:0]		DTS90PR[1:0]		DTS89PR[1:0]		DTS88PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS87PR[1:0]		DTS86PR[1:0]		DTS85PR[1:0]		DTS84PR[1:0]		DTS83PR[1:0]		DTS82PR[1:0]		DTS81PR[1:0]		DTS80PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.20 DTSPR5 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=80 to 95)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR6**

Access: DTSPR6 register can be read or written in 32-bit units

Address: <DTS_base> + 0078_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS111PR[1:0]		DTS110PR[1:0]		DTS109PR[1:0]		DTS108PR[1:0]		DTS107PR[1:0]		DTS106PR[1:0]		DTS105PR[1:0]		DTS104PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS103PR[1:0]		DTS102PR[1:0]		DTS101PR[1:0]		DTS100PR[1:0]		DTS99PR[1:0]		DTS98PR[1:0]		DTS97PR[1:0]		DTS96PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.21 DTSPR6 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=96 to 111)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

- **DTSPR7**

Access: DTSPR7 register can be read or written in 32-bit units

Address: <DTS_base> + $007C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTS127PR[1:0]		DTS126PR[1:0]		DTS125PR[1:0]		DTS124PR[1:0]		DTS123PR[1:0]		DTS122PR[1:0]		DTS121PR[1:0]		DTS120PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTS119PR[1:0]		DTS118PR[1:0]		DTS117PR[1:0]		DTS116PR[1:0]		DTS115PR[1:0]		DTS114PR[1:0]		DTS113PR[1:0]		DTS112PR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.22 DTSPR7 Register Contents

Bit Position	Bit Name	Function
31 to 0	DTSnPR[1:0] (n=112 to 127)	DTS channel n priority setting These bits configure the priority level of each DTS channel used for DTS channel arbitration. 00 is the highest priority, and 11 is the lowest.

8.3.2.6 DTSnnnCM — DTS Channel Master Setting Register (nnn = 000 to 127)

The DTS Channel Master Setting Register (DTSnnnCM) is a 32-bit read/write register that defines the channel master configuration for each DTS channel. The contents of this register is used for the master information inheritance function. For more information about the function, see **Section 8.7, Reliability Function**.

This register is placed in the DTSRAM and the lower 16-bit field of this register is shared with the DTS transfer count compare register (DTTCCnnn), one of DTS channel register. If this register is written, DTTCCnnn register is updated as well.

Access: DTSnnnCM register can be read or written in 32-bit units

Address: <DTS_base> + "$0200_H + 4_H \times$ [DTS channel number] (0200_H to $03FC_H$)"

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHAIN_RESTRICT	CHAIN_SPID[4:0]					CHAIN_UM	—	—	SPID[4:0]					UM	—
Value after reset	Undefined*1															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMC[15:0]															
Value after reset	Undefined*1															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.23 DTSnnnCM Register Contents

Bit Position	Bit Name	Function
31	CHAIN_RESTRICT	Chain function restriction setting 0: The chain function is not restricted. A chain from any channel is possible. 1: The chain function is restricted. A chain is permitted only when SPID set by CHAIN_SPID4-0 is the same as DTSnnnCM.SPID4-0 of the current channel in the chain and UM set by CHAIN_UM is the same as DTSnnnCM.UM of the current channel in the chain.
30 to 26	CHAIN_SPID[4:0]	Chain permission SPID setting These bits specify the SPID information that permits the chain function.
25	CHAIN_UM	Chain permission UM setting This bit specifies the UM information that permits the chain function.
24 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 18	SPID[4:0]	Channel master SPID setting These bits specify the SPID information used by the master assigned to the channel.
17	UM	Channel master UM setting This bit specifies the UM information of the master assigned to the channel.
16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CMC[15:0]	Transfer count compare In terms of contents, this field is the same as bits[15:0] in **(8), DTTCCnnn — DTS Transfer Count Compare Register**.

Note: DTS000CM to DTS127CM configure the channel master information of DTS channels 0 to 127 respectively. For information about the functions this register provides, see **Section 8.7, Reliability Function**.
The lowest 16 bits of this register are shared with the DTS Transfer Count Compare Register, one of DTS channel registers.
When writing to this register, the DTS Transfer Count Compare Register is updated as well.
Bits 24, 23 and 16 of this register are reserved, but these bits are readable and writable. It is recommended to write 0 to these bits and to ignore the value read from these bits.

8.3.3 Description of DTS Channel Registers

8.3.3.1 Transfer Information of DTS (TI)

(1) Structure of TI

The transfer information of DTS is called TI. One TI consists of 8 sets which are called TI-A, TI-B, TI-C, TI-D, TI-E, TI-F, TI-G, and TI-H. Each set consists of a 32-bit word. Eight sets of TI are assigned to each channel.

The following figure shows the structure of TI.

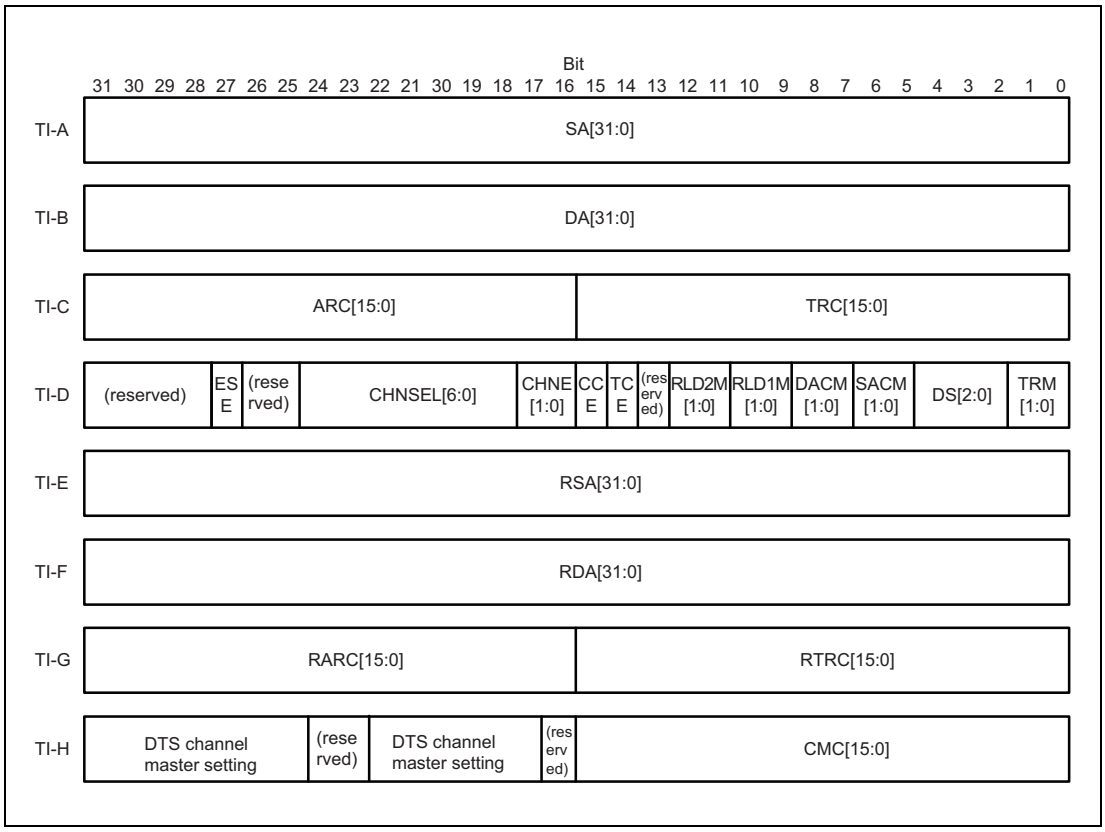


Figure 8.1 Structure of TI

(2) Organization of TI in DTSRAM

The user can indirectly access DTSRAM through the DTS channel registers for each channel and DTS channel master setting registers.

Therefore, direct access to TI structures in DTSRAM is usually not required.

As an exception, when an ECC error occurs while DTSRAM is being accessed, the address at which error occurred is stored in the DTSRAM 1st 1-bit Error Address Register (DR_00SEADR) or the DTSRAM 1st Fatal Error Address Register (DR_00DEADR). It is helpful to understand the address organization of TI in DTSRAM to distinguish which channel/TI has generated the error.

Figure 8.2 shows the address organization of TI in DTSRAM.

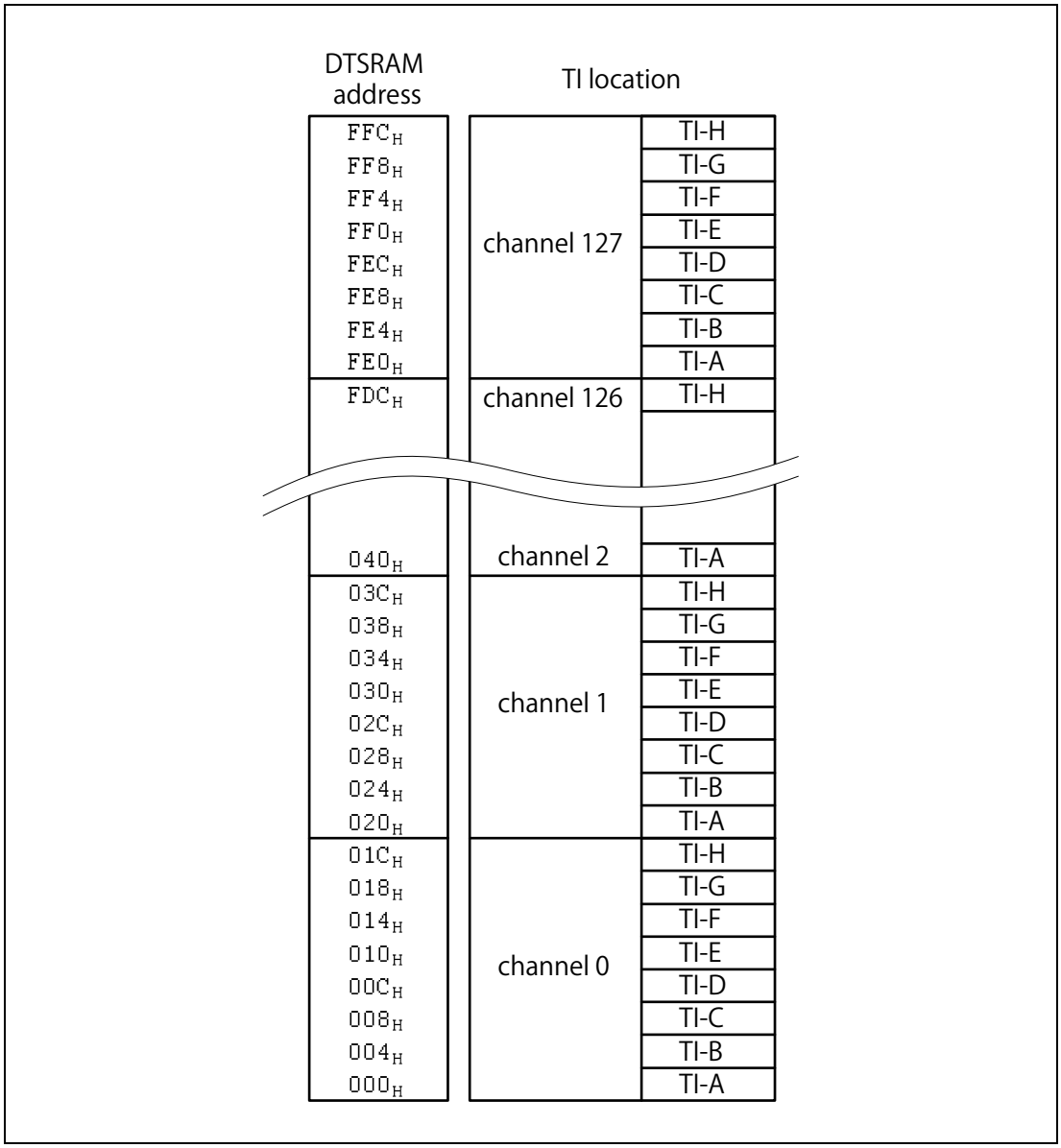


Figure 8.2 Organization of TI in DTSRAM

(3) How to Access TI

TI-A can be accessed through the DTS Source Address Register (DTSAnnn) for each channel.

TI-B can be accessed through the DTS Destination Address Register (DTDAnnn) for each channel.

TI-C can be accessed through the DTS Transfer Count Register (DTTCnnn) for each channel.

TI-D can be accessed through the DTS Transfer Control Register (DTTCTnnn) for each channel.

TI-E can be accessed through the DTS Reload Source Address register (DTRSAnnn) for each channel.

TI-F can be accessed through the DTS Reload Destination Address register (DTRDAnnn) for each channel.

TI-G can be accessed through the DTS Reload Transfer Count Register (DTRTCnnn) for each channel.

TI-H can be accessed through either the global Channel Master Setting Register (DTSnnnCM) or the channel specific Transfer Count Compare Register (DTTCCnnn).

(4) Cautions on Accessing TI

The value of the DTS Channel Master Setting Register and the value of the DTS Transfer Count Compare Register are both stored in TI-H.

Accessing the DTS Channel Master Setting Register (DTSnnnCM) is performed via a 32-bit access to the entire TI-H.

When writing to the DTS Channel Master Setting Register, the lower 16 bits (DTS transfer count compare or CMC) are also updated. When reading from the DTS Channel Master Setting Register, the lower 16 bits represent the CMC value.

When reading from the DTS Transfer Count Compare Register (DTTCCnnn), hardware performs a 32-bit read access from TI-H but only the lower 16 bits are used for the result of the register read. When writing to the DTS Transfer Count Compare Register (DTTCCnnn), hardware performs a read-modify-write access to TI-H.

Since the contents of TI after reset are undefined, writing to the DTS Transfer Count Compare Register (DTTCCnnn) before writing to the DTS Channel Master Setting Register can result in an ECC error due to the read part of the read-modify-write access.

Some bits of TI-H are reserved. It is recommended to write 0 to these bits and to ignore the value read from these bits.

After reset, the values of TI in DTSRAM are undefined. The first access to TI after reset should be a write access. Otherwise an ECC error will occur.

- DTS Source Address Register (DTSAnnn)
- DTS Destination Address Register (DTDAnnn)
- DTS Transfer Count Register (DTTCnnn)
- DTS Transfer Control Register (DTTCTnnn)
- DTS Reload Source Address Register (DTRSAnnn)
- DTS Reload Destination Address Register (DTRDAnnn)
- DTS Reload Transfer Count Register (DTRTCnnn)
- Channel Master Setting Registers (DTSnnnCM)

After reset a write access to the DTS Transfer Count Compare Register (DTTCCnnn) must follow a write access to the DTS Channel Master Setting Register (DTSnnnCM).

When accessing TI contents while a DTS transfer is in progress, the following implications should be considered.

- TI of the channel should not be updated by the CPU during the ongoing transfer. Otherwise a mismatch between the DTS transfer and the contents of TI may occur.
- If a TI access is requested from the CPU while a TI fetch or TI write back is being executed, the TI access is executed after the completion of the TI fetch or TI write back.
- If a TI fetch or TI write back is requested while a TI access request from the CPU is being processed, the TI fetch or TI write back is executed after the completion of the TI access.

8.3.3.2 Details of DTS Channel Registers

The “nnn” in the register symbols indicates the DTS channel number (nnn = 000 to 127).

(1) DTSAnnn — DTS Source Address Register

The DTS Source Address Register (DTSAnnn) is a 32-bit read/write register that specifies the DTS transfer source address of each channel.

Access: DTSAnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1000_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$ *1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SA[31:16]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.24 DTSAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	SA[31:0]	Source address These bits specify the DTS transfer source address. This register is updated at the timing of TI writeback and retains the DTS transfer source address for the next DTS transfer.

NOTE

DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	SA3	SA2	SA1	SA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

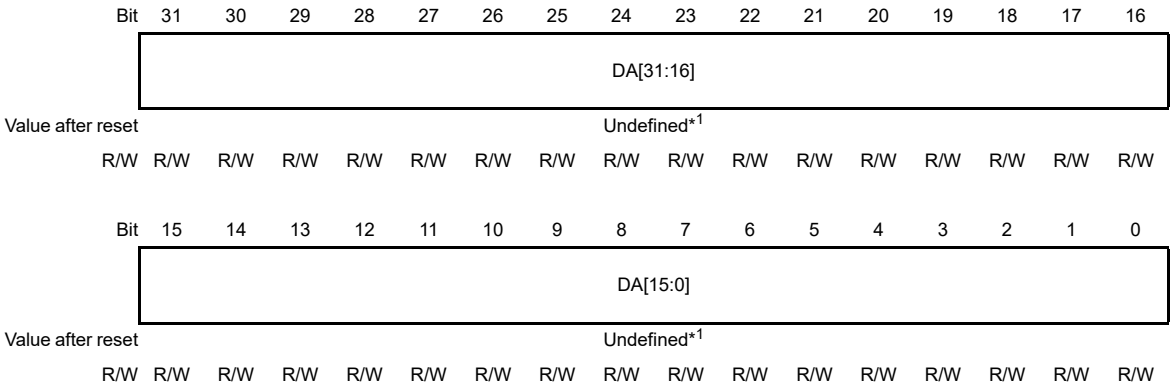
(2) DTDAnnn — DTS Destination Address Register

The DTS Destination Address Register (DTDAnnn) is a 32-bit read/write register that specifies the DTS transfer destination address of each channel.

Access: DTDAnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1004_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.



Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.25 DTDAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	DA[31:0]	Destination address These bits specify the DTS transfer destination address. This register is updated at the timing of TI writeback and retains the DTS transfer destination address for the next DTS transfer.

NOTES

1. If a DTS transfer error occurs in the read cycle of a DTS transfer, the write cycle is not executed, but the destination address is updated.
2. DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	DA3	DA2	DA1	DA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(3) DTTCnnn — DTS Transfer Count Register

The DTS Transfer Count Register (DTTCnnn) is a 32-bit read/write register that specifies the DTS transfer count of each channel.

Access: DTTCnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1008_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARC[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRC[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.26 DTTCnnn Register Contents

Bit Position	Bit Name	Function
31 to 16	ARC[15:0]	Address reload count These bits specify the transfer count for the address reload function. The number is used for both reload function 2 and block transfer 2. When reload function 2 or block transfer 2 is used, ARC[15:0] are decremented by one for every DTS cycle and updated at the timing of TI writeback. When neither reload function 2 nor block transfer 2 is used, ARC[15:0] are not updated. If ARC[15:0] are 0000_H, address reload is not performed and ARC[15:0] are not updated.
15 to 0	TRC[15:0]	Transfer count These bits configure the transfer count. TRC[15:0] are decremented by one whenever a DTS cycle is executed. These bits are updated at the timing of TI writeback. If the reload function is not used, the value at completion (0000_H) is retained after the completion of the last transfer. If 0000_H is set, DTS transfer is not executed even if a DTS transfer request is accepted.

TRC[15:0]	Operation
0000_H	Transfer is disabled or complete.
0001_H	The number of transfers or remaining transfers is 1.
:	:
$FFFF_H$	The number of transfers or remaining transfers is 65535.

NOTE

If a transfer error occurs in the read cycle of DTS transfer, the write cycle is not executed, but the transfer count and address reload count are updated.

(4) DTTCTnnn — DTS Transfer Control Register

The DTS Transfer Control Register (DTTCTnnn) is a 32-bit read/write register that defines DTS transfer control configuration of the channel.

Access: DTTCTnnn register can be read or written in 32-bit units

Address: <DTS_base> + $100C_H$ + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ESE	—	—	CHNSEL[6:0]							CHNE[1:0]	
Value after reset								Undefined*1								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCE	TCE	—	RLD2M[1:0]		RLD1M[1:0]		DACM[1:0]		SACM[1:0]		DS[2:0]			TRM[1:0]	
Value after reset								Undefined*1								
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.27 DTTCTnnn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value returned is undefined. The write value should be 0.
27	ESE	Setting of DTS error transfer disable This bit specifies whether to abort DTS transfer when a DTS transfer error occurs. If this bit is cleared to 0, DTS transfer continues when a DTS transfer error occurs. If this bit is set to 1, DTS transfer is aborted when a DTS transfer error occurs. 0: DTS transfer continues when a DTS transfer error occurs. 1: DTS transfer is aborted when a DTS transfer error occurs.
26 to 25	Reserved	When read, the value returned is undefined. The write value should be 0.
24 to 18	CHNSEL[6:0]	Next channel to chain These bits specify the next channel to chain. The next channel must be another channel in DTS. Setting the same channel for the next channel is not allowed. (Correct operation is not guaranteed in this case.)
17 to 16	CHNE[1:0]	Chain enable This bit selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DTS cycle at which the remaining transfer count is 1. 10: (Setting prohibited) 11: Always chain A chain request is generated at the completion of every DTS cycle.
15	CCE	Transfer count match interrupt enable If this bit is set, a transfer count match interrupt is generated at the completion of the DTS cycle in which the Transfer Count Compare Register and remaining transfer count have the same value.
14	TCE	Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer.

Table 8.27 DTTCTnnn Register Contents (2/2)

Bit Position	Bit Name	Function
13	Reserved	When read, the value returned is undefined. The write value should be 0.
12, 11	RLD2M[1:0]	Reload function 2 setting These bits configure reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DTS cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DTS cycle in which address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DTS cycle in which address reload count is 1.
10 to 9	RLD1M[1:0]	Reload function 1 setting These bits configure reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DTS cycle in which remaining transfer count is 1. (If reload function 2 is enabled, the address reload count is also reloaded.)
8 to 7	DACM[1:0]	Destination address count direction These bits specify the count direction of the destination address. 00: Increment 01: Decrement 10: Fixed 11: Prohibited (Operation not guaranteed)
6 to 5	SACM[1:0]	Source address count direction These bits specify the count direction of the source address. 00: Increment 01: Decrement 10: Fixed 11: Prohibited (Operation not guaranteed)
4 to 2	DS[2:0]	Transfer data size These bits specify the transfer data size. 000: 8 bits 001: 16 bits 010: 32 bits 011: 64 bits 100: 128 bits Other than the above: Prohibited (Operation not guaranteed)
1 to 0	TRM[1:0]	Transfer mode These bits specify the DTS transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Prohibited (Operation not guaranteed)

NOTES

1. If prohibited settings are used for some of the bits, the correct operation is not guaranteed.
2. Some bits in this register are unused, but these bits are readable and writable. It is recommended to write 0 to these bits and to ignore the value read from these bits.

(5) DTRSAnnn — DTS Reload Source Address Register

The DTS Reload Source Address Register (DTRSAnnn) is a 32-bit read/write register that specifies the DTS reload source address of each channel.

Access: DTRSAnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1010_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA[31:16]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSA[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.28 DTRSAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	RSA[31:0]	Reload source address These bits specify the source address to be reloaded when reload function 1 or reload function 2 is used.

NOTE

DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	RSA3	RSA2	RSA1	RSA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(6) DTRDAnnn — DTS Reload Destination Address Register

The DTS Reload Destination Address Register (DTRDAnnn) is a 32-bit read/write register that specifies the DTS reload destination address of each channel.

Access: DTRDAnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1014_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA[31:16]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDA[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.29 DTRDAnnn Register Contents

Bit Position	Bit Name	Function
31 to 0	RDA[31:0]	Reload destination address These bits specify the destination address to be reloaded when reload function 1 or reload function 2 is used.

NOTE

DTS transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size are as follows. The correct operation is not guaranteed if any other setting is specified.

Data Size	RDA3	RDA2	RDA1	RDA0
8 bits	0/1	0/1	0/1	0/1
16 bits	0/1	0/1	0/1	0
32 bits	0/1	0/1	0	0
64 bits	0/1	0	0	0
128 bits	0	0	0	0

(7) DTRTCnnn — DTS Reload Transfer Count Register

The DTS Reload Transfer Count Register (DTRTCnnn) is a 32-bit read/write register that specifies the DTS reload transfer count of each channel.

Access: DTRTCnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1018_H + 40_H × [channel number]

Value after reset: 0000 0000_H*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RARC[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTRC[15:0]															
Value after reset	Undefined*1															
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on the setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.30 DTRTCnnn Register Contents

Bit Position	Bit Name	Function
31 to 16	RARC[15:0]	Reload address reload count These bits specify the value to be reloaded to the address reload count when reload function 2 is used.
15 to 0	RTRC[15:0]	Reload transfer count These bits specify the value to be reloaded to the transfer count when reload function 1 is used.

RTRC[15:0]	Operation
0000_H	No DTS transfer
0001_H	1 transfer
:	:
$FFFF_H$	65535 transfers

(8) DTTCCnnn — DTS Transfer Count Compare Register

The DTS Transfer Count Compare Register (DTTCCnnn) is a 32-bit read/write register which configures the transfer count to be compared to transfer count register.

Access: DTTCCnnn register can be read or written in 32-bit units

Address: <DTS_base> + $101C_H$ + 40_H × [channel number]

Value after reset: $0000\ 0000_H$*1 RAM initialization is executed.
Undefined*1 RAM initialization is not executed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMC[15:0]															
Value after reset	Undefined*1															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The value after reset is dependent on setting of RAM initialization. See **Section 9.5.6, RAM Initialization** for details.

Table 8.31 DTTCCnnn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value returned is undefined. The write value should be 0.
15 to 0	CMC[15:0]	Transfer count compare These bits configure the transfer count to be compared to the Transfer Count Register. If the transfer count match interrupt enable (DTTCTnnn.CCE) bit is 1, the transfer count match interrupt is generated at the completion of the DTS cycle in which remaining transfer count is the same as the value in this register. If CMC[15:0] is 0000_H, comparison with the transfer count is disabled. In this case, the transfer count match interrupt is not generated.

Note: This register must be accessed after the DTS channel master setting register is set up. If this register is accessed without setting up the DTS channel master setting register after reset, an ECC error may occur.

(9) DTFSLnnn — DTSFSL Operation Setting Register

The DTSFSL Operation Setting Register (DTFSLnnn) is a 32-bit read/write register that controls the DTS transfer request of each channel.

Access: DTFSLnnn register can be read or written in 32-bit units

Address: <DTS_base> + 1020_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REQEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.32 DTFSLnnn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	REQEN	DTS transfer request enable This bit specifies whether a DTS transfer request of each channel is used as a candidate for DTS channel arbitration. 0: The DTS transfer request is not used as a candidate for DTS channel arbitration. 1: The DTS transfer request is used as a candidate for DTS channel arbitration. If this bit is 0, the request of the channel is not a candidate of DTS channel arbitration inside DTSFSL. DTS transfer request from the channel is not generated consequently, even if DTSFSL holds a request.

(10) DTFSTnnn — DTSFSL Transfer Status Register

The DTSFSL Transfer Status Register (DTFSTnnn) is a 32-bit read-only register that indicates the status of each channel.

Access: DTFSTnnn register is a read-only register that can be read in 32-bit units

Address: <DTS_base> + 1024_H + 40_H × [channel number]

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ERWR	—	—	—	ER	—	CC	TC	—	—	—	DRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.33 DTFSTnnn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	ERWR	DTS transfer error cycle When transfer error flag (ER) is set, this bit is also updated. This bit indicates whether a DTS transfer error has occurred in the read cycle or write cycle. If the ER bit is already set, this bit is not updated when another DTS transfer error occurs. This bit is also cleared when the ER bit is cleared. 0: A DTS transfer error occurred in the read cycle. 1: A DTS transfer error occurred in the write cycle
10 to 8	Reserved	When read, the value after reset is returned.
7	ER	Transfer error flag This bit is set when a DTS transfer error occurs. If this bit is 1 and the DTTCTnnn.ESE bit is set, a DTS transfer is not executed when a DTS transfer request occurs. This bit is cleared by writing 1 to the DTFSCnnn.ERC bit. 0: No DTS transfer error has occurred 1: A DTS transfer error has occurred
6	Reserved	When read, the value after reset is returned.
5	CC	Transfer count match flag This bit is set at the completion of the DTS transfer in which the remaining transfer count is the same as the value set in the transfer compare register. This bit is cleared by writing 1 to the DTFSCnnn.CCC bit. 0: No compare match with the Transfer Count Compare Register has occurred. 1: A compare match with the Transfer Count Compare Register has occurred.
4	TC	Transfer completion flag This bit is set at the completion of the last transfer and shows whether DTS transfer is complete. This bit is cleared by writing 1 to the DTFSCnnn.TCC bit. 0: DTS transfer is not complete 1: DTS transfer is complete
3 to 1	Reserved	When read, the value after reset is returned.

Table 8.33 DTFSTnnn Register Contents (2/2)

Bit Position	Bit Name	Function
0	DRQ	DTS transfer request flag This bit shows whether a DTS transfer request of this channel is being held. This bit is set when DTS transfer request is detected, or when the DTFSSnnn.DRQS bit is written by software. This bit is automatically cleared when DTS accepts a transfer request while DTSFSL is requesting DTS transfer for the channel in question. This bit also can be cleared by writing to the DTFSCnnn.DRQC bit. 0: A DTS transfer request was not detected. 1: A DTS transfer request was detected and is being held in DTSFSL.

(11) DTFSSnnn — DTSFSL Transfer Request Set Register

The DTSFSL Transfer Request Set Register (DTFSSnnn) is 32-bit write-only register that sets the DRQ bit in the DTFSTnnn register.

Access: DTFSSnnn register is a write-only register that can be written in 32-bit units

Address: <DTS_base> + 1028_H + 40_H × [channel number]

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 8.34 DTFSSnnn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DRQS	DTS transfer request set Writing 1 to this bit will generate a DTS transfer request. 0 is always read from this bit.

(12) DTFSCnnn — DTSFSL Transfer Status Clear Register

The DTSFSL Transfer Status Clear Register (DTFSCnnn) is 32-bit write-only register that clears the status flag bits in the DTFSTnnn register.

Access: DTFSCnnn register is a write-only register that can be written in 32-bit units

Address: <DTS_base> + $102C_H$ + 40_H × [channel number]

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ERC	—	CCC	TCC	—	—	—	DRQC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	W	W	R	R	R	W

Table 8.35 DTFSCnnn Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7	ERC	Transfer error flag (ER) clear The DTFSTnnn.ER bit can be cleared by writing 1 to this bit. 0 is always read from this bit.
6	Reserved	When writing, write the value after reset.
5	CCC	Transfer count match flag (CC) clear The DTFSTnnn.CC bit can be cleared by writing 1 to this bit. 0 is always read from this bit.
4	TCC	Transfer completion flag (TC) clear The DTFSTnnn.TC bit can be cleared by writing 1 to this bit. 0 is always read from this bit.
3 to 1	Reserved	When writing, write the value after reset.
0	DRQC	DTS transfer request clear The DTFSTnnn.DRQ bit can be cleared by writing 1 to this bit. 0 is always read from this bit.

8.3.4 Detail DTSSELm

8.3.4.1 DTSSELm — DTS Transfer Request Group Selection Register (m = 0 to 15)

DTSSELm register specifies the DTS transfer request group for each channel.

Access: DTSSELm register can be read or written in 8-, 16-, or 32-bit units

Address: <DMATRGSEL_base> + 200_H + (m × 4_H)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SEL7[1:0]		—	—	SEL6[1:0]		—	—	SEL5[1:0]		—	—	SEL4[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEL3[1:0]		—	—	SEL2[1:0]		—	—	SEL1[1:0]		—	—	SEL0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 8.36 DTSSELm Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	SEL7[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 7.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	SEL6[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 6.
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	SEL5[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 5.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	SEL4[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 4.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	SEL3[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 3.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	SEL2[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 2.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	SEL1[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 1.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SEL0[1:0]	These bits select the DTS transfer request group for DTS channel 8 x m + 0.

Bit Name	Function
SELn[1:0]	00_B: Select DTS transfer request group 0 01_B: Select DTS transfer request group 1 10_B: Select DTS transfer request group 2 11_B: Select DTS transfer request group 3

Figure 8.3 shows DTS transfer request group selection.

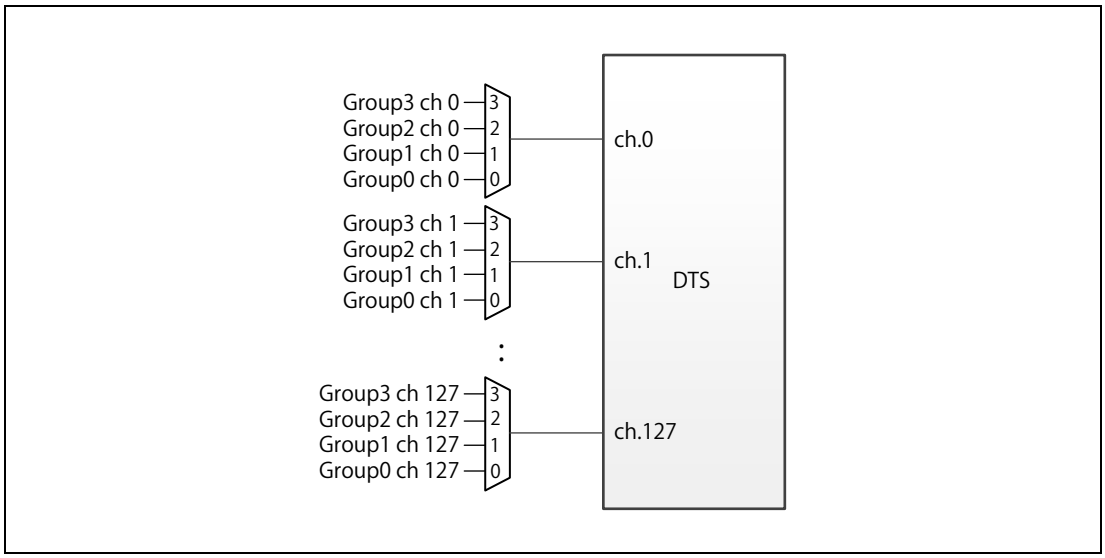


Figure 8.3 DTS Transfer Request Group Selection

8.4 Operation

8.4.1 Basic Operation of DTS Transfer

8.4.1.1 Transfer Mode

DTS has three transfer modes.

(1) Single Transfer

One DTS cycle is executed when a DTS transfer request is accepted.

(2) Block Transfer 1

The number of DTS cycles specified by the TRC bits in the DTS Transfer Count Register are executed when a DTS transfer request is accepted.

(3) Block Transfer 2

The number of DTS cycles specified by the ARC bits in the DTS Transfer Count Register are executed when a DTS transfer request is accepted. If the address reload count (ARC) is larger than the value of the transfer count (TRC), the number of DTS cycles is defined as the value of TRC.

8.4.1.2 Executing a DTS Cycle

DTS always executes the write cycle after the read cycle is complete. For example, if the transfer data size is 128 bits, the write cycle is executed after the read cycle for the 128-bit data is finished. The write cycle never starts before the read cycle is completed.

8.4.1.3 Updating Transfer Information

When a DTS cycle is executed, the DTS updates the transfer information as follows.

(1) Source Address and Destination Address

Transfer information (TI) will be updated as described in **Table 8.37** according to the settings of the DTS Transfer Control Register (DTTCTnnn) such as the source address count direction, the destination address count direction, and the transfer data size.

Table 8.37 Updating the Source Address and the Destination Address

Direction of Count	Transfer Data Size	Address after Update
Increment	8 bit	(address before update) + 0000 0001_H
	16 bit	(address before update) + 0000 0002_H
	32 bit	(address before update) + 0000 0004_H
	64 bit	(address before update) + 0000 0008_H
	128 bit	(address before update) + 0000 0010_H
Decrement	8 bit	(address before update) − 0000 0001_H
	16 bit	(address before update) − 0000 0002_H
	32 bit	(address before update) − 0000 0004_H
	64 bit	(address before update) − 0000 0008_H
	128 bit	(address before update) − 0000 0010_H
Fixed	—	Same as the address before update.

When the reload function is used, a specific update rule is applied other than the one described in **Table 8.37** for the last transfer and the address reload transfer. For details, see **Section 8.4.3, Reload Function**.

(2) Transfer Count/Address Reload Count

The transfer count is decremented by one for every DTS cycle.

The address reload count is decremented by one for every DTS cycle when reload function 2 or block transfer 2 is used. When neither reload function 2 nor block transfer 2 is used, it is not updated.

If the reload function is used, a specific update rule is applied for the last transfer and the address reload transfer.

For details, see **Section 8.4.3, Reload Function**.

(3) Other Transfer Information

Other transfer information is not updated during execution of a DTS cycle.

8.4.1.4 Last Transfer and Address Reload Transfer

The last transfer means the DTS cycle executed when the value in the Transfer Count Register, which shows the remaining number of transfers, is 1. The last transfer differs in operation compared to the other DTS cycles as follows.

- The transfer completion flag (DTFSTnnn.TC) is set when the last transfer is complete.
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When reload function 1 is enabled, reload function 1 is executed at the timing of the last transfer. For details, see **Section 8.4.3, Reload Function**.

Address reload transfer means the DTS cycle executed when reload function 2 is enabled and the address reload count is 1. Reload function 2 is executed after address reload transfer. For details, see **Section 8.4.3, Reload Function**.

8.4.1.5 Transfer Completion Interrupt and Transfer Count Match Interrupt Output

DTS can output a transfer completion interrupt and a transfer count match interrupt to outside DTS.

(1) Transfer Completion Interrupt Output

When the transfer completion interrupt output enable bit (DTTCTn.TCE) is set in the Transfer Control Register, DTS requests a DTS transfer completion interrupt when the last transfer is complete.

(2) Transfer Count Match Interrupt Output

When the transfer count match interrupt enable bit (DTTCTn.CCE) is set in the Transfer Control Register, DTS requests a DTS transfer count match interrupt at the completion of the DTS cycle in which the Transfer Count Compare Register and the transfer count have the same value.

Figure 8.4 shows the operation of the transfer completion interrupt and the transfer count match interrupt.

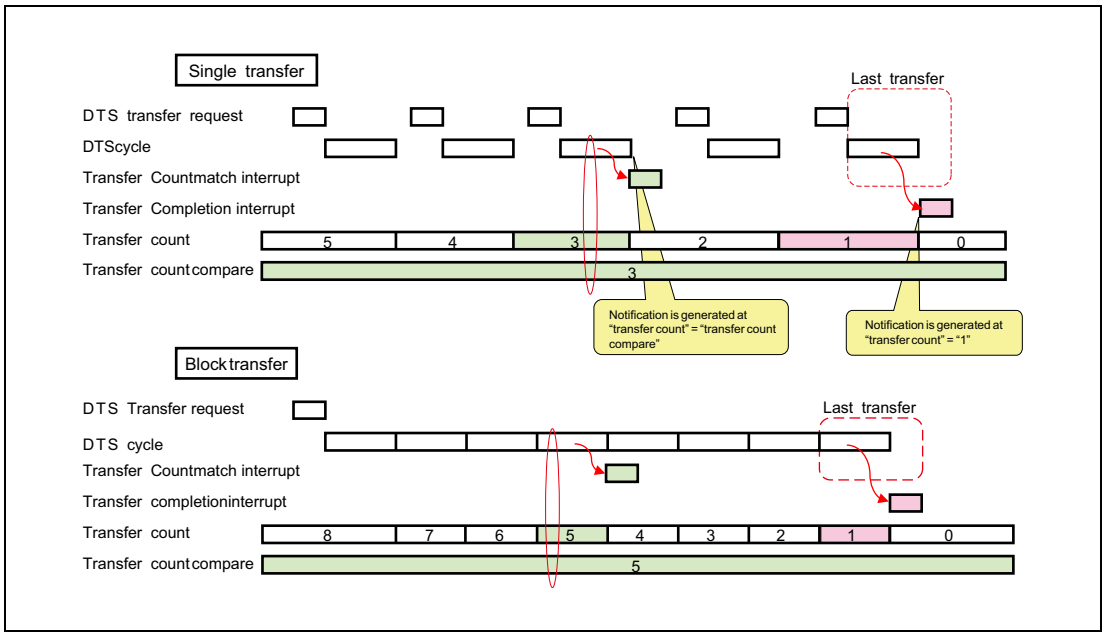


Figure 8.4 Transfer Completion Interrupt and Transfer Count Match Interrupt

8.4.1.6 Continuous Transfer

DTS does not have a configuration that supports continuous transfer.

DTS does not start DTS transfer when a DTS transfer request is generated while the transfer count is 0.

If the reload function 1 is used and the transfer count with the value other than 0 is reloaded to the transfer count bits (TRC[15:0]) in the DTS Transfer Count Register (DTTCnnn) when the last transfer is complete, the DTS can start the transfer after the next DTS transfer request is accepted.

8.4.2 Channel Priority Order

This subsection explains arbitration between multiple DTS channels.

8.4.2.1 DTS Channel Arbitration

If there are DTS transfer requests from multiple DTS channels, DTSFSL arbitrates those DTS channels. For each DTS channel, a priority can be selected from four levels using the DTS channel priority setting registers.

If there are DTS transfer requests from multiple DTS channels, arbitration is executed as follows.

The channel with the higher priority level in the setting of the DTS channel priority setting register has priority.

If two channels have the same priority level specified in the DTS channel priority setting register, the channel with the smaller channel number has priority.

DTSFSL sends DTS a DTS transfer request for the channel selected by arbitration. DTS executes DTS transfer when it accepts the DTS transfer request.

DTS transfer does not allow arbitration between DTS channels in the middle of a block transfer. If a higher priority DTS transfer request is issued during a lower priority block transfer, the higher priority DTS transfer has to wait until the current block transfer is complete*1.

Note 1. Completion of block transfer means either the last transfer or the last transfer for block transfer 1 or the address reload transfer for block transfer 2 is finished.

When the DTS is executing the block transfer 1 or the block transfer 2, a DTS cycle of a DTS channel with a higher priority does not take over the ongoing block transfer until the last transfer is completed.

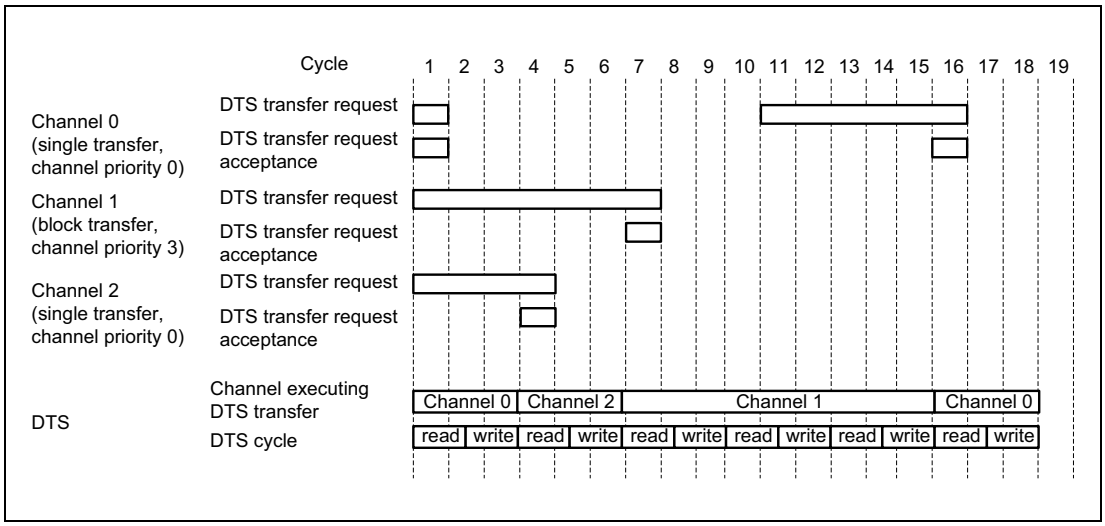


Figure 8.5 DTS Channel Arbitration

Cycle numbers shown in **Figure 8.5** are for explanation purposes only. They do not indicate the actual number of cycles necessary for executing DTS transfer.

In **Figure 8.5**, DTS transfer requests for channels 0, 1 and 2 are generated at Cycle 1. The channel priority for channels 0 and 2 is 0 and is higher than the channel priority for channel 1, which is 3. In addition, if two channels have the same priority, the channel with the smaller channel number has higher priority. Consequently, the priority order for arbitration is “channel 0 > channel 2 > channel 1”, and a DTS cycle for channel 0 starts because its priority is the highest. As a result of arbitration between channels 1 and 2 at cycle 4, a DTS cycle for channel 2 starts. A DTS cycle for channel 1 starts at Cycle 7, and continuous cycles follow until Cycle 15 because channel 1 uses block transfer. At Cycle 11, a DTS transfer request for channel 0 is generated. The DTS cycle for channel 1 is still ongoing and no arbitration is executed until the block transfer of channel 1 is complete.

At cycle 15, the block transfer of channel 1 is complete. At cycle 16, a DTS cycle for channel 0 starts.

8.4.3 Reload Function

8.4.3.1 Overview of the Reload Function

The reload function is the function which updates a portion of transfer information, more specifically, the source address, destination address, transfer count, and address reload count, to the predefined values at the time of specific transfer completion.

The reload function has two types of functions: reload function 1 and reload function 2.

8.4.3.2 Operation of Reload Function 1

When reload function 1 is enabled, the actions described in **Table 8.38** are executed at the timing of the last transfer according to the reload function 1 settings.

Table 8.38 Operation of Reload Function 1

Reload Function 1 Setting	Register	Action at the Last Transfer
00 (Reload function 1 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Transfer count	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 1 enabled. Reloading source address and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.
10 (Reload function 1 enabled. Reloading destination address and transfer count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.
11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	If reload function 2 is disabled: Not reloaded. If reload function 2 is enabled: The reload address reload count is copied to this.

Figure 8.6 shows the operation of reload function 1.

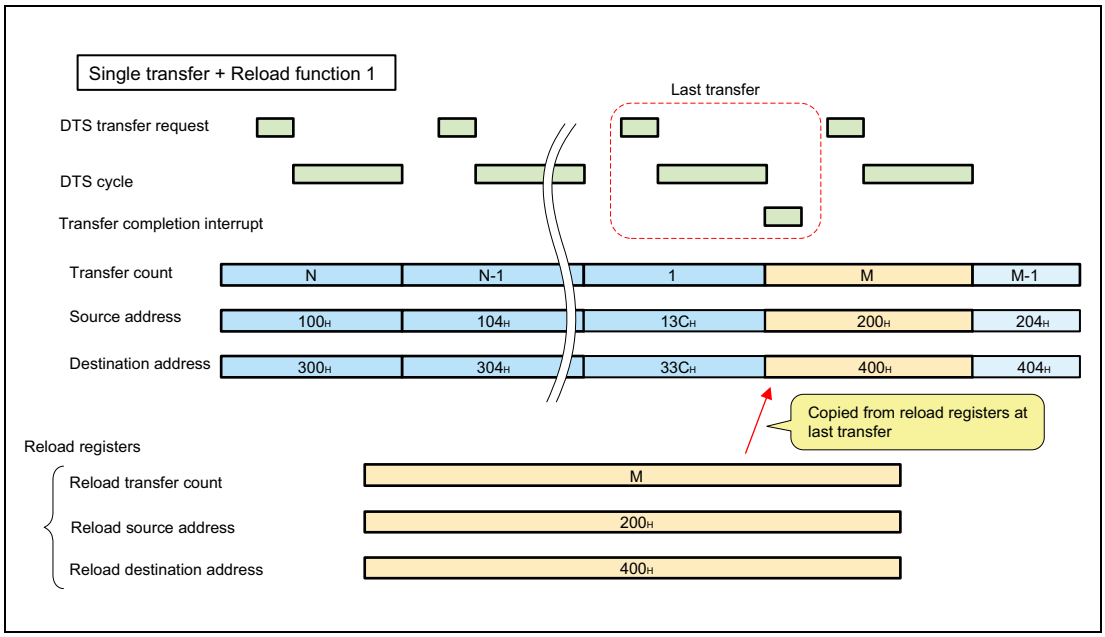


Figure 8.6 Operation of Reload Function 1

8.4.3.3 Reload Function 2

When reload function 2 is enabled, the actions described in **Table 8.39** are executed at the timing of address reload transfer according to the reload function 2 settings.

Table 8.39 Operation of Reload Function 2

Reload Function 2 Setting	Register	Action at Address Reload Transfer
00 (Reload function 2 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 2 enabled. Reloading source address.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Address reload count	The reload address reload count is copied to this.
10 (Reload function 2 enabled. Reloading destination address.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.
11 (Reload function 2 enabled. Reloading source address and destination address.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.

Figure 8.7 shows the operation of reload function 2.

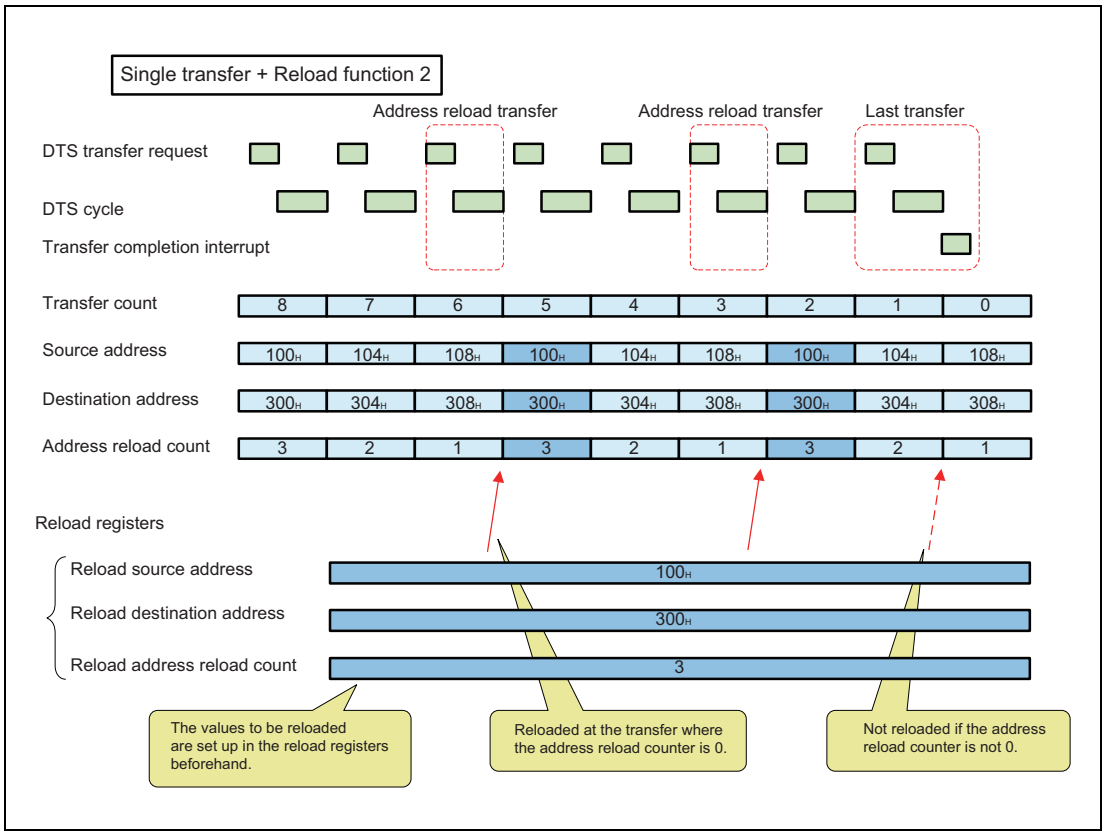


Figure 8.7 Operation of Reload Function 2

Figure 8.8 shows the operation when both reload function 1 and reload function 2 are used simultaneously.

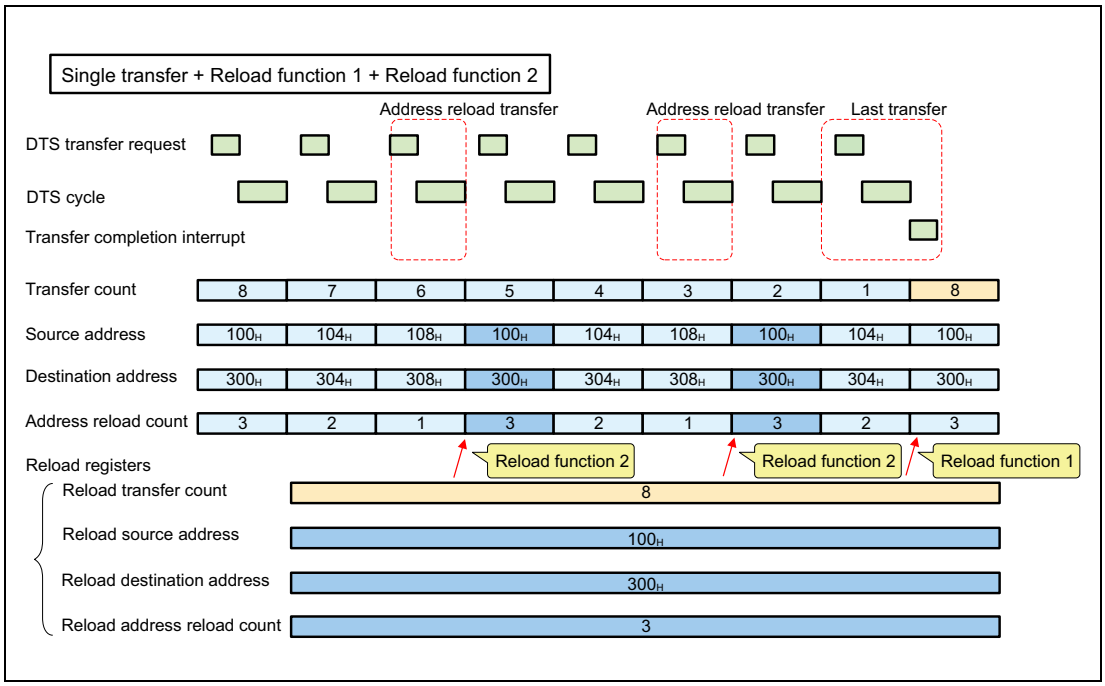


Figure 8.8 Operation When Combining Reload Function 1 and Reload Function 2

8.4.3.4 Timing of Setting DTS Reload Registers

It should be noted that the right timing of setting up the reload source address information, reload destination address information and reload transfer count information differs depending on the transfer mode.

In the single transfer mode, the TI fetched at the beginning of the last transfer or address reload transfer is used for reload at the completion of the DTS cycle. Therefore, if the reload function is used for a single transfer, the reload source address information, reload destination address information, and reload transfer count information in the TI must be set up before the beginning of the last transfer or address reload transfer.

For a block transfer, TI is fetched only at the beginning of the block transfer. The TI is used for the reload function at the last transfer or the address reload transfer. Therefore, when using the reload function for a block transfer, the reload contents in the TI must be set up before the beginning of the block transfer. If the reload contents is updated during the block transfer, those new settings cannot be reflected to the TI nor the next transfer.

8.4.4 Chain Function

8.4.4.1 Overview

DTS provides a function called chain function. When using the chain function, the completion of the DTS cycle or the last transfer for one channel can trigger a DTS transfer request for another channel. The DTS transfer request initiated by the chain function is called a chain request.

You can select the condition for generating a chain request from the following two options.

- Always chain: A chain request is generated at the completion of every DTS cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 8.9 shows the operation of the “always chain” option.

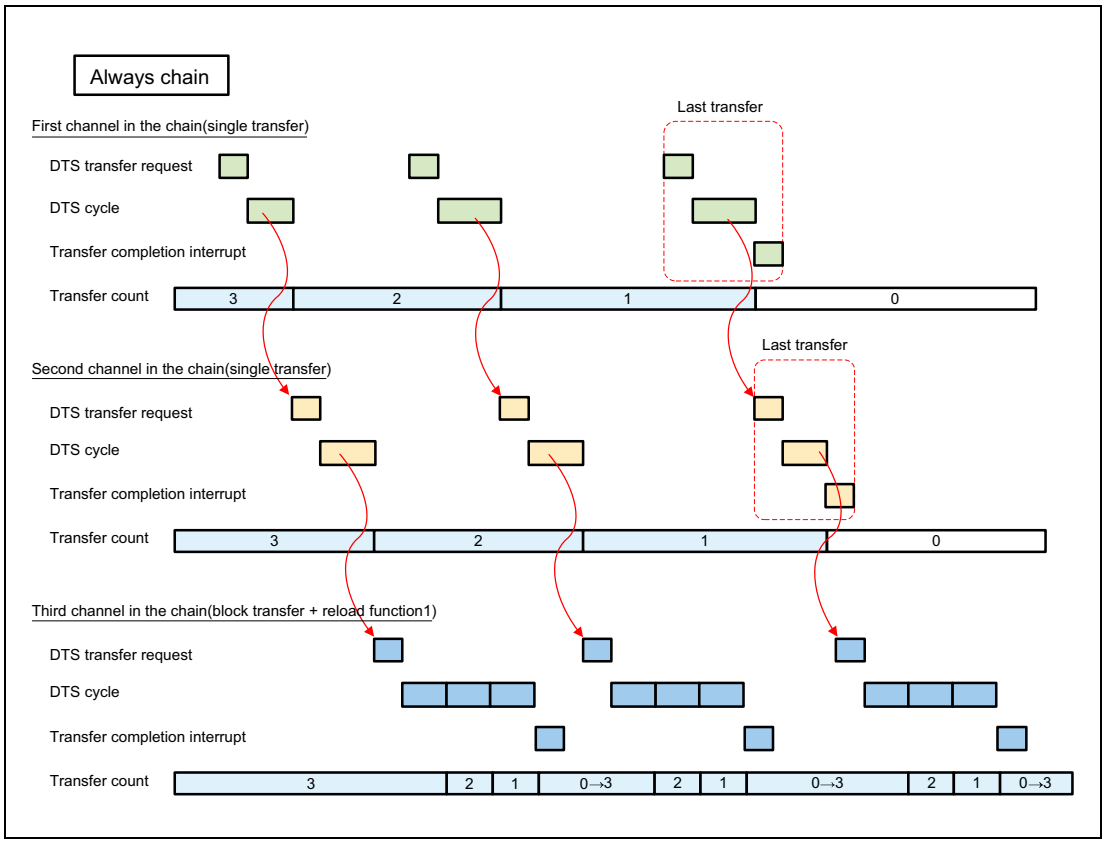


Figure 8.9 Operation of “Always Chain” Option

Figure 8.10 shows the operation of the “chain at the last transfer” option.

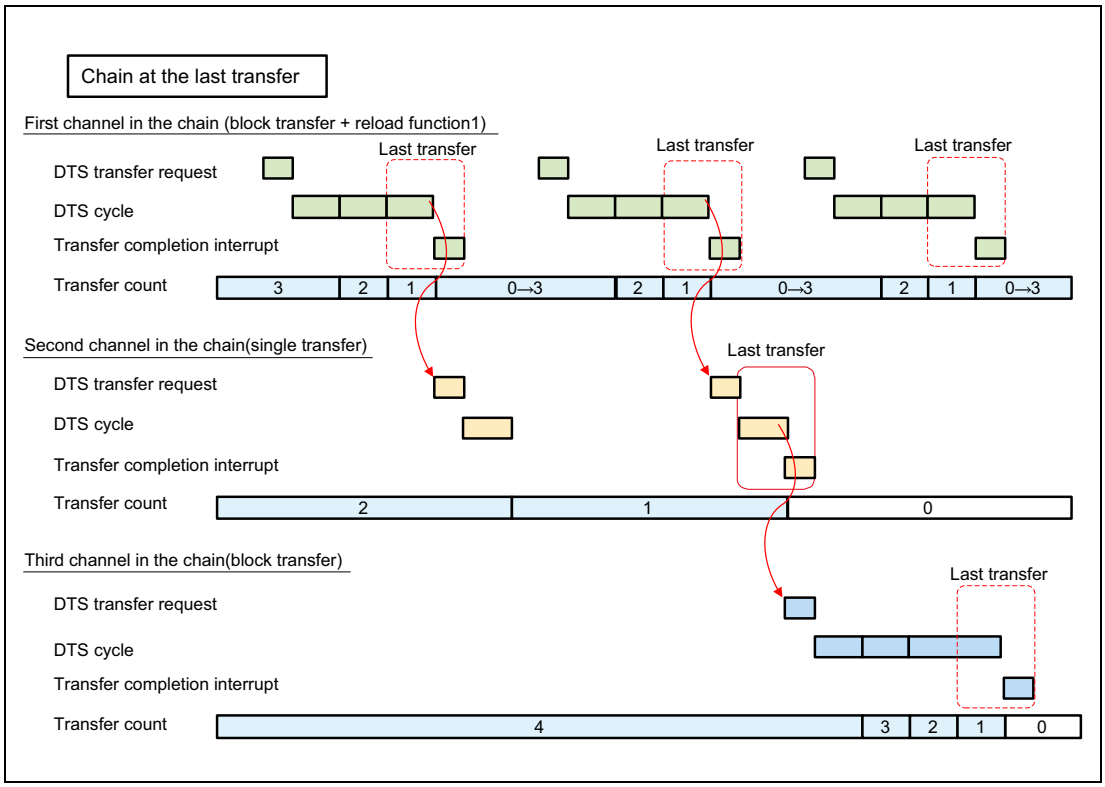


Figure 8.10 Operation of “Chain at the Last Transfer” Option

8.4.4.2 Setting Up the Chain Function

When using the chain function, it has to be enabled by the chain enable (DTTCTnnn.CHNE) and the next channel in the chain has to be specified in the chain selection (DTTCTnnn.CHNSEL) of the DTS Transfer Control Register.

If the reliability functions are required, the chain restriction (DTSnnnCM.CHAIN_RESTRICT, CHAIN_SPID[4:0], CHAIN_UM) also has to be specified in DTS Channel Master Setting Register.

For detailed information about reliability function, see **Section 8.7, Reliability Function** for details.

8.4.5 DTS Operation

8.4.5.1 Types of DTS Transfer Requests and Assigning DTS Transfer Requests

DTS starts DTS transfer by accepting a hardware DTS transfer request or software DTS transfer request.

A transfer request for DTS is retained in the transfer request pending bit of DTSFSL for each channel.

For DTSFSL, both a hardware DTS transfer request and a software DTS transfer request are retained in the same transfer request pending bit. When executing DTS transfer, DTS does not care whether the DTS transfer request is a hardware DTS transfer request or software DTS transfer request.

8.4.5.2 Generating and Accepting a DTS Transfer Request

DTS only supports edge detection for hardware DTS transfer requests.

When DTSFSL detects a hardware DTS transfer source*1 input, DTSFSL sets the transfer request pending bit and retains the hardware DTS transfer source as a DTS transfer request. If the transfer request pending bit is set and the transfer request enable bit (DTFSLnnn.REQEN) in the DTSFSL operation setting register is set, DTSFSL notifies DTS of the DTS transfer request.

Software can also generate a DTS transfer request by setting the transfer request pending bit (DTFSTnnn.DRQ) using the DTSFSL transfer request set register (DTFSSnnn).

DTSFSL can retain only one DTS transfer request per channel. If, while the transfer request pending bit for a channel is set, a new hardware DTS transfer source input for the same channel occurs, DTSFSL ignores the new hardware DTS transfer source input.

The transfer request pending bit is automatically cleared when DTS accepts the DTS transfer request. DTSFSL clears the transfer request pending bit automatically when DTS accepts the DTS transfer request regardless of the type of DTS transfer to be executed by DTS.

The transfer request pending bit can also be cleared using the DTSFSL transfer status clear register (DTFSCnnn). If the transfer request pending bit of a channel is cleared before DTS accepts the DTS transfer request, DTS transfer of the channel is not executed.

Note 1. For details, refer to Appendix file “DTS_Transfer_request_Table.xlsx”.

8.4.5.3 Executing DTS Transfer

When DTS accepts a DTS transfer request for a channel, DTS executes DTS transfer on the channel. If there are DTS transfer requests from multiple channels, DTSFSL arbitrates the DTS channels and selects the channel for the DTS transfer request.

While DTS is executing DTS transfer, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is set. In addition, the channel number of the currently ongoing DTS transfer is set in the DTS transfer channel (DTSSTS.DTSACH).

When the DTS transfer is complete or aborted because of a DTS transfer error or a register write and no channel is currently executing DTS transfer, the DTS transfer status (DTSSTS.DTSACT) bit is cleared.

8.4.5.4 DTSRAM Access

DTS accesses the DTSRAM when DTS transfer starts and finishes.

The DTS action of reading transfer information from the DTSRAM when DTS transfer starts is called TI fetch.

The DTS action of updating the transfer information on the DTSRAM when DTS transfer finishes is called TI writeback.

A single transfer performs a TI fetch at the beginning of a DTS cycle and a TI writeback at the end of a DTS cycle.

A block transfer performs a TI fetch at the beginning of the first DTS cycle and a TI writeback at the end of the DTS cycle that satisfies the block transfer completion condition (the last transfer or address reload transfer).

Therefore, in the case of a single transfer, the transfer information in DTSRAM is updated during each DTS cycle. In the case of block transfer, the transfer information in DTSRAM is updated after completion of the block transfer. If software reads the transfer information from DTSRAM during execution of a block transfer, the information represents the beginning of the block transfer.

8.5 Suspension, Resume, and Transfer Abort, and Clearing a DTS Transfer Request

8.5.1 Suspend, Resume, and Transfer Abort of a DTS Transfer

A DTS transfer being executed by DTS can be suspended by setting the DTS suspend bit (DTSCTL1.DTSUST) in DTS control register 1. If a DTS cycle is in progress, the DTS transfer is suspended after the DTS cycle is finished. If the current DTS cycle is a single transfer or a transfer that completes a block transfer (the last transfer or address reload transfer), the DTS transfer is suspended after a TI writeback and completion of the DTS cycle.

If the type of the current DTS cycle is other than the above, DTS transfer is suspended after the completion of the DTS cycle without a TI writeback. To resume DTS transfer after suspension, the DTS suspend bit in DTS control register 1 has to be cleared.

To abort DTS transfer by DTS, first suspend DTS as described above and then set the DTS transfer abort request bit (DTSCTL2.DTSTIT) in DTS control register 2. If a transfer is aborted, no TI writeback is executed. In addition, aborting a DTS transfer does not change the value of the DTS suspend bit (DTSCTL1.DTSUST). If it is desired to have the DTS accept another DTS transfer request after the abort, clear the DTS suspend bit.

Figure 8.11 shows an example of suspension, resume, and transfer abort of a DTS transfer.

In **Figure 8.11**, channels 0, 1, and 2 are executing block transfer. At time tick 1, a DTS transfer request for channel 1 is accepted and DTS transfer starts. At time tick 2, DTS transfer requests for channels 0 and 2 are generated. At time tick 3, the last transfer of channel 1 is complete, and as a result of DTS channel arbitration, a DTS transfer request for channel 0 is accepted, and DTS transfer of channel 0 starts because channel 0 has a higher priority than channel 2. At time tick 4, the last transfer of channel 0 is complete, and DTS transfer of channel 2 starts. At time tick 5, DTS is put into the suspended state, and the DTS transfer of channel 2 is suspended. At time tick 6, DTS transfer requests for channels 0 and 1 are generated. At time tick 7, the suspended state for DTS is cleared, and the DTS transfer of channel 2, which was suspended in the middle of a block transfer, is resumed. If DTS transfer is suspended in the middle of a block transfer, no DTS channel arbitration is executed when it is resumed. At time tick 8, the last transfer of channel 2 is complete, and as a result of DTS channel arbitration, a DTS transfer request for channel 0 is accepted and the DTS transfer starts because channel 0 has a higher priority than channel 2. At time tick 9, DTS is put into the suspended state, and at time tick 10, the suspended DTS transfer of channel 0 is aborted. When the suspended state of DTS is cleared at time tick 11, DTS transfer of channel 1 starts because there is no currently ongoing DTS transfer and channel 1 is the only channel with a DTS transfer request.

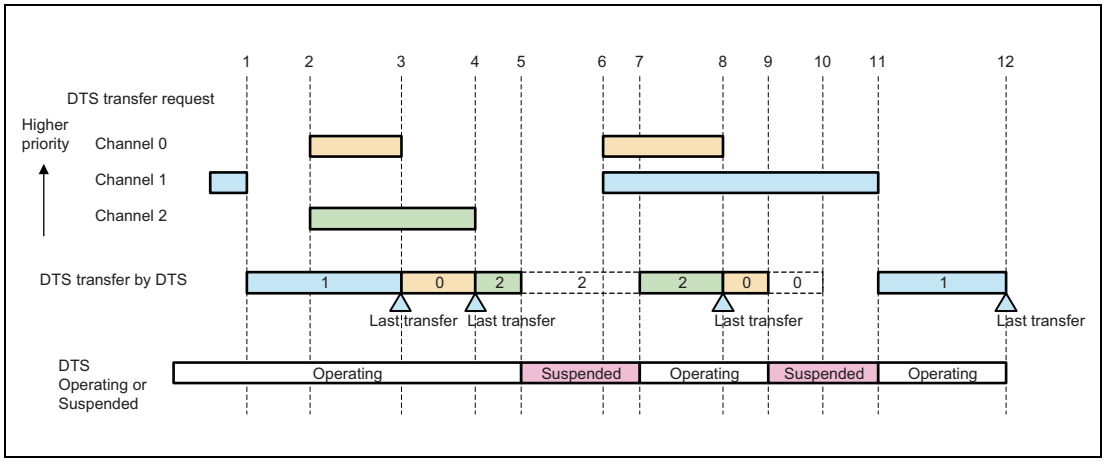


Figure 8.11 Example of Suspension, Resume, and Transfer Abort of a DTS Transfer

8.5.2 Masking and Clearing a Hardware DTS Transfer Request by DTSFSL

A DTS transfer request from a channel to DTS can be temporarily disabled (masked) by clearing the transfer request enable bit (DTFSLnnn.REQEN) in the DTSFSL operation setting register. In this case, the masked channel is excluded from DTSFSL arbitration.

A DTS transfer request held in DTSFSL can be cleared by setting the transfer request clear bit (DTFSCnnn.DRQC) in the DTSFSL transfer request clear register.

DTSFSL monitors hardware transfer source requests regardless of the state of DTS and the transfer request enable bit (DTFSLnnn.REQEN). DTSFSL sets a DTS transfer request for a channel when it detects a hardware transfer source request. When resuming or starting DTS transfer, a hardware DTS transfer request held in DTSFSL should be cleared as required.

8.5.3 List of Suspend, Resume, and Transfer Abort Functions

Table 8.40 List of Suspend, Resume, and Transfer Abort Functions

Function	How to Execute the Function	Operation	Possibility of DTS Transfer Abort
Suspension and resume of DTS	Setting and clearing DTSCTL1.DTSUST.	All channels of DTS transfer is suspended.	Possible (by setting DTSCTL2.DTSTIT during suspension)

8.6 Error Control

8.6.1 Types of Errors

DTS can generate the following types of errors:

- DTS Transfer Error
  This error is generated when an error is detected in the read cycle or write cycle in a DTS cycle or when an error is detected by checking the ECC in the read cycle.
  This error can be generated in all DTS channels during execution of DTS transfer.
- DTSRAM ECC Error
  This error is generated when an ECC error is detected in the DTSRAM read access by DTS.
  This error can be generated in the TI fetch during execution of DTS transfer for DTS or while software is accessing the DTS channel registers.
- DTSRAM Address Feedback Error
  This error is generated when an Address Feedback error is detected in the DTSRAM read or write access by DTS.
  This error can be generated in the TI fetch or TI writeback during execution of DTS transfer for DTS or while software is accessing the DTS channel registers. See **Section 44, Functional Safety** for details.

8.6.2 DTS Transfer Error

8.6.2.1 Operation of DTS When a DTS Transfer Error Occurs

When a DTS transfer error occurs in DTS, the transfer error flag (DTFSTnnn.ER) in the DTS transfer status register of the channel on which the DTS transfer error occurred is set. The transfer error cycle flag (DTSFSTnnn.ERWR) in the DTS transfer status register of the channel on which the DTS transfer error occurred shows whether the error occurred in the read cycle or the write cycle. The DTS Error register (DTSER) shows the smallest number of DTS channels on which the DTS transfer error occurred.

If a DTS transfer error occurs in a single transfer, a TI writeback is executed to finish the DTS cycle.

If a DTS transfer error occurs in the middle of a block transfer and DTS error transfer disable bit (DTTCTnnn.ESE) is set, the remaining DTS cycles in the block transfer are not executed, but a TI writeback is executed to finish the DTS cycle. At the same time, the DTS transfer status (DTSSTS.DTSACT) bit in the DTS status register is cleared. If a DTS transfer error occurs in the middle of a block transfer and DTS error transfer disable bit (DTTCTnnn.ESE) is cleared, the block transfer continues regardless of the DTS transfer error.

If a DTS transfer error occurs during the read cycle of a DTS cycle, the write cycle is not executed. If a DTS transfer error occurs during the write cycle of a DTS cycle, the validity of the write is not guaranteed.

Regardless of whether a DTS transfer error occurs in the read cycle or write cycle of a DTS cycle, the source address, destination address, transfer count, and address reload count registers are updated, and the TI is updated by a TI writeback.

Even if the transfer error flag (DTFSTnnn.ER) in the DTS transfer status register of the channel is set, a TI fetch is executed when DTS accepts a DTS transfer request for the channel. If, as a result of the TI fetch, the DTS error transfer disable bit (DTTCTnnn.ESE) is found to be set, a DTS cycle and a TI writeback are not executed. If the DTS error transfer disable bit (DTTCTnnn.ESE) is cleared, DTS transfer is executed.

8.6.3 DTSRAM Error

There are three types of DTSRAM errors detected in DTSRAM read access: an ECC correctable error, an ECC uncorrectable error and an address feedback error. There is one type of DTSRAM error detected in DTSRAM write access: an address feedback error.

If an ECC correctable 1-bit error is detected during a TI fetch, error corrected data is used, and DTS transfer continues. If an ECC correctable 1-bit error is detected during DTS channel register access from software, error corrected data is returned as read data.

If an ECC uncorrectable error or an address feedback error is detected during a TI fetch, handling of the DTS transfer request is terminated without executing a DTS cycle and TI writeback. If an address feedback error is detected during a TI writeback, the validity of the TI writeback is not guaranteed. If an ECC uncorrectable error or address feedback error is detected during a DTS channel register read from software, a peripheral bus error is reported. If an address feedback error is detected during a DTS channel register write from software, a peripheral bus error is not reported, but the validity of the write access is not guaranteed. In addition to an ECC 2-bit error, an ECC 1- bit error is treated as uncorrectable if 1-bit error correction is disabled in ECC.

8.7 Reliability Function

8.7.1 Overview

In this product, DTS provides the following reliability functions to support multi-core configurations (multiple master PEs).

- Register access protection function

  See **Section 44, Functional Safety**.

- Master information inheritance function
- SPID setting restriction

  See **Section 44, Functional Safety**.

- Restriction on chain function

8.7.2 Master Information Inheritance Function

The master information inheritance function uses the contents of the channel master setting registers as the sideband information of the data transfer interface.

DTS outputs the sideband information of data transfer for the access guard.

Table 8.41 shows the sideband information that DTS outputs.

Table 8.41 Sideband Information That Is Output from DTS

Meaning	Value Output from DTS
UM	Same as the UM bit value in the channel master setting register
SEC	Non secure
SPID	Same as the SPID bit value in the channel master setting register,

8.7.3 Restriction on Chain Function

The channel configuration of the chain function can be restricted. When chain function restriction is enabled, only a pre-allowed channel in master information can be set as the next channel.

The chain function is not restricted if the CHAIN_RESTRICT bit of the DTS Channel Master Setting Register (DTSnnnCM) is cleared. In this case, DTS channel can specify any other channel to be included in the chain function.

The chain function is restricted if the CHAIN_RESTRICT bit of the DTS Channel Master Setting Register (DTSnnnCM) is set. In this case, only the channels in which CHAIN_SPID4-0 and CHAIN_UM match can be included into the chain function by DTS channel.

When a chain is executed, DTS compares the channel master settings of the previous and next channels to decide if the chain is permissible.

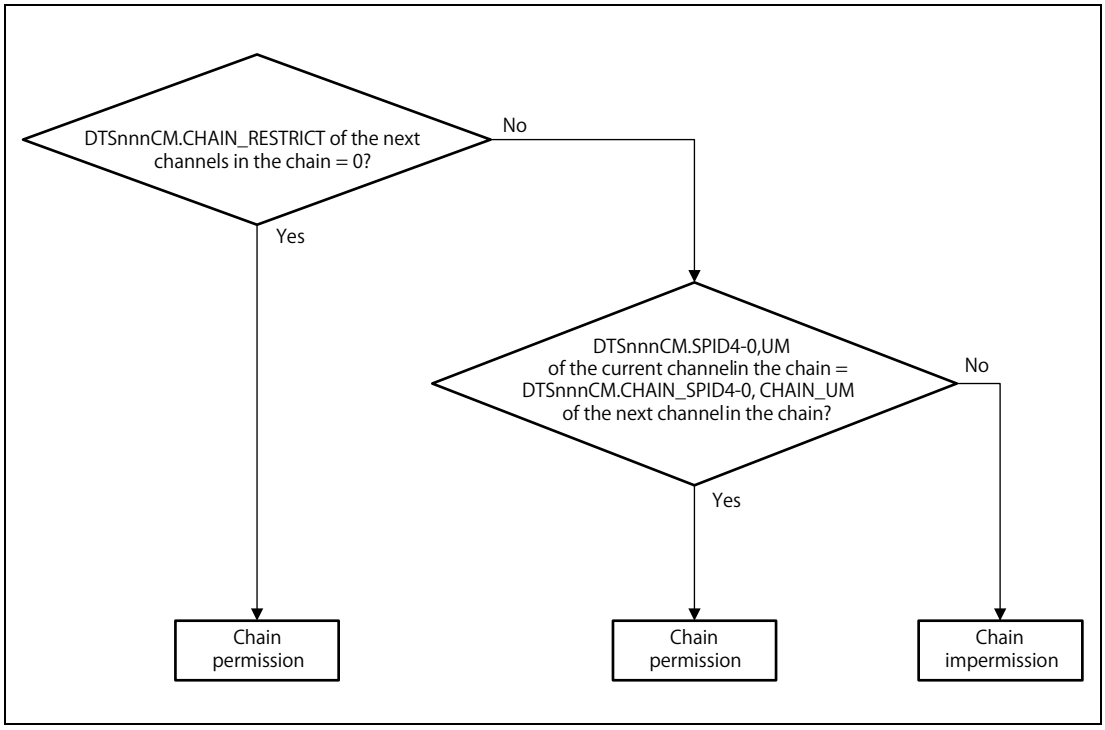


Figure 8.12 Chain Judgement Flow

8.8 Setting up DTS Transfer

8.8.1 Overview of Setting up DTS Transfer

Table 8.42 Overview of Setting up DTS Transfer (1/2)

Description	Register		Necessity of the Setting
Transfer request group selection setting	DTSSELm	DTS transfer request group selection	Mandatory (if using a hardware transfer request)
Overall DTS operation setting	DTSPRn	DTS channel priority setting	Mandatory (if DTS is used)
	DTSnnnCM	DTS channel master setting	Mandatory (if DTS is used)

Table 8.42 Overview of Setting up DTS Transfer (2/2)

Description	Register		Necessity of the Setting
Channel setting	DTSAnnn	DTS source address	Mandatory
	DTDAnnn	DTS destination address	Mandatory
	DTTCnnn	DTS transfer count	Mandatory
	DTTCTnnn	DTS transfer control	Mandatory
	DTRSAnnn	DTS reload source address	Mandatory if the reload function is used
	DTRDAnnn	DTS reload destination address	
	DTRTCnnn	DTS reload transfer count	
	DTTCCnnn	DTS transfer count compare	Mandatory if the transfer count match interrupt is used
Status clear	DTFSCnnn	DTSFSL transfer status clear	Recommended
Transfer request enable	DTFSLnnn	DTSFSL operation setting	Mandatory

8.8.2 Setting up the Transfer Request Group Selection

Set up the transfer request group selection before using DTS. The following registers must be set up to configure the transfer request group.

- DTS Transfer Request group selection Register (DTSSELm) (m = 0 to 15)
  This register configures the transfer request group selection.

8.8.3 Setting up the Overall DTS Operation

Set up the overall DTS operation before using DTS.

The following registers must be set up to configure the overall DTS operation.

- DTS channel priority setting registers (DTSPRn, n = 0 to 7)
  These registers configure the priority level of each DTS channel used for DTS channel arbitration.
- DTS channel master setting registers (DTSnnnCM)
  These registers configure channel assignment. (For details, see **Section 8.7, Reliability Function**.)

NOTE

It is important to properly configure the DTS channel master setting registers. Otherwise DTS transfer will not operate as expected.

It is recommended to clear following errors detected while setting up DTS:

- The DTSFSL Transfer Status Register (DTFSTnnn)

NOTE

Channel number information of the channel with error can be obtained by reading the DTS Error Register (DTSER).

8.8.4 Configuring the DTS Channel Settings

The DTS channel settings define the transfer information and transfer source for a DTS channel.

8.8.4.1 Configuring the DTS Channel Settings

The following procedure shows how to configure the DTS channel settings.

(1) Disable transfer requests by DTSFSL.

Clear the transfer request enable bit (DTFSLnnn.REQEN) in the DTSFSL operation setting register of the DTS channel being configured. This procedure is not mandatory but recommended in order to prevent DTS transfer requests during DTS channel configuration. It is also recommended to check the DTS status register (DTSSTS) and confirm that DTS transfer is not ongoing on the DTS channel being configured.

(2) Configure the transfer information (TI).

The following registers need to be set up to configure DTS channel transfer information.

- DTS Source Address Register (DTSAnnn)
- DTS Destination Address Register (DTDAnnn)
- DTS Transfer Count Register (DTTCnnn)
- DTS Transfer Control Register (DTTCTnnn)
- DTS Reload Source Address Register (DTRSAnnn)
- DTS Reload Destination Address Register (DTRDAnnn)
- DTS Reload Transfer Count Register (DTRTCnnn)
- DTS Transfer Count Compare Register (DTTCCnnn)

(3) Set the DTS transfer request.

DTS does not distinguish between hardware and software DTS transfer requests (no specific setting). Both hardware and software DTS transfer requests are retained in the same transfer request pending bit (DTFSTnnn.DRQ).

DTSFSL may retain a DTS transfer request that came before the transfer information is set up. If necessary, DTS transfer requests (DTFSTnnn.DRQ) retained in DTSFSL can be cleared by using the DTFSL transfer status clear register (DTFSCnnn).

(4) Enable transfer requests by DTSFSL.

Set the transfer request enable bit (DTFSLnnn.REQEN) in the DTSFSL operation setting register to enable DTS transfer requests for the DTS channel.

After the transfer request enable bit for DTSFSL is set, DTS can accept a DTS transfer request and start DTS transfer.

Section 9 Reset Controller

9.1 Features

The Reset Controller controls all factors that have an influence on the reset behavior of the device.

The device has several kinds of reset categories depending on the areas that are reset. Each reset category is triggered by one or multiple reset sources.

The relation between reset categories and their sources are shown in **Table 9.1**. The relation between reset categories and initialized area is shown in **Table 9.2**.

Table 9.1 Reset Categories and Reset Sources

No.	Reset Category	Source
1	Power On Reset*1	• POC Reset (POCRES)
		• Debugger Reset (by debugger)*2
2	System Reset 1	• External Reset
		• Primary VMON Reset (VCC/AWOVDD, ISOVDD, E0VCC)
		• Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]
3	System Reset 2	• Software System Reset (by SWSRESA, SWSRESA_ICUM)
		• ECM Reset (if RESC0 = 0)
		• WDTBA Reset
4	Application Reset	• Software Application Reset (by SWARESA, SWARESA_ICUM)
		• ECM Reset (if RESC0 = 1)
5	DeepSTOP Reset	• DeepSTOP Reset (by DeepSTOP mode trigger)
6	Module Reset	• Software Module Reset (by SWMRESA _<name>*3)
7	JTAG Reset	• JTAG Reset ($\overline{\text{TRST}}$)*2

Note 1. Power On Reset is asserted when SYSVCC supply voltage is under VPOC level and is released when SYSVCC supply is over VPOC level. About VPOC, refer to **Section 10, Power Supply Circuit**, **Section 55, Electrical Characteristics**.

Note 2. For details, refer to **Section 50, Debugging and Calibration**.

Note 3. In this section, <name> represents target module.

Table 9.2 Reset Categories and Initialized Area (1/3)
(√: Reset (Initialized), —: Without Reset)

Initialized Area (module)		Reset category						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PE0	PE0	√	√	√	√	√	—	—
	PE0 Local RAM	—	—	—	—	—	—	—
	Window Watchdog Timer 0	√	√	√	√	√	—	—
PE1	PE1	√	√	√	√	√	—	—
	PE1 Local RAM	—	—	—	—	—	—	—
	Window Watchdog Timer 1	√	√	√	√	√	—	—
PE2	PE2	√	√	√	√	√	—	—
	PE2 Local RAM	—	—	—	—	—	—	—
	Window Watchdog Timer 2	√	√	√	√	√	—	—

Table 9.2 Reset Categories and Initialized Area (2/3)
(√: Reset (Initialized), —: Without Reset)

Initialized Area (module)		Reset category						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PE3	PE3	√	√	√	√	√	—	—
	PE3 Local RAM	—	—	—	—	—	—	—
	Window Watchdog Timer 3	√	√	√	√	√	—	—
RAM	DTSRAM	√*9	√*9	√*9	√*2	√*9	—	—
	DPRAM	√*9	√*9	√*9	√*2	√*9	—	—
	Cluster RAM	—	—	—	—	—	—	—
	Retention RAM	—	—	—	—	—	—	—
Debug	Debug Function	√*3	—	—	—	—	—	√*1
	Debug RAM	—	—	—	—	—	—	—
	Emulation RAM	—	—	—	—	—	—	—
System Control	Operating Mode Controller	—	√*6	—	—	—	—	—
	Clock Controller (Clock for AWO area)	√	√	√*7	√*7	—	—	—
	Clock Controller (PLL, Clock for ISO area)	√	√	√	√*7	√	—	—
	Power Supply Voltage Monitor (VMON)	√	√*5	√*5	—	√*5	—	—
	Power Supply Voltage Monitor (DMON)	√*5	√*5	√*5	√*5	√*5	—	—
	Power Supply Voltage Monitor (VLVI)	√*5	√*5	√*5	—	√*5	—	—
	Standby Controller (Module Standby register)	√	√	√	√	√*8	√*8	—
	Clock Monitor (0, 1, 2)	√	√	√	√	—	—	—
	Clock Monitor (3, 4, 5, 6, 7, 8, 9)	√	√	√	√	√	—	—
IO	IO buffer (AWO area) Port Register (AWO area)	√	√	√	√	—	—	—
	IO buffer (ISO area)	√	√	√	√	—	—	—
	Port Register (ISO area)	√	√	√	√	√	—	—
	$\overline{\text{ERROROUT\_M}}$	√*4	√*4	√*4	—	√*4	—	—
	$\overline{\text{RESETOUT}}$	√	√	√	√	—	—	—
	$\overline{\text{VMONOUT}}$	√*3	—	—	—	—	—	—

Table 9.2 Reset Categories and Initialized Area (3/3)
(√: Reset (Initialized), —: Without Reset)

Initialized Area (module)		Reset category						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
Peripheral	BIST Skip Control Register	√	√	—	—	—	—	—
	BIST	√	√	√	—	√	—	—
	ECM Master/Checker Error Source Status Register	√	—	—	—	√	—	—
	ECM Error Output Clear Invalidation Configuration Register	√	√	√	—	√	—	—
	ECM	√	√	√	√	√	—	—
	GTM, MSPI, MMCA	√*9	√*9	√*9	√*2	√*9	√*2	—
	RTCA	√	√	√	—	—	—	—
	ADCJ2, TAUJ2, TAUJ3	√	√	√	√	—	—	—
	RSENT, RLIN3, ADCJ0, ADCJ1, CXPI, ENCA, PSI5, PSI5-S, PWM-Diag, RHSIF, RIIC, SCI3, SFMA, TAPA, TAUD, TAUJ0, TAUJ1, TPBA, TSG3, OSTM, RS-CANFD, FLXA, ETNB*10	√	√	√	√	√	√	—
	WDTBA, LPS	√	√	√	√	—	—	—
	SWDTA, Other Peripherals	√	√	√	√	√	—	—

Note 1. It is reset by $\overline{\text{TSRT}}$. For details, refer to the **Section 50, Debugging and Calibration**.
Note 2. The execution of RAM initialization is configurable by a register setting.
Note 3. Debugger Reset is excluded.
Note 4. For details, refer to the **Section 45, Error Control Module (ECM)**.
Note 5. For details, refer to the **Section 11, Power Supply Voltage Monitor**.
Note 6. External Reset only.
Note 7. For details, refer to the **Section 13, Clock Controller**.
Note 8. For details, refer to the **Section 15, Standby Controller (STBC)**.
Note 9. The execution of RAM initialization depends on Flash Option Byte.
Note 10. RAM initialization of RS-CANFD and FLXA are executed when each module startup. RAM initialization of ETNB is not executed by each reset factor.

9.2 Input/Output Pins

I/O pins related to reset are shown in **Table 9.3**.

Table 9.3 I/O Pins

Pin function name	Direction	Description
$\overline{\text{RESET}}$	Input	Reset Input
$\overline{\text{TRST}}$	Input	Debug Reset Input
$\overline{\text{RESETOUT}}$	Output	Reset Output

9.3 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA/DTS requests.

This module has no error notifications to ECM.

9.4 Registers

9.4.1 List of Registers

The register list related to reset is shown in **Table 9.4, List of Registers for Isolated area (ISO area)**, **Table 9.5, List of Registers for Always-On area (AWO area)**.

Table 9.4 List of Registers for Isolated area (ISO area) (1/3)

Register Name	Symbol	Address	Access Width	Value after reset	Access Protection	
					PBG	Other
RAM Initialization Mode Control Register for DPRAM	STAC_DPRAM	FF98 0800$_H$	32	0000 000x$_H$	PBG20 #2	—
RAM Initialization Mode Control Register for DTSRAM	STAC_DTSRAM	FF98 0810$_H$	32	0000 000x$_H$	PBG20 #2	—
RAM Initialization Mode Control Register for GTM	STAC_GTM	FF98 0830$_H$	32	0000 000x$_H$	PBG20 #2	—
RAM Initialization Mode Control Register for MSPI	STAC_MSPI	FF98 0850$_H$	32	000x xxxx$_H$	PBG20 #2	—
RAM Initialization Mode Control Register for MMCA	STAC_MMCA	FF98 0860$_H$	32	0000 000x$_H$	PBG20 #2	—
Software Module Reset Assertion Register for RS-CANFD	SWMRESA_RSCFD	FF98 0900$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for RS-CANFD	SWMRESS_RSCFD	FF98 0904$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for FLXA	SWMRESA_FLXA	FF98 0910$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for FLXA	SWMRESS_FLXA	FF98 0914$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for GTM	SWMRESA_GTM	FF98 0920$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for GTM	SWMRESS_GTM	FF98 0924$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for ETNB	SWMRESA_ETNB	FF98 0930$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for ETNB	SWMRESS_ETNB	FF98 0934$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for RSENT	SWMRESA_RSENT	FF98 0940$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for RSENT	SWMRESS_RSENT	FF98 0944$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for MSPI	SWMRESA_MSPI	FF98 0950$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for MSPI	SWMRESS_MSPI	FF98 0954$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for RLIN3	SWMRESA_RLIN3	FF98 0960$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for RLIN3	SWMRESS_RLIN3	FF98 0964$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for ADCJ (ISO)	SWMRESA_ADCJ_ISO	FF98 0970$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for ADCJ (ISO)	SWMRESS_ADCJ_ISO	FF98 0974$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for CXPI	SWMRESA_CXPI	FF98 0980$_H$	32	0000 0000$_H$	PBG20 #2	—

Table 9.4 List of Registers for Isolated area (ISO area) (2/3)

Register Name	Symbol	Address	Access Width	Value after reset	Access Protection	
					PBG	Other
Software Module Reset Status Register for CXPI	SWMRESS_CXPI	FF98 0984$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for MMCA	SWMRESA_MMCA	FF98 0990$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for MMCA	SWMRESS_MMCA	FF98 0994$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for ENCA	SWMRESA_ENCA	FF98 09A0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for ENCA	SWMRESS_ENCA	FF98 09A4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for PSI5	SWMRESA_PSI5	FF98 09B0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for PSI5	SWMRESS_PSI5	FF98 09B4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for PSI5-S	SWMRESA_PSI5S	FF98 09C0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for PSI5-S	SWMRESS_PSI5S	FF98 09C4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for PWM-Diag	SWMRESA_PWMD	FF98 09D0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for PWM-Diag	SWMRESS_PWMD	FF98 09D4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for RHSIF	SWMRESA_RHSIF	FF98 09E0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for RHSIF	SWMRESS_RHSIF	FF98 09E4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for RIIC	SWMRESA_RIIC	FF98 09F0$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for RIIC	SWMRESS_RIIC	FF98 09F4$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for SCI3	SWMRESA_SCI3	FF98 0A00$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for SCI3	SWMRESS_SCI3	FF98 0A04$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for SFMA	SWMRESA_SFMA	FF98 0A10$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for SFMA	SWMRESS_SFMA	FF98 0A14$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for TAPA	SWMRESA_TAPA	FF98 0A20$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for TAPA	SWMRESS_TAPA	FF98 0A24$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for TAUD	SWMRESA_TAUD	FF98 0A30$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for TAUD	SWMRESS_TAUD	FF98 0A34$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for TAUJ (ISO)	SWMRESA_TAUJ_ISO	FF98 0A40$_H$	32	0000 000x$_H$	PBG20 #2	—
Software Module Reset Status Register for TAUJ (ISO)	SWMRESS_TAUJ_ISO	FF98 0A44$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for TPBA	SWMRESA_TPBA	FF98 0A50$_H$	32	0000 0000$_H$	PBG20 #2	—

Table 9.4 List of Registers for Isolated area (ISO area) (3/3)

Register Name	Symbol	Address	Access Width	Value after reset	Access Protection	
					PBG	Other
Software Module Reset Status Register for TPBA	SWMRESS_TPBA	FF98 0A54$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for TSG3	SWMRESA_TSG3	FF98 0A60$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for TSG3	SWMRESS_TSG3	FF98 0A64$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Assertion Register for OSTM	SWMRESA_OSTM	FF98 0A80$_H$	32	0000 0000$_H$	PBG20 #2	—
Software Module Reset Status Register for OSTM	SWMRESS_OSTM	FF98 0A84$_H$	32	0000 0000$_H$	PBG20 #2	—
Reset Factor Clear Register	RESFC	FF98 0C00$_H$	32	0000 0000$_H$	PBG20 #2	—
Reset Factor Clear Register for Debugger Disconnection Reset	RESFDDC [For U2A16, U2A8 and U2A6 Only]	FF98 0C10$_H$	32	0000 0000$_H$	PBG20 #2	—
Reset Controller Register Key Code Protection Register 0	RESKCPROT0	FF98 0F00$_H$	32	0000 0000$_H$	PBG20 #2	—
Boot Control Register	BOOTCTRL	FFFB 2000$_H$	32	0000 0000$_H$	PBG00 #2	—

Table 9.5 List of Registers for Always-On area (AWO area)

Register Name	Symbol	Address	Access Width	Value after reset	Access Protection	
					PBG	Other
Software System Reset Assertion Register	SWSRESA	FF98 840C$_H$	32	0000 0000$_H$	PBG20 #2	RESKC PROT0
Software Application Reset Assertion Register	SWARESA	FF98 8410$_H$	32	0000 0000$_H$	PBG20 #2	RESKC PROT0
Reset Configuration Register	RESC	FF98 8414$_H$	32	0000 0000$_H$	PBG20 #2	RESKC PROT0
Reset Factor Register	RESF	FF98 8500$_H$	32	0000 x005$_H$	PBG20 #2	—
Reset Factor Register for Debugger Disconnection Reset	RESFDD [For U2A16, U2A8 and U2A6 Only]	FF98 8510$_H$	32	0000 0000$_H$	PBG20 #2	—

9.4.2 Reset of Registers

Register reset condition is shown in **Table 9.6, Reset of Register for Isolated area (ISO area)**, **Table 9.7, Reset of Register for Always-On area (AWO area)**.

Table 9.6 Reset of Register for Isolated area (ISO area) (1/2)

Register Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
STAC_DPRAM	√	√	√	—	√	—	—
STAC_DTSRAM	√	√	√	—	√	—	—
STAC_GTM	√	√	√	—	√	—	—
STAC_MSPI	√	√	√	—	√	—	—
STAC_MMCA	√	√	√	—	√	—	—
SWMRESA_RSCFD	√	√	√	—	√	—	—
SWMRESS_RSCFD	√	√	√	—	√	—	—
SWMRESA_FLXA	√	√	√	—	√	—	—
SWMRESS_FLXA	√	√	√	—	√	—	—
SWMRESA_GTM	√	√	√	—	√	—	—
SWMRESS_GTM	√	√	√	—	√	—	—
SWMRESA_ETNB	√	√	√	—	√	—	—
SWMRESS_ETNB	√	√	√	—	√	—	—
SWMRESA_RSENT	√	√	√	—	√	—	—
SWMRESS_RSENT	√	√	√	—	√	—	—
SWMRESA_MSPI	√	√	√	—	√	—	—
SWMRESS_MSPI	√	√	√	—	√	—	—
SWMRESA_RLIN3	√	√	√	—	√	—	—
SWMRESS_RLIN3	√	√	√	—	√	—	—
SWMRESA_ADCJ_ISO	√	√	√	—	√	—	—
SWMRESS_ADCJ_ISO	√	√	√	—	√	—	—
SWMRESA_CXPI	√	√	√	—	√	—	—
SWMRESS_CXPI	√	√	√	—	√	—	—
SWMRESA_MMCA	√	√	√	—	√	—	—
SWMRESS_MMCA	√	√	√	—	√	—	—
SWMRESA_ENCA	√	√	√	—	√	—	—
SWMRESS_ENCA	√	√	√	—	√	—	—
SWMRESA_PSI5	√	√	√	—	√	—	—
SWMRESS_PSI5	√	√	√	—	√	—	—
SWMRESA_PSI5S	√	√	√	—	√	—	—
SWMRESS_PSI5S	√	√	√	—	√	—	—
SWMRESA_PWMD	√	√	√	—	√	—	—
SWMRESS_PWMD	√	√	√	—	√	—	—
SWMRESA_RHSIF	√	√	√	—	√	—	—
SWMRESS_RHSIF	√	√	√	—	√	—	—
SWMRESA_RIIC	√	√	√	—	√	—	—
SWMRESS_RIIC	√	√	√	—	√	—	—
SWMRESA_SCI3	√	√	√	—	√	—	—

Table 9.6 Reset of Register for Isolated area (ISO area) (2/2)

Register Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
SWMRESS_SCI3	√	√	√	—	√	—	—
SWMRESA_SFMA	√	√	√	—	√	—	—
SWMRESS_SFMA	√	√	√	—	√	—	—
SWMRESA_TAPA	√	√	√	—	√	—	—
SWMRESS_TAPA	√	√	√	—	√	—	—
SWMRESA_TAUD	√	√	√	—	√	—	—
SWMRESS_TAUD	√	√	√	—	√	—	—
SWMRESA_TAUJ_ISO	√	√	√	—	√	—	—
SWMRESS_TAUJ_ISO	√	√	√	—	√	—	—
SWMRESA_TPBA	√	√	√	—	√	—	—
SWMRESS_TPBA	√	√	√	—	√	—	—
SWMRESA_TSG3	√	√	√	—	√	—	—
SWMRESS_TSG3	√	√	√	—	√	—	—
SWMRESA_OSTM	√	√	√	—	√	—	—
SWMRESS_OSTM	√	√	√	—	√	—	—
RESFC	√	√	√	√	√	—	—
RESFDDC [For U2A16, U2A8 and U2A6 Only]	√	√	√	√	√	—	—
RESKCPROT0	√	√	√	—	√	—	—
BOOTCTRL	√	√	√	√	√	—	—

Table 9.7 Reset of Register for Always-On area (AWO area)

Register Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RESC	√	√	—	—	—	—	—
RESF	√*1	√*1	—	—	—	—	—
RESFDD [For U2A16, U2A8 and U2A6 Only]	√	—	—	—	—	—	—

Note 1. For details, See **Section 9.4.63, RESF — Reset Factor Register**.

9.4.3 STAC_DPRAM — RAM Initialization Mode Control Register for DPRAM

This register is used to control the RAM initialization execution of the DPRAM. In Application Reset, initialization of DPRAM is executed depending on this register.

For details of RAM initialization, refer to **Section 9.5.6, RAM Initialization**.

Access: This register can be read or written in 32-bit units.

Address: FF98 0800$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZEROMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 9.8 STAC_DPRAM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for DPRAM. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

9.4.4 STAC_DTSRAM — RAM Initialization Mode Control Register for DTSRAM

This register is used to control the RAM initialization execution of the DTSRAM. In Application Reset, initialization of DTSRAM is executed depending on this register.

For details of RAM initialization, refer to **Section 9.5.6, RAM Initialization**.

Access: This register can be read or written in 32-bit units.

Address: FF98 0810$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZEROMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 9.9 STAC_DTSRAM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for DTSRAM. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

9.4.5 STAC_GTM — RAM Initialization Mode Control Register for GTM

This register is used to control the RAM initialization execution of the GTM. In Application Reset and Module Reset, initialization of RAM for GTM is executed depending on this register.

For details of RAM initialization, refer to **Section 9.5.6, RAM Initialization**.

Access: This register can be read or written in 32-bit units.

Address: FF98 0830$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZEROMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 9.10 STAC_GTM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for GTM0. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

9.4.6 STAC_MSPI — RAM Initialization Mode Control Register for MSPI

This register is used to control the RAM Initialization execution of the MSPI. In Application Reset and Module Reset, initialization of RAM for MSPI is executed depending on this register.
For details of RAM initialization, refer to **Section 9.5.6, RAM Initialization**.

Access: This register can be read or written in 32-bit units.
Address: FF98 0850_H
Value after reset: 000x $xxxx_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RZEROMD9[1:0]		RZEROMD8[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RZEROMD7[1:0]		RZEROMD6[1:0]		RZEROMD5[1:0]		RZEROMD4[1:0]		RZEROMD3[1:0]		RZEROMD2[1:0]		RZEROMD1[1:0]		RZEROMD0[1:0]	
Value after reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9.11 STAC_MSPI Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19, 18	RZEROMD9[1:0]	RAM Initialization Mode for MSPI9. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
17, 16	RZEROMD8[1:0]	RAM Initialization Mode for MSPI8. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
15, 14	RZEROMD7[1:0]	RAM Initialization Mode for MSPI7. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
13, 12	RZEROMD6[1:0]	RAM Initialization Mode for MSPI6. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
11, 10	RZEROMD5[1:0]	RAM Initialization Mode for MSPI5. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
9, 8	RZEROMD4[1:0]	RAM Initialization Mode for MSPI4. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

Table 9.11 STAC_MSPI Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	RZEROMD3[1:0]	RAM Initialization Mode for MSPI3. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
5, 4	RZEROMD2[1:0]	RAM Initialization Mode for MSPI2. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
3, 2	RZEROMD1[1:0]	RAM Initialization Mode for MSPI1. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled
1, 0	RZEROMD0[1:0]	RAM Initialization Mode for MSPI0. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

9.4.7 STAC_MMCA — RAM Initialization Mode Control Register for MMCA

This register is used to control the RAM Initialization execution of the MMCA. In Application Reset and Module Reset, initialization of RAM for MMCA is executed depending on this register.

For details of RAM initialization, refer to **Section 9.5.6, RAM Initialization**.

Access: This register can be read or written in 32-bit units.

Address: FF98 0860$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RZEROMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 9.12 STAC_MMCA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RZEROMD[1:0]	RAM Initialization Mode for MMCA. Initial value depends on Flash Option Byte. $x0_B$: Disabled 01_B : Prohibited 11_B : Enabled

9.4.8 SWMRESA_RSCFD— Software Module Reset Assertion Register for RS-CANFD

This register is used to generate a Module Reset for RS-CANFD.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FF98 0900_H
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_RSCFD_1	SWMRESA_RSCFD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.13 SWMRESA_RSCFD Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_RSCFD_1	Software Module Reset Trigger for RSCFD1 (CAN8 to 15) , E7RC11, E7RC12, and E7RC13 0 : No function 1 : Generate Module Reset
0	SWMRESA_RSCFD_0	Software Module Reset Trigger for RSCFD0 (CAN0 to 7), E7RC01, E7RC02, and E7RC03 0 : No function 1 : Generate Module Reset

9.4.9 SWMRESS_RSCFD — Software Module Reset Status Register for RS-CANFD

This register is used to show the execution status of a Module Reset for RS-CANFD.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0904$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_RSCFD_1	SWMRESS_RSCFD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.14 SWMRESS_RSCFD Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_RSCFD_1	Software Module Reset Status for RSCFD1 (CAN8 to 15), E7RC11, E7RC12, and E7RC13 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_RSCFD_0	Software Module Reset Status for RSCFD0 (CAN0 to 7), E7RC01, E7RC02, and E7RC03 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.10 SWMRESA_FLXA — Software Module Reset Assertion Register for FLXA

This register is used to generate a Module Reset for FLXA.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0910_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_FLXA_1	SWMRESA_FLXA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.15 SWMRESA_FLXA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_FLXA_1	Software Module Reset Trigger for FLXA1, E7FR10, E7FR11, and E7FR12 0 : No function 1 : Generate Module Reset
0	SWMRESA_FLXA_0	Software Module Reset Trigger for FLXA0, E7FR00, E7FR01, and E7FR02 0 : No function 1 : Generate Module Reset

9.4.11 SWMRESS_FLXA — Software Module Reset Status Register for FLXA

This register is used to show the execution status of a Module Reset for FLXA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0914$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_FLXA_1	SWMRESS_FLXA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.16 SWMRESS_FLXA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_FLXA_1	Software Module Reset Status for FLXA1, E7FR10, E7FR11, and E7FR12 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_FLXA_0	Software Module Reset Status for FLXA0, E7FR00, E7FR01, and E7FR02 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.12 SWMRESA_GTM — Software Module Reset Assertion Register for GTM

This register is used to generate a Module Reset for GTM.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0920_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_GTM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.17 SWMRESA_GTM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWMRESA_GTM_0	Software Module Reset Trigger for GTM0, GTM0_1, E7GT00, E7GT01, E7GT10, E7GT11, E7GT20, E7GT21, E7GT30, and E7GT31 0 : No function 1 : Generate Module Reset

9.4.13 SWMRESS_GTM — Software Module Reset Status Register for GTM

This register is used to show the execution status of a Module Reset for GTM.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0924$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_GTM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.18 SWMRESS_GTM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWMRESS_GTM_0	Software Module Reset Status for GTM0, GTM0_1, E7GT00, E7GT01, E7GT10, E7GT11, E7GT20, E7GT21, E7GT30, and E7GT31 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.14 SWMRESA_ETNB— Software Module Reset Assertion Register for ETNB

This register is used to generate a Module Reset for ETNB.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0930_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_ETNB_1	SWMRESA_ETNB_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.19 SWMRESA_ETNB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_ETNB_1	Software Module Reset Trigger for ETNB1, E7GE00, and E7GE01 0 : No function 1 : Generate Module Reset
0	SWMRESA_ETNB_0	Software Module Reset Trigger for ETNB0, E7ME00, and E7ME01 0 : No function 1 : Generate Module Reset

9.4.15 SWMRESS_ETNB — Software Module Reset Status Register for ETNB

This register is used to show the execution status of a Module Reset for ETNB.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0934$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_ETNB_1	SWMRESS_ETNB_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.20 SWMRESS_ETNB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_ETNB_1	Software Module Reset Status for ETNB1, E7GE00, and E7GE01 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_ETNB_0	Software Module Reset Status for ETNB0, E7ME00, and E7ME01 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.16 SWMRESA_RSENT — Software Module Reset Assertion Register for RSENT

This register is used to generate a Module Reset for RSENT.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FF98 0940_H
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SWMRESA_RSENT_7	SWMRESA_RSENT_6	SWMRESA_RSENT_5	SWMRESA_RSENT_4	SWMRESA_RSENT_3	SWMRESA_RSENT_2	SWMRESA_RSENT_1	SWMRESA_RSENT_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 9.21 SWMRESA_RSENT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7	SWMRESA_RSENT_7	Software Module Reset Trigger for RSENT7 0 : No function 1 : Generate Module Reset
6	SWMRESA_RSENT_6	Software Module Reset Trigger for RSENT6 0 : No function 1 : Generate Module Reset
5	SWMRESA_RSENT_5	Software Module Reset Trigger for RSENT5 0 : No function 1 : Generate Module Reset
4	SWMRESA_RSENT_4	Software Module Reset Trigger for RSENT4 0 : No function 1 : Generate Module Reset
3	SWMRESA_RSENT_3	Software Module Reset Trigger for RSENT3 0 : No function 1 : Generate Module Reset
2	SWMRESA_RSENT_2	Software Module Reset Trigger for RSENT2 0 : No function 1 : Generate Module Reset
1	SWMRESA_RSENT_1	Software Module Reset Trigger for RSENT1 0 : No function 1 : Generate Module Reset
0	SWMRESA_RSENT_0	Software Module Reset Trigger for RSENT0 0 : No function 1 : Generate Module Reset

Note . Register RSENTTSSEL is not initialized by Module Reset.

9.4.17 SWMRESS_RSENT — Software Module Reset Status Register for RSENT

This register is used to show the execution status of a Module Reset for RSENT.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0944$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SWMRESS_RSENT_7	SWMRESS_RSENT_6	SWMRESS_RSENT_5	SWMRESS_RSENT_4	SWMRESS_RSENT_3	SWMRESS_RSENT_2	SWMRESS_RSENT_1	SWMRESS_RSENT_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.22 SWMRESS_RSENT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	SWMRESS_RSENT_7	Software Module Reset Status for RSENT7 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
6	SWMRESS_RSENT_6	Software Module Reset Status for RSENT6 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
5	SWMRESS_RSENT_5	Software Module Reset Status for RSENT5 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
4	SWMRESS_RSENT_4	Software Module Reset Status for RSENT4 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
3	SWMRESS_RSENT_3	Software Module Reset Status for RSENT3 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_RSENT_2	Software Module Reset Status for RSENT2 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_RSENT_1	Software Module Reset Status for RSENT1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_RSENT_0	Software Module Reset Status for RSENT0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.18 SWMRESA_MSPI — Software Module Reset Assertion Register for MSPI

This register is used to generate a Module Reset for MSPI.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0950_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SWMRESA_MSPI_9	SWMRESA_MSPI_8	SWMRESA_MSPI_7	SWMRESA_MSPI_6	SWMRESA_MSPI_5	SWMRESA_MSPI_4	SWMRESA_MSPI_3	SWMRESA_MSPI_2	SWMRESA_MSPI_1	SWMRESA_MSPI_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W

Table 9.23 SWMRESA_MSPI Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When writing, write the value after reset.
9	SWMRESA_MSPI_9	Software Module Reset Trigger for MSPI9, MSPI9_INTF, and E7MS09 0 : No function 1 : Generate Module Reset
8	SWMRESA_MSPI_8	Software Module Reset Trigger for MSPI8, MSPI8_INTF, and E7MS08 0 : No function 1 : Generate Module Reset
7	SWMRESA_MSPI_7	Software Module Reset Trigger for MSPI7, MSPI7_INTF, and E7MS07 0 : No function 1 : Generate Module Reset
6	SWMRESA_MSPI_6	Software Module Reset Trigger for MSPI6, MSPI6_INTF, and E7MS06 0 : No function 1 : Generate Module Reset
5	SWMRESA_MSPI_5	Software Module Reset Trigger for MSPI5, MSPI5_INTF, and E7MS05 0 : No function 1 : Generate Module Reset
4	SWMRESA_MSPI_4	Software Module Reset Trigger for MSPI4, MSPI4_INTF, and E7MS04 0 : No function 1 : Generate Module Reset
3	SWMRESA_MSPI_3	Software Module Reset Trigger for MSPI3, MSPI3_INTF, and E7MS03 0 : No function 1 : Generate Module Reset
2	SWMRESA_MSPI_2	Software Module Reset Trigger for MSPI2, MSPI2_INTF, and E7MS02 0 : No function 1 : Generate Module Reset
1	SWMRESA_MSPI_1	Software Module Reset Trigger for MSPI1, MSPI1_INTF, and E7MS01 0 : No function 1 : Generate Module Reset
0	SWMRESA_MSPI_0	Software Module Reset Trigger for MSPI0, MSPI0_INTF, and E7MS00 0 : No function 1 : Generate Module Reset

Note . Register MSPITGCTLn, MSPITGDMAALT and MSPITGDTSALT are not initialized by Module Reset.

9.4.19 SWMRESS_MSPI — Software Module Reset Status Register for MSPI

This register is used to show the execution status of a Module Reset for MSPI.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0954$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SWMRESS_MSPI_9	SWMRESS_MSPI_8	SWMRESS_MSPI_7	SWMRESS_MSPI_6	SWMRESS_MSPI_5	SWMRESS_MSPI_4	SWMRESS_MSPI_3	SWMRESS_MSPI_2	SWMRESS_MSPI_1	SWMRESS_MSPI_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.24 SWMRESS_MSPI Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9	SWMRESS_MSPI_9	Software Module Reset Status for MSPI9, MSPI9_INTF, and E7MS09 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
8	SWMRESS_MSPI_8	Software Module Reset Status for MSPI8, MSPI8_INTF, and E7MS08 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
7	SWMRESS_MSPI_7	Software Module Reset Status for MSPI7, MSPI7_INTF, and E7MS07 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
6	SWMRESS_MSPI_6	Software Module Reset Status for MSPI6, MSPI6_INTF, and E7MS06 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
5	SWMRESS_MSPI_5	Software Module Reset Status for MSPI5, MSPI5_INTF, and E7MS05 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
4	SWMRESS_MSPI_4	Software Module Reset Status for MSPI4, MSPI4_INTF, and E7MS04 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
3	SWMRESS_MSPI_3	Software Module Reset Status for MSPI3, MSPI3_INTF, and E7MS03 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_MSPI_2	Software Module Reset Status for MSPI2, MSPI2_INTF, and E7MS02 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_MSPI_1	Software Module Reset Status for MSPI1, MSPI1_INTF, and E7MS01 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_MSPI_0	Software Module Reset Status for MSPI0, MSPI0_INTF, and E7MS00 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.20 SWMRESA_RLIN3 — Software Module Reset Assertion Register for RLIN3

This register is used to generate a Module Reset for RLIN3.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0960_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SWMRESA_RLIN3_23	SWMRESA_RLIN3_22	SWMRESA_RLIN3_21	SWMRESA_RLIN3_20	SWMRESA_RLIN3_19	SWMRESA_RLIN3_18	SWMRESA_RLIN3_17	SWMRESA_RLIN3_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWMRESA_RLIN3_15	SWMRESA_RLIN3_14	SWMRESA_RLIN3_13	SWMRESA_RLIN3_12	SWMRESA_RLIN3_11	SWMRESA_RLIN3_10	SWMRESA_RLIN3_9	SWMRESA_RLIN3_8	SWMRESA_RLIN3_7	SWMRESA_RLIN3_6	SWMRESA_RLIN3_5	SWMRESA_RLIN3_4	SWMRESA_RLIN3_3	SWMRESA_RLIN3_2	SWMRESA_RLIN3_1	SWMRESA_RLIN3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 9.25 SWMRESA_RLIN3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When writing, write the value after reset.
23	SWMRESA_RLIN3_23	Software Module Reset Trigger for RLIN323 0 : No function 1 : Generate Module Reset
22	SWMRESA_RLIN3_22	Software Module Reset Trigger for RLIN322 0 : No function 1 : Generate Module Reset
21	SWMRESA_RLIN3_21	Software Module Reset Trigger for RLIN321 0 : No function 1 : Generate Module Reset
20	SWMRESA_RLIN3_20	Software Module Reset Trigger for RLIN320 0 : No function 1 : Generate Module Reset
19	SWMRESA_RLIN3_19	Software Module Reset Trigger for RLIN319 0 : No function 1 : Generate Module Reset
18	SWMRESA_RLIN3_18	Software Module Reset Trigger for RLIN318 0 : No function 1 : Generate Module Reset
17	SWMRESA_RLIN3_17	Software Module Reset Trigger for RLIN317 0 : No function 1 : Generate Module Reset
16	SWMRESA_RLIN3_16	Software Module Reset Trigger for RLIN316 0 : No function 1 : Generate Module Reset
15	SWMRESA_RLIN3_15	Software Module Reset Trigger for RLIN315 0 : No function 1 : Generate Module Reset
14	SWMRESA_RLIN3_14	Software Module Reset Trigger for RLIN314 0 : No function 1 : Generate Module Reset

Table 9.25 SWMRESA_RLIN3 Register Contents (2/2)

Bit Position	Bit Name	Function
13	SWMRESA_RLIN3_13	Software Module Reset Trigger for RLIN313 0 : No function 1 : Generate Module Reset
12	SWMRESA_RLIN3_12	Software Module Reset Trigger for RLIN312 0 : No function 1 : Generate Module Reset
11	SWMRESA_RLIN3_11	Software Module Reset Trigger for RLIN311 0 : No function 1 : Generate Module Reset
10	SWMRESA_RLIN3_10	Software Module Reset Trigger for RLIN310 0 : No function 1 : Generate Module Reset
9	SWMRESA_RLIN3_9	Software Module Reset Trigger for RLIN39 0 : No function 1 : Generate Module Reset
8	SWMRESA_RLIN3_8	Software Module Reset Trigger for RLIN38 0 : No function 1 : Generate Module Reset
7	SWMRESA_RLIN3_7	Software Module Reset Trigger for RLIN37 0 : No function 1 : Generate Module Reset
6	SWMRESA_RLIN3_6	Software Module Reset Trigger for RLIN36 0 : No function 1 : Generate Module Reset
5	SWMRESA_RLIN3_5	Software Module Reset Trigger for RLIN35 0 : No function 1 : Generate Module Reset
4	SWMRESA_RLIN3_4	Software Module Reset Trigger for RLIN34 0 : No function 1 : Generate Module Reset
3	SWMRESA_RLIN3_3	Software Module Reset Trigger for RLIN33 0 : No function 1 : Generate Module Reset
2	SWMRESA_RLIN3_2	Software Module Reset Trigger for RLIN32 0 : No function 1 : Generate Module Reset
1	SWMRESA_RLIN3_1	Software Module Reset Trigger for RLIN31 0 : No function 1 : Generate Module Reset
0	SWMRESA_RLIN3_0	Software Module Reset Trigger for RLIN30 0 : No function 1 : Generate Module Reset

9.4.21 SWMRESS_RLIN3 — Software Module Reset Status Register for RLIN3

This register is used to show the execution status of a Module Reset for RLIN3.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0964$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SWMRESS_RLIN3_23	SWMRESS_RLIN3_22	SWMRESS_RLIN3_21	SWMRESS_RLIN3_20	SWMRESS_RLIN3_19	SWMRESS_RLIN3_18	SWMRESS_RLIN3_17	SWMRESS_RLIN3_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SWMRESS_RLIN3_15	SWMRESS_RLIN3_14	SWMRESS_RLIN3_13	SWMRESS_RLIN3_12	SWMRESS_RLIN3_11	SWMRESS_RLIN3_10	SWMRESS_RLIN3_9	SWMRESS_RLIN3_8	SWMRESS_RLIN3_7	SWMRESS_RLIN3_6	SWMRESS_RLIN3_5	SWMRESS_RLIN3_4	SWMRESS_RLIN3_3	SWMRESS_RLIN3_2	SWMRESS_RLIN3_1	SWMRESS_RLIN3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.26 SWMRESS_RLIN3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	SWMRESS_RLIN3_23	Software Module Reset Status for RLIN323 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
22	SWMRESS_RLIN3_22	Software Module Reset Status for RLIN322 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
21	SWMRESS_RLIN3_21	Software Module Reset Status for RLIN321 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
20	SWMRESS_RLIN3_20	Software Module Reset Status for RLIN320 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
19	SWMRESS_RLIN3_19	Software Module Reset Status for RLIN319 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
18	SWMRESS_RLIN3_18	Software Module Reset Status for RLIN318 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
17	SWMRESS_RLIN3_17	Software Module Reset Status for RLIN317 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
16	SWMRESS_RLIN3_16	Software Module Reset Status for RLIN316 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
15	SWMRESS_RLIN3_15	Software Module Reset Status for RLIN315 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
14	SWMRESS_RLIN3_14	Software Module Reset Status for RLIN314 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

Table 9.26 SWMRESS_RLIN3 Register Contents (2/2)

Bit Position	Bit Name	Function
13	SWMRESS_RLIN3_13	Software Module Reset Status for RLIN313 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
12	SWMRESS_RLIN3_12	Software Module Reset Status for RLIN312 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
11	SWMRESS_RLIN3_11	Software Module Reset Status for RLIN311 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
10	SWMRESS_RLIN3_10	Software Module Reset Status for RLIN310 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
9	SWMRESS_RLIN3_9	Software Module Reset Status for RLIN39 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
8	SWMRESS_RLIN3_8	Software Module Reset Status for RLIN38 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
7	SWMRESS_RLIN3_7	Software Module Reset Status for RLIN37 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
6	SWMRESS_RLIN3_6	Software Module Reset Status for RLIN36 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
5	SWMRESS_RLIN3_5	Software Module Reset Status for RLIN35 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
4	SWMRESS_RLIN3_4	Software Module Reset Status for RLIN34 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
3	SWMRESS_RLIN3_3	Software Module Reset Status for RLIN33 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_RLIN3_2	Software Module Reset Status for RLIN32 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_RLIN3_1	Software Module Reset Status for RLIN31 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_RLIN3_0	Software Module Reset Status for RLIN30 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.22 SWMRESA_ADCJ_ISO — Software Module Reset Assertion Register for ADCJ (ISO)

This register is used to generate a Module Reset for ADCJ (ISO).
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FF98 0970_H
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_ADCJ_1	SWMRESA_ADCJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.27 SWMRESA_ADCJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_ADCJ_1	Software Module Reset Trigger for ADCJ1 0 : No function 1 : Generate Module Reset
0	SWMRESA_ADCJ_0	Software Module Reset Trigger for ADCJ0 and AVSEG 0 : No function 1 : Generate Module Reset

9.4.23 SWMRESS_ADCJ_ISO — Software Module Reset Status Register for ADCJ (ISO)

This register is used to show the execution status of a Module Reset for ADCJ (ISO).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0974$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_ADCJ_1	SWMRESS_ADCJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.28 SWMRESS_ADCJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_ADCJ_1	Software Module Reset Status for ADCJ1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_ADCJ_0	Software Module Reset Status for ADCJ0 and AVSEG 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.24 SWMRESA_CXPI — Software Module Reset Assertion Register for CXPI

This register is used to generate a Module Reset for CXPI.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0980_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_CXPI_3	SWMRESA_CXPI_2	SWMRESA_CXPI_1	SWMRESA_CXPI_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 9.29 SWMRESA_CXPI Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	SWMRESA_CXPI_3	Software Module Reset Trigger for CXP13 0 : No function 1 : Generate Module Reset
2	SWMRESA_CXPI_2	Software Module Reset Trigger for CXP12 0 : No function 1 : Generate Module Reset
1	SWMRESA_CXPI_1	Software Module Reset Trigger for CXP11 0 : No function 1 : Generate Module Reset
0	SWMRESA_CXPI_0	Software Module Reset Trigger for CXP10 0 : No function 1 : Generate Module Reset

9.4.25 SWMRESS_CXPI — Software Module Reset Status Register for CXPI

This register is used to show the execution status of a Module Reset for CXPI.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0984$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_CXPI_3	SWMRESS_CXPI_2	SWMRESS_CXPI_1	SWMRESS_CXPI_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.30 SWMRESS_CXPI Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SWMRESS_CXPI_3	Software Module Reset Status for CXP13 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_CXPI_2	Software Module Reset Status for CXP12 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_CXPI_1	Software Module Reset Status for CXP11 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_CXPI_0	Software Module Reset Status for CXP10 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.26 SWMRESA_MMCA — Software Module Reset Assertion Register for MMCA

This register is used to generate a Module Reset for MMCA.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 0990_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_MMCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.31 SWMRESA_MMCA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWMRESA_MMCA_0	Software Module Reset Trigger for MMCA0, E7MM00, and E7MM01 0 : No function 1 : Generate Module Reset

9.4.27 SWMRESS_MMCA — Software Module Reset Status Register for MMCA

This register is used to show the execution status of a Module Reset for MMCA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0994$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_MMCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.32 SWMRESS_MMCA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWMRESS_MMCA_0	Software Module Reset Status for MMCA0, E7MM00, and E7MM01 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.28 SWMRESA_ENCA — Software Module Reset Assertion Register for ENCA

This register is used to generate a Module Reset for ENCA.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $09A0_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_ENCA_1	SWMRESA_ENCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.33 SWMRESA_ENCA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_ENCA_1	Software Module Reset Trigger for ENCA1 0 : No function 1 : Generate Module Reset
0	SWMRESA_ENCA_0	Software Module Reset Trigger for ENCA0 0 : No function 1 : Generate Module Reset

9.4.29 SWMRESS_ENCA — Software Module Reset Status Register for ENCA

This register is used to show the execution status of a Module Reset for ENCA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09A4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_ENCA_1	SWMRESS_ENCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.34 SWMRESS_ENCA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_ENCA_1	Software Module Reset Status for ENCA1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_ENCA_0	Software Module Reset Status for ENCA0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.30 SWMRESA_PSI5 — Software Module Reset Assertion Register for PSI5

This register is used to generate a Module Reset for PSI5.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $09B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_PSI5_3	SWMRESA_PSI5_2	SWMRESA_PSI5_1	SWMRESA_PSI5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 9.35 SWMRESA_PSI5 Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	SWMRESA_PSI5_3	Software Module Reset Trigger for PSI53 0 : No function 1 : Generate Module Reset
2	SWMRESA_PSI5_2	Software Module Reset Trigger for PSI52 0 : No function 1 : Generate Module Reset
1	SWMRESA_PSI5_1	Software Module Reset Trigger for PSI51 0 : No function 1 : Generate Module Reset
0	SWMRESA_PSI5_0	Software Module Reset Trigger for PSI50 0 : No function 1 : Generate Module Reset

Note: Register PSI5TSSEL is not initialized by Module Reset.

9.4.31 SWMRESS_PSI5 — Software Module Reset Status Register for PSI5

This register is used to show the execution status of a Module Reset for PSI5.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09B4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_PSI5_3	SWMRESS_PSI5_2	SWMRESS_PSI5_1	SWMRESS_PSI5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.36 SWMRESS_PSI5 Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SWMRESS_PSI5_3	Software Module Reset Status for PSI53 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_PSI5_2	Software Module Reset Status for PSI52 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_PSI5_1	Software Module Reset Status for PSI51 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_PSI5_0	Software Module Reset Status for PSI50 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.32 SWMRESA_PSI5S — Software Module Reset Assertion Register for PSI5-S

This register is used to generate a Module Reset for PSI5-S.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $09C0_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_PSI5S_1	SWMRESA_PSI5S_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.37 SWMRESA_PSI5S Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_PSI5S_1	Software Module Reset Trigger for PSI5S1 0 : No function 1 : Generate Module Reset
0	SWMRESA_PSI5S_0	Software Module Reset Trigger for PSI5S0 0 : No function 1 : Generate Module Reset

9.4.33 SWMRESS_PSI5S — Software Module Reset Status Register for PSI5-S

This register is used to generate a Module Reset for PSI5-S.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09C4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_PSI5S_1	SWMRESS_PSI5S_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.38 SWMRESS_PSI5S Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_PSI5S_1	Software Module Reset Status for PSI5S1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_PSI5S_0	Software Module Reset Status for PSI5S0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.34 SWMRESA_PWMD — Software Module Reset Assertion Register for PWM-Diag

This register is used to generate a Module Reset for PWM-Diag.
This register is always read as 0000 0000$_H$.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 09D0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_PWMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.39 SWMRESA_PWMD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWMRESA_PWMD	Software Module Reset Trigger for PWM-Diag (PWBA0, PWGC0-95, PWSD0, SLPW, and PWGCINTF) 0 : No function 1 : Generate Module Reset

9.4.35 SWMRESS_PWMD — Software Module Reset Status Register for PWM-Diag

This register is used to show the execution status of a Module Reset for PWM-Diag.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09D4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_PWMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.40 SWMRESS_PWMD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWMRESS_PWMD	Software Module Reset Status for PWM-Diag(PWBA0, PWGC0-95, PWSD0, SLPW, and PWGCINTF) 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.36 SWMRESA_RHSIF — Software Module Reset Assertion Register for RHSIF

This register is used to generate a Module Reset for RHSIF.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $09E0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_RHSIF_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.41 SWMRESA_RHSIF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWMRESA_RHSIF_0	Software Module Reset Trigger for RHSIF0 0 : No function 1 : Generate Module Reset

9.4.37 SWMRESS_RHSIF — Software Module Reset Status Register for RHSIF

This register is used to show the execution status of a Module Reset for RHSIF.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09E4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_RHSIF_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.42 SWMRESS_RHSIF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWMRESS_RHSIF_0	Software Module Reset Status for RHSIF0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.38 SWMRESA_RIIC — Software Module Reset Assertion Register for RIIC

This register is used to generate a Module Reset for RIIC.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $09F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_RIIC_1	SWMRESA_RIIC_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.43 SWMRESA_RIIC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_RIIC_1	Software Module Reset Trigger for RIIC1 0 : No function 1 : Generate Module Reset
0	SWMRESA_RIIC_0	Software Module Reset Trigger for RIIC0 0 : No function 1 : Generate Module Reset

9.4.39 SWMRESS_RIIC — Software Module Reset Status Register for RIIC

This register is used to show the execution status of a Module Reset for RIIC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 09F4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_RIIC_1	SWMRESS_RIIC_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.44 SWMRESS_RIIC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_RIIC_1	Software Module Reset Status for RIIC1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_RIIC_0	Software Module Reset Status for RIIC0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.40 SWMRESA_SCI3 — Software Module Reset Assertion Register for SCI3

This register is used to generate a Module Reset for SCI3.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A00_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_SCI3_2	SWMRESA_SCI3_1	SWMRESA_SCI3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Table 9.45 SWMRESA_SCI3 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	SWMRESA_SCI3_2	Software Module Reset Trigger for SCI32 0 : No function 1 : Generate Module Reset
1	SWMRESA_SCI3_1	Software Module Reset Trigger for SCI31 0 : No function 1 : Generate Module Reset
0	SWMRESA_SCI3_0	Software Module Reset Trigger for SCI30 0 : No function 1 : Generate Module Reset

9.4.41 SWMRESS_SCI3 — Software Module Reset Status Register for SCI3

This register is used to show the execution status of a Module Reset for SCI3.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A04$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_SCI3_2	SWMRESS_SCI3_1	SWMRESS_SCI3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.46 SWMRESS_SCI3 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SWMRESS_SCI3_2	Software Module Reset Status for SCI32 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_SCI3_1	Software Module Reset Status for SCI31 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_SCI3_0	Software Module Reset Status for SCI30 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.42 SWMRESA_SFMA — Software Module Reset Assertion Register for SFMA

This register is used to generate a Module Reset for SFMA.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A10_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_SFMA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.47 SWMRESA_SFMA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWMRESA_SFMA_0	Software Module Reset Trigger for SFMA0 0 : No function 1 : Generate Module Reset

9.4.43 SWMRESS_SFMA — Software Module Reset Status Register for SFMA

This register is used to show the execution status of a Module Reset for SFMA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A14$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_SFMA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.48 SWMRESS_SFMA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SWMRESS_SFMA_0	Software Module Reset Status for SFMA0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.44 SWMRESA_TAPA — Software Module Reset Assertion Register for TAPA

This register is used to generate a Module Reset for each TAPA.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FF98 $0A20_H$
Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_TAPA_3	SWMRESA_TAPA_2	SWMRESA_TAPA_1	SWMRESA_TAPA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 9.49 SWMRESA_TAPA Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	SWMRESA_TAPA_3	Software Module Reset Trigger for TAPA3 0 : No function 1 : Generate Module Reset
2	SWMRESA_TAPA_2	Software Module Reset Trigger for TAPA2 0 : No function 1 : Generate Module Reset
1	SWMRESA_TAPA_1	Software Module Reset Trigger for TAPA1 0 : No function 1 : Generate Module Reset
0	SWMRESA_TAPA_0	Software Module Reset Trigger for TAPA0 0 : No function 1 : Generate Module Reset

9.4.45 SWMRESS_TAPA — Software Module Reset Status Register for TAPA

This register is used to show the execution status of a Module Reset for TAPA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A24$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_TAPA_3	SWMRESS_TAPA_2	SWMRESS_TAPA_1	SWMRESS_TAPA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.50 SWMRESS_TAPA Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SWMRESS_TAPA_3	Software Module Reset Status for TAPA3 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_TAPA_2	Software Module Reset Status for TAPA2 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_TAPA_1	Software Module Reset Status for TAPA1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_TAPA_0	Software Module Reset Status for TAPA0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.46 SWMRESA_TAUD — Software Module Reset Assertion Register for TAUD

This register is used to generate a Module Reset for TAUD.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A30_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_TAUD_2	SWMRESA_TAUD_1	SWMRESA_TAUD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Table 9.51 SWMRESA_TAUD Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	SWMRESA_TAUD_2	Software Module Reset Trigger for TAUD2 and SELB_TAUD2I 0 : No function 1 : Generate Module Reset
1	SWMRESA_TAUD_1	Software Module Reset Trigger for TAUD1 0 : No function 1 : Generate Module Reset
0	SWMRESA_TAUD_0	Software Module Reset Trigger for TAUD0 0 : No function 1 : Generate Module Reset

9.4.47 SWMRESS_TAUD — Software Module Reset Status Register for TAUD

This register is used to show the execution status of a Module Reset for TAUD.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A34$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_TAUD_2	SWMRESS_TAUD_1	SWMRESS_TAUD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.52 SWMRESS_TAUD Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SWMRESS_TAUD_2	Software Module Reset Status for TAUD2 and SELB_TAUD2I 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_TAUD_1	Software Module Reset Status for TAUD1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_TAUD_0	Software Module Reset Status for TAUD0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.48 SWMRESA_TAUJ_ISO — Software Module Reset Assertion Register for TAUJ

This register is used to generate a Module Reset for each TAUJ (ISO).
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A40_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_TAUJ_1	SWMRESA_TAUJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.53 SWMRESA_TAUJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_TAUJ_1	Software Module Reset Trigger for TAUJ1 0 : No function 1 : Generate Module Reset
0	SWMRESA_TAUJ_0	Software Module Reset Trigger for TAUJ0 0 : No function 1 : Generate Module Reset

9.4.49 SWMRESS_TAUJ_ISO — Software Module Reset Status Register for TAUJ

This register is used to show the execution status of a Module Reset for TAUJ (ISO).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A44$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_TAUJ_1	SWMRESS_TAUJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.54 SWMRESS_TAUJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_TAUJ_1	Software Module Reset Status for TAUJ1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_TAUJ_0	Software Module Reset Status for TAUJ0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.50 SWMRESA_TPBA — Software Module Reset Assertion Register for TPBA

This register is used to generate a Module Reset for TPBA.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A50_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_TPBA_1	SWMRESA_TPBA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.55 SWMRESA_TPBA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_TPBA_1	Software Module Reset Trigger for TPBA1 0 : No function 1 : Generate Module Reset
0	SWMRESA_TPBA_0	Software Module Reset Trigger for TPBA0 0 : No function 1 : Generate Module Reset

9.4.51 SWMRESS_TPBA — Software Module Reset Status Register for TPBA

This register is used to show the execution status of a Module Reset for TPBA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A54$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_TPBA_1	SWMRESS_TPBA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.56 SWMRESS_TPBA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_TPBA_1	Software Module Reset Status for TPBA1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_TPBA_0	Software Module Reset Status for TPBA0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.52 SWMRESA_TSG3 — Software Module Reset Assertion Register for TSG3

This register is used to generate a Module Reset for TSG3.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A60_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESA_TSG3_1	SWMRESA_TSG3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 9.57 SWMRESA_TSG3 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SWMRESA_TSG3_1	Software Module Reset Trigger for TSG31 0 : No function 1 : Generate Module Reset
0	SWMRESA_TSG3_0	Software Module Reset Trigger for TSG30 0 : No function 1 : Generate Module Reset

9.4.53 SWMRESS_TSG3 — Software Module Reset Status Register for TSG3

This register is used to show the execution status of a Module Reset for TSG3.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A64$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWMRESS_TSG3_1	SWMRESS_TSG3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.58 SWMRESS_TSG3 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SWMRESS_TSG3_1	Software Module Reset Status for TSG31 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_TSG3_0	Software Module Reset Status for TSG30 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.54 SWMRESA_OSTM — Software Module Reset Assertion Register for OSTM

This register is used to generate a Module Reset for each OSTM.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0A80_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SWMRESA_OSTM_9	SWMRESA_OSTM_8	SWMRESA_OSTM_7	SWMRESA_OSTM_6	SWMRESA_OSTM_5	SWMRESA_OSTM_4	SWMRESA_OSTM_3	SWMRESA_OSTM_2	SWMRESA_OSTM_1	SWMRESA_OSTM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W

Table 9.59 SWMRESA_OSTM Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When writing, write the value after reset.
9	SWMRESA_OSTM_9	Software Module Reset Trigger for OSTM9 and IC0CKSEL9 0 : No function 1 : Generate Module Reset
8	SWMRESA_OSTM_8	Software Module Reset Trigger for OSTM8 and IC0CKSEL8 0 : No function 1 : Generate Module Reset
7	SWMRESA_OSTM_7	Software Module Reset Trigger for OSTM7 0 : No function 1 : Generate Module Reset
6	SWMRESA_OSTM_6	Software Module Reset Trigger for OSTM6 0 : No function 1 : Generate Module Reset
5	SWMRESA_OSTM_5	Software Module Reset Trigger for OSTM5 0 : No function 1 : Generate Module Reset
4	SWMRESA_OSTM_4	Software Module Reset Trigger for OSTM4 0 : No function 1 : Generate Module Reset
3	SWMRESA_OSTM_3	Software Module Reset Trigger for OSTM3 0 : No function 1 : Generate Module Reset
2	SWMRESA_OSTM_2	Software Module Reset Trigger for OSTM2 0 : No function 1 : Generate Module Reset
1	SWMRESA_OSTM_1	Software Module Reset Trigger for OSTM1 0 : No function 1 : Generate Module Reset
0	SWMRESA_OSTM_0	Software Module Reset Trigger for OSTM0 0 : No function 1 : Generate Module Reset

9.4.55 SWMRESS_OSTM — Software Module Reset Status Register for OSTM

This register is used to generate a Module Reset for each OSTM.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 0A84$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SWMRESS_OSTM_9	SWMRESS_OSTM_8	SWMRESS_OSTM_7	SWMRESS_OSTM_6	SWMRESS_OSTM_5	SWMRESS_OSTM_4	SWMRESS_OSTM_3	SWMRESS_OSTM_2	SWMRESS_OSTM_1	SWMRESS_OSTM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 9.60 SWMRESS_OSTM Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9	SWMRESS_OSTM_9	Software Module Reset Status for OSTM9 and IC0CKSEL9 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
8	SWMRESS_OSTM_8	Software Module Reset Status for OSTM8 and IC0CKSEL8 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
7	SWMRESS_OSTM_7	Software Module Reset Status for OSTM7 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
6	SWMRESS_OSTM_6	Software Module Reset Status for OSTM6 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
5	SWMRESS_OSTM_5	Software Module Reset Status for OSTM5 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
4	SWMRESS_OSTM_4	Software Module Reset Status for OSTM4 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
3	SWMRESS_OSTM_3	Software Module Reset Status for OSTM3 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
2	SWMRESS_OSTM_2	Software Module Reset Status for OSTM2 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
1	SWMRESS_OSTM_1	Software Module Reset Status for OSTM1 0 : Reset execution is not being processed. 1 : Reset execution is being processed.
0	SWMRESS_OSTM_0	Software Module Reset Status for OSTM0 0 : Reset execution is not being processed. 1 : Reset execution is being processed.

9.4.56 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $0C00_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRESFC0	HWBISTFC2	HWBISTFC1	HWBISTFC0	—	ARESFC2	—	ARESFC0	SRES2FC3	SRES2FC2	—	SRES2FC0	SRES1FC1	SRES1FC0	—	PRESFC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	R	W	R	W	W	W	R	W	W	W	R	W

Table 9.61 RESFC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15	DRESFC0	DeepSTOP Reset Flag Clear 0 : No function 1 : Clear flag
14	HWBISTFC2	HW BIST (Field BIST2) Execution Flag Clear 0 : No function 1 : Clear flag
13	HWBISTFC1	HW BIST (Field BIST1) Execution Flag Clear 0 : No function 1 : Clear flag
12	HWBISTFC0	HW BIST (Field BIST0) Execution Flag Clear 0 : No function 1 : Clear flag
11	Reserved	When writing, write the value after reset.
10	ARESFC2	ECM Application Reset Flag Clear 0 : No function 1 : Clear flag
9	Reserved	When writing, write the value after reset.
8	ARESFC0	Software Application Reset Flag Clear 0 : No function 1 : Clear flag
7	SRES2FC3	WDTBA Reset Flag Clear 0 : No function 1 : Clear flag
6	SRES2FC2	ECM System Reset 2 Flag Clear 0 : No function 1 : Clear flag
5	Reserved	When writing, write the value after reset.
4	SRES2FC0	Software System Reset Flag Clear 0 : No function 1 : Clear flag

Table 9.61 RESFC Register Contents (2/2)

Bit Position	Bit Name	Function
3	SRES1FC1	Primary VMON Reset Flag Clear 0 : No function 1 : Clear flag
2	SRES1FC0	External Reset Flag Clear 0 : No function 1 : Clear flag
1	Reserved	When writing, write the value after reset.
0	PRESFC0	Power On Reset Flag Clear 0 : No function 1 : Clear flag

9.4.57 RESFDDC — Reset Factor Clear Register for Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]

This register clears the reset flags of the RESFDD register.
This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FF98 $0C10_H$
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESFDDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.62 RESFDDC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	RESFDDC	Debugger Disconnection Reset Flag Clear 0: No function 1: Clear flag

9.4.58 RESKCPROT0 — Reset Controller Register Key Code Protection Register 0

The RESKCPROT register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 0F00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 9.63 RESKCPROT0 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0.
0	KCE	Key Code Enable bit 0 : Disables write access of protected registers 1 : Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.

Note 2. Write A5A5A501$_H$ to this register to enable writing protected registers.

9.4.59 BOOTCTRL — Boot Control Register

This register controls the Start-Up of each PE. At the time of reset (Power On Reset, System Reset 1/2, Application Reset, DeepSTOP Reset) release, Either PE0 or ICUHMA start according to the Flash Option Byte. For details of Flash Option, see the *RH850/U2A-EVA Group Security User's Manual: Hardware.*

After reset release, PE can be started selectively by asserting the corresponding bit of this register. For details of Startup Order, see **Section 9.5.7.1, CPU's Startup Order**.

Only 1-write is possible for these bits. 0-write is ignored.

Access: This register can be read or written in 32-bit units.

Address: FFFB_2000$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	BC3	BC2	BC1	BC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 9.64 BOOTCTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	BC3	Boot Control 3 This bit triggers the start-up of the PE3. (When OPBT3.PE3_DISABLE = 1, PE3 is inactive regardless of the value of this bit.) 0 : Inactive 1 : Active
2	BC2	Boot Control 2 This bit triggers the start-up of the PE2. (When OPBT3.PE2_DISABLE = 1, PE2 is inactive regardless of the value of this bit.) 0 : Inactive 1 : Active
1	BC1	Boot Control 1 This bit triggers the start-up of the PE1. (When OPBT3.PE1_DISABLE = 1, PE1 is inactive regardless of the value of this bit.) 0 : Inactive 1 : Active
0	BC0	Boot Control 0 This bit triggers the start-up of the PE0. 0 : Inactive 1 : Active

9.4.60 SWSRESA — Software System Reset Assertion Register

This register is used to generate a System Reset 2.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $840C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWSRESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.65 SWSRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWSRESA	Software System Reset Trigger 0 : No function 1 : Generate a System Reset 2

9.4.61 SWARESA — Software Application Reset Assertion Register

This register is used to generate an Application Reset.

Do not writing to SWARESA during Cyclic RUN mode.

This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 8410_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWARESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.66 SWARESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SWARESA	Software Application Reset Trigger 0 : No function 1 : Generate an Application Reset

9.4.62 RESC — Reset Configuration Register

This register contains configuration settings for the behavior of the device during reset.

Access: This register can be read or written in 32-bit units.

Address: FF98 8414$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 9.67 RESC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RESC0	ECM Reset Configuration in RUN mode*1 0 : ECM Reset Generates a System Reset 2 1 : ECM Reset Generates an Application Reset

Note 1. In other than RUN mode, ECM Reset Generates System Reset 2 regardless of the value of RESC0.

9.4.63 RESF — Reset Factor Register

This register indicates whether a Reset occurred since it was cleared last time.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 8500$_H$

Value after reset: 0000 X005$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICUMARESF0	ICUMSRESF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRESF0	HWBISTF2	HWBISTF1	HWBISTF0	—	ARESF2	—	ARESF0	SRES2F3	SRES2F2	—	SRES2F0	SRES1F1	SRES1F0	—	PRESF0
Value after reset	0*1	0*1	0*1	x	0	0*1	0	0*1	0*1	0*1	0	0*1	0*1	1*1	0	1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Value after Power On Reset

Table 9.68 RESF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	ICUMARESF0	ICUMHA Software Application Reset Flag This flag is a status controlled by ICUMHA. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*. This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
16	ICUMSRESF0	ICUMHA Software System Reset Flag This flag is a status controlled by ICUMHA. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*. This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
15	DRESF0	DeepSTOP Reset Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
14	HWBISTF2	HW BIST (Field BIST2) Execution Flag*1 This flag will always be asserted after a preceding DeepSTOP Reset if the Field BIST2 is executed. In this case the corresponding DeepSTOP Reset Flag indication is asserted as well. 0 : Field BIST2 was not executed 1 : Field BIST2 was executed
13	HWBISTF1	HW BIST (Field BIST1) Execution Flag*1 This flag will always be asserted after a preceding DeepSTOP Reset if the Field BIST1 is executed. In this case the corresponding DeepSTOP Reset Flag indication is asserted as well. 0 : Field BIST1 was not executed 1 : Field BIST1 was executed
12	HWBISTF0	HW BIST (Field BIST0) Execution Flag*1 This flag will always be asserted after a preceding Power On or System Reset 1/2 if the Field BIST0 is executed. In this case the corresponding Power On or System Reset 1/2 Flag indication is asserted as well. 0 : Field BIST0 was not executed 1 : Field BIST0 was executed

Table 9.68 RESF Register Contents (2/2)

Bit Position	Bit Name	Function
11	Reserved	When read, the value after reset is returned.
10	ARESF2	ECM Application Reset Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
9	Reserved	When read, the value after reset is returned.
8	ARESF0	Software Application Reset Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
7	SRES2F3	WDTBA Reset Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
6	SRES2F2	ECM System Reset 2 Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
5	Reserved	When read, the value after reset is returned.
4	SRES2F0	Software System Reset Flag This flag is cleared to 0 by Power On Reset, Primary VMON Reset. 0 : No reset occurred 1 : Reset has occurred
3	SRES1F1	Primary VMON Reset Flag This flag is only cleared by a Power On Reset 0 : No reset occurred 1 : Reset has occurred
2	SRES1F0	External Reset Flag This flag is also set by a Power On Reset. 0 : No reset occurred 1 : Reset has occurred
1	Reserved	When read, the value after reset is returned.
0	PRESF0	Power On Reset Flag This flag is set by a Power On Reset. 0 : No reset occurred 1 : Reset has occurred

Note 1. HWBISTFn is also set in the condition of “BIST not executed” by setting $\overline{\text{TRST}}$ pin to High. For the condition of BIST execution, refer to **Table 44.475** and **Table 44.476** about the condition.

Table 9.69 Reset Factor Register Value in Usecase

Reset Factor	RESF.PRESF0	RESF.SRES1F0	RESF.SRES1F1	RESF.SRES2F0	RESF.SRES2F2	RESF.SRES2F3	RESF.ARESF0	RESF.ARESF2	RESF.DRESF0	RESF.ICUMSRESF0	RESF.ICUMARESF0
Power On Reset	1	1	0	0	0	0	0	0	0	0	0
External Reset	—	1	—	X	X	X	X	X	X	X	X
Primary VMON Reset	—	—	1	0	0	0	0	0	0	0	0
Software System Reset	—	—	—	1	—	—	—	—	—	—	—
ICUM Software System Reset	—	—	—	—	—	—	—	—	—	1	—
ECM System Reset	—	—	—	—	1	—	—	—	—	—	—
WDTBA Reset	—	—	—	—	—	1	—	—	—	—	—
Software Application Reset	—	—	—	—	—	—	1	—	—	—	—
ECM Application Reset	—	—	—	—	—	—	—	1	—	—	—
ICUM Application System Reset	—	—	—	—	—	—	—	—	—	—	1
DeepSTOP Reset	—	—	—	—	—	—	—	—	1	—	—

Note: —: keep
X: Unknown

9.4.64 RESFDD — Reset Factor Register for Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]

This register indicates whether a Debugger Disconnection Reset occurred since it was cleared last time.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 8510$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESFDD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Value after Power On Reset

Table 9.70 RESFDD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	RESFDD	Debugger Disconnection Reset Flag This flag is cleared to 0 by Power On Reset. 0: No reset occurred 1: Reset has occurred

9.5 Operation

9.5.1 Reset Categories

(1) Power On Reset

The Power On Reset represents a Start-Up of the device during power up. This reset occurs when SYSVCC supply voltage is under a definite level and is released when SYSVCC supply voltage is over the definite level. In this case entire microcontroller is initialized.

VLVI, LS IntOSC, Mode latch are excluded from reset. Power On Reset sources are as follows.

- POC Reset: Power On Clear Reset
- Debugger Reset: Reset generated by a debugger command

(2) System Reset 1

System Reset 1 is identical to the Power On Reset with the following exceptions. The Reset Factor Register of the Reset Controller and the Error Source Status Register of the Error Control Module and a part of Voltage Monitor registers are excluded from this reset. The Operating mode controller is initialized only in case External Reset is a reset source of System Reset 1. System Reset 1 sources are as follows.

- External Reset
- Primary Voltage Monitor (VMON) Reset (VCC)
- Primary Voltage Monitor (VMON) Reset (Other: AWOVDD, ISOVDD, E0VCC)
- Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]

(3) System Reset 2

System Reset 2 is identical to the System Reset 1 with the following exceptions. A part of the Clock Controller registers is preserved. System Reset 2 sources are as follows.

- Software Reset by writing the associated register
- Error Control Module (ECM) Reset (The Reset Category is configured by the Reset Configuration Register)
- Watchdog Timer Reset (WDTBA) for Always-On area (AWO area)

(4) Application Reset

Application Reset is used for fast re-initialization of the application. This reset is basically identical to the System Reset 2 with the following exceptions.
The configuration data stored in the FLASH is not reloaded. Also, HW BIST is not executed too. Therefore, fast re-initialization is realized. Application Reset sources are as follows.

- Software Application Reset by writing the associated register
- Error Control Module (ECM) Reset (The Reset Category is configured by the Reset Configuration Register)

Application Reset is valid only in RUN mode.
ECM Reset is issued as System Reset 2 in Chip standby mode.

(5) DeepSTOP Reset

At the transition to DeepSTOP mode, initializes all the Isolated area (ISO area).

(6) Module Reset

Module Reset is used for re-initialization of dedicated device function like certain peripherals.

Table 9.71 Modules that participate in Module Reset (1/3)

Module Reset Assertion register	Bit	Module
SWMRESA_RSCFD	SWMRESA_RSCFD_1	RSCFD1, E7RC11, E7RC12, E7RC13
	SWMRESA_RSCFD_0	RSCFD0, E7RC01, E7RC02, E7RC03
SWMRESA_FLXA	SWMRESA_FLXA_1	FLXA1, E7FR10, E7FR11, E7FR12
	SWMRESA_FLXA_0	FLXA0, E7FR00, E7FR01, E7FR02
SWMRESA_GTM	SWMRESA_GTM_0	GTM0, GTM0_1, E7GT00, E7GT01, E7GT10, E7GT11, E7GT20, E7GT21, E7GT30, E7GT31
SWMRESA_ETNB	SWMRESA_ETNB_1	ETNB1, E7GE00, E7GE01
	SWMRESA_ETNB_0	ETNB0, E7ME00, E7ME01
SWMRESA_RSENT*1	SWMRESA_RSENT_7	RSENT7
	SWMRESA_RSENT_6	RSENT6
	SWMRESA_RSENT_5	RSENT5
	SWMRESA_RSENT_4	RSENT4
	SWMRESA_RSENT_3	RSENT3
	SWMRESA_RSENT_2	RSENT2
	SWMRESA_RSENT_1	RSENT1
	SWMRESA_RSENT_0	RSENT0
SWMRESA_MSPI*2	SWMRESA_MSPI_9	MSPI9, MSPI9_INTF, E7MS09
	SWMRESA_MSPI_8	MSPI8, MSPI8_INTF, E7MS08
	SWMRESA_MSPI_7	MSPI7, MSPI7_INTF, E7MS07
	SWMRESA_MSPI_6	MSPI6, MSPI6_INTF, E7MS06
	SWMRESA_MSPI_5	MSPI5, MSPI5_INTF, E7MS05
	SWMRESA_MSPI_4	MSPI4, MSPI4_INTF, E7MS04
	SWMRESA_MSPI_3	MSPI3, MSPI3_INTF, E7MS03
	SWMRESA_MSPI_2	MSPI2, MSPI2_INTF, E7MS02
	SWMRESA_MSPI_1	MSPI1, MSPI1_INTF, E7MS01
	SWMRESA_MSPI_0	MSPI0, MSPI0_INTF, E7MS00
SWMRESA_RLIN3	SWMRESA_RLIN3_23	RLIN323
	SWMRESA_RLIN3_22	RLIN322
	SWMRESA_RLIN3_21	RLIN321
	SWMRESA_RLIN3_20	RLIN320
	SWMRESA_RLIN3_19	RLIN319
	SWMRESA_RLIN3_18	RLIN318
	SWMRESA_RLIN3_17	RLIN317
	SWMRESA_RLIN3_16	RLIN316
	SWMRESA_RLIN3_15	RLIN315
	SWMRESA_RLIN3_14	RLIN314
	SWMRESA_RLIN3_13	RLIN313
	SWMRESA_RLIN3_12	RLIN312
	SWMRESA_RLIN3_11	RLIN311
	SWMRESA_RLIN3_10	RLIN310

Table 9.71 Modules that participate in Module Reset (2/3)

Module Reset Assertion register	Bit	Module
SWMRESA_RLIN3	SWMRESA_RLIN3_9	RLIN39
	SWMRESA_RLIN3_8	RLIN38
	SWMRESA_RLIN3_7	RLIN37
	SWMRESA_RLIN3_6	RLIN36
	SWMRESA_RLIN3_5	RLIN35
	SWMRESA_RLIN3_4	RLIN34
	SWMRESA_RLIN3_3	RLIN33
	SWMRESA_RLIN3_2	RLIN32
	SWMRESA_RLIN3_1	RLIN31
	SWMRESA_RLIN3_0	RLIN30
SWMRESA_ADCJ_ISO	SWMRESA_ADCJ_1	ADCJ1
	SWMRESA_ADCJ_0	ADCJ0, AVSEG
SWMRESA_CXPI	SWMRESA_CXPI_3	CXP13
	SWMRESA_CXPI_2	CXP12
	SWMRESA_CXPI_1	CXP11
	SWMRESA_CXPI_0	CXP10
SWMRESA_MMCA	SWMRESA_MMCA_0	MMCA0, E7MM00, E7MM01
SWMRESA_ENCA	SWMRESA_ENCA_1	ENCA1
	SWMRESA_ENCA_0	ENCA0
SWMRESA_PSI5*3	SWMRESA_PSI5_3	PSI53
	SWMRESA_PSI5_2	PSI52
	SWMRESA_PSI5_1	PSI51
	SWMRESA_PSI5_0	PSI50
SWMRESA_PSI5S	SWMRESA_PSI5S_1	PSI5S1
	SWMRESA_PSI5S_0	PSI5S0
SWMRESA_PWMD	SWMRESA_PWMD	PWM-Diag (PWBA0, PWGC0-95, PWSD0, SLPW, PWGCINTF)
SWMRESA_RHSIF	SWMRESA_RHSIF_0	RHSIF0
SWMRESA_RIIC	SWMRESA_RIIC_1	RIIC1
	SWMRESA_RIIC_0	RIIC0
SWMRESA_SCI3	SWMRESA_SCI3_2	SCI32
	SWMRESA_SCI3_1	SCI31
	SWMRESA_SCI3_0	SCI30
SWMRESA_SFMA	SWMRESA_SFMA_0	SFMA0
SWMRESA_TAPA	SWMRESA_TAPA_3	TAPA3
	SWMRESA_TAPA_2	TAPA2
	SWMRESA_TAPA_1	TAPA1
	SWMRESA_TAPA_0	TAPA0
SWMRESA_TAUD	SWMRESA_TAUD_2	TAUD2, SELB_TAUD2I
	SWMRESA_TAUD_1	TAUD1
	SWMRESA_TAUD_0	TAUD0
SWMRESA_TAUJ_ISO	SWMRESA_TAUJ_1	TAUJ1
	SWMRESA_TAUJ_0	TAUJ0

Table 9.71 Modules that participate in Module Reset (3/3)

Module Reset Assertion register	Bit	Module
SWMRESA_TPBA	SWMRESA_TPBA_1	TPBA1
	SWMRESA_TPBA_0	TPBA0
SWMRESA_TSG3	SWMRESA_TSG3_1	TSG31
	SWMRESA_TSG3_0	TSG30
SWMRESA_OSTM	SWMRESA_OSTM_9	OSTM9, IC0CKSEL9
	SWMRESA_OSTM_8	OSTM8, IC0CKSEL8
	SWMRESA_OSTM_7	OSTM7
	SWMRESA_OSTM_6	OSTM6
	SWMRESA_OSTM_5	OSTM5
	SWMRESA_OSTM_4	OSTM4
	SWMRESA_OSTM_3	OSTM3
	SWMRESA_OSTM_2	OSTM2
	SWMRESA_OSTM_1	OSTM1
	SWMRESA_OSTM_0	OSTM0

Note 1. Register RSENTTSSEL is not initialized by Module Reset.
Note 2. Register MSPITGCTLn, MSPITGDMAALT and MSPITGDTSALT are not initialized by Module Reset.
Note 3. Register PSI5TSSEL is not initialized by Module Reset.

(7) JTAG Reset

Debug Reset is used for re-initialization of debug module. For details, refer to **Section 50, Debugging and Calibration**.

9.5.2 Reset Sources

(1) POC Reset

The Power On Reset identifies a Start-Up of the device during power up. The power up condition is detected by a Power On Clear circuit (POC). It permanently compares the power supply voltage SYSVCC with an internal reference voltage (Vpoc). If SYSVCC lowers below the internal reference voltage (SYSVCC < Vpoc), Power On Reset is generated. For details, refer to **Section 55, Electrical Characteristics**.

(2) Debugger Reset

MCU transits to the internal reset state when the MCU is connected with the debug tool and receives the reset command from the debugger. It can trigger a Power On Reset. For details, refer to **Section 50, Debugging and Calibration** and the user's manual of the debugger.

(3) External Reset

A dedicated Reset Input pin ($\overline{\text{RESET}}$) is available for initialization. The associated buffer has input hysteresis. The External Reset is active low.

In case of an open reset input, the device will be initialized via the internal pull-down of the Reset Input Buffer. This ensures that the device is always in a safe state.

In case Power-up and Power-down, $\overline{\text{RESET}}$ should be low. In case Power-up, $\overline{\text{RESET}}$ must be asserted until all power supply exceeds the threshold.

Until VCC, SYSVCC, AWOVDD, ISOVDD and E0VCC exceed the threshold, Internal reset is generated.

After Internal reset and $\overline{\text{RESET}}$ are released, HW BIST and RAM initialization are executed.

For details on the required $\overline{\text{RESET}}$ low period and the threshold of all power supply refer to **Section 55, Electrical Characteristics**.

In case of External Reset excluding power-up/down, the low pulse width of the $\overline{\text{RESET}}$ input must be greater than the value of noise-filter characteristic to activate the reset. For details on the required $\overline{\text{RESET}}$ low pulse width (tWRSL), refer to **Section 55, Electrical Characteristics**.

(4) Primary VMON Reset

The VMON Reset can be asserted when each supply voltage is outside the operating range. The VMON Reset operation is independent from External Reset.
Following power domains can be detected.

- AWOVDD
- ISOVDD
- VCC
- E0VCC.

For details, refer to **Section 11, Power Supply Voltage Monitor**.

(5) Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]

A System Reset 1 will be trigged in case the debugger is disconnected from a device. The disconnection is detected by a falling edge of $\overline{\text{TRST}}$ during Security is Unlock.

The detection can be disable by FLASH Option Byte located to Security Setting Area. Refer to **Section 47.8, Security Setting Area**.

(6) Software Reset

The device supports multiple software resets that can trigger the reset of dedicated Reset Domains:

- Software System Reset Assertion Register (SWSRESA) triggers a System Reset 2.
- Software Application Reset Assertion Register (SWARESA) triggers an Application Reset.
- Software Module Reset Assertion Register n (SWMRESA_*) triggers reset of each peripheral domain.
- Software System Reset Assertion Register (SWSRESA_ICUM) for ICUMHA triggers a System Reset 2. For details, refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*.
- Software Application Reset Assertion Register (SWARESA_ICUM) for ICUMHA triggers an Application Reset. For details, refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Each software reset can be triggered when software writes to the associated Reset Assertion Register.

Each Reset Assertion Register is protected against unintended access by configuring the Key Code protection register. For details, see **Section 9.4.58, RESKCPROT0 — Reset Controller Register Key Code Protection Register 0** .

Before writing to module reset assertion register, verify that the corresponding peripheral module is in the idle state and other Module Reset is not being executed. (After writing to module reset assertion register, this register can automatically release the module reset.)

The procedure of Module reset execution is shown in **Figure 9.1**. For the procedure of checking idle state, see the section of corresponding peripheral module.

CAUTION

During the Module Reset, do not access that module.

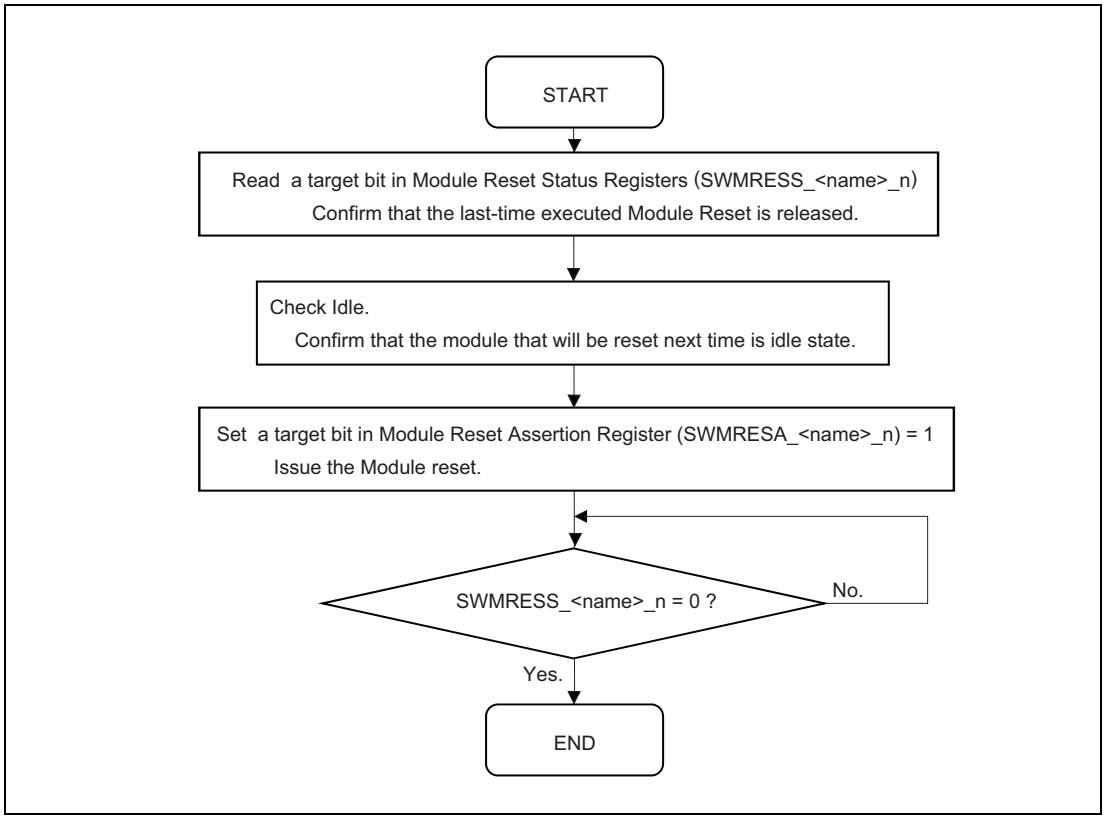


Figure 9.1 Flowchart of module Reset

(7) ECM Reset

All failures which are detectable by the Error Control Module (ECM) generate dedicated internal reset or interrupts (FENMI or EINT) instead. The failure signals and system behavior can be configured in the ECM (see **Section 45, Error Control Module (ECM)**). Optionally the device failures can be configured to be used as a Reset Source.

The ECM Reset can generate a System Reset 2 or an Application Reset. The behavior can be configured by the Reset Configuration Register.

(8) WDTBA Reset

The Window Watch Dog Timer (WDTBA) module can generate a WDTBA Reset.

(9) JTAG Reset (by $\overline{\text{TRST}}$)

A dedicated Reset Input pin ($\overline{\text{TRST}}$) is available for initialization of the debug function.

If $\overline{\text{TRST}}$ is open, the debug function will be always initialized via the internal pull-down of $\overline{\text{TRST}}$. For details on the filter characteristics, refer to **Section 55, Electrical Characteristics**. The Debug function is described in **Section 50, Debugging and Calibration**.

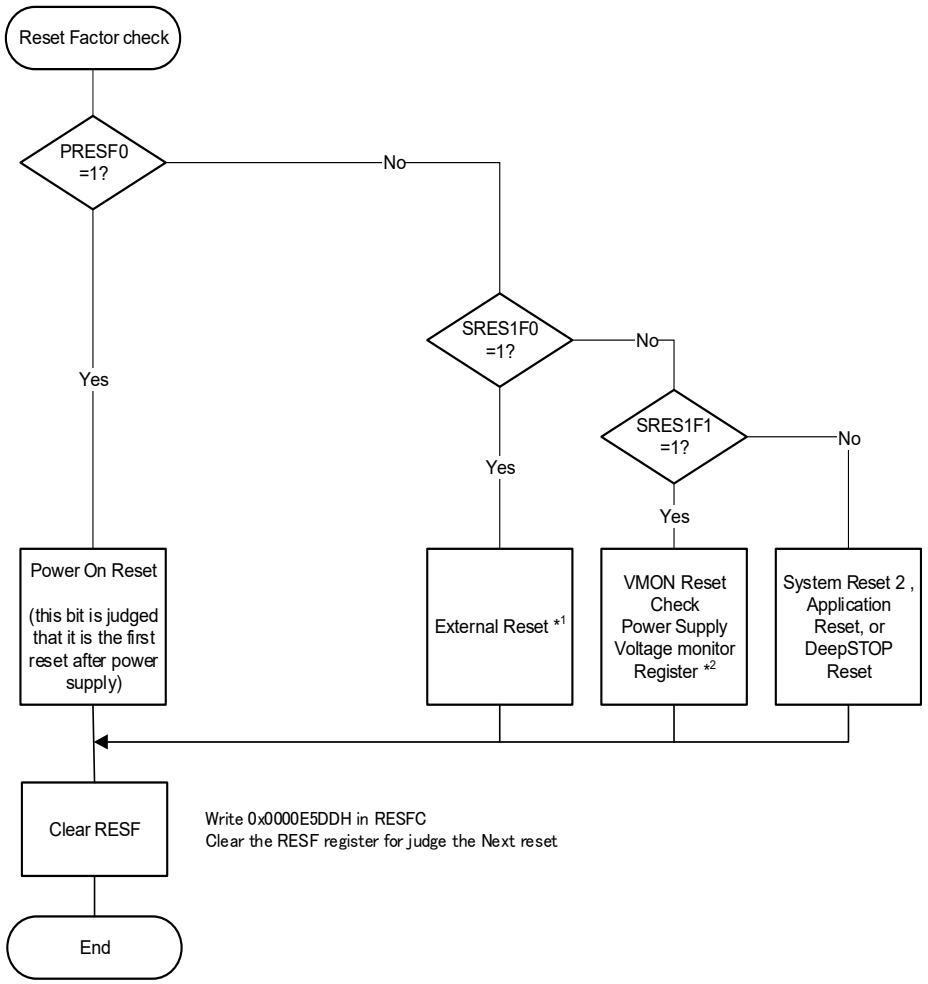
The Debug Reset is active low.

9.5.3 Reset Flags

Any reset root cause can be identified by SW in the Reset Factor Register (RESF) . Each reset source will be indicated by one bit when it occurs. The status can be read out after the reset has been executed.

For details, refer to the **Section 9.4.63, RESF — Reset Factor Register**.

The method for determining the reset factors is shown in **Figure 9.2**.



Note 1. SRES1F0 is a high priority flag. Do not use ARESF2 - 0 and SRES2F3 - 0 for judgment when SRES1F0 = 1 at the time of reset factor judgment.

Note 2. See **Section 11, Power Supply Voltage Monitor**.

Figure 9.2 Flowchart of Reset Factor Result

9.5.4 Read Configuration Data from FLASH

Reading configuration data from FLASH memory is executed by Power On Reset, System Reset 1, System Reset 2 and DeepSTOP Reset. If an ECC error is detected during FACI reset transfer, the ECC error is notified to ECM. The microcontroller will be kept in reset state and will not start up if the FACI reset transfer error is detected in Normal Operation Mode or User Boot Mode. For details, see **Section 51.17.1, FACI reset transfer**.

9.5.5 HW BIST

HW BIST is executed by Power On Reset, System Reset 1/2 and DeepSTOP Reset. In System Reset 2 and DeepSTOP Reset, HW BIST execution can be disabled depending on Field BIST0/1/2 (FBIST0/1/2) control register (BSEQ0CTL/BSEQ1CTL/ BSEQ2CTL). For details, see **Section 44.6, BIST**.

The HW BIST has three kinds of following.

- The Field BIST0 is executed before transition to RUN mode after Power On Reset and System Reset 1/2.
- The Field BIST1 is executed before returning to RUN mode from DeepSTOP mode.
- The Field BIST2 is executed before returning to Cyclic RUN mode from DeepSTOP mode.

For details, Refer to **Figure 15.1, Transition to Chip Standby Mode** in **Section 15, Standby Controller (STBC)**.

9.5.6 RAM Initialization

RAM can be initialized (to all0) after each reset occurs, depending on a Flash Option Byte or RAM Initialization Mode Control Register corresponding to each RAM. For details of target RAMs, see **Section 52, RAM**.

In Power On Reset and System reset 1/2, and DeepSTOP Reset, RAM initialization is executed according to Flash Option Byte value. For Flash Option Byte, see **Section 51, Flash Memory**.

In Application Reset and Module Reset, RAM initialization execution can be controlled by STAC register.

9.5.7 Start Up of Cores

9.5.7.1 CPU's Startup Order

Figure 9.3 shows CPU's startup order. Each CPU start in the following.

ICUMHA starts first if ICUMHA is enabled. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

(1) CPU0*1 starts after reset (Power On Reset, System Reset 1/2, Application Reset, DeepSTOP Reset).

(2) CPU0 execute Clock gear up*2.

(3) CPU0 starts each CPU by BOOTCTRL. Insert wait time (100 µs) between each CPU startup.

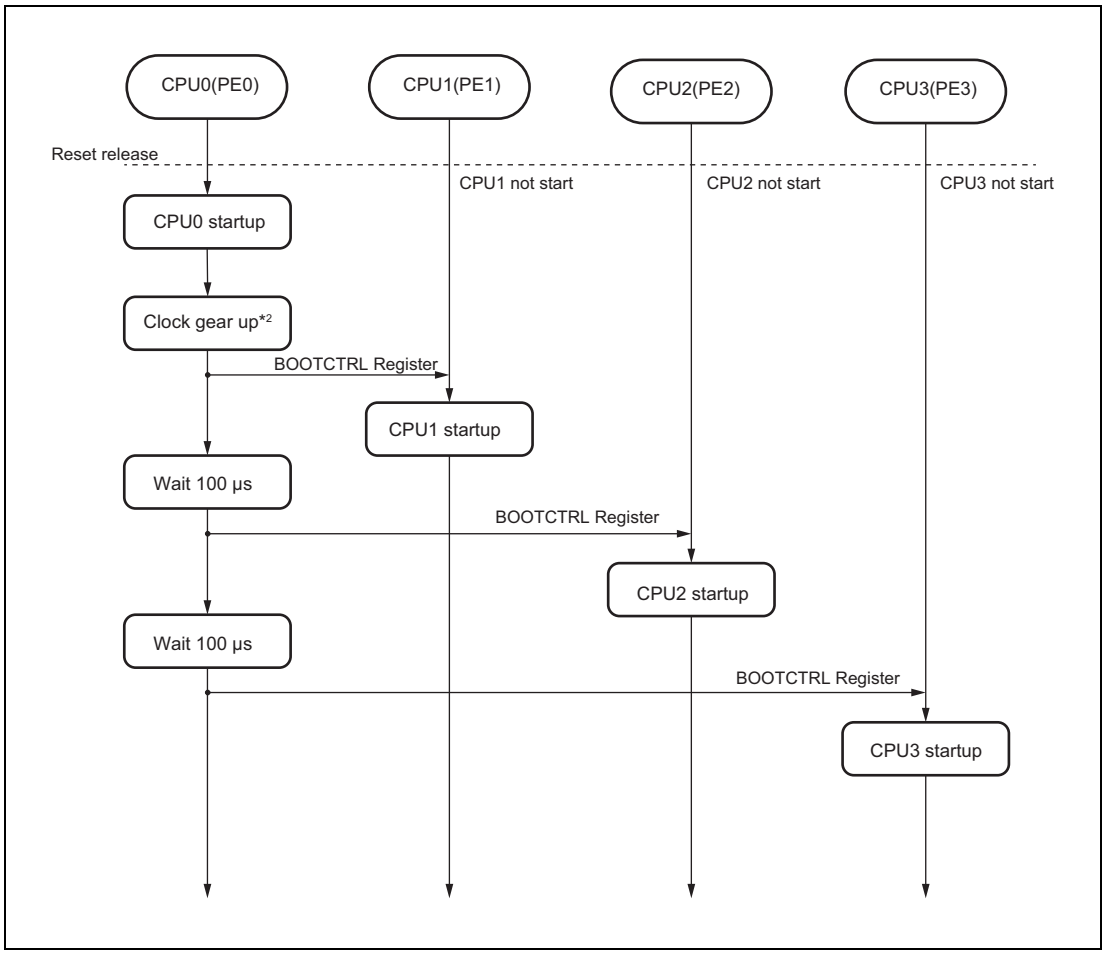


Figure 9.3 CPU's Startup Order in RUN Mode

Note 1. CPU0 activate setting is done by Flash Option Byte. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Note 2. For details of gear up flow, see **Section 13.6.5, Sequence for Shifting the CPU System Clock Gear Up/ Down**.

9.5.8 Reset Mask function

In On-chip debug mode, reset factor can be masked. For details, see **Section 50, Debugging and Calibration**.
Masked reset sources are shown in **Table 9.72**.

Table 9.72 Masked reset sources

No.	Category	Source	Reset Masked
1	Power On Reset	POC Reset (POCRES)	—
		Debugger Reset (by debugger)	—
2	System Reset 1	External Reset	√
		Primary VMON Reset（VCC/ AWOVDD, ISOVDD, E0VCC）	—
		Debugger Disconnection Reset [For U2A16, U2A8 and U2A6 Only]	—
3	System Reset 2	Software System Reset (by SWSRESA, SWSRESA_ICUM)	√
		ECM Reset (if RESC0 = 0)	√
		WDTBA Reset	√
4	Application Reset	Software Application Reset (by SWARESA, SWARESA_ICUM)	√
		ECM Reset (if RESC0 = 1)	√
5	DeepSTOP Reset	DeepSTOP Reset (by DeepSTOP mode trigger)	—
6	Module Reset	Software Module Reset (by SWMRESA_<name>)	—
7	JTAG Reset	JTAG Reset ($\overline{\text{TRST}}$)	—

9.5.9 Reset Output ($\overline{\text{RESETOUT}}$)

When a reset source of Power On Reset, System Reset 1/2, and Application Reset is generated, a reset output signal ($\overline{\text{RESETOUT}}$) is output to the outside. Reset output is used to reset external devices at the same time as a reset is generated inside the microcontroller. For details, see **Section 2.4.4, RESETOUT Function**.

Section 10 Power Supply Circuit

This section describes the power supply and power domains of the RH850/U2A-EVA series.

10.1 Function

This section describes the external voltage connection and internal voltage distribution required to operate the microcontroller. The power supply circuit has a POC (Power On Clear) circuit for safe startup.

The internal circuits are separated into two independent power domains, the Always-On area (AWO) and the Isolated area (ISO).

The power supply of the Always-On area is always on in all operating modes and stand-by modes.

The power supply of the Isolated area can be turned off to reduce the overall power consumption depending on the type of stand-by mode.

For operation of the device, the following voltages are required:

- Power supply voltage SYSVCC for system logic and on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in Always-On area power domain.
- Power supply voltage SVRDRVCC, SVRAVCC and SYSVCC for Switching Voltage Regulator (SVR).
- Power supply voltage ISOVDD for the digital circuits in ISO area power domain. The ISO area power domain is supplied by an external power supply.
- Power supply voltage EnVCC for the I/O ports.
- Power supply voltage AnVCC for the A/D converters and the separated I/O ports.
- Power supply voltage LVDVCC for the LVDS ports in Port Group 2.
- Power supply voltage E0VCC for the LVDS ports in Port Group 4.
- Power supply voltages EMUVCC and EMUVDD for the Aurora ports.
- Power supply voltage GETH0PVCC, GETH0BVCC and GETH0RVCC for SGMII.

10.2 Power Supply Pins

The table below lists all power supply pins and what they are used for.

10.2.1 External Pin List

Table 10.1 Power Supply Pins (1/2)

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	SYSVCC	Power supply for System Logic, Internal voltage regulator and SVR power POC, VMON, VLVI Power supply terminal for PORTS
	VCC	Power supply for FLASH
	SVRAVCC (AD_VCC) SVRDRVCC (DRV_VCC)	Power supply for SVR
	VDD	Power supply terminal for ISOVDD
	VSS	Common Ground
	SVRAVSS (AD_VSS) SVRDRVSS (DRV_VSS)	Ground for SVR
	AWOVCL	External buffer capacitance of regulator
Power supply for I/O port	E0VCC	port group JP0_xx, Pxx_xx*2
	E1VCC	port group Pxx_xx*2
	E2VCC	port group Pxx_xx*2
	VSS	Common Ground
Power supply for A/D converters	A0VCC	Analog circuits of ADCJ, port group AP0_xx, AP1_xx
	A0VSS	
	A0VREFH	
	A1VCC	Analog circuits of ADCJ, port group AP2_xx, AP3_xx
	A1VSS	
	A1VREFH	
	A2VCC	Analog circuits of ADCJ, port group AP4_xx, AP5_xx
	A2VSS	
	A2VREFH	
Power supply for LVDS	LVDVCC	LVDS port in Port Group 2
	E0VCC	LVDS port in Port Group 4
	VSS	Common Ground
Power supply for SGMII	GETH0PVCC	Power supply for SGMII domain P3_8*2
	GETH0RVCC	
	GETH0BVCC	
	GETH0VCL (SGVCL)	
	VSS	Common Ground

Table 10.1 Power Supply Pins (2/2)

Power Supply	Power Supply Pins	Power Supply for
Power supply for EMU*1	EMUVCC	Debug circuits of EMU (Aurora)
	EMUVDD	
	VSS	
Power supply for Debug*1	DVCC	Debug circuits
	DVDD	
	VSS	Common Ground
Power supply for ERAM*1,*3	ERAMVCC	ERAM circuits of FLASH
	ERAMVDD	
	VSS	

Note 1. Only RH850/U2A-EVA support

Note 2. For the power supply of each pin, see the appendix "E02_01_List_of_Pin_Assignment.xlsx".

Note 3. RH850/U2A6 support ERAM and its power is supplied by VDD and VSS.

Note: See **Section 55, Electrical Characteristics** for the voltage range of each power supply.

10.3 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

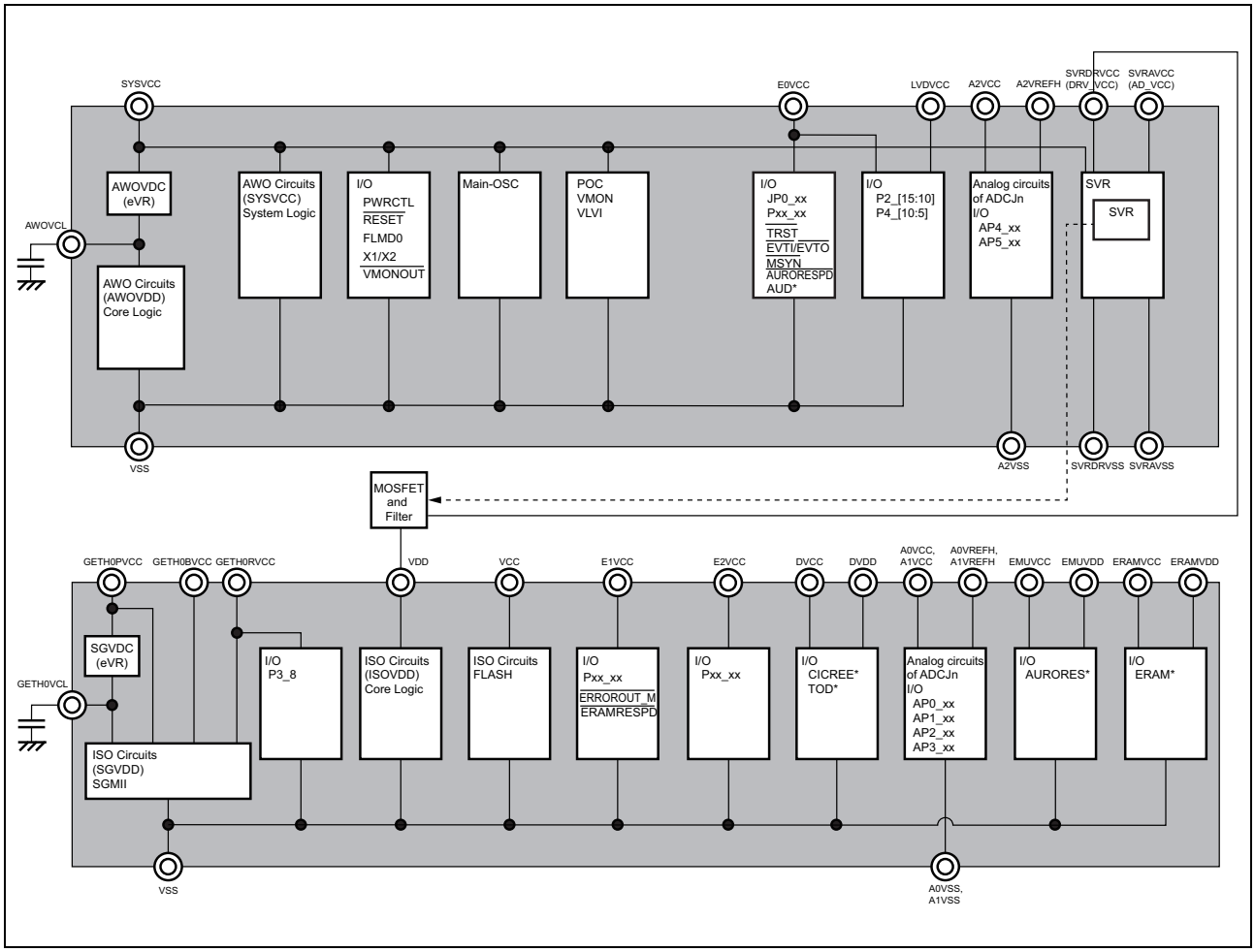


Figure 10.1 Overview of Power Supply Circuit (RH850/U2A-EVA BGA516)

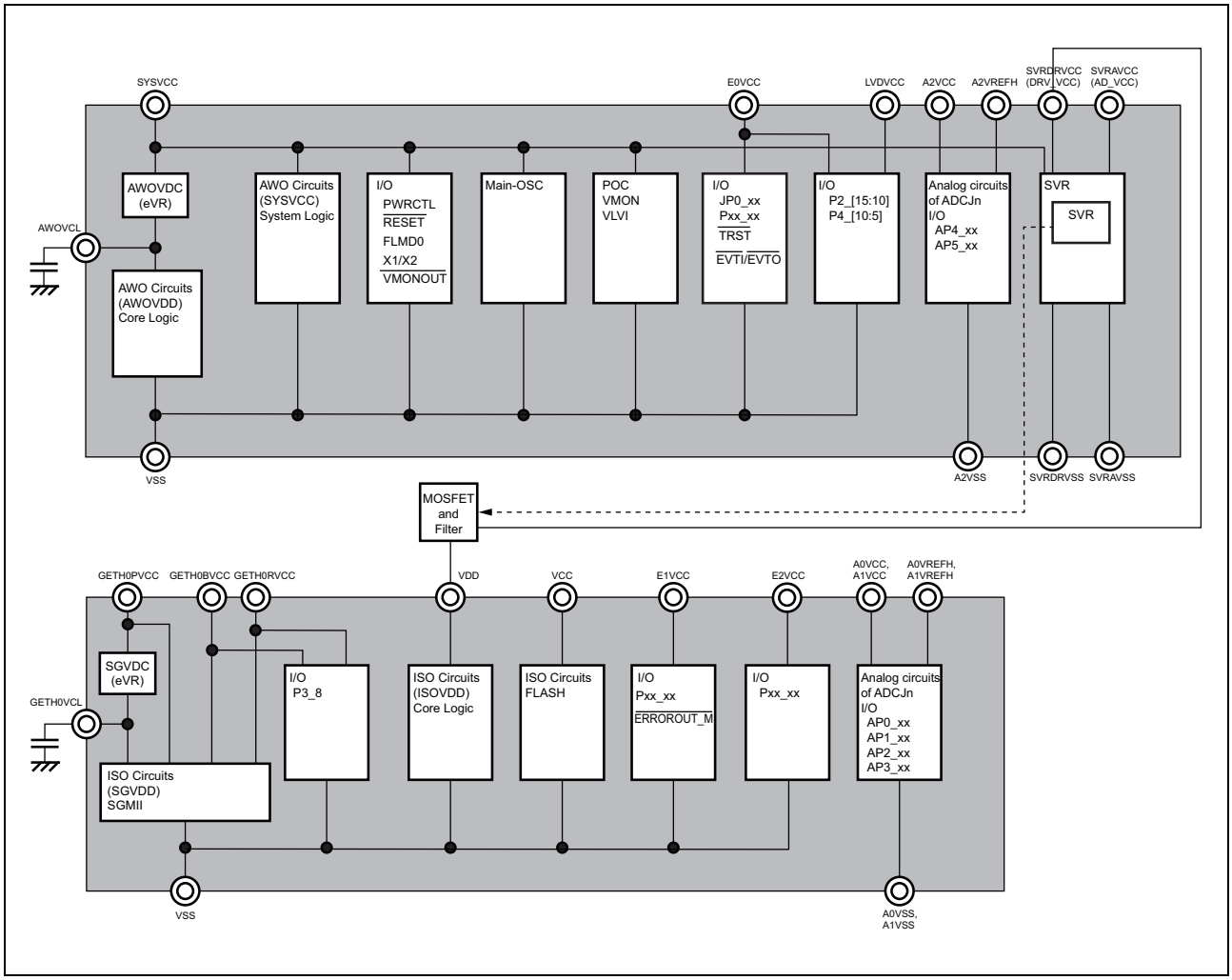


Figure 10.2 Overview of Power Supply Circuit (RH850/U2A16 BGA516)

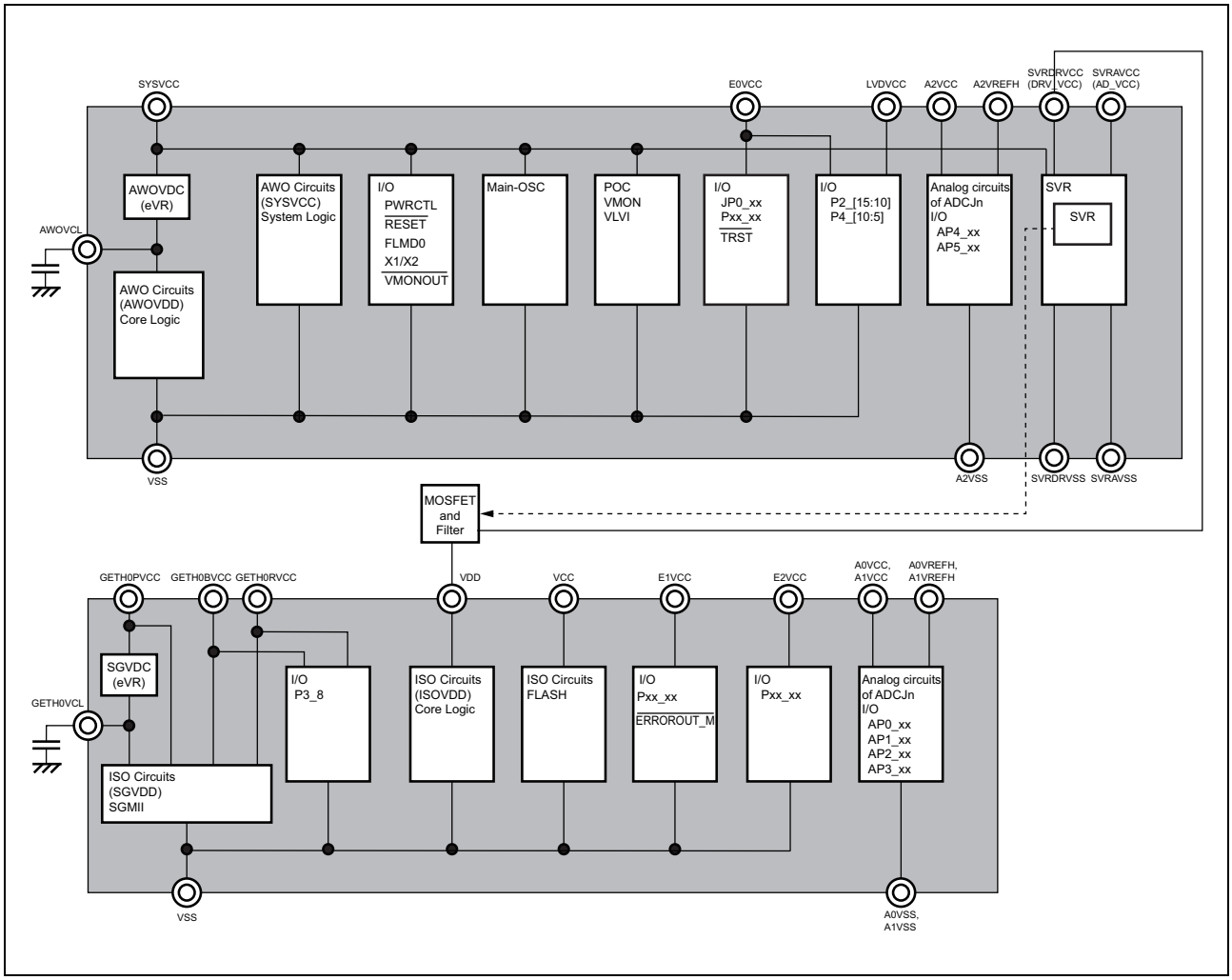


Figure 10.3 Overview of Power Supply Circuit (RH850/U2A16 BGA373)

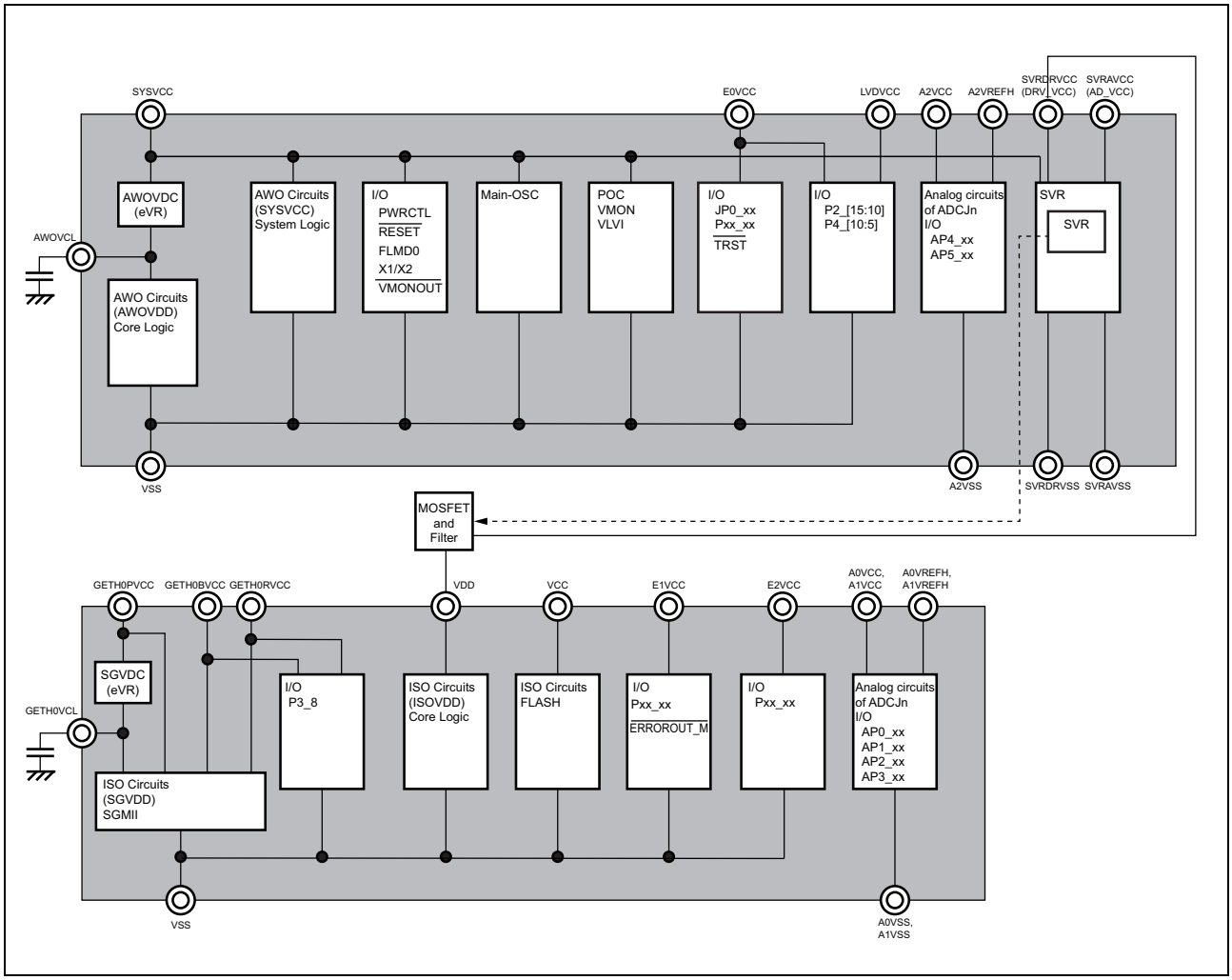


Figure 10.4 Overview of Power Supply Circuit (RH850/U2A16 BGA292)

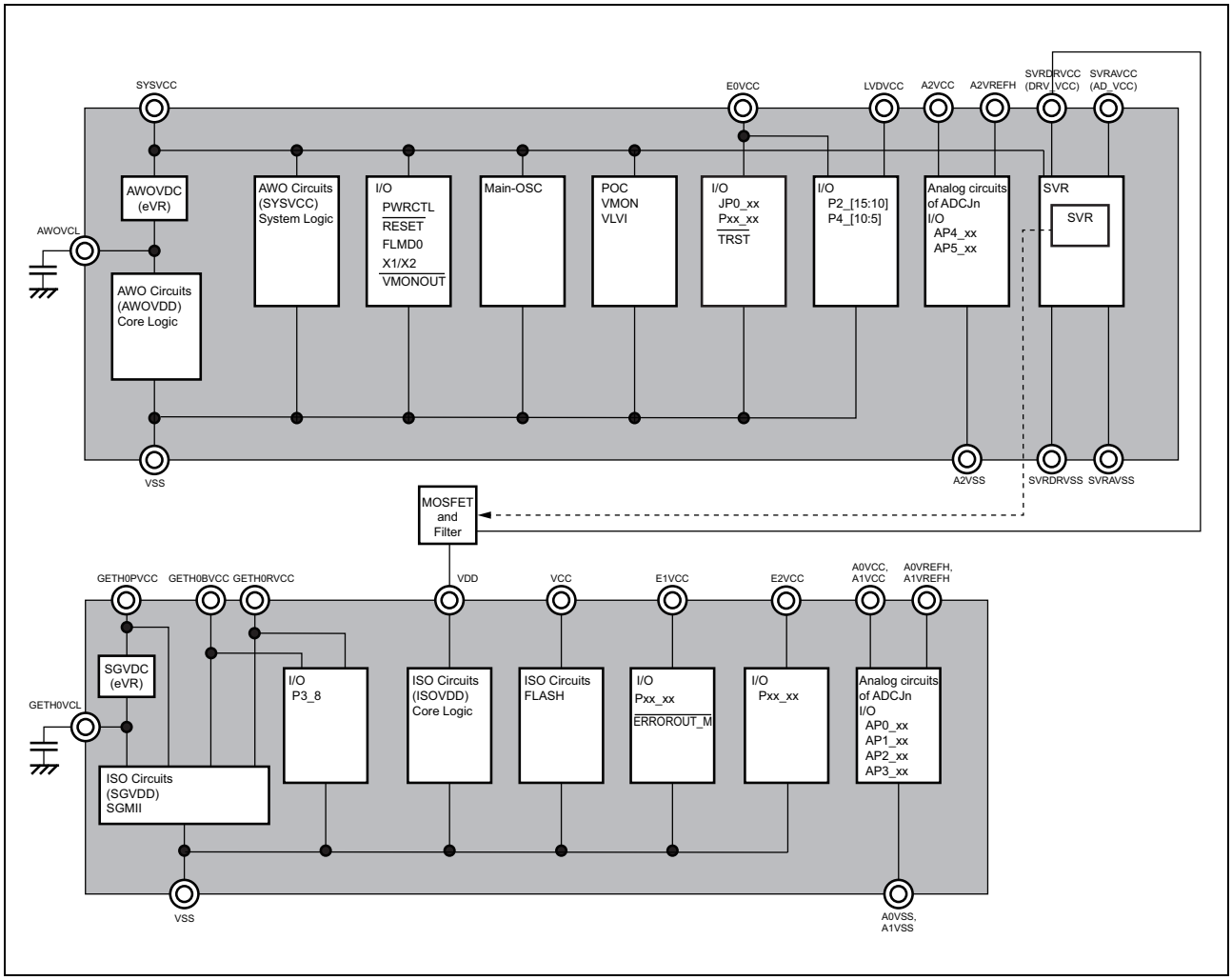


Figure 10.5 Overview of Power Supply Circuit (RH850/U2A8 BGA373)

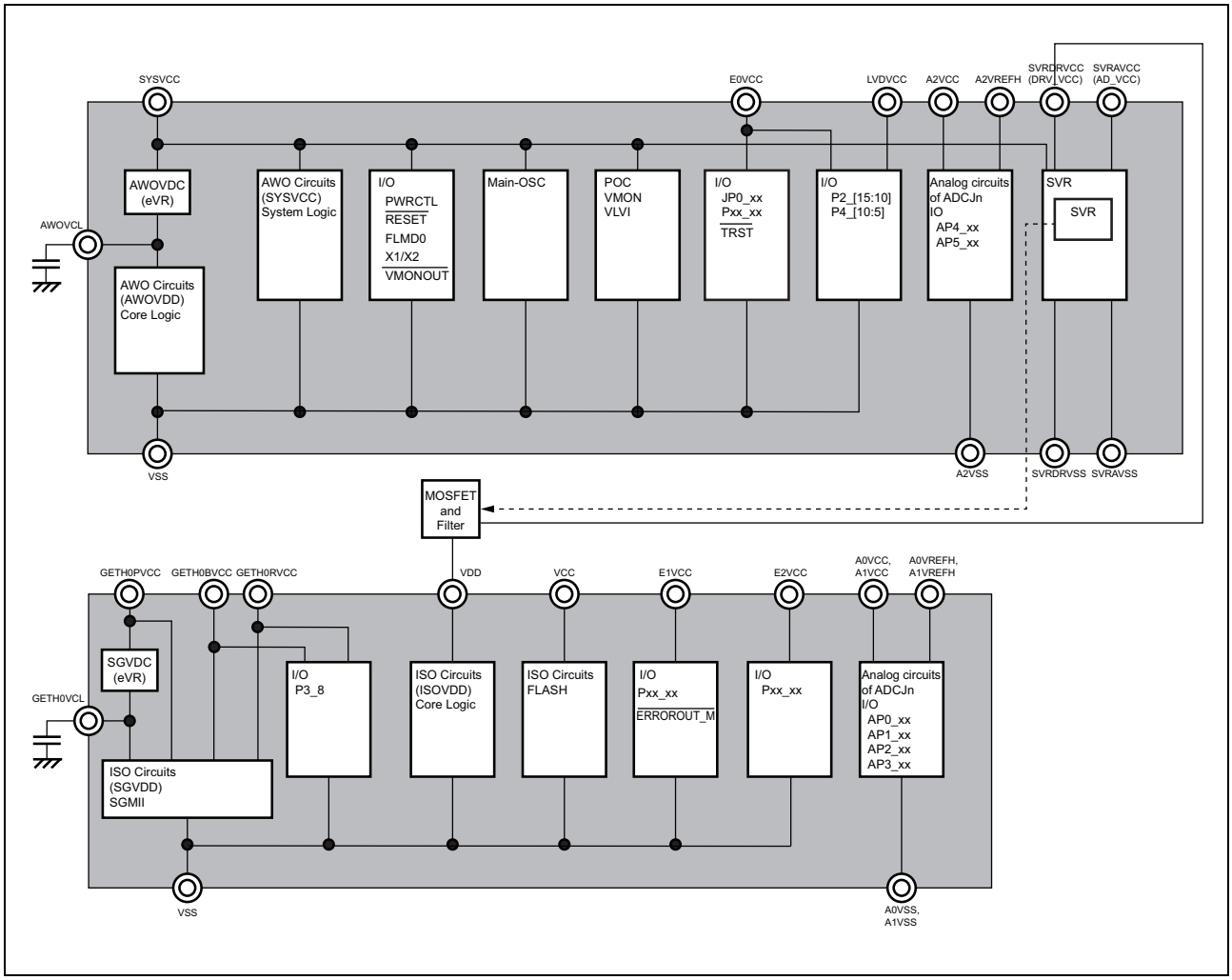


Figure 10.6 Overview of Power Supply Circuit (RH850/U2A8 BGA292)

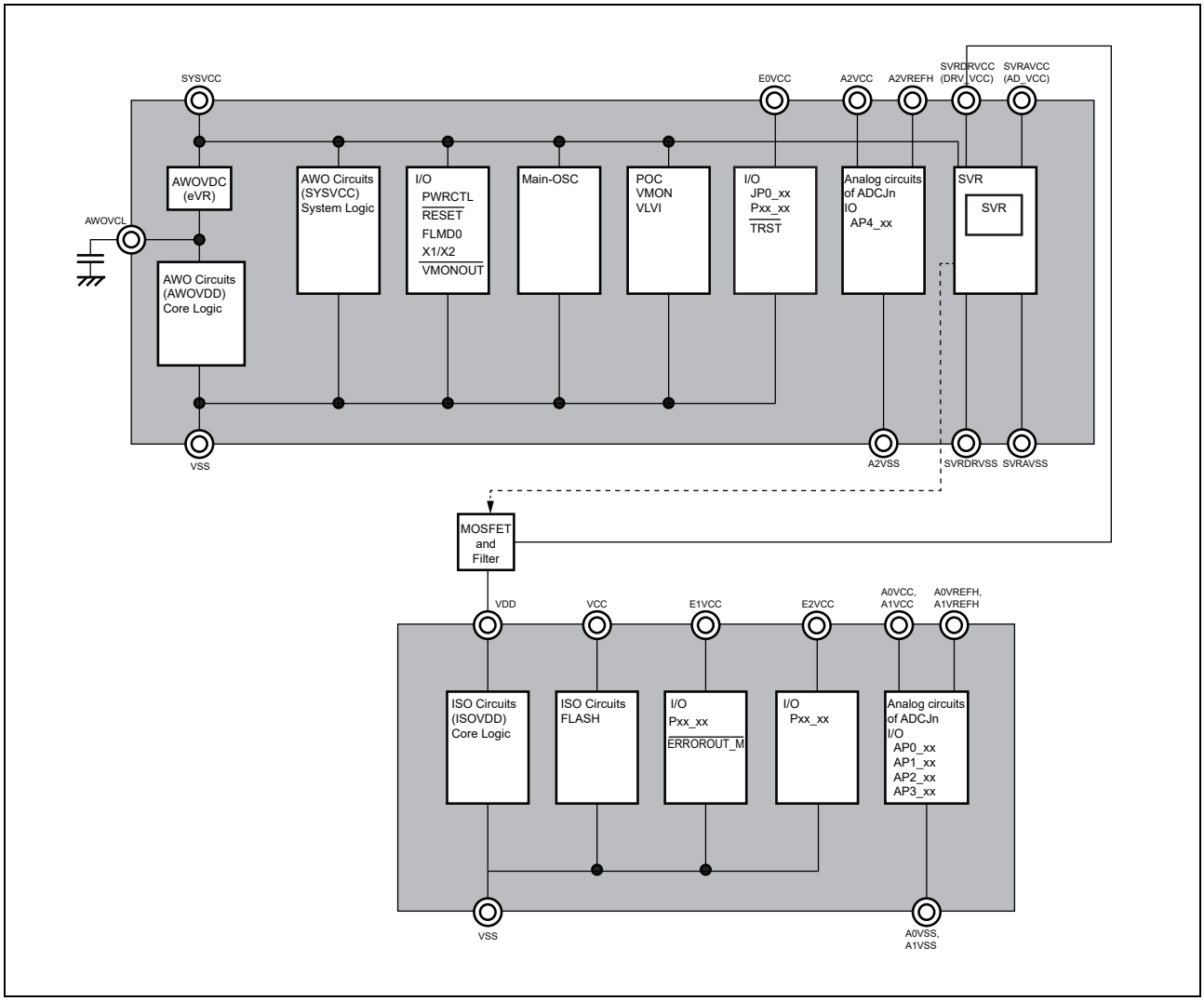


Figure 10.7 Overview of Power Supply Circuit (RH850/U2A6 BGA292)

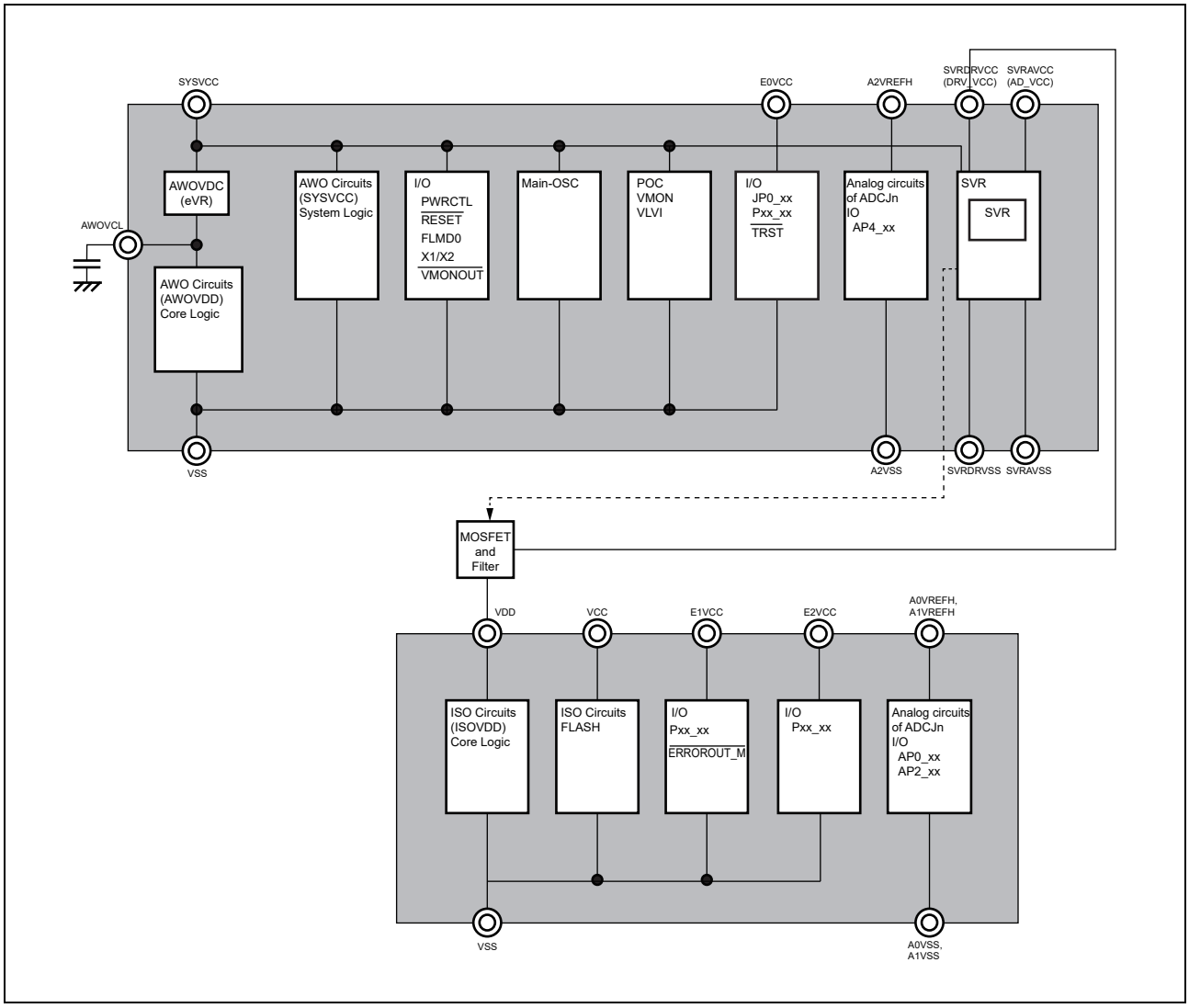


Figure 10.8 Overview of Power Supply Circuit (RH850/U2A6 QFP176)

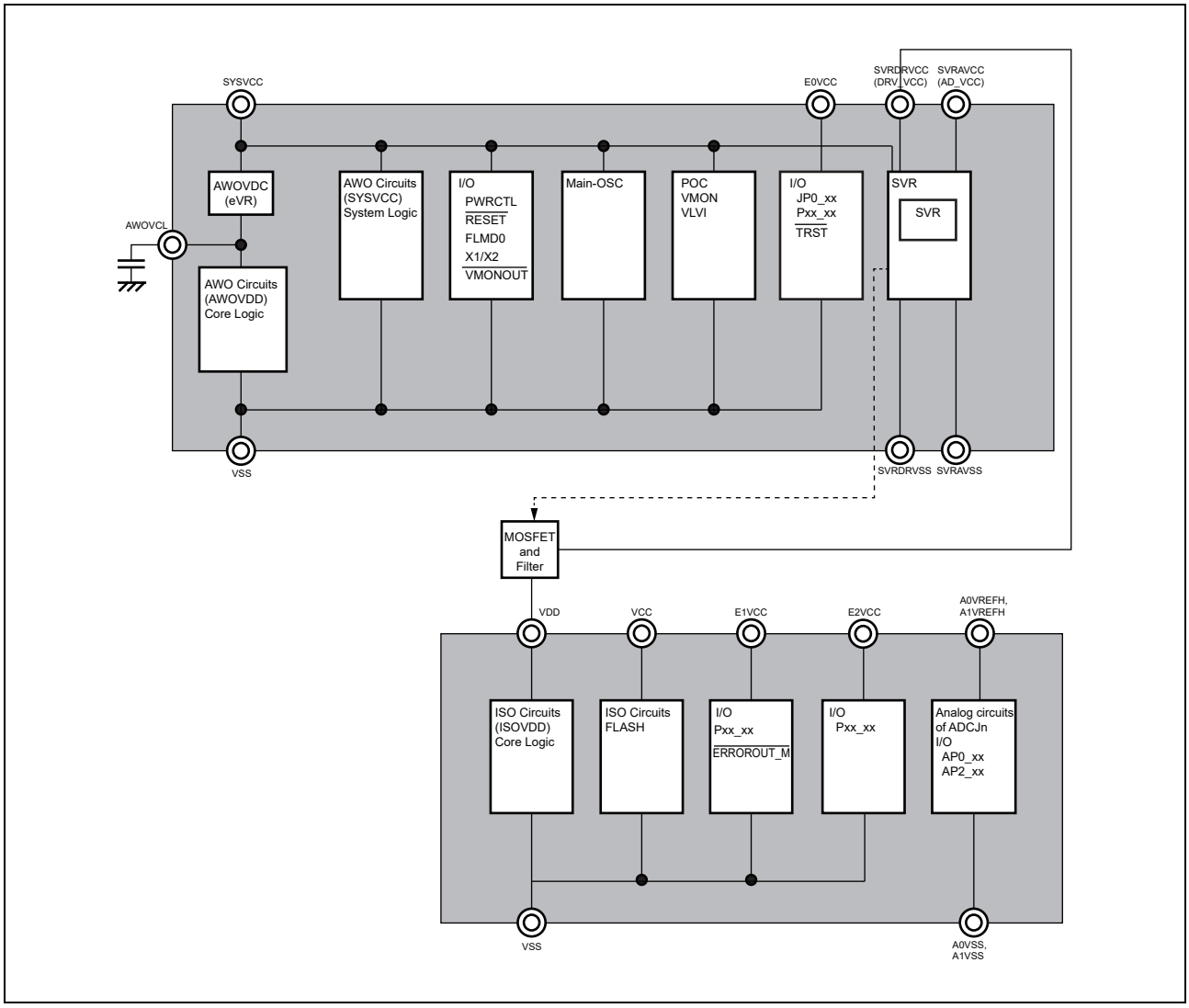


Figure 10.9 Overview of Power Supply Circuit (RH850/U2A6 BGA156)

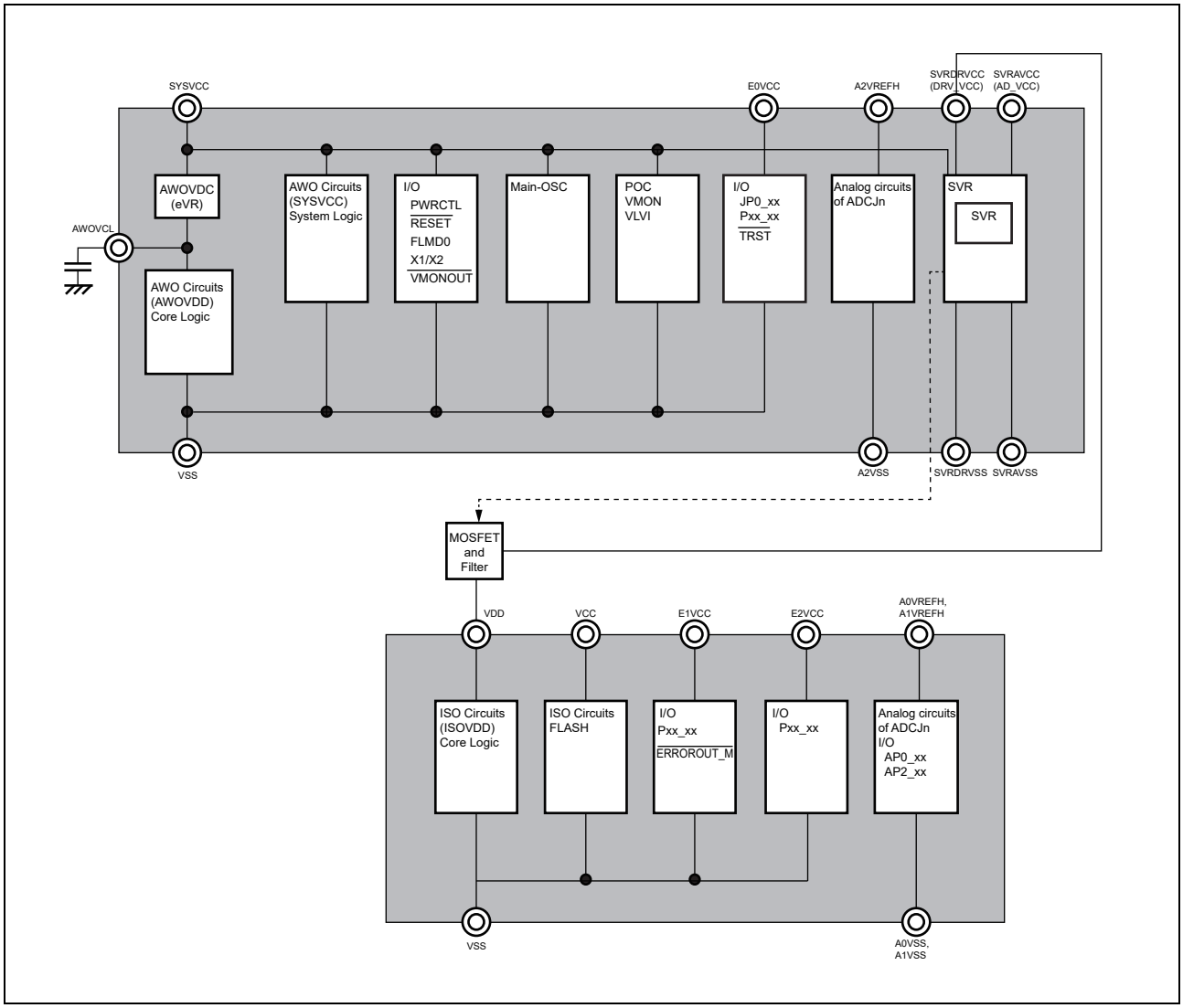


Figure 10.10 Overview of Power Supply Circuit (RH850/U2A6 QFP144)

10.4 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 10.2 Functional Modules and Power Domain

Power Domain	Functions
AWO area	• STBC, Reset controller • Retention RAM • CLMA0, CLMA1, CLMA2 • HV IntOSC, Main OSC, LS IntOSC, HS IntOSC • WDTBA, RTCA, TAUJ2, TAUJ3, ADCJ2 • SVR, LPS • POC, VMON, VLVI • AWO Port groups
ISO area	• CPU subsystem • Code Flash, Data Flash, Local RAM, Cluster RAM, Emulation RAM*1 • PLL, SGMII_PLL, HSIFPLL • CLMA3, CLMA4, CLMA5, CLMA6, CLMA7, CLMA8, CLMA9 • WDTB0, WDTB1, WDTB2, WDTB3 • RHSIF, TPBA, GTM • TAUD, TAUJ0, TAUJ1, OSTM, PWM-Diag, RS-CANFD, RLIN3, RIIC, LTSC • ADCJ0, ADCJ1, TAPA, TSG3, ENCA, SFMA, FLXA, ETNB, CXPI, RSENT, MMCA • SWDT, ICUM • PSI5, PSI5-S, SCI3 • MSPI, PIC, KCRC, OTS, ECM • ISO Port groups • DMON

Note 1. For U2A6 only

10.5 VDD Power Supply

10.5.1 Features

There are two ways to supply VDD power.

Power supply method without SVR. Refer to **Section 10.5.2, Without SVR**.

Power supply method using SVR. Refer to **Section 10.5.3, Using SVR**.

10.5.2 Without SVR

This section describes how to supply if you do not use SVR.

Connect the external power supply to VDD.

The supply voltage is supplied with the voltage specified by VDD.

Power is supplied from the VDD terminal to the ISO domain.

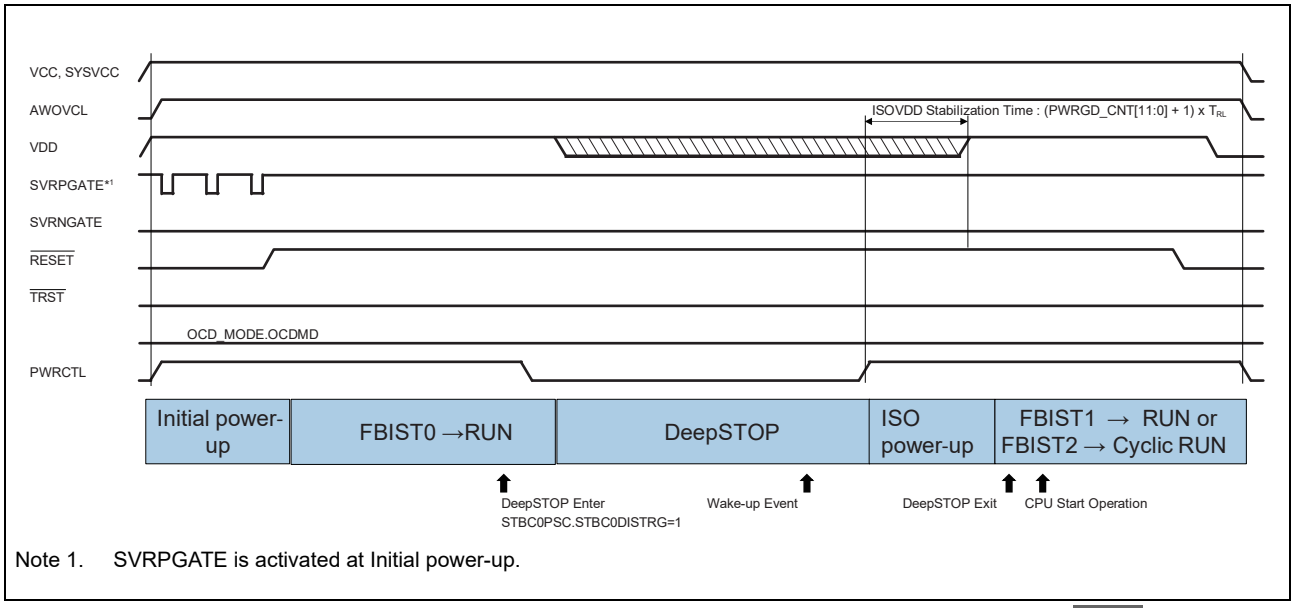
DeepSTOP mode allows the PWRCTL pin to control the VDD supply.

When the PWRCTL = L, external VDD supply to switch off.

When the PWRCTL = H, external VDD supply to switch on again.

DeepSTOP Reset will not be released until the VDD stabilizes when returning from DeepSTOP.

The waiting time for VDD stabilization can be set by PWRGD_CNT register.



Note 1. SVRPGATE is activated at Initial power-up.

Figure 10.11 External Power Supplies VDD when Non-Debug Mode ($\overline{TRST}$ = L, OCD_MODE.OCDMD = 0)

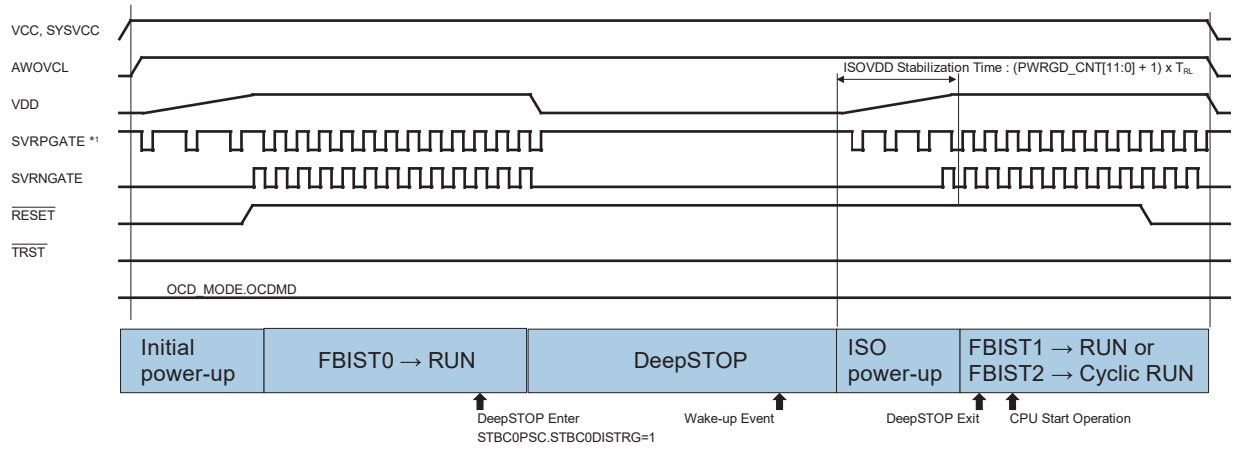
10.5.3 Using SVR

This section describes how to supply when using SVR.

SVR controls switching of external MOSFET.

Connect the output power from the external MOSFET to VDD.

Power is supplied from the VDD terminal to the ISO domain.



Note 1. SVRPGATE is activated at Initial power-up.

Figure 10.12 SVR Supplies VDD when Non-Debug Mode ($\overline{TRST}$ = L, OCD_MODE.OCDMD = 0)

For PWRGD_CNT refer to **Section 15.2.4.28, PWRGD_CNT — Power Good Counter Register**.

See the section below for more information

- **Section 11.2, Power On Clear**
- **Section 15.3.2, DeepSTOP Mode**
- **Section 9.5.1, Reset Categories**
- **Section 55, Electrical Characteristics**

10.6 SVR Controller

10.6.1 Features

SVR controller is Synchronous Rectification Step-Down Controller and can directly drive the MOSFET.

Connect output power of the MOSFET to the VDD, and power to the ISO area is supplied.

10.6.2 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA / DTS requests.

This module has no error notifications to ECM.

10.6.3 Input/Output Pins

This section describes the list of external pins for SVR converter.

Refer to the target specification of SVR for details of terminal characteristics.

10.6.3.1 External Supply pin

Table 10.3 External Supply pin

Port	I/O	Comment
SVRAVCC SYSVCC	—	3.3 V/5 V power supply for control blocks
SVRDRVCC	—	3.3 V/5 V power supply for gate driver
SVRAVSS SVRDRVSS	—	Ground for SVR

10.6.3.2 Gate driver pin

Table 10.4 Gate driver pin

Port	I/O	Comment
SVRPGATE	DO	Gate drive for the high-side PMOSFET
SVRNGATE	DO	Gate drive for the low-side NMOSFET

10.6.4 Setting Parameter

It is possible to set SVR parameters with the flash option byte.

Select combinations of Fsw, L, C, and MOSFET from data sheet and set them with the flash option byte.

A list of parameters is shown in the **Table 10.5**.

For details of flash option byte, see to **Section 51, Flash Memory**.

Table 10.5 SVR parameters list

	Control parameters of SVR	Comment
1	SVRENABLE	SVR enable setting
2	SVRAJDTP[2:0]	Adjusting dead time of High-side: OFF → Low-side: ON
3	SVRAJDTN[2:0]	Adjusting dead time of Low-side: OFF → High-side: ON
4	SVRAJPRDSR[3:0]	Adjusting slew rate (drive ability)
5	SVRADNSMP[7:0]	ADC sampling time setting
6	SVRADTHRESH[7:0]	ADC sampling start state setting
7	SVRADTHRESHE[7:0]	ADC conversion completion state setting
8	SVRFSWMODE[1:0]	Switching frequency setting
9	SVRKPVSCL[13:0]	Scaled KP value of VPID
10	SVRKIVSCL[13:0]	Scaled KI value of VPID
11	SVRKDVSCL[13:0]	Scaled KD value of VPID
12	SVRMAXDUTY[7:0]	Maximum on-duty setting
13	SVRMINSKIPDUTY[1:0]	Skip minimum pulse setting

10.6.5 Basic Function

This section describes the basic functions.

SVR is PMOS external digital control switching regulator. The functions and characteristics of SVR are as follows.

(1) SVR Enable setting

When SVR is used, set SVRENABLE of Option Byte to “1”. When SVR is not used, set SVRENABLE to “0”.

(2) Select the switching frequency

(3) Adjust pre-driver characteristics (slew rate, dead time can be adjusted)

It is possible to set SVR parameters with the flash option byte.

Select combinations of Fsw, L, C, PMOS and set them with the flash option byte.

(4) Starting mode with Soft Start and Power Supply mode with analog PFM and digital PWM

(a) Soft Start

This is the mode to raise the ISOVDD to the operation guaranteed voltage.

(b) PFM mode

This mode supplies constant ISOVDD regardless of SVR parameter.

(c) PWM mode

This is the mode in which ISOVDD is stably supplied using the parameters given to SVR.

10.7 Connection Example

10.7.1 Example of Power Supply Connection for RH850/U2A-EVA

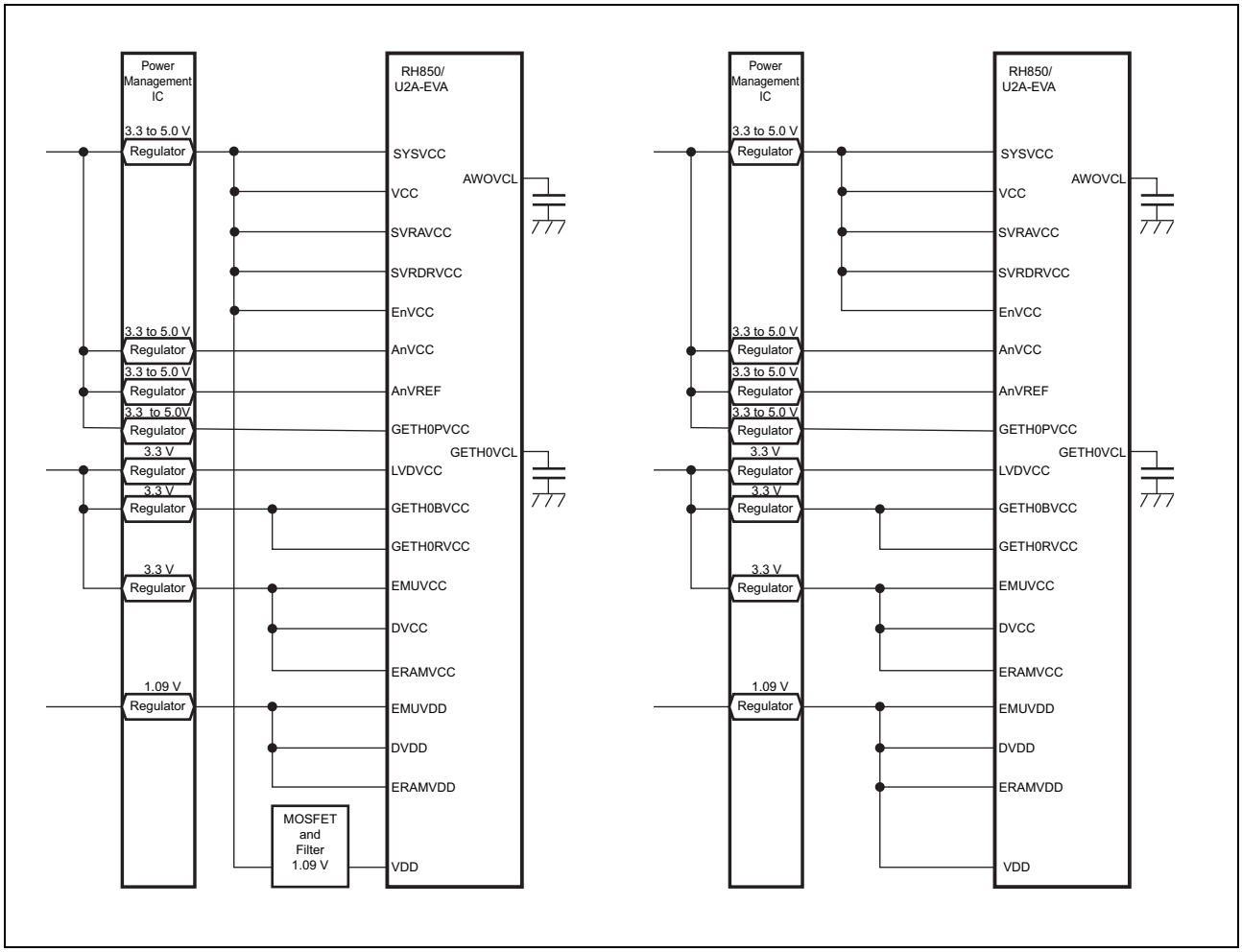


Figure 10.13 Example of Power Supply Connection for RH850/U2A-EVA

10.8 Power Up/Down Timing

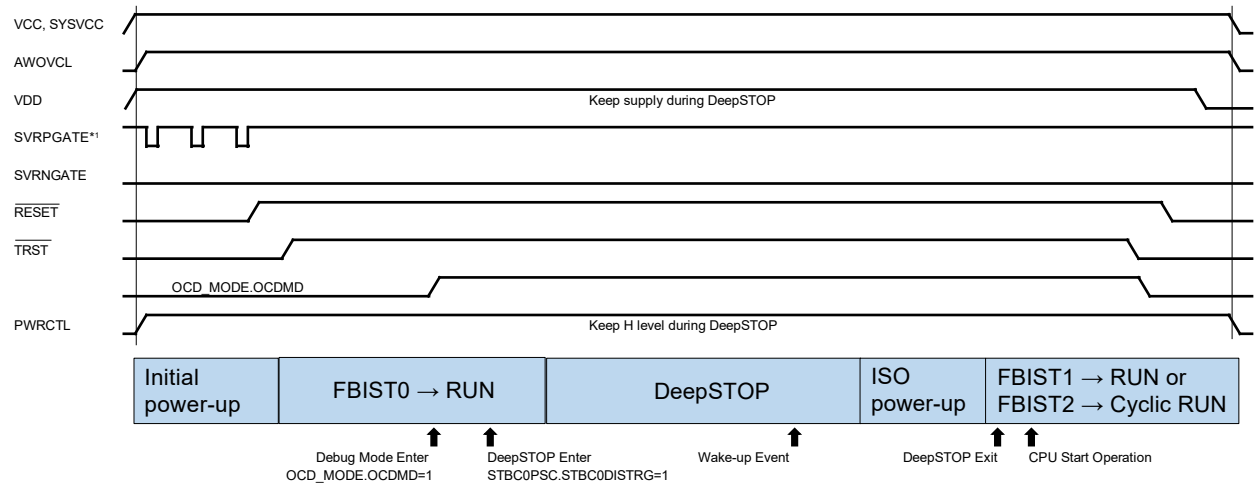
See details in **Section 55, Electrical Characteristics**.

10.9 Power Control in Debug Mode

In Debug mode, the power off control becomes invalid. Each power supply continues to be supplied even during DeepSTOP.

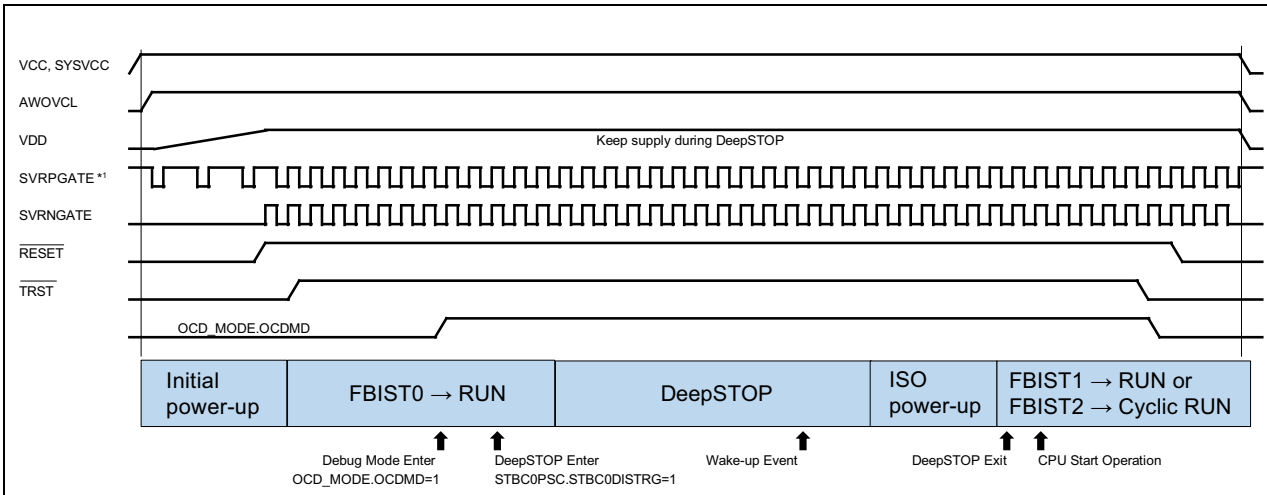
In the case of without SVR, PWRCTL signal keep to high level. For details, see **Figure 10.14, External Power Supplies VDD when Debug Mode (TRST = H, OCD_MODE.OCDMD = 1)**.

In the case of using SVR, SVR keeps operation and power supply from MOSFET continues. For details, see **Figure 10.15, SVR Supplies VDD when Debug Mode (TRST = H, OCD_MODE.OCDMD = 1)**.



Note 1. SVRPGATE is activated at Initial power-up.

Figure 10.14 External Power Supplies VDD when Debug Mode ($\overline{TRST}$ = H, OCD_MODE.OCDMD = 1)



Note 1. SVRPGATE is activated at Initial power-up.

Figure 10.15 SVR Supplies VDD when Debug Mode ($\overline{TRST}$ = H, OCD_MODE.OCDMD = 1)

10.10 SVR restriction

For products in **Table 10.6, Affected products**, as usage of SVR is restricted*1, using with external power supply is recommended.

For products in **Table 10.7, Not affected products**, either SVR (without restriction of below Note 1) or external power supply to VDD can be selected.

Note 1. In case of using SVR, when POCRES is issued, SYSVCC and SVRDRVCC should be raised up after VDD level is lower than 0.64 V.

Table 10.6 Affected products

Product Name	Note
R7F702300EBBG-C	U2A16 BGA516
R7F702300EBBB-C	U2A16 BGA373
R7F702300EABA-C	U2A16 BGA292
R7F702300AEBBC-C	U2A16 BGA516
R7F702300AEBBB-C	U2A16 BGA373
R7F702300AFABA-C	U2A16 BGA292
R7F702301EBBA-C	U2A8 BGA373
R7F702301EABG-C	U2A8 BGA292
R7F702301AEBBA-C	U2A8 BGA373
R7F702301AFABG-C	U2A8 BGA292

Table 10.7 Not affected products

Product Name	Note
R7F702Z19AEDBG	U2A-EVA BGA516
R7F702Z19BFDBG	U2A-EVA BGA516
R7F702300BEBBC-C	U2A16 BGA516
R7F702300BEBBB-C	U2A16 BGA373
R7F702300BFABA-C	U2A16 BGA292
R7F702301BEBBA-C	U2A8 BGA373
R7F702301BFABG-C	U2A8 BGA292
R7F702302FABB-C	U2A6 BGA292
R7F702302FABD-C	U2A6 BGA156
R7F702302FAFK-C	U2A6 QFP176
R7F702302FAFM-C	U2A6 QFP144

Section 11 Power Supply Voltage Monitor

11.1 Overall Configuration

- The power supply voltage monitor continuously monitors multiple external and internal supply voltages in order to ensure that the device operates with a supply voltage within the specified range.
- If the power supply voltage drops below or rise above the reference voltage or comparison voltage, an internal reset signal is generated.
- The following lists the power supply voltage monitor functions.

Table 11.1 Power Supply Voltage Monitor Functions

Function Name	Monitor Voltage	Signal Generated when voltage drops or voltage rise
Power-On Clear (POC)	SYSVCC	Internal reset signal (POCRES)
Primary detection of Voltage Monitor (VMON)	AWOVDD ISOVDD VCC E0VCC	Internal reset signal (VMON Reset), Monitor signal ($\overline{\text{VMONOUT}}$)
Delay Monitor (DMON)	ISOVDD	
Secondary detection of Voltage Monitor (SAR-ADC)	AWOVDD ISOVDD VCC E0VCC	Connect INTC2
RAM Retention Voltage Indicator (VLVI)	SYSVCC	—

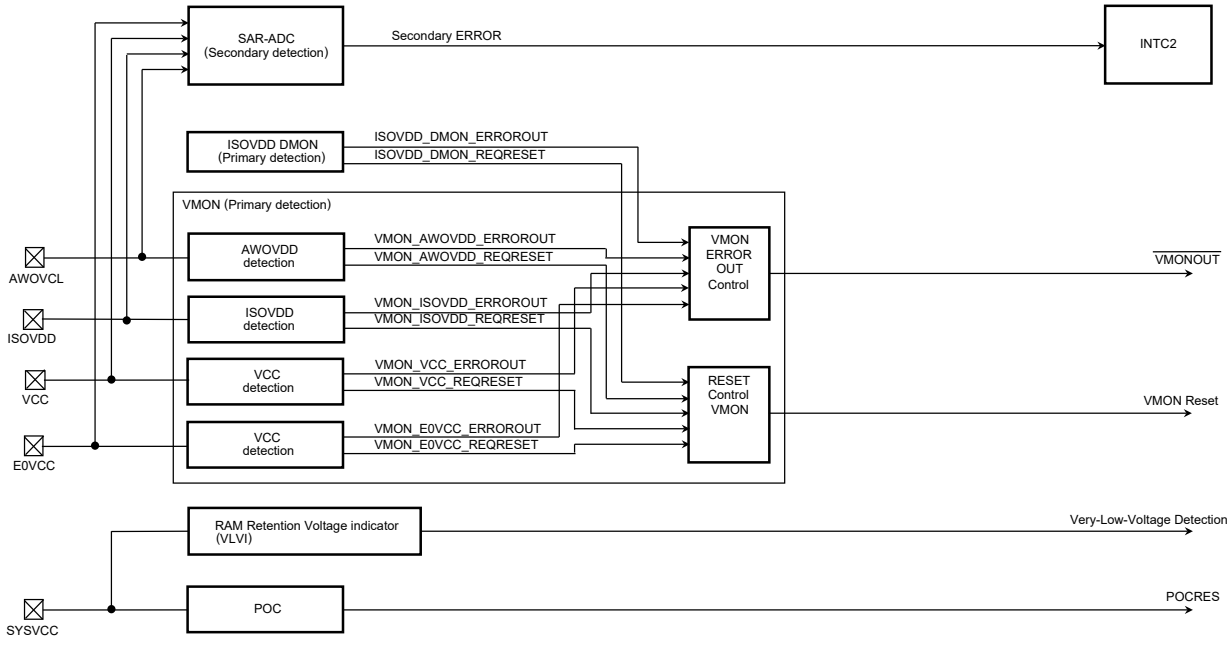
The block diagram of the power supply monitor is shown in **Figure 11.1**.

This MCU consist of POC, power supply voltage monitor (VMON, DMON and SAR-ADC) and VLVI.

Primary power supply voltage monitor can control on or off for the VMON Reset generated by high level/low level detection of ISOVDD, VCC, and E0VCC.

The detection of AWOVDD always generate a VMON Reset.

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage SYSVCC with the RAM retention voltage VVLVI.



Note: The device is kept in reset state under the control of $\overline{\text{RESET}}$, POC, or the configured voltage monitor VMON until reset status release. See **Section 55.2.13** and **Section 55.2.14** for the specification of the reset voltage levels.

Figure 11.1 Block Diagram of Power Supply Voltage Monitor (Overall)

11.2 Power On Clear

11.2.1 Features

The POC continuously monitors the external power supply voltage SYSVCC. This ensures that the microcontroller only operates at or above power-on-clear detection voltage (V_{POC}).

If SYSVCC falls below the POC detection voltage (SYSVCC < V_{POC}), Power On Reset is generated.

For details, see **Section 9.5.2, Reset Sources**.

11.2.2 Operation

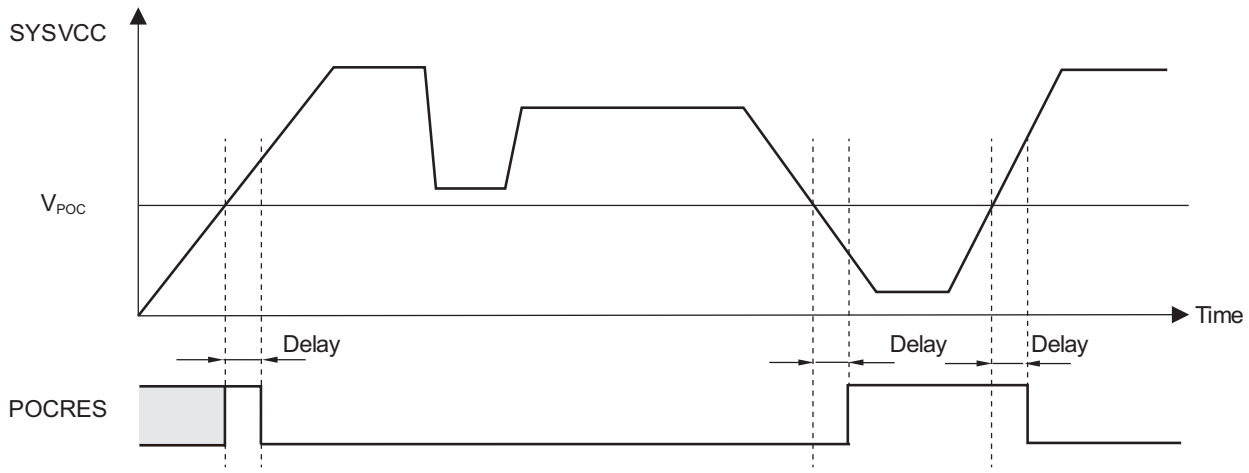
The power-on-clear circuit (POC) constantly compares the power supply voltage SYSVCC with the internal reference voltage V_{POC}. It ensures that the microcontroller only operates as long as the power supply exceeds a certain level.

If SYSVCC falls below the internal reference voltage (SYSVCC < V_{POC}), Power On Reset is generated and Power On Reset flag (PRESF0) is set.

For details, on the specification of the internal voltage reference level Power On Reset, see **Section 9.5.2, Reset Sources**.

The power-on-clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level V_{POC}.

The following figure illustrates the timing of a POCRES.



Note: Delay:
When the SYSVCC exceeds the V_{POC}, the POCRES is released after certain delay. When the SYSVCC falls below the V_{POC}, the Power On Reset flag POCRES is asserted after certain delay.
See **Section 55, Electrical Characteristics** for the delay time.

Figure 11.2 POC Reset Timing

11.3 Primary Detection of Voltage Monitor (VMON)

11.3.1 Features

- Violation occurrence of the operating voltage ranges is informed users of by:
  - Signal output pin ($\overline{\text{VMONOUT}}$).
  - Power supply voltage high and low-voltage flags.
  - VMON Reset.
- Diagnostic function:
  - VMON function is testable.
  - High and low-voltage error can be generated without influencing voltage itself.
  - VMON error test is done by changing reference voltage.
  - For diagnosis, the signal path to the $\overline{\text{VMONOUT}}$ pin can optionally be masked.

The primary detection of voltage monitor high-side (HDET) and low-side (LDET) voltage detectors, which detect if the monitored voltage is higher or lower than the specified voltage. See **Figure 11.3** and **Figure 11.4** for the detection level.

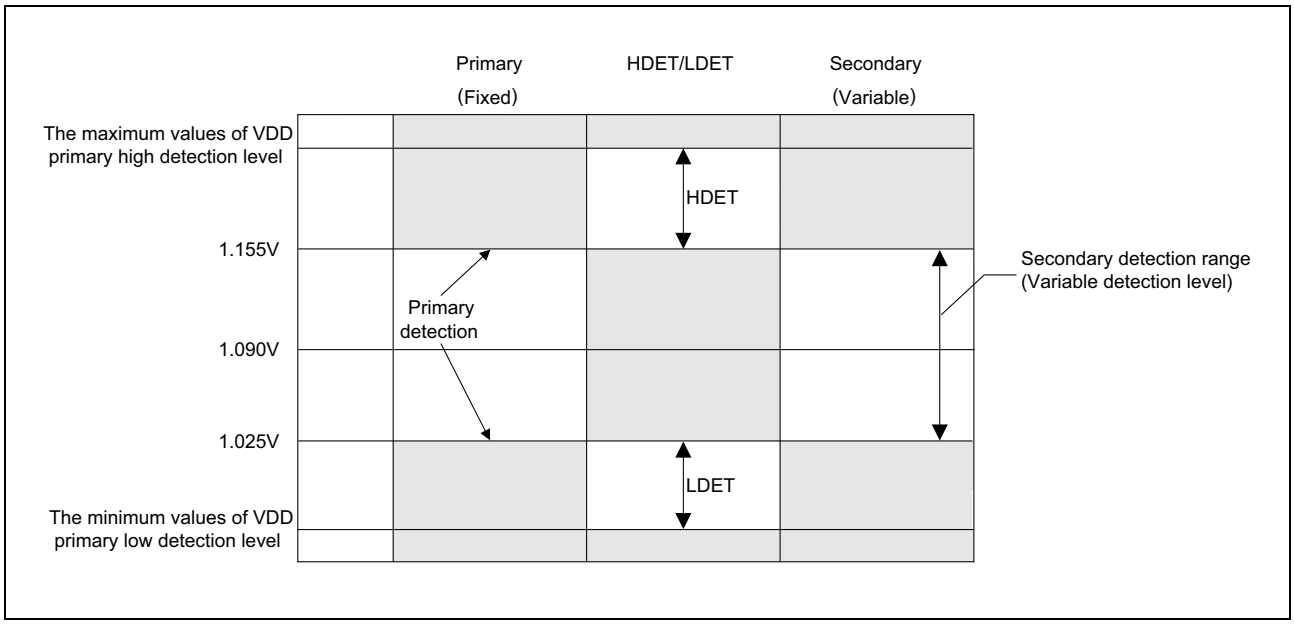


Figure 11.3 VMON Detection Level (ISOVDD/AWOVDD)

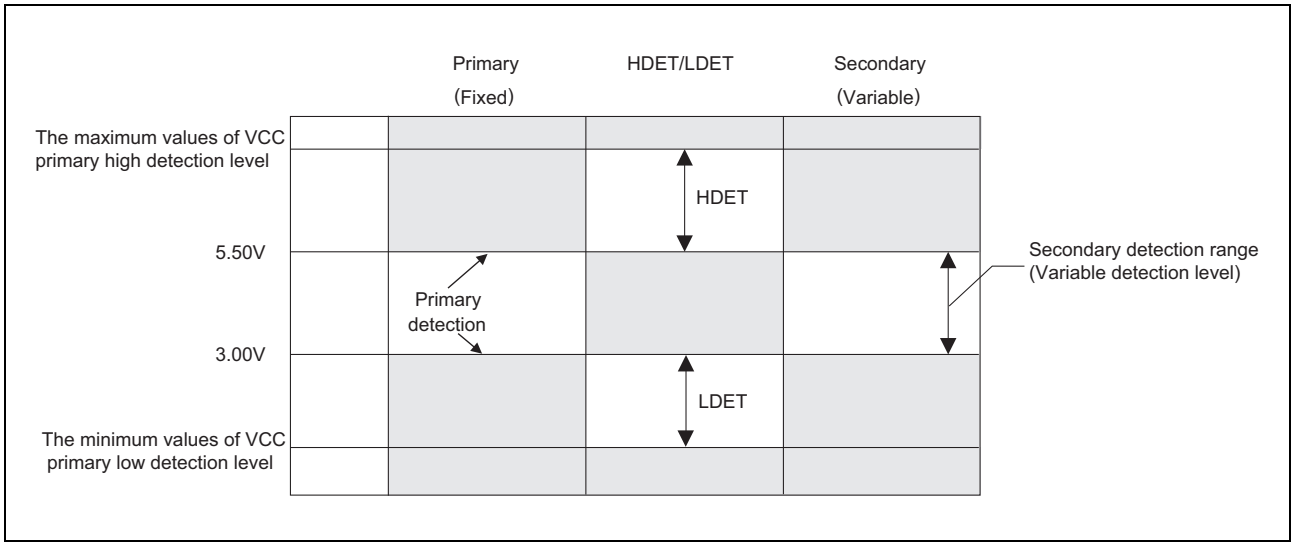


Figure 11.4 VMON Detection Level (E0VCC/VCC)

CAUTION

If system enters DeepSTOP mode, VMON function detection of ISOVDD part is masked.

See Section 11.3.6.12 of Table 11.16.

When not using VMON in DeepSTOP mode, VMON function of AWOVDD, VCC and E0VCC part are also masked and VMONOUT is kept to high.

11.3.2 Clock Supply

The clock supply to the VMON is shown in **Table 11.2**.

Table 11.2 Clock Supply

Module Name	Module Clock Name	Clock Name
VMON	Digital noise filter clock	CLK_HVIOSC
	Register access clock	CLK_LSB

11.3.3 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA / DTS requests.

This module has no error notifications to ECM.

11.3.4 External Input/Outputs

I/O pins related to reset are shown in **Table 11.3.**

Table 11.3 I/O pins

Module Name	Specification	Description
VMON	$\overline{\text{VMONOUT}}$	VMON error detection output

11.3.5 Overview

11.3.5.1 Block Diagram

Figure 11.5, **Figure 11.6**, **Figure 11.7** and **Figure 11.8** show the block diagram of the power supply voltage monitor.

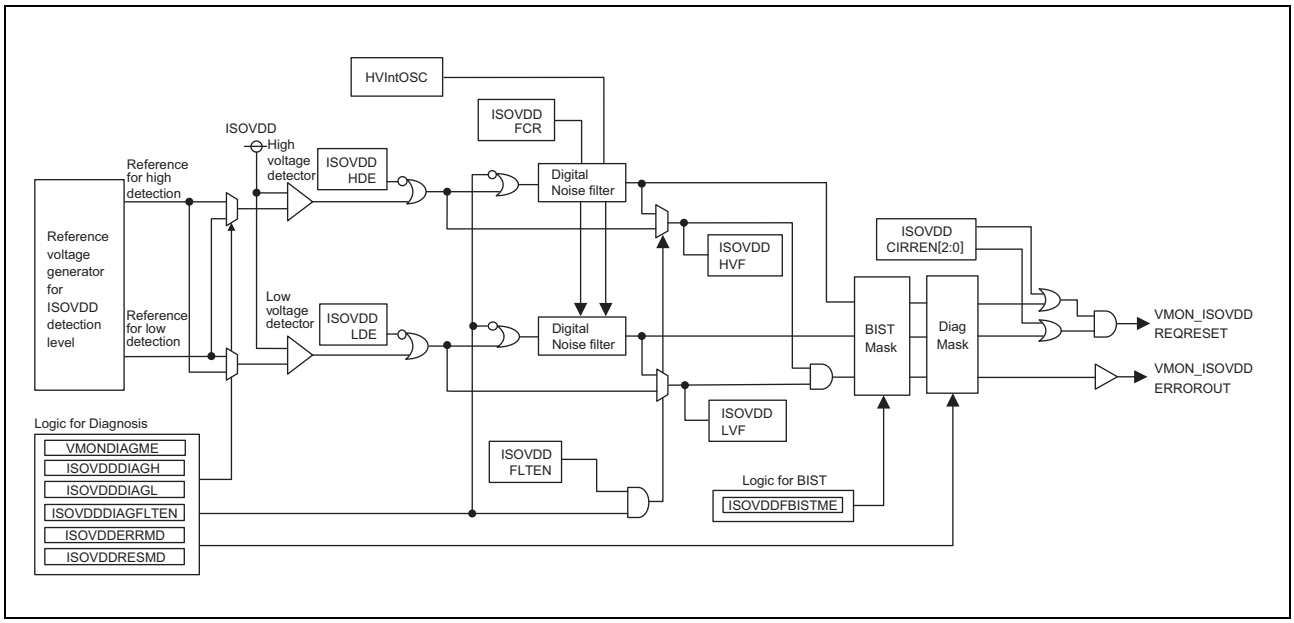


Figure 11.5 Block Diagram of the Power Supply Voltage Monitor (ISOVDD)

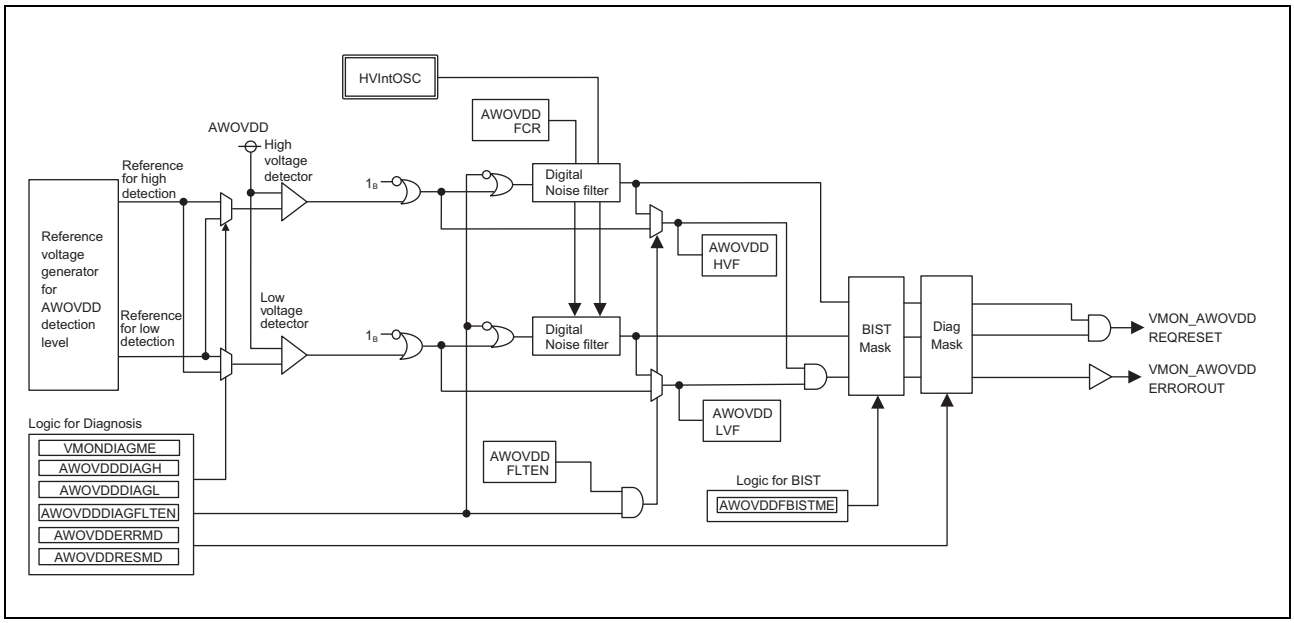


Figure 11.6 Block Diagram of the Power Supply Voltage Monitor (AWOVDD)

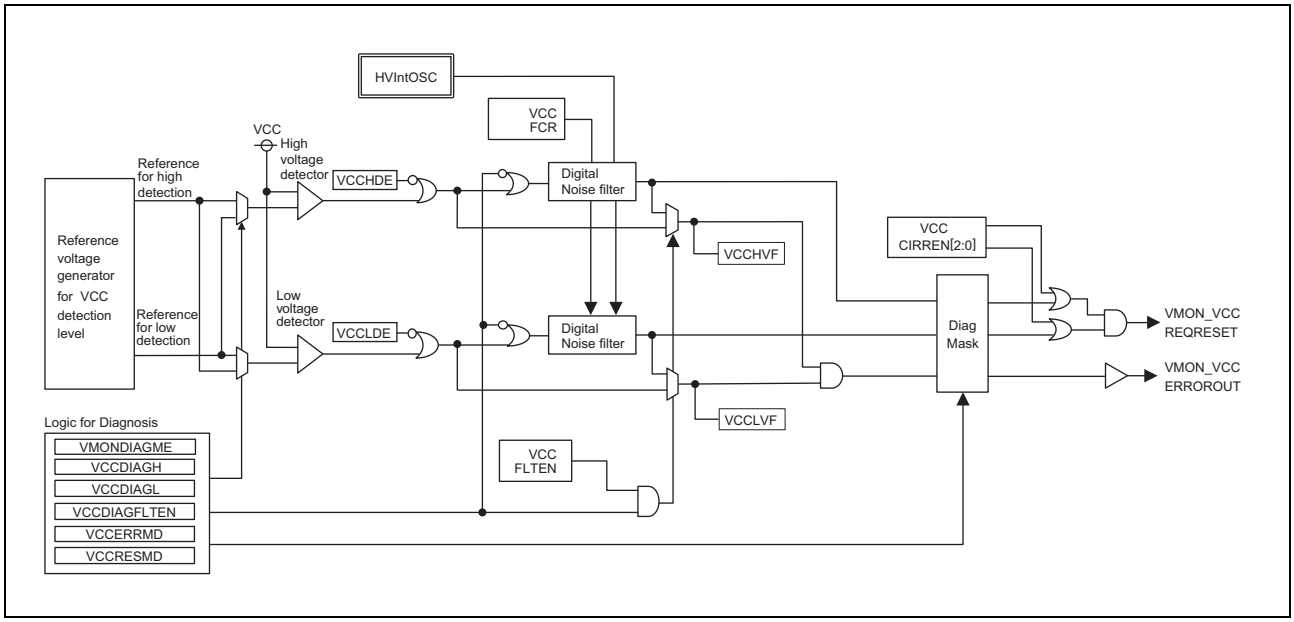


Figure 11.7 Block Diagram of the Power Supply Voltage Monitor (VCC)

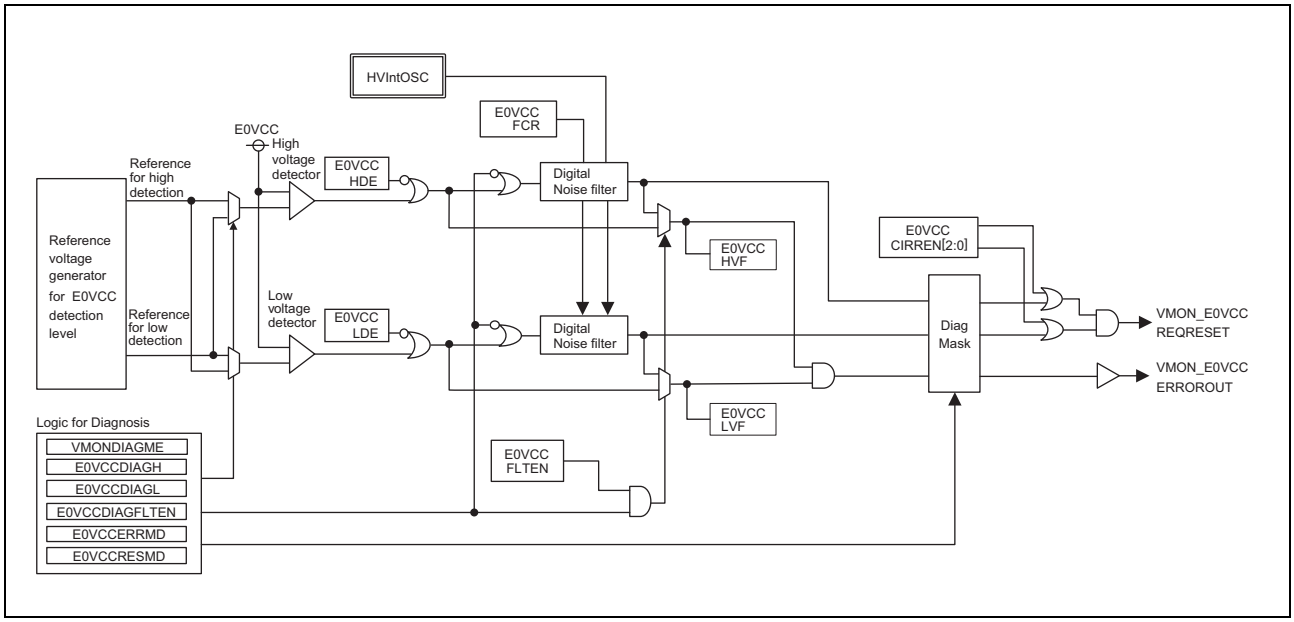


Figure 11.8 Block Diagram of the Power Supply Voltage Monitor (E0VCC)

11.3.6 Registers

11.3.6.1 List of Registers

A register list related to VMON is shown in **Table 11.4**.

Table 11.4 List of Registers

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 3000$_H$	VMONF	VMON Factor Register	8	00$_H$	PBG20 #2	
FF98 3004$_H$	VMONFC	VMONF Clear Register	8	00$_H$	PBG20 #2	VMONKCPROT
FF98 3008$_H$	VMONDIAGME	VMONDIAG Monitor Enable Register	8	00$_H$	PBG20 #2	
FF98 300C$_H$	VMONDIAGMEW	VMONDIAGME Write Register	8	00$_H$	PBG20 #2	VMONKCPROT
FF98 3010$_H$	VMONDMASK	VMON Detection Output Diagnosis MASK Register	8	00$_H$	PBG20 #2	
FF98 3014$_H$	VMONDIAG	VMON DIAG Mode Setting Register	8	00$_H$	PBG20 #2	
FF98 3018$_H$	VMONDIAGFE	VMON DIAG Filter Enable Register	8	0F$_H$	PBG20 #2	
FF98 301C$_H$	VMONOUTF	VMONOUT Factor Register	8	00$_H$	PBG20 #2	
FF98 3020$_H$	VMONOUTFC	VMONOUTF Clear Register	8	00$_H$	PBG20 #2	VMONKCPROT
FF98 3024$_H$	VMONMON	VMON Monitor Register	8	0x$_H$	PBG20 #2	
FF98 3080$_H$	ISOVDDDE	ISOVDD Detection Enable Register	8	0x$_H$	PBG20 #2	VMONKCPROT
FF98 3084$_H$	ISOVDDFCR	ISOVDD Monitor Filter Control Register	8	0x$_H$	PBG20 #2	
FF98 3088$_H$	AWOVDDDE	AWOVDD Detection Enable Register	8	0x$_H$	PBG20 #2	VMONKCPROT
FF98 308C$_H$	AWOVDDFCR	AWOVDD Monitor Filter Control Register	8	0x$_H$	PBG20 #2	
FF98 3100$_H$	VCCDE	VCC Detection Enable Register	8	0x$_H$	PBG20 #2	VMONKCPROT
FF98 3104$_H$	VCCFCR	VCC Monitor Filter Control Register	8	0x$_H$	PBG20 #2	
FF98 3180$_H$	E0VCCDE	E0VCC Detection Enable Register	8	0x$_H$	PBG20 #2	VMONKCPROT
FF98 3184$_H$	E0VCCFCR	E0VCC Monitor Filter Control Register	8	0x$_H$	PBG20 #2	
FF98 31C0$_H$	VMONDSCR	VMON DeepSTOP Detections and Filters Control Register	8	0x$_H$	PBG20 #2	VMONKCPROT
FF98 3F00$_H$	VMONKCPROT	VMON Register Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	

CAUTION

If system enters DeepSTOP mode, VMON function of ISOVDD area is masked.

Register reset conditions are shown in **Table 11.5**

Table 11.5 Register Reset Conditions

Register Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
VMONF	√	—	—	—	—	—	—
VMONFC	√	√	—	—	—	—	—
VMONDIAGME	√	√*5	—	—	√*2	—	—
VMONDIAGMEW	√	√*5	—	—	√*2	—	—
VMONDMASK	√	√	—	—	√*2	—	—
VMONDIAG	√	√	—	—	√*2	—	—
VMONDIAGFE	√	√	—	—	√*2	—	—
VMONOUTF	√	—	—	—	—	—	—
VMONOUTFC	√	√	—	—	—	—	—
VMONMON	√*4	—	—	—	—	—	—
ISOVDDDE	√	√*1	—	—	—	—	—
ISOVDDFCR	√	√*1	—	—	—	—	—
AWOVDDDE	√	√*1	—	—	—	—	—
AWOVDDFCR	√	√*1	—	—	—	—	—
VCCDE	√	√*1	—	—	—	—	—
VCCFCR	√	√*1	—	—	—	—	—
E0VCCDE	√	√*1	—	—	—	—	—
E0VCCFCR	√	√*1	—	—	—	—	—
VMONDSCR	√	√*1	—	—	—	—	—
VMONKCPROT	√	√	√	—	√	—	—

Note 1. VMON Reset is excluded.

Note 2. If VMON Reset which excepts AWOVDD detection during DeepSTOP transition, VMONDIAGME and VMONDIAGMEW may not be initialized even if DeepSTOP Reset occurs.
If System Reset 2 occurs during DeepSTOP transition, VMONDIAGME, VMONDIAGMEW, VMONDMASK, VMONDIAG, and VMONDIAGFE may not be initialized even if DeepSTOP Reset occurs.

Note 3. External Reset only.

Note 4. POC Reset only.

Note 5. Only External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection.

11.3.6.2 VMONF — VMON Factor Register

VMONF register indicates that an error is detected by VMON. Each flag of VMONF can be cleared individually by writing '1' to the corresponding bit in VMONFC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	E0VCCHVF	E0VCCLVF	VCCHVF	VCCLVF	ISOVDDHVF	ISOVDDLVF	AWOVDDHVF	AWOVDDLVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 11.6 VMONF Register Contents

Bit Position	Bit Name	Function
7	E0VCCHVF	E0VCC high voltage detection flag 0: No high E0VCC voltage violation detected 1: High E0VCC voltage violation occurred
6	E0VCCLVF	E0VCC low voltage detection flag 0: No low E0VCC voltage violation detected 1: Low E0VCC voltage violation occurred
5	VCCHVF	VCC high voltage detection flag 0: No high VCC voltage violation detected 1: High VCC voltage violation occurred
4	VCCLVF	VCC low voltage detection flag 0: No low VCC voltage violation detected 1: Low VCC voltage violation occurred
3	ISOVDDHVF	ISOVDD high voltage detection flag 0: No high ISOVDD voltage violation detected 1: High ISOVDD voltage violation occurred
2	ISOVDDLVF	ISOVDD low voltage detection flag 0: No low ISOVDD voltage violation detected 1: Low ISOVDD voltage violation occurred
1	AWOVDDHVF	AWOVDD high voltage detection flag 0: No high AWOVDD voltage violation detected 1: High AWOVDD voltage violation occurred
0	AWOVDDLVF	AWOVDD low voltage detection flag 0: No low AWOVDD voltage violation detected 1: Low AWOVDD voltage violation occurred

11.3.6.3 VMONFC — VMONF Clear Register

VMONFC is a register to clear the VMONF register. The read value of this register is always 00_H.

Access: This register is a write-only register that can be written in 8-bit units.

Address: FF98 3004_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	E0VCCHVFC	E0VCCLVFC	VCCHVFC	VCCLVFC	ISOVDDHVFC	ISOVDDLVFC	AWOVDDHVFC	AWOVDDLVFC
Value after reset	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Table 11.7 VMONFC Register Contents

Bit Position	Bit Name	Function
7	E0VCCHVFC	Clear E0VCC high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear E0VCCHVF
6	E0VCCLVFC	Clear E0VCC low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear E0VCCLVF
5	VCCHVFC	Clear VCC high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear VCCHVF
4	VCCLVFC	Clear VCC low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear VCCLVF
3	ISOVDDHVFC	Clear ISOVDD high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear ISOVDDHVF
2	ISOVDDLVFC	Clear ISOVDD low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear ISOVDDLVF
1	AWOVDDHVFC	Clear AWOVDD high voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear AWOVDDHVF
0	AWOVDDLVFC	Clear AWOVDD low voltage detection flag 0: Writing 0 has no effect 1: Writing 1 will clear AWOVDDLVF

11.3.6.4 VMONDIAGME — VMON DIAG Monitor Enable Register

VMONDIAGME is a read only register to control VMON DIAG function.

This register reflects the settings written to the VMONDIAGMEW.

If VMON Reset which excepts AWOVDD detection or System Reset 2 occur during DeepSTOP transition, this register may not be initialized even if DeepSTOP Reset occurs.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3008$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONDIAGME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 11.8 VMONDIAGME Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	VMONDIAGME	Permit DIAG function 0: DIAG function of the Voltage Monitor can be enabled. 1: DIAG function of the Voltage Monitor cannot be enabled.

11.3.6.5 VMONDIAGMEW — VMONDIAGME Write Register

VMONDIAGMEW is a register to set values of VMONDIAGME. Writing is permitted only once after Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset were released. Subsequent write operation is ignored. The read value of this register is always 00_H.

If VMON Reset which excepts AWOVDD detection or System Reset 2 occur during DeepSTOP transition, this register may not be initialized even if DeepSTOP Reset occurs.

Access: This register is a write-only register that can be written in 8-bit units.

Address: FF98 $300C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONDIAGMEW
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 11.9 VMONDIAGMEW Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	VMONDIAGMEW	The data written in this bit is set to VMONDIAGME.

11.3.6.6 VMONDMASK — VMON Detection Output Diagnosis Mask Register

VMONDMASK is a register to mask $\overline{\text{VMONOUT}}$ and VMON Reset when VMONDIAGME = 0.

VMONDMASK setting is ignored if VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDMASK is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

If System Reset 2 occur during DeepSTOP transition, this register may not be initialized even if DeepSTOP Reset occurs.

Access: This register can be read or written in 8-bit units

Address: FF98 3010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	E0VCCRESMD	VCCRESMD	ISOVDDRESMD	AWOVDDRESMD	E0VCCERRMD	VCCERRMD	ISOVDDERRMD	AWOVDDERRMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.10 VMONDMASK Register Contents

Bit Position	Bit Name	Function
7	E0VCCRESMD	This bit masks VMON_E0VCC_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_E0VCC_REQ_RESET is not masked. 1: VMON_E0VCC_REQ_RESET is masked.
6	VCCRESMD	This bit masks VMON_VCC_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VCC_REQ_RESET is not masked. 1: VMON_VCC_REQ_RESET is masked.
5	ISOVDDRESMD	This bit masks VMON_ISOVDD_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_ISOVDD_REQ_RESET is not masked. 1: VMON_ISOVDD_REQ_RESET is masked.
4	AWOVDDRESMD*1	This bit masks VMON_AWOVDD_REQ_RESET when VMONDIAGME.VMONDIAGME = 0. 0: VMON_AWOVDD_REQ_RESET is not masked. 1: VMON_AWOVDD_REQ_RESET is masked.
3	E0VCCERRMD	This bit masks VMON_E0VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_E0VCC_ERROROUT is not masked. 1: VMON_E0VCC_ERROROUT is masked.
2	VCCERRMD	This bit masks VMON_VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_VCC_ERROROUT is not masked. 1: VMON_VCC_ERROROUT is masked.
1	ISOVDDERRMD	This bit masks VMON_ISOVDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_ISOVDD_ERROROUT is not masked. 1: VMON_ISOVDD_ERROROUT is masked.
0	AWOVDDERRMD	This bit masks VMON_AWOVDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: VMON_AWOVDD_ERROROUT is not masked. 1: VMON_AWOVDD_ERROROUT is masked.

Note 1. Set AWO Reset to 1 during diagnosis.

CAUTION

In the case of Debugger Reset, External Reset, Debugger Disconnection Reset, or DeepSTOP Reset is inserted during VMON diagnosis, VMON Reset might occur or VMONOUT might become to L even if each bit (*RESMD and *ERRMD) of VMONDMASK register is set to 1.

11.3.6.7 VMONDIAG — VMONDIAG Mode Setting Register

VMONDIAG forces VMON comparators to output error. This register is valid only when VMONDIAGME = 0.

VMONDIAG's setting is ignored if VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDIAG is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

If System Reset 2 occur during DeepSTOP transition, this register may not be initialized even if DeepSTOP Reset occurs.

Access: This register can be read or written in 8-bit units

Address: FF98 3014$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	E0VCCDIAGH	E0VCCDIAGL	VCCDIAGH	VCCDIAGL	ISOVDDDIAGH	ISOVDDDIAGL	AWOVDDDIAGH	AWOVDDDIAGL
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.11 VMONDIAG Register Contents (1/2)

Bit Position	Bit Name	Function
7	E0VCCDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: E0VCC for high voltage monitors normally. 1: E0VCC for high voltage will detect violation condition because detection level changes.
6	E0VCCDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: E0VCC for low voltage monitors normally. 1: E0VCC for low voltage will detect violation condition because detection level changes.
5	VCCDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VCC for high voltage monitors normally. 1: VCC for high voltage will detect violation condition because detection level changes.
4	VCCDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: VCC for low voltage monitors normally. 1: VCC for low voltage will detect violation condition because detection level changes.
3	ISOVDDDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: ISOVDD for high voltage monitors normally. 1: ISOVDD for high voltage will detect violation condition because detection level changes.
2	ISOVDDDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: ISOVDD for low voltage monitors normally. 1: ISOVDD for low voltage will detect violation condition because detection level changes.

Table 11.11 VMONDIAG Register Contents (2/2)

Bit Position	Bit Name	Function
1	AWOVDDDIAGH	This bit can change the detection level of high voltage detector to low level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: AWOVDD for high voltage monitors normally. 1: AWOVDD for high voltage will detect violation condition because detection level changes.
0	AWOVDDDIAGL	This bit can change the detection level of low voltage detector to high level intentionally when VMONDIAGME.VMONDIAGME = 0. 0: AWOVDD for low voltage monitors normally. 1: AWOVDD for low voltage will detect violation condition because detection level changes.

11.3.6.8 VMONDIAGFE — VMON DIAG Filter Enable Register

VMONDIAGFE specifies whether output filter for $\overline{\text{VMONOUT}}$ is enabled.

This register is valid only when VMONDIAGME = 0. The setting of this register is ignored if

VMONDIAGME.VMONDIAGME = 1.

Writing to VMONDIAGFE is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

If System Reset 2 occur during DeepSTOP transition, this register may not be initialized even if DeepSTOP Reset occurs.

When each bit is set to 0, VMON Reset assertion which is controlled by the corresponding bit does not occur.

Access: This register can be read or written in 8-bit units

Address: FF98 3018$_H$

Value after reset: 0F$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	E0VCCDIAGFLTEN	VCCDIAGFLTEN	ISOVDDDIAGFLTEN	AWOVDDDIAGFLTEN
Value after reset	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 11.12 VMONDIAGFE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	E0VCCDIAGFLTEN	Enable output filter for VMON_E0VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_E0VCC_ERROROUT 1: Enable output filter for VMON_E0VCC_ERROROUT
2	VCCDIAGFLTEN	Enable output filter for VMON_VCC_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_VCC_ERROROUT 1: Enable output filter for VMON_VCC_ERROROUT
1	ISOVDDDIAGFLTEN	Enable output filter for VMON_ISOVDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_ISOVDD_ERROROUT 1: Enable output filter for VMON_ISOVDD_ERROROUT
0	AWOVDDDIAGFLTEN	Enable output filter for VMON_AWOVDD_ERROROUT when VMONDIAGME.VMONDIAGME = 0. 0: Disable output filter for VMON_AWOVDD_ERROROUT 1: Enable output filter for VMON_AWOVDD_ERROROUT

11.3.6.9 VMONOUTF — VMONOUT Factor Register

VMONOUTF register indicates that an error is detected by VMON.

VMONOUTF can be cleared individually by writing “1” to the corresponding bit in VMONOUTFC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 301C$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONOFLG
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 11.13 VMONOUTF Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	VMONOFLG	$\overline{\text{VMONOUT}}$ flag When $\overline{\text{VMONOUT}}$ pin has changed to low level, this flag is set to 1. 0: $\overline{\text{VMONOUT}}$ doesn’t have changed to low level. 1: $\overline{\text{VMONOUT}}$ has changed to low level.

11.3.6.10 VMONOUTFC — VMONOUTF Clear Register

VMONOUTFC is a register to clear the VMONOUTF register.

The read value of the register is always 00_H.

Access: This register is a write-only register that can be written in 8-bit units

Address: FF98 3020_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONOFLGC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 11.14 VMONOUTFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	VMONOFLGC	Clear $\overline{\text{VMONOUT}}$ flag 0: Writing 0 has no effect. 1: Writing 1 will clear VMONOFLG.

11.3.6.11 VMONMON — VMON Monitor Register

VMONMON register indicates the value of $\overline{\text{VMONOUT}}$ pin.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3024_H

Value after reset: $0x_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VMONOUT
Value after reset	0	0	0	0	0	0	0	x
R/W	R	R	R	R	R	R	R	R

Table 11.15 VMONMON Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	VMONOUT	$\overline{\text{VMONOUT}}$ pin level 0: $\overline{\text{VMONOUT}}$ Output level is at low level. 1: $\overline{\text{VMONOUT}}$ Output level is at high level.

11.3.6.12 ISOVDDDE — ISOVDD Detection Enable Register

ISOVDDDE is register to control VMON Reset, mask of VMON signal and filter of VMON signal.

Bit 3, 1, 0 are set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Writing to ISOVDDDE is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

Access: This register can be read or written in 8-bit units

Address: FF98 3080_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	ISOVDDCIRREN[2:0]			ISOVDDFBISTME	ISOVDDFLTEN	—	ISOVDDHDE	ISOVDDLDE
Value after reset	0	0	0	0	x	0	x	x
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 11.16 ISOVDDDE Register Contents

Bit Position	Bit Name	Function
7 to 5	ISOVDDCIRREN[2:0]	Permit VMON Reset by ISOVDD detection. ISOVDDCIRREN 2 to 0 101_B: VMON Reset by ISOVDD high voltage detection is permitted 110_B: VMON Reset by ISOVDD low voltage detection is permitted 111_B: VMON Reset by ISOVDD high/low voltage detection is permitted Other value: VMON Reset by ISOVDD detection is not permitted
4	ISOVDDFBISTME	When this bit is set to 0, VMON_ISOVDD_ERROROUT and VMON_ISOVDD_REQRESET is masked when field-BIST is carried out after VMON Reset. When this bit is set to 1, VMON_ISOVDD_ERROROUT and VMON_ISOVDD_REQRESET is not masked when field-BIST is carried out after VMON Reset. 0: During FBIST the VMON_ISOVDD_ERROROUT and VMON_ISOVDD_REQRESET can be masked 1: During FBIST the VMON_ISOVDD_ERROROUT and VMON_ISOVDD_REQRESET cannot be masked **NOTE:** VMON_ISOVDD_ERROROUT output and VMON_ISOVDD_REQRESET are always masked when field-BIST is carried out after Power On Reset, External Reset, or Debugger Disconnection Reset. VMON_ISOVDD_ERROROUT output and VMON_ISOVDD_REQRESET are always not masked when field-BIST is carried out after System Reset 2 or after DeepSTOP.
3	ISOVDDFLTEN	Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF. 0: Disable output filter for $\overline{\text{VMONOUT}}$ and VMONF. 1: Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ISOVDDHDE	ISOVDD High voltage detection enable 0: Disable ISOVDD high voltage detection 1: Enable ISOVDD high voltage detection
0	ISOVDDLDE	ISOVDD Low voltage detection enable 0: Disable ISOVDD low voltage detection 1: Enable ISOVDD low voltage detection

CAUTION

If system enters DeepSTOP mode, VMON function of ISOVDD part is masked.

11.3.6.13 ISOVDDFCR — ISOVDD Monitor Filter Control Register

ISOVDDFCR is a register to adjust the filtering width of ISOVDD Monitor digital noise filter.

This register is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3084_H

Value after reset: $0x_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	ISOVDDFLTW[1:0]		ISOVDDCLKSEL[1:0]	
Value after reset	0	0	0	0	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Table 11.17 ISOVDDFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	ISOVDDFLTW [1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by cycle-count of a clock selected by ISOVDDCLKSEL1, 0.) The following values are the Minimum Filtering Width (Cycle) 00_B: 20 cycle 01_B: 13 cycle 10_B: 8 cycle 11_B: 3 cycle
1, 0	ISOVDDCLKSEL[1:0]	These bits select a clock of the digital noise filter from the following clocks. The following values are the Clock of the Digital Filter 00_B: CLK_HVIOSC frequency/256 01_B: CLK_HVIOSC frequency/128 10_B: CLK_HVIOSC frequency/64 11_B: CLK_HVIOSC frequency/32 **CAUTION** A frequency of CLK_HVIOSC is typ. 16 MHz.

CAUTION

When using SVR Controller with Switching frequency = 868 kHz and using DeepSTOP Mode, set the filtering width to 50 µs or higher.

11.3.6.14 AWOVDDDE — AWOVDD Detection Enable Register

AWOVDDDE is register to control VMON Reset, mask of VMON signal and filter of VMON signal.

Bit 3 is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Writing to AWOVDDDE is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

Access: This register can be read or written in 8-bit units

Address: FF98 3088_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	AWOVDDFBISTME	AWOVDDFLTEN	—	—	—
Value after reset	0	0	0	0	x	0	0	0
R/W	R	R	R	R/W	R	R	R	R

Table 11.18 AWOVDDDE Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	AWOVDDFBISTME	When this bit is set to 0, VMON_AWOVDD_ERROROUT and VMON_AWOVDD_REQRESET is masked when field-BIST is carried out after VMON Reset. When this bit is set to 1, VMON_AWOVDD_ERROROUT and VMON_AWOVDD_REQRESET is not masked when field-BIST is carried out after VMON Reset. 0: During FBIST the VMON_AWOVDD_ERROROUT and VMON_AWOVDD_REQRESET can be masked 1: During FBIST the VMON_AWOVDD_ERROROUT and VMON_AWOVDD_REQRESET cannot be masked **NOTE:** VMON_AWOVDD_ERROROUT output and VMON_AWOVDD_REQRESET are always masked when field-BIST is carried out after Power On Reset, External Reset, or Debugger Disconnection Reset. VMON_AWOVDD_ERROROUT output and VMON_AWOVDD_REQRESET are always not masked when field-BIST is carried out after System Reset 2 or after DeepSTOP.
3	AWOVDDFLTEN	Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF. 0: Disable output filter for $\overline{\text{VMONOUT}}$ and VMONF. 1: Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
2 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

11.3.6.15 AWOVDDFCR — AWOVDD Monitor Filter Control Register

AWOVDDFCR is a register to adjust the filtering width of AWOVDD Monitor digital noise filter.

This register is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 308C$_H$

Value after reset: 0x$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AWOVDDFLTW[1:0]		AWOVDDCLKSEL[1:0]	
Value after reset	0	0	0	0	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Table 11.19 AWOVDDFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	AWOVDDFLTW [1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by cycle-count of a clock selected by AWOVDDCLKSEL1, 0.) The following values are the Minimum Filtering Width (Cycle) 00_B: 20 cycle 01_B: 13 cycle 10_B: 8 cycle 11_B: 3 cycle
1, 0	AWOVDDCLKSEL[1:0]	These bits select a clock of the digital noise filter from the following clocks. The following values are the Clock of the Digital Filter 00_B: CLK_HVIOSC frequency/256 01_B: CLK_HVIOSC frequency/128 10_B: CLK_HVIOSC frequency/64 11_B: CLK_HVIOSC frequency/32 **CAUTION** A frequency of CLK_HVIOSC is typ. 16 MHz.

11.3.6.16 VCCDE — VCC Detection Enable Register

VCCDE is register to control VMON Reset, mask of VMON signal and filter of VMON signal.

Bit 3, 1, 0 are set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Writing to VCCDE is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

Access: This register can be read or written in 8-bit units

Address: FF98 3100_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	VCCCIRREN[2:0]			—	VCCFLTEN	—	VCCHDE	VCCLDE
Value after reset	0	0	0	0	x	0	x	x
R/W	R/W	R/W	R/W	R	R	R	R	R

Table 11.20 VCCDE Register Contents

Bit Position	Bit Name	Function
7 to 5	VCCCIRREN [2:0]	Permit VMON Reset by VCC voltage detection. VCCCIRREN 2 to 0 101_B: VMON Reset by VCC high voltage detection is permitted 110_B: VMON Reset by VCC low voltage detection is permitted 111_B: VMON Reset by VCC high/low voltage detection is permitted Other value: VMON Reset by VCC detection is not permitted
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	VCCFLTEN	Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF 0 : Disable output filter for $\overline{\text{VMONOUT}}$ and VMONF 1 : Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	VCCHDE	VCC high voltage detection enable 0: Disable VCC high voltage detection 1: Enable VCC high voltage detection
0	VCCLDE	VCC Low voltage detection enable 0: Disable VCC low voltage detection 1: Enable VCC low voltage detection

11.3.6.17 VCCFCR — VCC Monitor Filter Control Register

VCCFCR is a register to adjust the filtering width of VCC Monitor digital noise filter.

This register is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3104$_H$

Value after reset: 0x$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	VCCFLTW[1:0]		VCCCLKSEL[1:0]	
Value after reset	0	0	0	0	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Table 11.21 VCCFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	VCCFLTW [1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by cycle-count of a clock selected by VCCCLKSEL1, 0.) The following values are the Minimum Filtering Width (Cycle) 00_B: 20 cycle 01_B: 13 cycle 10_B: 8 cycle 11_B: 3 cycle
1, 0	VCCCLKSEL [1:0]	These bits select a clock of the digital noise filter from the following clocks. The following values are the Clock of the Digital Filter 00_B: CLK_HVIOSC frequency/256 01_B: CLK_HVIOSC frequency/128 10_B: CLK_HVIOSC frequency/64 11_B: CLK_HVIOSC frequency/32 **CAUTION** A frequency of CLK_HVIOSC is typ. 16 MHz.

11.3.6.18 E0VCCDE — E0VCC Detection Enable Register

VCCDE is register to control VMON Reset, mask of VMON signal and filter of VMON signal.

Bit 3, 1, 0 are set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Writing to E0VCCDE is permitted during a period which starts from release of Power On Reset, External Reset, Debugger Disconnection Reset, VMON Reset which is AWOVDD detection, and DeepSTOP Reset to end of writing in VMONDIAGMEW.

Access: This register can be read or written in 8-bit units

Address: FF98 3180_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	E0VCCCIRREN[2:0]			—	E0VCCFLTEN	—	E0VCCHDE	E0VCCLDE
Value after reset	0	0	0	0	x	0	x	x
R/W	R/W	R/W	R/W	R	R	R	R	R

Table 11.22 E0VCCDE Register Contents

Bit Position	Bit Name	Function
7 to 5	E0VCCCIRREN [2:0]	Permit VMON Reset by E0VCC voltage detection. E0VCCCIRREN 2 to 0 101_B: VMON Reset by E0VCC high voltage detection is permitted 110_B: VMON Reset by E0VCC low voltage detection is permitted 111_B: VMON Reset by E0VCC high/low voltage detection is permitted Other value: VMON Reset by E0VCC detection is not permitted
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	E0VCCFLTEN	Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF 0 : Disable output filter for $\overline{\text{VMONOUT}}$ and VMONF 1 : Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	E0VCCHDE	E0VCC high voltage detection enable 0: Disable E0VCC high voltage detection 1: Enable E0VCC high voltage detection
0	E0VCCLDE	E0VCC Low voltage detection enable 0: Disable E0VCC low voltage detection 1: Enable E0VCC low voltage detection

11.3.6.19 E0VCCFCR — E0VCC Monitor Filter Control Register

E0VCCFCR is a register to adjust the filtering width of E0VCC Monitor digital noise filter.

This register is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3184$_H$

Value after reset: 0x$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	E0VCCFLTW[1:0]		E0VCCCLKSEL[1:0]	
Value after reset	0	0	0	0	x	x	x	x
R/W	R	R	R	R	R	R	R	R

Table 11.23 E0VCCFCR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	E0VCCFLTW [1:0]	These bits select the minimum filtering width of digital noise filter. (The minimum filtering width is given by cycle-count of a clock selected by E0VCCCLKSEL1, 0) The following values are the Minimum Filtering Width (Cycle) 00_B: 20 cycle 01_B: 13 cycle 10_B: 8 cycle 11_B: 3 cycle
1, 0	E0VCCCLKSEL [1:0]	These bits select a clock of the digital noise filter from the following clocks. The following values are the Clock of the Digital Filter 00_B: CLK_HVIOSC frequency/256 01_B: CLK_HVIOSC frequency/128 10_B: CLK_HVIOSC frequency/64 11_B: CLK_HVIOSC frequency/32 **CAUTION** A frequency of CLK_HVIOSC is typ. 16 MHz.

11.3.6.20 VMONDSCR — VMON DeepSTOP Control Register

VMONDSCR is register to control VMON Reset, mask of VMON signal.

Bit 0 is set by flash option byte when Power On Reset, External Reset, or Debugger Disconnection Reset is released.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 31C0$_H$

Value after reset: 0X$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DSDETEN
Value after reset	0	0	0	0	0	0	0	x
R/W	R	R	R	R	R	R	R	R

Table 11.24 VMONDSCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	DSDETEN	VMON control during DeepSTOP 0: VMON disable (STOP) 1: VMON continue

11.3.6.21 VMONKCPROT — VMON Register Key Code Protection Register

The VMONKCPROT register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 3F00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 11.25 VMONKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0*1.
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writes to protected registers.
Write A5A5A501$_H$ to this register to enable writes to protected registers.

11.3.7 Operation

11.3.7.1 Basic Function

VMON watches AWOVDD, ISOVDD, VCC and E0VCC voltage.

VMON has high and low voltage detection circuit. If it detects voltage error, it records the error to VMONF register and notifies the error to the outside via $\overline{\text{VMONOUT}}$ pin. It also generates internal reset (VMON Reset) triggered by detection of voltage error. **Table 11.26** shows the symbol of primary high detection level, and primary low detection level for each power domain.

Table 11.26 Definition of Power domain detection Symbol name

Power domain	High detection level	Low detection level	High detect flag	Low detect flag
AWOVDD	$V_{AWOVDDMH}$	$V_{AWOVDDML}$	AWOVDDHVF	AWOVDDLVF
ISOVDD	$V_{ISOVDDMH}$	$V_{ISOVDDML}$	ISOVDDHVF	ISOVDDLVF
VCC	V_{VCCMH}	V_{VCCML}	VCCHVF	VCCLVF
E0VCC	$V_{E0VCCMH}$	$V_{E0VCCML}$	E0VCCHVF	E0VCCLVF

CAUTION

If system enters DeepSTOP mode, VMON function of ISOVDD area is masked.

(1) VMON detection flag

VMONF.ISOVDDHVF is set to “1” when ISOVDD becomes higher than high voltage detection level of VMON.

VMONF.ISOVDDLVF is set to “1” when ISOVDD becomes lower than low voltage detection level of VMON.

Table 11.26 shows the symbol of high detection flag, and low detection flag for each power domain.

These flags can be cleared to “0” by writing in the corresponding bit of VMONFC.

These flags are also cleared by Power On Reset (not cleared by other resets).

Figure 11.9 shows the operation example of VMONF.

(2) $\overline{\text{VMONOUT}}$ pin

While monitor voltage is higher than high voltage detection level or lower than low voltage detection level of VMON, $\overline{\text{VMONOUT}}$ pin outputs low.

If VMON does not detect voltage error $\overline{\text{VMONOUT}}$ pin outputs high.

To avoid unwanted operation of VMON circuit caused by instable power supply at power up, $\overline{\text{VMONOUT}}$ is fixed to low during Power On Reset period except when triggered by Debugger Reset.

$\overline{\text{VMONOUT}}$ outputs error detection result even during other reset period except Power On Reset.

Figure 11.9 shows the operation example of that is $\overline{\text{VMONOUT}}$ operated by ISOVDD.

Similarly, for other power supply, the operation timing of $\overline{\text{VMONOUT}}$ is the same as that for ISOVDD.

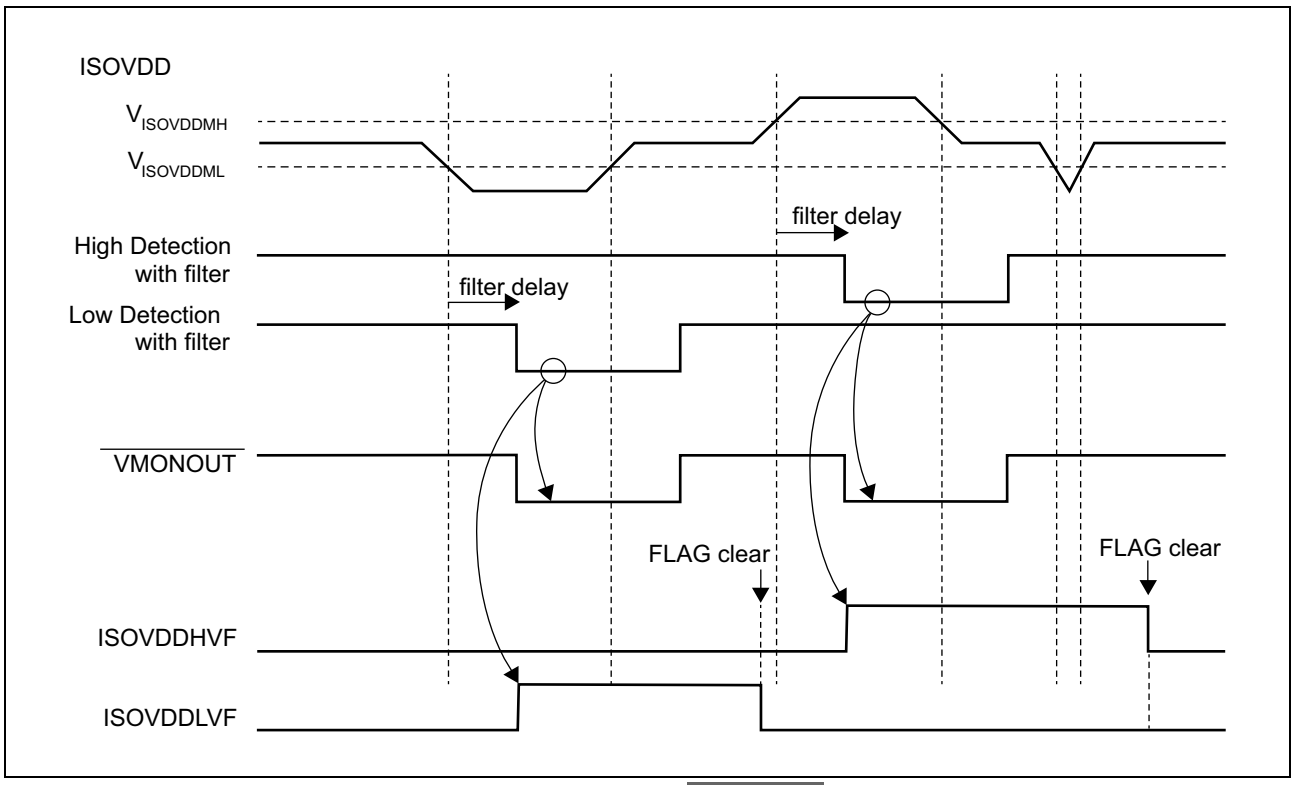


Figure 11.9 Example of VMONF and $\overline{\text{VMONOUT}}$ Operation (ISOVDD)

(3) VMON Reset

If ISOVDD voltage becomes higher than high voltage detection level of VMON when "101_B" or "111_B" were set to ISOVDDDE.ISOVDDCIRREN[2:0] in advance, VMON Reset occurs.

Similarly, if ISOVDD voltage becomes lower than low voltage detection level of VMON when "110_B" or "111_B" were set to ISOVDDDE.ISOVDDCIRREN[2:0] in advance, VMON Reset occurs.

If VCC voltage becomes higher than high voltage detection level of VMON when "101_B" or "111_B" were set to VCCDE.VCCCIRREN[2:0] in advance, VMON Reset occurs.

Similarly, if VCC voltage becomes lower than low voltage detection level of VMON when "110_B" or "111_B" were set to VCCDE.VCCCIRREN[2:0] in advance, VMON Reset occurs

If E0VCC voltage becomes higher than high voltage detection level of VMON when "101_B" or "111_B" were set to E0VCCDE.E0VCCCIRREN[2:0] in advance, VMON Reset occurs.

Similarly, if E0VCC voltage becomes lower than low voltage detection level of VMON when "110_B" or "111_B" were set to E0VCCDE.E0VCCCIRREN[2:0] in advance, VMON Reset occurs.

VMON Reset is released after the specified time has been passed since VMON detects the normal voltage.

NOTE

LDET and HDET of AWOVDD will always generate the VMON Reset at the time of low voltage detection, except for during the diagnosis.

Figure 11.10 shows the operation example of VMON Reset that is operated by ISOVDD.

Similarly, for other power supply, the operation timing of VMON Reset is the same as that for ISOVDD.

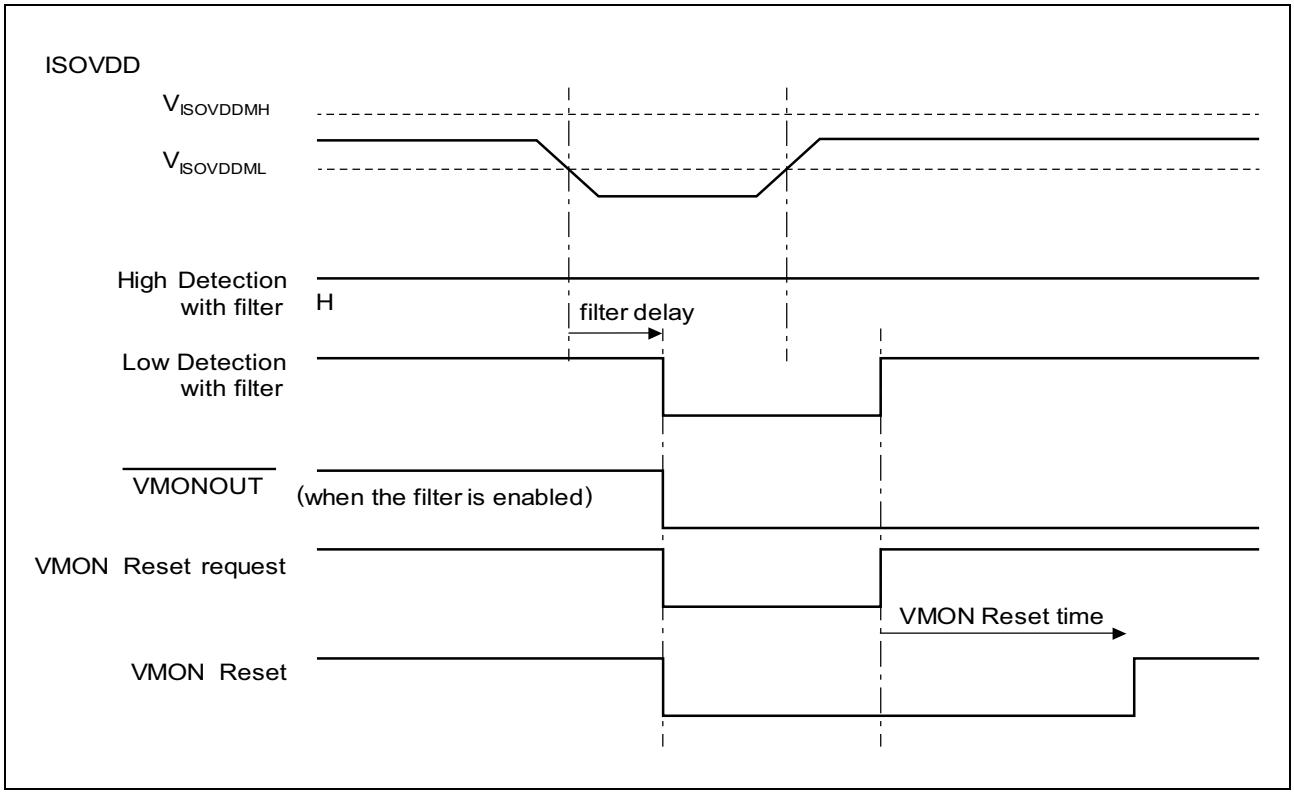


Figure 11.10 Example of VMON Reset Operation (ISOVDD)

(4) Digital Noise Filter

The output signal of the voltage detector passes through a digital noise filter to remove an unintended glitch. See "**Section 55, Electrical Characteristics**" for further information of the filters characteristics.

Digital noise filter clock and the cycle are controlled by setting of the following registers. (These registers are read only and set by flash option byte.)

- ISOVDDFCR
- AWOVDDFCR
- VCCFCR
- E0VCCFCR

The relation of filter enable is shown in **Figure 11.5**, **Figure 11.6**, **Figure 11.7** and **Figure 11.8**.

11.3.7.2 VMON function in BIST and Serial Programming Mode (Mask of $\overline{\text{VMONOUT}}$ pin and VMON Reset)

$\overline{\text{VMONOUT}}$ output and VMON Reset by detection of ISOVDD and AWOVDD voltage fluctuations are always masked during BIST after Power On Reset, External Reset, and Debugger Disconnection Reset.

It is possible to mask $\overline{\text{VMONOUT}}$ output and VMON Reset by setting ISOVDDDE.ISOVDDFBISTME and AWOVDDDE.AWOVDDFBISTME to “0” during BIST after VMON Reset. In this case, $\overline{\text{VMONOUT}}$ is fixed to high level and VMON Reset does not occur.

NOTE

$\overline{\text{VMONOUT}}$ output and VMON Reset by detection of ISOVDD and AWOVDD voltage fluctuations are not always masked during BIST after System Reset 2 and DeepSTOP Reset occur. $\overline{\text{VMONOUT}}$ output and VMON Reset by detection of VCC and E0VCC voltage fluctuations are always not masked during BIST.

In Serial Programming Mode, $\overline{\text{VMONOUT}}$ output is fixed to high even by detection of ISOVDD/AWOVDD/E0VCC/VCC voltage fluctuations during writing and erasing operations to the Flash memory.

11.3.7.3 Diagnosis Function

(1) Change of VMON detection level for the error injection

Table 11.27 List of Setting register

Condition	Detection target	Setting register
VMONDIAGME.VMONDIAGME is “0”	ISOVDD	VMONDIAG.ISOVDDDIAGH or VMONDIAG.ISOVDDDIAGL
VMONDIAGME.VMONDIAGME is “0”	AWOVDD	VMONDIAG.AWOVDDDIAGH or VMONDIAG.AWOVDDDIAGL
VMONDIAGME.VMONDIAGME is “0”	VCC	VMONDIAG.VCCDIAGH or VMONDIAG.VCCDIAGL
VMONDIAGME.VMONDIAGME is “0”	E0VCC	VMONDIAG.E0VCCDIAGH or VMONDIAG.E0VCCDIAGL

VMON error detection signal can be intentionally generated by setting these registers even when each power supply voltage is in the operating range.

NOTE

During diagnosis, Abnormal voltage can not be detected by VMON.

(2) Mask of $\overline{\text{VMONOUT}}$ and VMON Reset

Table 11.28 List of VMONOUT mask

Condition	Mask target
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.ISOVDDERRMD is set to “1”	VMON_ISOVDD_ERROROUT
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.AWOVDDERRMD is set to “1”	VMON_AWOVDD_ERROROUT
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VCCERRMD is set to “1”	VMON_VCC_ERROROUT
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.E0VCCERRMD is set to “1”	VMON_E0VCC_ERROROUT

If above four settings are done, $\overline{\text{VMONOUT}}$ is dependent on the setting of DMON during diagnosis.

Table 11.29 List of VMON Reset mask

Condition	Mask target
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.ISOVDDRESMD is set to “1”	VMON_ISOVDD_REQ_RESET
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.AWOVDDRESMD is set to “1”	VMON_AWOVDD_REQ_RESET
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.VCCRESMD is set to “1”	VMON_VCC_REQ_RESET
VMONDIAGME.VMONDIAGME is “0” and VMONDMASK.E0VCCRESMD is set to “1”	VMON_E0VCC_REQ_RESET

(3) The flow of VMON diagnosis application

VMONDIAGME.VMONDIAGME is initialized to “0” by Power On Reset, External Reset, Debugger Disconnection Reset, and VMON Reset which is AWOVDD detection. So VMON diagnosis function is enabled at start-up after these resets.

The user’s application controls the diagnostic function by VMONDMASK, VMONDIAG and VMONDIAGFE.

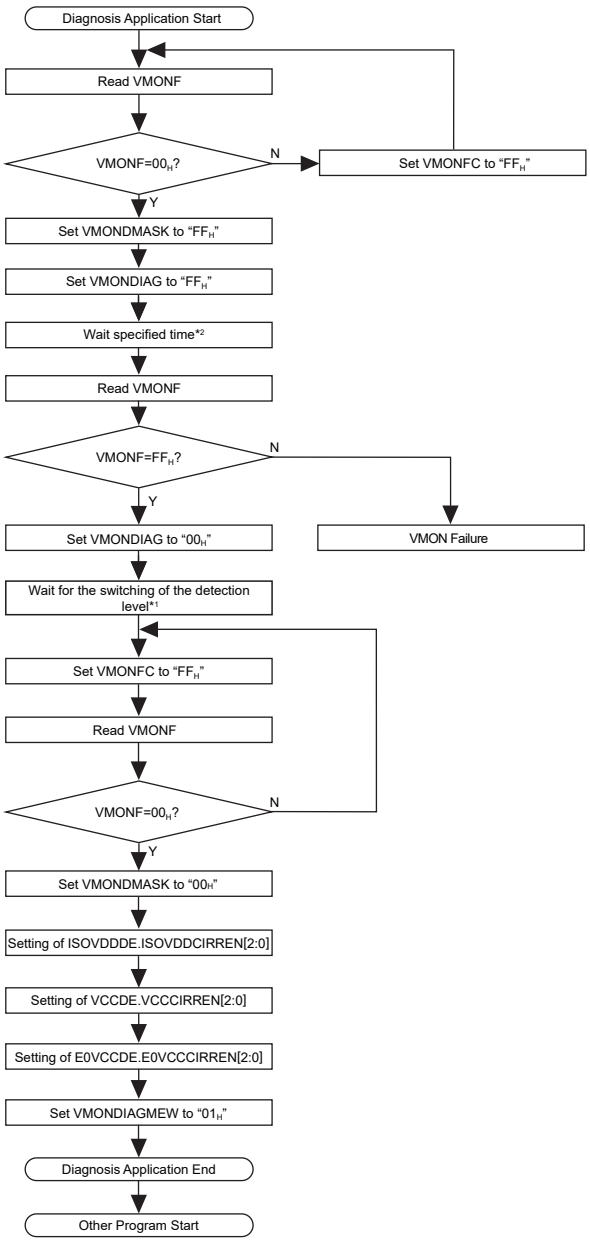
Figure 11.11 shows the example of the diagnosis application.

Figure 11.12 shows the example of the $\overline{\text{VMONOUT}}$ diagnosis application.

Figure 11.13 shows the example of the VMON Reset diagnosis application.

After diagnosis application was over, write “1” to VMONDIAGME.VMONDIAGME and disable VMON diagnosis function immediately.

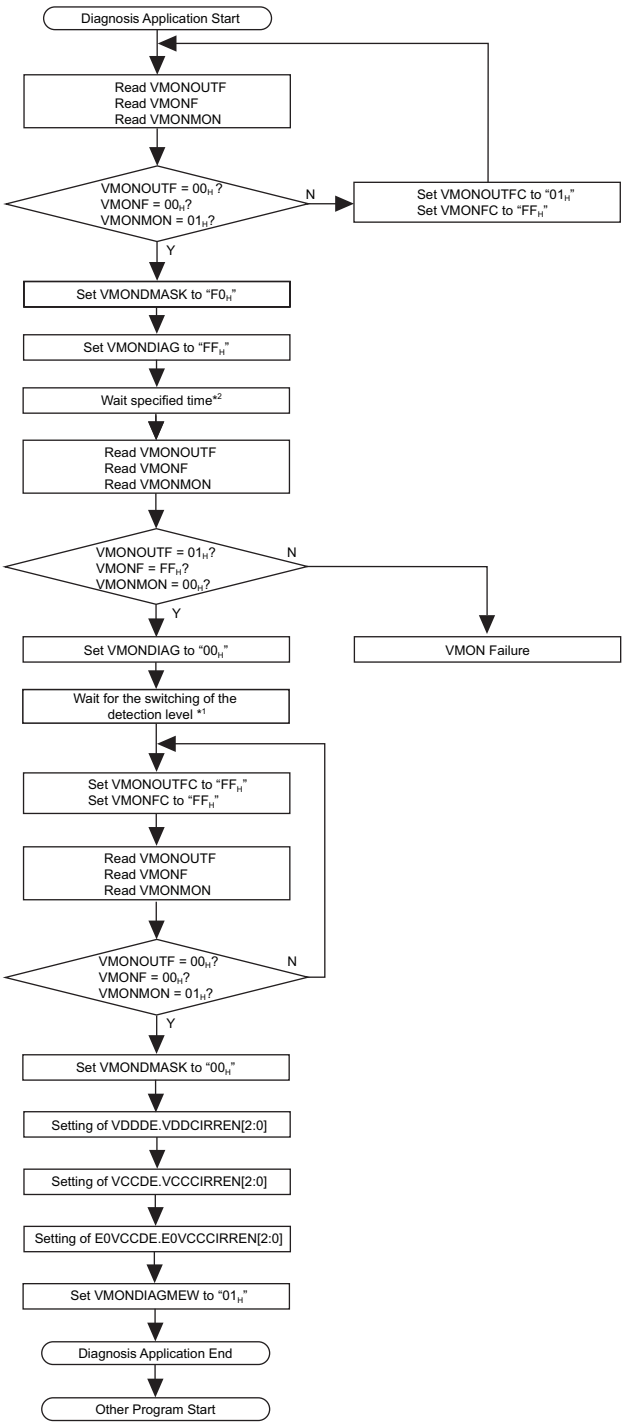
This function protects each flag against erroneous operation when each power supply is out of the operating range.



Note 1. The processing must produce waiting period of at least 30 µs.

Note 2. Wait more than the following time.
Wait time = 10 + 0.125 ×Dclk × (Wfil+1) [µs]
Dclk: Divided value. Value is selected by nnnFCR.nnnCLKSEL[1:0]
Wfil: Filtering Width. Value is set by nnnFCR.nnnFLTW[1:0]
nnn: ISOVDD/AWOVDD/VCC/E0VCC

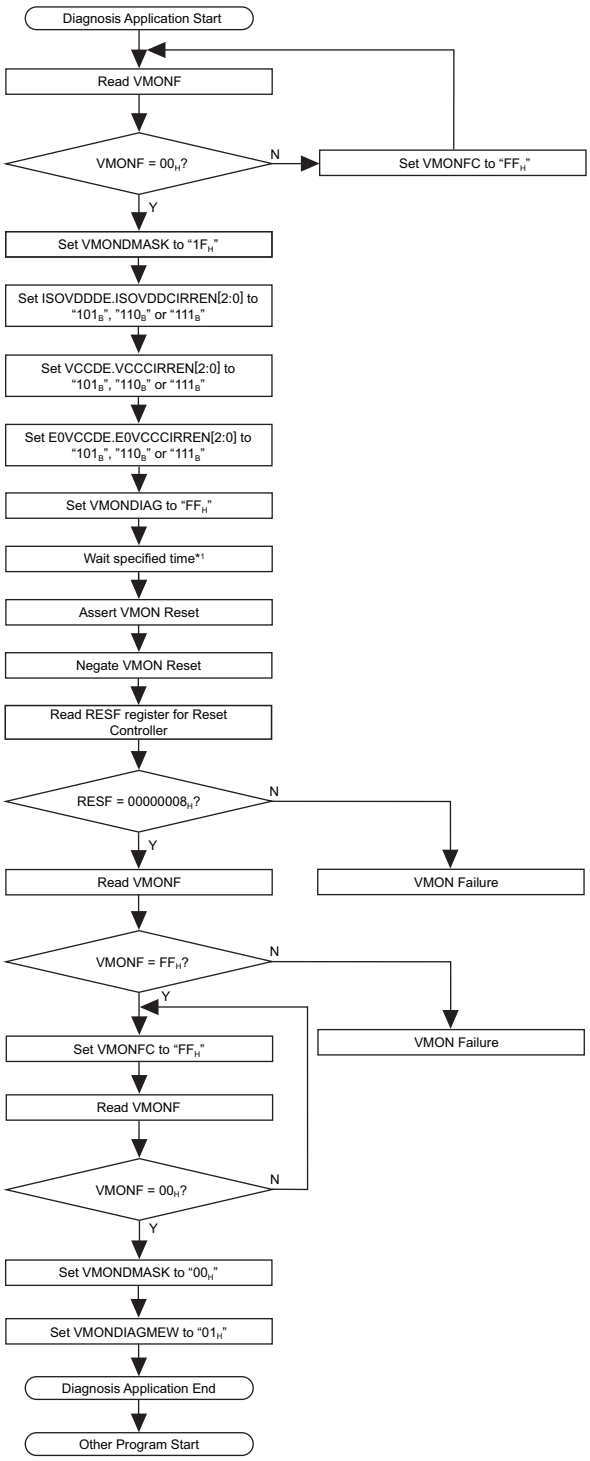
Figure 11.11 The Example of the Flowchart of VMON Diagnosis Application



Note 1. The processing must produce waiting period of at least 30µs.

Note 2. Wait more than the following time.
Wait time = 10 + 0.125 ×Dclk × (Wfil+1) [µs]
Dclk: Divided value. Value is selected by nnnFCR.nnnCLKSEL[1:0]
Wfil: Filtering Width. Value is set by nnnFCR.nnnFLTW[1:0]
nnn: ISOVDD/AWOVDD/VCC/E0VCC

Figure 11.12 The Example of the Flowchart of VMON Diagnosis Application (VMONOUT)



Note 1. Wait more than the following time.
Wait time = 10 + 0.125 ×Dclk × (Wfil+1) [µs]
Dclk: Divided value. Value is selected by nnnFCR.nnnCLKSEL[1:0]
Wfil: Filtering Width. Value is set by nnnFCR.nnnFLTW[1:0]
nnn: ISOVDD/AWOVDD/VCC/E0VCC

Figure 11.13 The Example of the Flowchart of VMON Diagnosis Application (VMON Reset)

(4) The flow of VMON re-diagnosis after DeepSTOP

The VMON detection of voltage monitor is used for monitoring the domains ISOVDD, AWOVDD, E0VCC and VCC.

When MCU return to RUN form DeepSTOP, VMON will be able to re-diagnosis.

11.3.7.4 Function in Standby

If system enters DeepSTOP mode, VMON function of ISOVDD detection was stopped.

The following shows the control of detections for per system mode.

Table 11.30 Function in Standby

State	ISOVDD Detection	AWOVDD Detection	VCC Detection	E0VCC Detection
DeepSTOP	— (Stopped)	DSDETEN	DSDETEN	DSDETEN
		DSDETEN = 0 Stopped	DSDETEN = 0 Stopped	DSDETEN = 0 Stopped
		DSDETEN = 1 Continue. Follow the values below AWOVDDFLTEN	DSDETEN = 1 Continue. Follow the values bellow VCCHDE VCCLDE VCCFLTEN	DSDETEN = 1 Continue. Follow the values below E0VCCHDE E0VCCLDE E0VCCFLTEN

In DeepSTOP mode, it is controlled by the value of DSDETEN to stop the detections or continue detections.

11.3.7.5 VMON function of mask

The following list shows the mask correspondence of the VMON function in the **Table 11.31**.

Table 11.31 VMON function of mask

Power Supply	VMON Function	Power On Reset	Serial Programming mode	Common Case	Field-BIST0	Field-BIST 1/2	Diagnosis	Deep STOP
ISOVDD	$\overline{\text{VMONOUT}}$	— (Always error)	Always masked	ISOVDDH/LDE	ISOVDDFBISTME	—	ISOVDDERRMD	Always masked
	VMON Reset	Always masked	—	ISOVDDH/LDE ISOVDDCIRREN	ISOVDDFBISTME	—	ISOVDDRESMD	Always masked
	VMON Flag	Always masked	—	ISOVDDH/LDE	—	—	—	Always masked
AWOVDD	$\overline{\text{VMONOUT}}$	— (Always error)	Always masked	—	AWOVDDFBISTME	—	AWOVDDERRMD	DSDETEN
	VMON Reset	Always masked	—	—	AWOVDDFBISTME	—	AWOVDDRESMD	DSDETEN
	VMON Flag	Always masked	—	—	—	—	—	DSDETEN
VCC	$\overline{\text{VMONOUT}}$	— (Always error)	Always masked	VCCH/LDE	—	—	VCCERRMD	DSDETEN
	VMON Reset	Always masked	—	VCCH/LDE VCCCIRREN	—	—	VCCRESMD	DSDETEN
	VMON Flag	Always masked	—	VCCH/LDE	—	—	—	DSDETEN
E0VCC	$\overline{\text{VMONOUT}}$	— (Always error)	Always masked	E0VCCH/LDE	—	—	E0VCCERRMD	DSDETEN
	VMON Reset	Always masked	—	E0VCCH/LDE E0VCCCIRREN	—	—	E0VCCRESMD	DSDETEN
	VMON Flag	Always masked	—	E0VCCH/LDE	—	—	—	DSDETEN

Note: —: Not masked, Other: how to mask (refer each register spec about polarity of setting)
Mask setting described in left side column is prior to that in right side column in the table.
(e.g. If $\overline{\text{VMONOUT}}$ function is masked by setting 0 to VCCH/LDE, DSDETEN is ignored in DeepSTOP)

11.3.8 Usage Notes

11.3.8.1 Flag Bit (VMONF)

Setting 1 to VMONF by error detection is prior to clearing VMONF by VMONFC.

These functions protect each flag against erroneous operation when each power supply is out of the operating range.

When Monitor voltages are recovered to the operating range, the MCU need to be initialized by External Reset or VMON Reset, or the MCU may still continue erroneous operation and then unintended clear of flag may occur.

11.3.8.2 AWOVDD Primary Low Detection

Because of internal circuit guard of this MCU, VMON Reset might occur in shorter time than the filter width when AWOVDD voltage is lower than the lowest level of VDD primary low detection range.

In that case, AWOVDDLVF will not be set by the AWOVDD primary low detection.

11.4 Delay Monitor (DMON)

11.4.1 Features

DMON can detect abnormalities in delay time of transistor devices by using an on-chip delay monitor while the microcontroller is working. It can only monitor the delay time on the ISOVDD domain. It detects an error if the monitored delay time is lower than a threshold according to the fabrication process conditions.

DMON assists the power supply voltage monitor (VMON) which detects the low level voltage of ISOVDD.

When DMON detects the delay error within the ranges of VMON detection, it notifies VMON about the error detection to operate VMON Reset normally.

When MainOSC error was detected in CLMA0, DMON possibility that output an error.

11.4.2 Clock Supply

The clock supply to the DMON is shown in the following table.

Table 11.32 Clock Supply

Module Name	Module Clock Name	Clock Name
DMON	MainOSC	CLK_MOSC
	Register access clock	CLK_LSB

11.4.3 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA / DTS requests.

This module has no error notifications to ECM.

11.4.4 External Input/Outputs

This module has no external input/output pins.

11.4.5 Overview

The On-Chip delay monitor measures the delay time related to the power supply voltage. It converts analog delay time into digital codes (DCODE) and outputs the DCODE into a comparator. The DCODE is compared with the threshold which is a fixed value. When the DCODE becomes lower than the threshold, an error output signal (DMON_ERROROUT) which and a reset request signal (DMON_REQRESET) are generated to notify the VMON.

11.4.5.1 Block Diagram

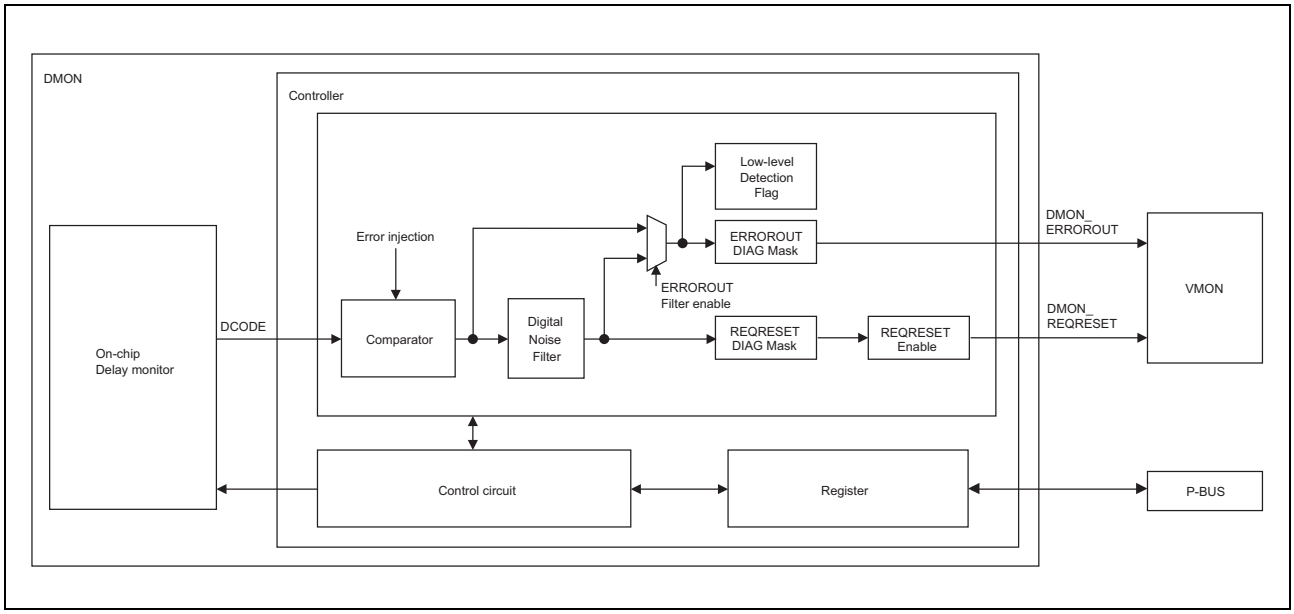


Figure 11.14 Block Diagram of the Delay Monitor

11.4.6 Registers

11.4.6.1 List of Registers

The register list related to DMON is shown in **Table 11.33**.

Table 11.33 List of Registers (DMON)

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 2800$_H$	DMONF	DMON Factor Register	32	0000 0000$_H$	PBG20 #2	
FF98 2804$_H$	DMONFC	DMONF Clear Register	32	0000 0000$_H$	PBG20 #2	DMONKCPROT0
FF98 2808$_H$	DMONDIAGME	DMON DIAG Monitor Enable Register	32	0000 0000$_H$	PBG20 #2	
FF98 280C$_H$	DMONDIAGMEW	DMONDIAGME Write Register	32	0000 0000$_H$	PBG20 #2	DMONKCPROT0
FF98 2810$_H$	DMONDMASK	DMON Detection Output Diagnosis Mask Register	32	0000 0011$_H$	PBG20 #2	
FF98 2814$_H$	DMONDIAG	DMONDIAG Mode Setting Register	32	0000 0000$_H$	PBG20 #2	
FF98 2818$_H$	DMONDE	DMON Detection Enable Register	32	0000 0000$_H$	PBG20 #2	DMONKCPROT0
FF98 281C$_H$	DMONFCR	DMON Filter Control Register	32	0000 0000$_H$	PBG20 #2	
FF98 2900$_H$	DMONTEST	Delay Monitor Test Register	32	0000 0000$_H$	PBG20 #2	DMONKCPROT2
FF98 2F00$_H$	DMONKCPROT0	DMON Register 0 Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	
FF98 3F80$_H$	DMONKCPROT2	DMON Register 2 Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	

Register reset conditions are shown in **Table 11.34**.

Table 11.34 Register Reset Conditions for DMON

Register Name	Reset Category							DMONTEST.RESDMON Reset Target
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset	RESDMON = 1
DMONF	√	√*1	—	—	√	—	—	—
DMONFC	√	√	—	—	√	—	—	√
DMONDIAGME	√	√	√	—	√	—	—	√
DMONDIAGMEW	√	√	√	—	√	—	—	√
DMONDIAG	√	√	√	—	√	—	—	√
DMONDMASK	√	√	√	—	√	—	—	√
DMONDE	√	√	√	—	√	—	—	√
DMONFCR	√	√	√	—	√	—	—	√
DMONTEST	√	√	√	—	√	—	—	—
DMONKCPROT0	√	√	√	√	√	—	—	√
DMONKCPROT2	√	√	√	√	√	—	—	—

Note 1. Only VMON Reset which is AWOVDD detection.

11.4.6.2 DMONF — DMON Factor Register

DMONF register indicates that an error is detected by DMON. Each flag of DMONF can be cleared individually by writing “1” to the corresponding bit in DMONFC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 2800$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONLVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.35 DMONF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DMONLVF	Lower limit delay detection flag 0: No lower limit delay error detected 1: Lower limit delay error occurred

Note 1. DMONLVF is guaranteed only when VMON detection of ISOVDD does not occur.

11.4.6.3 DMONFC — DMONF Clear Register

DMONFC is a register to clear the DMONF register. The read value of this register is always $0000\ 0000_H$.

Access: This register is a write-only register that can be written in 32-bit units.

Address: $FF98\ 2804_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONLVFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 11.36 DMONFC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DMONLVFC	Clear lower limit delay detection flag 0: Writing 0 has no effect 1: Writing 1 will clear DMONLVF

11.4.6.4 DMONDIAGME — DMON DIAG Monitor Enable Register

DMONDIAGME is a read only register to control DIAG function.

This register reflects the settings written to the DMONDIAGMEW.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 2808$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONDIAGME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.37 DMONDIAGME Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DMONDIAGME	Permit DIAG function 0: DIAG function of the DMON can be enabled 1: DIAG function of the DMON cannot be enabled

11.4.6.5 DMONDIAGMEW — DMONDIAGME Write Register

DMONDIAGMEW is a register to set values of DMONDIAGME. Writing is permitted only once after Power On Reset, System Reset 1/2, DeepSTOP Reset or DMON Reset (RESDMON = 1) was released. Subsequent write operation is ignored. The read value of this register is always 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FF98 $280C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONDIAGMEW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 11.38 DMONDIAGMEW Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DMONDIAGMEW	The data written in this bit is set to DMONDIAGME.DMONDIAGME.

11.4.6.6 DMONDMASK — DMON Detection Output Diagnosis Mask Register

DMONDMASK is a register to mask DMON_REQRESET and DMON_ERROROUT when DMONDIAGME.DMONDIAGME = 0. DMONDMASK setting is ignored if DMONDIAGME.DMONDIAGME = 1.

Access: This register can be read or written in 32-bit units.

Address: FF98 2810$_H$

Value after reset: 0000 0011$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DMON RESMD	—	—	—	DMON ERRMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Table 11.39 DMONDMASK Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	DMONRESMD	This bit masks DMON_REQRESET when DMONDIAGME.DMONDIAGME = 0. 0: DMON_REQRESET is not masked. 1: DMON_REQRESET is masked.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DMONERRMD	This bit masks DMON_ERROROUT when DMONDIAGME.DMONDIAGME = 0. 0: DMON_ERROROUT is not masked. 1: DMON_ERROROUT is masked.

11.4.6.7 DMONDIAG — DMONDIAG Mode Setting Register

DMONDIAG register is used for controlling diagnostic function of DMON.

This register is valid only when DMONDIAGME.DMONDIAGME = 0. The setting of DMONDIAG is ignored if DMONDIAGME.DMONDIAGME = 1.

Access: This register can be read or written in 32-bit units.

Address: FF98 2814$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMON DIAGL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.40 DMONDIAG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DMONDIAGL	This bit is used for error injection of diagnosis function when DMONDIAGME = 0. 0: Normal operation. 1: Execute error injection.

11.4.6.8 DMONDE — DMON Detection Enable Register

DMONDE is a register to control DMON detection, the mask of DMON signal and filter of DMON signal.

Writing in DMONDE is permitted during a period before writing in DMONDIAGMEW after Power On Reset, System Reset 1/2, DeepSTOP Reset or DMON Reset (RESDMON = 1) was released. Writing in this register is ignored, after writing in DMONDIAGMEW.

Access: This register can be read or written in 32-bit units.

Address: FF98 2818$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DMONCIRREN1	—	DMONCIRREN0	—	DMONFLTEN	—	—	DMONLDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W

Table 11.41 DMONDE Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DMONCIRREN1	Permit DMON_REQRESET notification. DMONCIRREN[1:0] 00_B, 01_B, 10_B: DMON_REQRESET is not permitted 11_B: DMON_REQRESET is permitted
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DMONCIRREN0	Refer to bit7 (DMONCIRREN1)
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	DMONFLTEN	Enable output filter for DMON_ERROROUT and DMON Error Detection Flag 0: Disable output filter for DMON_ERROROUT and DMON Error Detection Flag 1: Enable output filter for DMON_ERROROUT and DMON Error Detection Flag
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DMONLDE	DMON Low delay detection enable 0: Disable DMON lower limit delay detection 1: Enable DMON lower limit delay detection

CAUTION

Restrictions

- When ISOVDDDE.ISOVDDLDE = “0”, setting the DMONLDE to “1” is prohibited.
- When ISOVDDDE.ISOVDDCIRREN[2:0]! = “110_B” or “111_B”, setting the DMONCIRREN[1:0] to “11_B” is prohibited.

11.4.6.9 DMONFCR — DMON Filter Control Register

DMONFCR is a register to adjust the filtering width of DMON digital noise filter.

Writing in DMONFCR is permitted during a period before writing in DMONDIAGMEW after Power On Reset, System Reset 1/2, DeepSTOP Reset or DMON Reset (RESDMON = 1) was released. Writing in this register is ignored, after writing in DMONDIAGMEW.

Access: This register can be read or written in 32-bit units.

Address: FF98 281C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMONFLTW[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.42 DMONFCR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMONFLTW [2:0]	These bits select the minimum filtering width of digital noise filter. The minimum filtering width is given by number of sampling times 000_B: 1 sampling time 001_B: 2 sampling times 010_B: 4 sampling times 011_B: 8 sampling times 100_B: 16 sampling times 101_B: 32 sampling times 110_B: 64 sampling times 111_B: 128 sampling times **CAUTION** A sampling period is 5 µs (16 MHz), 4 µs (20/40 MHz), 3.33 µs (24 MHz).

11.4.6.10 DMONTEST — Delay Monitor Test Register

This register is used for the self-test of the Delay monitors.

Access: This register can be read or written in 32-bit units.

Address: FF98 2900$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESDMON
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.43 DMONTEST Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RESDMON	DMON test reset signal control Asserting this bit re-initializes the Delay Monitor in order to continue normal operation. 0: Reset signal for DMON is released. 1: Reset signal for DMON is asserted.

11.4.6.11 DMONKCPROT0 — DMON Register 0 Key Code Protection Register

The DMONKCPROT0 register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 2F00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 11.44 DMONKCPROT0 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0. *1
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writes to protected registers.
Write A5A5A501$_H$ to this register to enable writes to protected registers.

11.4.6.12 DMONKCPROT2 — DMON Register 2 Key Code Protection Register

The DMONKCPROT2 register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 3F80$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 11.45 DMONKCPROT2 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0*1.
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writes to protected registers.
Write A5A5A501$_H$ to this register to enable writes to protected registers.

11.4.7 Operation

11.4.7.1 Basic Function

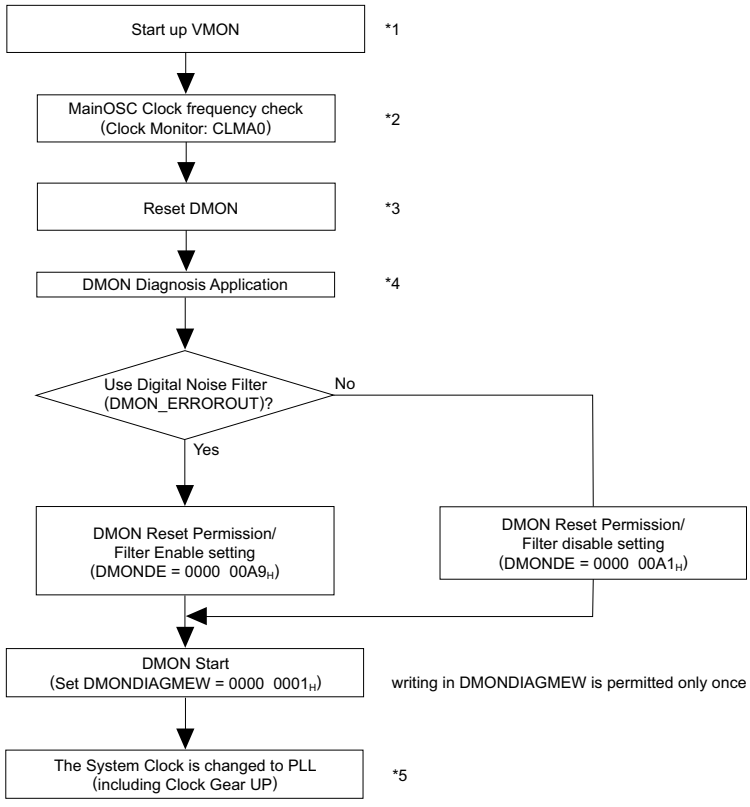
DMON monitors the delay time depending on ISOVDD supply voltage.

If it detects the delay error, it records the error to DMONF register. It also generates DMON_ERROROUT and DMON_REQRESET triggered by detection of the lower limit delay. It notifies the error to VMON.

Figure 11.15 shows the flowchart of starting up DMON. DMON begins to operate after VMON starts up and the Clock Monitor confirms that MainOSC clock is normally operating. The user must invoke DMON after starting up VMON, checking Clock Monitor whether MainOSC clock is normally operating, and resetting DMON. After DMON starts up, the system clock needs to be changed to PLL and the clock gear up is carried out.

The timing chart of DMON operation is shown in **Figure 11.16**.

When Power On Reset, System Reset 1/2, DeepSTOP Reset or DMON Reset (RESDMON = 1) is asserted, DMON is also initialized by these resets. After restart from the reset, the start-up sequence needs to be executed again.



Note 1. Refer to **Section 11.3**.

Note 2. Refer to **Section 14, Clock Monitor (CLMA)**.

Note 3. Refer to **Section 11.4.7.4**.

Note 4. Refer to **Section 11.4.7.2**.

Note 5. Refer to **Section 13, Clock Controller**.

Figure 11.15 Flowchart of the DMON Start-Up

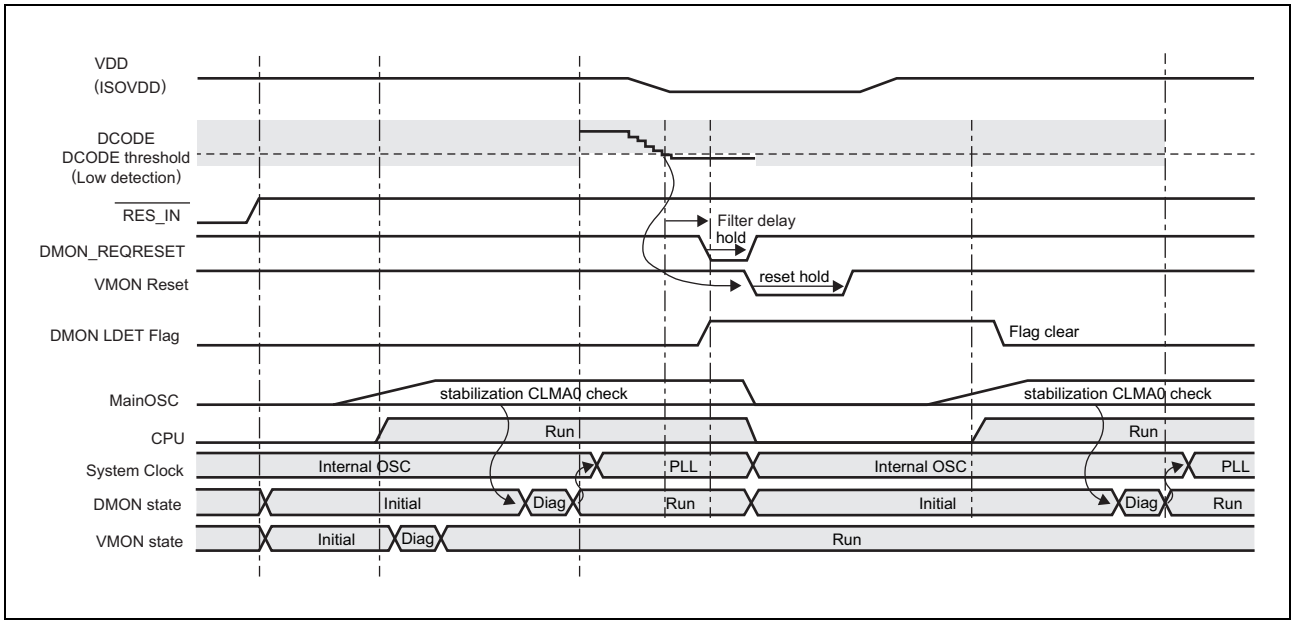
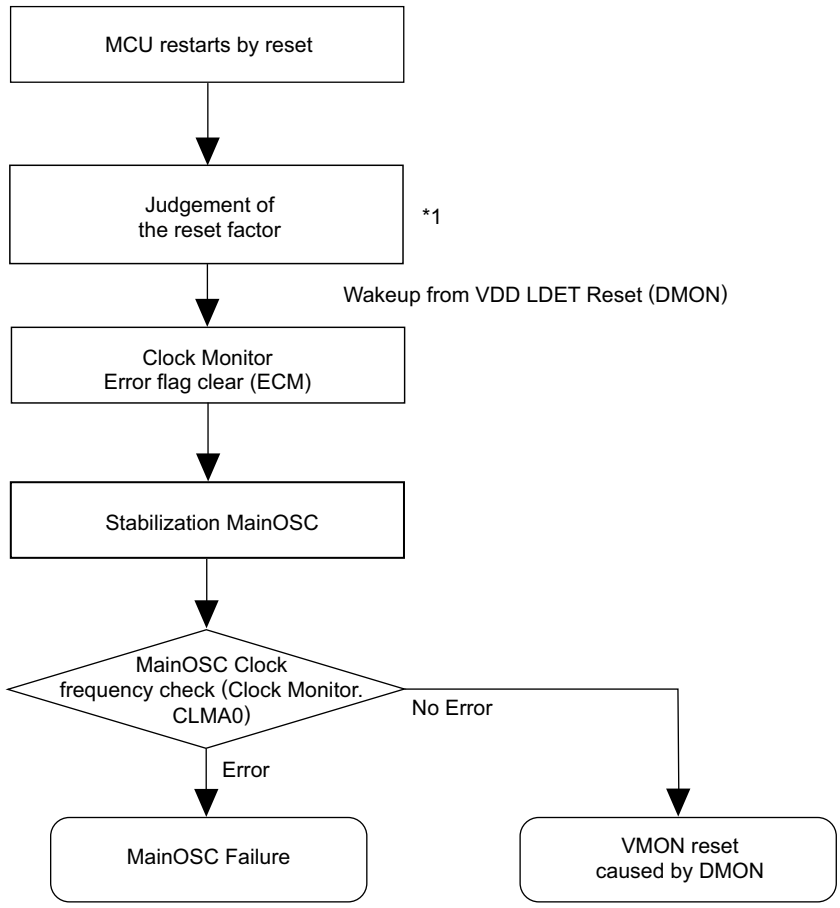


Figure 11.16 Timing Chart of the DMON Function

DMON generates the sampling period from MainOSC clock (CLK_MOSC). If the frequency of MainOSC is faster than the spec, the DCODE will become smaller than expected value due to shortening of sampling period. Therefore, there is a possibility that DMON detects the delay error before the error detection of clock monitor for MainOSC and the MCU is initialized by VMON Reset. If the user distinguishes between the failure of MainOSC and the reset factor caused by DMON error after the MCU restarts from VMON Reset, the user needs to confirm DMON error flag and check the frequency of MainOSC by clock monitor (CLMA0). **Figure 11.17** shows the example of flowchart to distinguish the factor.



Note 1. Refer to **Section 11.6.1**.

Figure 11.17 The Example of the Flowchart to Distinguish the Failure of Main OSC

(1) DMON Error Detection Flag

DMON sets “1” to DMONF.DMONLVF when the DCODE becomes lower than the low detection threshold level.

The flag can be cleared to “0” by writing in the corresponding bit of DMONFC.

The flag is also cleared by Power On Reset. **Figure 11.18** shows the operation example of DMONF.

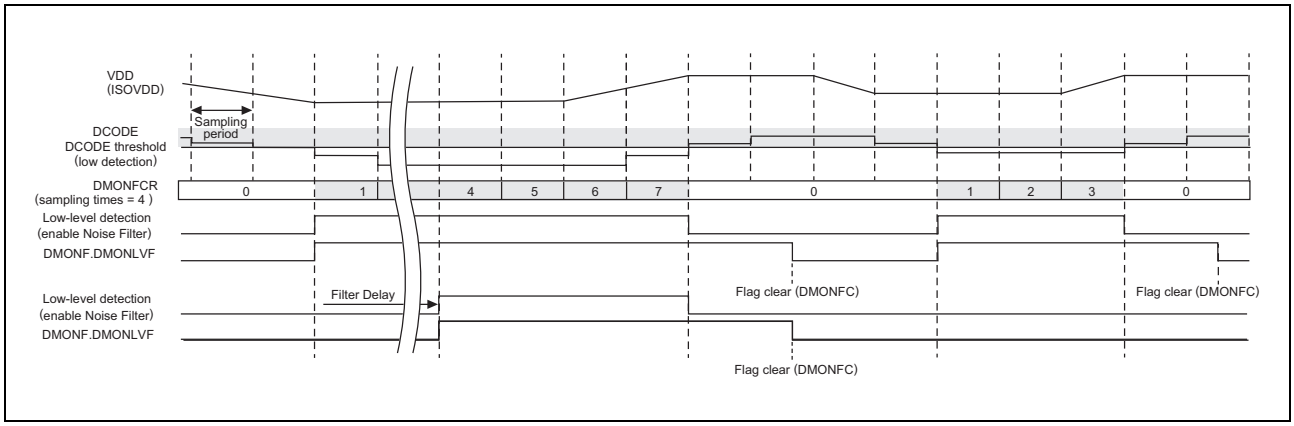


Figure 11.18 DMON Error Detection Flag

(2) DMON_REQRESET Output

If the DCODE value becomes lower than the DCODE threshold level when DMONDE.DMONCIRREN is set to "11_B", DMON notifies VMON that the DMON_REQRESET has occurred. After that, VMON Reset occurs and the MCU including DMON is initialized. DMON continues to hold the DMON_REQRESET state until DMON is initialized by VMON Reset. **Figure 11.19** shows the operation example of the DMON_REQRESET.

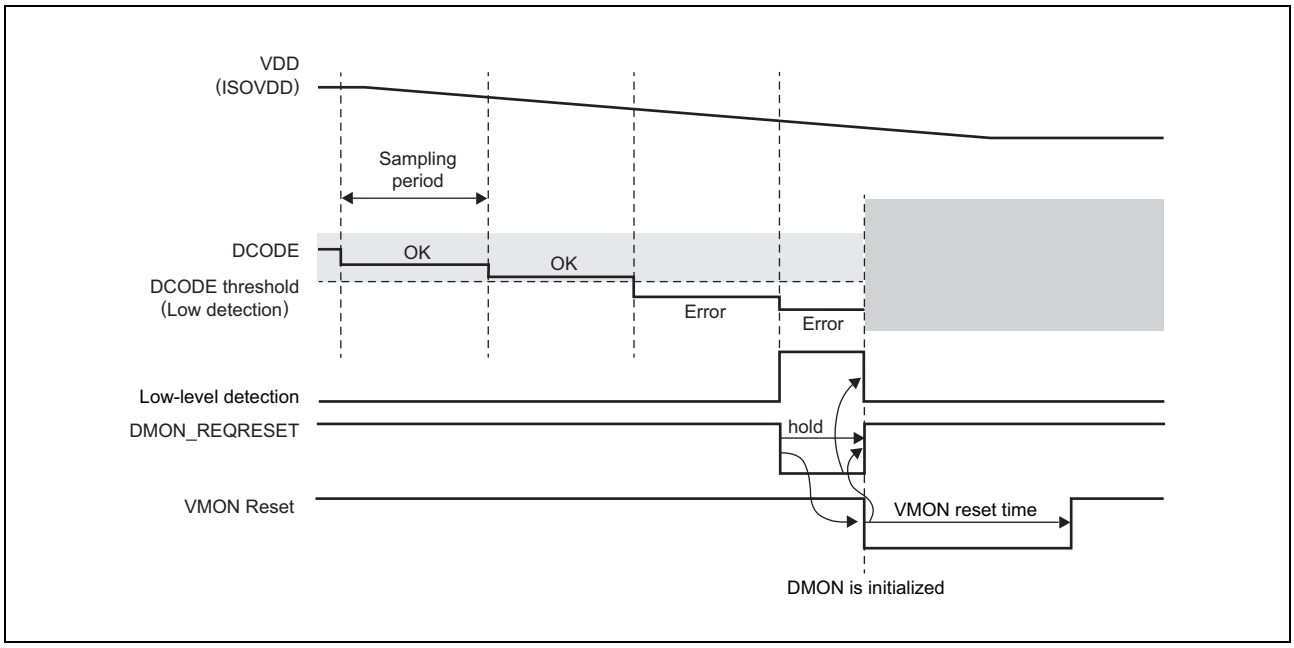


Figure 11.19 DMON_REQRESET Operation

(3) The Digital Noise Filter

The output signal of the DMON comparators passes through a digital noise filter to remove unintended glitches.

Whether to use filtered output or to bypass the filter is selectable by DMONDE.DMONFLTEN.

The Digital noise filter period can be changed by DMONFCR register.

11.4.7.2 Diagnosis Function

(1) Change of DMON detection level for the error injection

When DMONDIAGME.DMONDIAGME is set to “0”, Lower limit delay threshold level can be changed by DMONDIAG.DMONDIAGL.

DMON error detection signal can be intentionally generated by setting this register when VMON didn’t detect the voltage error and the clock monitor CLMA0 didn’t detect the frequency abnormalities.

(2) Mask of DMON_ERROROUT and DMON_REQRESET

When DMONDIAGME.DMONDIAGME is set to “0” and DMONDMASK.DMONERRMD is set to “1”, DMON_ERROROUT can be masked.

When DMONDIAGME.DMONDIAGME is set to “0” and DMONDMASK.DMONRESMD is set to “1”, DMON_REQRESET can be masked.

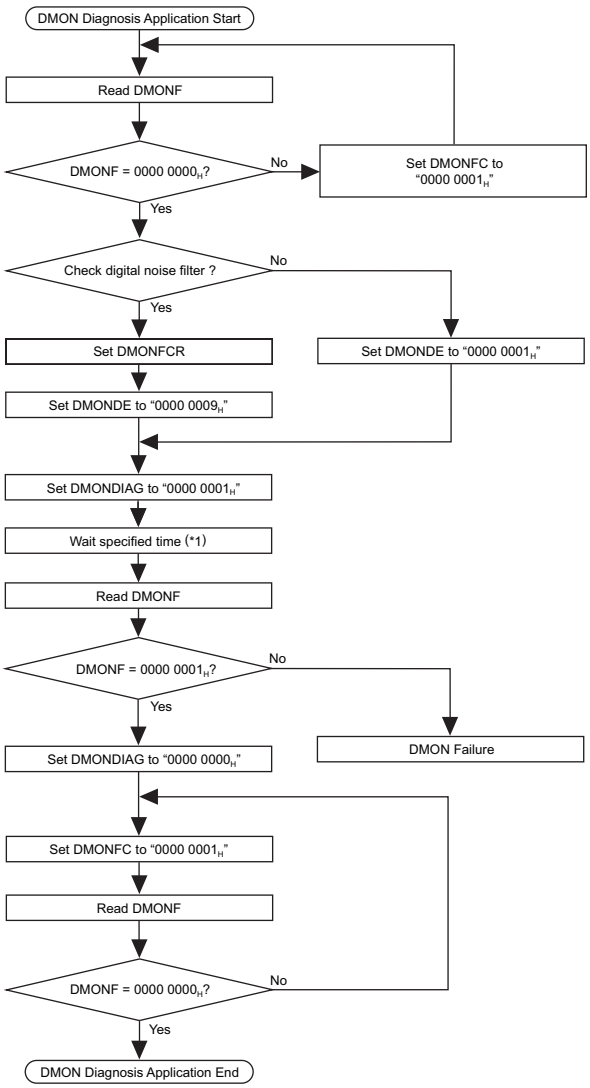
(3) The flow of DMON diagnosis application

DMONDIAGME.DMONDIAGME is initialized to “0” by Power On Reset, System Reset 1/2, DeepSTOP Reset or DMON Reset (RESDMON = 1). So DMON diagnosis function is enabled at start-up after these resets.

The user’s application controls the diagnostic function by DMONDMASK and DMONDIAG. **Figure 11.20** shows the example of the diagnosis application.

After diagnosis application was over, write “1” to DMONDIAGMEW.DMONDIAGMEW and disable DMON diagnosis function immediately.

This function protects each flag against erroneous operation when each power supply is out of the operating range.



Note 1. The diagnosis test wait time is changed by the filter setting. It is shown in **Table 11.46.**

Figure 11.20 The Example of the Flowchart of DMON Diagnosis Application

Table 11.46 The Wait Time of Diagnosis Test

DMONFCR.DMONFLTW	Wait time
000_B	3 sampling times
001_B	4 sampling times
010_B	6 sampling times
011_B	10 sampling times
100_B	18 sampling times
101_B	34 sampling times
110_B	66 sampling times
111_B	130 sampling times

CAUTION

A sampling period is 5 µs(16 MHz), 4 µs(20/40 MHz), 3.33 µs(24 MHz)

11.4.7.3 Function in Standby

It is necessary to stop the DMON operation before stopping the MainOSC by resetting DMON.DMON can be reset when DMONTEST.RESDMON is set to “1”.

11.4.7.4 Procedures to Reset DMON

DMONTEST.RESDMON can reset Delay Monitor.

[Procedure]

Example case:

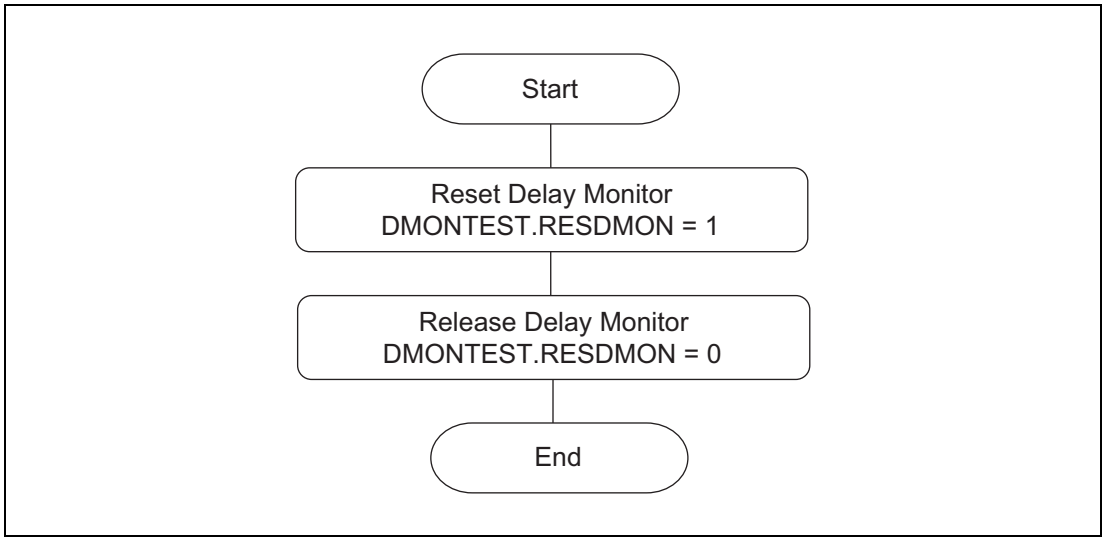


Figure 11.21 Flow Chart to Reset DMON

11.5 RAM Retention Voltage Indicator (Very Low Voltage Indicator, VLVI)

11.5.1 Features

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage SYSVCC with the RAM retention voltage V_{VLVI}.

See **Section 55, Electrical Characteristics** for the specification of the RAM retention voltage level V_{VLVI}.

11.5.2 Clock Supply

The clock supply to the VLVI is shown in the following table.

Table 11.47 Clock Supply

Module Name	Module Clock Name	Supply Clock Name
VLVI	Register access clock	CLK_LSB

11.5.3 Interrupt Requests and Error Notifications

This module has no interrupt and sDMA / DTS requests.

This module has no error notifications to ECM.

11.5.4 External Input/Outputs

This module has no external input/output pins.

11.5.5 Registers

11.5.5.1 List of Registers

The register list related to VLVI is shown in **Table 11.48**.

Table 11.48 List of Registers (VLVI)

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 3300$_H$	VLVF	Very-Low-Voltage Detection Register	8	01$_H$	PBG20 #2	
FF98 3308$_H$	VLVFC	Very-Low-Voltage Detection Clear Register	8	00$_H$	PBG20 #2	VLVIKCPROT
FF98 3F40$_H$	VLVIKCPROT	VLVI Register Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	

Register reset conditions are shown in **Table 11.49**.

Table 11.49 Register Reset Conditions for VLVI

Register Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
VLVF	—	—	—	—	—	—	—
VLVFC	√	√	—	—	—	—	—
VLVIKCPROT	√	√	√	—	√	—	—

11.5.5.2 VLVF — Very-Low-Voltage Detection Register

The very-low-voltage detection register (VLVF) shows the state of the RAM retention voltage detection.

This register is set upon detection of a voltage below the RAM retention voltage (V_{VLVI}).

If VLVF is set, the retention RAM content cannot be guaranteed.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FF98 3300$_H$

Value after reset: 01$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VLVF
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 11.50 VLVF Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	VLVF	Very-Low-Voltage Detection Flag 0: Very-low-voltage is not detected. 1: Very-low-voltage is detected.

11.5.5.3 VLVFC— Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

Access: This register is a write-only register that can be written in 8-bit units.

Address: FF98 3308$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VLVFC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 11.51 VLVFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

11.5.5.4 VLVIKCPROT — VLVI Register Key Code Protection Register

The VLVIKCPROT register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units

Address: FF98 3F40$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 11.52 VLVIKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0 *1.
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writes to protected registers.
Write A5A5A501$_H$ to this register to enable writes to protected registers.

11.5.6 Operation

If the power supply voltage SYSVCC does not fall below V_{VLVI}, the content of the retention RAM (RRAM) is retained. See **Section 9.5.2, Reset Sources**.

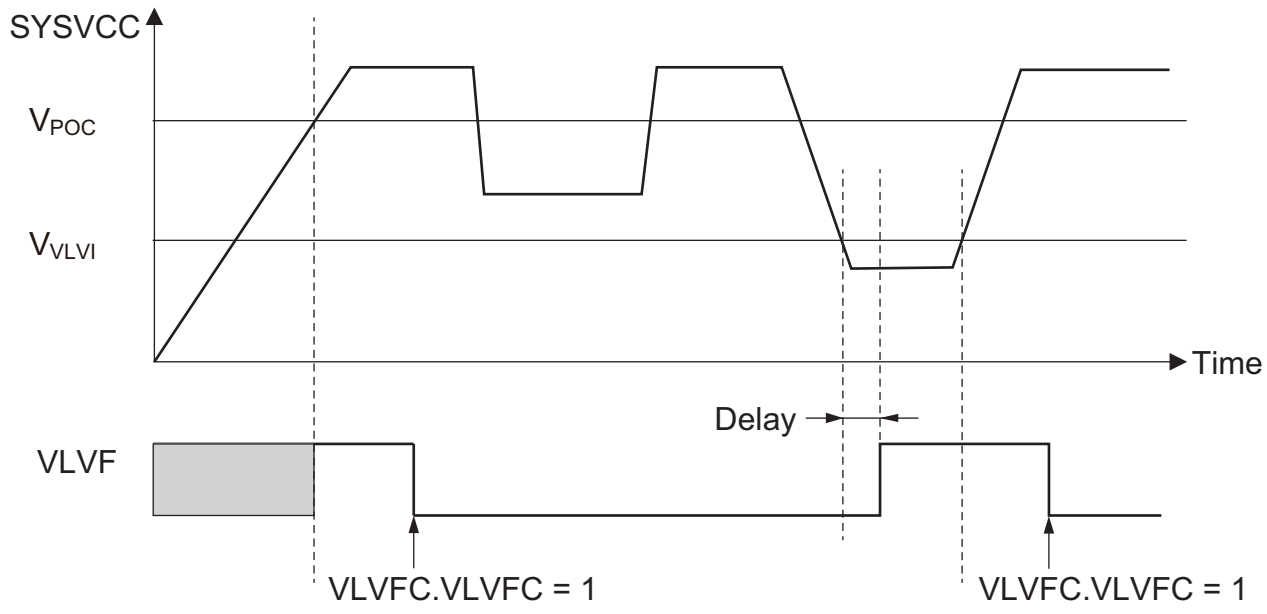
If SYSVCC falls below V_{VLVI}, the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

If SYSVCC falls below the RAM retention voltage (SYSVCC < V_{VLVI}), the VLVF.VLVF bit is set.

After that, even if SYSVCC exceeds V_{VLVI}, the VLVF.VLVF bit is not cleared automatically. It is cleared by

- setting VLVFC.VLVFC bit to 1.

The following **Figure 11.22** illustrates the timing of VLVF.



Note: Delay:
When SYSVCC falls below V_{VLVI}, the VLVF bit is set to 1 after a certain delay.
See **Section 55, Electrical Characteristics** for delay time.

Figure 11.22 VLVF Operation Timing

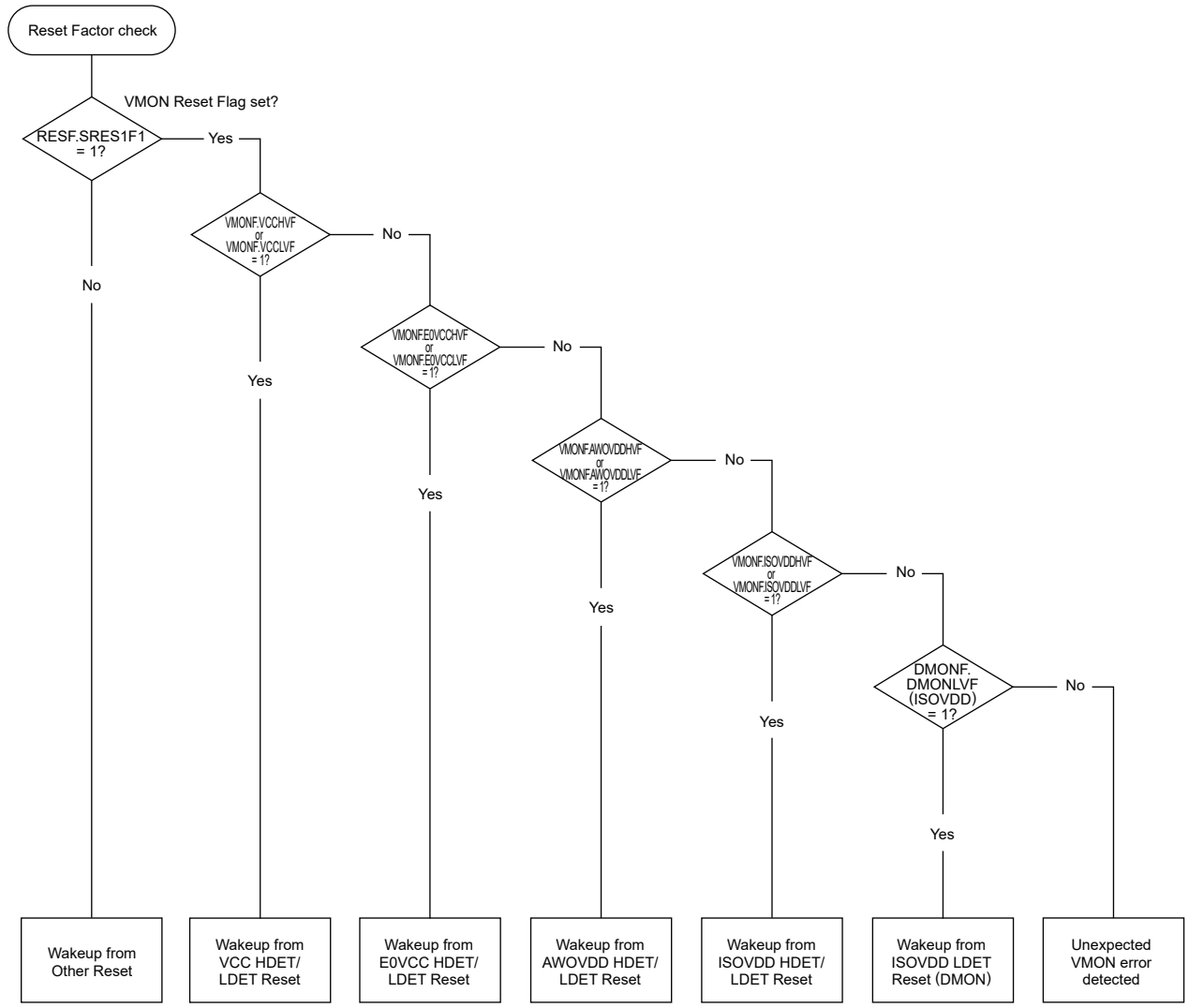
11.6 Usage Notes

11.6.1 Judgment Method of the Reset Factors for Voltage Monitor

If RESF.SRES1F1 bit is “1”, it can be known that wake-up from VMON Reset occurs.

Check the VMONF and DMONF for the detection power.

Judgment method of the reset factors is shown in **Figure 11.23**.



Note: If high voltage or low voltage of multiple power supplies is detected, the corresponding flag is set.

Figure 11.23 Flowchart of Reset Factor Result

Section 12 Temperature Sensor (OTS)

12.1 Features of Temperature Sensor for RH850/U2A-EVA

12.1.1 Number of Channels

Table 12.1 Number of Channels

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)	RH850/U2A6 (292 pins)	RH850/U2A6 (176 pins)	RH850/U2A6 (156 pins)	RH850/U2A6 (144 pins)
Number of Channels	1	1	1	1	1	1	1	1	1	1
Name	OTS0	OTS0	OTS0	OTS0	OTS0	OTS0	OTS0	OTS0	OTS0	OTS0

12.1.2 Register Base Addresses

Table 12.2 Register base addresses

Base Address Name	Base Address	Bus Group
<OTS0_base>	FFC0 0200_H	P-Bus Group 6L

12.1.3 Clock Supply

The clocks supplied to the temperature sensor are listed in the following table.

Table 12.3 Clock Supply

Unit Name	Unit Clock Name	Clock Name
OTS0	Operation clock	CLK_LSB
	Register access clock	CLK_LSB

12.1.4 Interrupt Requests and Error Notifications

The temperature sensor interrupt requests are listed in the following table.

Table 12.4 Interrupt Requests

Unit Interrupt Name	Unit Interrupt Signal Outline	Interrupt Number
INTOTSOTE	Temperature sensor error interrupt	657
INTOTSOTI	Temperature measurement end interrupt	658
INTOTSOTULI	Triggered if the state machine changes the state due to a temperature rise or drop in the guaranteed temperature range	659

The temperature sensor error notifications are listed in the following table.

Table 12.5 Error Notifications

Error Name	Description	Error number	Error response
Temperature sensor error	Abnormal temperature error (OTABE)	76	—

12.1.5 Reset Sources

The registers that constitute the temperature sensor are initialized by the reset sources listed below.

Table 12.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
OTS0	All registers	√	√	√	√	√	—	—

12.1.6 External Input/Output Pins

This module has no external input/output pin.

12.2 Overview

12.2.1 Functional Overview

The following describes the features of the temperature sensor.

- Temperature data register
  A temperature data register stores a temperature measurement value.
- Temperature measurement mode
  Single measurement mode: Used to measure temperature only once.
  Continuous measurement mode: Used to measure temperature continuously.
- Supporting temperature measurement end interrupt
  Each time temperature measurement ends, the temperature sensor can generate an interrupt request (INTOTSOTI) to be sent to the INTC.
- Supporting abnormal temperature error signal and temperature rise/drop interrupt
  Six temperature threshold values (high-temperature border AU > high-temperature border AL > high-temperature border BU > high-temperature border BL > low-temperature border AU > low-temperature border AL) should be set in advance. Furthermore, the temperature sensor has four temperature states (high temperature A, high temperature B, normal temperature, and low temperature A). The temperature state is updated at each temperature measurement according to six temperature threshold values. The following describes the conditions for generating an abnormal temperature error signal and a temperature rise/drop interrupt.
  - An abnormal temperature error signal (OTABE) is output at a transition from a state other than high temperature A to high temperature A or at a transition from a state other than low temperature A to low temperature A.
  - A temperature rise/drop interrupt (INTOTSOTULI) is output at a transition from high temperature A to high temperature B or normal temperature, or at a transition from high temperature B to normal temperature, that is, when the temperature drops.
  - A temperature rise/drop interrupt (INTOTSOTULI) is output at a transition from low temperature A to high temperature B or normal temperature, or at a transition from normal temperature to high temperature B, that is, when the temperature rises.

Temperature status can be monitored by reading the temperature status register.

- Diagnosis function
  When temperature is measured in continuous measurement mode, if the difference between the measured value and the previously measured value is larger than the specified temperature difference limit value, a temperature sensor error interrupt (INTOTSOTE) is output.
- Reducing errors by temperature correction
  Three temperature correction coefficients are stored during device manufacturing in coefficient registers A to C.

12.2.2 Block Diagram

Figure 12.1 shows a block diagram of the temperature sensor.

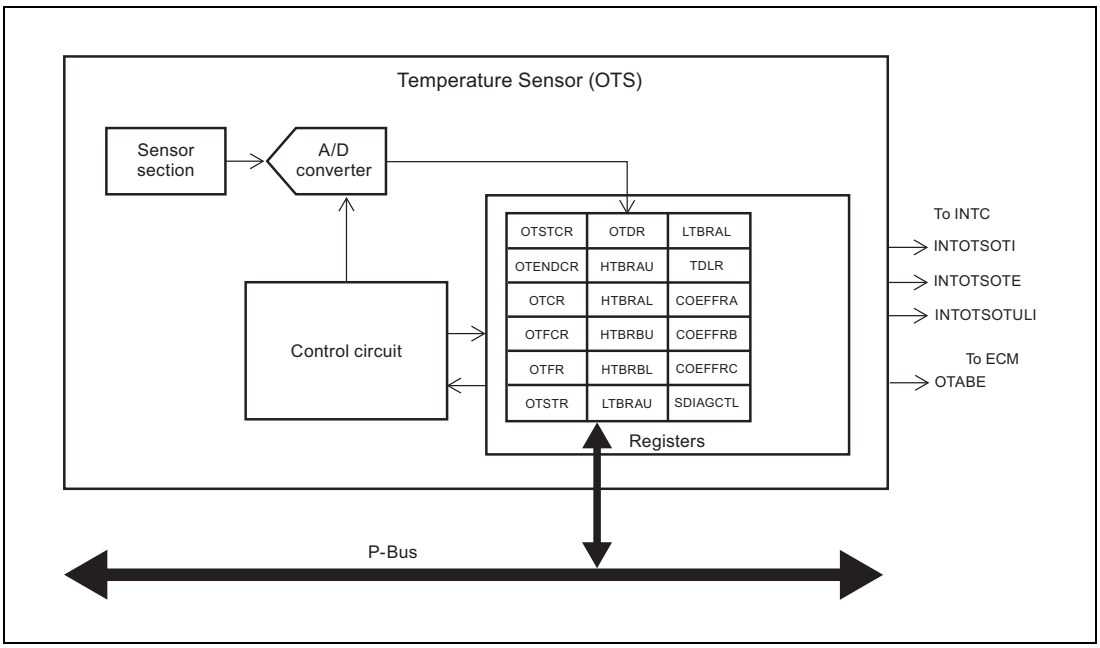


Figure 12.1 Temperature Sensor Block Diagram

12.3 Register

12.3.1 List of Registers

The following table lists the registers of the temperature sensor.

<OTS0_base> is defined in **Section 12.1.2, Register Base Addresses**.

Table 12.7 List of Registers

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
OTS0	Temperature measurement start control register	OTS0OTSTCR	<OTS0_base> +00_H	8	PBG6L0#2	—
OTS0	Temperature measurement end control register	OTS0OTENDCR	<OTS0_base> +04_H	8	PBG6L0#2	—
OTS0	Temperature sensor control register	OTS0OTCR	<OTS0_base> +08_H	8	PBG6L0#2	—
OTS0	Temperature sensor flag clear register	OTS0OTFCR	<OTS0_base> +$0C_H$	8	PBG6L0#2	—
OTS0	Temperature sensor flag register	OTS0OTFR	<OTS0_base> +10_H	8	PBG6L0#2	—
OTS0	Temperature status register	OTS0OTSTR	<OTS0_base> +14_H	8	PBG6L0#2	—
OTS0	Temperature data register	OTS0OTDR	<OTS0_base> +18_H	16	PBG6L0#2	—
OTS0	High-temperature border AU register	OTS0HTBRAU	<OTS0_base> +$1C_H$	16	PBG6L0#2	—
OTS0	High-temperature border AL register	OTS0HTBRAL	<OTS0_base> +20_H	16	PBG6L0#2	—
OTS0	High-temperature border BU register	OTS0HTBRBU	<OTS0_base> +24_H	16	PBG6L0#2	—
OTS0	High-temperature border BL register	OTS0HTBRBL	<OTS0_base> +28_H	16	PBG6L0#2	—
OTS0	Low-temperature border AU register	OTS0LTBRAU	<OTS0_base> +$2C_H$	16	PBG6L0#2	—
OTS0	Low-temperature border AL register	OTS0LTBRAL	<OTS0_base> +30_H	16	PBG6L0#2	—
OTS0	Temperature difference limiting register	OTS0TDLR	<OTS0_base> +34_H	16	PBG6L0#2	—
OTS0	Coefficient A register	OTS0COEFFRA	<OTS0_base> +38_H	16	PBG6L0#2	—
OTS0	Coefficient B register	OTS0COEFFRB	<OTS0_base> +$3C_H$	16	PBG6L0#2	—
OTS0	Coefficient C register	OTS0COEFFRC	<OTS0_base> +40_H	16	PBG6L0#2	—
OTS0	Self-diagnosis control register	OTS0SDIAGCTL	<OTS0_base> +$A0_H$	8	PBG6L0#2	—

12.3.2 OTS0OTSTCR — Temperature Measurement Start Control Register

OTS0OTSTCR is an 8-bit write-only control register to start the temperature sensor.

Access: This register can be written in 8-bit units.

Address: <OTS0_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OTST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 12.8 OTS0OTSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OTST	Temperature Measurement Start Condition for starting temperature measurement by OTST. Write 1 to OTST when OTACT = 0

12.3.3 OTS0OTENDCR — Temperature Measurement End Control Register

OTS0OTENDCR is an 8-bit write-only control register to terminate the temperature sensor.

Access: This register can be written in 8-bit units.

Address: <OTS0_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OTEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 12.9 OTS0OTENDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OTEND	Temperature Measurement End Condition for terminating temperature measurement by OTEND. Write 1 to OTEND and then wait for 2 cycles of the OTS clock.

12.3.4 OTS0OTCR — Temperature Sensor Control Register

OTS0OTCR is an 8-bit readable/writable register to control the temperature sensor.

Access: This register can be read or written in 8-bit units.

Address: <OTS0_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	OTEE	OTULIE	OTABEE	SDE	OTMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 12.10 OTS0OTCR Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	OTEE	Temperature Sensor Error Enable 0: Disabled 1: Enabled When OTMD = 0 (single measurement mode) or SDE = 0 (diagnosis disabled), no temperature sensor error is generated regardless of the OTEE setting.
3	OTULIE	Temperature Rise/Drop Interrupt Enable 0: Disabled 1: Enabled
2	OTABEE	Temperature Alarm Error Enable 0: Disabled 1: Enabled
1	SDE	Diagnosis Enable 0: Disabled 1: Enabled When OTMD = 0 (single measurement mode), diagnosis is not performed regardless of the SDE setting.
0	OTMD	Temperature Measurement Mode 0: Single temperature measurement mode 1: Continuous temperature measurement mode

CAUTION

To prevent malfunction, set OTS0OTCR while the OTACT bit in OTS0OTFR is 0.

12.3.5 OTS0OTFCR — Temperature Sensor Flag Clear Register

OTS0OTFCR is an 8-bit write-only register to clear the OTS0OTFR register.

Access: This register can be written in 8-bit units.

Address: <OTS0_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SDERC	—	OTFC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	W	R	W

Table 12.11 OTS0OTFCR Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When writing, write the value after reset.
2	SDERC	Diagnosis Error Clear Writing 0: Does not clear the diagnosis error. Writing 1: Clears the diagnosis error.
1	Reserved	When writing, write the value after reset.
0	OTFC	Temperature Measurement End Flag Clear Writing 0: Does not clear the temperature measurement end flag. Writing 1: Clears the temperature measurement end flag.

12.3.6 OTS0OTFR — Temperature Sensor Flag Register

OTS0OTFR is an 8-bit read-only register to indicate temperature sensor flags.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <OTS0_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SDER	OTACT	OTF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.12 OTS0OTFR Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	SDER	Diagnosis Error Setting condition A diagnosis error occurred. Clearing condition Write SDERC to 1 in OTS0OTFCR
1	OTACT	Temperature Sensor Status 0: The temperature sensor is in the idle state. 1: The temperature sensor is operating.
0	OTF	Temperature Measurement End Flag Setting condition A temperature measurement value is written to OTS0OTDR. Clearing condition Write OTFC to 1 in OTS0OTFCR. Reading of the OTS0OTDR register.

CAUTION

SDER is updated when a temperature measurement value is written to OTS0OTDR.

12.3.7 OTS0OTSTR — Temperature Status Register

OTS0OTSTR is an 8-bit read-only register to indicate temperature sensor status.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <OTS0_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSTAT[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 12.13 OTS0OTSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1, 0	TSTAT[1:0]	Temperature Status 00: Normal temperature (value after reset) 01: High temperature B 10: High temperature A 11: Low temperature A When temperature is measured, temperature status is determined by the temperature measurement value and the status is reflected in TSTAT[1:0]. **Figure 12.2** shows the status transitions of TSTAT. **Table 12.14** shows the temperature alarm error and temperature rise/drop interrupt generating conditions.

CAUTION

The TSTAT[1:0] bits are updated when a temperature measurement value is written to OTS0OTDR.

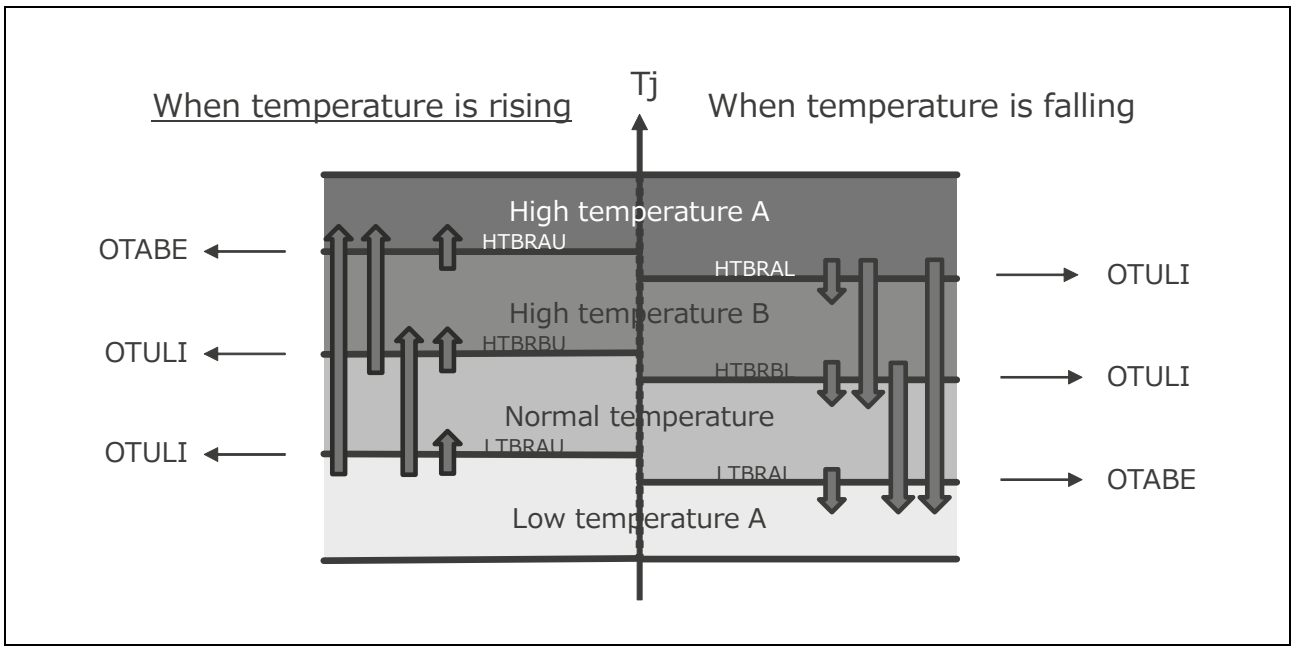


Figure 12.2 Temperature Status and Transition States

Table 12.14 Conditions for Generating Abnormal Temperature Error and Temperature Rise/Drop Interrupts

		After Status Transition			
		High temp. A	**High temp. B**	**Normal temp.**	**Low temp. A**
Before status transition	High temp. A	N/A	INTOTSOTULI	INTOTSOTULI	OTABE
	High temp. B	OTABE	N/A	INTOTSOTULI	OTABE
	Normal temp.	OTABE	INTOTSOTULI	N/A	OTABE
	Low temp. A	OTABE	INTOTSOTULI	INTOTSOTULI	N/A

[Legend]

OTABE: Occurrence of Abnormal temperature error

INTOTSOTULI: Occurrence of temperature rise or drop interrupt

N/A: Neither OTABE nor INTOTSOTULI occurs.

12.3.8 OTS0OTDR — Temperature Data Register

OTS0OTDR is a 16-bit read-only register that stores a temperature measurement value.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <OTS0_base> + 18_H

Value after reset: 0000_H

Signed fixed-point format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OTD*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

↑

Decimal point position

Note 1. OTD[15]: Sign bit (always 0)

Table 12.15 OTS0OTDR Register Contents

Bit Position	Bit Name	Function
15 to 0	OTD	Temperature Data Value These bits specify a value for temperature data. OTD[15] is always 0. The temperature can be calculated by the following formula. $T = AX^2 + BX + C$ – T[K]: Temperature value after correction – X: OTS0OTDR – A to C: Correction coefficients A to C Refer to **Section 12.3.12, OTS0COEFFRn — Coefficient n Register (n = A, B, C)**.

NOTE

When the CPU reads the register while the temperature sensor module is updating the register, the value before updating will be read. **Figure 12.3, Example of Temperature Measurement Operation (Single Measurement Mode)** shows the updating of this register.

12.3.9 OTS0HTBRmn — High-Temperature Border mn Register (m = A or B, n = U or L)

OTS0HTBRmn is a 16-bit readable/writable register to specify the temperature border of high temperature m.

Access: This register can be read or written in 16-bit units.

Address: OTS0HTBRAU: <OTS0_base> + $1C_H$, OTS0HTBRAL: <OTS0_base> + 20_H
OTS0HTBRBU: <OTS0_base> + 24_H, OTS0HTBRBL: <OTS0_base> + 28_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HTBmn*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. HTBmn[15]: Sign bit (always 0)

Table 12.16 OTS0HTBRmn Register Contents

Bit Position	Bit Name	Function
15 to 0	HTBmn	High-Temperature Border mn These bits specify the high-temperature border mn. Specify it so that the following condition is met. OTS0HTBRAU > OTS0HTBRAL > OTS0HTBRBU > OTS0HTBRBL > OTS0LTBRAU > OTS0LTBRAL The HTBmn format is the same as the OTS0OTDR format. HTBmn[15] is always 0. For details, see **Section 12.3.7, OTS0OTSTR — Temperature Status Register**.

CAUTION

To prevent malfunction, set OTS0HTBRmn while the OTACT bit in OTS0OTFR is 0.

The new calculation of the border registers has to be done according to the reverse quadratic equation of the temperature register

$T = AX^2 + BX + C$

- [K]: Temperature value after correction
- X: OTS0OTDR
- A to C: Correction coefficients A to C

This leads to the formula:

$$X = \left(\frac{-B + \sqrt{B^2 - 4A \times (C - T)}}{2A} \right)$$

A, B, C = Correction coefficients A to C

(= decimalpoint - corrected raw value of CoeffRegisters (= raw value shifted right by 4)

T = Temperature [in K]

X = BorderRegister

Example:

- OTS0COEFFRA = 2106_H, OTS0COEFFRB = $2D87_H$, OTS0COEFFRC = 0040_H
- Temperature Border = 153°Celsius

→

- A = OTS0COEFFRA/(2^4) = 528.375
- B = OTS0COEFFRB/(2^4) = 728.4375
- C = OTS0COEFFRC/(2^4) = 4

$$X = \frac{-728.4 + \sqrt{728.4^2 - 4 \times 528.4 \times (4 - (153 + 273.15))}}{2 \times 528.4} = 0.43942$$

This value has to be decimal point corrected = multiplied with 2^{15}.

Calculated High Temperature Border Register:

OTS0HTBRmn = 0.43942×2^{15} = 14399

12.3.10 OTS0LTBRAn — Low-Temperature Border An Register (n = U or L)

OTS0LTBRAn is a 16-bit readable/writable register to specify the temperature border of low temperature A.

Access: This register can be read or written in 16-bit units.

Address: OTS0LTBRAU: <OTS0_base> + $2C_H$, OTS0LTBRAL: <OTS0_base> + 30_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTBAn*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. LTBAn[15]: Sign bit (always 0)

Table 12.17 OTS0LTBRAn Register Contents

Bit Position	Bit Name	Function
15 to 0	LTBAn	Low-Temperature Border An These bits specify the low-temperature border An. Specify it so that the following condition is met. OTS0HTBRAU > OTS0HTBRAL > OTS0HTBRBU > OTS0HTBRBL > OTS0LTBRAU > OTS0LTBRAL The LTBAn format is the same as the OTS0OTDR format. LTBAn[15] is always 0. For details, see **Section 12.3.7, OTS0OTSTR — Temperature Status Register**.

CAUTION

To prevent malfunction, set OTS0LTBRAn while the OTACT bit in OTS0OTFR is 0.

12.3.11 OTS0TDLR — Temperature Difference Limiting Register

OTS0TDLR is a 16-bit readable/writable register to specify the temperature difference limit value for diagnosis.

Access: This register can be read or written in 16-bit units.

Address: <OTS0_base> + 34_H

Value after reset: $7FFF_H$

Signed fixed-point format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDL*1															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

↑
Decimal point position

Note 1. TDL[15]: Sign bit (always 0)

Table 12.18 OTS0TDLR Register Contents

Bit Position	Bit Name	Function
15 to 0	TDL	Temperature Difference Limit Value These bits specify a value for limiting the temperature difference between present temperature measurement value and previous temperature measurement value in continuous measurement mode. When the following condition is met, SDER (diagnosis error) is set to 1. OTS0TDLR < | present temperature measurement value – previous temperature measurement value | To calculate from the temperature, calculate back from the formula described in the OTS0COEFFRn register ($T[K] = AX^2 + BX + C$). For details, see **Section 12.3.12, OTS0COEFFRn — Coefficient n Register (n = A, B, C)**.The TDL format is the same as the OTS0OTDR format. TDL[15] is always 0. For details, see **12.3.7, OTS0OTSTR — Temperature Status Register**.

CAUTION

To prevent malfunction, set OTS0TDLR while the OTACT bit in OTS0OTFR is 0.

12.3.12 OTS0COEFFRn — Coefficient n Register (n = A, B, C)

OTS0COEFFRn is a 16-bit read-only register to store coefficients for correction calculation.

Coefficients are stored in OTS0COEFFRn during device manufacturing.

Access: This register is a read-only register that can be read in 16-bit units.

Address: OTS0COEFFRA: <OTS0_base> + 38_H, OTS0COEFFRB: <OTS0_base> + $3C_H$, OTS0COEFFRC: <OTS0_base> + 40_H

Value after reset: The initial value will be stored at delivery inspection for each devices

Signed fixed-point format

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COEFF[15:0]*1															
Value after reset	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

↑
Decimal point position

Note 1. COEFF[15]: Sign bit

Table 12.19 OTS0COEFFRn Register Contents

Bit Position	Bit Name	Function
15 to 0	COEFF[15:0]	Coefficients Value for correction calculation These bits specify a value for coefficients Value.

- Reducing errors by temperature correction
  Three temperature correction coefficients are stored during device manufacturing in coefficient registers A to C.
  Errors in temperature measurement values can be reduced by making a correction by using the temperature correction calculation formula that uses the temperature correction coefficients. However, temperature corrections must be made by the CPU. By using this process, the Temperature Border setting can be improved.
- Correction calculation formula: $T[K] = AX^2 + BX + C$
  − T: Temperature value after correction
  − X: Temperature value before correction (OTS0OTDR)
  − A to C: Coefficients A to C (OTS0COEFFRn n = A, B, C)

  For example:
  $OTS0OTDR = 36DA_H$, $OTS0COEFFRA = 2106_H$, $OTS0COEFFRB = 2D87_H$, $OTS0COEFFC = 0040_H$

  $X = OTS0OTDR / (2^{15}) = 0.428528$

  $A = OTS0COEFFRA / (2^4) = 528.375$

  $B = OTS0COEFFRB / (2^4) = 728.4375$

  $C = OTS0COEFFRC / (2^4) = 4$

  $T[°C] = T[K] – 273.15 = AX^2 + BX + C – 273.15 = 140.035$ °C

12.3.13 OTS0SDIAGCTL — Self-Diagnosis Control Register

OTS0SDIAGCTL is an 8-bit readable/writable register to control the input of AD converter for self-diagnosis.

Access: This register can be read or written in 8-bit units.

Address: <OTS0_base> + $A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADCIN[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 12.20 OTS0SDIAGCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ADCIN[1:0]	AD Converter Input Control for Self-Diagnosis. 00: No pull-up/pull-down (normal operation) 01: Pull-down 10: Pull-up 11: Setting prohibited

12.4 Operation

12.4.1 Temperature Measurement Sequence

The temperature sensor is equipped with a Single measurement mode and a Continuous measurement mode.

The measurement start trigger and measurement end trigger of each mode is shown below.

Table 12.21 Measurement Start/End Trigger

Measurement Mode	Measurement Start Trigger	Measurement End Trigger
Single measurement mode OTS0OTCR.OTMD = 0	Write 1_B in OTS0OTSTCR.OTST.	1) Automatic stop after one temperature measurement ends 2) Write 1_B in OTS0OTENDCR.OTEND.*1*2
Continuous measurement mode OTS0OTCR.OTMD = 1	Write 1_B in OTS0OTSTCR.OTST.	Write 1_B in OTS0OTENDCR.OTEND

Note 1. If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND, data during the measurement will be discarded.

Note 2. If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND before the first measurement has been completed, the measurement operation will not output the measurement result at all.

The time until the temperature data is output requires 512 states (9.984 ms) of the OTS clock to convert the measured temperature to digital data.

There is a measurement stabilization period of 3 OTS clock states before the temperature is sampled after the measurement process is started by writing to the OTS0OTSSTCR.OTST bit.

In Single measurement mode, to update the temperature measurement result requires 515 OTS clock states, including the measurement preparation period, until the temperature measurement operation ends.

In Continuous measurement mode, 515 of the OTS clock states, including the 3-state measurement stabilization period, are required only for the first temperature measurement, the second and subsequent updates of the temperature measurement result require 512 states, and the measurement operation is repeated until 1 is written to OTENDCR.OTEND.

12.4.2 Examples of Temperature Measurement Operation

The following figure shows the temperature measurement operation in single measurement mode.

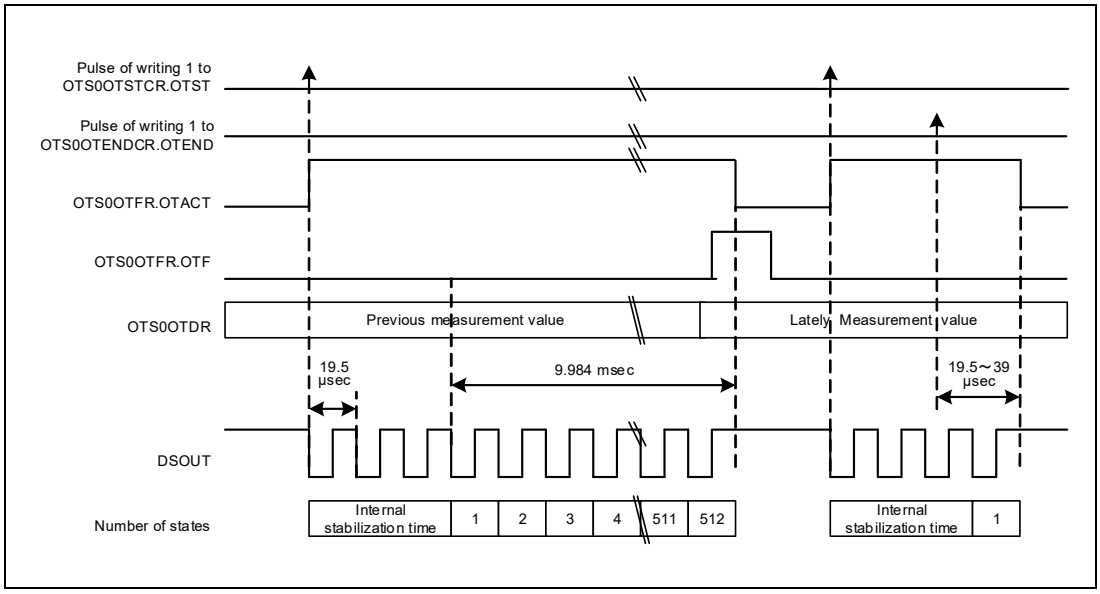


Figure 12.3 Example of Temperature Measurement Operation (Single Measurement Mode)

NOTE

If “1” is written to OTENDCR.OTEND at the time of internal cycle 511 or 512, request INTOTSOTI is issued and OTS0OTFR.OTF is set after updating the temperature measurement result as with the case of automatic stop after the temperature measurement.

DSOUT is the operation clock of the temperature sensor obtained by dividing CLK_LSB.

The period of DSOUT is 19.5 µs (When CLK_LSB 40 MHz)

The following figure shows temperature measurement operation in continuous measurement mode. There is no internal stabilization time in the second and subsequent temperature measurements.

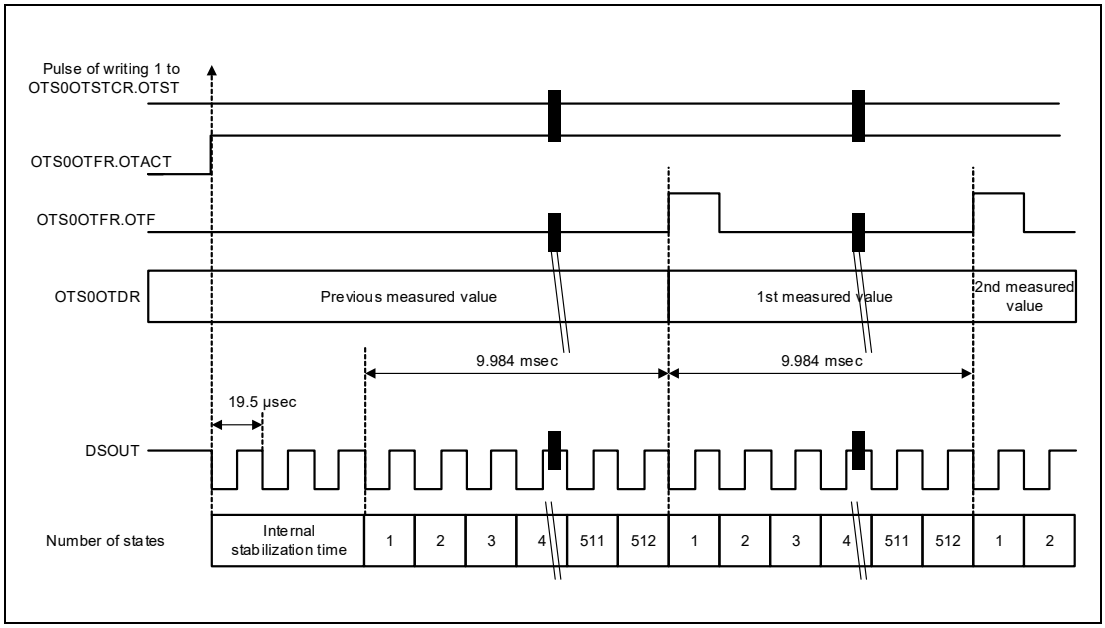


Figure 12.4 Example of Temperature Measurement Operation (Continuous Measurement Mode)

12.4.3 Temperature Measurement End Interrupt Request

The temperature sensor can generate a temperature measurement end interrupt request (INTOTSOTI) to be sent to the INTC. At this time, the OTS0OTFR.OTF is set.

The table below shows the assertion condition and negated conditions of INTOTSOTI and OTS0OTFR.OTF.

Table 12.22 Temperature Measurement End Conditions

Measurement End Notification Signal	Assert Conditions	Negated Conditions	Enable Control
Temperature measurement end interrupt (INTOTSOTI)	Each time temperature measurement ends	After the assertion, one cycle after the register access clock	No enable control bit.
Temperature measurement end flag (OTS0OTFR.OTF)	Each time temperature measurement ends	1) Reading of the OTS0OTDR register 2) Write 1_B in OTS0OTFCR.OTFC	No enable control bit.

For the OTS0OTFR.OTF bit, if the assertion conditions and negated conditions conflict, negated condition takes precedence.

If the temperature measurement has been stopped by the OTS0OTENDCR.OTEND, neither the INTOTSOTI nor the OTS0OTFR.OTF will occur.

However, after writing "1" to the OTS0OTENDCR.OTEND, when the measurement is within 2 OTS clock cycles of being completed, the measurement result is updated and the INTOTSOTI and OTS0OTFR.OTF are asserted.

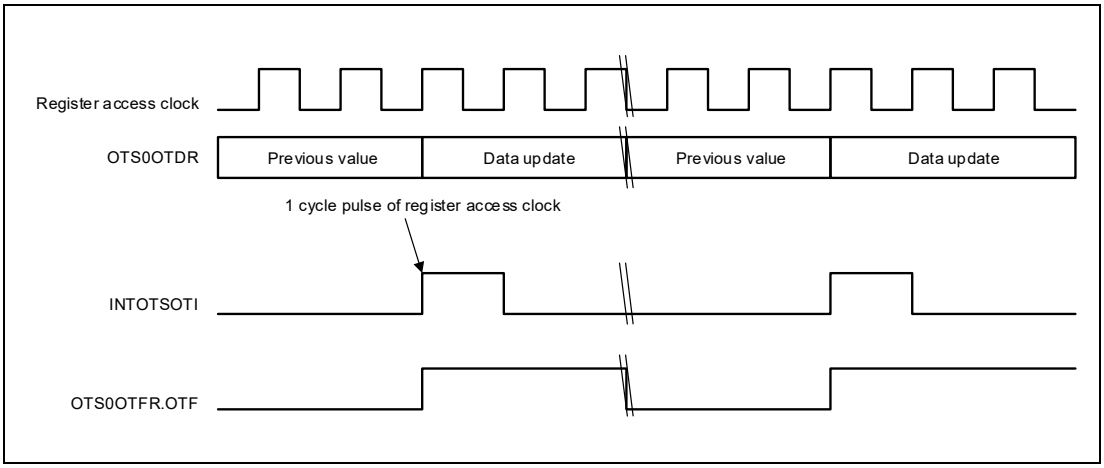


Figure 12.5 Example of Occurrence of a Temperature Measurement End Interrupt

12.4.4 Temperature Alarm Error and Temperature Rise /Drop Interrupt and Temperature Sensor Error Interrupt Requests

The temperature sensor can generate a temperature rise or drop interrupt request (INTOTSOTULI), a temperature sensor error interrupt request (INTOTSOTE) and an abnormal temperature error (OTABE) to be sent to ECM. Setting OTULIE in OTS0OTCR to 1 enables INTOTSOTULI. Setting OTULIE to 0 disables INTOTSOTULI. Setting OTEE in OTS0OTCR to 1 enables INTOTSOTE. Setting OTEE to 0 disables INTOTSOTE. Setting OTABEE in OTS0OTCR to 1 enables OTABE. Setting OTABEE to 0 disables OTABE.

The table below shows the assert conditions and negated conditions of INTOTSOTULI and OTABE.

Table 12.23 Temperature Sensor Interrupts

Temperature State Transition Notification Signal	Assert Conditions	Negated Conditions	Enable Control
Temperature rise or drop interrupt (INTOTSOTULI)	Temperature state transition from a state other than the high temperature B to high temperature B. Temperature state transition from a state other than the normal temperature to normal temperature.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTULIE
Abnormal temperature error (OTABE)	Temperature state transition from a state other than the high temperature A to high temperature A. Temperature state transition from a state other than the low temperature A to low temperature A.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTABEE
Temperature sensor error interrupt (INTOTSOTE)	After the OTDR register is updated, when the difference between the temperature measurements exceeds a set value TDL.	After the assertion, one cycle after the register access clock	OTS0OTCR.OTEE and OTS0OTCR.SDE
Diagnostic error flag (OTS0OTFR.SDER)	After the OTDR register is updated, when the difference between the temperature measurements exceeds a set value TDL.	Write 1B in OTS0OTFCR.SDERC	OTS0OTCR.SDE

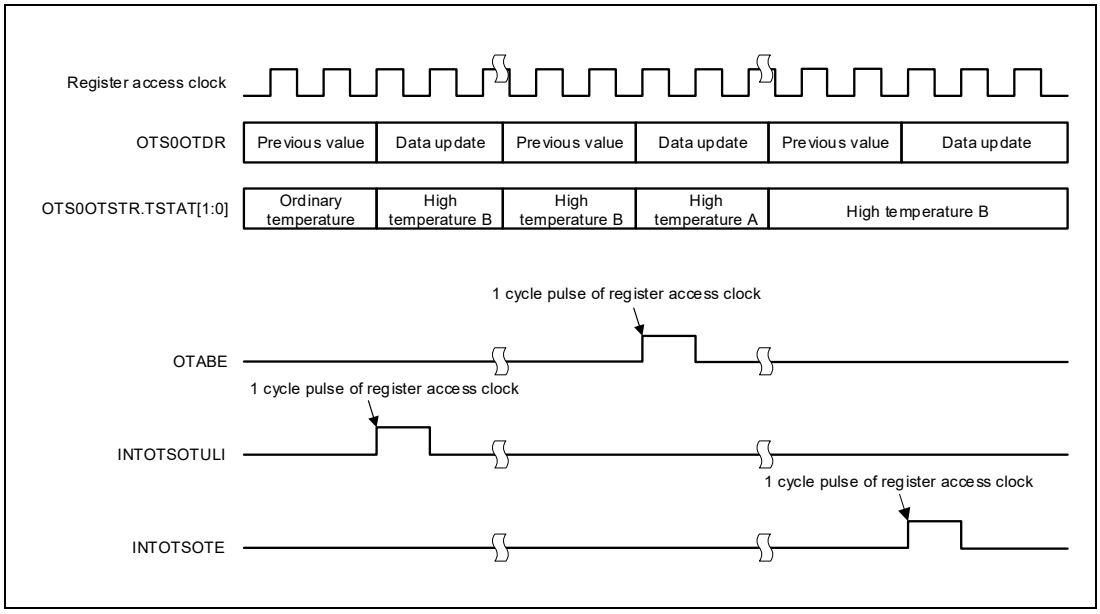


Figure 12.6 Example of Occurrence of a Temperature Alarm Error, a Temperature Rise/Drop Interrupt and a Temperature Sensor Error Interrupt

12.4.5 Self-Diagnosis Sequence

Self-diagnosis of the temperature sensor can be performed by the sequence shown in **Figure 12.7**.

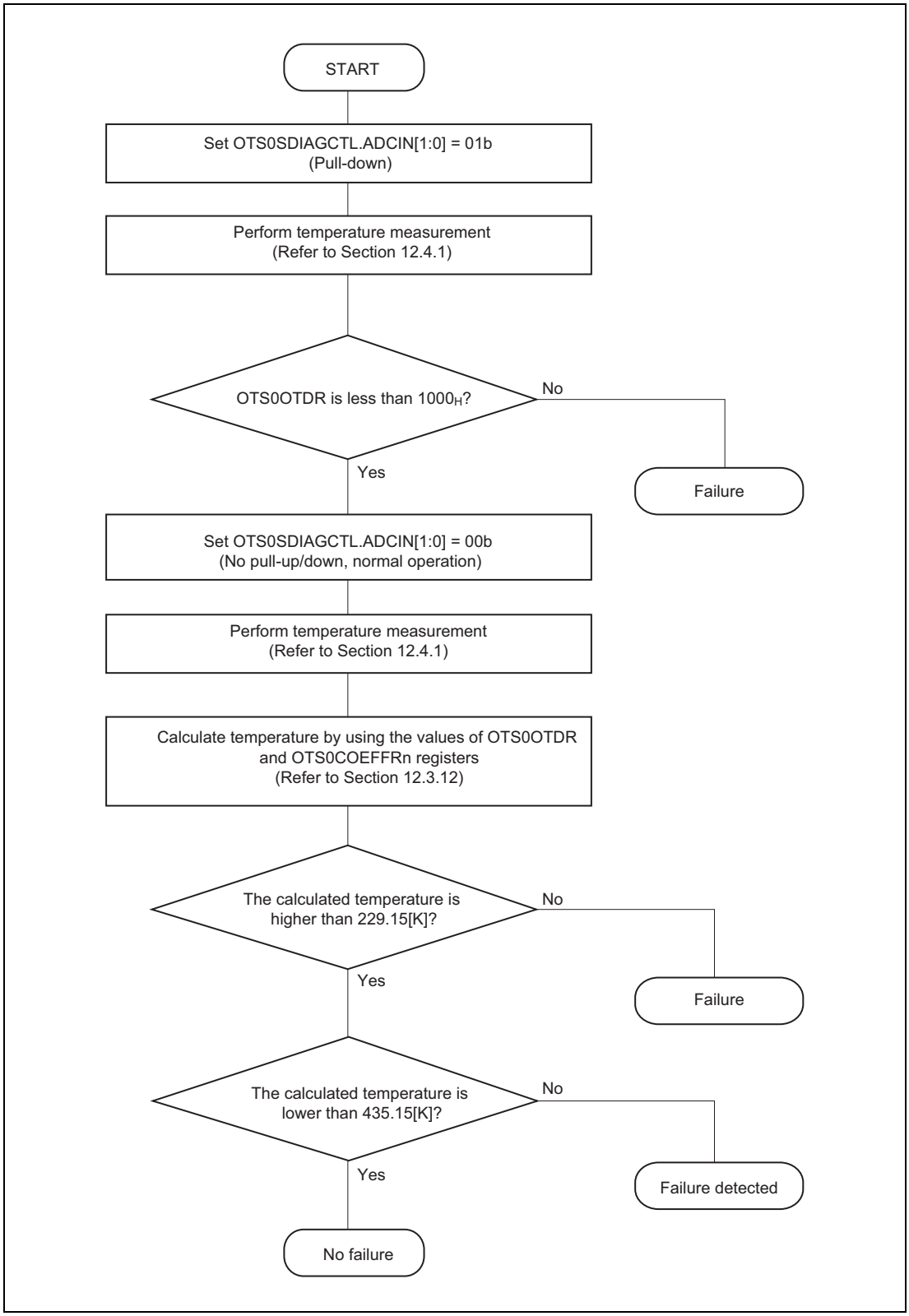


Figure 12.7 Self-Diagnosis Sequence

Section 13 Clock Controller

This section contains a generic description of the clock controller.

The first part in this section describes the specific features of the clock controller of the RH850/U2A-EVA microcontrollers. The ensuing sections describe the clock oscillation circuit, clock selectors, and clock output function that make up the clock controller.

13.1 Features of Clock Controller

The clock controller of the RH850/U2A-EVA microcontrollers has the following features.

- Five on-chip clock oscillators
  - Main Oscillator (Main OSC) with an oscillation frequency of 16, 20, 24, and 40 MHz
  - High Speed Internal Oscillator (HS IntOSC) with a nominal frequency of 200 MHz (Typ.)
  - Low Speed Internal Oscillator (LS IntOSC) with a nominal frequency of 240 kHz (Typ.)
  - High Voltage Internal Oscillator (HV IntOSC) with a nominal frequency of 16 MHz (Typ.)
  - Phase Locked Loop (PLL)

  The configuration of Main OSC, PLL, and HV IntOSC is variable by Option Byte OPBT10, OPBT11, and OPBT4.

  For details of Option Byte, see **Section 51, Flash Memory**, **Table 13.5, CPU System Clock Setting Table**, **Section 13.4.1, Main Oscillator (Main OSC)**, **Section 13.4.4, High Voltage Internal Oscillator (HV IntOSC)**, and **Section 13.4.5, Phase Locked Loop (PLL)**.
- Fine management of clock supply to peripheral modules
- On-chip clock monitor that detects clock anomalies when the Main Oscillator, High Speed Internal Oscillator, Low Speed Internal Oscillator, or PLL are in use. See **Section 14, Clock Monitor (CLMA)**.
- Clock output (FOUT)

13.1.1 External Input/Output Pins

Table 13.1 shows the pins related to the clock controller.

Table 13.1 Pins Related to the Clock Controller

Pin Function Name	Direction	Function
X1	Input	Main OSC crystal resonator/External clock input
X2	Output	Main OSC crystal resonator
EXTCLK0O	Output	Clock output 0 (FOUT)
EXTCLK1O	Output	Clock output 1 (FOUT)

13.2 Type of Clocks

Table 13.2 shows the List of Clock Sources, **Table 13.3** shows the List of Clocks.

Table 13.2 List of Clock Sources

Clock Name	Symbol	Clock Frequency (MHz)			Remarks
Main OSC clock	CLK_MOSC	16, 20, 24, 40			
HS IntOSC clock	CLK_HSIOSC	200			
LS IntOSC clock	CLK_LSIOSC	0.24			
HV IntOSC clock	CLK_HVIOSC	16			
PLL clock	CLK_PLL	800	640	480	Clock source: CLK_MOSC
	CLK_PLLO	800, 400	640, 320	480, 240	Clock source: CLK_PLL
Internal OSC clock	CLK_IOSC	200, 0.24			Clock source: CLK_HSIOSC or CLK_LSIOSC
System clock	CLK_SYS	800, 400	640, 320	480, 240	Clock source: CLK_PLLO
		200, 0.24			Clock source: CLK_IOSC

Table 13.3 List of Clocks

Clock Name	Symbol	Maximum Clock Frequency (MHz)			Remarks
CPU clock	CLK_CPU	400	320	240	The maximum clock frequency depends on the Option Byte CKDIVMD.
System Bus clock	CLK_SBUS	200	160	80	
H-BUS clock	CLK_HBUS	100	80	80	
Peripheral ultra high speed clock	CLK_UHSB	160	160	160	
Peripheral high speed clock	CLK_HSB	80	80	80	
Peripheral low speed clock	CLK_LSB	40	40	40	
RLIN3 comm. clock	CLK_RLIN	80			Refer to "**13.6.4, Clock Settings**" for clock source and divider.
RS-CANFD comm. clock	CLK_RCANOSC	40			
ADCJn (n = 0, 1) conv. clock	CLK_ADC	40			
MSPI comm. clock	CLK_MSPI	80			
ECM delay timer clock	CLK_ECMCNT	10			
WDTBn (n = 0 to 3), SWDTA, ICUM_WDTA0 count clock	CLK_WDT	10			
WDTBA count clock	CLKA_WDT	0.24			
TAUJn (n = 2, 3) operating clock	CLKA_TAUJ	80			
RTCA count clock	CLKA_RTCA	2.5			
ADCJ2 operating clock	CLKA_ADC	40			
Clock output 0 (FOUT)	EXTCLK0O	24			
Clock output 1 (FOUT)	EXTCLK1O	24			
LPS operating clock	CLKA_LPS	10			

Refer to **Table 13.4** about clock operation in chip standby mode.

Table 13.4 Clock Operation in Chip Standby Mode

Power Domain	Clock	STOP	DeepSTOP	Cyclic RUN	Cyclic STOP
AWO	CLK_MOSC	Operable*1	Operable*1	Operable	Operable*1
AWO	CLK_HSIOSC	Operable*1	Operable*1	Operable	Operable*1
AWO	CLK_LSIOSC	Operable	Operable	Operable	Operable
ISO	CLK_PLLO	Operable*1	Inoperable	Inoperable	Inoperable
AWO	CLK_IOSC	Operable	Operable	Operable	Operable
ISO	CLK_SYS	Operable	Inoperable	Operable	Operable
ISO	CLK_CPU	Inoperable	Inoperable	Operable	Inoperable
ISO	CLK_SBUS	Inoperable	Inoperable	Operable	Inoperable
ISO	CLK_HBUS	Inoperable	Inoperable	Operable	Inoperable
ISO	CLK_UHSB	Operable	Inoperable	Operable	Operable
ISO	CLK_HSB	Operable	Inoperable	Operable	Operable
ISO	CLK_LSB	Operable	Inoperable	Operable	Operable
AWO	EXTCLK0O	Operable	Operable	Operable	Operable
AWO	EXTCLK1O	Operable	Operable	Operable	Operable
AWO	CLKA_LPS	Operable*2	Operable*2	Operable	Operable*2

Note 1. When the Stop Mask Register is set to 1, the oscillator continues operation in chip standby mode.

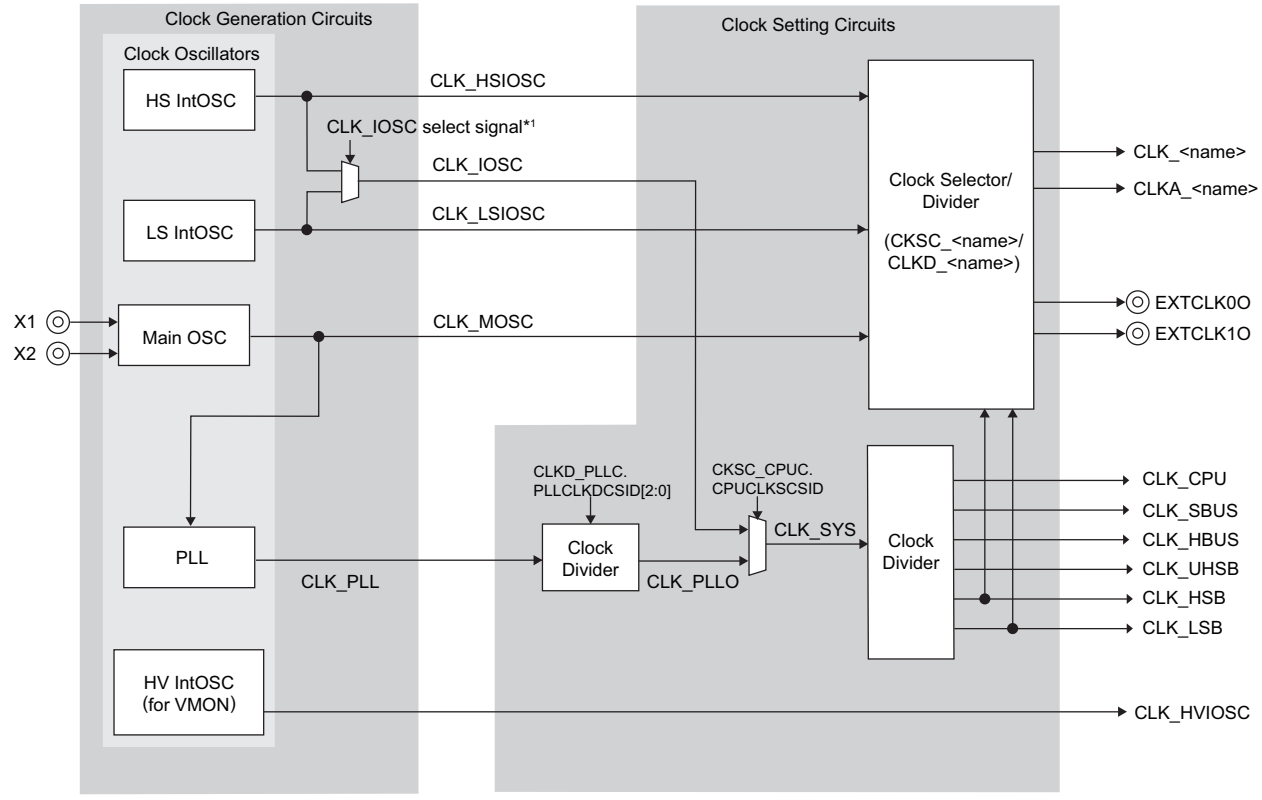
Note 2. When CLK_HSIOSC stops in chip standby mode , CLKA_LPS is CLK_LSIOSC. In other cases, CLKA_LPS is CLK_HSIOSC/20.

13.3 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is composed of clock generation circuits including clock oscillators, and clock setting circuits.

Figure 13.1 shows the configuration of the clock controller.



Note 1. For details, see **Section 13.3.1, Clock Generation Circuits, Internal OSC clock (CLK_IOSC)**.

Figure 13.1 Clock Controller Configuration

13.3.1 Clock Generation Circuits

Five clock oscillators are provided:

Four clock oscillators are located in the Always-On area (AWO area) and PLL is located in the Isolated area (ISO area).

Main Oscillator (Main OSC)

The Main OSC generates the clock CLK_MOSC, which runs at a frequency of 16, 20, 24, and 40 MHz. Generation of the clock CLK_MOSC requires the connection of an external resonator to X1 and X2. The clock CLK_MOSC is used as the reference clock for the PLL.

High Speed Internal Oscillator (HS IntOSC)

The HS IntOSC generates the clock CLK_HSIOSC, which runs at a frequency of 200 MHz (Typ.).

Low Speed Internal Oscillator (LS IntOSC)

The LS IntOSC generates the clock CLK_LSIOSC, which runs at a frequency of 240 kHz (Typ.). It starts operation at power up and cannot be stopped, hence it is always operating.

High Voltage Internal Oscillator (HV IntOSC)

The HV IntOSC generates the clock CLK_HVIOSC, which runs at a frequency of 16 MHz (Typ.).

Phase Locked Loop (PLL)

The PLL circuits generate high speed operation clock CLK_PLL for normal operation of the microcontroller.

Internal OSC clock (CLK_IOSC)

This clock is selected from CLK_LSIOSC and CLK_HSIOSC and used as startup clock for System clock (CLK_SYS).
The clock source of CLK_IOSC is CLK_HSIOSC except in the following case. In the following case, the clock source of CLK_IOSC is CLK_LSIOSC.

- CLK_HSIOSC stops in chip standby mode*1

Note 1. For the condition, see **Section 13.4.2, High Speed Internal Oscillator (HS IntOSC)**, HS IntOSC STOP Requests in Chip Standby Mode.

13.3.2 Clock Setting Circuits

The clocks generated by the clock oscillators are input to the clock selectors CKSC _<*name*> and the clock dividers CLKD_<*name*>.

The clocks for the CPU System and the peripheral modules are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

Note that not all available clocks generated by the clock oscillators are input to each clock selector or clock divider.

CPU System clock (CLK_CPU, CLK_SBUS, CLK_HBUS, CLK_UHSB, CLK_HSB, CLK_LSB)

The System clock (CLK_SYS) is selected from the PLL clock (CLK_PLLO) and the Internal OSC clock (CLK_IOSC) by the CKSC_CPUC register setting, and the CPU System clock (CLK_CPU, CLK_SBUS, CLK_HBUS, CLK_UHSB, CLK_HSB, CLK_LSB) is divided from the System clock (CLK_SYS).

For the CPU System clock setting table, see **Table 13.5**.

Table 13.5 CPU System Clock Setting Table

Clock Source	CLK_IOSC						CLK_PLLO					
	LS IntOSC			HS IntOSC			PLL					
Clock Setting	Register CKSC_CPUC.CPUCLKSCSID *2											
	1_B						0_B					
	Option Byte CKDIVMD *1											
	11_B	10_B	$0X_B$	11_B	10_B	$0X_B$	11_B		10_B		$0X_B$	
	—						Register CLKD_PLLC.PLLCLKDCSID[2:0] *3					
							001B	010B	001B	010B	001B	010B
CLK_CPU (MHz)	0.12	0.12	0.12	100	100	100	400	200	320	160	240	120
CLK_SBUS (MHz)	0.06	0.06	0.04	50	50	33.3	200	100	160	80	80	40
CLK_HBUS (MHz)	0.03	0.03	0.04	25	25	33.3	100	50	80	40	80	40
CLK_UHSB (MHz)	0.048	0.06	0.08	40	50	66.7	160	80	160	80	160	80
CLK_HSB (MHz)	0.024	0.03	0.04	20	25	33.3	80	40	80	40	80	40
CLK_LSB (MHz)	0.012	0.015	0.02	10	12.5	16.7	40	20	40	20	40	20

Note 1. For details, see **Section 51, Flash Memory**

Note 2. For details, see **Section 13.5.5.1, System Clock (CLK_SYS)**

Note 3. For details, see **Section 13.5.5.2, PLL Clock (CLK_PLLO)**

13.4 Clock Oscillators

13.4.1 Main Oscillator (Main OSC)

The Main Oscillator generates the clock CLK_MOSC. CLK_MOSC is also used as the PLL input clock. **Figure 13.2** shows the basic configuration and signals of the Main OSC.

The block diagram of Main OSC configuration and the sequence of Main OSC stabilization are shown in **Figure 13.3** and **Figure 13.4** respectively.

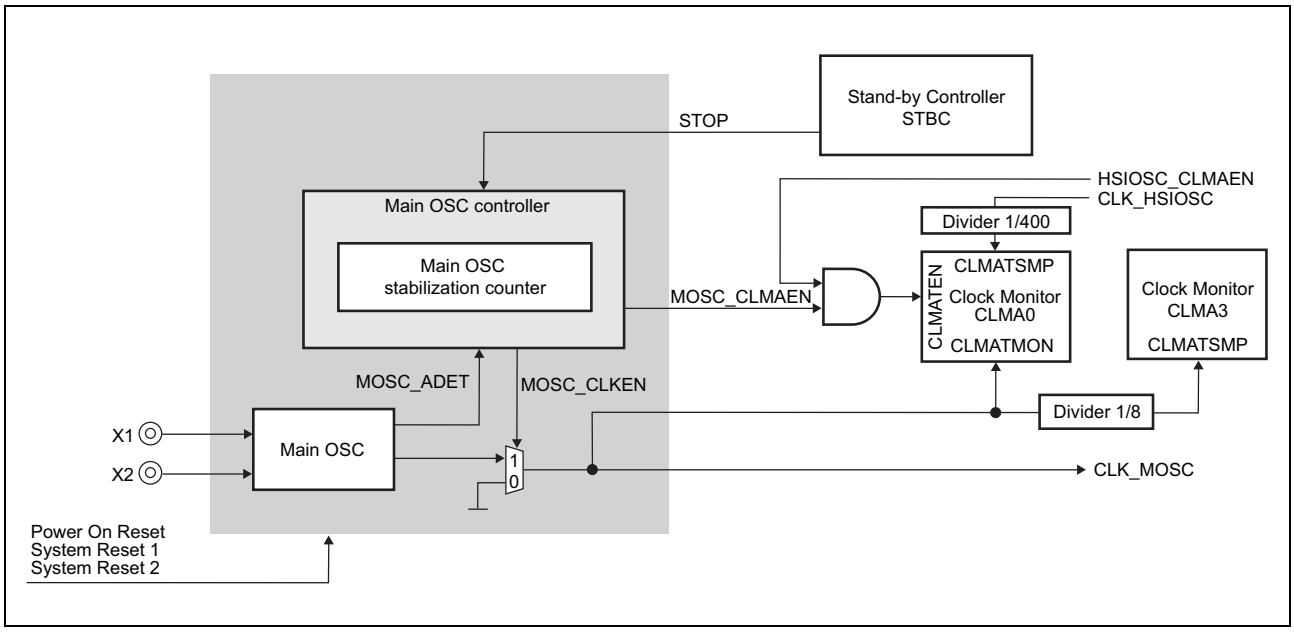


Figure 13.2 Main Oscillator (Main OSC)

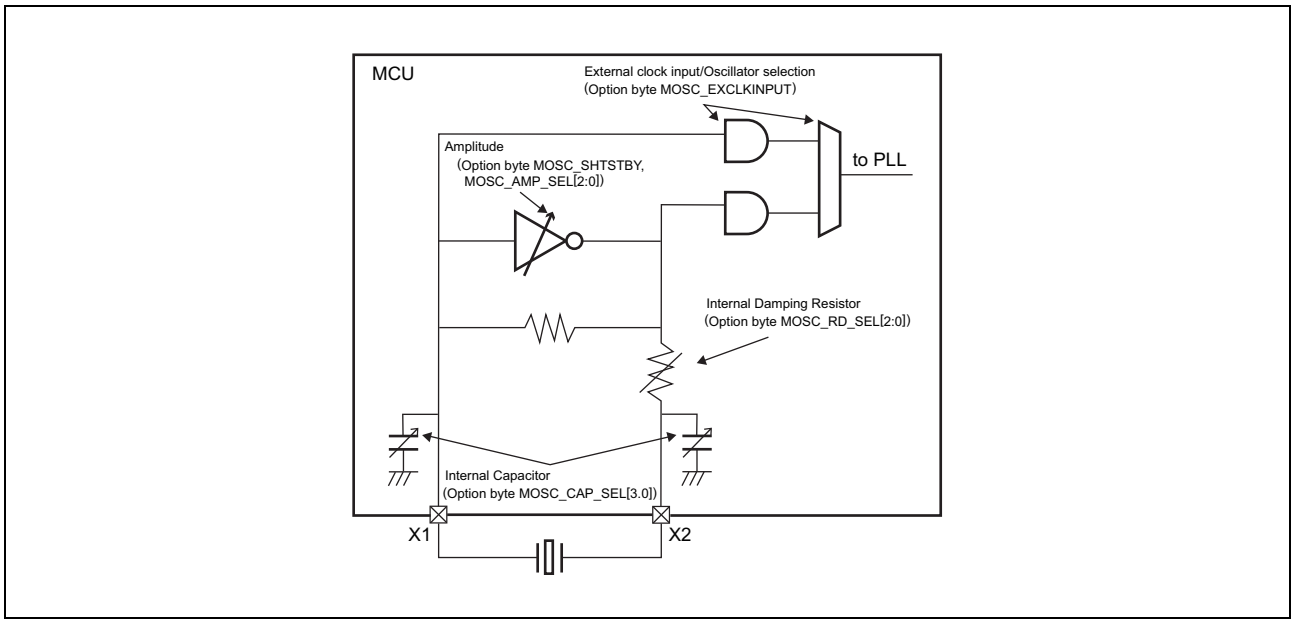


Figure 13.3 Block Diagram of Main OSC Configuration

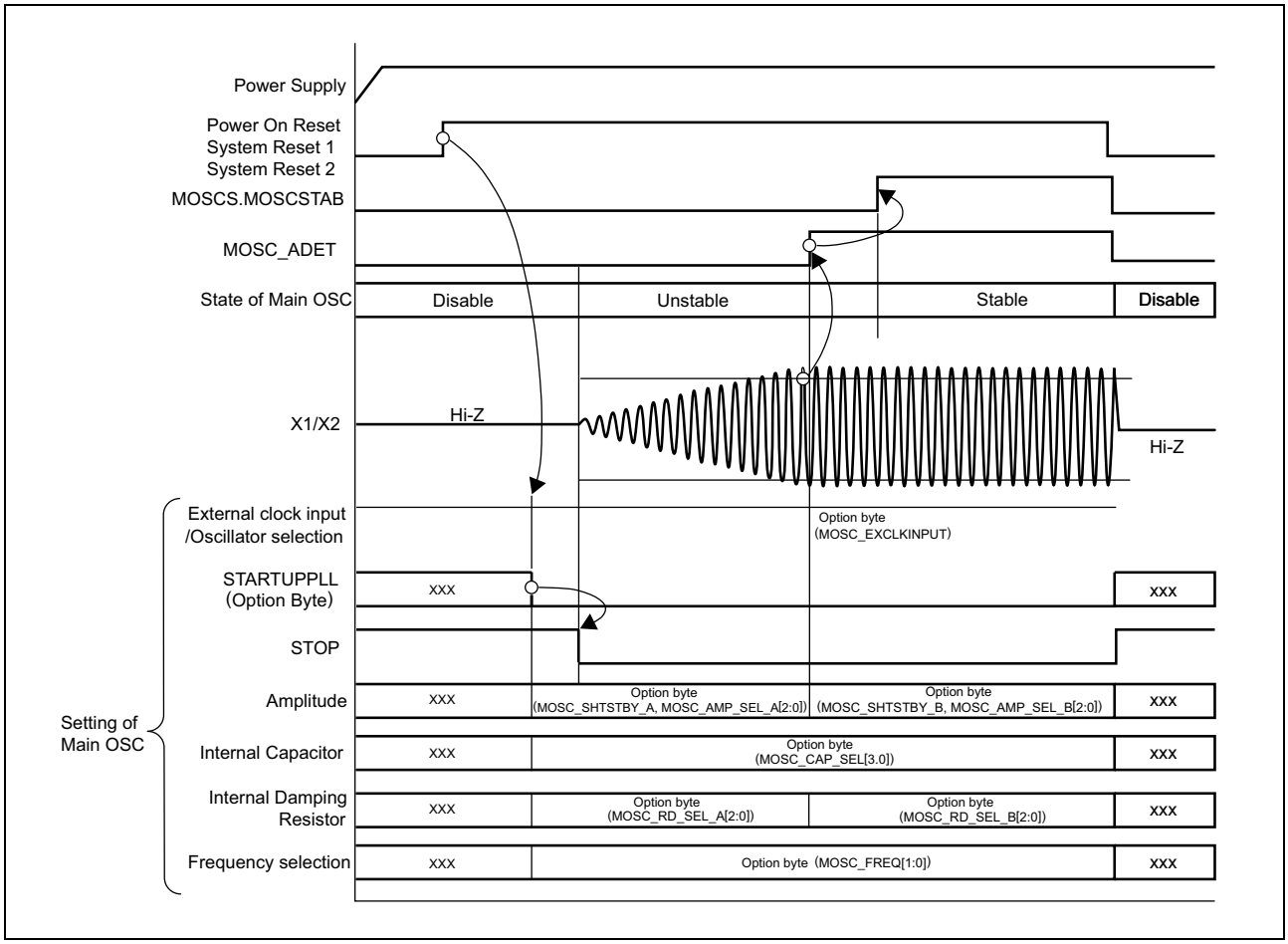


Figure 13.4 Sequence of Main OSC Stabilization

Main OSC

The Main OSC is initialized by Power On Reset and System Reset 1 and System Reset 2.

Whether to start or stop the Main OSC initial state after Reset release is determined by Option Byte STARTUPPLL.

After CPU Startup, it is controlled by the Main OSC Enable trigger bit (MOSCE.MOSCENTRG) or Main OSC Disable Trigger (MOSCE.MOSCDISTRG).

If DMON is enabled, the Main OSC shall not be set to disable.

Main OSC Stabilization

After the Main OSC starts oscillation, the X1/X2 signal amplitude are growing. When detecting enough signal amplitude level, the MOSC_ADET signal switches from 0 to 1 and the Main OSC stabilization counter starts counting the stabilization time. When the Main OSC stabilization counter reaches the predefined value, the Main OSC is assumed to be stable and the MOSC_CLKEN signal switches from 0 to 1 to enable the Main OSC clock CLK_MOSC.

As long as the Main OSC is not stable, the MOSC_CLKEN signal disables the Main OSC clock CLK_MOSC.

When the Main OSC is stopped by transition to chip standby mode (MOSCSTPM.MOSCSTPMSK = 0), MOSC_CLMAEN signal switches from 1 to 0 to disable the Clock Monitor CLMA0.

The stable and active state of the Main OSC is indicated by the Main OSC stable state bit (MOSCS.MOSCSTAB = 1).

For details of Option Byte, see **Section 51, Flash Memory**. SHTSTBY_A for start-up must be “1” and SHTSTBY_B for stable must be “0”.

Main OSC STOP Requests in Chip Standby Mode

The STOP signal from the Chip Standby Controller requests the Main OSC Controller to switch off the CLK_MOSC clock in chip standby modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).

The stop request mask bit MOSCSTPM.MOSCSTPMSK controls whether the Main OSC is stopped during chip standby mode or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0:
  The STOP request signal is not masked, so the Main OSC is stopped in chip standby mode.
  If the Main OSC is in operation before chip standby mode, it is automatically re-started after wake-up from chip standby mode, and the Main OSC stabilization counter counts the oscillation stabilization time.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is masked, so the Main OSC continues to operate in chip standby mode.

Clock Monitor Control

The MOSC_CLMAEN signal and the HSIOSC_CLMAEN signal enable or disable supervision by the Clock Monitor CLMA0 when chip standby mode.

If CLMA0 is used in chip standby mode, set MOSCSTPM.MOSCSTPMSK and HSOSCSTPM.HSOSCSTPMSK to 1.

Main OSC Enable/Disable Trigger

The Main OSC can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger MOSCE.MOSCENTRG = 1 starts the Main OSC.
  Note that setting the enable trigger is only effective if the Main OSC is unstable, i.e. if MOSCS.MOSCSTAB = 0.
- Disable trigger MOSCE.MOSCDISTRG = 1 stops the Main OSC.
  Note that setting the disable trigger is only effective if the Main OSC is stable, i.e. if MOSCS.MOSCSTAB = 1.

Direct Clock Input to X1 (EXCLK Mode)

A clock waveform from an external clock source can be supplied to X1 pin. In this case, set the MOSC_EXCLKINPUT bit of OPBT10 register to 0 before clock input to X1 pin is supplied.
see **Section 51, Flash Memory**.

The behavior of CLK_MOSC is the same between EXCLK mode and OSC mode.
Enable trigger MOSCE.MOSCENTRG = 1 starts the Main OSC (CLK_MOSC).
Disable trigger MOSCE.MOSCDISTRG=1 stops the Main OSC (CLK_MOSC).
The STOP signal from the Chip Standby Controller requests the Main OSC Controller to switch off the CLK_MOSC clock in chip standby modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).

13.4.2 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock CLK_HSIOSC. CLK_HSIOSC has a nominal frequency of 200 MHz.

Figure 13.5 shows the basic configuration and signals of the HS IntOSC.

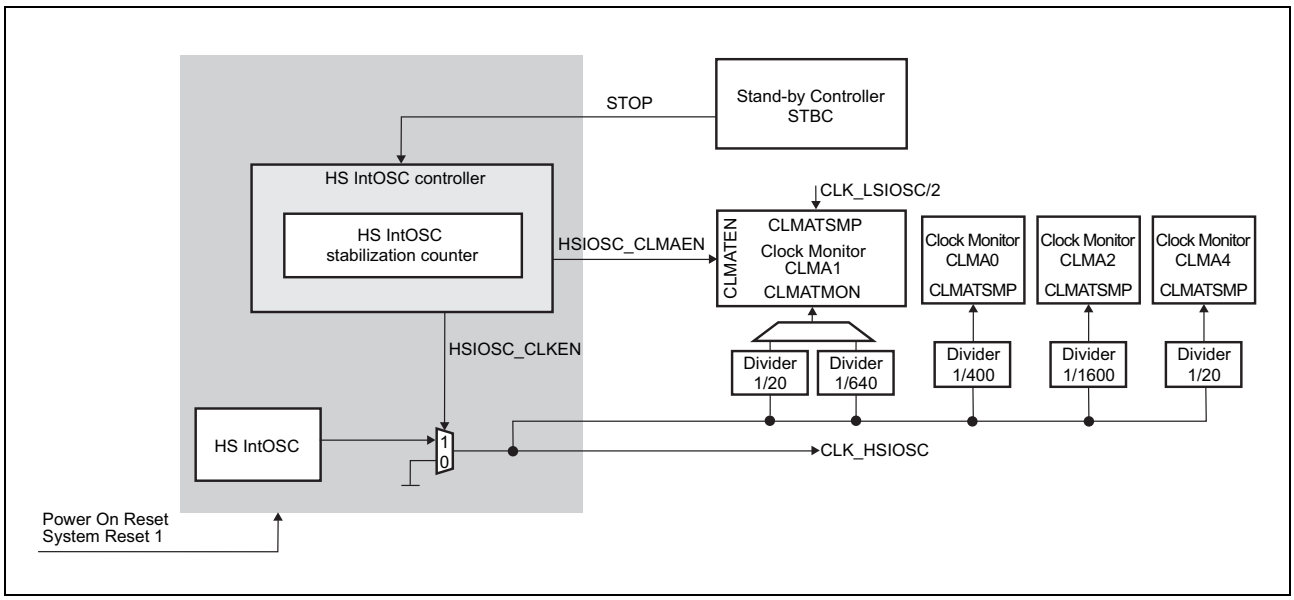


Figure 13.5 High Speed Internal Oscillator (HS IntOSC)

After Power On Reset or System Reset 1 release, the HS IntOSC starts operation.

HS IntOSC Stabilization

The HS IntOSC stabilization counter starts counting the stabilization time, after the HS IntOSC enabled. When the HS IntOSC stabilization counter reaches the predefined value, the HS IntOSC is assumed to be stable and the HSIOSC_CLKEN signal switches from 0 to 1 to enable the HS IntOSC clock CLK_HSIOSC.

As long as the HS IntOSC is not stable, the HSIOSC_CLKEN signal disables the HS IntOSC clock CLK_HSIOSC.

When the HS IntOSC is stopped by transition to chip standby mode (HSOSCSTPM.HSOSCSTPMSK = 0), HSIOSC_CLMAEN signal switches from 1 to 0 to disable the Clock Monitor CLMAn (n = 0, 1, 2).

The stable and active state of the HS IntOSC is indicated by the HS IntOSC clock stable state bit (HSOSCS.HSOSCSTAB = 1).

HS IntOSC STOP Requests in Chip Standby Mode

The STOP signal from the Chip Standby Controller requests the HS IntOSC Controller to switch off the CLK_HSIOSC clock in chip standby modes (STOP mode, DeepSTOP mode and Cyclic STOP mode).
The stop request mask bit HSOSCSTPM.HSOSCSTPMSK controls whether the HS IntOSC is stopped during chip standby mode or continues operation:

- HSOSCSTPM.HSOSCSTPMSK = 0:
  The STOP request signal is not masked, so the HS IntOSC is stopped during chip standby mode and automatically restarted after wake-up from chip standby mode.
  The STOP request is masked under the following conditions, even if HSOSCSTPM.HSOSCSTPMSK = 0. Therefore, the HS IntOSC will continue to operate even in chip standby mode.
  - If the low power sampler (LPS) is operating
- HSOSCSTPM.HSOSCSTPMSK = 1:
  The STOP request signal is masked, so the HS IntOSC continues to operate during chip standby mode.

Clock Monitor Control

The HSIOSC_CLMAEN signal enables or disables supervision by the Clock Monitor CLMAn (n = 0, 1, 2) when chip standby mode.

If CLMA1 is used in chip standby mode, set HSOSCSTPM.HSOSCSTPMSK to 1.

13.4.3 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock CLK_LSIOSC. CLK_LSIOSC has a nominal frequency of 240 kHz.

Figure 13.6 shows the basic configuration and signals of the LS IntOSC.

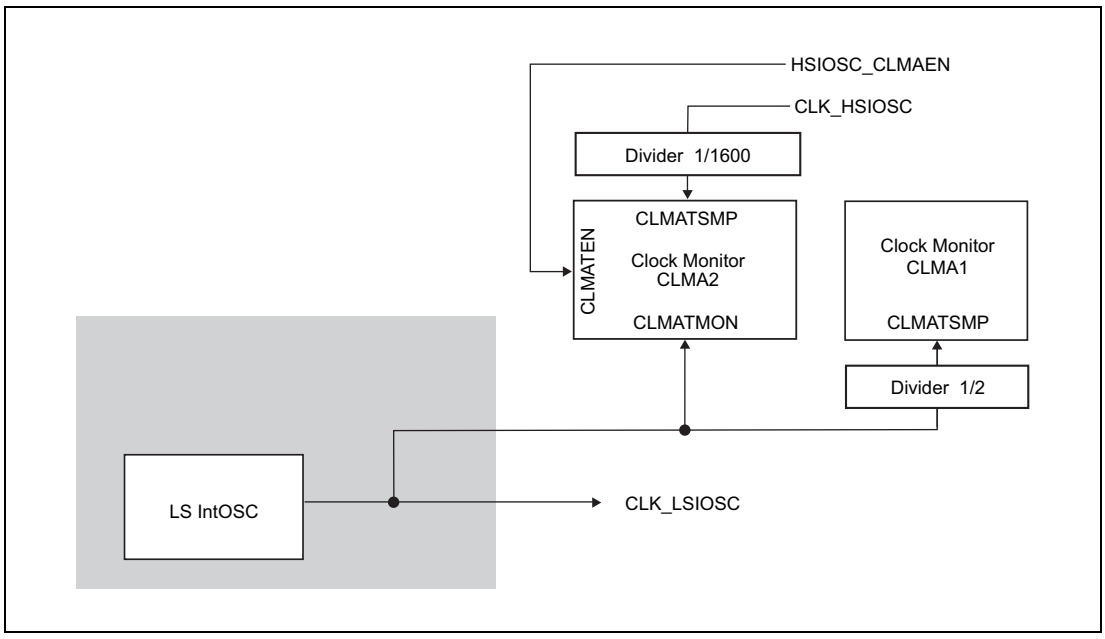


Figure 13.6 Low Speed Internal Oscillator (LS IntOSC)

After power supply the LS IntOSC starts operation. It cannot be stopped.

The LS IntOSC clock CLK_LSIOSC is used as the sampling clock for the Clock Monitor CLMA1.

Clock Monitor Control

The HSIOSC_CLMAEN signal enables or disables supervision by the Clock Monitor CLMA2 when chip standby mode.

If CLMA2 is used in chip standby mode, set HSOSCSTPM.HSOSCSTPMSK to 1.

13.4.4 High Voltage Internal Oscillator (HV IntOSC)

The High Voltage Internal Oscillator generates the clock CLK_HVIOSC. CLK_HVIOSC has a nominal frequency of 16 MHz.

Figure 13.7 shows the basic configuration and signals of the HV IntOSC.

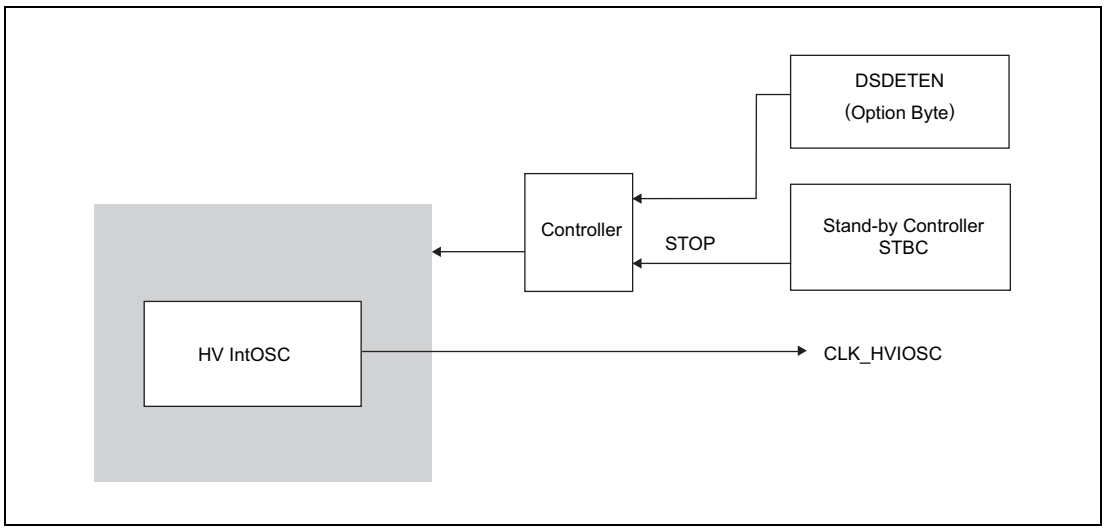


Figure 13.7 High Voltage Internal Oscillator (HV IntOSC)

After power supply the HV IntOSC starts operation.

HV IntOSC in Chip Standby Mode

The HV IntOSC is automatically disabled depends on Option Byte (DSDETEN) value when transitioning to DeepSTOP mode.
For details, see **Section 11.3.7.4, Function in Standby**.

13.4.5 Phase Locked Loop (PLL)

Main OSC is input to a phase-locked loop (PLL) clock oscillator as reference clock.
The PLL output clock CLK_PLL serve as the main operation clocks for the microcontroller.

Figure 13.8 shows the basic configuration and signals of the PLL.

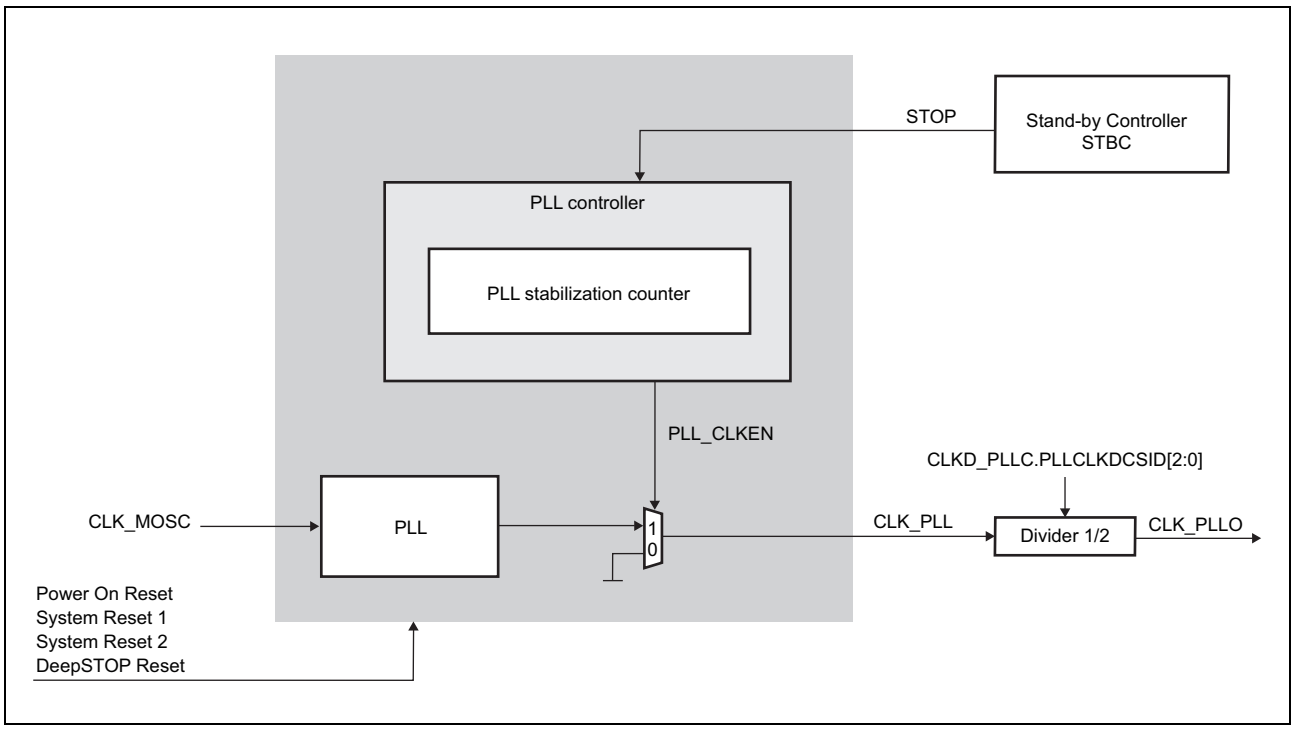


Figure 13.8 PLL

PLL Enable

The PLL is initialized by Power On Reset and System Reset 1 and System Reset 2 and DeepSTOP Reset.
Whether to start or stop the PLL initial state after Reset release is determined by Option Byte STARTUPPLL.

After CPU Startup, it is controlled by the PLL Enable trigger bit (PLLE.PLLENTRG) or PLL Disable Trigger (PLLE.PLLDISTRG).

PLL Stabilization

The PLL stabilization counter starts counting the stabilization time, after the PLL enabled.

When the PLL stabilization counter reaches the predefined value, the PLL is assumed to be stable and the PLL_CLKEN signal switches from 0 to 1 to enable the PLL clock CLK_PLL.

As long as the PLL is not stable, the PLL_CLKEN signal disables the PLL clock CLK_PLL.

The stable and active state of the PLL is indicated by the PLL clock stable state bit (PLLS.PLLCLKSTAB = 1).

PLL in Chip Standby Mode

When the chip standby mode transitions to DeepSTOP mode, the PLL is automatically disabled.

After restoring from DeepSTOP mode the PLL needs to be reconfigured.

When the chip standby mode transitions from STOP mode to RUN mode, the PLL automatically resumes.

In Cyclic RUN and Cyclic STOP mode, the PLL is not available. Do not enable the PLL by PLL Enable register.

PLL STOP Requests in Chip Standby Mode

The STOP signal from the Chip Standby Controller requests the PLL Controller to switch off the CLK_PLLO clock in chip standby modes (STOP mode, DeepSTOP mode and Cyclic STOP mode). The stop request mask bit PLLSTPM.PLLSTPMSK controls whether the PLL is stopped during chip standby mode or continues operation:

Table 13.6 PLL Stop Mask Setting

Main OSC		PLL		Behavior of the Main OSC and the PLL in chip standby mode
Status Register	Stop Mask Register	Status Register	Stop Mask Register	
MOSCSTAB	MOSCSTPMSK	PLLCLKSTAB	PLLSTPMSK	
1	0	1	0	The Main OSC and the PLL are stopped during chip standby mode.
			1	Setting prohibited.
		0	0 or 1	The Main OSC is stopped during chip standby mode. The PLL continues to stop.
	1	1	0	The Main OSC continues to operate during chip standby mode. The PLL is stopped during chip standby mode.
			1	The Main OSC and the PLL continues to operate during chip standby mode.
		0	0 or 1	The Main OSC continues to operate during chip standby mode. The PLL continues to stop.
0	0 or 1	0	0 or 1	The Main OSC and the PLL continues to stop.

PLL Enable/Disable Trigger

The PLL can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLLE.PLLENTRG = 1 starts the PLL
  Note that setting the enable trigger is only effective if the PLL is unstable, i.e. if PLLS.PLLCLKSTAB = 0.
- Disable trigger PLLE.PLLDISTRG = 1 stops the PLL
  Note that setting the disable trigger is only effective if the PLL is stable, i.e. if PLLS.PLLCLKSTAB = 1.

13.5 Registers

13.5.1 Register Protection

Write protection prevents inadvertent change of register content due to erroneous software execution.

Registers CLKKCPROT1 have to be written first in order to unlock protected registers and allow change of register content.

13.5.2 List of Registers

The registers of the clock controller are listed below.

Table 13.7 List of Clock Controller Registers for ISO area

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 0000$_H$	PLLE	PLL Enable Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 0004$_H$	PLLS	PLL Status Register	32	0000 000X$_H$	PBG20 #2	
FF98 000C$_H$	PLLSTPM	PLL Stop Mask Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 0100$_H$	CKSC_CPUC	CLK_CPU Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0108$_H$	CKSC_CPUS	CLK_CPU Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 0120$_H$	CLKD_PLLC	CLK_PLLO Divider Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0128$_H$	CLKD_PLLS	CLK_PLLO Divider Status Register	32	0000 0003$_H$	PBG20 #2	
FF98 0140$_H$	CKSC_RLINC	CLK_RLIN Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0148$_H$	CKSC_RLINS	CLK_RLIN Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 0150$_H$	CKSC_RCANC	CLK_RCANOSC Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0158$_H$	CKSC_RCANS	CLK_RCANOSC Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 0160$_H$	CKSC_ADCC	CLK_ADC Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0168$_H$	CKSC_ADCS	CLK_ADC Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 0170$_H$	CKSC_MSPIC	CLK_MSPI Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 0178$_H$	CKSC_MSPIS	CLK_MSPI Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 0700$_H$	CLKKCPROT1	Clock Controller Register Key Code Protection Register 1	32	0000 0000$_H$	PBG20 #2	

Table 13.8 List of Clock Controller Registers for AWO area

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 8000$_H$	MOSCE	Main OSC Enable Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 8004$_H$	MOSCS	Main OSC Status Register	32	0000 000X$_H$	PBG20 #2	
FF98 800C$_H$	MOSCSTPM	Main OSC Stop Mask Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 8100$_H$	HSOSCS	HS IntOSC Status Register	32	0000 0000$_H$	PBG20 #2	
FF98 8104$_H$	HSOSCSTPM	HS IntOSC Stop Mask Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 8200$_H$	CKSC_AWDTC	CLKA_WDT Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8208$_H$	CKSC_AWDTS	CLKA_WDT Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8210$_H$	CKSC_ATAUJC	CLKA_TAUJ Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8218$_H$	CKSC_ATAUJS	CLKA_TAUJ Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8220$_H$	CKSC_ARTCAC	CLKA_RTCA Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8228$_H$	CKSC_ARTCAS	CLKA_RTCA Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8230$_H$	CKSC_AADCC	CLKA_ADC Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8238$_H$	CKSC_AADCS	CLKA_ADC Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8240$_H$	CLKD_AADCC	CLKA_ADC Divider Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8248$_H$	CLKD_AADCS	CLKA_ADC Divider Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8250$_H$	CKSC_FOUT0C	FOUT0 Clock Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8258$_H$	CKSC_FOUT0S	FOUT0 Clock Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8260$_H$	CLKD_FOUT0C	FOUT0 Clock Divider Control Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 8268$_H$	CLKD_FOUT0S	FOUT0 Clock Divider Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8270$_H$	CKSC_FOUT1C	FOUT1 Clock Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8278$_H$	CKSC_FOUT1S	FOUT1 Clock Selector Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8280$_H$	CLKD_FOUT1C	FOUT1 Clock Divider Control Register	32	0000 0000$_H$	PBG20 #2	CLKKCPROT1
FF98 8288$_H$	CLKD_FOUT1S	FOUT1 Clock Divider Status Register	32	0000 0001$_H$	PBG20 #2	
FF98 8300$_H$	CKSC_WDTC	CLK_WDT Selector Control Register	32	0000 0001$_H$	PBG20 #2	CLKKCPROT1
FF98 8308$_H$	CKSC_WDTS	CLK_WDT Selector Status Register	32	0000 0001$_H$	PBG20 #2	

13.5.3 Reset of Registers

Register reset condition is shown in **Table 13.9** and **Table 13.10**.

Table 13.9 Register Reset Condition for ISO area

Symbol	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PLLE	√	√	√	—	√	—	—
PLLS	√	√	√	—	√	—	—
PLLSTPM	√	√	√	—	√	—	—
CKSC_CPUC	√	√	√	—	√	—	—
CKSC_CPUS	√	√	√	—	√	—	—
CLKD_PLLC	√	√	√	—	√	—	—
CLKD_PLLS	√	√	√	—	√	—	—
CKSC_RLINC	√	√	√	√	√	—	—
CKSC_RLINS	√	√	√	√	√	—	—
CKSC_RCANC	√	√	√	√	√	—	—
CKSC_RCANS	√	√	√	√	√	—	—
CKSC_ADCC	√	√	√	√	√	—	—
CKSC_ADCS	√	√	√	√	√	—	—
CKSC_MSPIC	√	√	√	√	√	—	—
CKSC_MSPIS	√	√	√	√	√	—	—
CLKKCPROT1	√	√	√	—	√	—	—

Table 13.10 Register Reset Condition for AWO area

Symbol	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
MOSCE	√	√	√	—	—	—	—
MOSCS	√	√	√	—	—	—	—
MOSCSTPM	√	√	√	—	—	—	—
HSOSCS	√	√	—	—	—	—	—
HSOSCSTPM	√	√	—	—	—	—	—
CKSC_AWDTC	√	√	√	√	—	—	—
CKSC_AWDTS	√	√	√	√	—	—	—
CKSC_ATAUJC	√	√	√	√	—	—	—
CKSC_ATAUJS	√	√	√	√	—	—	—
CKSC_ARTCAC	√	√	√	—	—	—	—
CKSC_ARTCAS	√	√	√	—	—	—	—
CKSC_AADCC	√	√	√	√	—	—	—
CKSC_AADCS	√	√	√	√	—	—	—
CLKD_AADCC	√	√	√	√	—	—	—
CLKD_AADCS	√	√	√	√	—	—	—
CKSC_FOUT0C	√	√	√	√	—	—	—
CKSC_FOUT0S	√	√	√	√	—	—	—
CLKD_FOUT0C	√	√	√	√	—	—	—
CLKD_FOUT0S	√	√	√	√	—	—	—
CKSC_FOUT1C	√	√	√	√	—	—	—
CKSC_FOUT1S	√	√	√	√	—	—	—
CLKD_FOUT1C	√	√	√	√	—	—	—
CLKD_FOUT1S	√	√	√	√	—	—	—
CKSC_WDTC	√	√	√	√	—	—	—
CKSC_WDTS	√	√	√	√	—	—	—

13.5.4 Clock Oscillator Registers

13.5.4.1 PLLE — PLL Enable Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: FF98 0000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLDIS TRG	PLLEN TRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 13.11 PLLE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	PLLDISTRG*1	PLL Disable Trigger 0: No function 1: Stops PLL
0	PLLENTRG*2	PLL Enable Trigger 0: No function 1: Starts PLL

Note 1. Follow the procedure given below for stopping the PLL by using PLLDISTRG.
1. Confirm that the PLL is stable. See **Note 2.** about PLL is stable.
2. Select CLK_IOSC for clock source of CLK_SYS (CKSC_CPUS.CPUCLKSACT = 1).
3. Stop the PLL (PLLE.PLLDISTRG = 1).
4. Confirm that the PLL has been stopped (PLLS.PLLCLKEN = 0 & PLLS.PLLCLKSTAB = 0).

Note 2. Follow the procedure given below for starting the PLL by using PLLENTRG.
1. Confirm that the Main OSC is stable. See **Note 2.** of **Table 13.14** about Main OSC is stable.
2. Confirm that the PLL is stopped (PLLS.PLLCLKEN = 0 & PLLS.PLLCLKSTAB = 0).
3. Start the PLL (PLLE.PLLENTRG = 1).
4. Confirm that the PLL has been stable (PLLS.PLLCLKEN = 1 & PLLS.PLLCLKSTAB = 1).
5. Wait 1.4us or Read the PLLS.PLLCLKSTAB = 1 four times.

CAUTION

It is prohibited to write 1 to PLLDISTRG and PLLENTRG at the same time.

13.5.4.2 PLLS — PLL Status Register

This register provides stable status information about the PLL.

Access: This register can be read in 32-bit units.

Address: FF98 0004$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLCLKSTAB	PLLCLKEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*2	x*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Initial value depends on Option Byte (OPBT11.STARTUPPLL) except DeepSTOP Reset and serial programming mode.
Initial value when returning from DeepSTOP to RUN/Cyclic RUN is 0.
Initial value in serial programming mode is 0.

Note 2. In case the initial value of PLLCLKEN bit is 1, the reading value depends on the read timing and the stabilization time.

Table 13.12 PLLS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	PLLCLKSTAB	PLL clock stable state 0: PLL clock is unstable. 1: PLL clock is stable.*1
0	PLLCLKEN	PLL clock enable state 0: PLLE.PLLDISTRG is set to 1. 1: PLLE.PLLENTRG is set to 1.

Note 1. Follow the procedure given below for checking stable.
1. Confirm that the PLL has been stable (PLLS.PLLCLKEN = 1 & PLLS.PLLCLKSTAB = 1).
2. Wait 1.4 µs or Read the PLLS.PLLCLKSTAB = 1 four times.

CAUTION

PLLS.PLLSTAB is kept high even if setting PLLE.PLLDISTRG to 1 when PLL is selected for System clock source PLLS.PLLCLKEN is cleared to 0.

13.5.4.3 PLLSTPM — PLL Stop Mask Register

Access: This register can be read or written in 32-bit units.

Address: FF98 000C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.13 PLLSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PLLSTPMSK	PLL Stop Request Mask 0: PLL stops operation in chip standby mode 1: PLL continues operation in chip standby mode

13.5.4.4 MOSCE — Main OSC Enable Register

This register is used to start and stop the Main OSC.

Access: This register is a write-only register that can be write in 32-bit units.

Address: FF98 8000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCDISTRG	MOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 13.14 MOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	MOSCDISTRG *1	Main OSC Disable Trigger 0: No function 1: Stops Main OSC
0	MOSCENTRG*2	Main OSC Enable Trigger 0: No function 1: Starts Main OSC

Note 1. Follow the procedure given below for stopping the Main OSC by using MOSCDISTRG.
1. Confirm that the Main OSC is stable. See **Note 2.** about Main OSC is stable.
2. Select a clock source other than MOSC. Select Main OSC (CLK_MOSC) divided by 1 for CLK_RCANOSC.
3. Stop the Main OSC (MOSCE.MOSCDISTRG = 1).
4. Confirm that the Main OSC has been stopped (MOSCS.MOSCEN = 0 & MOSCS.MOSCSTAB = 0).

Note 2. Follow the procedure given below for starting the Main OSC by using MOSCENTRG.
1. Confirm that the Main OSC is stopped (MOSCS.MOSCEN = 0 & MOSCS.MOSCSTAB = 0).
2. Start the Main OSC (MOSCE.MOSCENTRG = 1).
3. Confirm that the Main OSC has been stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).
4. Wait 1.4us or Read the MOSCS.MOSCSTAB = 1 four times.

CAUTIONS

- **It is prohibited to write 1 to MOSCDISTRG and MOSCENTRG at the same time.**
- **Don't disable Main OSC while CLK_MOSC is selected for CLKA_ADC.**

13.5.4.5 MOSCS — Main OSC Status Register

This register provides stable status information about the Main OSC.

Access: This register can be read in 32-bit units.

Address: FF98 8004$_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTAB	MOSCEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*2	x*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Initial value depends on Option Byte (OPBT11.STARTUPPLL) except serial programming mode. Initial value in serial programming mode is 0.

Note 2. In case the initial value of MOSCEN bit is 1, the reading value depends on the read timing and the stabilization time.

Table 13.15 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	MOSCSTAB	Main OSC stable state 0: Main OSC clock is unstable. 1: Main OSC clock is stable.*1
0	MOSCEN	Main OSC enable state 0: MOSCE.MOSCDISTRG is set to 1. 1: MOSCE.MOSCENTRG is set to 1.

Note 1. Follow the procedure given below for checking stable.
1. Confirm that the Main OSC has been stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).
2. Wait 1.4 µs or Read the MOSCS.MOSCSTAB = 1 four times.

CAUTION

MOSCS.MOSCSTAB is kept high even if setting MOSCE.MOSCDISTRG to 1 when PLL is operating MOSCS.MOSCEN is cleared to 0.

13.5.4.6 MOSCSTPM — Main OSC Stop Mask Register

Access: This register can be read or written in 32-bit units.

Address: FF98 800C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.16 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCSTPMSK	Main OSC Stop Request Mask 0: Main OSC stops operation in chip standby mode. 1: Main OSC continues operation in chip standby mode.

13.5.4.7 HSOSCS — HS IntOSC Status Register

This register provides stable status information about the HS IntOSC.

Access: This register can be read in 32-bit units.

Address: FF98 8100_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSOSC STAB	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value becomes 1 before CPU reset is released.

Table 13.17 HSOSCS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	HSOSCSTAB	HS IntOSC clock stable State 0: HS IntOSC clock is unstable 1: HS IntOSC clock is stable
0	Reserved	When read, the value after reset is returned.

13.5.4.8 HSOSCSTPM — HS IntOSC Stop Mask Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8104_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSOSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.18 HSOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	HSOSCSTPMSK	HS IntOSC Stop Request Mask 0: HS IntOSC stops operation in chip standby mode 1: HS IntOSC continues operation in chip standby mode

13.5.5 Clock Selector/Divider Control Registers

13.5.5.1 System Clock (CLK_SYS)

(1) CKSC_CPUC — CLK_SYS Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0100_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.19 CKSC_CPUC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CPUCLKSCSID	Clock source control These bits select the ID of a clock source for CLK_SYS ID = 0: CLK_PLLO ID = 1: CLK_IOSC (default)

CAUTION

The clock source selected for the CLK_SYS should not be stopped by software.

(2) CKSC_CPUS — CLK_SYS Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0108$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.20 CKSC_CPUS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CPUCLKSACT	Clock source status for CLK_SYS Indicates the ID of the selected clock source of the CLK_SYS. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CLK_PLLO ID = 1: CLK_IOSC refer to **Table 13.5**

13.5.5.2 PLL Clock (CLK_PLLO)

(1) CLKD_PLLC — CLK_PLLO Divider Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0120_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLCLKDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 13.21 CLKD_PLLC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PLLCLKDCSID [2:0]	Clock Divider control These bits select the ID of a clock divider for CLK_PLLO 001_B: No division (default) 010_B: Divided by 2 Setting other than the above is prohibited.

CAUTIONS

1. **There is limitation that PLLCLKDCSID[2:0] = 010_B setting is only permitted when setting CPU System clock Gear Up/Down. For details of clock Gear Up/Down, see Section 13.6.5, Sequence for Shifting the CPU System Clock Gear Up/Down.**
2. **This register can be set during PLL is stable. See Note 2. of Table 13.14 about PLL is stable.**
   The setting during PLL is not stable is not guaranteed.
3. **This register can be set during CLKD_PLLS.PLLCLKDSYNC = 1 which indicates the divider is stable.**
   The behavior is not guaranteed if setup is performed while CLKD_PLLS.PLLCLKDSYNC = 0.

(2) CLKD_PLLS — CLK_PLLO Divider Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0128_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLCLKDSYNC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.22 CLKD_PLLS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	PLLCLKDSYNC	Divider clock synchronized 0: Clock output does not correspond to the actual divisor setting in PLLCLKDCSID 1: Clock output corresponds to the actual divisor setting in PLLCLKDCSID
0	Reserved	When read, the value after reset is returned.

13.5.5.3 RLIN Clock (CLK_RLIN)

(1) CKSC_RLINC — CLK_RLIN Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0140_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLINSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 13.23 CKSC_RLINC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RLINSCSID[1:0]	Clock source control These bits select the ID of a clock source for CLK_RLIN*1 ID = 00_B: CLK_MOSC ID = 01_B: CLK_HSB (default) ID = 10_B: CLK_MOSC/4 ID = 11_B: CLK_MOSC/8

Note 1. When the module operates in chip standby mode, select a clock source other than CLK_HSB.

CAUTIONS

- **In the case of RLINSCSID[1:0] = 00_B or 10_B or 11_B,**
  The CKSC_RLINC register must be set so that the relationship between frequency (1) and (2) is retained within the range of "(1) ≥ (2) ×2 ".
  (1) Frequency [MHz] specified by the CLK_HSB.
  (2) Frequency [MHz] specified by the CLK_RLIN.
- **Write this register only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).**

(2) CKSC_RLINS — CLK_RLIN Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0148$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLINSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.24 CKSC_RLINS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1 , 0	RLINSACT[1:0]	Clock source status for CLK_RLIN Indicates the ID of the selected clock source of the CLK_RLIN. The value indicated by these bits shows the actual ID that is currently selected. ID = 00_B: CLK_MOSC ID = 01_B: CLK_HSB ID = 10_B: CLK_MOSC/4 ID = 11_B: CLK_MOSC/8

13.5.5.4 RS-CANFD Clock (CLK_RCANOSC)

(1) CKSC_RCANC — CLK_RCANOSC Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0150_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCANSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 13.25 CKSC_RCANC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RCANSCSID [1:0]	Clock source control These bits select the ID of a clock source for CLK_RCANOSC ID = 00_B: Setting prohibited ID = 01_B: CLK_MOSC (default) ID = 10_B: CLK_MOSC/2 ID = 11_B: CLK_MOSC/4

CAUTION

Write this register only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).

(2) CKSC_RCANS — CLK_RCANOSC Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0158$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCANSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.26 CKSC_RCANS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	RCANSACT[1:0]	Clock source status for CLK_RCANOSC Indicates the ID of the selected clock source of the CLK_RCANOSC. The value indicated by these bits shows the actual ID that is currently selected. ID = 01_B: CLK_MOSC ID = 10_B: CLK_MOSC/2 ID = 11_B: CLK_MOSC/4

13.5.5.5 ADCJn Clock (CLK_ADC)

(1) CKSC_ADCC — CLK_ADC Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0160_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCSCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.27 CKSC_ADCC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ADCSCSID	Clock source control These bits select the ID of a clock source for CLK_ADC ID = 0_B: CLK_LSB ID = 1_B: CLK_LSB/2 (default)

(2) CKSC_ADCS — CLK_ADC Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0168_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCSACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.28 CKSC_ADCS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ADCSACT	Clock source status for CLK_ADC Indicates the ID of the selected clock source of the CLK_ADC. The value indicated by these bits shows the actual ID that is currently selected. ID = 0_B: CLK_LSB ID = 1_B: CLK_LSB/2

13.5.5.6 MSPI Clock (CLK_MSPI)

(1) CKSC_MSPIC — CLK_MSPI Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 0170_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSPISCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.29 CKSC_MSPIC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MSPISCSID	Clock source control These bits select the ID of a clock source for CLK_MSPI ID = 0: CLK_MOSC ID = 1: CLK_HSB (default)

CAUTION

Write this register only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).

(2) CKSC_MSPIS — CLK_MSPI Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 0178$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSPIS ACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.30 CKSC_MSPIS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	MSPISACT	Clock source status for CLK_MSPI Indicates the ID of the selected clock source of the CLK_MSPI. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CLK_MOSC ID = 1: CLK_HSB

13.5.5.7 WDTBA Clock (CLKA_WDT)

(1) CKSC_AWDTC — CLKA_WDT Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8200_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTSCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.31 CKSC_AWDTC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AWDTSCSID	Clock source control These bits select the ID of a clock source for CLKA_WDT ID = 0: CLK_LSIOSC/1 ID = 1: CLK_LSIOSC/128 (default)

(2) CKSC_AWDTS — CLKA_WDT Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8208$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTSACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.32 CKSC_AWDTS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	AWDTSACT	Clock source status for CLKA_WDT Indicates the ID of the selected clock source of the CLKA_WDT. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CLK_LSIOSC/1 ID = 1: CLK_LSIOSC/128

13.5.5.8 TAUJ Clock (CLKA_TAUJ)

(1) CKSC_ATAUJC— CLKA_TAUJ Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8210_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 13.33 CKSC_ATAUJC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ATAUJSCSID [1:0]	Clock source control These bits select the ID of a clock source for CLKA_TAUJ*1 ID = 00_B: CLK_LSIOSC ID = 01_B: CLK_HSIOSC/20 (default) ID = 10_B: CLK_MOSC ID = 11_B: CLK_HSB

Note 1. When the module operates in chip standby mode, select a clock source other than CLK_HSB.

CAUTION

Select CLK_MOSC only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).
Don't disable Main OSC while CLK_MOSC is selected for CLKA_TAUJ.

(2) CKSC_ATAUJS — CLKA_TAUJ Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8218$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.34 CKSC_ATAUJS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ATAUJSACT[1:0]	Clock source status for CLKA_TAUJ Indicates the ID of the selected clock source of the CLKA_TAUJ. The value indicated by these bits shows the actual ID that is currently selected. ID = 00$_B$: CLK_LSIOSC ID = 01$_B$: CLK_HSIOSC/20 ID = 10$_B$: CLK_MOSC ID = 11$_B$: CLK_HSB

13.5.5.9 RTCA Clock (CLKA_RTCA)

(1) CKSC_ARTCAC — CLKA_RTCA Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8220_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.35 CKSC_ARTCAC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARTCASCSID	Clock source control These bits select the ID of a clock source for CLKA_RTCA ID = 0: CLK_MOSC/16 ID = 1: CLK_LSIOSC (default)

CAUTION

Write this register only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).

(2) CKSC_ARTCAS — CLKA_RTCA Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8228$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.36 CKSC_ARTCAS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ARTCASACT	Clock source status for CLKA_RTCA Indicates the ID of the selected clock source of the CLKA_RTCA. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CLK_MOSC/16 ID = 1: CLK_LSIOSC

13.5.5.10 ADCJ2 Clock (CLKA_ADC)

(1) CKSC_AADCC — CLKA_ADC Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8230$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 13.37 CKSC_AADCC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCSCSID [1:0]	Clock source control These bits select the ID of a clock source for CLKA_ADC*1 ID = 00_B: CLK_MOSC ID = 01_B: CLK_HSIOSC/20 (default) ID = 10_B: CLK_LSB ID = 11_B: Setting prohibited

Note 1. When the module operates in chip standby mode, select a clock source other than CLK_LSB.

CAUTION

Select CLK_MOSC only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).
Don't disable Main OSC while CLK_MOSC is selected for CLKA_ADC.
CLK_MOSC can be selected only when the frequency is 16 MHz, 20 MHz, or 24 MHz.

(2) CKSC_AADCS — CLKA_ADC Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8238$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.38 CKSC_AADCS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCSACT[1:0]	Clock source status for CLKA_ADC Indicates the ID of the selected clock source of the CLKA_ADC. The value indicated by these bits shows the actual ID that is currently selected. ID = 00_B: CLK_MOSC ID = 01_B: CLK_HSIOSC/20 ID = 10_B: CLK_LSB

(3) CLKD_AADCC — CLKA_ADC Divider Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8240$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCDCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.39 CLKD_AADCC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AADCDCSID*1	Clock Divider control These bits select the ID of a clock divider for CLKA_ADC ID = 0: CKSC_AADCC selection/2 ID = 1: CKSC_AADCC selection/1 (default)

Note 1. Make sure the operation frequency of CLKA_ADC. For details, see **Section 55.4, A/D Converter Characteristics**.

(4) CLKD_AADCS — CLKA_ADC Divider Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8248$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCDACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.40 CLKD_AADCS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	AADCDACT	Clock divider status for CLKA_ADC Indicates the ID of the selected clock source of the CLKA_ADC. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CKSC_AADCC selection/2 ID = 1: CKSC_AADCC selection/1

13.5.5.11 FOUT0 Clock (EXTCLK0O)

(1) CKSC_FOUT0C — FOUT0 Clock Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8250_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT0SCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 13.41 CKSC_FOUT0C Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	FOUT0SCSID [2:0]	Source Clock Setting for EXTCLK0O*1 ID = 000_B: CLK_MOSC ID = 001_B: CLK_HSB (default) ID = 011_B: CLK_LSIOSC ID = 100_B: CLK_HSIOSC/20 ID = 110_B: CLK_MOSC ID = 111_B: CLK_MOSC Setting other than the above is prohibited.

Note 1. When the module operates in chip standby mode, select a clock source other than CLK_HSB.

CAUTION

Set this register when EXTCLK0O is stopped.

Select CLK_MOSC only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).

(2) CKSC_FOUT0S — FOUT0 Clock Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8258$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT0SACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.42 CKSC_FOUT0S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	FOUT0SACT [2:0]	Clock source status for FOUT0 Indicates the ID of the selected clock source of the FOUT0. The value indicated by these bits shows the actual ID that is currently selected. ID = 000_B: CLK_MOSC ID = 001_B: CLK_HSB ID = 011_B: CLK_LSIOSC ID = 100_B: CLK_HSIOSC/20 ID = 110_B: CLK_MOSC ID = 111_B: CLK_MOSC

(3) CLKD_FOUT0C — FOUT0 Clock Divider Control Register

This register defines the clock divisor.

Access: This register can be read or written in 32-bit units.

Address: FF98 8260_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FOUT0DIV[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.43 CLKD_FOUT0C Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	FOUT0DIV[9:0]	Clock Divider N 000_H: Clock output is stopped 001_H: N = 1 002_H: N = 2 : $3FE_H$: N = 1022 $3FF_H$: N = 1023

CAUTION

This register must not be written with a new value while the CLKD_FOUT0S.FOUT0SYNC is 0.

(4) CLKD_FOUT0S — FOUT0 Clock Divider Status Register

This register indicates the clock output status.

Access: This register can be read in 32-bit units.

Address: FF98 8268$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT0CLKACT	FOUT0SYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.44 CLKD_FOUT0S Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUT0CLKACT	Clock Divider Active 0: Frequency output is stopped. 1: Frequency output is ongoing.
0	FOUT0SYNC	Clock Divider Synchronized 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

13.5.5.12 FOUT1 Clock (EXTCLK1O)

(1) CKSC_FOUT1C — FOUT1 Clock Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8270$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT1SCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 13.45 CKSC_FOUT1C Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	FOUT1SCSID [2:0]	Source Clock Setting for EXTCLK1O*1 ID = 000_B: CLK_MOSC ID = 001_B: CLK_HSB (default) ID = 011_B: CLK_LSIOSC ID = 100_B: CLK_HSIOSC/20 ID = 110_B: CLK_MOSC ID = 111_B: CLK_MOSC Setting other than the above is prohibited.

Note 1. When the module operates in chip standby mode, select a clock source other than CLK_HSB.

CAUTION

Set this register when EXTCLK1O is stopped.

Select CLK_MOSC only when Main OSC is stable (MOSCS.MOSCEN = 1 & MOSCS.MOSCSTAB = 1).

(2) CKSC_FOUT1S — FOUT1 Clock Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8278$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT1SACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.46 CKSC_FOUT1S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	FOUT1SACT[2:0]	Clock source status for FOUT1 Indicates the ID of the selected clock source of the FOUT1. The value indicated by these bits shows the actual ID that is currently selected. ID = 000_B: CLK_MOSC ID = 001_B: CLK_HSB ID = 011_B: CLK_LSIOSC ID = 100_B: CLK_HSIOSC/20 ID = 110_B: CLK_MOSC ID = 111_B: CLK_MOSC

(3) CLKD_FOUT1C — FOUT1 Clock Divider Control Register

This register defines the clock divisor.

Access: This register can be read or written in 32-bit units.

Address: FF98 8280$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FOUT1DIV[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.47 CLKD_FOUT1C Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	FOUT1DIV[9:0]	Clock Divider N 000_H: Clock output is stopped 001_H: N = 1 002_H: N = 2 : $3FE_H$: N = 1022 $3FF_H$: N = 1023

CAUTION

This register must not be written with a new value while the CLKD_FOUT1S.FOUT1SYNC is 0.

(4) CLKD_FOUT1S — FOUT1 Clock Divider Status Register

This register indicates the clock output status.

Access: This register can be read in 32-bit units.

Address: FF98 8288$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUT1 CLKACT	FOUT1 SYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.48 CLKD_FOUT1S Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUT1CLKACT	Clock Divider Active 0: Frequency output is stopped. 1: Frequency output is ongoing.
0	FOUT1SYNC	Clock Divider Synchronized 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

13.5.5.13 WDTBn Clock (CLK_WDT)

(1) CKSC_WDTC — CLK_WDT Selector Control Register

Access: This register can be read or written in 32-bit units.

Address: FF98 8300$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTSCSID
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.49 CKSC_WDTC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	WDTSCSID	Clock source control These bits select the ID of a clock source for CLK_WDT ID = 0: CLK_HSIOSC/20 ID = 1: CLK_HSIOSC/640 (default)

CAUTION

This register is not initialized by DeepSTOP Reset.

(2) CKSC_WDTS — CLK_WDT Selector Status Register

Access: This register can be read in 32-bit units.

Address: FF98 8308$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTSACT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.50 CKSC_WDTS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	WDTSACT	Clock source status for CLK_WDT Indicates the ID of the selected clock source of the CLK_WDT. The value indicated by these bits shows the actual ID that is currently selected. ID = 0: CLK_HSIOSC/20 ID = 1: CLK_HSIOSC/640

CAUTION

This register is not initialized by DeepSTOP Reset.

13.5.6 Protection Register

(1) CLKKCPROT1 — Clock Controller Register Key Code Protection Register 1

The CLKKCPROT1 register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 0700$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 13.51 CLKKCPROT1 Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0.
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing to protected registers.
Write A5A5A501$_H$ to this register to enable writing to protected registers.

13.6 Operation

13.6.1 Clock Setting

13.6.1.1 Overview of Clock Selector/Divider Control Register

The clocks CLK_<*name*>, CLKA_<*name*> and EXTCLKnO (n = 0, 1) can be controlled by the following registers:

- Clock selector control registers
  These registers select the clock from the available source clocks.
  - CKSC_<*name*>C
  - CKSC_A<*name*>C
  - CKSC_FOUTnC (n = 0, 1)
- Clock divider control registers
  These registers specify the clock division ratio for the selected source clock.
  - CLKD_ <*name*> C
  - CLKD_A<*name*>C
  - CLKD_FOUTnC (n = 0, 1)
- Clock selector status registers/Clock divider status registers
  These registers return the currently active source clock selection and division status, respectively
  - CKSC_ <*name*>S/CLKD_ <*name*>S
  - CKSC_ A<*name*>S/CLKD_ A<*name*>S
  - CKSC_FOUTnS (n = 0, 1)/CLKD_FOUTnS (n = 0, 1)

NOTE

For details of the clocks with clock selector and divider, see **Table 13.52, List of Selectable Clocks**.

13.6.1.2 Setting Procedure for Clock

Procedure of setting up clock is described as below:

1. Clock selector setting
   - Select a source clock. (CKSC_ <*name*>C, CKSC_A<*name*>C, CKSC_FOUTnC (n = 0, 1))
   - Confirm completion of selection. (CKSC_ <*name*>S, CKSC_A<*name*>S, CKSC_FOUTnS (n = 0, 1))*1
2. Clock divider setting
   - Select a clock divider. (CLKD_ <*name*>C, CLKD_A<*name*>C, CLKD_FOUTnC (n = 0, 1))
   - Confirm completion of selection. (CLKD_ <*name*>S, CLKD_A<*name*>S, CLKD_FOUTnS (n = 0, 1))*2

Note 1. Continue processing after status register is updated with the new values written to control register.

Note 2. Continue processing after status register is updated with the new values written to control register (CLKD_AADCS = CLKD_AADCC).
Continue processing after status register indicates the synchronization with control register (CLKD_PLLS.PLLCLKDSYNC = 1, CLKD_FOUT0S.FOUT0SYNC = 1, CLKD_FOUT1S.FOUT1SYNC = 1).

CAUTION

The source clock to be selected must be operating before performing these settings. The behavior and performance are not guaranteed if setup is performed while the source clock is stopped.
Access to a peripheral module is prohibited while the clock is not supplied to the module.

13.6.2 Stopping the Clock in Chip Standby Mode

In chip standby modes (STOP mode, DeepSTOP mode, and Cyclic STOP mode), the peripheral clock which has the stop mask bit (MSR_<name>.STPMSK_<name>) can be configured to stop or continue in response to clock stop requests from the chip standby controller.

The clock stop mask bits are used to determine the operation status of the clock in chip standby mode: For details of chip standby mode, See **Section 15, Standby Controller (STBC)**.

The clocks CLK_CPU, CLK_SBUS and CLK_HBUS are stopped in STOP mode and Cyclic STOP mode. The clocks CLK_HSB, CLK_LSB for CLKA_<name> and EXTCLKnO (n = 0, 1) are stopped in DeepSTOP mode.

CAUTION

If the source clock of the peripheral clock stops during chip standby mode, MSR_<name>.STPMSK_<name> should be set to 0.

13.6.3 Stopping the Clock in Module Standby Mode

It is necessary to cancel the module standby mode in order to operate some peripheral modules.

For details of module standby mode, See **Section 15, Standby Controller (STBC)**.

13.6.4 Clock Settings

The following table shows a selectable source clock, a frequency division ratio, and a register to be used for each clock.

Table 13.52 List of Selectable Clocks (1/2)

Symbol	Clock Selector Control Register		Clock Divider Control Register		Maximum Frequency	Functional Module
CLK_CPU CLK_SBUS CLK_HBUS CLK_UHSB CLK_HSB CLK_LSB	CKSC_CPUC	CLK_PLLO	CLKD_PLLC	Refer to **Table 13.5**	400/320/240 MHz The maximum clock frequency depends on the Option Byte CKDIVMD.	CPU System
		CLK_IOSC				
CLK_RLIN	CKSC_RLINC	CLK_MOSC	—	**1/1**	80 MHz	RLIN3
		CLK_MOSC/4				
		CLK_MOSC/8				
		CLK_HSB				
CLK_RCANOSC	CKSC_RCANC	**CLK_MOSC**	—	**1/1**	40 MHz	RS-CANFD
		CLK_MOSC/2				
		CLK_MOSC/4				
CLK_ADC	CKSC_ADCC	CLK_LSB	—	**1/1**	40MHz	ADCJ0,1
		CLK_LSB/2				
CLK_MSPI	CKSC_MSPIC	CLK_MOSC	—	**1/1**	80 MHz	MSPI
		CLK_HSB				
CLK_ECMCNT	—	**CLK_HSIOSC/20**	—	**1/1**	10 MHz	ECM
CLK_WDT	CKSC_WDTC	CLK_HSIOSC/20	—	**1/1**	10 MHz	WDTB0,1,2,3 SWDTA ICUM_WDTA0
		CLK_HSIOSC/640				
CLKA_WDT	CKSC_AWDTC	CLK_LSIOSC	—	**1/1**	240 kHz	WDTBA
		CLK_LSIOSC/128				
CLKA_TAUJ	CKSC_ATAUJC	CLK_MOSC	—	**1/1**	80 MHz	TAUJ2 TAUJ3
		CLK_HSIOSC/20				
		CLK_LSIOSC				
		CLK_HSB				
CLKA_RTCA	CKSC_ARTCAC	CLK_MOSC/16	—	**1/1**	2.5 MHz	RTCA
		CLK_LSIOSC				
CLKA_ADC	CKSC_AADCC	CLK_MOSC	CLKD_AADCC	**1/1**	40 MHz	ADCJ2
		CLK_HSIOSC/20		1/2		
		CLK_LSB				
EXTCLK0O	CKSC_FOUT0C	CLK_MOSC	CLKD_FOUT0C	1/1 ~ 1/1023	24 MHz	FOUT0
		CLK_HSIOSC/20				
		CLK_LSIOSC				
		CLK_HSB		**stop**		

Table 13.52 List of Selectable Clocks (2/2)

Symbol	Clock Selector Control Register		Clock Divider Control Register		Maximum Frequency	Functional Module
EXTCLK1O	CKSC_FOUT1C	CLK_MOSC	CLKD_FOUT1C	1/1 ~ 1/1023	24 MHz	FOUT1
		CLK_HSIOSC/20				
		CLK_LSIOSC				
		CLK_HSB		**Stop**		
CLKA_LPS*1	—	**CLK_LSIOSC or CLK_HSIOSC/20**	—	1/1	10 MHz	LPS

Note: The items written in bold are the initial setting clocks for each register.

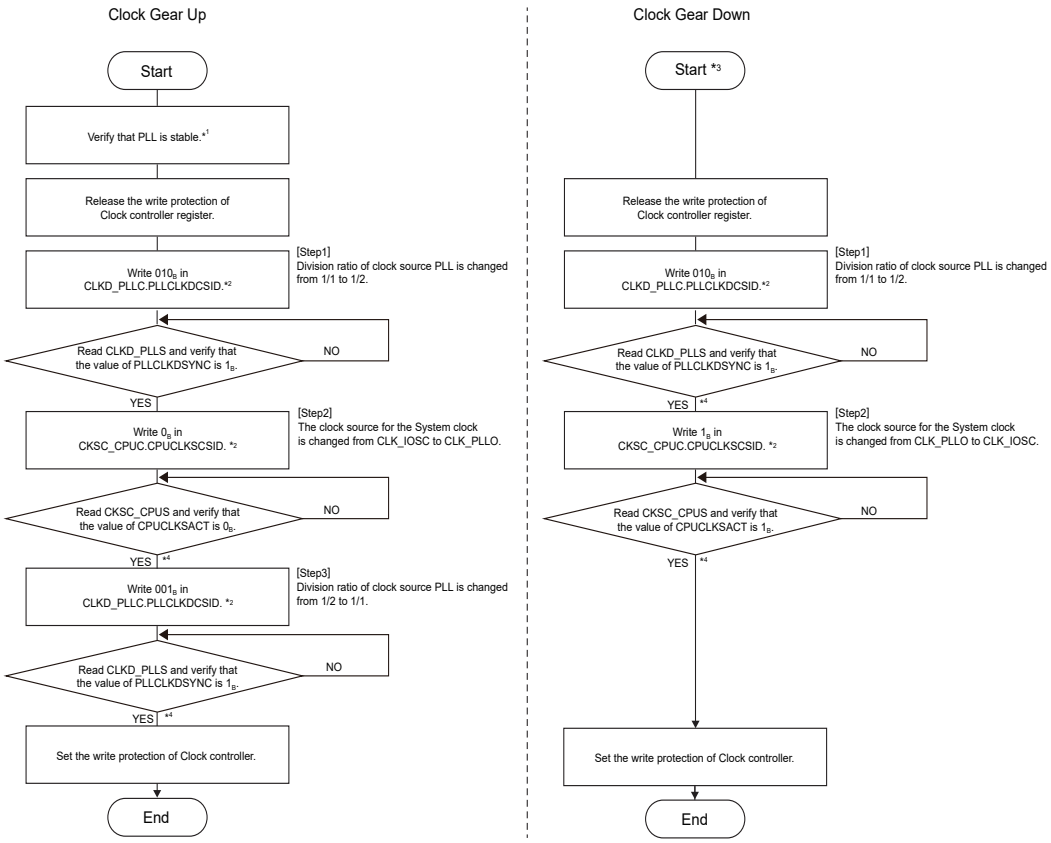
Note 1. When CLK_HSIOSC stops in chip standby mode , CLKA_LPS is CLK_LSIOSC. In other cases, CLKA_LPS is CLK_HSIOSC/20.

CAUTION

Refer to Section 13.6.2 and Section 13.6.3 and Stop Mask Register (MOSCSTPM, HSOSCSTPM, PLLSTPM) to stop the clocks.

13.6.5 Sequence for Shifting the CPU System Clock Gear Up/Down

CPU clock (CLK_CPU) is CLK_HSIOSC/2 when CPU startup.
Follow **Figure 13.9** below when shifting the CPU System clock .



Note: There are modules which can not operate when either CLK_IOSC or CLK_PLL/2 is selected to CLK_SYS.

Note 1. After release from the reset state, verify that PLL operation is stable before executing the sequence to shift the clock gear up. See **Section 13.5.4.2, PLLS — PLL Status Register**. For enabling PLL, see **Section 13.5.4.1, PLLE — PLL Enable Register**.

Note 2. The CKSC_CPUC and CLKD_PLLC registers are protected by the CLKKCPROT1 register. Execute the key code protection unlocking when setting these registers.

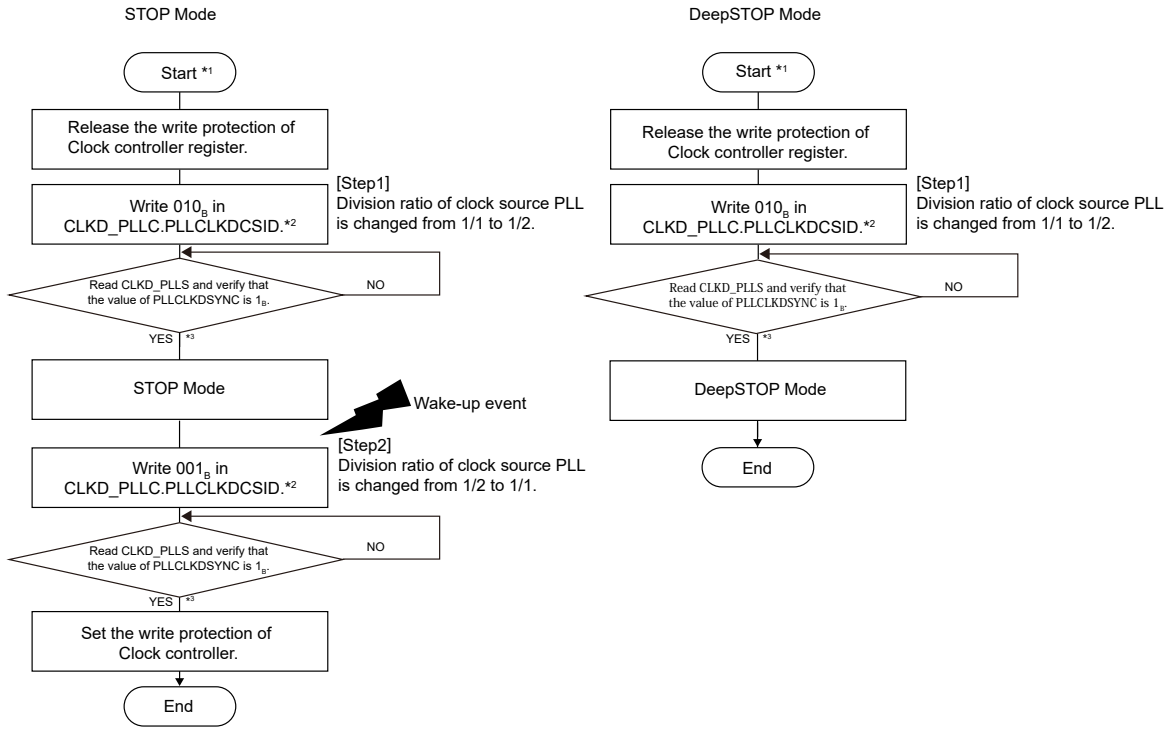
Note 3. CLK_PLLO is selected as the clock source for the System clock, and No division is selected as the division ratio of clock source PLL.

Note 4. Wait 100 µs after verifying

Figure 13.9 Shifting the CPU System Clock Gear Up/Down

13.6.6 CPU System Clock Setting in STOP/DeepSTOP Mode

Transition before STOP/DeepSTOP mode, follow the procedure below in case that CLK_PLLO is selected as the clock source for the System clock, and No division is selected as the division ratio of clock source PLL.



Note 1. CLK_PLLO is selected as the clock source for the System clock, and No division is selected as the division ratio of clock source PLL.

Note 2. The CKSC_CPUC and CLKD_PLLC registers are protected by the CLKKCPROT1 register. Execute the key code protection unlocking when setting these registers.

Note 3. Wait 100 µs after verifying

Figure 13.10 CPU System Clock Setting in STOP/DeepSTOP Mode

Section 14 Clock Monitor (CLMA)

The following sections describe clock monitor A (CLMA).

The first section describes the attributes specific to the RH850/U2A-EVA microcontrollers, including the number of channels, register base addresses, and input/output signal names.

The ensuing sections describe the functions relevant to all operations.

14.1 Features of RH850/U2A-EVA CLMA

14.1.1 Number of Channels

The RH850/U2A-EVA microcontrollers incorporate CLMA with the following number of channels.

Table 14.1 Number of Channels

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of channels	10 (n = 0 to 9)	10 (n = 0 to 9)	10 (n = 0 to 9)	10 (n = 0 to 9)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)
Name	CLMAn									

14.1.2 Register Base Addresses

The CLMA base addresses are listed in the following table.

The CLMA register addresses are expressed as offsets from the base addresses.

Table 14.2 Register Base Addresses

Base Address Name	Base Address	Bus Group
<CLMAC_base>*1	FF98 9000$_H$	P-Bus Group 2L
<CLMA0_base>	FF98 9100$_H$	P-Bus Group 2L
<CLMA1_base>	FF98 9200$_H$	P-Bus Group 2L
<CLMA2_base>	FF98 9300$_H$	P-Bus Group 2L
<CLMA3_base>	FF98 1800$_H$	P-Bus Group 2L
<CLMA4_base>	FF98 1900$_H$	P-Bus Group 2L
<CLMA5_base>	FF98 1A00$_H$	P-Bus Group 2L
<CLMA6_base>	FF98 1B00$_H$	P-Bus Group 2L
<CLMA7_base>	FF98 1C00$_H$	P-Bus Group 2L
<CLMA8_base>	FF98 1D00$_H$	P-Bus Group 2L
<CLMA9_base>	FF98 1E00$_H$	P-Bus Group 2L

Note 1. Common registers CLMATEST, CLMATESTS.

14.1.3 Clock Supply

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.
Each clock frequency see **Section 13, Clock Controller**.

Table 14.3 Clock Supply

Channel Name	CLMATMON (monitored clock)	CLMATSMP (sampling clock)	PCLK (register access clock)
CLMA0	CLK_MOSC	CLK_HSIOSC/400	CLK_LSB
CLMA1	CLK_WDT (HS IntOSC 1/20, 1/640)	CLK_LSIOSC/2	CLK_LSB
CLMA2	CLK_LSIOSC	CLK_HSIOSC/1600	CLK_LSB
CLMA3	CLK_LSB	CLK_MOSC/8	CLK_LSB
CLMA4	CLK_LSB	CLK_HSIOSC/20	CLK_LSB
CLMA5	CLK_UHSB/2 (Target: GTM Main clock)	CLK_LSB/4	CLK_LSB
CLMA6	CLK_CPU/4 (Target: PE0 checker clock)	CLK_LSB/4	CLK_LSB
CLMA7	CLK_CPU/4 (Target: PE1 checker clock)	CLK_LSB/4	CLK_LSB
CLMA8	CLK_CPU/4 (Target: PE2 checker clock)	CLK_LSB/4	CLK_LSB
CLMA9	CLK_CPU/4 (Target: PE3 checker clock)	CLK_LSB/4	CLK_LSB

NOTE

- The PLL lock loss error status can be detected by CLMA3, because the PLL clock is monitored by these clock monitors.
  (When the PLL clock is selected as the system clock (CLK_SYS).)
- CLMA3 needs to execute a special flow after reset release. Refer to **Section 14.7.1, Notes on CLMA3 after Reset release** for details.
- When CLK_SYS source is PLL clock, CLMA3 can be used.
- When CLK_SYS source is CLK_IOSC, CLMA4 can be used.
- When DCLS of PEn (n = 1 to 3) is disabled by setting bits 19 to 17 in FLASH Option Byte 9, CLMAn (n = 7 to 9) can not be used.
  For details of FLASH Option Byte 9, see **Section 51.12.15, OPBT9 — Option Byte 9**.
- For procedures of clock change, refer to **Section 14.7.3, Notes on Procedures to Clock Change**.

14.1.4 Interrupt Requests and Error Notifications

This module has no interrupt requests.

The error notifications of this module are listed in the following table.

Table 14.4 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
Clock monitor error for CLK_MOSC	CLMA0 over/under frequency error	48	—
Clock monitor error for CLK_WDT	CLMA1 over/under frequency error	49	—
Clock monitor error for CLK_LSIOSC	CLMA2 over/under frequency error	50	—
Clock monitor error for CLK_LSB	CLMA3 over/under frequency error	51	—
	CLMA4 over/under frequency error		
Clock monitor error for CLK_UHSB	CLMA5 over/under frequency error	52	—
Clock monitor error for CLK_CPU (PE0)	CLMA6 over/under frequency error	229	—
Clock monitor error for CLK_CPU (PE1)	CLMA7 over/under frequency error	261	—
Clock monitor error for CLK_CPU (PE2)	CLMA8 over/under frequency error	293	—
Clock monitor error for CLK_CPU (PE3)	CLMA9 over/under frequency error	325	—

14.1.5 Internal Input/Output Signals

Table 14.5 Internal Input/Output Signals

Clock monitor	Output to
CLMA0	ECM Error Input No.48
CLMA1	ECM Error Input No.49
CLMA2	ECM Error Input No.50
CLMA3	ECM Error Input No.51
CLMA4	ECM Error Input No.51
CLMA5	ECM Error Input No.52
CLMA6	ECM Error Input No.229
CLMA7	ECM Error Input No.261
CLMA8	ECM Error Input No.293
CLMA9	ECM Error Input No.325

Note: CLMA notifies ECM of high detection error and low detection error.

14.2 Overview

14.2.1 Block Diagram

The main components of the clock monitor are shown in **Figure 14.1**.

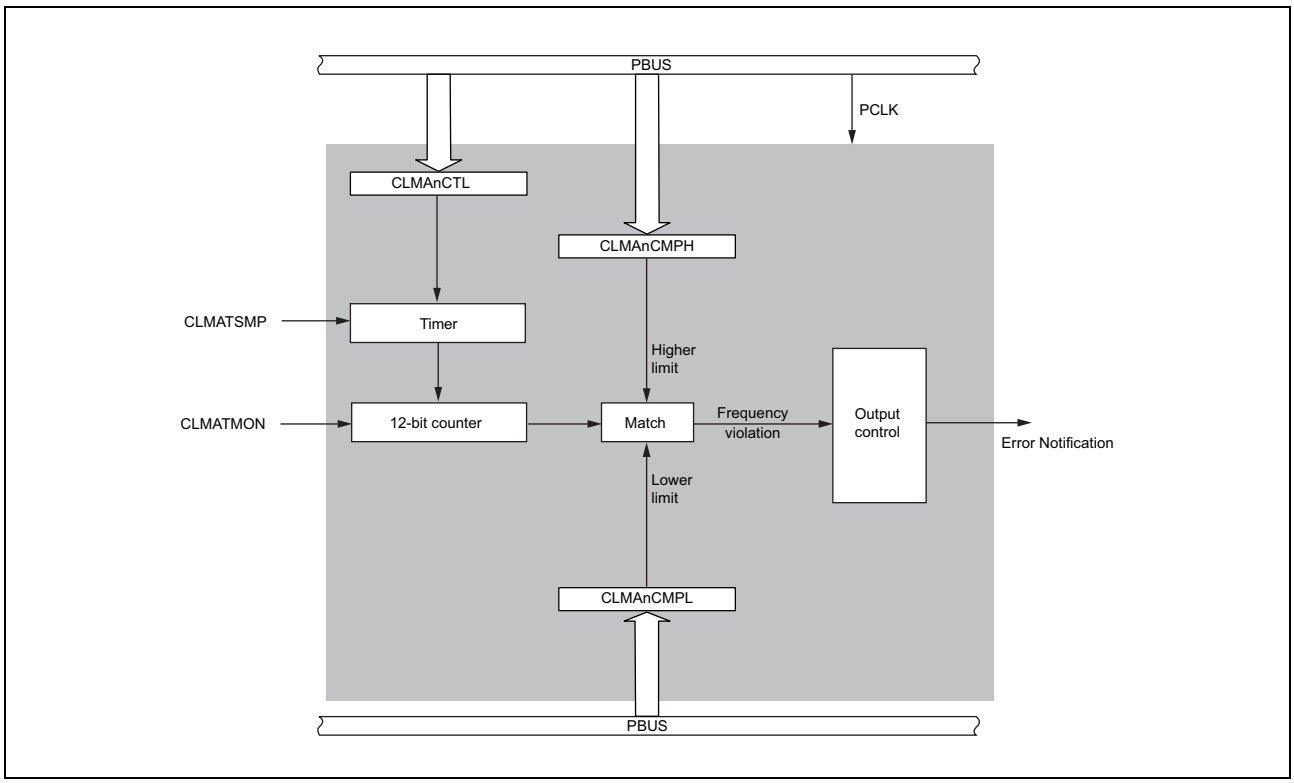


Figure 14.1 Block Diagram of Clock Monitor

14.2.2 Functional Overview

Clock monitor CLMA detects frequency abnormalities in the monitored clock.

It uses sampling clock CLMATSMP to monitor whether the frequency of input clock CLMATMON is within a specific range.

Upon detection of an abnormal clock, CLMAn sends the error notification to ECM.

The accuracy of each clock monitor is shown below.

Table 14.6 Accuracy of clock frequency used by Clock Monitor

Channel Name	CLMATMON (monitored clock)	CLMATSMP (sampling clock)
CLMA0	±0.1% *1	±5.0%
CLMA1	±5.0%	±10%
CLMA2	±10%	±5.0%
CLMA3	±0.1% *1	±0%
CLMA4	±0.1% *1*2	±0%
CLMA5	±0.1% *1	±0%
CLMA6	±0.1% *1	±0%
CLMA7	±0.1% *1	±0%
CLMA8	±0.1% *1	±0%
CLMA9	±0.1% *1	±0%

Note 1. Depends on the accuracy of the external parts.
Note 2. This is the value of “CLK_LSB = CLK_IOSC”.

14.3 Enabling CLMA

To enable CLMAn, CLMAnCTL which is a register to control CLMAn should be set 01_H. The control register CLMAnCTL is a write protection register to enable CLMAn. Refer to **Section 14.6.1** about procedure to enable CLMAn.

CLMA0 to CLMA2 are in the AWO area. When the monitored clock or sampling clock is stopped by transition to Chip Standby mode (MOSCSTPM.MOSCSTPMSK = 0, HSOSCSTPM.HSOSCSTPMSK = 0), the corresponding clock monitor is automatically stopped.

After returning from Chip Standby mode, and the monitored clock and sampling clock start oscillation again and become stable, the clock monitor also starts operation.

CLMA3 to CLMA9 are in the ISO area, and the software must stop these CLMAs before the system enters Chip Standby mode.

NOTE

- CLMAn can only be disabled by reset. Writing 0 to CLMAnCTL cannot disable CLMAn.
- The initial value of CLMAnCTL is 00_H. (00_H means CLMAn is disabled)

14.4 Functions

The Clock Monitor CLMAn is used to verify whether the frequency of a clock (CLMATMON) is within the specified range.

14.4.1 Detection of Abnormal Clock Frequencies

14.4.1.1 Detection Method

(1) CLMAn counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMP and then compares the counter with the configured limits:
- CLMAnCMPL.CLMAnCMPL[11:0] defines the lower limit.
- CLMAnCMPH.CLMAnCMPH[11:0] defines the higher limit.

(2) When the frequency of CLMATMON is lower than the limit, the counter falls below CLMAnCMPL.CLMAnCMPL[11:0].

(3) When the frequency of CLMATMON is higher than the limit, the counter exceeds CLMAnCMPH.CLMAnCMPH[11:0].

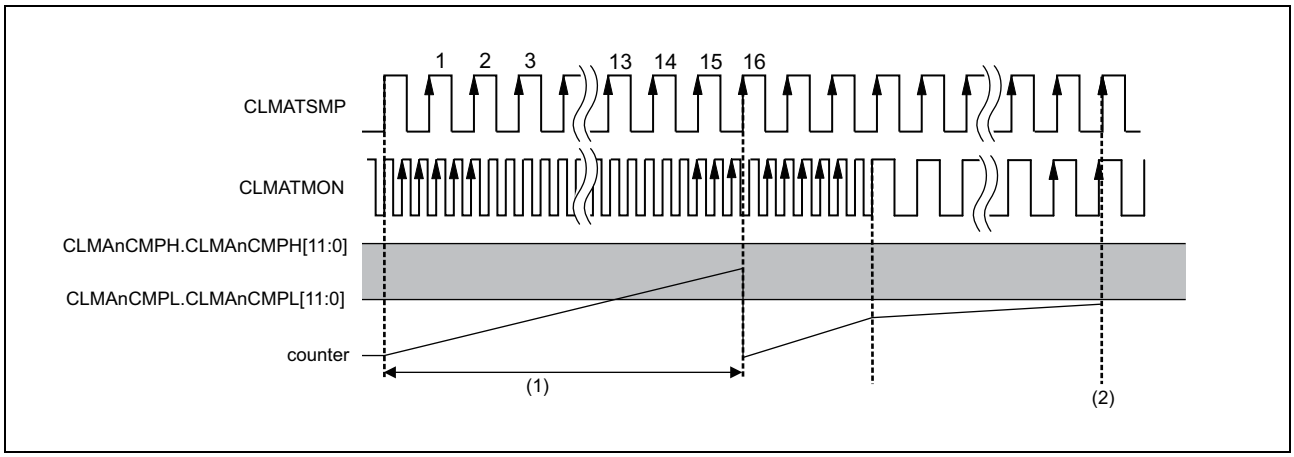


Figure 14.2 Example: $f_{CLMATMON}$ is Lower than the specified limit.

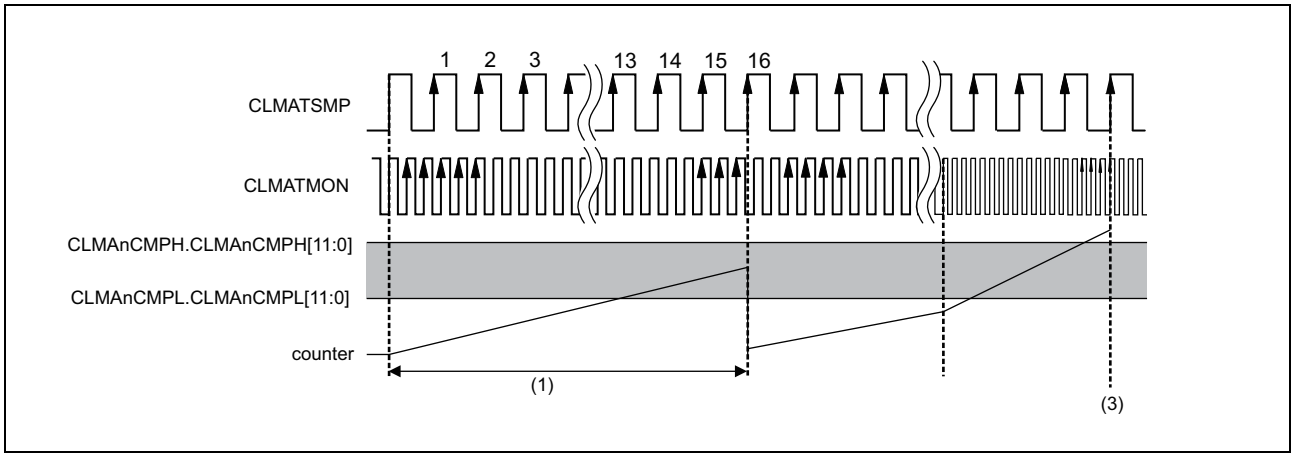


Figure 14.3 Example: $f_{CLMATMON}$ is Higher than the specified limit

NOTE

Even if the $f_{CLMATMON}$ exceeds or falls below the specified limits during a sample period, the counter might be within the valid range.
Abnormal $f_{CLMATMON}$ is detected after one sampling interval later.

14.4.1.2 Calculation of limits CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]

The compare registers CLMAnCMPL and CLMAnCMPH are configured with the minimum and maximum number of clock cycles of CLMATMON that are assumed to be valid within 16 cycles of the sampling clock CLMATSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMATSMP}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the limit values can be calculated by the following formulas:

$$\text{Lower limit} = N_{min} = \frac{f_{CLMATMON(min)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Higher limit} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

Example

For $f_{CLMATSMP}$ = 240 kHz (±10%)/2 and $f_{CLMATMON}$ = 200 MHz (±5.0%) /640 the recommended limit values are the following:

$$N_{min} = (296.9/132) \times 16 - 1$$
$$= 34.98$$
$$CLMAnCMPL = 34 = 0022_H$$

$$N_{max} = (328.1/108) \times 16 + 1$$
$$= 49.61$$
$$CLMAnCMPH = 50 = 0032_H$$

Minimum limits

The following restrictions must be taken into account:

- $CLMAnCMPL \geq 0001_H$
- $CLMAnCMPH \geq CLMAnCMPL + 0003_H$

14.4.2 Error Notification

Once error is detected, i.e. frequency of monitored clock (CLMATMON) exceeds the higher limit or falls below the lower limit, CLMAn sends the error notification to ECM.

In addition, the error notification is not negated until CLMAn is reset.

14.4.3 Self-Test

This MCU implements a self-diagnosis function for the clock monitors. It allows isolating the clock monitor from the system and executing functional test patterns via software. Each clock monitor can be tested individually. For performing these tests, the clock monitor inputs can be controlled by dedicated control registers, while its outputs can be observed in status registers.

Two registers are implemented for the self-diagnosis of the Clock Monitor, the Clock Monitor Test Register (CLMATEST) and the Clock Monitor Test Status Register (CLMATESTS). Refer to the corresponding register descriptions for details.

14.5 Registers

14.5.1 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

Registers CLMAKCPROT have to be written first in order to unlock protected registers and allow change of register content.

14.5.2 List of Registers

The following table lists the CLMA registers.

<CLMAn_base> is defined in **Section 14.1.2, Register Base Addresses**.

Table 14.7 List of Registers

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
<CLMAC_base> + 00_H	CLMATEST	Clock Monitor Test Register	32	0000 0000_H	PBG20 #2	CLMAKCPROT
<CLMAC_base> + 04_H	CLMATESTS	Clock Monitor Test Status Register	32	0000 0000_H	PBG20 #2	
<CLMAn_base> + 00_H	CLMAnCTL	CLMAn Control Register	8	00_H	PBG20 #2	CLMAKCPROT
<CLMAn_base> + 08_H	CLMAnCMPL	CLMAn Comparison Register L	16	0001_H	PBG20 #2	
<CLMAn_base> + $0C_H$	CLMAnCMPH	CLMAn Comparison Register H	16	$03FF_H$	PBG20 #2	
FF98 2700_H	CLMAKCPROT	Clock Monitor Register Key Code Protection Register	32	0000 0000_H	PBG20 #2	

Note: (n = 0 to 9)

14.5.3 Reset of Registers

Table 14.8 Register reset conditions

Channel Name	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
CLMAC	√	√	√	√	—	—	—
CLMA0	√	√	√	√	—	—	—
CLMA1	√	√	√	√	—	—	—
CLMA2	√	√	√	√	—	—	—
CLMA3	√	√	√	√	√	—	—
CLMA4	√	√	√	√	√	—	—
CLMA5	√	√	√	√	√	—	—
CLMA6	√	√	√	√	√	—	—
CLMA7	√	√	√	√	√	—	—
CLMA8	√	√	√	√	√	—	—
CLMA9	√	√	√	√	√	—	—
CLMAKCPROT	√	√	√	—	√	—	—

CLMATEST.RESCLM can also reset each clock monitor (CLMAn). Then, before clock frequency is changed by register for clock during operations, it is necessary to go through special procedure and to retry setting parameters for CLMAn.

About special procedure, refer to **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM**.

Table 14.9 Individual reset conditions

Register Name	CLMATEST.RESCLM Reset Target									
	CLMA0T ESEN = 1	CLMA1T ESEN = 1	CLMA2T ESEN = 1	CLMA3T ESEN = 1	CLMA4T ESEN = 1	CLMA5T ESEN = 1	CLMA6T ESEN = 1	CLMA7T ESEN = 1	CLMA8T ESEN = 1	CLMA9T ESEN = 1
CLMATEST	—	—	—	—	—	—	—	—	—	—
CLMATESTS	√	√	√	√	√	√	√	√	√	√
CLMA0CTL	√	—	—	—	—	—	—	—	—	—
CLMA0CMPL	√	—	—	—	—	—	—	—	—	—
CLMA0CMPH	√	—	—	—	—	—	—	—	—	—
CLMA1CTL	—	√	—	—	—	—	—	—	—	—
CLMA1CMPL	—	√	—	—	—	—	—	—	—	—
CLMA1CMPH	—	√	—	—	—	—	—	—	—	—
CLMA2CTL	—	—	√	—	—	—	—	—	—	—
CLMA2CMPL	—	—	√	—	—	—	—	—	—	—
CLMA2CMPH	—	—	√	—	—	—	—	—	—	—
CLMA3CTL	—	—	—	√	—	—	—	—	—	—
CLMA3CMPL	—	—	—	√	—	—	—	—	—	—
CLMA3CMPH	—	—	—	√	—	—	—	—	—	—
CLMA4CTL	—	—	—	—	√	—	—	—	—	—
CLMA4CMPL	—	—	—	—	√	—	—	—	—	—
CLMA4CMPH	—	—	—	—	√	—	—	—	—	—
CLMA5CTL	—	—	—	—	—	√	—	—	—	—
CLMA5CMPL	—	—	—	—	—	√	—	—	—	—
CLMA5CMPH	—	—	—	—	—	√	—	—	—	—
CLMA6CTL	—	—	—	—	—	—	√	—	—	—
CLMA6CMPL	—	—	—	—	—	—	√	—	—	—
CLMA6CMPH	—	—	—	—	—	—	√	—	—	—
CLMA7CTL	—	—	—	—	—	—	—	√	—	—
CLMA7CMPL	—	—	—	—	—	—	—	√	—	—
CLMA7CMPH	—	—	—	—	—	—	—	√	—	—
CLMA8CTL	—	—	—	—	—	—	—	—	√	—
CLMA8CMPL	—	—	—	—	—	—	—	—	√	—
CLMA8CMPH	—	—	—	—	—	—	—	—	√	—
CLMA9CTL	—	—	—	—	—	—	—	—	—	√
CLMA9CMPL	—	—	—	—	—	—	—	—	—	√
CLMA9CMPH	—	—	—	—	—	—	—	—	—	√
CLMAKCPROT	—	—	—	—	—	—	—	—	—	—

14.5.4 CLMATEST – Clock Monitor Test Register

This register is used for the self-test of the clock monitors. Each Clock Monitor can be tested individually.

Access: This register can be read or written in 32-bit units.

Address: <CLMAC_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CLMA9 TESEN	CLMA8 TESEN	CLMA7 TESEN	CLMA6 TESEN	CLMA5 TESEN	CLMA4 TESEN	CLMA3 TESEN	CLMA2 TESEN	CLMA1 TESEN	CLMA0 TESEN	ERRMSK	MONCLKMSK	RESCLM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.10 CLMATEST Register contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	CLMA9TESEN	CLMA9 self-test enable/disable 0: Test disabled. 1: Test enabled.
11	CLMA8TESEN	CLMA8 self-test enable/disable 0: Test disabled. 1: Test enabled.
10	CLMA7TESEN	CLMA7 self-test enable/disable 0: Test disabled. 1: Test enabled.
9	CLMA6TESEN	CLMA6 self-test enable/disable 0: Test disabled. 1: Test enabled.
8	CLMA5TESEN	CLMA5 self-test enable/disable 0: Test disabled. 1: Test enabled.
7	CLMA4TESEN	CLMA4 self-test enable/disable 0: Test disabled. 1: Test enabled.
6	CLMA3TESEN	CLMA3 self-test enable/disable 0: Test disabled. 1: Test enabled.
5	CLMA2TESEN	CLMA2 self-test enable/disable 0: Test disabled. 1: Test enabled.
4	CLMA1TESEN	CLMA1 self-test enable/disable 0: Test disabled. 1: Test enabled.
3	CLMA0TESEN	CLMA0 self-test enable/disable 0: Test disabled. 1: Test enabled.

Table 14.10 CLMATEST Register contents (2/2)

Bit Position	Bit Name	Function
2	ERRMSK *1	CLMA test error mask setting Asserting this bit prevents the detected error from being forwarded into the device. CLMA test reset signal mask setting 0: Error signal generation enabled 1: Error signal generation disabled (masked)
1	MONCLKMSK*1	Monitor clock mask setting Fixes the clock input to the CLMAn to low level. 0: Monitored clock enabled 1: Monitored clock disabled (masked)
0	RESCLM *1	CLMAn test reset signal control Asserting this bit re-initializes the Clock Monitor. 0: Reset signal for CLMAn is released. 1: Reset signal for CLMAn is asserted.

Note 1. These bits are valid for CLMAn which is in self-test mode by setting “1” to CLMAnTESEN.

14.5.5 CLMATESTS – Clock Monitor Test Status Register

This register is used for the self-test of the clock monitors. It monitors the error detection flags which are otherwise forwarded to the ECM module. Once error is detected, this register keeps the status until CLMAn is reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CLMAC_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLMA9 ERRS	CLMA8 ERRS	CLMA7 ERRS	CLMA6 ERRS	CLMA5 ERRS	CLMA4 ERRS	CLMA3 ERRS	CLMA2 ERRS	CLMA1 ERRS	CLMA0 ERRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.11 CLMATESTS Register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9	CLMA9ERRS	CLMA9 error status 0: No error detected 1: Error detected
8	CLMA8ERRS	CLMA8 error status 0: No error detected 1: Error detected
7	CLMA7ERRS	CLMA7 error status 0: No error detected 1: Error detected
6	CLMA6ERRS	CLMA6 error status 0: No error detected 1: Error detected
5	CLMA5ERRS	CLMA5 error status 0: No error detected 1: Error detected
4	CLMA4ERRS	CLMA4 error status 0: No error detected 1: Error detected
3	CLMA3ERRS	CLMA3 error status 0: No error detected 1: Error detected
2	CLMA2ERRS	CLMA2 error status 0: No error detected 1: Error detected
1	CLMA1ERRS	CLMA1 error status 0: No error detected 1: Error detected
0	CLMA0ERRS	CLMA0 error status 0: No error detected 1: Error detected

14.5.6 CLMAnCTL – CLMAn Control Register

This register enables the clock monitor CLMAn.

Access: This register can be read or written in 8-bit units

Address: <CLMAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnCLME
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 14.12 CLMAnCTL Register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CLMAnCLME*1	Enable or disables the clock monitor. 0: Disables CLMAn. 1: Enables CLMAn. This bit can only be cleared by a reset.

Note 1. (n = 0 to 9)

14.5.7 CLMAnCMPL – CLMAn Comparison Register L

This register specifies the lower limit of monitored clock frequency.

Write access is permitted only when the CLMAn is disabled (CLMAnCTL.CLMAnCLME = 0).

For details, see **Section 14.4.1.2, Calculation of limits CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + 08_H

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPL[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.13 CLMAnCMPL Register contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPL*1	Specifies the lower limit • The recommended value is $f_{CLMATMON\ (min)}$ / $f_{CLMATSMP\ (max)}$ × 16 - 1. • The minimum value is 0001_H.

Note 1. (n = 0 to 9)

14.5.8 CLMAnCMPH – CLMAn Comparison Register H

This register specifies the higher limit of monitored clock frequency.

Write access is permitted only when the CLMAn is disabled (CLMAnCTL.CLMAnCLME = 0).

For details, see **Section 14.4.1.2, Calculation of limits CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + $0C_H$

Value after reset: $03FF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPH[11:0]											
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.14 CLMAnCMPH Register contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPH[*1]	Specifies the higher limit • The recommended value is $f_{CLMATMON\ (max)}$ / $f_{CLMATSMP\ (min)}$ × 16 + 1. • The minimum value is CLMAnCMPL + 0003_H.

Note 1. (n = 0 to 9)

14.5.9 CLMAKCPROT – Clock Monitor Register Key Code Protection Register

This register is used for protection against writing to the target registers due to program malfunction and the like. See **Table 14.7, List of Registers** for the target registers.

Access: This register can be read or written in 32-bit units.

Address: FF98 2700$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 14.15 CLMAKCPROT Register contents

Bit Position	Bit Name	Function
31 to 1	KCPROT*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0. *1
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing to protected registers.
Write A5A5A501$_H$ to this register to enable writing to protected registers.

14.6 Operation

14.6.1 Procedures to Enable CLMAn

[Procedure]

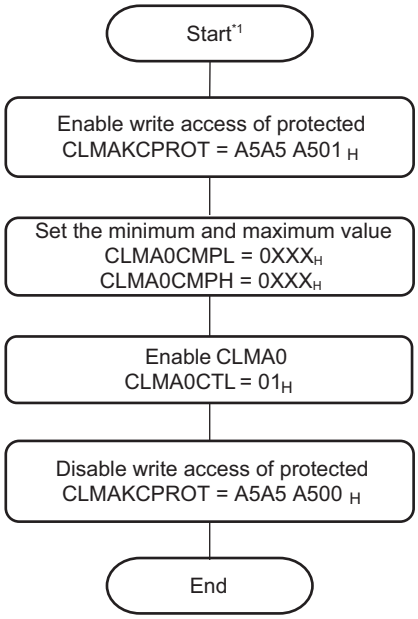
(1) Enable write access of protected registers with setting of CLMAKCPROT = A5A5 $A501_H$.

(2) Set the minimum and maximum number of clock cycles to CLMAnCMPL and CLMAnCMPH.

(3) Enable CLMAn by setting 01_H to CLMAnCTL.CLMAnCLME.

(4) Disable write access of protected registers with setting of CLMAKCPROT.KCE = A5A5 $A500_H$.

Example case: Enable CLMA0



Note 1. Initialize CLMA3 and clear the error status of ECM error source status register after reset release of CLMA3. See **Section 14.7.1, Notes on CLMA3 after Reset release** for details.

Figure 14.4 Flow Chart of the Enable CLMA0

CAUTION

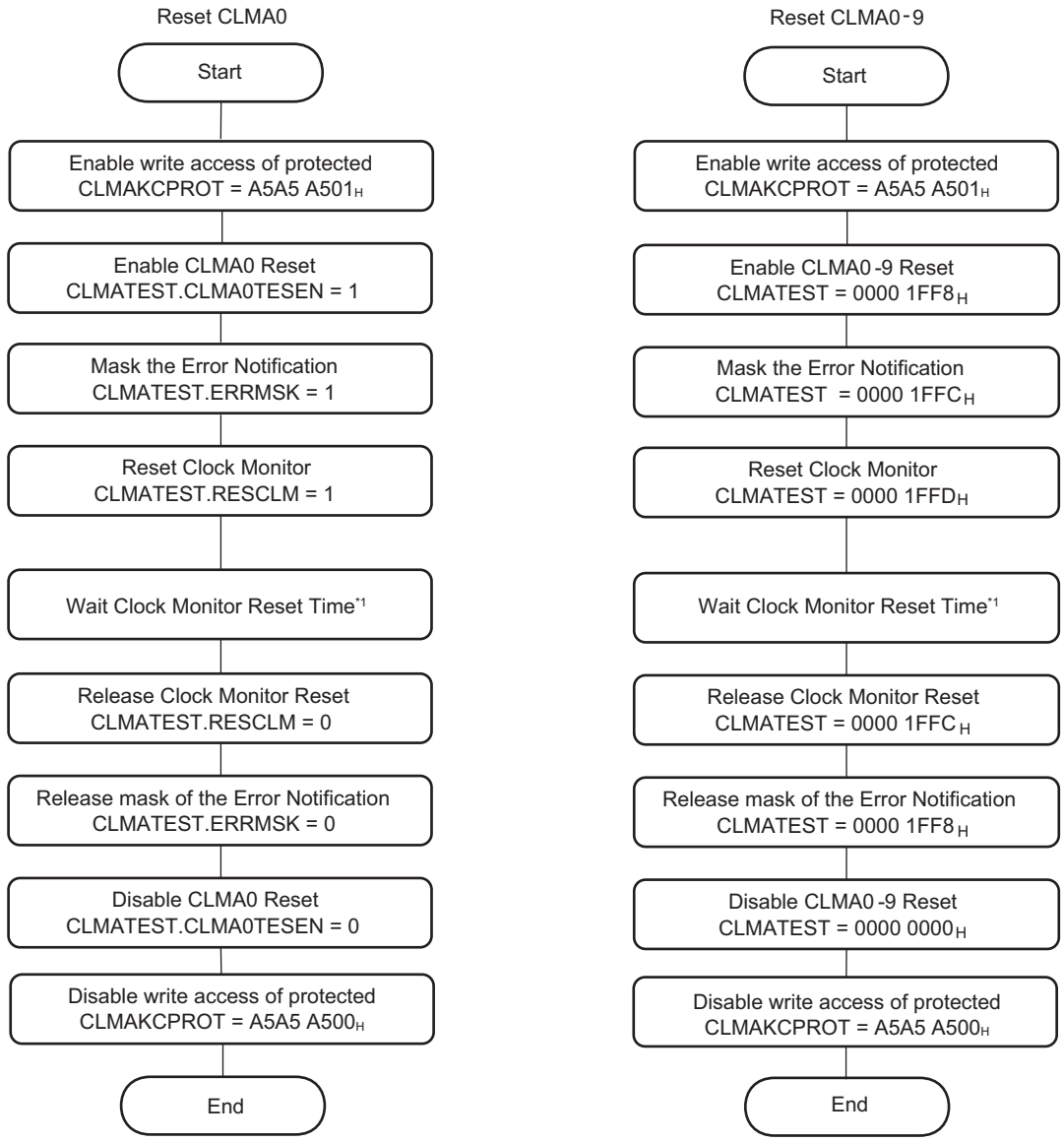
- **CLMA0 and CLMA3 operation must be started after users check stable state of Main OSC by referring to MOSCS.MOSCSTAB = 1.**
- **CLMA0, CLMA1, CLMA2 and CLMA4 operation must be started after users check stable state of HS IntOSC by referring to HSOSCS.HSOSCSTAB = 1.**
- **CLMAn (n = 3 to 9) operation must be started after users check that clock gear-up procedure has been finished. For details of clock gear-up sequence, see Section 13, Clock Controller.**

14.6.2 Procedures to Reset by CLMATEST.RESCLM

CLMATEST.RESCLM can also reset each clock monitor. Before changing clock frequency via register during operation. It is necessary to go through the procedure as below and to retry setting parameters for CLMAn.

[Procedure]

Example case: Reset CLMA0 or Reset CLMA0 to CLMA9

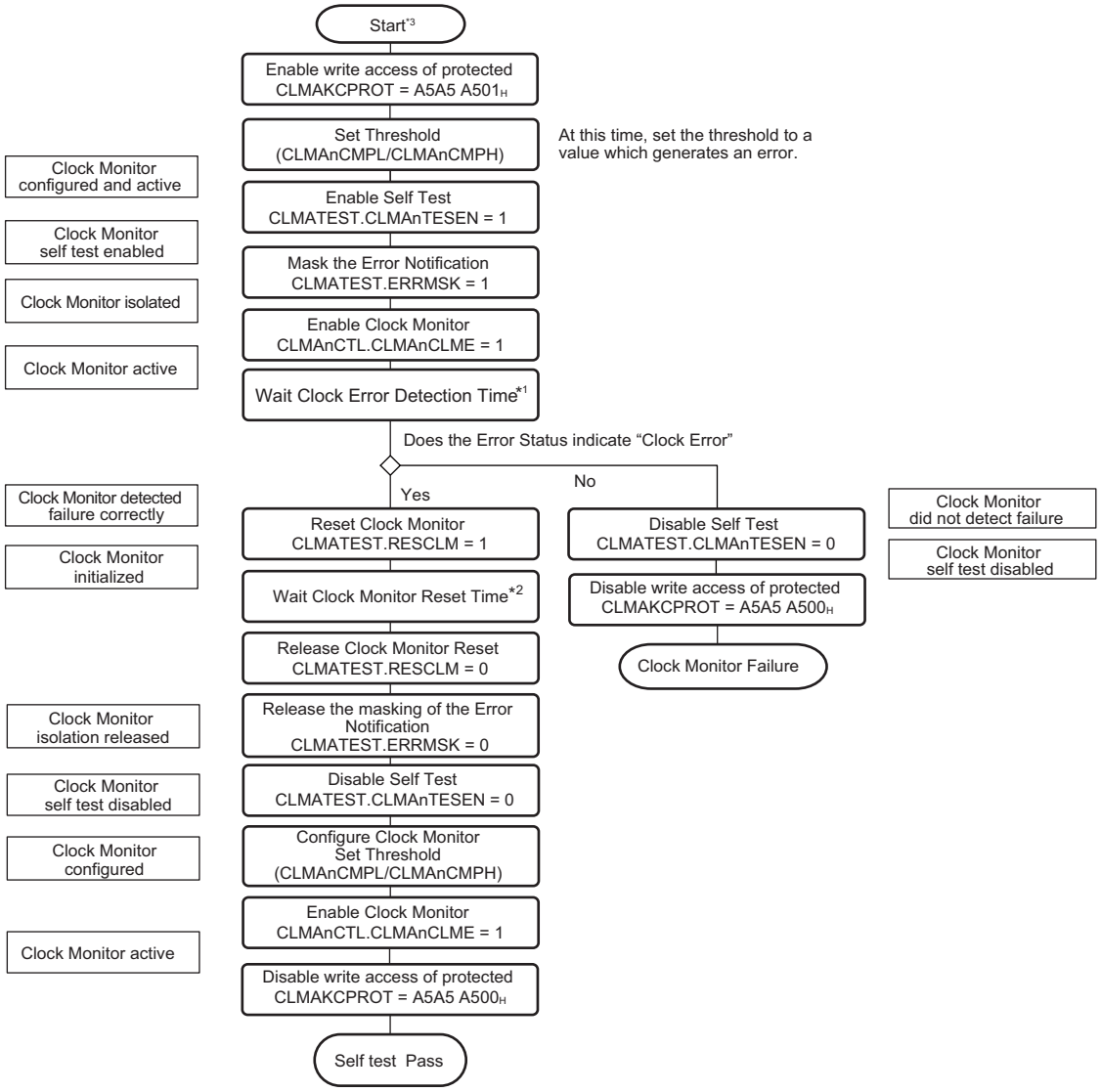


Note 1. CLMATEST.RESCLM must be kept asserted for at least 2 cycles at the corresponding sampling clock in order to make the CLMA initialized.

Figure 14.5 Flow Chart of the Reset by CLMATEST.RESCLM

14.6.3 Procedures to do Self-Test

Self-diagnosis of the clock monitor is available as described below. In the description, the clock monitor to be self-diagnosed is operating.



Note 1. After CLMAnCTL.CLMAnCLME bit is set to “1”, it requires more than or equal to 32 cycles at the sampling clock to detect the error status via CLMATESTS.

Note 2. CLMATEST.RESCLM must be kept asserted for at least 2 cycles at the corresponding sampling clock in order to make the CLMA initialized.

Note 3. Initialize CLMA3 and clear the error status of ECM error source status register after reset release of CLMA3. See **Section 14.7.1, Notes on CLMA3 after Reset release** for details.

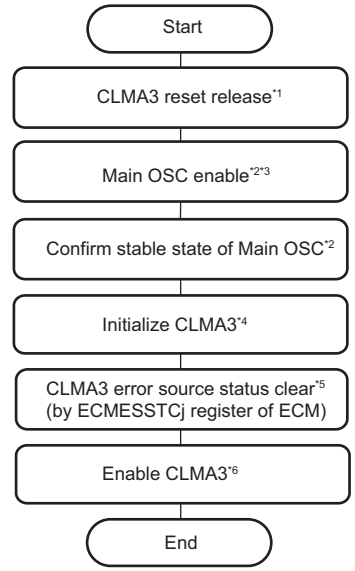
Figure 14.6 Flow Chart of the Self-Diagnosis

14.7 Usage Notes for CLMAn

14.7.1 Notes on CLMA3 after Reset release

After power-up or resume from power-down of MCU, the ECM error input of CLMA3 becomes undefined value and the ECM error source status register may be flagged.

So initialize CLMA3 and clear the error status of ECM error source status register after reset release of CLMA3. **Figure 14.7** shows the flow for clearing ECM error source status.

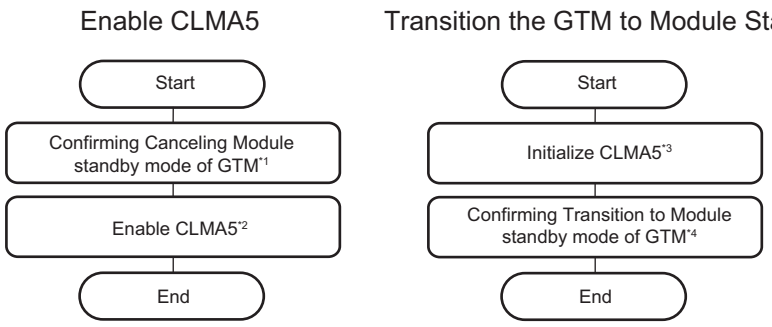


Note 1. See **Table 14.8** Register reset conditions for CLMA3 reset.
Note 2. See **Section 13.5.4.4, MOSCE — Main OSC Enable Register** for details.
Note 3. Setting is unnecessary if Main OSC is already enabled.
Note 4. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.
Note 5. See **Section 45.3.13, ECMESSTCj — ECM Error Source Status Clear Trigger Register j** for details.
Note 6. See **Section 14.6.1, Procedures to Enable CLMAn** and **Section 14.6.3, Procedures to do Self-Test** for details.

Figure 14.7 Flow Chart for clearing ECM error source status after CLMA3 reset release

14.7.2 Notes on CLMA5 in Module standby mode

Enable CLMA5 after confirming canceling module standby mode of GTM.
Stop the CLMA5 prior to transition the GTM to module standby.



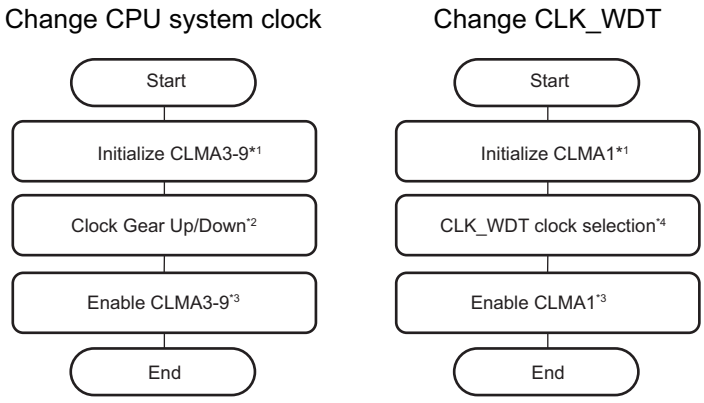
Note 1. See **Section 15.1.2.1, Canceling to Module Standby Mode** for details.
Note 2. See **Section 14.6.1, Procedures to Enable CLMAn** for details.
Note 3. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.
Note 4. See **Section 15.1.2.2, Transition to Module Standby Mode** for details.

Figure 14.8 Flow Chart of the CLMA5 in Module standby mode

14.7.3 Notes on Procedures to Clock Change

Initialize the clock monitor before the monitored clock or sampling clock is changed (including when the oscillator is stopped by software), and enable the clock monitor after the monitored clock and sampling clock are stable.

Clock change of the clock monitor is available as described below.



Note 1. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.
Note 2. See **Section 13.6.5, Sequence for Shifting the CPU System Clock Gear Up/Down** for details.
Note 3. See **Section 14.6.1, Procedures to Enable CLMAn** for details.
Note 4. See **Section 13.6.1.2, Setting Procedure for Clock** for details.

Figure 14.9 Flow Chart of the Clock Change

Section 15 Standby Controller (STBC)

This section describes the functions of the standby controller (STBC), the registers, and various chip standby modes.

15.1 Functions

15.1.1 Types of Chip Standby Mode

The RH850/U2A-EVA supports STOP mode and DeepSTOP mode for system-level low power status. In addition, the RH850/U2A-EVA supports cyclic operation (Cyclic RUN mode and Cyclic STOP mode) which supports low-power operation of limited functions. Transition between each mode is described in the **Section 15.1.1.4, Transition to Chip Standby Mode**.

- RUN mode
  RUN mode is a normal operation mode where the CPU is operating and all of other modules can operate. The CPU can enter “HALT” state by executing the “HALT” instruction to stop its operation in this mode.
- STOP mode
  STOP mode is a chip-level chip standby mode in which the clock supply to a certain clock can be stopped.
  STOP mode is entered by setting the STBC0STPT.STBC0STPTRG bit is set to 1.
  The clock supply to a certain clock can continue even in STOP mode by MSR_<name>*1.STPMSK_<name> bit setting. For details on the MSR_<name> register, see **Section 15.1.2.3, Behavior of Module Standby Mode during Chip Standby Mode**.
- DeepSTOP mode
  DeepSTOP mode is a chip-level chip standby mode to reduce power consumption further than STOP mode. In addition to the clock supply stop, the power supply to the Isolated area is switched off.
  DeepSTOP mode is entered by setting the STBC0PSC.STBC0DISTRG bit is set to 1.
- Cyclic RUN mode
  Cyclic RUN mode is a low-power operation mode in which limited modules can operate at low speed. In this mode, only the CPU0(PE0), the peripheral functions in the Always-On area, RLIN3, MSPI can operate. Data/Code flash memory and PLL are not available. The CPU1(PE1)/ CPU2(PE2)/CPU3(PE3) is also not available.
  The CPU0(PE0) executes the instructions in the retention RAM.
  In this mode, the CPU0(PE0) can issue the “HALT” instruction to enter HALT state.
  The mode transition to Cyclic RUN mode from DeepSTOP mode is triggered by Wake-Up Factor 1, and the mode transition to Cyclic RUN mode from Cyclic STOP mode is triggered by either Wake-Up Factor 0 or Wake-Up Factor 1.
  MSPI cannot be Wake-Up Factor because it does not have wake-up function.
- Cyclic STOP mode
  Cyclic STOP mode is a STOP mode in cyclic operation, and the CPU0(PE0) halts its operation.
  This mode is entered by setting the STBC0STPT.STBC0STPTRG bit to 1 in Cyclic RUN mode.

Note 1. In this section, <name> represents target module.

15.1.1.1 Wake-Up Control

(1) Wake-Up Factors for Chip Standby Modes

The standby controller can initiate return from chip standby mode by Wake-Up Factor 0 and Wake-Up Factor 1.

- Wake-Up Factor 0

  Wake-Up Factor 0 is trigger which can initiate return from STOP/DeepSTOP mode to RUN mode and can initiate return from Cyclic STOP mode to Cyclic RUN mode.

  When transition from Cyclic operation (Cyclic RUN mode and Cyclic STOP mode) to RUN mode, return to RUN mode via DeepSTOP mode.

  When the transition from Cyclic STOP to Cyclic RUN is made by Wake-Up Factor 0, if the transition to DeepSTOP by STBC0PSC.STBC0DISTRG is made without clearing Wake-Up Factor 0, the transition to RUN mode is made.

- Wake-Up Factor 1

  Wake-Up Factor 1 is trigger which can initiate return from DeepSTOP mode to Cyclic RUN mode and can initiate return from Cyclic STOP mode to Cyclic RUN mode.

(2) Setting of Wake-Up Factors

Wake-Up Factors for returning from chip standby modes are controlled by the following standby controller registers:

- Wake-Up Factor registers: WUF0_A0, WUF0_A1, WUF0_A2, WUF0_I0, WUF0_I1, WUF1_A0, WUF1_A1, WUF1_A2, WUF1_I0, WUF1_I1

  Upon occurrence of an effective Wake-Up Factor, the associated Wake-Up Factor flag is set to 1. By checking these registers and their flags, it is possible to identify the Wake-Up Factor.

- Wake-Up Factor mask registers: WUFMSK0_A0, WUFMSK0_A1, WUFMSK0_A2, WUFMSK0_I0, WUFMSK0_I1, WUFMSK1_A0, WUFMSK1_A1, WUFMSK1_A2, WUFMSK1_I0, WUFMSK1_I1

  Each bit of these registers is assigned to a certain Wake-Up Factor. Wake-up by this factor is enabled if its mask bit is set to 0. Wake-Up Factors assigned to both Wake-Up Factor 0 and 1 should not be enabled at the same time.

- Wake-Up Factor clear registers: WUFC0_A0, WUFC0_A1, WUFC0_A2, WUFC0_I0, WUFC0_I1, WUFC1_A0, WUFC1_A1, WUFC1_A2, WUFC1_I0, WUFC1_I1

  By setting the applicable bits in these registers to 1, the Wake-Up Factor bit (WUFy) in the Wake-Up Factor registers (WUF0_A0, WUF0_A1, WUF0_A2, WUF0_I0, WUF0_I1, WUF1_A0, WUF1_A1, WUF1_A2, WUF1_I0, WUF1_I1) can be cleared.

NOTE

The Wake-Up Factor flags in the Wake-Up Factor registers (WUF0_A0, WUF0_A1, WUF0_A2, WUF0_I0, WUF0_I1, WUF1_A0, WUF1_A1, WUF1_A2, WUF1_I0, WUF1_I1) only indicate the occurrence of Wake-Up Factor. These flags do not indicate a transition from chip standby mode to normal mode.

The assignment of the Wake-Up Factors to the control register bits and status register bits are shown in the following tables.

For details about the wake-up control and status registers, see below.

- **Section 15.2.4.35, WUF0_Ax/WUF1_Ax/WUF0_Ix/WUF1_Ix — Wake-Up Factor Registers**
- **Section 15.2.4.36, WUFMSK0_Ax/WUFMSK1_Ax/WUFMSK0_Ix/WUFMSK1_Ix — Wake-Up Factor Mask Registers**
- **Section 15.2.4.37, WUFC0_Ax/WUFC1_Ax/WUFC0_Ix/WUFC1_Ix — Wake-Up Factor Clear Registers**.

Table 15.1 Wake-Up Factor (WUF0/1_A0)

Wake-Up Factor	Module	Bit Assignment			Wake-Up Factor 0			Wake-Up Factor 1	
		Status Register	Mask Register	Clear Register	STOP to RUN	DeepSTOP to RUN	Cyclic STOP to Cyclic RUN	DeepSTOP to Cyclic RUN	Cyclic STOP to Cyclic RUN
TNMI	Port	WUF0/1_A0[0]	WUFMSK0/1_A0[0]	WUFC0/1_A0[0]	√	√	√	√	√
INTWDTBA	WDTBA	WUF0/1_A0[5]	WUFMSK0/1_A0[5]	WUFC0/1_A0[5]	√	√	√	√	√
INTP0	Port	WUF0/1_A0[6]	WUFMSK0/1_A0[6]	WUFC0/1_A0[6]	√	√	√	√	√
INTP1	Port	WUF0/1_A0[7]	WUFMSK0/1_A0[7]	WUFC0/1_A0[7]	√	√	√	√	√
INTP2	Port	WUF0/1_A0[8]	WUFMSK0/1_A0[8]	WUFC0/1_A0[8]	√	√	√	√	√
INTP3	Port	WUF0/1_A0[9]	WUFMSK0/1_A0[9]	WUFC0/1_A0[9]	√	√	√	√	√
INTP4	Port	WUF0/1_A0[10]	WUFMSK0/1_A0[10]	WUFC0/1_A0[10]	√	√	√	√	√
INTP5	Port	WUF0/1_A0[11]	WUFMSK0/1_A0[11]	WUFC0/1_A0[11]	√	√	√	√	√
INTP6	Port	WUF0/1_A0[12]	WUFMSK0/1_A0[12]	WUFC0/1_A0[12]	√	√	√	√	√
INTP7	Port	WUF0/1_A0[13]	WUFMSK0/1_A0[13]	WUFC0/1_A0[13]	√	√	√	√	√
INTP8	Port	WUF0/1_A0[14]	WUFMSK0/1_A0[14]	WUFC0/1_A0[14]	√	√	√	√	√
INTP9	Port	WUF0/1_A0[15]	WUFMSK0/1_A0[15]	WUFC0/1_A0[15]	√	√	√	√	√
INTP10	Port	WUF0/1_A0[16]	WUFMSK0/1_A0[16]	WUFC0/1_A0[16]	√	√	√	√	√
INTP11	Port	WUF0/1_A0[17]	WUFMSK0/1_A0[17]	WUFC0/1_A0[17]	√	√	√	√	√
INTP12	Port	WUF0/1_A0[18]	WUFMSK0/1_A0[18]	WUFC0/1_A0[18]	√	√	√	√	√
INTP13	Port	WUF0/1_A0[19]	WUFMSK0/1_A0[19]	WUFC0/1_A0[19]	√	√	√	√	√
INTP14	Port	WUF0/1_A0[20]	WUFMSK0/1_A0[20]	WUFC0/1_A0[20]	√	√	√	√	√
INTP15	Port	WUF0/1_A0[21]	WUFMSK0/1_A0[21]	WUFC0/1_A0[21]	√	√	√	√	√

Table 15.2 Wake-Up Factor (WUF0/1_A1)

Wake-Up Factor	Module	Bit Assignment			Wake-Up Factor 0			Wake-Up Factor 1	
		Status Register	Mask Register	Clear Register	STOP to RUN	DeepSTOP to RUN	Cyclic STOP to Cyclic RUN	DeepSTOP to Cyclic RUN	Cyclic STOP to Cyclic RUN
INTP16	Port	WUF0/1_A1[0]	WUFMSK0/1_A1[0]	WUFC0/1_A1[0]	√	√	√	√	√
INTP17	Port	WUF0/1_A1[1]	WUFMSK0/1_A1[1]	WUFC0/1_A1[1]	√	√	√	√	√
INTP18	Port	WUF0/1_A1[2]	WUFMSK0/1_A1[2]	WUFC0/1_A1[2]	√	√	√	√	√
INTP19	Port	WUF0/1_A1[3]	WUFMSK0/1_A1[3]	WUFC0/1_A1[3]	√	√	√	√	√
INTP20	Port	WUF0/1_A1[4]	WUFMSK0/1_A1[4]	WUFC0/1_A1[4]	√	√	√	√	√
INTP21	Port	WUF0/1_A1[5]	WUFMSK0/1_A1[5]	WUFC0/1_A1[5]	√	√	√	√	√
INTP22	Port	WUF0/1_A1[6]	WUFMSK0/1_A1[6]	WUFC0/1_A1[6]	√	√	√	√	√
INTP23	Port	WUF0/1_A1[7]	WUFMSK0/1_A1[7]	WUFC0/1_A1[7]	√	√	√	√	√
INTP24	Port	WUF0/1_A1[8]	WUFMSK0/1_A1[8]	WUFC0/1_A1[8]	√	√	√	√	√
INTP25	Port	WUF0/1_A1[9]	WUFMSK0/1_A1[9]	WUFC0/1_A1[9]	√	√	√	√	√
INTP26	Port	WUF0/1_A1[10]	WUFMSK0/1_A1[10]	WUFC0/1_A1[10]	√	√	√	√	√
INTP27	Port	WUF0/1_A1[11]	WUFMSK0/1_A1[11]	WUFC0/1_A1[11]	√	√	√	√	√
INTP28	Port	WUF0/1_A1[12]	WUFMSK0/1_A1[12]	WUFC0/1_A1[12]	√	√	√	√	√
INTP29	Port	WUF0/1_A1[13]	WUFMSK0/1_A1[13]	WUFC0/1_A1[13]	√	√	√	√	√
INTP30	Port	WUF0/1_A1[14]	WUFMSK0/1_A1[14]	WUFC0/1_A1[14]	√	√	√	√	√
INTP31	Port	WUF0/1_A1[15]	WUFMSK0/1_A1[15]	WUFC0/1_A1[15]	√	√	√	√	√
INTP32	Port	WUF0/1_A1[16]	WUFMSK0/1_A1[16]	WUFC0/1_A1[16]	√	√	√	√	√
INTP33	Port	WUF0/1_A1[17]	WUFMSK0/1_A1[17]	WUFC0/1_A1[17]	√	√	√	√	√
INTP34	Port	WUF0/1_A1[18]	WUFMSK0/1_A1[18]	WUFC0/1_A1[18]	√	√	√	√	√
INTP35	Port	WUF0/1_A1[19]	WUFMSK0/1_A1[19]	WUFC0/1_A1[19]	√	√	√	√	√
INTP36	Port	WUF0/1_A1[20]	WUFMSK0/1_A1[20]	WUFC0/1_A1[20]	√	√	√	√	√
INTP37	Port	WUF0/1_A1[21]	WUFMSK0/1_A1[21]	WUFC0/1_A1[21]	√	√	√	√	√
INTP38	Port	WUF0/1_A1[22]	WUFMSK0/1_A1[22]	WUFC0/1_A1[22]	√	√	√	√	√
INTP39	Port	WUF0/1_A1[23]	WUFMSK0/1_A1[23]	WUFC0/1_A1[23]	√	√	√	√	√

Table 15.3 Wake-Up Factor (WUF0/1_A2)

Wake-Up Factor	Module	Bit Assignment			Wake-Up Factor 0			Wake-Up Factor 1	
		Status Register	Mask Register	Clear Register	STOP to RUN	DeepSTOP to RUN	Cyclic STOP to Cyclic RUN	DeepSTOP to Cyclic RUN	Cyclic STOP to Cyclic RUN
WUTRG0	LPS0	WUF0/1_A2[0]	WUFMSK0/1_A2[0]	WUFC0/1_A2[0]	√	√	√	√	√
WUTRG1	LPS0	WUF0/1_A2[1]	WUFMSK0/1_A2[1]	WUFC0/1_A2[1]	√	√	√	√	√
INTDCUTDI	JTAG	WUF0/1_A2[2]	WUFMSK0/1_A2[2]	WUFC0/1_A2[2]	√	√	√	√	√
INTTAUJ2I0	TAUJ2	WUF0/1_A2[3]	WUFMSK0/1_A2[3]	WUFC0/1_A2[3]	√	√	√	√	√
INTTAUJ2I1	TAUJ2	WUF0/1_A2[4]	WUFMSK0/1_A2[4]	WUFC0/1_A2[4]	√	√	√	√	√
INTTAUJ2I2	TAUJ2	WUF0/1_A2[5]	WUFMSK0/1_A2[5]	WUFC0/1_A2[5]	√	√	√	√	√
INTTAUJ2I3	TAUJ2	WUF0/1_A2[6]	WUFMSK0/1_A2[6]	WUFC0/1_A2[6]	√	√	√	√	√
INTTAUJ3I0	TAUJ3	WUF0/1_A2[7]	WUFMSK0/1_A2[7]	WUFC0/1_A2[7]	√	√	√	√	√
INTTAUJ3I1	TAUJ3	WUF0/1_A2[8]	WUFMSK0/1_A2[8]	WUFC0/1_A2[8]	√	√	√	√	√
INTTAUJ3I2	TAUJ3	WUF0/1_A2[9]	WUFMSK0/1_A2[9]	WUFC0/1_A2[9]	√	√	√	√	√
INTTAUJ3I3	TAUJ3	WUF0/1_A2[10]	WUFMSK0/1_A2[10]	WUFC0/1_A2[10]	√	√	√	√	√
INTRTCA01S	RTCA0	WUF0/1_A2[11]	WUFMSK0/1_A2[11]	WUFC0/1_A2[11]	√	√	√	√	√
INTRTCA0AL	RTCA0	WUF0/1_A2[12]	WUFMSK0/1_A2[12]	WUFC0/1_A2[12]	√	√	√	√	√
INTRTCA0R	RTCA0	WUF0/1_A2[13]	WUFMSK0/1_A2[13]	WUFC0/1_A2[13]	√	√	√	√	√
INTADCJ2I0	ADCJ2	WUF0/1_A2[14]	WUFMSK0/1_A2[14]	WUFC0/1_A2[14]	√*1	√*1	√*1	√*1	√*1
INTADCJ2I1	ADCJ2	WUF0/1_A2[15]	WUFMSK0/1_A2[15]	WUFC0/1_A2[15]	√*1	√*1	√*1	√*1	√*1
INTADCJ2I2	ADCJ2	WUF0/1_A2[16]	WUFMSK0/1_A2[16]	WUFC0/1_A2[16]	√*1	√*1	√*1	√*1	√*1
INTADCJ2I3	ADCJ2	WUF0/1_A2[17]	WUFMSK0/1_A2[17]	WUFC0/1_A2[17]	√*1	√*1	√*1	√*1	√*1
INTADCJ2I4	ADCJ2	WUF0/1_A2[18]	WUFMSK0/1_A2[18]	WUFC0/1_A2[18]	√*1	√*1	√*1	√*1	√*1

Note 1. These Wake-Up Factors are only available in LPS analog input mode.

Table 15.4 Wake-Up Factor (WUF0/1_I0)

Wake-Up Factor	Module	Bit Assignment			Wake-Up Factor 0			Wake-Up Factor 1	
		Status Register	Mask Register	Clear Register	STOP to RUN	DeepSTOP to RUN	Cyclic STOP to Cyclic RUN	DeepSTOP to Cyclic RUN	Cyclic STOP to Cyclic RUN
INTRCANGRECC0	RSCFD0	WUF0_I0[0]	WUFMSK0_I0[0]	WUFC0_I0[0]	√	—	—	—	—
INTRCAN0REC	RSCFD0	WUF0_I0[1]	WUFMSK0_I0[1]	WUFC0_I0[1]	√	—	—	—	—
INTRCAN1REC	RSCFD0	WUF0_I0[2]	WUFMSK0_I0[2]	WUFC0_I0[2]	√	—	—	—	—
INTRCAN2REC	RSCFD0	WUF0_I0[3]	WUFMSK0_I0[3]	WUFC0_I0[3]	√	—	—	—	—
INTRCAN3REC	RSCFD0	WUF0_I0[4]	WUFMSK0_I0[4]	WUFC0_I0[4]	√	—	—	—	—
INTRCAN4REC	RSCFD0	WUF0_I0[5]	WUFMSK0_I0[5]	WUFC0_I0[5]	√	—	—	—	—
INTRCAN5REC	RSCFD0	WUF0_I0[6]	WUFMSK0_I0[6]	WUFC0_I0[6]	√	—	—	—	—
INTRCAN6REC	RSCFD0	WUF0_I0[7]	WUFMSK0_I0[7]	WUFC0_I0[7]	√	—	—	—	—
INTRCAN7REC	RSCFD0	WUF0_I0[8]	WUFMSK0_I0[8]	WUFC0_I0[8]	√	—	—	—	—
INTRCANGRECC1	RSCFD1	WUF0_I0[9]	WUFMSK0_I0[9]	WUFC0_I0[9]	√	—	—	—	—
INTRCAN8REC	RSCFD1	WUF0_I0[10]	WUFMSK0_I0[10]	WUFC0_I0[10]	√	—	—	—	—
INTRCAN9REC	RSCFD1	WUF0_I0[11]	WUFMSK0_I0[11]	WUFC0_I0[11]	√	—	—	—	—
INTRCAN10REC	RSCFD1	WUF0_I0[12]	WUFMSK0_I0[12]	WUFC0_I0[12]	√	—	—	—	—
INTRCAN11REC	RSCFD1	WUF0_I0[13]	WUFMSK0_I0[13]	WUFC0_I0[13]	√	—	—	—	—
INTRCAN12REC	RSCFD1	WUF0_I0[14]	WUFMSK0_I0[14]	WUFC0_I0[14]	√	—	—	—	—
INTRCAN13REC	RSCFD1	WUF0_I0[15]	WUFMSK0_I0[15]	WUFC0_I0[15]	√	—	—	—	—
INTRCAN14REC	RSCFD1	WUF0_I0[16]	WUFMSK0_I0[16]	WUFC0_I0[16]	√	—	—	—	—
INTRCAN15REC	RSCFD1	WUF0_I0[17]	WUFMSK0_I0[17]	WUFC0_I0[17]	√	—	—	—	—

Table 15.5 Wake-Up Factor (WUF0/1_I1)

Wake-Up Factor	Module	Bit Assignment			Wake-Up Factor 0			Wake-Up Factor 1	
		Status Register	Mask Register	Clear Register	STOP to RUN	DeepSTOP to RUN	Cyclic STOP to Cyclic RUN	DeepSTOP to Cyclic RUN	Cyclic STOP to Cyclic RUN
INTRLIN30	RLIN30	WUF0/1_I1[0]	WUFMSK0/1_I1[0]	WUFC0/1_I1[0]	√	—	√	—	√
INTRLIN31	RLIN31	WUF0/1_I1[1]	WUFMSK0/1_I1[1]	WUFC0/1_I1[1]	√	—	√	—	√
INTRLIN32	RLIN32	WUF0/1_I1[2]	WUFMSK0/1_I1[2]	WUFC0/1_I1[2]	√	—	√	—	√
INTRLIN33	RLIN33	WUF0/1_I1[3]	WUFMSK0/1_I1[3]	WUFC0/1_I1[3]	√	—	√	—	√
INTRLIN34	RLIN34	WUF0/1_I1[4]	WUFMSK0/1_I1[4]	WUFC0/1_I1[4]	√	—	√	—	√
INTRLIN35	RLIN35	WUF0/1_I1[5]	WUFMSK0/1_I1[5]	WUFC0/1_I1[5]	√	—	√	—	√
INTRLIN36	RLIN36	WUF0/1_I1[6]	WUFMSK0/1_I1[6]	WUFC0/1_I1[6]	√	—	√	—	√
INTRLIN37	RLIN37	WUF0/1_I1[7]	WUFMSK0/1_I1[7]	WUFC0/1_I1[7]	√	—	√	—	√
INTRLIN38	RLIN38	WUF0/1_I1[8]	WUFMSK0/1_I1[8]	WUFC0/1_I1[8]	√	—	√	—	√
INTRLIN39	RLIN39	WUF0/1_I1[9]	WUFMSK0/1_I1[9]	WUFC0/1_I1[9]	√	—	√	—	√
INTRLIN310	RLIN310	WUF0/1_I1[10]	WUFMSK0/1_I1[10]	WUFC0/1_I1[10]	√	—	√	—	√
INTRLIN311	RLIN311	WUF0/1_I1[11]	WUFMSK0/1_I1[11]	WUFC0/1_I1[11]	√	—	√	—	√
INTRLIN312	RLIN312	WUF0/1_I1[12]	WUFMSK0/1_I1[12]	WUFC0/1_I1[12]	√	—	√	—	√
INTRLIN313	RLIN313	WUF0/1_I1[13]	WUFMSK0/1_I1[13]	WUFC0/1_I1[13]	√	—	√	—	√
INTRLIN314	RLIN314	WUF0/1_I1[14]	WUFMSK0/1_I1[14]	WUFC0/1_I1[14]	√	—	√	—	√
INTRLIN315	RLIN315	WUF0/1_I1[15]	WUFMSK0/1_I1[15]	WUFC0/1_I1[15]	√	—	√	—	√
INTRLIN316	RLIN316	WUF0/1_I1[16]	WUFMSK0/1_I1[16]	WUFC0/1_I1[16]	√	—	√	—	√
INTRLIN317	RLIN317	WUF0/1_I1[17]	WUFMSK0/1_I1[17]	WUFC0/1_I1[17]	√	—	√	—	√
INTRLIN318	RLIN318	WUF0/1_I1[18]	WUFMSK0/1_I1[18]	WUFC0/1_I1[18]	√	—	√	—	√
INTRLIN319	RLIN319	WUF0/1_I1[19]	WUFMSK0/1_I1[19]	WUFC0/1_I1[19]	√	—	√	—	√
INTRLIN320	RLIN320	WUF0/1_I1[20]	WUFMSK0/1_I1[20]	WUFC0/1_I1[20]	√	—	√	—	√
INTRLIN321	RLIN321	WUF0/1_I1[21]	WUFMSK0/1_I1[21]	WUFC0/1_I1[21]	√	—	√	—	√
INTRLIN322	RLIN322	WUF0/1_I1[22]	WUFMSK0/1_I1[22]	WUFC0/1_I1[22]	√	—	√	—	√
INTRLIN323	RLIN323	WUF0/1_I1[23]	WUFMSK0/1_I1[23]	WUFC0/1_I1[23]	√	—	√	—	√

CAUTION

For the pins of the function used for the Wake-Up Factors from DeepSTOP, use the multiplexed functions of the ports assigned to the Always-On area or assigned to "Wake-up factors". See Section 2.2.1, Pin List and Function assignment.

15.1.1.2 On-Chip Debug Wake-Up

The On-Chip Debug Module (OCD) generates a wake-up event while the microcontroller runs the application program in the following cases:

- The debugger issues a stop request
- A breakpoint is hit

In either case all chip standby modes are terminated, provided the OCD debug event is enabled as a Wake-Up Factor via the WUFMSK0/1_A2 register.

CAUTION

If the OCD wake-up event is disabled, it is not possible to wake-up from chip standby modes via an On-chip debugger request.
The OCD wake-up event can be enabled as a Wake-Up Factor for all chip standby modes by setting WUFMSK0/1_A2[2] = 0.
When the hot plug-in function is used, make sure to enable the OCD wake-up event and return from chip standby mode by INTDCUTDI interrupt.

15.1.1.3 I/O Buffer Control

In DeepSTOP mode, the I/O buffers of port groups in the Isolated area transition to I/O buffer hold state.
For port groups in the Isolated area supporting I/O buffer hold state, see **Section 2.2.2, Pin Status**.

(1) I/O Buffer Hold State

During the I/O buffer hold state, the I/O buffer maintains the state before it enters this state. Therefore, no external or internal signal can change the state of the I/O buffer until the I/O buffer hold state is terminated.

After wake-up from DeepSTOP, the I/O buffers remain in I/O buffer hold state.
And releases I/O buffer hold state for each port group by setting the IOHOLDn register.
To release I/O buffer hold state, follow the steps shown below.

1. Re-configure the peripheral or port function.
2. Set IOHOLDn.IOHOLD_xxx = 0. (n = 0, 2)

CAUTION

The data from the I/O buffer can be used from when the common I/O buffer input is enabled in the above step 1. Follow the steps shown above even if the LVDS buffer is used.

The following table is a summary of the I/O buffer in the Isolated area during DeepSTOP mode and after wake-up.

Table 15.6 Buffer Operation during DeepSTOP Mode and after Wake-Up (I/O buffers in the Isolated area)

Chip Standby mode	Before DeepSTOP mode	During DeepSTOP mode	After Wake-Up*1
DeepSTOP mode	Normal operation	I/O buffer hold state*2	I/O buffer hold state*2

Note 1. Set the IOHOLDn.IOHOLD_xxx (n = 0, 2) bit to "0" to release the I/O buffer hold state.

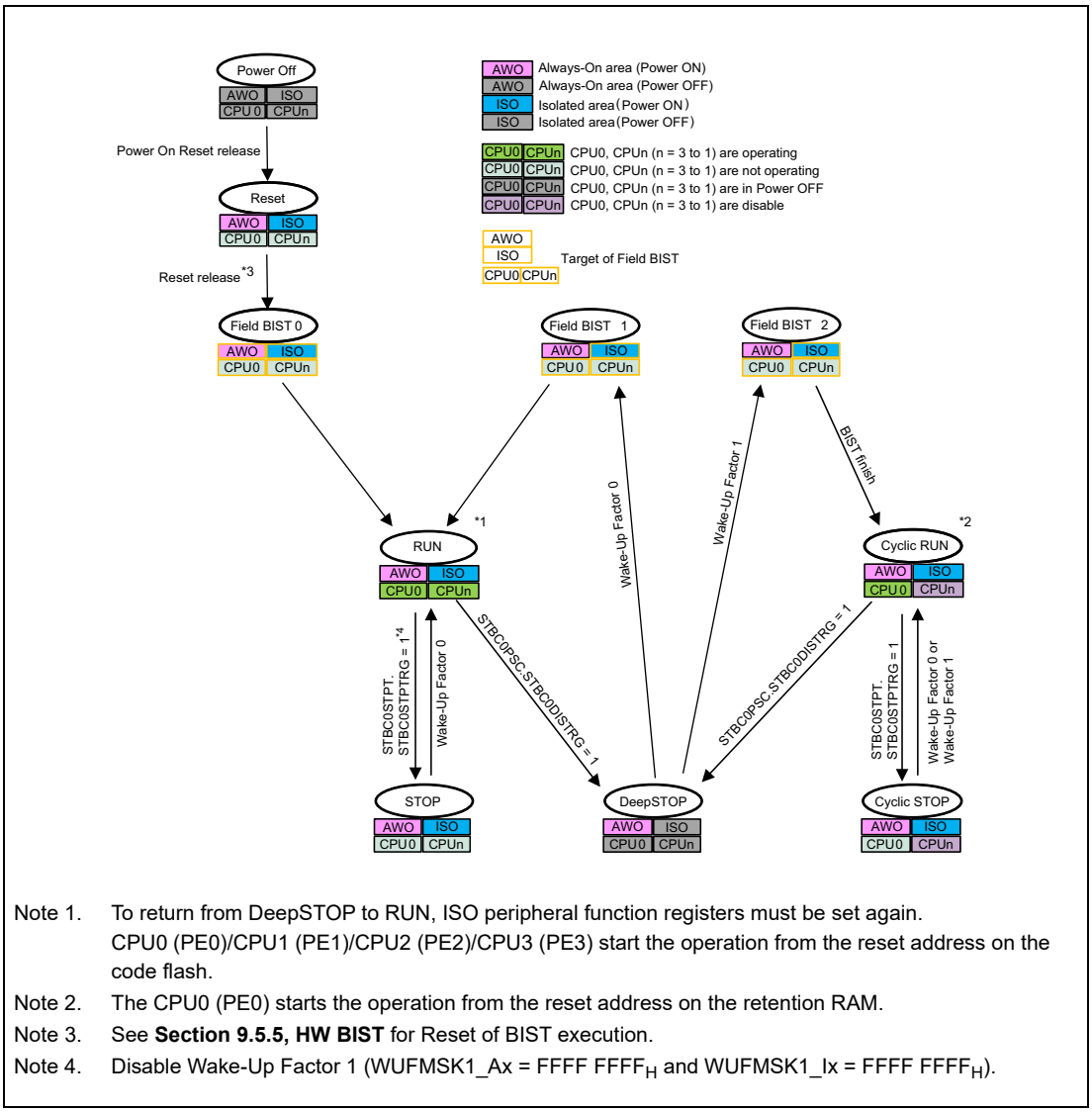
Note 2. I/O buffer hold sate is not supported when LVDS is used. The state is Hi-Z.

The port groups in the Always-On area don't support I/O buffer hold state. They continue operation and remain its state before entering DeepSTOP. In the case an alternative function of modules in Isolated area is assigned to the pin in the Always-On area, the state of the I/O buffer may change in the transition to the DeepSTOP due to initialization of the modules in the Isolated area by DeepSTOP Reset To avoid this behavior, it is recommended to change to function of modules in Always-On area (e. g. Port mode) before entering DeepSTOP.

15.1.1.4 Transition to Chip Standby Mode

The figure below shows transition between RUN mode and chip standby mode.

When Wake-Up Factor 0 and Wake-Up Factor 1 are set at the same time, Wake-Up Factor 0 takes precedence.



Note 1. To return from DeepSTOP to RUN, ISO peripheral function registers must be set again. CPU0 (PE0)/CPU1 (PE1)/CPU2 (PE2)/CPU3 (PE3) start the operation from the reset address on the code flash.

Note 2. The CPU0 (PE0) starts the operation from the reset address on the retention RAM.

Note 3. See **Section 9.5.5, HW BIST** for Reset of BIST execution.

Note 4. Disable Wake-Up Factor 1 (WUFMSK1_Ax = FFFF FFFF$_H$ and WUFMSK1_Ix = FFFF FFFF$_H$).

Figure 15.1 Transition to Chip Standby Mode

Table 15.7 CPU operating in each Chip Standby Mode

	CPU0(PE0)	CPU1(PE1)	CPU2(PE2)	CPU3(PE3)
RUN mode	Run	Run*1	Run*1	Run*1
STOP mode	Stop	Stop	Stop	Stop
DeepSTOP mode	Power off	Power off	Power off	Power off
Cyclic RUN mode	Run*2	Disable	Disable	Disable
Cyclic STOP mode	Stop	Disable	Disable	Disable

Note 1. After writing "1" to the bit of the BOOTCTRL register corresponding to each CPU, CPU start operating.

Note 2. The beginning address of Reset vector is Retention RAM (FE80 0000$_H$).

15.1.2 Types of Module Standby Mode

This function stops the clocks for peripheral modules by register setting in order to reduces power consumption.
Table 15.8, Modules that participate in Module Standby mode shows the modules that participate in the module standby modes.

After reset is released, all peripherals enter module standby modes.
Register access to the module in module standby mode is prohibited.

Example of the procedure of module standby mode is shown below.

15.1.2.1 Canceling to Module Standby Mode

1. Check the Software Module Reset Status Register of modules that participate in the module standby mode is 0 (SWMRESS_<name>. SWMRESS_<name>_n = 0).
2. Enable writing to protected registers (MSRKCPROT = $A5A5A501_H$).
3. Start all target clocks that cancels module standby mode (MSR_<name>.MS_<name>_n = 0).
4. Read the value of MSR_<name>.MS_<name>_n bit and check the value is 0.
5. Disable writing to protected registers (MSRKCPROT = $A5A5A500_H$).
6. Read the value of MSRKCPROT and check the value is 00000000_H.

CAUTION

ADCJ2 and TAUJ2/TAUJ3 need to wait for time before register setting after module standby cancel.
Read the register of the target module once for wait time.

Follow the steps shown below.

1. **Canceling module standby mode (Execute flow (2)(3)(4)(5)(6) above).**
2. **Read the register of ADCJ2 or TAUJ2/TAUJ3 once (Dummy read).**
3. **Register setting of ADCJ2 or TAUJ2/TAUJ3.**

15.1.2.2 Transition to Module Standby Mode

1. Check the modules that participate in the module standby mode have completed the operation and are in idle state and no other module or external pin may activate the module.
   For details of the way to confirm the idle state of the modules, see the section related to each module.
2. Check the Software Module Reset Status Register of modules that participate in the module standby mode is 0 (SWMRESS_<name>. SWMRESS_<name>_n = 0).
3. Stop all target clocks that participate in the module standby mode (MSR_<name>.MS_<name>_n = 1).

Table 15.8 Modules that participate in Module Standby mode (1/3)

Module Standby Register	Bit Name	Module Name
MSR_RSCFD	MS_RSCFD_1	RSCFD1, E7RC11, E7RC12, E7RC13
	MS_RSCFD_0	RSCFD0, E7RC01, E7RC02, E7RC03
MSR_FLXA	MS_FLXA_1	FLXA1, E7FR10, E7FR11, E7FR12
	MS_FLXA_0	FLXA0, E7FR00, E7FR01, E7FR02
MSR_GTM	MS_GTM_0	GTM0, GTM0_1
		E7GT00, E7GT01, E7GT10, E7GT11
		E7GT20, E7GT21, E7GT30, E7GT31
MSR_ETNB	MS_ETNB_1	ETNB1, E7GE00, E7GE01
	MS_ETNB_0	ETNB0, E7ME00, E7ME01
MSR_RSENT*1	MS_RSENT_7	RSENT7
	MS_RSENT_6	RSENT6
	MS_RSENT_5	RSENT5
	MS_RSENT_4	RSENT4
	MS_RSENT_3	RSENT3
	MS_RSENT_2	RSENT2
	MS_RSENT_1	RSENT1
	MS_RSENT_0	RSENT0
MSR_MSPI*2	MS_MSPI_9	MSPI9, MSPI9_INTF, E7MS09
	MS_MSPI_8	MSPI8, MSPI8_INTF, E7MS08
	MS_MSPI_7	MSPI7, MSPI7_INTF, E7MS07
	MS_MSPI_6	MSPI6, MSPI6_INTF, E7MS06
	MS_MSPI_5	MSPI5, MSPI5_INTF, E7MS05
	MS_MSPI_4	MSPI4, MSPI4_INTF, E7MS04
	MS_MSPI_3	MSPI3, MSPI3_INTF, E7MS03
	MS_MSPI_2	MSPI2, MSPI2_INTF, E7MS02
	MS_MSPI_1	MSPI1, MSPI1_INTF, E7MS01
	MS_MSPI_0	MSPI0, MSPI0_INTF, E7MS00
MSR_RLIN3	MS_RLIN3_23	RLIN323
	MS_RLIN3_22	RLIN322
	MS_RLIN3_21	RLIN321
	MS_RLIN3_20	RLIN320
	MS_RLIN3_19	RLIN319
	MS_RLIN3_18	RLIN318
	MS_RLIN3_17	RLIN317
	MS_RLIN3_16	RLIN316
	MS_RLIN3_15	RLIN315
	MS_RLIN3_14	RLIN314
	MS_RLIN3_13	RLIN313
	MS_RLIN3_12	RLIN312
	MS_RLIN3_11	RLIN311
	MS_RLIN3_10	RLIN310
	MS_RLIN3_9	RLIN39
	MS_RLIN3_8	RLIN38

Table 15.8 Modules that participate in Module Standby mode (2/3)

Module Standby Register	Bit Name	Module Name
MSR_RLIN3	MS_RLIN3_7	RLIN37
	MS_RLIN3_6	RLIN36
	MS_RLIN3_5	RLIN35
	MS_RLIN3_4	RLIN34
	MS_RLIN3_3	RLIN33
	MS_RLIN3_2	RLIN32
	MS_RLIN3_1	RLIN31
	MS_RLIN3_0	RLIN30
MSR_ADCJ_ISO	MS_ADCJ_1	ADCJ1
	MS_ADCJ_0	ADCJ0, AVSEG
MSR_CXPI	MS_CXPI_3	CXP13
	MS_CXPI_2	CXP12
	MS_CXPI_1	CXP11
	MS_CXPI_0	CXP10
MSR_MMCA	MS_MMCA_0	MMCA0, E7MM00, E7MM01
MSR_ENCA	MS_ENCA_1	ENCA1
	MS_ENCA_0	ENCA0
MSR_PSI5*3	MS_PSI5_3	PSI53
	MS_PSI5_2	PSI52
	MS_PSI5_1	PSI51
	MS_PSI5_0	PSI50
MSR_PSI5S	MS_PSI5S_1	PSI5S1
	MS_PSI5S_0	PSI5S0
MSR_PWMD	MS_PWMD	PWM-Diag (PWBA0, PWGC0-95, PWSD0, SLPW, PWGCINTF)
MSR_RHSIF	MS_RHSIF_0	RHSIF0
MSR_RIIC	MS_RIIC_1	RIIC1
	MS_RIIC_0	RIIC0
MSR_SCI3	MS_SCI3_2	SCI32
	MS_SCI3_1	SCI31
	MS_SCI3_0	SCI30
MSR_TAPA	MS_TAPA_3	TAPA3
	MS_TAPA_2	TAPA2
	MS_TAPA_1	TAPA1
	MS_TAPA_0	TAPA0
MSR_TAUD	MS_TAUD_2	TAUD2, SELB_TAUD2I
	MS_TAUD_1	TAUD1
	MS_TAUD_0	TAUD0
MSR_TAUJ_ISO	MS_TAUJ_1	TAUJ1
	MS_TAUJ_0	TAUJ0
MSR_TPBA	MS_TPBA_1	TPBA1
	MS_TPBA_0	TPBA0
MSR_TSG3	MS_TSG3_1	TSG31
	MS_TSG3_0	TSG30

Table 15.8 Modules that participate in Module Standby mode (3/3)

Module Standby Register	Bit Name	Module Name
MSR_OSTM	MS_OSTM_9	OSTM9, IC0CKSEL9
	MS_OSTM_8	OSTM8, IC0CKSEL8
	MS_OSTM_7	OSTM7
	MS_OSTM_6	OSTM6
	MS_OSTM_5	OSTM5
	MS_OSTM_4	OSTM4
	MS_OSTM_3	OSTM3
	MS_OSTM_2	OSTM2
	MS_OSTM_1	OSTM1
	MS_OSTM_0	OSTM0
MSR_ADCJ_AWO	MS_ADCJ_2	ADCJ2, ADCJ2SGTSELx
MSR_RTCA	MS_RTCA_0	RTCA0
MSR_TAUJ_AWO	MS_TAUJ_3	TAUJ3, SELB_TAUJ3I
	MS_TAUJ_2	TAUJ2, SELB_TAUJ2I
MSR_WDTB_AWO	—*4	WDTBA

Note 1. Register RSENTTSSEL is always accessible regardless of the value of MSR_RSENT.

Note 2. Register MSPITGCTLn, MSPITGDMAALT, and MSPITGDTSALT are always accessible regardless of the value of MSR_MSPI.

Note 3. Register PSI5TSSEL is always accessible regardless of the value of MSR_PSI5.

Note 4. Module Standby Mode is not available. Only Chip Standby Mode is available.

15.1.2.3 Behavior of Module Standby Mode during Chip Standby Mode

Depending on the state of each bit (STPMSK_<name>,MS_<name>_n) of Module Standby Register.

Clock supply state in each mode of chip standby mode is shown in **Table 15.9**.

Table 15.9 Clock supply state in each mode at Chip Standby Mode

Module Standby Register		Chip Standby Mode	
MS_<name>_n	STPMSK_<name>*1	RUN, Cyclic RUN	STOP, DeepSTOP, Cyclic STOP
1	0	Stop	Stop
1	1	Stop	Stop
0	0	Supply	Stop
0	1	Supply	Supply

Note 1. If the source clock of the peripheral clock stops during chip standby mode, MSR_<name>.STPMSK_<name> should be set to 0.

For description of MS_<name>_n and STPMSK_<name>, refer to **Section 15.2.4, Details of Standby Controller Control Registers**.

Table 15.10, Clock operation propriety table of Module standby register target Module shows whether operation propriety to clock in each state.

Table 15.10 Clock operation propriety table of Module standby register target Module (1/2)

Power Domain	module	Module standby register		RUN	Cyclic RUN	STOP	Deep STOP	Cyclic STOP
		STPMSK_<name>	MS_<name>_n					
ISO	RS-CANFD	√	√	Operable	—	Operable	—	—
ISO	FLXA	—	√	Operable	—	—	—	—
ISO	GTM	—	√	Operable	—	—	—	—
ISO	ETNB	—	√	Operable	—	—	—	—
ISO	RSENT	—	√	Operable	—	—	—	—
ISO	MSPI	—	√	Operable	Operable	—	—	—
ISO	RLIN3	√	√	Operable	Operable	Operable	—	Operable
ISO	ADCJ0/1	—	√	Operable	—	—	—	—
ISO	CXPI	—	√	Operable	—	—	—	—
ISO	MMCA	—	√	Operable	—	—	—	—
ISO	ENCA	—	√	Operable	—	—	—	—
ISO	PSI5	—	√	Operable	—	—	—	—
ISO	PSI5-S	—	√	Operable	—	—	—	—
ISO	PWM-Diag	—	√	Operable	—	—	—	—
ISO	RHSIF	—	√	Operable	—	—	—	—
ISO	RIIC	—	√	Operable	—	—	—	—
ISO	SCI3	—	√	Operable	—	—	—	—
ISO	SFMA	—	—	Operable	—	—	—	—
ISO	TAPA	—	√	Operable	—	—	—	—
ISO	TAUD	—	√	Operable	—	—	—	—
ISO	TAUJ0/1	—	√	Operable	—	—	—	—
ISO	TPBA	—	√	Operable	—	—	—	—
ISO	TSG3	—	√	Operable	—	—	—	—

Table 15.10 Clock operation propriety table of Module standby register target Module (2/2)

Power Domain	module	Module standby register		RUN	Cyclic RUN	STOP	Deep STOP	Cyclic STOP
		STPMSK_<name>	MS_<name>_n					
ISO	WDTB0	—	—	Operable	Operable	—	—	—
ISO	WDTB1/2/3	—	—	Operable	—	—	—	—
ISO	OSTM	—	√	Operable	—	—	—	—
AWO	WDTBA	√	—	Operable	Operable	Operable	Operable	Operable
AWO	TAUJ2/3	√	√	Operable	Operable	Operable	Operable	Operable
AWO	RTCA	√	√	Operable	Operable	Operable	Operable	Operable
AWO	ADCJ2	√	√	Operable	Operable	Operable	Operable	Operable
ISO	PIC	—	—	Operable	—	—	—	—
ISO	ECM	—	—	Operable	Operable	Operable*1	—	Operable*1
ISO	KCRC	—	—	Operable	—	—	—	—
ISO	OTS	—	—	Operable	—	—	—	—
AWO	LPS	—	—	Operable	Operable	Operable	Operable	Operable
AWO	FOUT	—	—	Operable	Operable	Operable	Operable	Operable

Note 1. ECM delay timer clock (CLK_ECMCNT) stops.

15.1.3 Clock Supply

The clock supply to the standby controller is shown in the following table.

Table 15.11 Clock Supply

Module Name	Module Clock Name	Supply Clock Name
Standby controller	Register access clock	CLK_LSB

15.2 Registers

15.2.1 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

By releasing the protection of STBCKCPROT, MSRKCPROT, it can be written.

15.2.2 List of Registers

The following table lists the standby controller registers.

Table 15.12 List of Registers (1/2)

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 1000$_H$	MSR_RSCFD	Module Standby Register for RS-CANFD	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1010$_H$	MSR_FLXA	Module Standby Register for FLXA	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1020$_H$	MSR_GTM	Module Standby Register for GTM	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 1030$_H$	MSR_ETNB	Module Standby Register for ETNB	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1040$_H$	MSR_RSENT	Module Standby Register for RSENT	32	0000 00FF$_H$	PBG20 #2	MSRKCPROT
FF98 1050$_H$	MSR_MSPI	Module Standby Register for MSPI	32	0000 03FF$_H$	PBG20 #2	MSRKCPROT
FF98 1060$_H$	MSR_RLIN3	Module Standby Register for RLIN3	32	00FF FFFF$_H$	PBG20 #2	MSRKCPROT
FF98 1070$_H$	MSR_ADCJ_ISO	Module Standby Register for ADCJ0/ ADCJ1 (ISO area)	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1080$_H$	MSR_CXPI	Module Standby Register for CXPI	32	0000 000F$_H$	PBG20 #2	MSRKCPROT
FF98 1090$_H$	MSR_MMCA	Module Standby Register for MMCA	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 10A0$_H$	MSR_ENCA	Module Standby Register for ENCA	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 10B0$_H$	MSR_PSI5	Module Standby Register for PSI5	32	0000 000F$_H$	PBG20 #2	MSRKCPROT
FF98 10C0$_H$	MSR_PSI5S	Module Standby Register for PSI5-S	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 10D0$_H$	MSR_PWMD	Module Standby Register for PWM-Diag	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 10E0$_H$	MSR_RHSIF	Module Standby Register for RHSIF	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 10F0$_H$	MSR_RIIC	Module Standby Register for RIIC	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1100$_H$	MSR_SCI3	Module Standby Register for SCI3	32	0000 0007$_H$	PBG20 #2	MSRKCPROT
FF98 1120$_H$	MSR_TAPA	Module Standby Register for TAPA	32	0000 000F$_H$	PBG20 #2	MSRKCPROT
FF98 1130$_H$	MSR_TAUD	Module Standby Register for TAUD	32	0000 0007$_H$	PBG20 #2	MSRKCPROT
FF98 1140$_H$	MSR_TAUJ_ISO	Module Standby Register for TAUJ0/ TAUJ1 (ISO area)	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1150$_H$	MSR_TPBA	Module Standby Register for TPBA	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1160$_H$	MSR_TSG3	Module Standby Register for TSG3	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 1180$_H$	MSR_OSTM	Module Standby Register for OSTM	32	0000 03FF$_H$	PBG20 #2	MSRKCPROT
FF98 1700$_H$	STBCKCPROT	Standby controller Register Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	
FF98 1710$_H$	MSRKCPROT	Module Standby Register Key Code Protection Register	32	0000 0000$_H$	PBG20 #2	
FF98 8C00$_H$	STBC0PSC	Power Save Control Register	32	0000 0000$_H$	PBG20 #2	STBCKCPROT
FF98 8C04$_H$	STBC0STPT	Power Stop Trigger Register	32	0000 0000$_H$	PBG20 #2	STBCKCPROT

Table 15.12 List of Registers (2/2)

Address	Register Name	Description	Access Width	Value after reset	Access Protection	
					PBG	Other
FF98 8C08$_H$	PWRGD_CNT	Power good counter register	32	0000 0FFF$_H$	PBG20 #2	STBCKCPROT
FF98 8D00$_H$	IOHOLD0	I/O Buffer Hold Control Register 0	32	0000 0000$_H$	PBG20 #2	STBCKCPROT
FF98 8D08$_H$	IOHOLD2	I/O Buffer Hold Control Register 2	32	0000 0000$_H$	PBG20 #2	STBCKCPROT
FF98 8E00$_H$	MSR_ADCJ_AWO	Module Standby Register for ADCJ2 (AWO area)	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 8E10$_H$	MSR_RTCA	Module Standby Register for RTCA	32	0000 0001$_H$	PBG20 #2	MSRKCPROT
FF98 8E20$_H$	MSR_TAUJ_AWO	Module Standby Register for TAUJ2/ TAUJ3 (AWO area)	32	0000 0003$_H$	PBG20 #2	MSRKCPROT
FF98 8E30$_H$	MSR_WDTB_AWO	Module Standby Register for WDTB (AWO area)	32	0000 0000$_H$	PBG20 #2	MSRKCPROT
FF98 E000$_H$	WUF0_A0	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E010$_H$	WUF0_A1	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E020$_H$	WUF0_A2	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E120$_H$	WUF1_A0	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E130$_H$	WUF1_A1	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E140$_H$	WUF1_A2	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E200$_H$	WUF0_I0	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E210$_H$	WUF0_I1	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E320$_H$	WUF1_I0	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E330$_H$	WUF1_I1	Wake-Up Factor Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E004$_H$	WUFMSK0_A0	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E014$_H$	WUFMSK0_A1	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E024$_H$	WUFMSK0_A2	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E124$_H$	WUFMSK1_A0	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E134$_H$	WUFMSK1_A1	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E144$_H$	WUFMSK1_A2	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E204$_H$	WUFMSK0_I0	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E214$_H$	WUFMSK0_I1	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E324$_H$	WUFMSK1_I0	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E334$_H$	WUFMSK1_I1	Wake-Up Factor Mask Registers	32	FFFF FFFF$_H$	PBG20 #2	
FF98 E008$_H$	WUFC0_A0	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E018$_H$	WUFC0_A1	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E028$_H$	WUFC0_A2	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E128$_H$	WUFC1_A0	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E138$_H$	WUFC1_A1	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E148$_H$	WUFC1_A2	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E208$_H$	WUFC0_I0	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E218$_H$	WUFC0_I1	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E328$_H$	WUFC1_I0	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E338$_H$	WUFC1_I1	Wake-Up Factor Clear Registers	32	0000 0000$_H$	PBG20 #2	
FF98 E1A0$_H$	WUFMON	Wake-Up Factor Monitor Register	32	0000 0000$_H$	PBG20 #2	

15.2.3 Reset of Registers

Register reset condition is shown in **Table 15.13, Register reset condition**.

Table 15.13 Register reset condition (1/2)

Symbol	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
MSR_RSCFD	√	√	√	√	√	√*1	—
MSR_FLXA	√	√	√	√	√	√*1	—
MSR_GTM	√	√	√	√	√	√*1	—
MSR_ETNB	√	√	√	√	√	√*1	—
MSR_RSENT	√	√	√	√	√	√*1	—
MSR_MSPI	√	√	√	√	√	√*1	—
MSR_RLIN3	√	√	√	√	√	√*1	—
MSR_ADCJ_ISO	√	√	√	√	√	√*1	—
MSR_CXPI	√	√	√	√	√	√*1	—
MSR_MMCA	√	√	√	√	√	√*1	—
MSR_ENCA	√	√	√	√	√	√*1	—
MSR_PSI5	√	√	√	√	√	√*1	—
MSR_PSI5S	√	√	√	√	√	√*1	—
MSR_PWMD	√	√	√	√	√	√*1	—
MSR_RHSIF	√	√	√	√	√	√*1	—
MSR_RIIC	√	√	√	√	√	√*1	—
MSR_SCI3	√	√	√	√	√	√*1	—
MSR_TAPA	√	√	√	√	√	√*1	—
MSR_TAUD	√	√	√	√	√	√*1	—
MSR_TAUJ_ISO	√	√	√	√	√	√*1	—
MSR_TPBA	√	√	√	√	√	√*1	—
MSR_TSG3	√	√	√	√	√	√*1	—
MSR_OSTM	√	√	√	√	√	√*1	—
STBCKCPROT	√	√	√	—	√	—	—
MSRKCPROT	√	√	√	—	√	—	—
STBC0PSC	√	√	√	√	—	—	—
STBC0STPT	√	√	√	√	—	—	—
PWRGD_CNT	√	√*2	—	—	—	—	—
IOHOLD0	√	√	√	√	—	—	—
IOHOLD2	√	√	√	√	—	—	—
MSR_ADCJ_AWO	√	√	√	√	—	—	—
MSR_RTCA	√	√	√	—	—	—	—
MSR_TAUJ_AWO	√	√	√	√	—	—	—
MSR_WDTB_AWO	√	√	√	√	—	—	—
WUF0_A0	√	√	√	√	—	—	—
WUF0_A1	√	√	√	√	—	—	—
WUF0_A2	√	√	√	√	—	—	—
WUF1_A0	√	√	√	√	—	—	—
WUF1_A1	√	√	√	√	—	—	—

Table 15.13 Register reset condition (2/2)

Symbol	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
WUF1_A2	√	√	√	√	—	—	—
WUF0_I0	√	√	√	√	—	—	—
WUF0_I1	√	√	√	√	—	—	—
WUF1_I0	√	√	√	√	—	—	—
WUF1_I1	√	√	√	√	—	—	—
WUFMSK0_A0	√	√	√	√	—	—	—
WUFMSK0_A1	√	√	√	√	—	—	—
WUFMSK0_A2	√	√	√	√	—	—	—
WUFMSK1_A0	√	√	√	√	—	—	—
WUFMSK1_A1	√	√	√	√	—	—	—
WUFMSK1_A2	√	√	√	√	—	—	—
WUFMSK0_I0	√	√	√	√	—	—	—
WUFMSK0_I1	√	√	√	√	—	—	—
WUFMSK1_I0	√	√	√	√	—	—	—
WUFMSK1_I1	√	√	√	√	—	—	—
WUFC0_A0	—	—	—	—	—	—	—
WUFC0_A1	—	—	—	—	—	—	—
WUFC0_A2	—	—	—	—	—	—	—
WUFC1_A0	—	—	—	—	—	—	—
WUFC1_A1	—	—	—	—	—	—	—
WUFC1_A2	—	—	—	—	—	—	—
WUFC0_I0	—	—	—	—	—	—	—
WUFC0_I1	—	—	—	—	—	—	—
WUFC1_I0	—	—	—	—	—	—	—
WUFC1_I1	—	—	—	—	—	—	—
WUFMON	√	√	√	√	—	—	—

Note 1. The reset is controlled bit by bit. The reset by Module Reset register bit (SWMRESA_<name>_n) initializes Module Standby register bit (MS_<name>_n).

Note 2. Only VMON Reset which is AWOVDD detection.

15.2.4 Details of Standby Controller Control Registers

15.2.4.1 MSR_RSCFD — Module Standby Register for RS-CANFD

This register is used to control the stop modes of the RS-CANFD.

Access: This register can be read or written in 32-bit units.

Address: FF98 1000_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_RSCFD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RSCFD_1	MS_RSCFD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.14 MSR_RSCFD Register Contents

Bit Position	Bit Name	Function
31	STPMSK_RSCFD	This bit controls the operation of RS-CANFD Clock in STOP mode with MS_RSCFD_n (n = 0, 1) = 0. 0 : The clocks are stopped in STOP mode. 1 : The clocks are supplied in STOP mode.
30 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_RSCFD_1	Setting this bit to 1 stops all clocks connected to RSCFD1 (CAN8 to CAN15), E7RC11, E7RC12, and E7RC13. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSCFD.SWMRESA_RSCFD_1 = 1, this bit is returned to the value after reset.
0	MS_RSCFD_0	Setting this bit to 1 stops all clocks connected to RSCFD0 (CAN0 to CAN7), E7RC01, E7RC02, and E7RC03. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSCFD.SWMRESA_RSCFD_0 = 1, this bit is returned to the value after reset.

15.2.4.2 MSR_FLXA — Module Standby Register for FLXA

This register is used to control the stop modes of the FLXA.

Access: This register can be read or written in 32-bit units.

Address: FF98 1010$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_FLXA_1	MS_FLXA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.15 MSR_FLXA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_FLXA_1	Setting this bit to 1 stops all clocks connected to FLXA1, E7FR10, E7FR11, and E7FR12. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_FLXA.SWMRESA_FLXA_1 = 1, this bit is returned to the value after reset.
0	MS_FLXA_0	Setting this bit to 1 stops all clocks connected to FLXA0, E7FR00, E7FR01, and E7FR02. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_FLXA.SWMRESA_FLXA_0 = 1, this bit is returned to the value after reset.

15.2.4.3 MSR_GTM — Module Standby Register for GTM

This register is used to control the stop modes of the GTM.

Access: This register can be read or written in 32-bit units.

Address: FF98 1020_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_GTM_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.16 MSR_GTM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_GTM_0	Setting this bit to 1 stops all clocks connected to GTM0, GTM0_1, E7GT00, E7GT01, E7GT10, E7GT11, E7GT20, E7GT21, E7GT30, and E7GT31. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_GTM.SWMRESA_GTM_0 = 1, this bit is returned to the value after reset.

15.2.4.4 MSR_ETNB — Module Standby Register for ETNB

This register is used to control the stop modes of the ETNB.

Access: This register can be read or written in 32-bit units.

Address: FF98 1030_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ETNB_1	MS_ETNB_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.17 MSR_ETNB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_ETNB_1	Setting this bit to 1 stops all clocks connected to ETNB1, E7GE00, and E7GE01. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ETNB.SWMRESA_ETNB_1 = 1, this bit is returned to the value after reset.
0	MS_ETNB_0	Setting this bit to 1 stops all clocks connected to ETNB0, E7ME00, and E7ME01. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ETNB.SWMRESA_ETNB_0 = 1, this bit is returned to the value after reset.

15.2.4.5 MSR_RSENT — Module Standby Register for RSENT

This register is used to control the stop modes of the RSENT.

Access: This register can be read or written in 32-bit units.

Address: FF98 1040$_H$

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MS_RSENT_7	MS_RSENT_6	MS_RSENT_5	MS_RSENT_4	MS_RSENT_3	MS_RSENT_2	MS_RSENT_1	MS_RSENT_0
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.18 MSR_RSENT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MS_RSENT_7	Setting this bit to 1 stops all clocks connected to RSENT7. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_7 = 1, this bit is returned to the value after reset.
6	MS_RSENT_6	Setting this bit to 1 stops all clocks connected to RSENT6. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_6 = 1, this bit is returned to the value after reset.
5	MS_RSENT_5	Setting this bit to 1 stops all clocks connected to RSENT5. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_5 = 1, this bit is returned to the value after reset.
4	MS_RSENT_4	Setting this bit to 1 stops all clocks connected to RSENT4. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_4 = 1, this bit is returned to the value after reset.
3	MS_RSENT_3	Setting this bit to 1 stops all clocks connected to RSENT3. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_3 = 1, this bit is returned to the value after reset.

Table 15.18 MSR_RSENT Register Contents (2/2)

Bit Position	Bit Name	Function
2	MS_RSENT_2	Setting this bit to 1 stops all clocks connected to RSENT2. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_2 = 1, this bit is returned to the value after reset.
1	MS_RSENT_1	Setting this bit to 1 stops all clocks connected to RSENT1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_1 = 1, this bit is returned to the value after reset.
0	MS_RSENT_0	Setting this bit to 1 stops all clocks connected to RSENT0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RSENT.SWMRESA_RSENT_0 = 1, this bit is returned to the value after reset.

NOTE

Register RSENTTSSEL is always accessible regardless of the value of MSR_RSENT.

15.2.4.6 MSR_MSPI — Module Standby Register for MSPI

This register is used to control the stop modes of the MSPI.

Access: This register can be read or written in 32-bit units.

Address: FF98 1050$_H$

Value after reset: 0000 03FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MS_MSPI_9	MS_MSPI_8	MS_MSPI_7	MS_MSPI_6	MS_MSPI_5	MS_MSPI_4	MS_MSPI_3	MS_MSPI_2	MS_MSPI_1	MS_MSPI_0
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.19 MSR_MSPI Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	MS_MSPI_9	Setting this bit to 1 stops all clocks connected to MSPI9, MSPI9_INTF, and E7MS09. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_9 = 1, this bit is returned to the value after reset.
8	MS_MSPI_8	Setting this bit to 1 stops all clocks connected to MSPI8, MSPI8_INTF, and E7MS08. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_8 = 1, this bit is returned to the value after reset.
7	MS_MSPI_7	Setting this bit to 1 stops all clocks connected to MSPI7, MSPI7_INTF, and E7MS07. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_7 = 1, this bit is returned to the value after reset.
6	MS_MSPI_6	Setting this bit to 1 stops all clocks connected to MSPI6, MSPI6_INTF, and E7MS06. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_6 = 1, this bit is returned to the value after reset.
5	MS_MSPI_5	Setting this bit to 1 stops all clocks connected to MSPI5, MSPI5_INTF, and E7MS05. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_5 = 1, this bit is returned to the value after reset.

Table 15.19 MSR_MSPI Register Contents (2/2)

Bit Position	Bit Name	Function
4	MS_MSPI_4	Setting this bit to 1 stops all clocks connected to MSPI4, MSPI4_INTF, and E7MS04 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_4 = 1, this bit is returned to the value after reset.
3	MS_MSPI_3	Setting this bit to 1 stops all clocks connected to MSPI3, MSPI3_INTF, and E7MS03. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_3 = 1, this bit is returned to the value after reset.
2	MS_MSPI_2	Setting this bit to 1 stops all clocks connected to MSPI2, MSPI2_INTF, and E7MS02. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_2 = 1, this bit is returned to the value after reset.
1	MS_MSPI_1	Setting this bit to 1 stops all clocks connected to MSPI1, MSPI1_INTF, and E7MS01 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_1 = 1, this bit is returned to the value after reset.
0	MS_MSPI_0	Setting this bit to 1 stops all clocks connected to MSPI0, MSPI0_INTF, and E7MS00. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MSPI.SWMRESA_MSPI_0 = 1, this bit is returned to the value after reset.

NOTE

Register MSPITGCTLn, MSPITGDMAALT, and MSPITGDTSALT are always accessible regardless of the value of MSR_MSPI.

15.2.4.7 MSR_RLIN3 — Module Standby Register for RLIN3

This register is used to control the stop modes of the RLIN3.

Access: This register can be read or written in 32-bit units.

Address: FF98 1060$_H$

Value after reset: 00FF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_RLIN3	—	—	—	—	—	—	—	MS_RLIN3_23	MS_RLIN3_22	MS_RLIN3_21	MS_RLIN3_20	MS_RLIN3_19	MS_RLIN3_18	MS_RLIN3_17	MS_RLIN3_16
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MS_RLIN3_15	MS_RLIN3_14	MS_RLIN3_13	MS_RLIN3_12	MS_RLIN3_11	MS_RLIN3_10	MS_RLIN3_9	MS_RLIN3_8	MS_RLIN3_7	MS_RLIN3_6	MS_RLIN3_5	MS_RLIN3_4	MS_RLIN3_3	MS_RLIN3_2	MS_RLIN3_1	MS_RLIN3_0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.20 MSR_RLIN3 Register Contents (1/3)

Bit Position	Bit Name	Function
31	STPMSK_RLIN3	This bit controls the operation of RLIN3 Clock in STOP and Cyclic STOP mode with MS_RLIN3_n (n = 0 to 23) = 0. 0 : The clocks are stopped in STOP and Cyclic STOP mode. 1 : The clocks are supplied in STOP and Cyclic STOP mode.
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	MS_RLIN3_23	Setting this bit to 1 stops all clocks connected to RLIN323. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_23 = 1, this bit is returned to the value after reset.
22	MS_RLIN3_22	Setting this bit to 1 stops all clocks connected to RLIN322. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_22 = 1, this bit is returned to the value after reset.
21	MS_RLIN3_21	Setting this bit to 1 stops all clocks connected to RLIN321. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_21 = 1, this bit is returned to the value after reset.
20	MS_RLIN3_20	Setting this bit to 1 stops all clocks connected to RLIN320. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_20 = 1, this bit is returned to the value after reset.
19	MS_RLIN3_19	Setting this bit to 1 stops all clocks connected to RLIN319. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_19 = 1, this bit is returned to the value after reset.

Table 15.20 MSR_RLIN3 Register Contents (2/3)

Bit Position	Bit Name	Function
18	MS_RLIN3_18	Setting this bit to 1 stops all clocks connected to RLIN318. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_18 = 1, this bit is returned to the value after reset.
17	MS_RLIN3_17	Setting this bit to 1 stops all clocks connected to RLIN317. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_17 = 1, this bit is returned to the value after reset.
16	MS_RLIN3_16	Setting this bit to 1 stops all clocks connected to RLIN316. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_16 = 1, this bit is returned to the value after reset.
15	MS_RLIN3_15	Setting this bit to 1 stops all clocks connected to RLIN315. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_15 = 1, this bit is returned to the value after reset.
14	MS_RLIN3_14	Setting this bit to 1 stops all clocks connected to RLIN314. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_14 = 1, this bit is returned to the value after reset.
13	MS_RLIN3_13	Setting this bit to 1 stops all clocks connected to RLIN313. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_13 = 1, this bit is returned to the value after reset.
12	MS_RLIN3_12	Setting this bit to 1 stops all clocks connected to RLIN312. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_12 = 1, this bit is returned to the value after reset.
11	MS_RLIN3_11	Setting this bit to 1 stops all clocks connected to RLIN311. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_11 = 1, this bit is returned to the value after reset.
10	MS_RLIN3_10	Setting this bit to 1 stops all clocks connected to RLIN310. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_10 = 1, this bit is returned to the value after reset.
9	MS_RLIN3_9	Setting this bit to 1 stops all clocks connected to RLIN39. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_9 = 1, this bit is returned to the value after reset.

Table 15.20 MSR_RLIN3 Register Contents (3/3)

Bit Position	Bit Name	Function
8	MS_RLIN3_8	Setting this bit to 1 stops all clocks connected to RLIN38. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_8 = 1, this bit is returned to the value after reset.
7	MS_RLIN3_7	Setting this bit to 1 stops all clocks connected to RLIN37. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_7 = 1, this bit is returned to the value after reset.
6	MS_RLIN3_6	Setting this bit to 1 stops all clocks connected to RLIN36. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_6 = 1, this bit is returned to the value after reset.
5	MS_RLIN3_5	Setting this bit to 1 stops all clocks connected to RLIN35. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_5 = 1, this bit is returned to the value after reset.
4	MS_RLIN3_4	Setting this bit to 1 stops all clocks connected to RLIN34. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_4 = 1, this bit is returned to the value after reset.
3	MS_RLIN3_3	Setting this bit to 1 stops all clocks connected to RLIN33. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_3 = 1, this bit is returned to the value after reset.
2	MS_RLIN3_2	Setting this bit to 1 stops all clocks connected to RLIN32. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_2 = 1, this bit is returned to the value after reset.
1	MS_RLIN3_1	Setting this bit to 1 stops all clocks connected to RLIN31. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_1 = 1, this bit is returned to the value after reset.
0	MS_RLIN3_0	Setting this bit to 1 stops all clocks connected to RLIN30. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RLIN3.SWMRESA_RLIN3_0 = 1, this bit is returned to the value after reset.

15.2.4.8 MSR_ADCJ_ISO — Module Standby Register for ADCJ0/ADCJ1 (ISO area)

This register is used to control the stop modes of the ADCJ0/ADCJ1 (ISO area).

Access: This register can be read or written in 32-bit units.

Address: FF98 1070_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ADCJ_1	MS_ADCJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.21 MSR_ADCJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_ADCJ_1	Setting this bit to 1 stops all clocks connected to ADCJ1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ADCJ_ISO.SWMRESA_ADCJ_1 = 1, this bit is returned to the value after reset.
0	MS_ADCJ_0	Setting this bit to 1 stops all clocks connected to ADCJ0 and AVSEG. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ADCJ_ISO.SWMRESA_ADCJ_0 = 1, this bit is returned to the value after reset.

15.2.4.9 MSR_CXPI — Module Standby Register for CXPI

This register is used to control the stop modes of the CXPI.

Access: This register can be read or written in 32-bit units.

Address: FF98 1080_H

Value after reset: 0000 $000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MS_CXPI_3	MS_CXPI_2	MS_CXPI_1	MS_CXPI_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 15.22 MSR_CXPI Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MS_CXPI_3	Setting this bit to 1 stops all clocks connected to CXP13. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_CXPI.SWMRESA_CXPI_3 = 1, this bit is returned to the value after reset.
2	MS_CXPI_2	Setting this bit to 1 stops all clocks connected to CXP12. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_CXPI.SWMRESA_CXPI_2 = 1, this bit is returned to the value after reset.
1	MS_CXPI_1	Setting this bit to 1 stops all clocks connected to CXP11. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_CXPI.SWMRESA_CXPI_1 = 1, this bit is returned to the value after reset.
0	MS_CXPI_0	Setting this bit to 1 stops all clocks connected to CXP10. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_CXPI.SWMRESA_CXPI_0 = 1, this bit is returned to the value after reset.

15.2.4.10 MSR_MMCA — Module Standby Register for MMCA

This register is used to control the stop modes of the MMCA.

Access: This register can be read or written in 32-bit units.

Address: FF98 1090_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_MMCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.23 MSR_MMCA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_MMCA_0	Setting this bit to 1 stops all clocks connected to MMCA0, E7MM00, and E7MM01. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_MMCA.SWMRESA_MMCA_0 = 1, this bit is returned to the value after reset.

15.2.4.11 MSR_ENCA — Module Standby Register for ENCA

This register is used to control the stop modes of the ENCA.

Access: This register can be read or written in 32-bit units.

Address: FF98 10A0$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ENCA_1	MS_ENCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.24 MSR_ENCA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_ENCA_1	Setting this bit to 1 stops all clocks connected to ENCA1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ENCA.SWMRESA_ENCA_1 = 1, this bit is returned to the value after reset.
0	MS_ENCA_0	Setting this bit to 1 stops all clocks connected to ENCA0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_ENCA.SWMRESA_ENCA_0 = 1, this bit is returned to the value after reset.

15.2.4.12 MSR_PSI5 — Module Standby Register for PSI5

This register is used to control the stop modes of the PSI5.

Access: This register can be read or written in 32-bit units.

Address: FF98 10B0$_H$

Value after reset: 0000 000F$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MS_PSI5_3	MS_PSI5_2	MS_PSI5_1	MS_PSI5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 15.25 MSR_PSI5 Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MS_PSI5_3	Setting this bit to 1 stops all clocks connected to PSI53. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5.SWMRESA_PSI5_3 = 1, this bit is returned to the value after reset.
2	MS_PSI5_2	Setting this bit to 1 stops all clocks connected to PSI52. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5.SWMRESA_PSI5_2 = 1, this bit is returned to the value after reset.
1	MS_PSI5_1	Setting this bit to 1 stops all clocks connected to PSI51. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5.SWMRESA_PSI5_1 = 1, this bit is returned to the value after reset.
0	MS_PSI5_0	Setting this bit to 1 stops all clocks connected to PSI50. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5.SWMRESA_PSI5_0 = 1, this bit is returned to the value after reset.

NOTE

Register PSI5TSSEL is always accessible regardless of the value of MSR_PSI5.

15.2.4.13 MSR_PSI5S — Module Standby Register for PSI5S

This register is used to control the stop modes of the PSI5S.

Access: This register can be read or written in 32-bit units.

Address: FF98 $10C0_H$

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_PSI5S_1	MS_PSI5S_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.26 MSR_PSI5S Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_PSI5S_1	Setting this bit to 1 stops all clocks connected to PSI5S1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5S.SWMRESA_PSI5S_1 = 1, this bit is returned to the value after reset.
0	MS_PSI5S_0	Setting this bit to 1 stops all clocks connected to PSI5S0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PSI5S.SWMRESA_PSI5S_0 = 1, this bit is returned to the value after reset.

15.2.4.14 MSR_PWMD — Module Standby Register for PWM-Diag

This register is used to control the stop modes of the PWM-Diag.

Access: This register can be read or written in 32-bit units.

Address: FF98 $10D0_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_PWMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.27 MSR_PWMD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_PWMD	Setting this bit to 1 stops all clocks connected to PWM-Diag (PWBA0, PWGC0-95, PWSD0, SLPW, and PWGCINTF). 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_PWMD.SWMRESA_PWMD = 1, this bit is returned to the value after reset.

15.2.4.15 MSR_RHSIF — Module Standby Register for RHSIF

This register is used to control the stop modes of the RHSIF.

Access: This register can be read or written in 32-bit units.

Address: FF98 $10E0_H$

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RHSIF_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.28 MSR_RHSIF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_RHSIF_0	Setting this bit to 1 stops communication clocks (Internal) connected to RHSIF0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RHSIF.SWMRESA_RHSIF_0 = 1, this bit is returned to the value after reset.

15.2.4.16 MSR_RIIC — Module Standby Register for RIIC

This register is used to control the stop modes of the RIIC.

Access: This register can be read or written in 32-bit units.

Address: FF98 $10F0_H$

Value after reset: $0000\ 0003_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RIIC_1	MS_RIIC_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.29 MSR_RIIC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_RIIC_1	Setting this bit to 1 stops all clocks connected to RIIC1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RIIC.SWMRESA_RIIC_1 = 1, this bit is returned to the value after reset.
0	MS_RIIC_0	Setting this bit to 1 stops all clocks connected to RIIC0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_RIIC.SWMRESA_RIIC_0 = 1, this bit is returned to the value after reset.

15.2.4.17 MSR_SCI3 — Module Standby Register for SCI3

This register is used to control the stop modes of the SCI3.

Access: This register can be read or written in 32-bit units.

Address: FF98 1100$_H$

Value after reset: 0000 0007$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_SCI3_2	MS_SCI3_1	MS_SCI3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 15.30 MSR_SCI3 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	MS_SCI3_2	Setting this bit to 1 stops all clocks connected to SCI32. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_SCI3.SWMRESA_SCI3_2 = 1, this bit is returned to the value after reset.
1	MS_SCI3_1	Setting this bit to 1 stops all clocks connected to SCI31. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_SCI3.SWMRESA_SCI3_1 = 1, this bit is returned to the value after reset.
0	MS_SCI3_0	Setting this bit to 1 stops all clocks connected to SCI30. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_SCI3.SWMRESA_SCI3_0 = 1, this bit is returned to the value after reset.

15.2.4.18 MSR_TAPA — Module Standby Register for TAPA

This register is used to control the stop modes of the TAPA.

Access: This register can be read or written in 32-bit units.

Address: FF98 1120$_H$

Value after reset: 0000 000F$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MS_TAPA_3	MS_TAPA_2	MS_TAPA_1	MS_TAPA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 15.31 MSR_TAPA Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MS_TAPA_3	Setting this bit to 1 stops all clocks connected to TAPA3. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAPA.SWMRESA_TAPA_3 = 1, this bit is returned to the value after reset.
2	MS_TAPA_2	Setting this bit to 1 stops all clocks connected to TAPA2. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAPA.SWMRESA_TAPA_2 = 1, this bit is returned to the value after reset.
1	MS_TAPA_1	Setting this bit to 1 stops all clocks connected to TAPA1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAPA.SWMRESA_TAPA_1 = 1, this bit is returned to the value after reset.
0	MS_TAPA_0	Setting this bit to 1 stops all clocks connected to TAPA0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAPA.SWMRESA_TAPA_0 = 1, this bit is returned to the value after reset.

15.2.4.19 MSR_TAUD — Module Standby Register for TAUD

This register is used to control the stop modes of the TAUD.

Access: This register can be read or written in 32-bit units.

Address: FF98 1130_H

Value after reset: 0000 0007_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_TAUD_2	MS_TAUD_1	MS_TAUD_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 15.32 MSR_TAUD Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	MS_TAUD_2	Setting this bit to 1 stops all clocks connected to TAUD2 and SELB_TAUD2I. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAUD.SWMRESA_TAUD_2 = 1, this bit is returned to the value after reset.
1	MS_TAUD_1	Setting this bit to 1 stops all clocks connected to TAUD1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAUD.SWMRESA_TAUD_1 = 1, this bit is returned to the value after reset.
0	MS_TAUD_0	Setting this bit to 1 stops all clocks connected to TAUD0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAUD.SWMRESA_TAUD_0 = 1, this bit is returned to the value after reset.

15.2.4.20 MSR_TAUJ_ISO — Module Standby Register for TAUJ0/TAUJ1 (ISO area)

This register is used to control the stop modes of the TAUJ0/TAUJ1 (ISO area).

Access: This register can be read or written in 32-bit units.

Address: FF98 1140$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_TAUJ_1	MS_TAUJ_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.33 MSR_TAUJ_ISO Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_TAUJ_1	Setting this bit to 1 stops all clocks connected to TAUJ1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAUJ_ISO.SWMRESA_TAUJ_1 = 1, this bit is returned to the value after reset.
0	MS_TAUJ_0	Setting this bit to 1 stops all clocks connected to TAUJ0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TAUJ_ISO.SWMRESA_TAUJ_0 = 1, this bit is returned to the value after reset.

15.2.4.21 MSR_TPBA — Module Standby Register for TPBA

This register is used to control the stop modes of the TPBA.

Access: This register can be read or written in 32-bit units.

Address: FF98 1150_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_TPBA_1	MS_TPBA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.34 MSR_TPBA Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_TPBA_1	Setting this bit to 1 stops all clocks connected to TPBA1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TPBA.SWMRESA_TPBA_1 = 1, this bit is returned to the value after reset.
0	MS_TPBA_0	Setting this bit to 1 stops all clocks connected to TPBA0. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TPBA.SWMRESA_TPBA_0 = 1, this bit is returned to the value after reset.

15.2.4.22 MSR_TSG3 — Module Standby Register for TSG3

This register is used to control the stop modes of the TSG3.

Access: This register can be read or written in 32-bit units.

Address: FF98 1160_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_TSG3_1	MS_TSG3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.35 MSR_TSG3 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_TSG3_1	Setting this bit to 1 stops all clocks connected to TSG31. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TSG3.SWMRESA_TSG3_1 = 1, this bit is returned to the value after reset.
0	MS_TSG3_0	Setting this bit to 1 stops all clocks connected to TSG30. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_TSG3.SWMRESA_TSG3_0 = 1, this bit is returned to the value after reset.

15.2.4.23 MSR_OSTM — Module Standby Register for OSTM

This register is used to control the stop modes of the OSTM.

Access: This register can be read or written in 32-bit units.

Address: FF98 1180$_H$

Value after reset: 0000 03FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MS_OSTM_9	MS_OSTM_8	MS_OSTM_7	MS_OSTM6_	MS_OSTM_5	MS_OSTM_4	MS_OSTM_3	MS_OSTM_2	MS_OSTM_1	MS_OSTM_0
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.36 MSR_OSTM Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	MS_OSTM_9	Setting this bit to 1 stops all clocks connected to OSTM9 and IC0CKSEL9. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_9 = 1, this bit is returned to the value after reset.
8	MS_OSTM_8	Setting this bit to 1 stops all clocks connected to OSTM8 and IC0CKSEL8. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_8 = 1, this bit is returned to the value after reset.
7	MS_OSTM_7	Setting this bit to 1 stops all clocks connected to OSTM7. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_7 = 1, this bit is returned to the value after reset.
6	MS_OSTM_6	Setting this bit to 1 stops all clocks connected to OSTM6. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_6 = 1, this bit is returned to the value after reset.
5	MS_OSTM_5	Setting this bit to 1 stops all clocks connected to OSTM5. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_5 = 1, this bit is returned to the value after reset.
4	MS_OSTM_4	Setting this bit to 1 stops all clocks connected to OSTM4. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_4 = 1, this bit is returned to the value after reset.

Table 15.36 MSR_OSTM Register Contents (2/2)

Bit Position	Bit Name	Function
3	MS_OSTM_3	Setting this bit to 1 stops all clocks connected to OSTM3. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_3 = 1, this bit is returned to the value after reset.
2	MS_OSTM_2	Setting this bit to 1 stops all clocks connected to OSTM2. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_2 = 1, this bit is returned to the value after reset.
1	MS_OSTM_1	Setting this bit to 1 stops all clocks connected to OSTM1. 0 : The clocks are supplied. 1 : The clocks are stopped. When written to SWMRESA_OSTM.SWMRESA_OSTM_1 = 1, this bit is returned to the value after reset.
0	MS_OSTM_0	Setting this bit to 1 stops all clocks connected to OSTM0. 0 : The clocks are supplied. 1 : The clocks are stopped When written to SWMRESA_OSTM.SWMRESA_OSTM_0 = 1, this bit is returned to the value after reset.

15.2.4.24 STBCKCPROT — Standby controller Register Key Code Protection Register

STBCKCPROT register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 1700$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STBCKCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STBCKCPROT[15:1]															STBCKCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 15.37 STBCKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	STBCKCPROT* [1]	Enable or disable modification of the STBCKCE bit. The value written is not retained. These bits are always read as 0.
0	STBCKCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing to protected registers.
Write A5A5A501$_H$ to this register to enable writing to protected registers.

15.2.4.25 MSRKCPROT — Module Standby Register Key Code Protection Register

MSRKCPROT register is used for protection against writing operation to the registers which may have a material effect on the system so that the application system is not incorrectly stopped due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FF98 1710$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 15.38 MSRKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT*1	Enable or disable modification of the KCE bit. The value written is not retained. These bits are always read as 0.
0	KCE	Key Code Enable bit 0: Disables write access of protected registers 1: Enables write access of protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing to protected registers.
Write A5A5A501$_H$ to this register to enable writing to protected registers.

15.2.4.26 STBC0PSC — Power Save Control Register

This register is used for transition to DeepSTOP mode.

Access: This register can be read or written in 32-bit units.

Address: FF98 8C00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0 DISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 15.39 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	STBC0DISTRG*1	0 : No effect 1 : Transition to DeepSTOP mode This bit is cleared automatically after transition to the DeepSTOP mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. Before setting the STBC0PSC.STBC0DISTRG bit to 1, confirm Main OSC and PLL are in a stable state. Refer to **Section 13.5.4.1, PLLE — PLL Enable Register** for confirmation of the stable state of PLL. Refer to **Section 13.5.4.4, MOSCE — Main OSC Enable Register** for confirmation of the stable state of Main OSC.

15.2.4.27 STBC0STPT — Power Stop Trigger Register

This register is used for STOP mode or Cyclic STOP mode transition.

Access: This register can be read or written in 32-bit units.

Address: FF98 8C04$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0STPTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.40 STBC0STPT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STBC0STPTRG*1	0 : No effect. 1 : Transition to STOP / Cyclic STOP mode – In RUN mode: Transition to STOP mode – In Cyclic RUN mode: Transition to Cyclic STOP mode This bit is cleared automatically after transition to the STOP/Cyclic STOP mode.

Note 1. Before setting the STBC0STPT.STBC0STPTRG bit to 1, confirm Main OSC and PLL are in a stable state. Refer to **Section 13.5.4.1, PLLE — PLL Enable Register** for confirmation of the stable state of PLL. Refer to **Section 13.5.4.4, MOSCE — Main OSC Enable Register** for confirmation of the stable state of Main OSC.

15.2.4.28 PWRGD_CNT — Power Good Counter Register

This register determines a minimum waiting period for ISOVDD to be stable after PWRCTL is high level.

This register is used when returning from DeepSTOP mode.

Access: This register can be read or written in 32-bit units.

Address: FF98 8C08$_H$

Value after reset: 0000 0FFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWRGD_CNT[11:0]											
Value after reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.41 PWRGD_CNT Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWRGD_CNT*1	Power good counter value Wait time = (PWRGD_CNT[11:0] + 1) x 1/f_{RL}*1 f_{RL}: Low Speed Internal Oscillator (LS IntOSC) frequency.

Note 1. When using SVR, set the value according to the SVR guideline (Application Note) for this product. When not using SVR, set the wait time until the VDD input voltage becomes stable after the PWRCTL pin outputs a high level. See **Section 10.5, VDD Power Supply**.

15.2.4.29 IOHOLD0 — I/O Buffer Hold Control Register 0

This register specifies the hold state of the I/O buffer in DeepSTOP mode.

Access: This register can be read or written in 32-bit units.

Address: FF98 8D00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	IOHOLD_P24	IOHOLD_P23	IOHOLD_P22	IOHOLD_P21	IOHOLD_P20	IOHOLD_P19	IOHOLD_P18	IOHOLD_P17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IOHOLD_P12	IOHOLD_P11	IOHOLD_P10	IOHOLD_P9	—	—	—	IOHOLD_P5	IOHOLD_P4	IOHOLD_P3	IOHOLD_P2	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R

Table 15.42 IOHOLD0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	IOHOLD_P24	This bit indicates the I/O hold state of each port group "P24". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
23	IOHOLD_P23	This bit indicates the I/O hold state of each port group "P23". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
22	IOHOLD_P22	This bit indicates the I/O hold state of each port group "P22". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
21	IOHOLD_P21	This bit indicates the I/O hold state of each port group "P21". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
20	IOHOLD_P20	This bit indicates the I/O hold state of each port group "P20". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
19	IOHOLD_P19	This bit indicates the I/O hold state of each port group "P19". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.

Table 15.42 IOHOLD0 Register Contents (2/3)

Bit Position	Bit Name	Function
18	IOHOLD_P18	This bit indicates the I/O hold state of each port group "P18". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
17	IOHOLD_P17	This bit indicates the I/O hold state of each port group "P17". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
16 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	IOHOLD_P12	This bit indicates the I/O hold state of each port group "P12". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
11	IOHOLD_P11	This bit indicates the I/O hold state of each port group "P11". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
10	IOHOLD_P10	This bit indicates the I/O hold state of each port group "P10". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
9	IOHOLD_P9	This bit indicates the I/O hold state of each port group "P9". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
8 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	IOHOLD_P5	This bit indicates the I/O hold state of each port group "P5". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
4	IOHOLD_P4	This bit indicates the I/O hold state of each port group "P4". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.

Table 15.42 IOHOLD0 Register Contents (3/3)

Bit Position	Bit Name	Function
3	IOHOLD_P3	This bit indicates the I/O hold state of each port group "P3". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
2	IOHOLD_P2	This bit indicates the I/O hold state of each port group "P2". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

15.2.4.30 IOHOLD2 — I/O Buffer Hold Control Register 2

This register specifies the hold state of the I/O buffer in DeepSTOP mode.

Access: This register can be read or written in 32-bit units.

Address: FF98 8D08$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IOHOLD_AP3	IOHOLD_AP2	IOHOLD_AP1	IOHOLD_AP0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 15.43 IOHOLD2 Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	IOHOLD_AP3	This bit indicates the I/O hold state of each port group "AP3". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
2	IOHOLD_AP2	This bit indicates the I/O hold state of each port group "AP2". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
1	IOHOLD_AP1	This bit indicates the I/O hold state of each port group "AP1". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.
0	IOHOLD_AP0	This bit indicates the I/O hold state of each port group "AP0". 0 : I/O hold state is released 1 : I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is ignored. To release the I/O hold state after the wake-up, set this bit to 0 by software.

15.2.4.31 MSR_ADCJ_AWO — Module Standby Register for ADCJ2 (AWO area)

This register is used to control the stop modes of the ADCJ2 (AWO area).

Access: This register can be read or written in 32-bit units.

Address: FF98 8E00$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_ADCJ_AWO	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_ADCJ_2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.44 MSR_ADCJ_AWO Register Contents

Bit Position	Bit Name	Function
31	STPMSK_ADCJ_AWO	This bit controls the operation of ADCJ2 Clock in STOP and DeepSTOP and Cyclic STOP mode with MS_ADCJ_2 = 0. 0 : The clocks are stopped in STOP and DeepSTOP and Cyclic STOP mode. 1 : The clocks are supplied in STOP and DeepSTOP and Cyclic STOP mode.
30 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_ADCJ_2	Setting this bit to 1 stops all clocks connected to ADCJ2 and ADCJ2SGTSELx. 0 : The clocks are supplied. 1 : The clocks are stopped.

NOTE

When operating LPS in Analog Input Mode

- Set the MS_ADCJ_2 = 0.
- Even though setting the STPMSK_ADCJ_AWO = 0, ADCJ2 Clock can operate only while LPS is operating (during A/D conversion) in chip standby mode.

15.2.4.32 MSR_RTCA — Module Standby Register for RTCA

This register is used to control the stop modes of the RTCA.

Access: This register can be read or written in 32-bit units.

Address: FF98 8E10$_H$

Value after reset: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_RTCA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_RTCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.45 MSR_RTCA Register Contents

Bit Position	Bit Name	Function
31	STPMSK_RTCA	This bit controls the operation of RTCA Clock in STOP and DeepSTOP and Cyclic STOP mode with MS_RTCA_0 = 0. 0 : The clocks are stopped in STOP and DeepSTOP and Cyclic STOP mode. 1 : The clocks are supplied in STOP and DeepSTOP and Cyclic STOP mode.
30 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MS_RTCA_0	Setting this bit to 1 stops all clocks connected to RTCA0. 0 : The clocks are supplied. 1 : The clocks are stopped.

15.2.4.33 MSR_TAUJ_AWO — Module Standby Register for TAUJ2/TAUJ3 (AWO area)

This register is used to control the stop modes of the TAUJ2/TAUJ3 (AWO area).

Access: This register can be read or written in 32-bit units.

Address: FF98 8E20$_H$

Value after reset: 0000 0003$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_TAUJ_AWO	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MS_TAUJ_3	MS_TAUJ_2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 15.46 MSR_TAUJ_AWO Register Contents

Bit Position	Bit Name	Function
31	STPMSK_TAUJ_AWO	This bit controls the operation of TAUJ2/TAUJ3 Clock in STOP and DeepSTOP and Cyclic STOP mode with MS_TAUJ_n (n = 2, 3) = 0. 0 : The clocks are stopped in STOP and DeepSTOP and Cyclic STOP mode. 1 : The clocks are supplied in STOP and DeepSTOP and Cyclic STOP mode.
30 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MS_TAUJ_3	Setting this bit to 1 stops all clocks connected to TAUJ3 and SELB_TAUJ3I. 0 : The clocks are supplied. 1 : The clocks are stopped.
0	MS_TAUJ_2	Setting this bit to 1 stops all clocks connected to TAUJ2 and SELB_TAUJ2I. 0 : The clocks are supplied. 1 : The clocks are stopped.

15.2.4.34 MSR_WDTB_AWO — Module Standby Register for WDTB (AWO area)

This register is used to control the stop modes of the WDTB (AWO area).

Access: This register can be read or written in 32-bit units.

Address: FF98 8E30$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STPMSK_WDTB_AWO	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.47 MSR_WDTB_AWO Register Contents

Bit Position	Bit Name	Function
31	STPMSK_ WDTB_AWO	This bit controls the operation of WDTBA Clock in STOP and DeepSTOP and Cyclic STOP mode. 0 : The clocks are stopped in STOP and DeepSTOP and Cyclic STOP mode. 1 : The clocks are supplied in STOP and DeepSTOP and Cyclic STOP mode.
30 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

15.2.4.35 WUF0_Ax/WUF1_Ax/WUF0_Ix/WUF1_Ix — Wake-Up Factor Registers

These registers indicate the generation of Wake-Up Factors.

Access: This register is a read-only register that can be read in 32-bit units.

Address:

WUF0_A0 : FF98 E000$_H$	WUF0_A1 : FF98 E010$_H$
WUF0_A2 : FF98 E020$_H$	WUF1_A0 : FF98 E120$_H$
WUF1_A1 : FF98 E130$_H$	WUF1_A2 : FF98 E140$_H$
WUF0_I0 : FF98 E200$_H$	WUF0_I1 : FF98 E210$_H$
WUF1_I0 : FF98 E320$_H$	WUF1_I1 : FF98 E330$_H$

Value after reset:

WUF0_A0 : 0000 0000$_H$	WUF0_A1 : 0000 0000$_H$
WUF0_A2 : 0000 0000$_H$	WUF1_A0 : 0000 0000$_H$
WUF1_A1 : 0000 0000$_H$	WUF1_A2 : 0000 0000H
WUF0_I0 : 0000 0000$_H$	WUF0_I1 : 0000 0000$_H$
WUF1_I0 : 0000 0000$_H$	WUF1_I1 : 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFn31	WUFn30	WUFn29	WUFn28	WUFn27	WUFn26	WUFn25	WUFn24	WUFn23	WUFn22	WUFn21	WUFn20	WUFn19	WUFn18	WUFn17	WUFn16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFn15	WUFn14	WUFn13	WUFn12	WUFn11	WUFn10	WUFn9	WUFn8	WUFn7	WUFn6	WUFn5	WUFn4	WUFn3	WUFn2	WUFn1	WUFn0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.48 WUF0_Ax/WUF1_Ax/WUF0_Ix/WUF1_Ix Register Contents (1/3)

Bit Position	Bit Name	Function
31	WUFn31	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
30	WUFn30	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
29	WUFn29	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
28	WUFn28	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
27	WUFn27	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
26	WUFn26	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
25	WUFn25	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
24	WUFn24	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
23	WUFn23	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated

Table 15.48 WUF0_Ax/WUF1_Ax/WUF0_Ix/WUF1_Ix Register Contents (2/3)

Bit Position	Bit Name	Function
22	WUFn22	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
21	WUFn21	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
20	WUFn20	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
19	WUFn19	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
18	WUFn18	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
17	WUFn17	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
16	WUFn16	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
15	WUFn15	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
14	WUFn14	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
13	WUFn13	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
12	WUFn12	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
11	WUFn11	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
10	WUFn10	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
9	WUFn9	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
8	WUFn8	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
7	WUFn7	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
6	WUFn6	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
5	WUFn5	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
4	WUFn4	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated

Table 15.48 WUF0_Ax/WUF1_Ax/WUF0_Ix/WUF1_Ix Register Contents (3/3)

Bit Position	Bit Name	Function
3	WUFn3	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
2	WUFn2	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
1	WUFn1	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated
0	WUFn0	Indicates the generation of a Wake-Up Factor. 0 : Wake-Up Factor is not generated 1 : Wake-Up Factor is generated

NOTE

While the WUFMSKnx (x = 0 to 31) bit in the Wake-Up Factor mask register is 1, WUFnx (x = 0 to 31) is not set to 1 at the generation of a Wake-Up Factor.

WUF0_Ax : WUF0_A0/WUF0_A1/WUF0_A2

WUF1_Ax : WUF1_A0/WUF1_A1/WUF1_A2

WUF0_Ix : WUF0_I0/WUF0_I1

WUF1_Ix : WUF1_I0/WUF1_I1

Wake-Up Factors

As for the assignment of Wake-Up Factors to the Wake-Up Factor register bits, see **Section 15.1.1, Types of Chip Standby Mode**.

The bit to which a Wake-Up Factor is not assigned is read as the value “0”.

15.2.4.36 WUFMSK0_Ax/WUFMSK1_Ax/WUFMSK0_Ix/WUFMSK1_Ix — Wake-Up Factor Mask Registers

These registers enable Wake-Up Factors.

Access: This register can be read or written in 32-bit units.

Address:
WUFMSK0_A0 : FF98 E004$_H$ WUFMSK0_A1 : FF98 E014$_H$
WUFMSK0_A2 : FF98 E024$_H$ WUFMSK1_A0 : FF98 E124$_H$
WUFMSK1_A1 : FF98 E134$_H$ WUFMSK1_A2 : FF98 E144$_H$
WUFMSK0_I0 : FF98 E204$_H$ WUFMSK0_I1 : FF98 E214$_H$
WUFMSK1_I0 : FF98 E324$_H$ WUFMSK1_I1 : FF98 E334$_H$

Value after reset:
WUFMSK0_A0 : FFFF FFFF$_H$ WUFMSK0_A1 : FFFF FFFF$_H$
WUFMSK0_A2 : FFFF FFFF$_H$ WUFMSK1_A0 : FFFF FFFF$_H$
WUFMSK1_A1 : FFFF FFFF$_H$ WUFMSK1_A2 : FFFF FFFF$_H$
WUFMSK0_I0 : FFFF FFFF$_H$ WUFMSK0_I1 : FFFF FFFF$_H$
WUFMSK1_I0 : FFFF FFFF$_H$ WUFMSK1_I1 : FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFMSKn31	WUFMSKn30	WUFMSKn29	WUFMSKn28	WUFMSKn27	WUFMSKn26	WUFMSKn25	WUFMSKn24	WUFMSKn23	WUFMSKn22	WUFMSKn21	WUFMSKn20	WUFMSKn19	WUFMSKn18	WUFMSKn17	WUFMSKn16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFMSKn15	WUFMSKn14	WUFMSKn13	WUFMSKn12	WUFMSKn11	WUFMSKn10	WUFMSKn9	WUFMSKn8	WUFMSKn7	WUFMSKn6	WUFMSKn5	WUFMSKn4	WUFMSKn3	WUFMSKn2	WUFMSKn1	WUFMSKn0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.49 WUFMSK0_Ax/WUFMSK1_Ax/WUFMSK0_Ix/WUFMSK1_Ix Register Contents (1/3)

Bit Position	Bit Name	Function
31	WUFMSKn31	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
30	WUFMSKn30	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
29	WUFMSKn29	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
28	WUFMSKn28	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
27	WUFMSKn27	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
26	WUFMSKn26	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
25	WUFMSKn25	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
24	WUFMSKn24	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
23	WUFMSKn23	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled

Table 15.49 WUFMSK0_Ax/WUFMSK1_Ax/WUFMSK0_Ix/WUFMSK1_Ix Register Contents (2/3)

Bit Position	Bit Name	Function
22	WUFMSKn22	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
21	WUFMSKn21	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
20	WUFMSKn20	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
19	WUFMSKn19	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
18	WUFMSKn18	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
17	WUFMSKn17	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
16	WUFMSKn16	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
15	WUFMSKn15	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
14	WUFMSKn14	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
13	WUFMSKn13	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
12	WUFMSKn12	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
11	WUFMSKn11	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
10	WUFMSKn10	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
9	WUFMSKn9	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
8	WUFMSKn8	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
7	WUFMSKn7	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
6	WUFMSKn6	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
5	WUFMSKn5	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled

Table 15.49 WUFMSK0_Ax/WUFMSK1_Ax/WUFMSK0_Ix/WUFMSK1_Ix Register Contents (3/3)

Bit Position	Bit Name	Function
4	WUFMSKn4	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
3	WUFMSKn3	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
2	WUFMSKn2	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
1	WUFMSKn1	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled
0	WUFMSKn0	Enables/disables a Wake-Up Factor. 0 : Wake-Up Factor is enabled 1 : Wake-Up Factor is disabled

NOTE

While the WUFMSKnx (x = 0 to 31) bit is 1, WUFnx (x = 0 to 31) of the Wake-Up Factor register is not set to 1 at the generation of a Wake-Up Factor.

WUFMSK0_Ax : WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2

WUFMSK1_Ax : WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2

WUFMSK0_Ix : WUFMSK0_I0/WUFMSK0_I1

WUFMSK1_Ix : WUFMSK1_I0/WUFMSK1_I1

Wake-Up Factors

As for the assignment of Wake-Up Factors to the Wake-Up Factor register bits, see **Section 15.1.1, Types of Chip Standby Mode**.

When writing to these registers, write the value “1” to the bits to which Wake-Up Factors are not assigned.

15.2.4.37 WUFC0_Ax/WUFC1_Ax/WUFC0_Ix/WUFC1_Ix — Wake-Up Factor Clear Registers

These registers clear the WUFnx (x = 0 to 31) bits in the Wake-Up Factor registers.

Access: This register is a write-only register that can be write in 32-bit units.

Address:
WUFC0_A0 : FF98 E008$_H$ WUFC0_A1 : FF98 E018$_H$
WUFC0_A2 : FF98 E028$_H$ WUFC1_A0 : FF98 E128$_H$
WUFC1_A1 : FF98 E138$_H$ WUFC1_A2 : FF98 E148$_H$
WUFC0_I0 : FF98 E208$_H$ WUFC0_I1 : FF98 E218$_H$
WUFC1_I0 : FF98 E328$_H$ WUFC1_I1 : FF98 E338$_H$

Value after reset:
WUFC0_A0 : 0000 0000$_H$ WUFC0_A1 : 0000 0000$_H$
WUFC0_A2 : 0000 0000$_H$ WUFC1_A0 : 0000 0000$_H$
WUFC1_A1 : 0000 0000$_H$ WUFC1_A2 : 0000 0000$_H$
WUFC0_I0 : 0000 0000$_H$ WUFC0_I1 : 0000 0000$_H$
WUFC1_I0 : 0000 0000$_H$ WUFC1_I1 : 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFCn 31	WUFCn 30	WUFCn 29	WUFCn 28	WUFCn 27	WUFCn 26	WUFCn 25	WUFCn 24	WUFCn 23	WUFCn 22	WUFCn 21	WUFCn 20	WUFCn 19	WUFCn 18	WUFCn 17	WUFCn 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFCn 15	WUFCn 14	WUFCn 13	WUFCn 12	WUFCn 11	WUFCn 10	WUFCn 9	WUFCn 8	WUFCn 7	WUFCn 6	WUFCn 5	WUFCn 4	WUFCn 3	WUFCn 2	WUFCn 1	WUFCn 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 15.50 WUFC0_Ax/WUFC1_Ax/WUFC0_Ix/WUFC1_Ix Register Contents (1/3)

Bit Position	Bit Name	Function
31	WUFCn31	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
30	WUFCn30	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
29	WUFCn29	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
28	WUFCn28	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
27	WUFCn27	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
26	WUFCn26	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
25	WUFCn25	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
24	WUFCn24	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
23	WUFCn23	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared

Table 15.50 WUFC0_Ax/WUFC1_Ax/WUFC0_Ix/WUFC1_Ix Register Contents (2/3)

Bit Position	Bit Name	Function
22	WUFCn22	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
21	WUFCn21	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
20	WUFCn20	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
19	WUFCn19	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
18	WUFCn18	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
17	WUFCn17	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
16	WUFCn16	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
15	WUFCn15	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
14	WUFCn14	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
13	WUFCn13	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
12	WUFCn12	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
11	WUFCn11	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
10	WUFCn10	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
9	WUFCn9	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
8	WUFCn8	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
7	WUFCn7	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
6	WUFCn6	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
5	WUFCn5	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
4	WUFCn4	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared

Table 15.50 WUFC0_Ax/WUFC1_Ax/WUFC0_Ix/WUFC1_Ix Register Contents (3/3)

Bit Position	Bit Name	Function
3	WUFCn3	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
2	WUFCn2	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
1	WUFCn1	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared
0	WUFCn0	Clears the Wake-Up Factor bit WUFy in the Wake-Up Factor registers. 0 : WUFy is not modified 1 : WUFy is cleared

NOTE

WUFC0_Ax : WUFC0_A0/WUFC0_A1/WUFC0_A2
WUFC1_Ax : WUFC1_A0/WUFC1_A1/WUFC1_A2
WUFC0_Ix : WUFC0_I0/WUFC0_I1
WUFC1_Ix : WUFC1_I0/WUFC1_I1

Wake-Up Factors

As for the assignment of Wake-Up Factors to the Wake-Up Factor register bits, see **Section 15.1.1, Types of Chip Standby Mode**.

When writing to these registers, write the value “0” to the bits to which Wake-Up Factors are not assigned.

15.2.4.38 WUFMON — Wake-Up Factor Monitor Register

This register is used to monitor Wake-Up Factor Register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF98 E1A0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MONWUF1_I1	MONWUF1_I0	—	—	—	—	—	MONWUF1_A2	MONWUF1_A1	MONWUF1_A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MONWUF0_I1	MONWUF0_I0	—	—	—	—	—	MONWUF0_A2	MONWUF0_A1	MONWUF0_A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.51 WUFMON Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	MONWUF1_I1	When one of WUF1_I1 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF1_I1 register is cleared, this bit becomes 0. 0 : WUF1_I1 register is not active. 1 : WUF1_I1 register is active.
24	MONWUF1_I0	When one of WUF1_I0 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF1_I0 register is cleared, this bit becomes 0. 0 : WUF1_I0 register is not active. 1 : WUF1_I0 register is active.
23 to 19	Reserved	When read, the value after reset is returned.
18	MONWUF1_A2	When one of WUF1_A2 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF1_A2 register is cleared, this bit becomes 0. 0 : WUF1_A2 register is not active. 1 : WUF1_A2 register is active.
17	MONWUF1_A1	When one of WUF1_A1 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF1_A1 register is cleared, this bit becomes 0. 0 : WUF1_A1 register is not active. 1 : WUF1_A1 register is active.
16	MONWUF1_A0	When one of WUF1_A0 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF1_A0 register is cleared, this bit becomes 0. 0 : WUF1_A0 register is not active. 1 : WUF1_A0 register is active.
15 to 10	Reserved	When read, the value after reset is returned.
9	MONWUF0_I1	When one of WUF0_I1 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF0_I1 register is cleared, this bit becomes 0. 0 : WUF0_I1 register is not active. 1 : WUF0_I1 register is active.
8	MONWUF0_I0	When one of WUF0_I0 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF0_I0 register is cleared, this bit becomes 0. 0 : WUF0_I0 register is not active. 1 : WUF0_I0 register is active.
7 to 3	Reserved	When read, the value after reset is returned.

Table 15.51 WUFMON Register Contents (2/2)

Bit Position	Bit Name	Function
2	MONWUF0_A2	When one of WUF0_A2 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF0_A2 register is cleared, this bit becomes 0. 0 : WUF0_A2 register is not active. 1 : WUF0_A2 register is active.
1	MONWUF0_A1	When one of WUF0_A1 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF0_A1 register is cleared, this bit becomes 0. 0 : WUF0_A1 register is not active. 1 : WUF0_A1 register is active.
0	MONWUF0_A0	When one of WUF0_A0 registers [31: 0] becomes 1, this bit is set to 1. Also, when the WUF0_A0 register is cleared, this bit becomes 0. 0 : WUF0_A0 register is not active. 1 : WUF0_A0 register is active.

15.3 Chip Standby Mode Transition

This section explains the mode transition procedures.

CAUTION

Transition to chip standby mode should be performed by CPU0(PE0). When CPU0(PE0) shifts to chip standby mode, CPU1(PE1)/CPU2(PE2)/CPU3(PE3) will also shift to chip standby mode.

15.3.1 STOP Mode

In STOP mode, most of the clock supplies to the Always-On area and the Isolated area are stopped. The clock stop mask bit in module standby register control clock supply to related clocks in chip standby mode. Stop all of the peripheral functions before transition to STOP mode if the clock supply to the function will be stopped in STOP mode.

The transition procedure (example) to STOP mode is shown below.

Preparation for standby

- Stop DMON.
- Stop CLMA (3 to 9).
- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (EICn.EIRFn = 0).
  - Mask the interrupts for non-Wake-Up Factors (EICn.EIMKn = 1).
  - Release the masks of the interrupts for Wake-Up Factors (EICn.EIMKn = 0).
- Set the wake-up related registers.
  - Clear the Wake-Up Factor flags (the WUFC0_A0/WUFC0_A1/WUFC0_A2/WUFC0_I0/WUFC0_I1 registers).
  - Mask the non-Wake-Up Factor (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1 registers).
  - Release the masks of the Wake-Up Factors (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1 registers).

NOTE

Disable Wake-Up Factor 1 (WUFMSK1_Ax = FFFF FFFF$_H$ and WUFMSK1_Ix = FFFF FFFF$_H$).

- Set the clock stop mask bit in module standby register to select the clocks to be stopped and the ones to continue operating (using the MSR_<name>.STPMSK_<name> bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register

to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register, the HSOSCSTPMSK bit in the HSOSCSTPM register and the PLLSTPMSK bit in the PLLSTPM register).

Set a clock according to the method described in **Section 13.6.6, CPU System Clock Setting in STOP/DeepSTOP Mode**.

Start of Chip standby

Set the STBC0STPTRG bit in the STBC0STPT register to 1 to transition to STOP mode.

End of Chip standby

When a Wake-Up Factor is generated, the microcontroller returns from STOP mode.

Wake-up handling

The generation of the Wake-Up Factors can be determined by the Wake-Up Factor flags (WUF0_A0, WUF0_A1, WUF0_I0).

When an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.

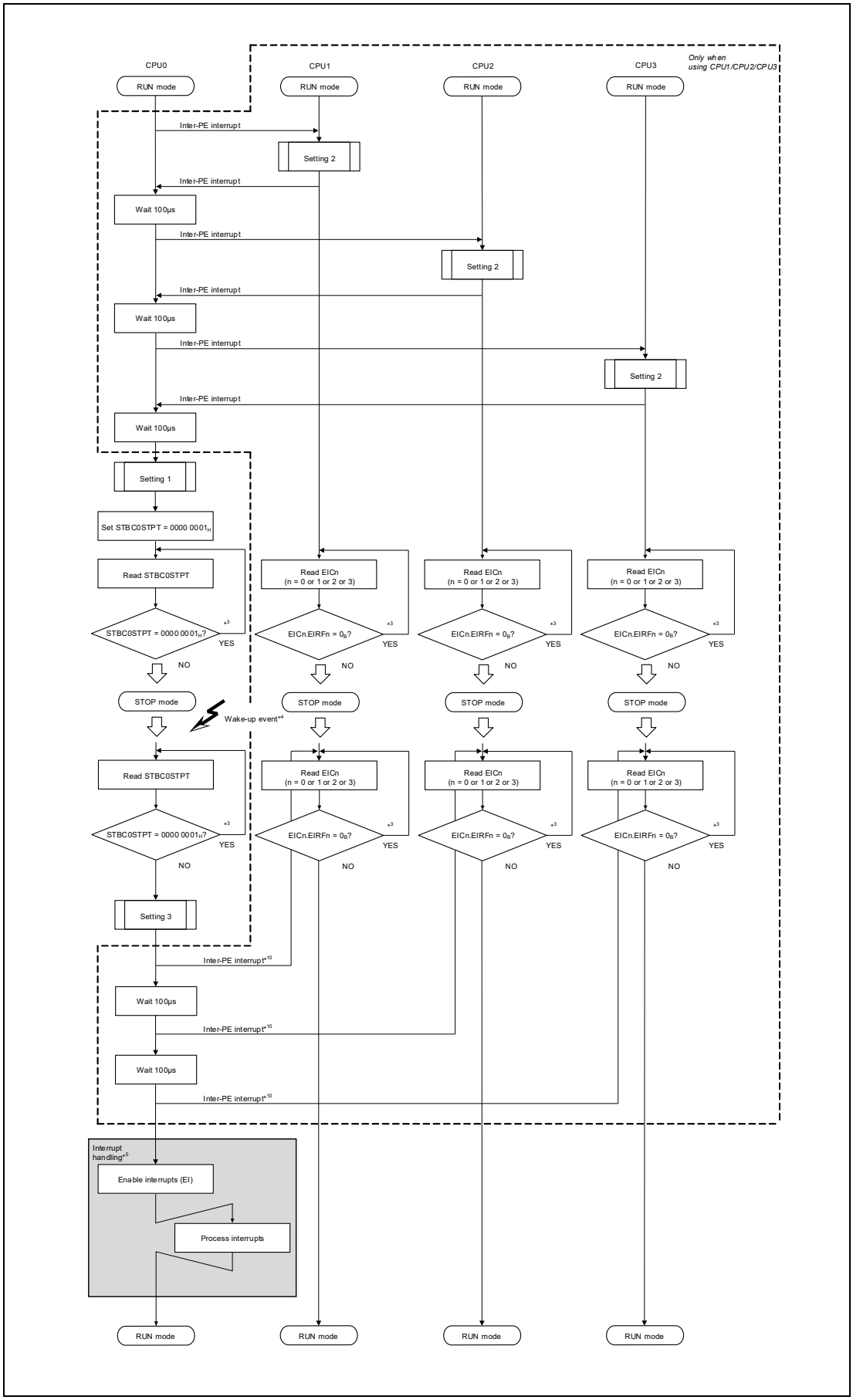
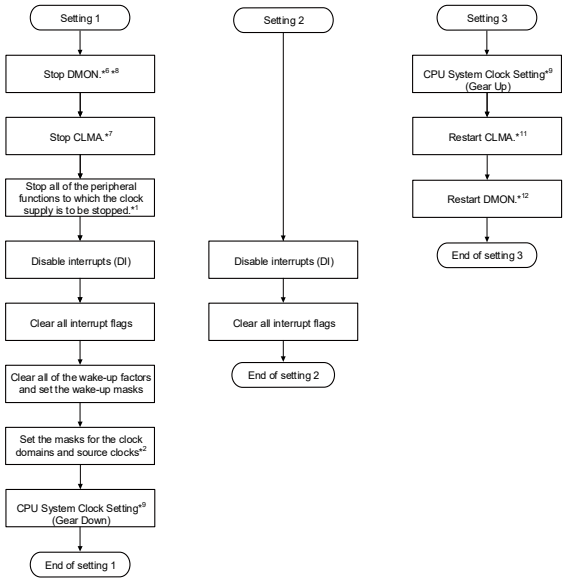


Figure 15.2 Example of STOP Mode Transition (1/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the STOP mode, the operation of the peripheral function may be incorrect. Before the transition to the STOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. Though the clock mask can be set before this flow is started, it must be set before $0000\ 0001_H$ is written to STBC0STPT.

Note 3. The clock supply to the CPU is stopped and the operation transitions to the STOP mode while checking that STBC0STPT = $0000\ 0001_H$ and EICn.EIRFn = 0_B. After returning from the STOP mode, the CPU starts the program from this loop.

Note 4. STBC0STPT is set to $0000\ 0000_H$ at the generation of a Wake-Up Factor. The generated Wake-Up Factor can be checked by the WUFxxx registers.

Note 5. This processing is optional. It is required when executing the interrupt handling after the wake-up.

Note 6. See **Section 11.4.7.4, Procedures to Reset DMON** for details.

Note 7. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.

Note 8. In case of stop Main OSC at chip standby mode, stop DMON.

Note 9. See **Section 13.6.6, CPU System Clock Setting in STOP/DeepSTOP Mode** for details.

Note 10. EICn.EIRFn can be set to 1_B by the Inter-PE interrupt. EICn.EIRFn should be cleared by software.

Note 11. See **Section 14.6.1, Procedures to Enable CLMAn** for details.

Note 12. See **Section 11.4.7.1, Basic Function** for details.

Figure 15.2 Example of STOP Mode Transition (2/2)

15.3.2 DeepSTOP Mode

In DeepSTOP mode, the clock supply to all areas and the power supply to the Isolated area are stopped. However, clock supply to peripheral functions in the Always-On area can be continued by setting the clock stop mask bit in module standby register.

The transition procedure (example) to DeepSTOP mode is shown below.

Preparation for standby

- Stop DMON.
- Stop CLMA (3 to 9).
- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (EICn.EIRFn = 0).(n = 0 to 695)
  - Mask the interrupts for non-Wake-Up Factors (EICn.EIMKn = 1).(n = 0 to 695)
  - Release the masks of the interrupts for Wake-Up Factors (EICn.EIMKn = 0).(n = 0 to 695)
- Set the wake-up related registers.
  - Clear the Wake-Up Factor flags (the WUFC0_A0/WUFC0_A1/WUFC0_A2/WUFC0_I0/WUFC0_I1/ WUFC1_A0/WUFC1_A1/WUFC1_A2/WUFC1_I0/WUFC1_I1 registers).
  - Mask the non-Wake-Up Factor (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1/ WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 registers).
  - Release the masks of the Wake-Up Factors (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1/ WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 registers).

CAUTION

When a Wake-Up Factor is assigned to both Wake-Up Factor 0 registers and Wake-Up Factor 1 registers, it can be used only in one of them.

- Set the clock stop mask bit in module standby register to select the clocks to be stopped and the ones to continue operating (using the MSR_<name>.STPMSK_<name> bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the HSOSCSTPMSK bit in the HSOSCSTPM register).

Set a clock according to the method described in **Section 13.6.6, CPU System Clock Setting in STOP/DeepSTOP Mode**.

Start of standby

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to transition to DeepSTOP mode.

End of standby

When a Wake-Up Factor is generated, the microcontroller returns from DeepSTOP mode.

Wake-up handling

- When returned from DeepSTOP mode due to Wake-Up Factor 0, the microcontroller starts the operation from the reset vector address.
  If one of the following interrupts has been generated before recovery from DeepSTOP mode to RUN mode, the microcontroller restarts operation from the exception handler address:
  - FENMI: FENMI handler address ($E0_H$)
  - FEINT: FEINT handler address ($F0_H$)

  Note that the General-purpose registers and local RAM are undefined value after return from DeepSTOP mode.
- When returned from DeepSTOP mode due to Wake-Up Factor 1, the device switches to CyclicRUN mode, the CPU0 (PE0) starts the operation from the reset address on the retention RAM. For details, refer to **Section 15.3.3, Cyclic RUN Mode**.
- The generation of the Wake-Up Factors can be determined by the Wake-Up Factor flags (WUF0_A0, WUF0_A1, WUF1_A0, WUF1_A1).
- The ports in the Isolated area maintain the I/O buffer hold state.
  Release the I/O buffer hold state by executing the following steps:
  1. Re-configure the peripheral functions and port functions.
  2. Set IOHOLDn.IOHOLD_xxx = 0 (n = 0, 2).
- To execute an interrupt of the Wake-Up Factor after the wake-up, evaluate the information of Wake-Up Factor flag by software and set the interrupt request flag in the interrupt control register. Then, when an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.

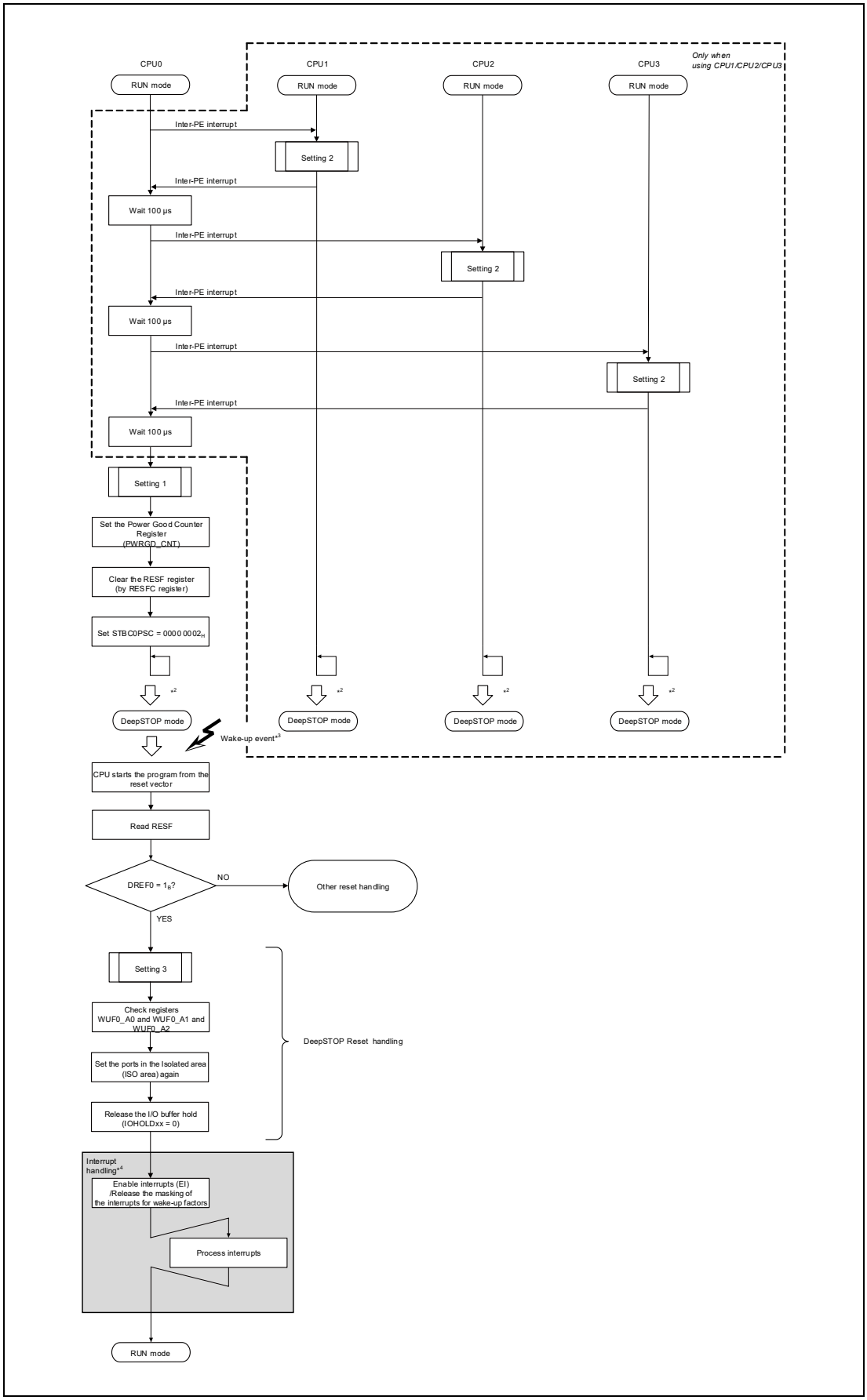
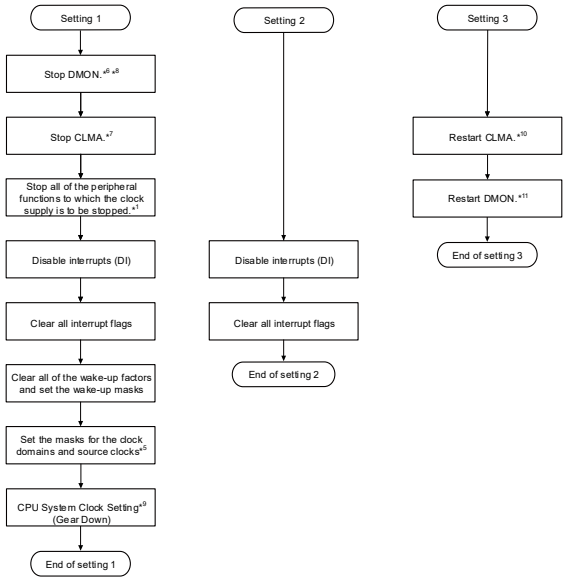


Figure 15.3 Example of DeepSTOP Mode Transition (1/2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DeepSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DeepSTOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DeepSTOP mode by the unconditional loop.

Note 3. The CPU starts the program from the reset vector after the generation of a Wake-Up Factor. The return from the DeepSTOP mode by a reset can be checked by the RESF register. In addition, the generated Wake-Up Factor can be checked by the WUF0_A0 and WUF0_A1 and WUF0_A2 registers.

Note 4. This processing is optional. It is required when executing the interrupt handling after the wake-up. To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. After that, the interrupt should be enabled.

Note 5. Though the clock mask can be set before this flow is started, it must be set before 0000 0002_H is written to STBC0PSC.

Note 6. See **Section 11.4.7.4, Procedures to Reset DMON** for details.

Note 7. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.

Note 8. In case of stop Main OSC at chip standby mode, stop DMON.

Note 9. See **Section 13.6.6, CPU System Clock Setting in STOP/DeepSTOP Mode** for details.

Note 10. See **Section 14.6.1, Procedures to Enable CLMAn** for details.

Note 11. See **Section 11.4.7.1, Basic Function** for details.

Figure 15.3 Example of DeepSTOP Mode Transition (2/2)

15.3.3 Cyclic RUN Mode

In Cyclic RUN mode, CPU0(PE0), the peripheral functions in the Always-On area (AWO area), RLIN3, and MSPI are available. In this mode, PLL and Flash Memory are not available.The CPU1(PE1)/CPU2(PE2)/CPU3(PE3) is also not available. The transition procedure (example) to Cyclic RUN mode is shown below.

Preparation for Cyclic RUN

Stop DMON and CLMA (3 to 9). Allocate the program for Cyclic RUN to the retention RAM. The reset vector base address (RBASE) in Cyclic RUN operation is set to the first address of the retention RAM (FE80 0000_H). Note that neither the code flash memory nor the data flash memory is available in Cyclic RUN mode.

The instruction to transition to DeepSTOP mode should be arranged in the interrupt exception handler or a polling routine of interrupt request which is used as the source of returning to the RUN mode.

CAUTION

Do not change the PSW.EBV bit from its value after reset in Cyclic RUN mode (Do not set the PSW.EBV bit to 1 in Cyclic RUN mode).

- Set the wake-up related registers.
  - Clear the Wake-Up Factor flags (the WUFC1_A0/WUFC1_A1/WUFC1_A2/WUFC1_I0/WUFC1_I1 register).
  - Mask the non-Wake-Up Factor (the WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 register).
  - Release the masks of the Wake-Up Factors (the WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 register).
- Transition to DeepSTOP mode. For details on the transition to DeepSTOP mode, see **Section 15.3.2, DeepSTOP Mode.**

Start of Cyclic RUN

The operation transitions to Cyclic RUN mode from DeepSTOP mode at the generation of Wake-Up Factor 1.

The operation transitions to Cyclic RUN mode from Cyclic STOP mode at the generation of Wake-Up Factors 0 and 1.

The microcontroller starts operation from the reset vector address of Cyclic RUN mode (the first address of the retention RAM (FE80 0000_H). If one of the following interrupts has been generated during recovery from DeepSTOP mode to Cyclic RUN mode, the microcontroller restart operation from the exception handler address:

- FENMI: FENMI handler address in Cyclic RUN mode (FE80 0000_H + $E0_H$).
- FEINT: FEINT handler address in Cyclic RUN mode (FE80 0000_H + $F0_H$).

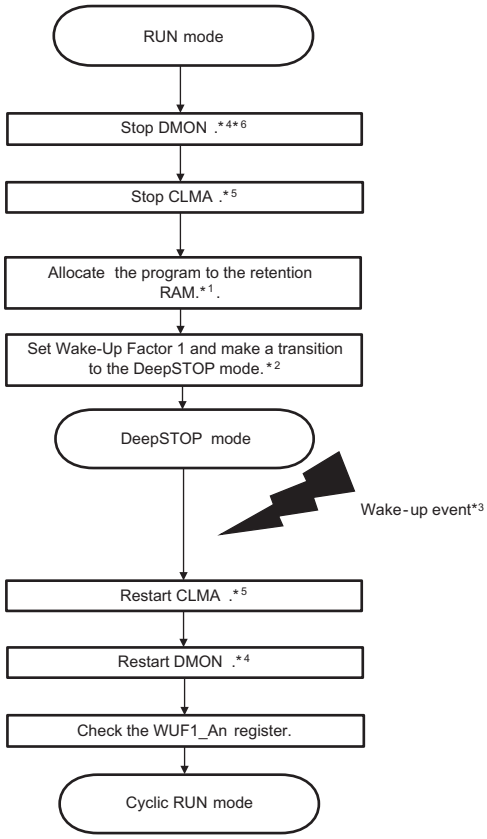
Note that the General-purpose registers and local RAM are undefined value after the transition to Cyclic RUN mode from DeepSTOP mode.

End of Cyclic RUN

The Cyclic RUN mode ends at the transition to the Cyclic STOP mode by setting the STBC0STPT.STBC0STPTRG bit to 1, or at the transition to the DeepSTOP mode by setting the STBC0PSC.STBC0DISTRG bit to 1.

Wake-up handling

The generation of the Wake-Up Factors can be determined by the Wake-Up Factor flags (WUF1_A0/ WUF1_A1/WUF1_A2/WUF1_I0/WUF1_I1) .



Note 1. When the mode transitions from the Cyclic RUN mode to the RUN mode by a Wake-Up Factor, the transition to the DeepSTOP mode should be made in the processing of the interrupt vector for the Wake-Up Factor. In that case, allocate the interrupt processing program to the retention RAM as well.

Note 2. Before the transition to the DeepSTOP mode, clear the flag for Wake-Up Factor 1 in the WUFC1_Ax register and set Wake-Up Factor 1 that is to be used in the WUFMSK1_A0/ WUFMSK1_I0 register. Other parts of the procedure for transition to DeepSTOP mode are the same as the normal procedure.

Note 3. The CPU starts the program from the top address (FE80 0000_H) of the retention RAM after the generation of a Wake-Up Factor. The generated Wake-Up Factor can be checked by the WUF1_A0/ WUF1_A1/WUF1_A2/WUF1_I0/WUF1_I1 register.

Note 4. See **Section 11.4.7.4, Procedures to Reset DMON** for details.

Note 5. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.

Note 6. In case of stop Main OSC at chip standby mode, DMON stops.

Figure 15.4 Example of Cyclic RUN Mode Transition

15.3.4 Cyclic STOP Mode

In Cyclic STOP mode, the peripheral functions in the Always-On area (AWO area), and RLIN3 are available. The transition procedure (example) to Cyclic STOP mode is shown below.

Preparation for Cyclic STOP

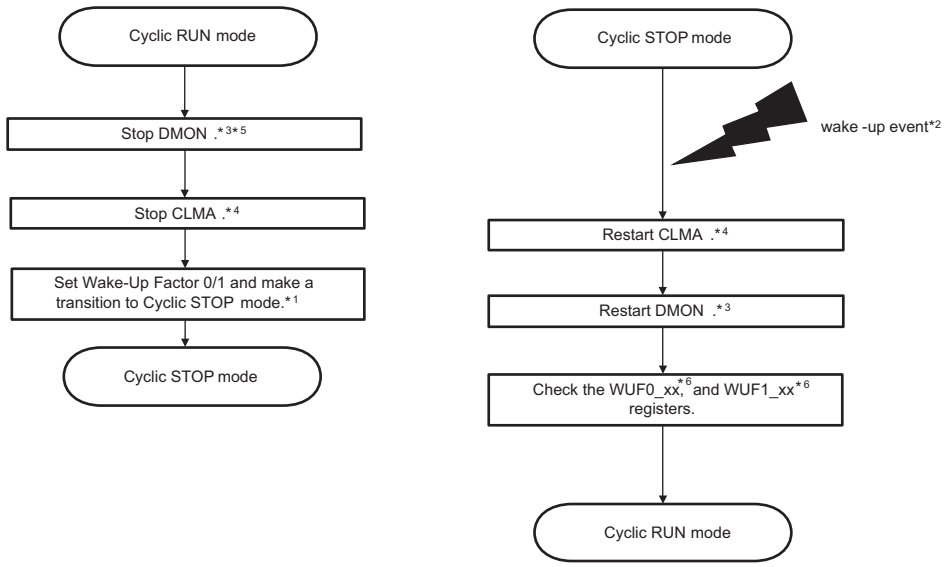
- Stop DMON.
- Stop CLMA (3 to 9).
- Transition to Cyclic RUN mode.
- Set the wake-up related registers.
  - Clear the Wake-Up Factor flags
    (the WUFC0_A0/WUFC0_A1/WUFC0_A2/WUFC0_I0/WUFC0_I1/
    WUFC1_A0/WUFC1_A1/WUFC1_A2/WUFC1_I0/WUFC1_I1 register).
  - Mask the non-Wake-Up Factor
    (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1/
    WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 register).
  - Release the masks of the Wake-Up Factors
    (the WUFMSK0_A0/WUFMSK0_A1/WUFMSK0_A2/WUFMSK0_I0/WUFMSK0_I1/
    WUFMSK1_A0/WUFMSK1_A1/WUFMSK1_A2/WUFMSK1_I0/WUFMSK1_I1 register).

Start of Cyclic STOP

Set the STBC0STPT.STBC0STPTRG bit to 1 to transition to Cyclic STOP mode.

End of Cyclic STOP

The operation transitions to Cyclic RUN mode at the generation of Wake-Up Factor 0 or 1.



Note 1. Set Wake-Up Factors 0 and 1 to be used for transitioning to RUN mode and Cyclic RUN mode, respectively. When the mode transitions to RUN mode by Wake-Up Factor 0, the transition processing to DeepSTOP mode should be added in Cyclic RUN mode.

Note 2. When a Wake-Up Factor is generated in Cyclic STOP mode, the mode transitions to Cyclic RUN mode and the operation starts immediately after the processing shifted to Cyclic STOP mode. The generated Wake-Up Factors can be checked by the WUF0_A0/WUF0_A1/WUF0_A2/WUF0_I0/WUF0_I1/ WUF1_A0/WUF1_A1/WUF1_A2/WUF1_I0/WUF1_I1 registers.

Note 3. See **Section 11.4.7.4, Procedures to Reset DMON** for details.

Note 4. See **Section 14.6.2, Procedures to Reset by CLMATEST.RESCLM** for details.

Note 5. In case of stop Main OSC at chip standby mode, stop DMON.

Note 6. "xx" is A0 , A1, A2 , I0 , I1.

Figure 15.5 Example of Cyclic STOP Mode Transition

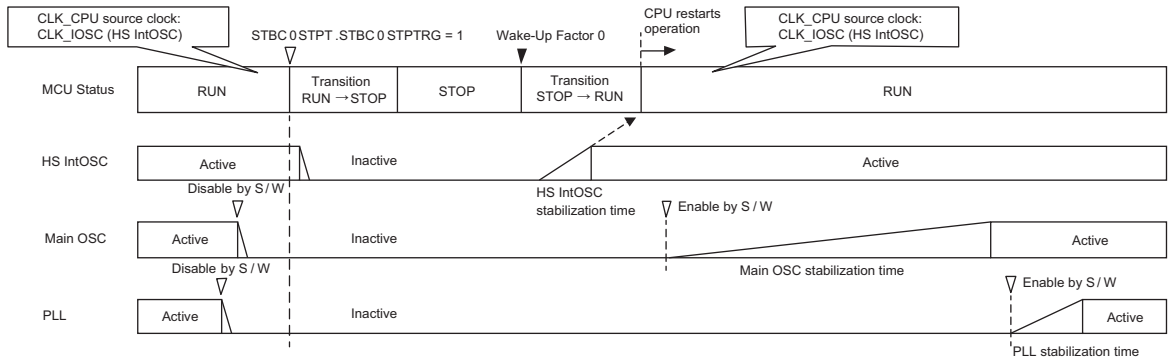
15.4 Clock Oscillator Behavior During Chip Standby Mode Transition

The following figures explain clock oscillator behavior during chip standby mode transition. The clock oscillators restart operation automatically if they are used before entering standby.

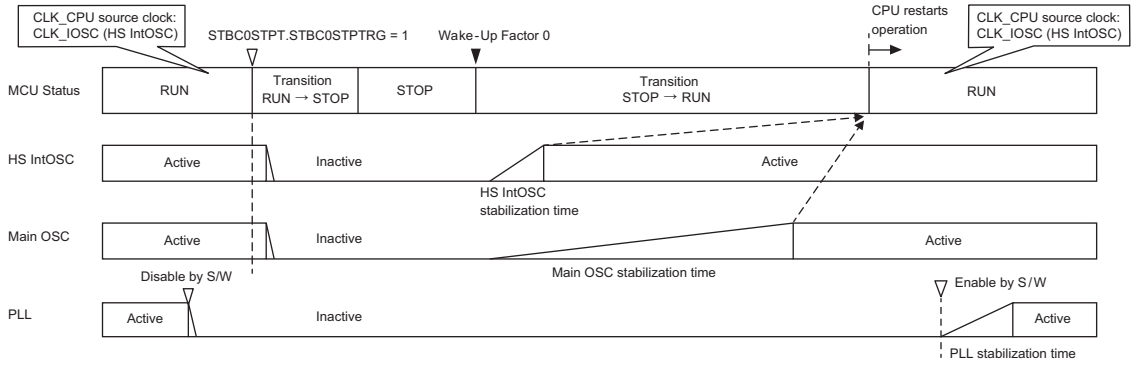
- If Main OSC and PLL are enabled before entering STOP mode, they restart oscillation automatically during wake-up from STOP mode, and CPU restarts operation after oscillations of these clock sources become stable.
- If Main OSC is enabled before entering DeepSTOP mode, it restarts oscillation automatically during wake-up from DeepSTOP mode, and CPU restarts operation after oscillation of Main OSC becomes stable. PLL is not restarted automatically even if it is enabled before entering DeepSTOP mode.
- If Main OSC is enabled before entering Cyclic STOP mode, it restarts oscillation automatically during wake-up from Cyclic STOP mode. CPU restarts operation after oscillation of Main OSC becomes stable.

Note that behavior of HS IntOSC, Main OSC and PLL in the following figures is in the case they stop oscillation during chip standby mode. HS IntOSC, Main OSC and PLL continue oscillation during chip standby mode if their stop mask register is set to 1 (HSOSCSTPMSK bit of HSOSCSTPM register, MOSCSTPM bit of MOSCSTPMSK register and PLLSTPMSK bit of PLLSTPM register respectively).

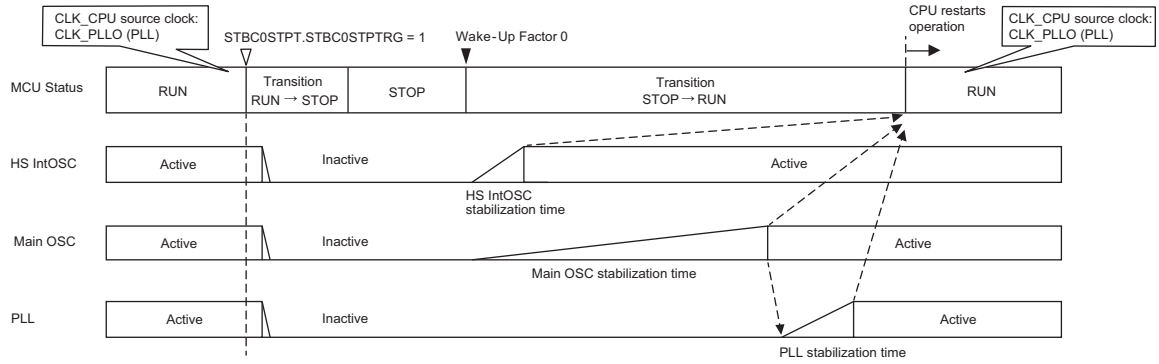
For details on the stop mask register, see **Section 13, Clock Controller**.



(a) PLL and Main OSC are disabled before entering STOP mode.
Restart Main OSC and PLL by software after wake-up.
CLK_CPU source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

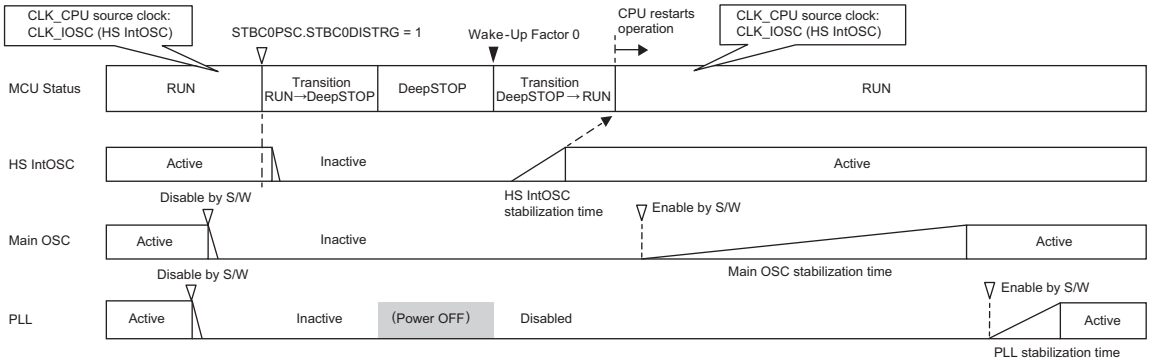


(b) PLL is disabled, and Main OSC is enabled before entering STOP mode.
Main OSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CLK_CPU source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

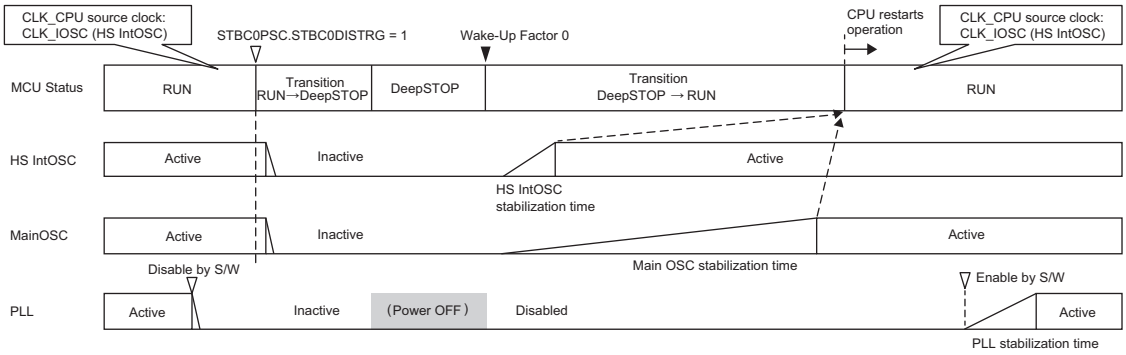


(c) PLL and Main OSC are enabled before entering STOP mode.
PLL and Main OSC are restarted automatically during transition to RUN mode, and become stable CPU restarts operation.
CLK_CPU source clock after wake-up from STOP mode is the same source clock before entering STOP mode.

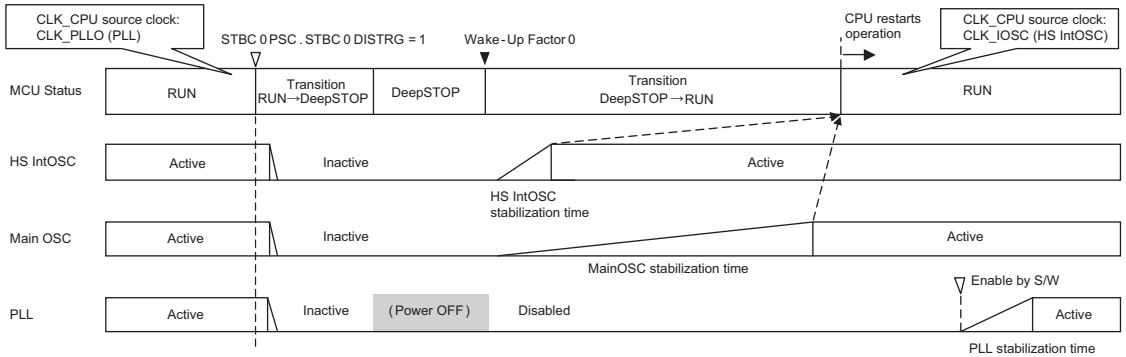
Figure 15.6 Clock oscillators behavior in Chip Standby Mode transition (RUN → STOP → RUN)



(a) PLL and Main OSC are disabled before entering DeepSTOP mode.
Restart Main OSC and PLL by software after wake-up.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

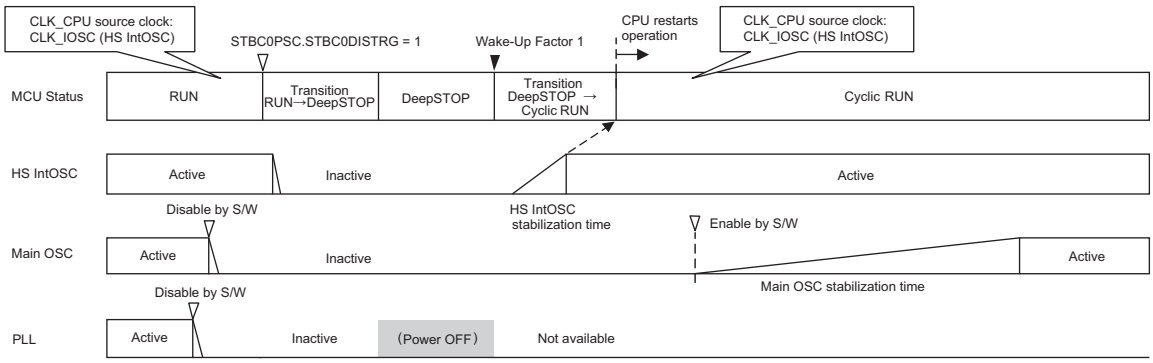


(b) PLL is disabled, and Main OSC is enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

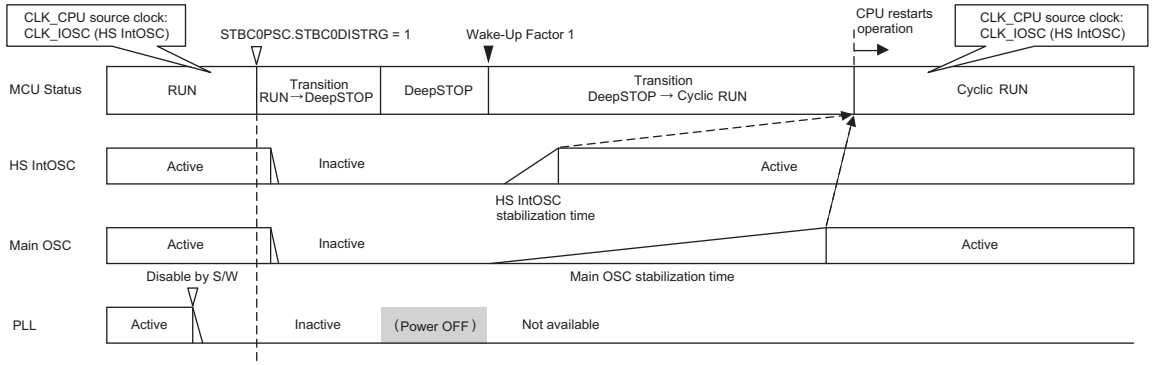


(c) PLL and Main OSC are enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to RUN mode, and becomes stable before CPU restarts operation.
Restart PLL by software after wake-up.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

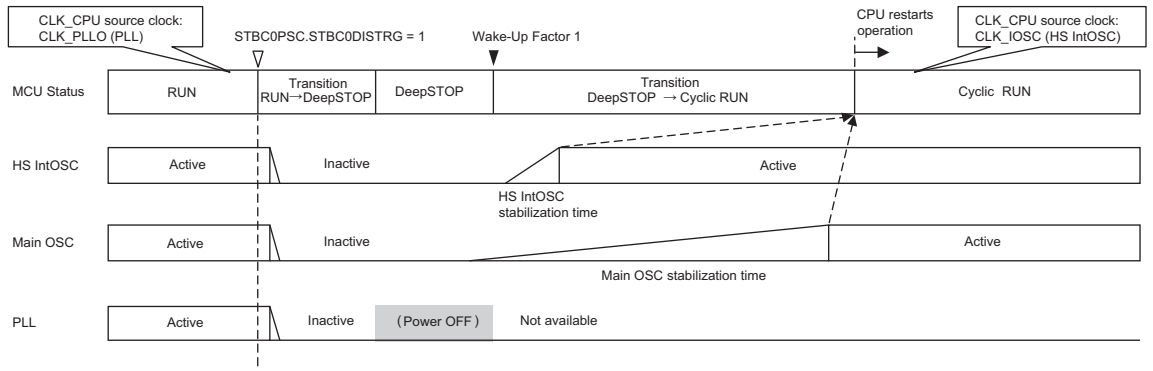
Figure 15.7 Clock oscillators behavior in Chip Standby Mode transition (RUN → DeepSTOP → RUN)



(a) PLL and Main OSC are disabled before entering DeepSTOP mode.
Restart Main OSC by software after wake-up. PLL is not available in Cyclic RUN mode.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

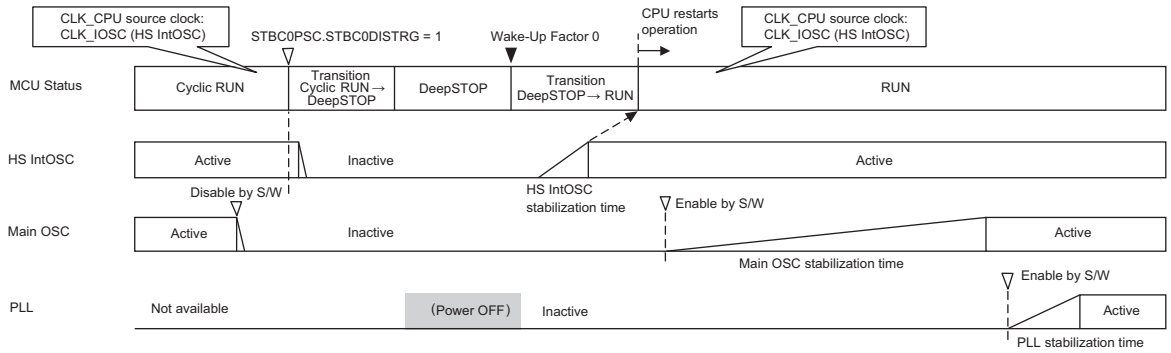


(b) PLL is disabled, and Main OSC is enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation. PLL is not available in Cyclic RUN mode.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

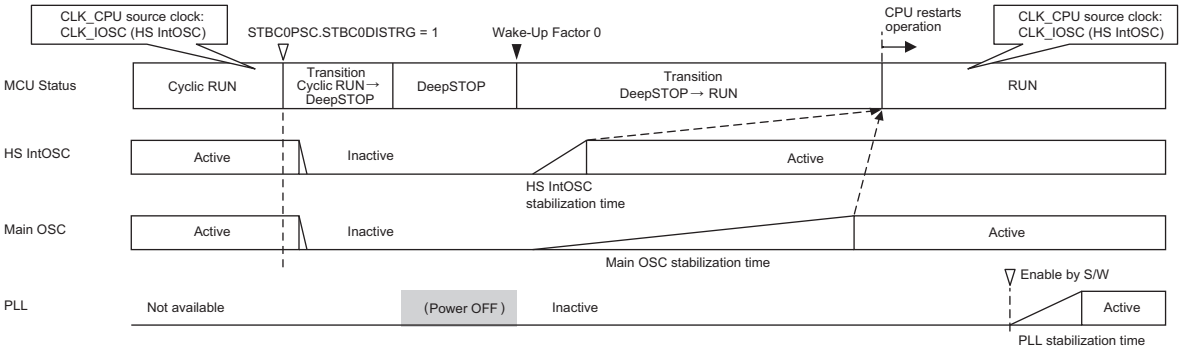


(c) PLL and Main OSC are enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation. PLL is not available in Cyclic RUN mode.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

Figure 15.8 Clock oscillators behavior in Chip Standby Mode transition (RUN → DeepSTOP → Cyclic RUN)

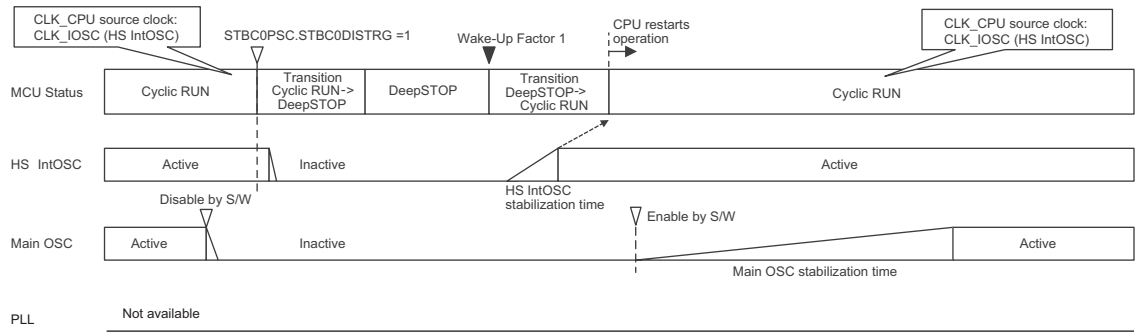


(a) Main OSC is disabled before entering DeepSTOP mode.
Restart Main OSC and PLL by software after wake-up.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

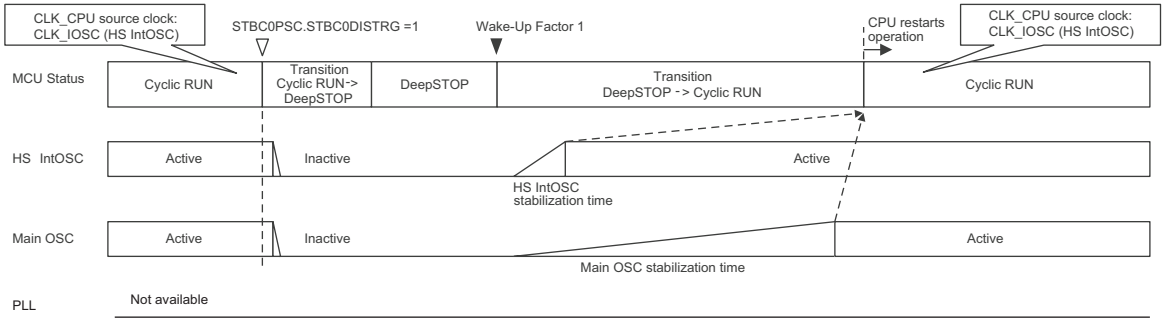


(b) Main OSC is enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to RUN mode, and becomes stable before CPU starts operation.
Restart PLL by software after wake-up.
CLK_CPU source clock after wake-up from DeepSTOP mode is CLK_IOSC (HS IntOSC).

Figure 15.9 Clock oscillators behavior in Chip Standby Mode transition (Cyclic RUN → DeepSTOP → RUN)

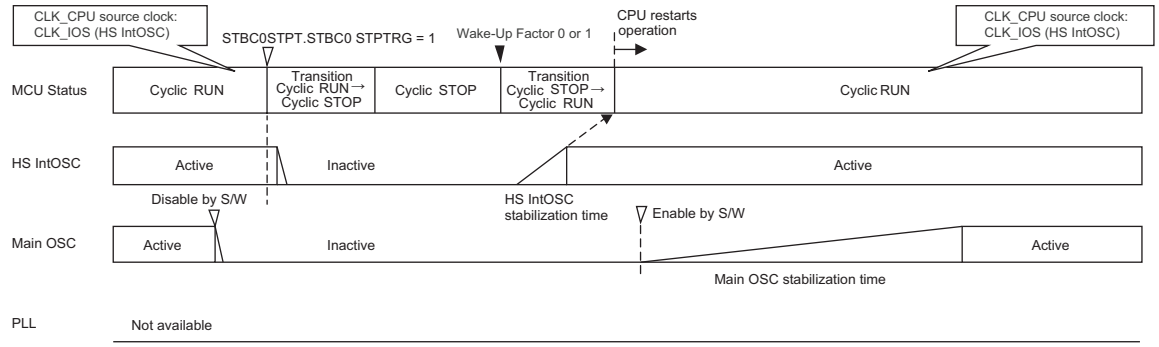


(a) Main OSC is disabled before entering DeepSTOP mode.
Restart Main OSC by software after wake-up. CLK_CPU source clock after wake-up from DeepSTOP mode is the same source clock before entering DeepSTOP mode.

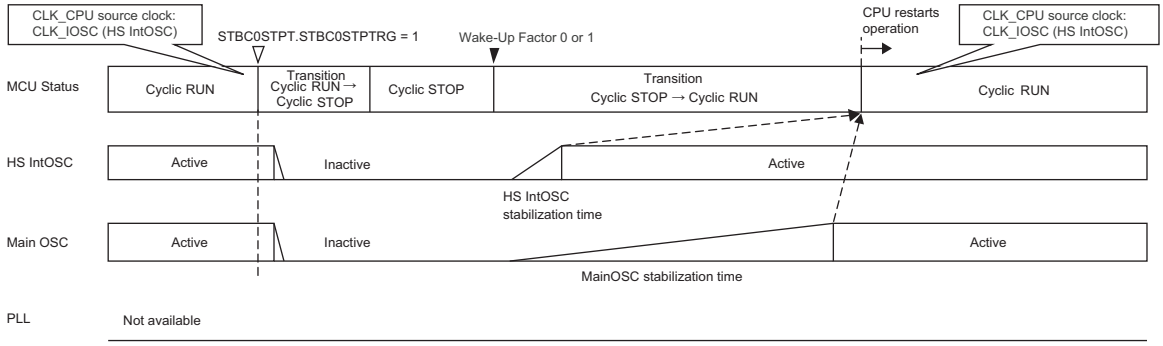


(b) Main OSC is enabled before entering DeepSTOP mode.
Main OSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation. CLK_CPU source clock after wake-up from DeepSTOP mode is the same source clock before entering DeepSTOP mode.

Figure 15.10 Clock oscillators behavior in Chip Standby Mode transition (Cyclic RUN → DeepSTOP → Cyclic RUN)



(a) Main OSC is disabled before entering Cyclic STOP mode.
Restart Main OSC by software after wake-up.
CLK_CPU source clock after wake-up from Cyclic STOP mode is the same source clock before entering Cyclic STOP mode.



(b) Main OSC is enabled before entering Cyclic STOP mode.
Main OSC is restarted automatically during transition to Cyclic RUN mode, and becomes stable before CPU starts operation.
CLK_CPU source clock after wake-up from Cyclic STOP mode is the same source clock before entering Cyclic STOP mode.

Figure 15.11 Clock oscillators behavior in Chip Standby Mode transition (Cyclic RUN → Cyclic STOP → Cyclic RUN)

15.5 Cautions when Using Chip Standby Modes

15.5.1 Cautions Concerning Transitioning to DeepSTOP Mode When Using a Debugger

When using a debugger, executing a program that causes the mode to transition to DeepSTOP mode immediately after the program is started may cause improper communication between the OCD emulator and microcontroller because the microcontroller will enter DeepSTOP mode before the preparations for communication between the OCD emulator and microcontroller are completed.

The communication preparation period depends on the OCD emulator's host PC environment and the operating frequency of the microcontroller, so when performing debugging that causes the program to enter DeepSTOP mode immediately after the program starts, insert a wait between reset release and the DeepSTOP execution instruction so that the debugger starts normally.

In DeepSTOP mode, the debugging controller stops. For return from DeepSTOP mode by the debugger, see **Section 15.1.1.2, On-Chip Debug Wake-Up**.

Section 16 Low-Power Sampler (LPS)

This section contains a generic description of the low-power sampler (LPS).

The first part of this section describes the features specific RH850/U2A such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the LPS.

16.1 Features LPS for RH850/U2A

16.1.1 Number of Units

This microcontroller has the following number of LPS units.

Table 16.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	0	1 (n = 0)
Name	LPSn									

Table 16.2 Unit Configurations and Channels

Unit Name LPSn	Number of Channels per Unit	Function	Channel Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
LPS0	1	Digital port input m for port polling	DPINm	24 ch	24 ch	15 ch	5 ch	15 ch	5 ch	5 ch	5 ch	0 ch	5 ch
		Analog input m for A/D converter	ADCJ2Im	20 ch	20 ch	20 ch	5 ch	20 ch	5 ch	5 ch	5 ch	0 ch	0 ch

Table 16.3 Indices

Index	Description
n	Throughout this section, the individual LPS units are identified by the index “n” (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index “m” (m = 0 to 23) and the number of analog input channels for A/D converter is indicated by the index “m” (m = 0 to 19)
k	The external multiplexer select output signal is indicated by the index “k”.
x	LPS sequence start trigger input signal is indicated by the index “x”.
y	Throughout this section, the individual TAUJ units are identified by the index “y”.

NOTE

Descriptions of functions and registers in this section are based on the maximum configurations. Adjust the indices in the text to the proper value for each product. When writing a value to a register that will result in writing to bits outside the range of the index for the product you are using, write the value after reset to these bits.

The following table shows the values indicated by the indices of each product.

Table 16.4 Indices of Products

Index	Indices of Each Product									
	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
m	0 to 23*1 0 to 19*2	0 to 23*1 0 to 19*2	0 to 14*1 0 to 19*2	0 to 2, 11 to 12*1 0 to 4*2	0 to 14*1 0 to 19*2	0 to 2, 11 to 12*1 0 to 4*2	0 to 2, 11 to 12*1 0 to 4*2	0 to 2, 11 to 12*1 0 to 4*2	—	0 to 2, 11 to 12*1 —*2
k	0 to 2	0 to 2	0 to 2	0 to 2	0 to 2	0 to 2	0 to 2	0 to 2	—	0 to 2
x	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3	0 to 3	—	0 to 3
y	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3	—	2, 3

Note 1. Digital port input m for port polling

Note 2. Analog input m for A/D converter

16.1.2 Register Base Address

The LPS base address is shown in the following table.

LPS register addresses are given as an offset from the base address.

Table 16.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<LPS0_base>	FF9A 4000$_H$	P-Bus Group 2L

16.1.3 Clock Supply

The LPS clock supply is shown in the following table.

If the operation request signal for the low-power sampler (LPS) is at the active level, the clock for CLKA_ADC for which the HS IntOSC is selected also operates.

Table 16.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
LPSn	Register access clock	CLK_LSB, CLKA_LPS
	Operating clock	CLKA_LPS

16.1.4 Interrupt Requests and Error Notifications

The LPS interrupt requests are listed in the following table.

Table 16.7 Interrupt and DMA/DTS Requests

Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
LPS0				
INTCWEND	Port polling end interrupt (LPS)	693	—	—
INTDPE	Digital port error interrupt (LPS)	694	—	—
INTADCJ2I0*1	ADCJ2 SG0 end interrupt	239	Group0-14	Group0-12
INTADCJ2I1*1	ADCJ2 SG1 end interrupt	240	Group0-15	Group0-13
INTADCJ2I2*1	ADCJ2 SG2 end interrupt	241	Group0-16	Group0-14
INTADCJ2I3*1	ADCJ2 SG3 end interrupt	242	Group0-17	Group0-15
INTADCJ2I4*1	ADCJ2 SG4 end interrupt	243	Group0-18	Group0-16

Note 1. These signals are output from ADCJ2.

This module has no error notifications.

16.1.5 Reset Sources

The LPS reset sources are shown in the following table. The LPS is initialized by the reset source.

Table 16.8 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
LPS0	All registers	√	√	√	√	—	—	—

16.1.6 External Input/Output Signals

External input/output signals of LPS are listed below.

Table 16.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
LPS0		
DPO	Port output signal for digital input	DPO
DPSELk	External multiplexer select output signal for digital port	SELDPk
DPINm	Digital port input signal	DPINm
APO	Port output signal for analog input	APO
ADCJ2SELk*1	External analog multiplexer (MPX) output pin	ADCJ2SELk
ADCJ2Im*1	ADCJ input channel signal	ADCJ2Im

Note 1. These signals are input/output of ADCJ2. For details, see **Section 43.1.6, External Input/Output Signals**.

16.1.7 Internal Input/Output Signals

Internal input/output signals for connecting the LPS and STBC or the LPS and TAUJ are listed below.

Table 16.10 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
WUTRG0	LPS wake-up source trigger 0 output signal	STBC
WUTRG1	LPS wake-up source trigger 1 output signal	STBC
INTTAUJylx	LPS sequence start trigger x input signal	TAUJy

16.2 Overview

16.2.1 Functional Overview

To monitor the external input without consuming CPU resources, the low-power sampler (LPS) can check the digital input ports and analog input ports without using the CPU. The figure below shows a connection example between the main components of the LPS and the external circuit.

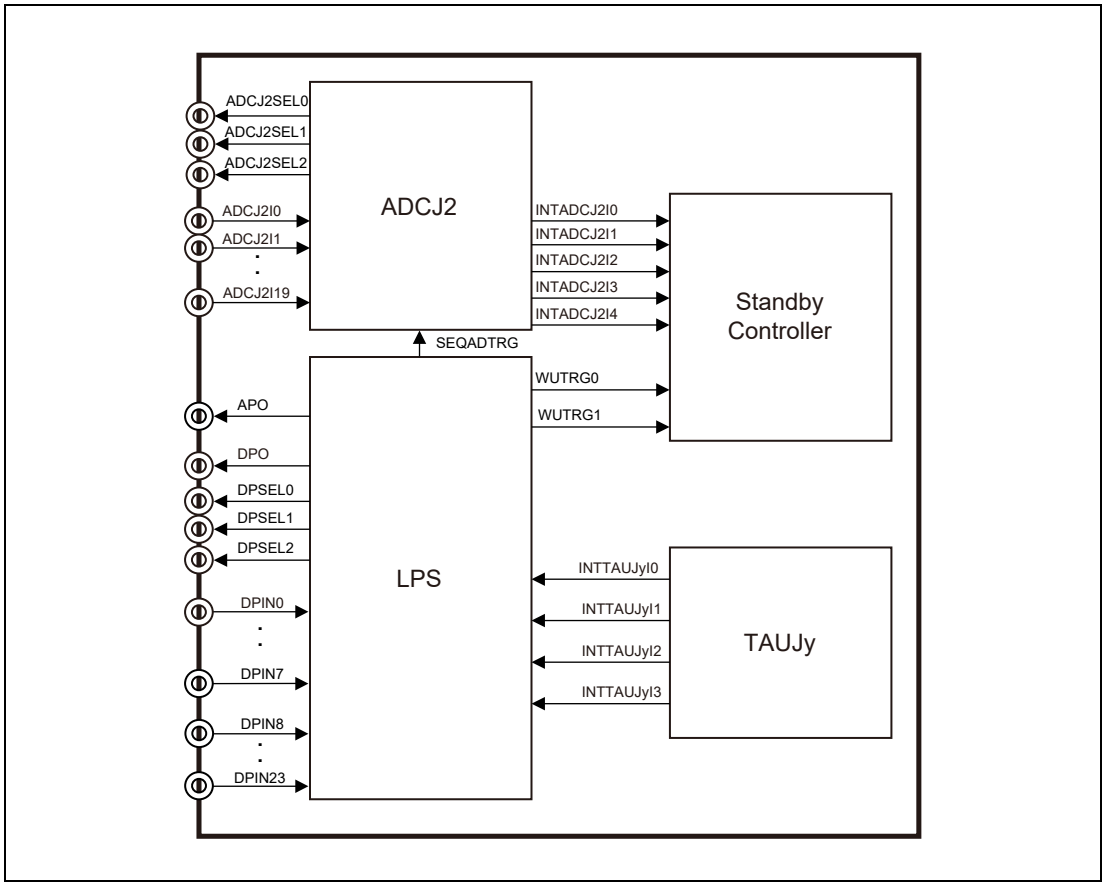


Figure 16.1 Block Diagram of the LPS

16.3 Registers

16.3.1 List of Registers

LPS registers are listed in the following table.

For details about <LPS0_base>, see **Section 16.1.2, Register Base Address**.

Table 16.11 List of Registers

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
LPS0	LPS control register	SCTLR	<LPS0_base > + 00_H	PBG20 #7	
	Event flag register	EVFR	<LPS0_base > + 04_H	PBG20 #7	
	DPIN select register 0	DPSELR0	<LPS0_base > + 08_H	PBG20 #7	
	DPIN select register M	DPSELRM	<LPS0_base > + $0C_H$	PBG20 #7	
	DPIN select register H	DPSELRH	<LPS0_base > + 10_H	PBG20 #7	
	DPIN data set register 0	DPDSR0	<LPS0_base > + 14_H	PBG20 #7	
	DPIN data set register M	DPDSRM	<LPS0_base > + 18_H	PBG20 #7	
	DPIN data set register H	DPDSRH	<LPS0_base > + $1C_H$	PBG20 #7	
	DPIN data input monitor register 0	DPDIMR0	<LPS0_base > + 20_H	PBG20 #7	
	DPIN data input monitor register 1	DPDIMR1	<LPS0_base > + 24_H	PBG20 #7	
	DPIN data input monitor register 2	DPDIMR2	<LPS0_base > + 28_H	PBG20 #7	
	DPIN data input monitor register 3	DPDIMR3	<LPS0_base > + $2C_H$	PBG20 #7	
	DPIN data input monitor register 4	DPDIMR4	<LPS0_base > + 30_H	PBG20 #7	
	DPIN data input monitor register 5	DPDIMR5	<LPS0_base > + 34_H	PBG20 #7	
	DPIN data input monitor register 6	DPDIMR6	<LPS0_base > + 38_H	PBG20 #7	
	DPIN data input monitor register 7	DPDIMR7	<LPS0_base > + $3C_H$	PBG20 #7	
	Count value register	CNTVAL	<LPS0_base > + 40_H	PBG20 #7	
	LPS operation status register	SOSTR	<LPS0_base > + 44_H	PBG20 #7	

16.3.2 SCTLR — LPS Control Register

This register is used to configure the LPS.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 00_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TJIS2	NUMDP 2	NUMDP 1	NUMDP 0	TJIS1	TJIS0	ADEN	DPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.12 SCTLR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	NUMDP[2:0]	These bits specify the number of times the port is read in digital input mode. If two or more times are specified, the external multiplexer is controlled by the DPSEL[2:0] pins. The bits for which comparison is enabled in the DPSELR0, DPSELRM, and DPSELRH registers are compared regardless of the repeat number setting, and WUTRG will be generated according to the results. **NUMDP[2:0] / Number of Times the Port Is Read** 000_B One time 001_B Two times 010_B Three times 011_B Four times 100_B Five times 101_B Six times 110_B Seven times 111_B Eight times These bits should be set before the TAUJy and sequence operations are started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0). (When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)
7, 3, 2	TJIS[2:0]	Sequence Start Trigger Select 000: INTTAUJ2I0 001: INTTAUJ2I1 010: INTTAUJ2I2 011: INTTAUJ2I3 100: INTTAUJ3I0 101: INTTAUJ3I1 110: INTTAUJ3I2 111: INTTAUJ3I3 These bits should be set before the sequence operation is started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0). (When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)

Table 16.12 SCTLR Register Contents (2/2)

Bit Position	Bit Name	Function
1	ADEN	0: Analog input mode is disabled 1: Analog input mode is enabled
0	DPEN	0: Digital input mode is disabled 1: Digital input mode is enabled

16.3.3 EVFR — Event Flag Register

This register indicates the result of comparing the data sequentially captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DINEVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.13 EVFR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DINEVF	This bit indicates the result of comparing the data captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers. Read: 0: The result of comparison is a match. 1: The result of comparison is a mismatch. Write: 0: Clear the bit. 1: Prohibited. This bit is set to 1 when a mismatch is detected even in one bit. Only 0 can be written to clear this bit.

16.3.4 DPSELR0 — DPIN Select Register 0

This register specifies the compare target bits in the DPDSR0 and DPDIMR0 registers.

Write to the DPSELR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 08_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0EN_23	D0EN_22	D0EN_21	D0EN_20	D0EN_19	D0EN_18	D0EN_17	D0EN_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0EN_15	D0EN_14	D0EN_13	D0EN_12	D0EN_11	D0EN_10	D0EN_9	D0EN_8	D0EN_7	D0EN_6	D0EN_5	D0EN_4	D0EN_3	D0EN_2	D0EN_1	D0EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.14 DPSELR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0EN_n (n = 23 to 0)	These bits enable or disable comparing each bit of the first data captured at the digital input pins and stored in the DPDIMR0 register with the comparison target data in the DPDSR0 register. 0: Disables comparison. 1: Enables comparison.

16.3.5 DPSELRM — DPIN Select Register M

This register specifies the compare target bits in the DPDSRM and DPDIMRm (m = 4 to 1) registers.

Write to the DPSELRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPSELRM can be read or written in 32-bit units.
DPSELRML and DPSELRMH can be read or written in 16-bit units.
DPSELR1, DPSELR2, DPSELR3, and DPSELR4 can be read or written in 8-bit units.

Address: DPSELRM: <LPS0_base> + $0C_H$
DPSELRML: <LPS0_base> + $0C_H$, DPSELRMH: <LPS0_base> + $0E_H$
DPSELR1: <LPS0_base> + $0C_H$, DPSELR2: <LPS0_base> + $0D_H$, DPSELR3: <LPS0_base> + $0E_H$,
DPSELR4: <LPS0_base> + $0F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4EN_7	D4EN_6	D4EN_5	D4EN_4	D4EN_3	D4EN_2	D4EN_1	D4EN_0	D3EN_7	D3EN_6	D3EN_5	D3EN_4	D3EN_3	D3EN_2	D3EN_1	D3EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2EN_7	D2EN_6	D2EN_5	D2EN_4	D2EN_3	D2EN_2	D2EN_1	D2EN_0	D1EN_7	D1EN_6	D1EN_5	D1EN_4	D1EN_3	D1EN_2	D1EN_1	D1EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.15 DPSELRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fifth data captured at the digital input pins and stored in the DPDIMR4 register with the comparison target data in the DPDSR4 register. 0: Disables comparison. 1: Enables comparison.
23 to 16	D3EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fourth data captured at the digital input pins and stored in the DPDIMR3 register with the comparison target data in the DPDSR3 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D2EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the third data captured at the digital input pins and stored in the DPDIMR2 register with the comparison target data in the DPDSR2 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D1EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the second data captured at the digital input pins and stored in the DPDIMR1 register with the comparison target data in the DPDSR1 register. 0: Disables comparison. 1: Enables comparison.

16.3.6 DPSELRH — DPIN Select Register H

This register specifies the compare target bits in the DPDSRH and DPDIMRm (m = 7 to 5) registers.

Write to the DPSELRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPSELRH can be read or written in 32-bit units.
DPSELRHL and DPSELRHH can be read or written in 16-bit units.
DPSELR5, DPSELR6, and DPSELR7 can be read or written in 8-bit units.

Address: DPSELRH: <LPS0_base> + 10_H
DPSELRHL: <LPS0_base> + 10_H, DPSELRHH: <LPS0_base> + 12_H
DPSELR5: <LPS0_base> + 10_H, DPSELR6: <LPS0_base> + 11_H, DPSELR7: <LPS0_base> + 12_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7EN_7	D7EN_6	D7EN_5	D7EN_4	D7EN_3	D7EN_2	D7EN_1	D7EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6EN_7	D6EN_6	D6EN_5	D6EN_4	D6EN_3	D6EN_2	D6EN_1	D6EN_0	D5EN_7	D5EN_6	D5EN_5	D5EN_4	D5EN_3	D5EN_2	D5EN_1	D5EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.16 DPSELRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the eighth data captured at the digital input pins and stored in the DPDIMR7 register with the compare target data in the DPDSR7 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D6EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the seventh data captured at the digital input pins and stored in the DPDIMR6 register with the compare target data in the DPDSR6 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D5EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the sixth data captured at the digital input pins and stored in the DPDIMR5 register with the compare target data in the DPDSR5 register. 0: Disables comparison. 1: Enables comparison.

16.3.7 DPDSR0 — DPIN Data Set Register 0

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR0 register.

Write to the DPDSR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 14_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0_23	D0_22	D0_21	D0_20	D0_19	D0_18	D0_17	D0_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0_15	D0_14	D0_13	D0_12	D0_11	D0_10	D0_9	D0_8	D0_7	D0_6	D0_5	D0_4	D0_3	D0_2	D0_1	D0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.17 DPDSR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0_n (n = 23 to 0)	Data to be compared with the first digital port input (DPINm)

16.3.8 DPDSRM — DPIN Data Set Register M

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR4 to DPDIMR1 registers.

Write to the DPDSRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRM can be read or written in 32-bit units.
DPDSRML and DPDSRMH can be read or written in 16-bit units.
DPDSR1, DPDSR2, DPDSR3, and DPDSR4 can be read or written in 8-bit units.

Address: DPDSRM: <LPS0_base> + 18_H
DPDSRML: <LPS0_base> + 18_H, DPDSRMH: <LPS0_base> + $1A_H$
DPDSR1: <LPS0_base> + 18_H, DPDSR2: <LPS0_base> + 19_H, DPDSR3: <LPS0_base> + $1A_H$,
DPDSR4: <LPS0_base> + $1B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4_7	D4_6	D4_5	D4_4	D4_3	D4_2	D4_1	D4_0	D3_7	D3_6	D3_5	D3_4	D3_3	D3_2	D3_1	D3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2_7	D2_6	D2_5	D2_4	D2_3	D2_2	D2_1	D2_0	D1_7	D1_6	D1_5	D1_4	D1_3	D1_2	D1_1	D1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.18 DPDSRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4_n (n = 7 to 0)	Data to be compared with the fifth digital port input (DPINm)
23 to 16	D3_n (n = 7 to 0)	Data to be compared with the fourth digital port input (DPINm)
15 to 8	D2_n (n = 7 to 0)	Data to be compared with the third digital port input (DPINm)
7 to 0	D1_n (n = 7 to 0)	Data to be compared with the second digital port input (DPINm)

16.3.9 DPDSRH — DPIN Data Set Register H

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR7 to DPDIMR5 registers.

Write to the DPDSRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRH can be read or written in 32-bit units.
DPDSRHL and DPDSRHH can be read or written in 16-bit units.
DPDSR5, DPDSR6, and DPDSR7 can be read or written in 8-bit units.

Address: DPDSRH: <LPS0_base> + $1C_H$
DPDSRHL: <LPS0_base> + $1C_H$, DPDSRHH: <LPS0_base> + $1E_H$
DPDSR5: <LPS0_base> + $1C_H$, DPDSR6: <LPS0_base> + $1D_H$, DPDSR7: <LPS0_base> + $1E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7_7	D7_6	D7_5	D7_4	D7_3	D7_2	D7_1	D7_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6_7	D6_6	D6_5	D6_4	D6_3	D6_2	D6_1	D6_0	D5_7	D5_6	D5_5	D5_4	D5_3	D5_2	D5_1	D5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.19 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7_n (n = 7 to 0)	Data to be compared with the eighth digital port input (DPINm)
15 to 8	D6_n (n = 7 to 0)	Data to be compared with the seventh digital port input (DPINm)
7 to 0	D5_n (n = 7 to 0)	Data to be compared with the sixth digital port input (DPINm)

16.3.10 DPDIMR0 — DPIN Data Input Monitor Register 0

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 23)) in digital input mode. DPDIMR0 stores the data acquired for the first time.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <LPS0_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0M_23	D0M_22	D0M_21	D0M_20	D0M_19	D0M_18	D0M_17	D0M_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0M_15	D0M_14	D0M_13	D0M_12	D0M_11	D0M_10	D0M_9	D0M_8	D0M_7	D0M_6	D0M_5	D0M_4	D0M_3	D0M_2	D0M_1	D0M_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.20 DPDIMR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	D0M_n (n = 23 to 0)	The first digital port input (DPINm) data

16.3.11 DPDIMR1 — DPIN Data Input Monitor Register 1

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR1 stores the data acquired for the second time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D1M_7	D1M_6	D1M_5	D1M_4	D1M_3	D1M_2	D1M_1	D1M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.21 DPDIMR1 Register Contents

Bit Position	Bit Name	Function
7 to 0	D1M_n (n = 7 to 0)	The second digital port input (DPINm) data

16.3.12 DPDIMR2 — DPIN Data Input Monitor Register 2

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR2 stores the data acquired for the third time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 28_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D2M_7	D2M_6	D2M_5	D2M_4	D2M_3	D2M_2	D2M_1	D2M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.22 DPDIMR2 Register Contents

Bit Position	Bit Name	Function
7 to 0	D2M_n (n = 7 to 0)	The third digital port input (DPINm) data

16.3.13 DPDIMR3 — DPIN Data Input Monitor Register 3

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR3 stores the data acquired for the fourth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D3M_7	D3M_6	D3M_5	D3M_4	D3M_3	D3M_2	D3M_1	D3M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.23 DPDIMR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	D3M_n (n = 7 to 0)	The fourth digital port input (DPINm) data

16.3.14 DPDIMR4 — DPIN Data Input Monitor Register 4

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR4 stores the data acquired for the fifth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 30_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D4M_7	D4M_6	D4M_5	D4M_4	D4M_3	D4M_2	D4M_1	D4M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.24 DPDIMR4 Register Contents

Bit Position	Bit Name	Function
7 to 0	D4M_n (n = 7 to 0)	The fifth digital port input (DPINm) data

16.3.15 DPDIMR5 — DPIN Data Input Monitor Register 5

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR5 stores the data acquired for the sixth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 34_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D5M_7	D5M_6	D5M_5	D5M_4	D5M_3	D5M_2	D5M_1	D5M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.25 DPDIMR5 Register Contents

Bit Position	Bit Name	Function
7 to 0	D5M_n (n = 7 to 0)	The sixth digital port input (DPINm) data

16.3.16 DPDIMR6 — DPIN Data Input Monitor Register 6

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR6 stores the data acquired for the seventh time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D6M_7	D6M_6	D6M_5	D6M_4	D6M_3	D6M_2	D6M_1	D6M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.26 DPDIMR6 Register Contents

Bit Position	Bit Name	Function
7 to 0	D6M_n (n = 7 to 0)	The seventh digital port input (DPINm) data

16.3.17 DPDIMR7 — DPIN Data Input Monitor Register 7

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode. DPDIMR7 stores the data acquired for the eighth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $3C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D7M_7	D7M_6	D7M_5	D7M_4	D7M_3	D7M_2	D7M_1	D7M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.27 DPDIMR7 Register Contents

Bit Position	Bit Name	Function
7 to 0	D7M_n (n = 7 to 0)	The eighth digital port input (DPINm) data

16.3.18 CNTVAL — Count Value Register

This register specifies the stabilization time of the external circuits (digital signal source and analog signal source).

- In digital mode
  The time from when the DPO output is set to 1 to the time when the port input is acquired for the first time
- In analog mode
  The time from when the APO output is set to 1 to the time when the LPS outputs the A/D conversion trigger to the ADCJ2

Write to the CNTVAL register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 16-bit units.

Address: <LPS0_base> + 40_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT17	CNT16	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT07	CNT06	CNT05	CNT04	CNT03	CNT02	CNT01	CNT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.28 CNTVAL Register Contents

Bit Position	Bit Name	Function
15 to 8	CNT1n (n = 7 to 0)	These bits set the stabilization time of the external circuit (analog signal source). Stabilization time = (1/(f_{RH}/20)) × 16 × CNT1n (set value)
7 to 0	CNT0n (n = 7 to 0)	These bits set the stabilization time of the external circuit (digital signal source). Stabilization time = (1/(f_{RH}/20)) × 16 × CNT0n (set value)

16.3.19 SOSTR — LPS Operation Status Register

This register indicates the operating state of the LPS.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 44_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SOF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.29 SOSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	SOF	LPS Operation Status Flag 0: Initial state before the occurrence of the start trigger 1: LPS operation is in progress (after the start trigger occurs) If the start trigger occurs while the SOF bit is set to 1 (during the LPS operation), the start trigger is canceled.

NOTE

When DPEN or ADEN is set to 0 for stop of the LPS, SOF bit is set to 0 after the bit becomes 1 once by hardware.

16.4 Digital Input Mode

With the digital input port DPINm and the externally connected multiplexer, up to 80 input ports can be monitored as shown in **Table 16.30, Combination of Monitored Ports**.

Port DPSELk is used to switch the external multiplexer. The DPSELk output is switched for the number of times specified in the SCTLR register.

TAUJy is used to set the timing to check the value input to the port.

Table 16.30 Combination of Monitored Ports

Combination (Number of Ports x Number of Checks)	Ports Used	Total Number
Direct mode When input ports are checked simultaneously without using the external multiplexer Up to 24 ports × 1	DPIN23 to DPIN0	Up to 24
Multiplexer mode When input ports are checked by using a small number of pins and the external multiplexer Up to 8 ports × 8	DPIN7 to DPIN0 DPSEL2 to DPSEL0	Up to 64
MIX mode When input ports are checked using a combination of the above two modes Up to 24 ports x 1 + Up to 8 ports x 7	DPIN23 to DPIN0*1 DPSEL2 to DPSEL0	Up to 80

Note 1. DPIN23 to DPIN8 are checked only for the first time.

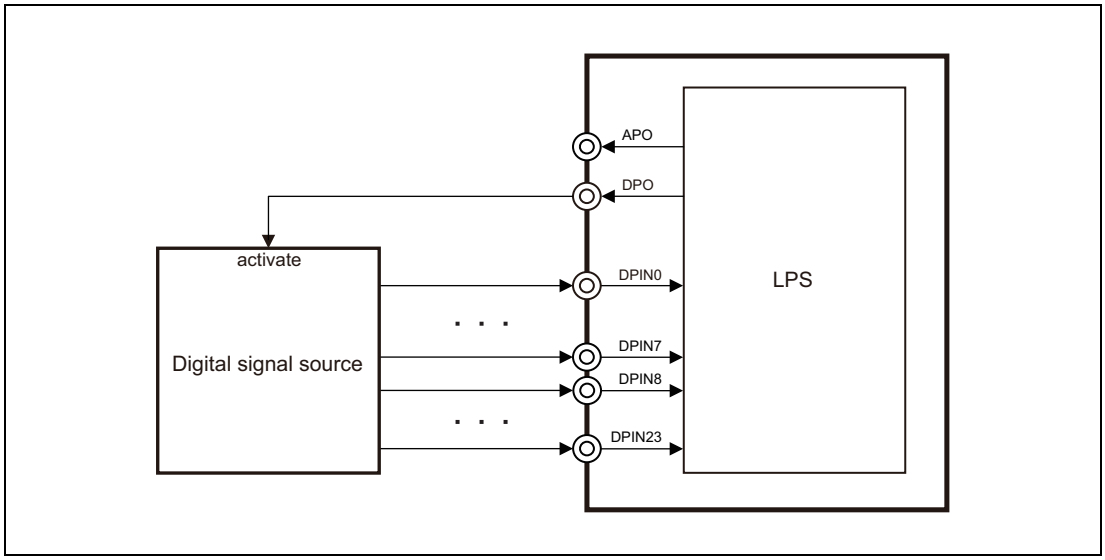


Figure 16.2 Direct Mode Connection Example

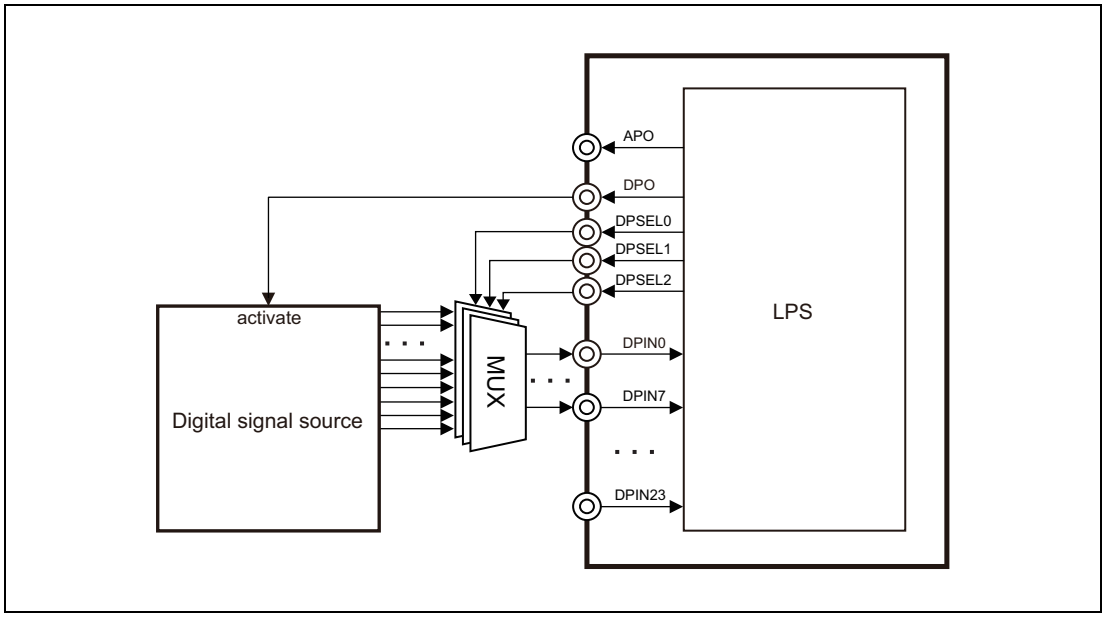


Figure 16.3 Multiplexer Mode Connection Example

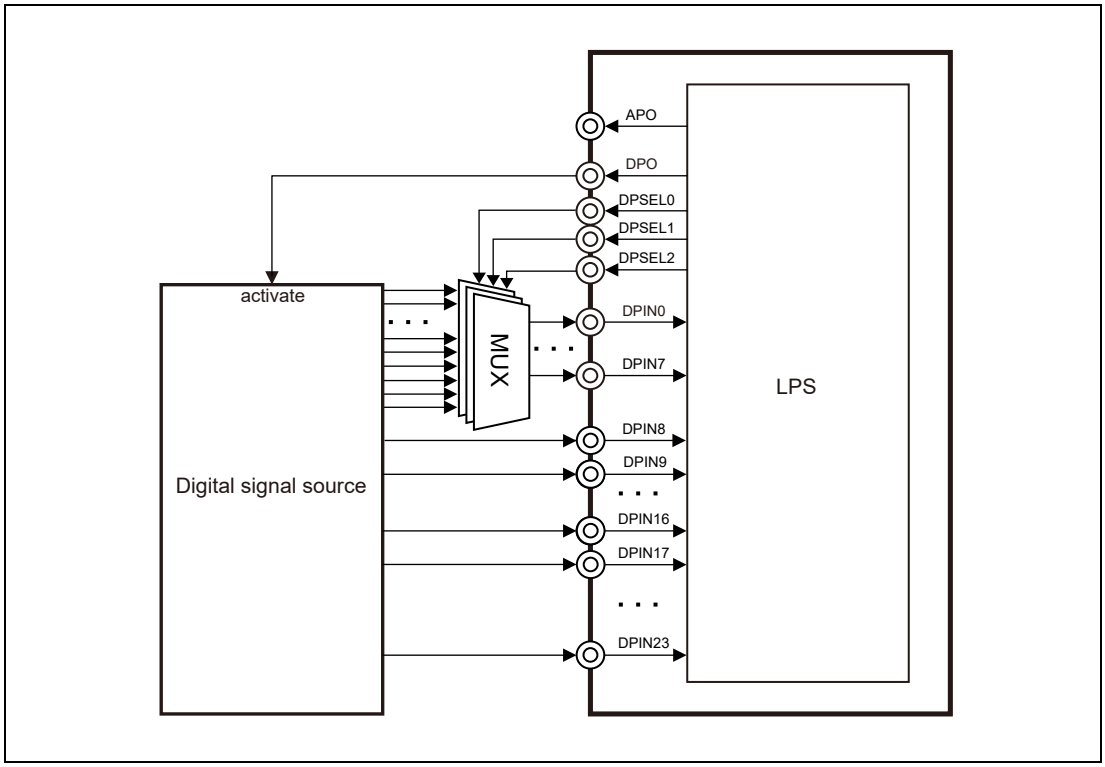


Figure 16.4 MIX Mode Connection Example

Preparation

- Set NUMDP[2:0] and TJIS[2:0] bits in the SCTLR register to specify the number of times the port is to be read, and the TAUJy interrupt to be used as sequence start trigger.
- Set TAUJy to interval timer mode.
- Set the wait time of the digital signal source by using the lower 8 bits in the CNTVAL register.
- Set expected values in the DPDSR0, DPDSRM and DPDSRH registers.
- Set the ports to be checked in the DPSELR0, DPSELRM, and DPSELRH registers.

Start

- Start the TAUJy.
- Set the SCTLR.DPEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJy. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in standby mode, it can only resume operation while the sequencer is running.

Upon completion of checking all ports that have been set, the INTCWEND interrupt occurs. In addition, if the input value of the port is different from the expected value set by the DPDSR0, DPDSRM, or DPDSRH register, the wake-up factor WUTRG0 occurs. The following figures show an example of the operation in digital input mode.

Stop

To stop the LPS operation in Digital Input Mode (by changing the SCTLR.DPEN bit setting from 1 to 0), follow the procedure shown below. In this example, the P6_4 pin is used as DPO.

1. Set the port register to specify low level output on the pin (P6.P6_4 = 0).
2. Change the setting for the P6_4 pin from the alternative port mode to the port mode (PMC6.PMC6_4 = 0).
3. Set SCTLR.DPEN = 0.

NOTE

The above procedure applies when the P6_4 pin is used as DPO. If the P6_12 pin is used as DPO, specify the P6_12 pin settings in the same way.

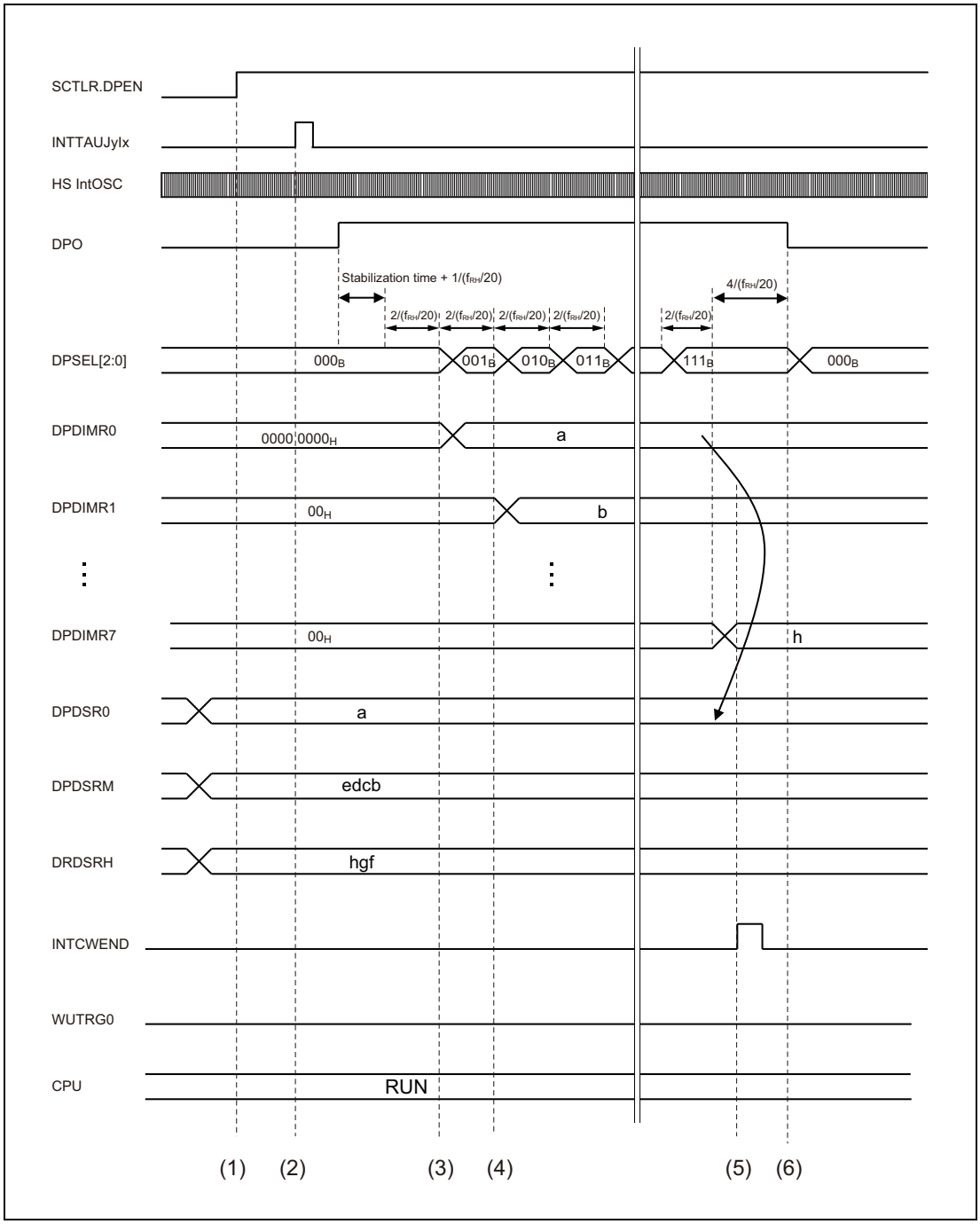


Figure 16.5 Operation of Digital Input Mode (8 Ports × 8) when the Input Value is not Changed (RUN Mode)

(1) Set the SCTLR.DPEN bit to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the sequencer outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(3) After the completion of the signal source stabilization, the LPS stores the DPIN[7:0] input value to the DPDIMR0 register and increments the DPSEL[2:0] pins to switch the external multiplexer.

(4) After the switching of the DPSEL[2:0] pins, the LPS stores the values in the DPDIMRn registers in order from DPDIMR1 and continues to increment the DPSEL[2:0] pins.

(5) After the value is stored up to the DPDIMR7 register, the INTCWEND interrupt is generated and the value is compared with the expected value set in the DPDSR0, DPDSRM, and DPDSRH registers.

(6) When the value is not different from the expected value, the wake-up factor WUTRG0 is not generated. The LPS stops the DPO output and returns to the waiting state for the trigger.

High level width of DPO pin is calculated by the following formula. For the Stabilization time, see **Section 16.3.18, CNTVAL — Count Value Register**.

High level width of DPO pin

= Stabilization time + $(1/(f_{RH}/20)) \times 1 + (1/(f_{RH}/20)) \times 2 \times$ (SCTLR.NUMDP (set value) + 1) + $(1/(f_{RH}/20)) \times 4$

= Stabilization time + $(1/(f_{RH}/20)) \times$ (2 × SCTLR.NUMDP (set value) + 7)

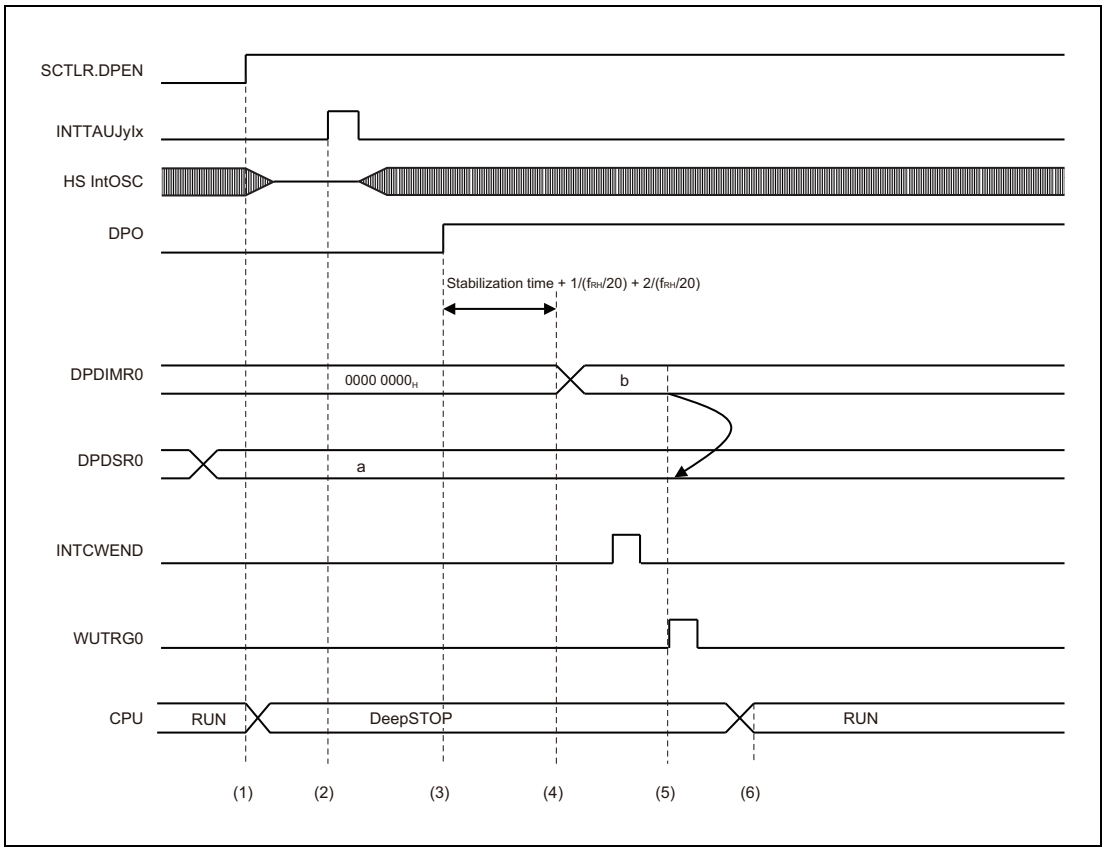


Figure 16.6 Operation of Digital Input Mode (24 Ports × 1) when the Input Value is Changed (DeepSTOP Mode)

(1) Set the STBC0PSC.STBC0DISTRG bit to 1 to transition to the DeepSTOP mode, while the SCTLR.DPEN bit is set to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(3) After the completion of the HS IntOSC stabilization time, the LPS outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(4) After the completion of the signal source stabilization, the LPS stores the DPIN[23:0] input value to the DPDIMR0 register and the INTCWEND interrupt is generated.

(5) The value stored in the DPDIMR0 register is compared with the expected value set in the DPDSR0 register. When the value is different from the expected value, the wake-up factor WUTRG0 is generated.

(6) The CPU returns to RUN mode at the generation of WUTRG0. The DPO pin is driven high until the EVFR.DINEVF bit is cleared to 0 by software.

16.4.1 Digital Port Error Interrupt

A level sensitive interrupt indicating a data comparison mismatch is generated. This interrupt is generated not only in standby mode but also in RUN mode. The set and clear conditions are shown below.

Table 16.31 Digital Port Error Interrupt

Unit Interrupt Signal	Set Condition	Clear Condition
INTDPE	When EVFR.DINEVF is set to 1 by hardware	When EVFR.DINEVF is cleared to 0 by software

16.5 Analog Input Mode

The analog input port ADCJ2Im (m = 0 to 19) can be monitored.

TAUJy is used to set the timing to check the value input to the port.

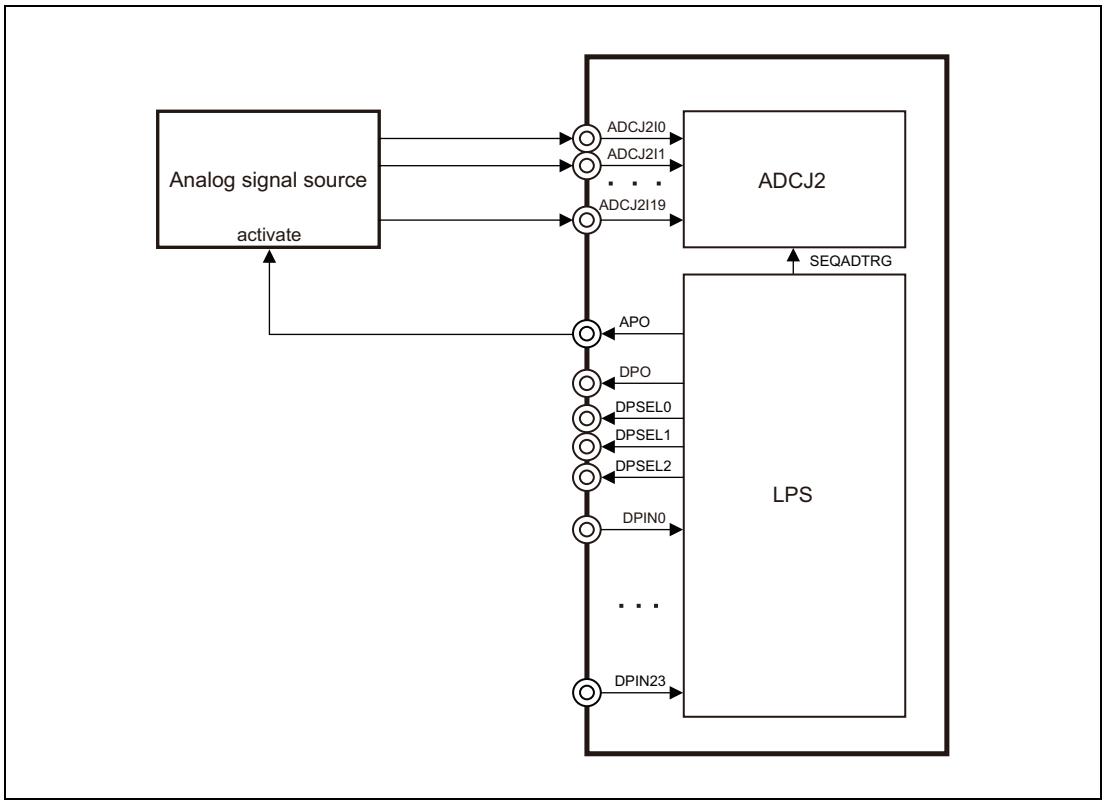


Figure 16.7 Analog Input Mode Connection Example 1

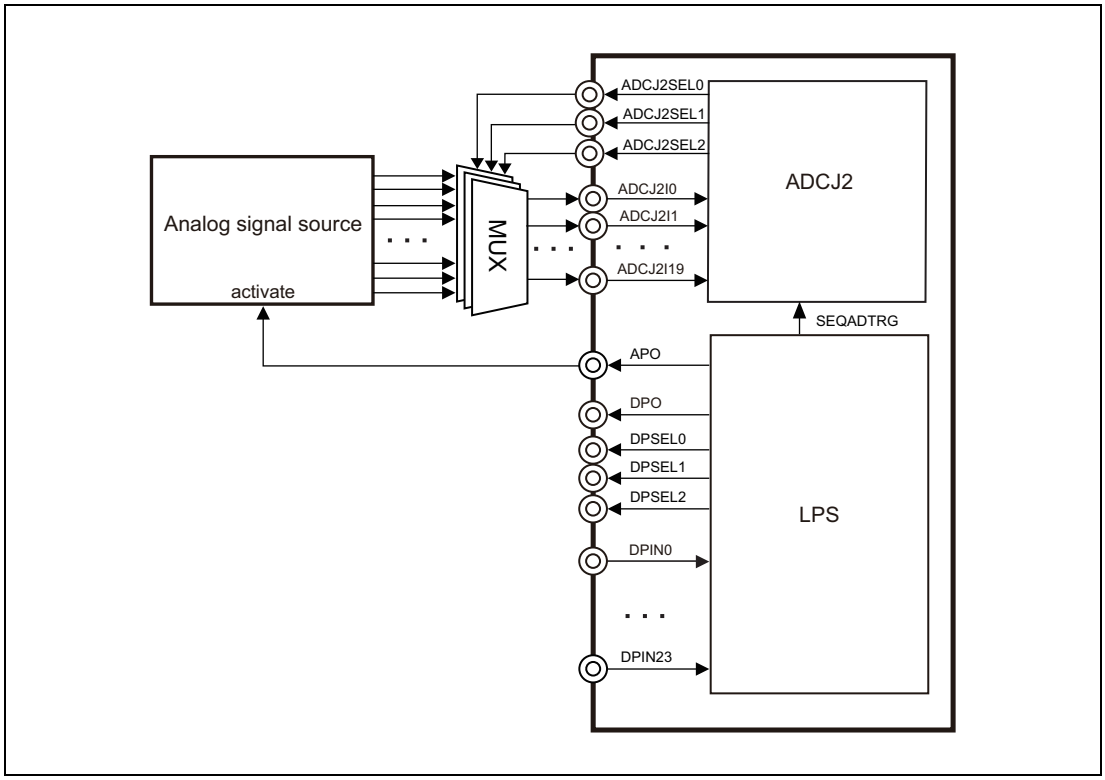


Figure 16.8 Analog Input Mode Connection Example 2

Preparation

- Set TJIS[2:0] bits in the SCTLR register to specify the TAUJy interrupt to be used as sequence start trigger.
- Set TAUJy to interval timer mode.
- Set the wait time of the analog signal source by using the upper 8 bits in the CNTVAL register.
- Set the ADCJ2.

CAUTIONS

1. **When the LPS is in use, the A/D conversion completion interrupt (INTADCJ2Ix) should be output after the conversion of all channels of the LPS has been completed. The setting is as follows.**
   - **Set the ADIE bit in virtual channel register j (ADCJ2VCRj) to 0 (a scan group x end interrupt (INTADCJ2Ix) is not generated when A/D conversion for virtual channel j ends in SGx.).**
   - **Set the ADIE bit in the scan group x control register (ADCJ2SGCRx) to 1 (INTADCJ2Ix is output when the scan for SGx ends).**
2. **Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[2:0] bits to the completion of A/D conversion for all channels of LPS, do not forcibly end A/D conversion by using the ADCJ2ADHALTR bit.**
3. **When the LPS is in use, do not use the following modes.**

- **Continuous scan mode**
  (the setting ADCJ2SGCRx.SCANMD = 1 is prohibited)
- **Multicycle scan mode with 2 or more cycles**
  (the settings except ADCJ2SGMCYCRx.MCYC = 00_H are prohibited)

Start

- Start the TAUJy.
- Set the SCTLR.ADEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJy. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in standby mode, it can only resume operation while the sequencer is running.

To detect whether the analog input value differs from the expected value, use the A/D error interrupt request (INTADCJ2ERR) of the A/D converter.

In addition, if the analog input value is different from the expected value, the wake-up factor WUTRG1 occurs.

For details on the A/D error interrupt request (INTADCJ2ERR), see **Section 43.4.19.2, A/D Error Interrupt Request (INT_ADE)***1. The following figures show an example of the operation in analog input mode.

Note 1. In **Section 43, Analog to Digital Converter (ADCJ)**, the name of the A/D error interrupt request is described as “INT_ADE”.

Stop

To stop the LPS operation in Analog Input Mode (by changing the SCTLR.ADEN bit setting from 1 to 0), follow the procedure shown below. Note that the P6_8 pin is used as APO.

1. Set the port register to specify low level output on the pin (P6.P6_8 = 0).
2. Change the setting for the P6_8 pin from the alternative port mode to the port mode (PMC6.PMC6_8 = 0).
3. Set SCTLR.ADEN = 0.

NOTE

The above procedure applies when the P6_8 pin is used as APO. If the P6_13 pin is used as APO, specify the P6_13 pin settings in the same way.

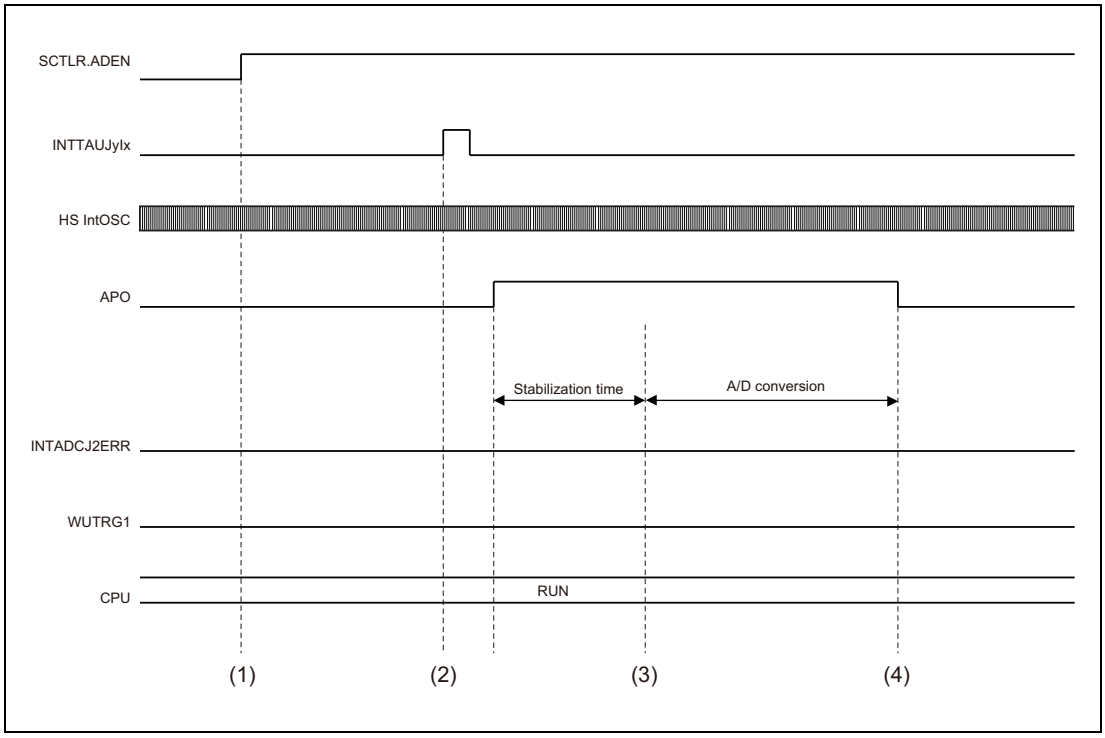


Figure 16.9 Operation of Analog Input Mode when the Conversion Result is within the Expected Range (RUN Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.

(3) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCJ2Im (m = 0 to 19), set in the A/D converter scan group, is started.

(4) When the INTADCJ2ERR interrupt is not generated as a result of A/D conversion, the LPS halts the A/D converter and resets the APO pin.

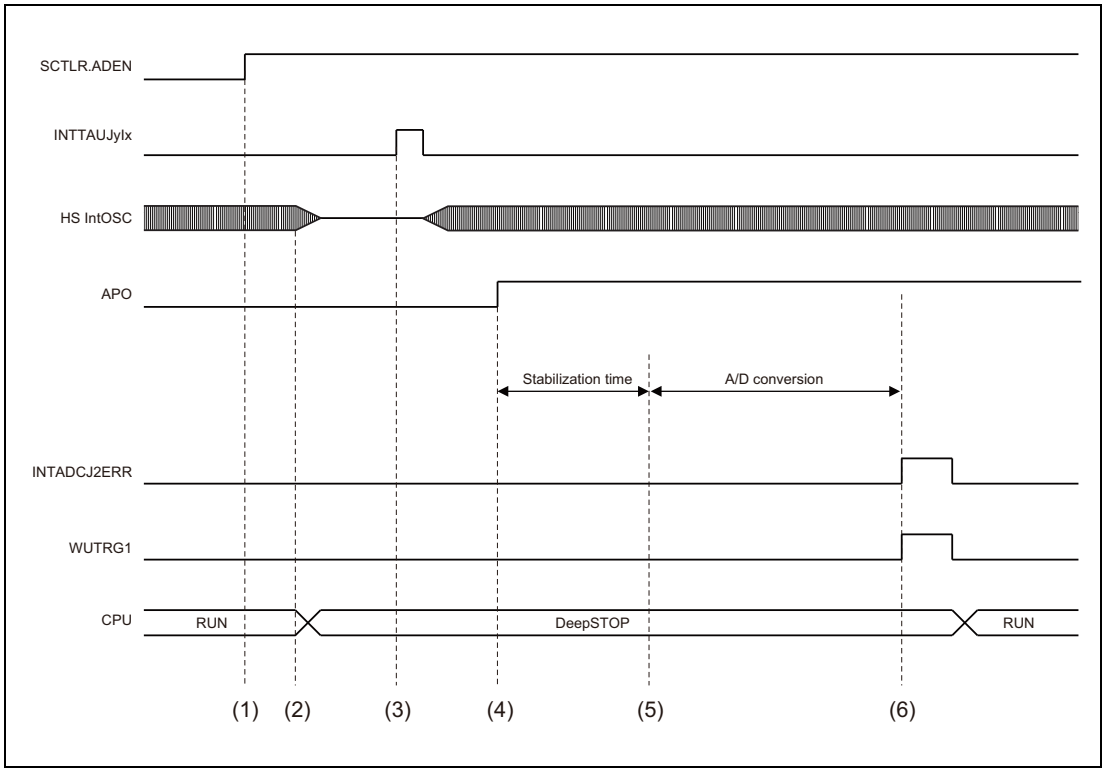


Figure 16.10 Operation of Analog Input Mode when the Conversion Result is not within the Expected Range (DeepSTOP Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) Set the STBC0PSC.STBC0DISTRG bit to 1 by software to transition to the DeepSTOP mode.

(3) When the INTTAUJyIx interrupt specified by the SCTLR.TJIS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(4) After the completion of the HS IntOSC stabilization, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.

(5) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCJ2Im (m = 0 to 19), set in the A/D converter scan group, is started.

(6) When the INTADCJ2ERR interrupt is generated as a result of A/D conversion, the wake-up factor WUTRG1 is generated and the CPU returns to RUN mode. The APO pin is driven high until the upper limit/lower limit error flag of the A/D converter is cleared to 0 by software.

NOTE

WUTRG1 is generated when the first ADCJ2 SG interrupt occurred after ADCJ2SGULCRx.UE or ADCJ2SGULCRx.LE is set.

Section 17 Serial Flash Memory Interface A (SFMA)

This section contains a generic description of the Serial Flash Memory Interface A (SFMA).

The first section describes all properties specific to the RH850/U2A-EVA, such as units, register base addresses, input/output signal names, etc.

The remainder of the section describes the function and registers of the SFMA.

17.1 Features SFMA of RH850/U2A-EVA

17.1.1 Number of Units and Channels

This microcontroller has the following number of SFMA units.

Table 17.1 Number of Channels

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)	RH850/U2A6 (292 pins)	RH850/U2A6 (176 pins)	RH850/U2A6 (156 pins)	RH850/U2A6 (144 pins)
Number of channels	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	—	—
Name	SFMAn									

Table 17.2 Index

Index	Description
n	Throughout this section, the individual SFMA units are identified by the index “n” (n = 0); for example, SFMAnCMNCR is the SFMAn common control register.

17.1.2 Register Base Address

SFMAn base address is listed in the following table.

SFMAn register addresses are given as offsets from the base address.

Table 17.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<SFMA0_base>	$1004\ 0000_H$	H-Bus Group 3

Note: Throughout this section, <SFMAn_base> is also same with <SFMA0_base> with index n = 0.

17.1.3 Clock Supply

The SFMAn clock supply is shown in the following table.

Table 17.4 Clock Supplies

Unit Name	Unit Clock Name	Supply Clock Name	Description
SFMAn	Bϕ	CLK_HSB	SFMAn clock
	Register access clock	CLK_HSB	SFMAn clock

17.1.4 Reset Sources

SFMAn reset sources are listed in the following table. SFMAn is initialized by these reset sources.

Table 17.5 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
SFMAn	All registers	√	√	√	√	√	√	—

17.1.5 External Input/Output Signals

External input/output signals of SFMAn are listed below.

Table 17.6 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
SFMA0		
SPBCLK	Clock output	SFMA0CLK
SPBSSL	Slave select signal output	SFMA0SSL
SPBMO/SPBIO0	Master transmit data/data 0	SFMA0IO0
SPBMI/SPBIO1	Master input data/data 1	SFMA0IO1
SPBIO2	Data 2	SFMA0IO2
SPBIO3	Data 3	SFMA0IO3

17.1.6 Interrupts and Error Notifications

SFMAn has no interrupts and DMA/DTS requests.

SFMAn has no error notifications.

17.2 Overview

17.2.1 Functional Overview

The Serial Flash Memory Interface outputs control signals to the serial flash memory connected to the SPI multi I/O bus space, thus enabling direct connection of the serial flash memory.

This module allows the connected serial flash memory to be accessed by directly reading the SPI multi I/O bus space, or using SPI operating mode to transmit and receive data.

- Serial Flash Memory Interface

  One serial flash memory device can be connected.

  A data bus size of 1 bit, 2 bits, or 4 bits can be selected.

- External Address Space Read Mode

  A maximum of 4-Gbyte address space is supported.

  The SPBSSL pin can be automatically controlled through access address monitoring.

  Efficient data reception due to built-in read cache (64-bit line × 16 entries).

- SPI Operating Mode

  Desired read/write access to serial flash memory is possible.

- Bit rate

  SPBCLK is generated by frequency division of Bϕ by the internal baud rate generator.

  SPBCLK frequency division ratio can be set from 2 to 4080.

- SPBSSL Pin Control

  Delay from SPBSSL signal assertion to SPBCLK operation (clock delay) can be set.
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)

  Delay from SPBCLK stop to SPBSSL output negation (SPBSSL negation delay) can be set.
  Range: 1.5 to 8.5 SPBCLK cycles (set in SPBCLK-cycle units)

  SPBSSL output assertion wait before next access (next access delay) can be set.
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)

  SPBSSL polarity can be changed.

17.2.2 Block Diagram

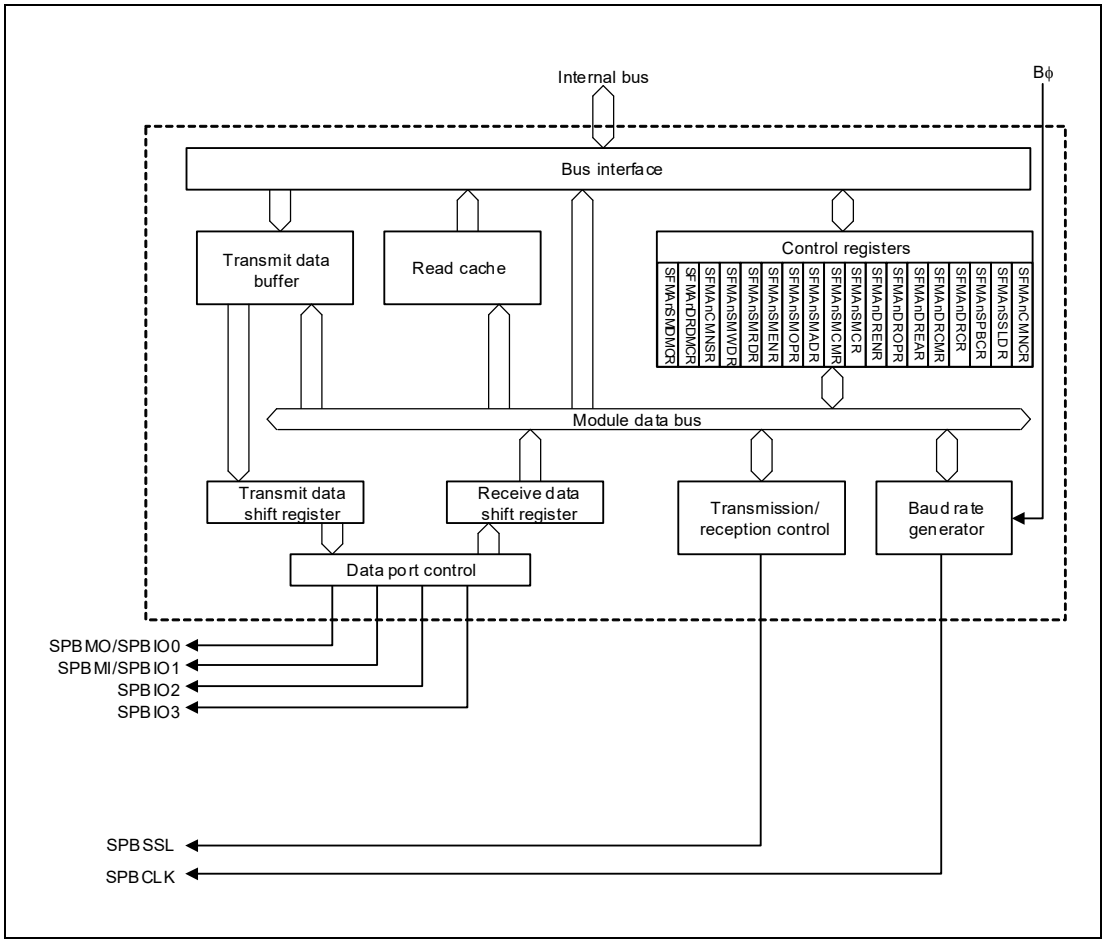


Figure 17.1 SFMA Block Diagram

17.3 Registers

17.3.1 List of Registers

SFMA registers are listed in the following table.

For details about <SFMAn_base>, see **Section 17.1.2, Register Base Address**.

Table 17.7 List of Registers

Module Name	Register Name	Symbol	Address size	Access size	Access Protection	
					HBG	Other
SFMAn	SFMAn Common control register	SFMAnCMNCR	<SFMAn_base> + 00_H	32	HBG96	—
SFMAn	SFMAn SSL delay register	SFMAnSSLDR	<SFMAn_base> + 04_H	32	HBG96	—
SFMAn	SFMAn Bit rate register	SFMAnSPBCR	<SFMAn_base> + 08_H	32	HBG96	—
SFMAn	SFMAn Data read control register	SFMAnDRCR	<SFMAn_base> + $0C_H$	32	HBG96	—
SFMAn	SFMAn Data read command setting register	SFMAnDRCMR	<SFMAn_base> + 10_H	32	HBG96	—
SFMAn	SFMAn Data read extended address setting register	SFMAnDREAR	<SFMAn_base> + 14_H	32	HBG96	—
SFMAn	SFMAn Data read option setting register	SFMAnDROPR	<SFMAn_base> + 18_H	32	HBG96	—
SFMAn	SFMAn Data read enable setting register	SFMAnDRENR	<SFMAn_base> + $1C_H$	32	HBG96	—
SFMAn	SFMAn SPI mode control register	SFMAnSMCR	<SFMAn_base> + 20_H	32	HBG96	—
SFMAn	SFMAn SPI mode command setting register	SFMAnSMCMR	<SFMAn_base> + 24_H	32	HBG96	—
SFMAn	SFMAn SPI mode address setting register	SFMAnSMADR	<SFMAn_base> + 28_H	32	HBG96	—
SFMAn	SFMAn SPI mode option setting register	SFMAnSMOPR	<SFMAn_base> + $2C_H$	32	HBG96	—
SFMAn	SFMAn SPI mode enable setting register	SFMAnSMENR	<SFMAn_base> + 30_H	32	HBG96	—
SFMAn	SFMAn SPI mode read data register	SFMAnSMRDR	<SFMAn_base> + 38_H	8, 16,32	HBG96	—
SFMAn	SFMAn SPI mode write data register	SFMAnSMWDR	<SFMAn_base> + 40_H	8, 16,32	HBG96	—
SFMAn	SFMAn Common status register	SFMAnCMNSR	<SFMAn_base> + 48_H	32	HBG96	—
SFMAn	SFMAn Data read dummy cycle setting register	SFMAnDRDMCR	<SFMAn_base> + 58_H	32	HBG96	—
SFMAn	SFMAn SPI mode dummy cycle setting register	SFMAnSMDMCR	<SFMAn_base> + 60_H	32	HBG96	—

17.3.2 SFMAnCMNCR — SFMAn Common Control Register

SFMAnCMNCR is a 32-bit register that controls the SPI multi-I/O bus controller. The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 00_H

Value after reset: $01AA\ 4000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD	—	—	—	—	—	—	—	MOIIO3[1:0]		MOIIO2[1:0]		MOIIO1[1:0]		MOIIO0[1:0]	
Value after reset	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO3FV[1:0]		IO2FV[1:0]		—	—	IO0FV[1:0]		—	CPHAT	CPHAR	SSLP	CPOL	—	—	—
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R

Table 17.8 SFMAnCMNCR Register Contents (1/2)

Bit Position	Bit Name	Function
31	MD	Operating Mode Switch Switches the operating modes. 0: External address space read mode 1: SPI operating mode
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23, 22	MOIIO3[1:0]	SPBSSL Output Idle Value Fix SPBIO3 Fixes the output value of SPBIO3 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
21, 20	MOIIO2[1:0]	SPBSSL Output Idle Value Fix SPBIO2 Fixes the output value of SPBIO2 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
19, 18	MOIIO1[1:0]	SPBSSL Output Idle Value Fix SPBIO1 Fixes the output value of SPBIO1 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
17, 16	MOIIO0[1:0]	SPBSSL Output Idle Value Fix SPBIO0 Fixes the output value of SPBIO0 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z

Table 17.8 SFMAnCMNCR Register Contents (2/2)

Bit Position	Bit Name	Function
15, 14	IO3FV[1:0]	SPBIO3 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO3 pin for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
13, 12	IO2FV[1:0]	SPBIO2 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO2 pin for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	IO0FV[1:0]	SPBIO0 Fixed Value for 1-bit Size Input Fixes the output value of SPBIO0 pin for 1-bit size input. 00: Output value 0 01: Output value 1 10: Output value is kept at a preceding value. 11: Output value Hi-Z
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	CPHAT	Output Shift Sets the SPBCLK edge of the output data. CPHAT and CPHAR should be set according to the table in the description of CPHAR. 0: Data transmission at even edge 1: Data transmission at odd edge
5	CPHAR	Input Latch Sets the SPBCLK edge of the reception data. CPHAT and CPHAR should be set according to the following table. 0: Data reception at odd edge 1: Data reception at even edge
4	SSLP	SPBSSL Signal Polarity Sets the polarity of SPBSSL signal. 0: Active low SPBSSL signal 1: Active high SPBSSL signal
3	CPOL	SPBSSL Negation Period SPBCLK Output Direction Sets the SPBCLK output direction during SPBSSL negation period. 0: SPBCLK output is 0 during SPBSSL negation period. 1: SPBCLK output is 1 during SPBSSL negation period.
2 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CPHAT	CPHAR	
0	0	Setting enabled
0	1	Setting enabled
1	0	Setting prohibited
1	1	Setting enabled

17.3.3 SFMAnSSLDR — SFMAn SSL Delay Register

SFMAnSSLDR is a 32-bit register that adjusts the timing between the SPBSSL signal and the SPBCLK signal.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 04_H

Value after reset: $0007\ 0707_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Value after reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 17.9 SFMAnSSLDR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	SPNDL[2:0]	Next Access Delay Sets the period from transfer end to next transfer start (next access). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	SLNDL[2:0]	SPBSSL Negation Delay Sets the period from the time the last SPBCLK edge is sent of a transfer to SPBSSL pin negation (SPBSSL negation delay). 000: 1.5 SPBCLK cycles 001: 2.5 SPBCLK cycles 010: 3.5 SPBCLK cycles 011: 4.5 SPBCLK cycles 100: 5.5 SPBCLK cycles 101: 6.5 SPBCLK cycles 110: 7.5 SPBCLK cycles 111: 8.5 SPBCLK cycles
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.9 SFMAnSSLDR Register Contents (2/2)

Bit Position	Bit Name	Function
2 to 0	SCKDL[2:0]	Clock Delay Sets the period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles

17.3.4 SFMAnSPBCR — SFMAn Bit Rate Register

SFMAnSPBCR is a 32-bit register that sets the bit rate.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 08_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPBR[7:0]								—	—	—	—	—	—	BRDV[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 17.10 SFMAnSPBCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	SPBR[7:0]	Bit Rate Sets the bit rate. The bit rate is determined by a combination of these bits with the BRDV[1:0] bits. Setting SPBR[7:0] to 0 is prohibited. SPBR[7:0] needs to be changed to non-zero value before SPI communication starts.
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	BRDV[1:0]	Bit Rate Frequency Division Sets the bit rate. The bit rate is determined by a combination of these bits with the SPBR[7:0] bits. The SPBR value is used to set the base bit rate. The BRDV value is used to select a division ratio of the base bit rate from among no division, 2, 4, and 8. 00: Base bit rate 01: Base bit rate divided by 2 10: Base bit rate divided by 4 11: Base bit rate divided by 8

Bit Rate

SPBR[7:0] and BRDV[1:0] are used for setting the bit rate.
The following formula is used to calculate the bit rate when SPBR[7:0] ≠ 0.

$$\text{Bit rate} = B\phi / (2 \times n \times 2^N)$$

n: SPBR[7:0] setting (1, …, 255)

N: BRDV[1:0] setting (0 to 3)

17.3.5 SFMAnDRCR — SFMAn Data Read Control Register

SFMAnDRCR is a 32-bit register that sets the operation in external address space read mode.

The bits except the SSLN bit should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SSLN	—	—	—	—	RBURST[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RCF	RBE	—	—	—	—	—	—	—	SSLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R/W	R	R	R	R	R	R	R	R/W

Table 17.11 SFMAnDRCR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	SSLN	SPBSSL Negation Asserted SPBSSL can be negated by writing 1 to this bit when both the RBE and SSLE bits are 1. This bit is always read as 0. **NOTE:** To start next access after SPBSSL negation using this bit, read SSLF in SFMAnCMNSR = 0 to confirm that the SPBSSL has been negated.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	RBURST[3:0]	Read Data Burst Length Sets the burst length (data unit count) when reading. This bit is enabled when the RBE bit is set to 1. 0000: 1 data unit 0001: 2 continuous data units : 1110: 15 continuous data units 1111: 16 continuous data units One data unit is 64 bits long.
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	RCF	Read cache Flush When 1 is written to this bit, all the entries in the read cache are cleared. This bit is always read as 0. **NOTE:** After flushing the read cache by writing 1 to the RCF bit, read the SFMAnDRCR before proceeding to read from the external address space.
8	RBE	Read Burst Turns burst ON or OFF when reading. 0: Data is read according to the access size. 1: Read cache is enabled, and as many data units as the burst count specified in RBURST[3:0] bits is read.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.11 SFMAnDRCR Register Contents (2/2)

Bit Position	Bit Name	Function
0	SSLE	SPBSSL Negation Sets the conditions for SPBSSL negation during read burst. SPBSSL is negated for each access during normal read. 0: SPBSSL is negated after transfer of data set in burst length. 1: SPBSSL is negated when the accessed address is not continuous with the previously transferred address.

17.3.6 SFMAnDRCMR — SFMAn Data Read Command Setting Register

SFMAnDRCMR is a 32-bit register that sets the commands issued in external address space read mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 10_H

Value after reset: $0003\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.12 SFMAnDRCMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

17.3.7 SFMAnDREAR — SFMAn Data Read Extended Address Setting Register

SFMAnDREAR is a 32-bit register that sets the address when the serial flash address is output in 32-bit mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 14_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	EAV[5:0]						—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.13 SFMAnDREAR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 17	EAV[5:0]	32-Bit Extended Upper Address Fixed Value Sets the serial flash address [31:26] fixed values to EAV[5:0]. This setting is valid when the ADE[3] bit in SFMAnDRENR is 1.
16 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

17.3.8 SFMAnDROPR — SFMAn Data Read Option Setting Register

SFMAnDROPR is a 32-bit register that sets the option data in external address space read mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.14 SFMAnDROPR Register Contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.3.9 SFMAnDRENR — SFMAn Data Read Enable Setting Register

SFMAnDRENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and read data in external address space read mode and enables outputting them other than read data.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + $1C_H$

Value after reset: $0000\ 4700_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		—	—	ADB[1:0]		—	—	OPDB[1:0]		—	—	DRDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	ADE[3:0]				OPDE[3:0]				—	—	—	—
Value after reset	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 17.15 SFMAnDRENR Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	CDB[1:0]	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	ADB[1:0]	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	OPDB[1:0]	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 17.15 SFMAnDRENR Register Contents (2/2)

Bit Position	Bit Name	Function
17, 16	DRDB[1:0]	Data Read Bit Size Sets the data read size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	DME	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. **NOTE:** A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	Address Enable Sets the address to be output. Be sure to use the following settings; otherwise, the operation is not guaranteed. 0000: Output disabled 0111: Address[23:0] 1111: Address[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

17.3.10 SFMAnSMCR — SFMAn SPI Mode Control Register

SFMAnSMCR is a 32-bit register that sets the operation in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SSLKP	—	—	—	—	—	SPIRE	SPIWE	SPIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	W

Table 17.16 SFMAnSMCR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SSLKP	SPBSSL Signal Level Determines the SPBSSL status after the end of transfer. 0: SPBSSL signal is negated at the end of transfer. 1: SPBSSL signal level is maintained from the end of transfer to the start of next access.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	SPIRE	Data Read Enable Enables reading in SPI operating mode. 0: Data reading disabled 1: Data reading enabled **NOTE:** When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
1	SPIWE	Data Write Enable Enables writing in SPI operating mode. 0: Data writing disabled 1: Data writing enabled **NOTE:** When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
0	SPIE	SPI Data Transfer Enable Data is transferred by setting this bit to 1. This bit is enabled only when the TEND bit in SFMAnCMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0. This bit is always read as 0. **NOTE:** When the SPBSSL pin is de-asserted, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0. When the SPBSSL pin is asserted, follow the notes described in **Section 17.5.2, Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode**.

17.3.11 SFMAnSMCMR — SFMAn SPI Mode Command Setting Register

SFMAnSMCMR is a 32-bit register that sets the commands issued in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 24_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.17 SFMAnSMCMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

17.3.12 SFMAnSMADR — SFMAn SPI Mode Address Setting Register

SFMAnSMADR is a 32-bit register that sets the addresses in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 28_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:24]								ADR[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.18 SFMAnSMADR Register Contents

Bit Position	Bit Name	Function
31 to 24	ADR[31:24]	Address Sets the value of bits 31 to 24 when the serial flash address is output in 32-bit units. This setting is valid when ADE[3] in SFMAnSMENR is 1.
23 to 0	ADR[23:0]	Address Sets the address.

17.3.13 SFMAnSMOPR — SFMAn SPI Mode Option Setting Register

SFMAnSMOPR is a 32-bit register that sets the option data in SPI operating mode.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + $2C_H$

Value after reset: $0000\ 0000_H$

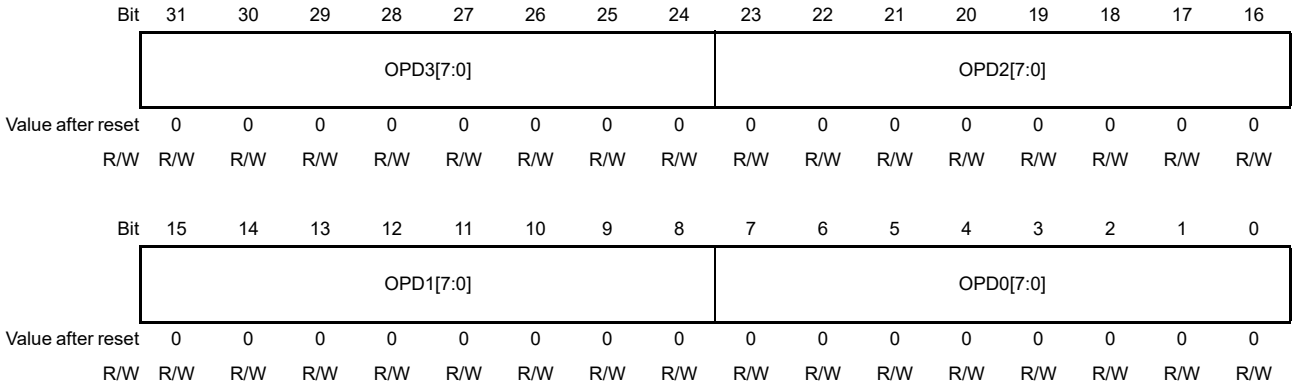


Table 17.19 SFMAnSMOPR Register Contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.3.14 SFMAnSMENR — SFMAn SPI Mode Enable Setting Register

SFMAnSMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in SPI operating mode and enables their output. SFMAnSMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 30_H

Value after reset: 0000 4000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		—	—	ADB[1:0]		—	—	OPDB[1:0]		—	—	SPIDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	ADE[3:0]				OPDE[3:0]				SPIDE[3:0]			
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.20 SFMAnSMENR Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	CDB[1:0]	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	ADB[1:0]	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	OPDB[1:0]	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited

Table 17.20 SFMAnSMENR Register Contents (2/2)

Bit Position	Bit Name	Function
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	SPIDB[1:0]	Transfer Data Bit Size Sets the transfer data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15	DME	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. **NOTE:** Dummy cycle insertion is prohibited for write in SPI operating mode including the case in which a transfer ends with a dummy cycle. **NOTE:** A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	Address Enable Sets the address to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 0100: ADR[23:16] 0110: ADR[23:8] 0111: ADR[23:0] 1111: ADR[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	SPIDE[3:0]	Transfer Data Enable Sets valid transfer data. The following settings must be used. Otherwise, the operation is not guaranteed. 0000: Not transferred 1000: 8 bits transferred (enables data at address 0 of the SPI mode read/write data register) 1100: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data register) 1111: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data register) Other than above: Setting prohibited

17.3.15 SFMAnSMRDR — SFMAn SPI Mode Read Data Register

SFMAnSMRDR is a 32-bit register that stores the read data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SFMAnSMENR). Be sure to access from address 0.

The settings of this register should be read when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: SFMAnSMRDR is a read-only register that can be read in 32-bit units.
SFMAnSMRDRL, SFMAnSMRDRH are read-only registers that can be read in 16-bit units.
SFMAnSMRDRLL, SFMAnSMRDRLH, SFMAnSMRDRHL, SFMAnSMRDRHH are read-only registers that can be read in 8-bit units.

Address: SFMAnSMRDR: <SFMAn_base> + 38_H
SFMAnSMRDRL: <SFMAn_base> + 38_H
SFMAnSMRDRLL: <SFMAn_base> + 38_H
SFMAnSMRDRLH: <SFMAn_base> + 39_H
SFMAnSMRDRH: <SFMAn_base> + $3A_H$
SFMAnSMRDRHL: <SFMAn_base> + $3A_H$
SFMAnSMRDRHH: <SFMAn_base> + $3B_H$

Value after reset: XXXX $XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDATA[31:16]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDATA[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.21 SFMAnSMRDR Register Contents

Bit Position	Bit Name	Function
31 to 0	RDATA[31:0]	Read Data Holds the data read in SPI operating mode.

Note: The contents of this register are modified upon completion of reception in SPI operating mode. Be sure to read data when reception in SPI operating mode is completed.

17.3.16 SFMAnSMWDR — SFMAn SPI Mode Write Data Register

SFMAnSMWDR is a 32-bit register that sets the write data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SFMAnSMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: SFMAnSMWDR can be read or written in 32-bit units.
SFMAnSMWDRL, SFMAnSMWDRH can be read or written in 16-bit units.
SFMAnSMWDRLL, SFMAnSMWDRLH, SFMAnSMWDRHL, SFMAnSMWDRHH can be read or written in 8-bit units.

Address: SFMAnSMWDR: <SFMAn_base> + 40_H
SFMAnSMWDRL: <SFMAn_base> + 40_H
SFMAnSMWDRLL: <SFMAn_base> + 40_H
SFMAnSMWDRLH: <SFMAn_base> + 41_H
SFMAnSMWDRH: <SFMAn_base> + 42_H
SFMAnSMWDRHL: <SFMAn_base> + 42_H
SFMAnSMWDRHH: <SFMAn_base> + 43_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.22 SFMAnSMWDR Register Contents

Bit Position	Bit Name	Function
31 to 0	WDATA[31:0]	Write Data Holds the data to be written in SPI operating mode.

17.3.17 SFMAnCMNSR — SFMAn Common Status Register

SFMAnCMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <SFMAn_base> + 48_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSLF	TEND
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 17.23 SFMAnCMNSR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SSLF	SPBSSL Pin Monitor 0: SPBSSL pin is negated 1: SPBSSL pin is asserted
0	TEND	Transfer End Flag Indicates whether the data transfer has ended. 0: Indicates that data transfer is in progress 1: Indicates that data transfer has ended

17.3.18 SFMAnDRDMCR — SFMAn Data Read Dummy Cycle Setting Register

SFMAnDRDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in external address space read mode.

The settings of this register are enabled when the DME bit in the data read enable setting register (SFMAnDRENR) is 1.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 58_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 17.24 SFMAnDRDMCR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DMDB[1:0]	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (SFMAnCMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMCYC[2:0]	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (SFMAnDRENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.3.19 SFMAnSMDMCR — SFMAn SPI Mode Dummy Cycle Setting Register

SFMAnSMDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in SPI operating mode.

The settings of this register are enabled when the DME bit in the SPI mode enable setting register (SFMAnSMENR) is 1.

The settings of this register should be changed when the TEND flag in SFMAnCMNSR is 1; otherwise, the operation cannot be guaranteed.

Access: This register can be read or written in 32-bit units.

Address: <SFMAn_base> + 60_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMDB[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 17.25 SFMAnSMDMCR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DMDB[1:0]	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV, IO2FV, and IO3FV bits in the common control register (SFMAnCMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DMCYC[2:0]	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the SPI mode enable setting register (SFMAnSMENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.4 Operation

17.4.1 System Configuration

With this module, one serial flash memory can be directly connected (data size of 1, 2, and 4 bits).

The example of system configuration is shown in **Figure 17.2**.

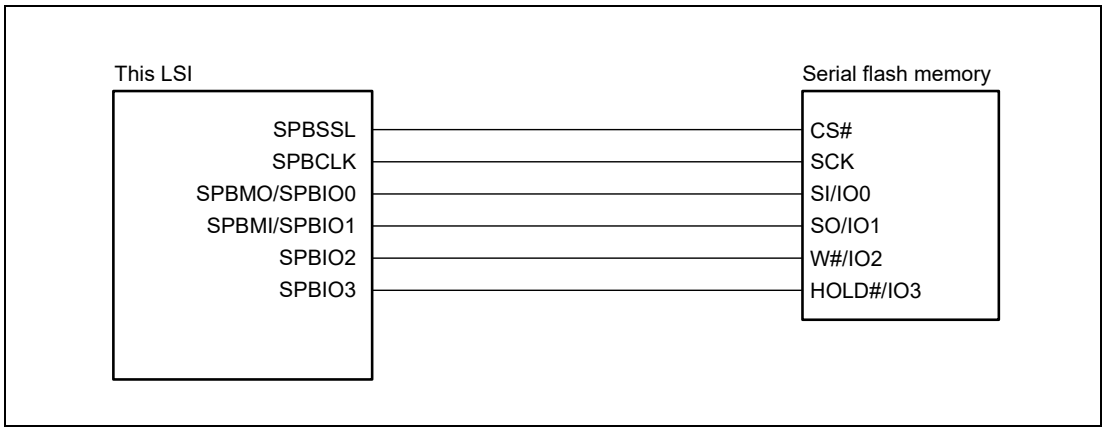


Figure 17.2 System Configuration Example with 4-Bit Data Size

17.4.2 Address Map

In external address space read mode, the serial flash connected is assigned in the SPI multi-I/O bus space. In combination with SFMAnDREAR, a maximum of 4 Gbytes can be accessed.

Table 17.26 Address Map

Internal Address	Max. Access Area
3800 0000_H to 3BFF $FFFF_H$	4 Gbytes

17.4.3 32-bit Serial Flash Addresses

Since the SPI multi-I/O bus space is 64 Mbytes, only a part of the 32-bit serial flash address area can be directly accessed. Here, the fixed value set in the pertinent register is used as the upper bit value of a 32-bit address.

To output serial flash addresses in 32 bits, set the ADE[3] bit in SFMAnDRENR to 1, and set the serial flash address[31:26] to the value of EAV[5:0] in SFMAnDREAR as a fixed value.

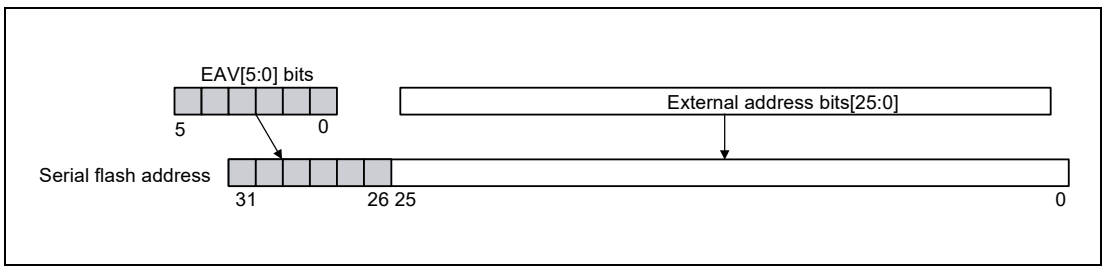


Figure 17.3 32-Bit Address Setting

17.4.4 Operating Modes

This module has two operating modes: external address space read mode and SPI operating mode.

In external address space read mode, a read access to the SPI multi-I/O bus space is converted into SPI communication and data is received. After data acquisition, data is returned to the bus master that is the issuing source. For details, see **Section 17.4.5, External Address Space Read Mode**.

In SPI operating mode, arbitrary SPI communication is carried out using register settings. For details, see **Section 17.4.8, SPI Operating Mode**.

17.4.5 External Address Space Read Mode

A read access to the SPI multi-I/O bus space can be converted into SPI communication in external address space read mode. Further, the commands, optional commands, option data, and dummy cycle issued for reading can be modified using registers.

In external address space read mode, either normal read operation or burst read operation can be selected. The transfer format is determined based on the common control register (SFMAnCMNCR), SSL delay register (SFMAnSSLDR), bit rate setting register (SFMAnSPBCR), data read control register (SFMAnDRCR), data read command setting register (SFMAnDRCMR), data read extended address setting register (SFMAnDREAR), data read option setting register (SFMAnDROPR), data read enable setting register (SFMAnDRENR), and data read dummy cycle setting register (SFMAnDRDMCR).

17.4.5.1 Normal Read Operation

When the RBE bit in SFMAnDRCR is set to 0, normal read operation is performed.

In the normal read operation, the data of 8 bits, 16 bits, and 32 bits are read for respectively a byte, a half-word, and a word read access. After reading, the SPBSSL pin is negated.

The normal read operation timing is shown in **Figure 17.4**.

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay), t2 is the time period from transmission of the last SPBCLK edge of a transfer to SPBSSL pin negation (SPBSSL negation delay), and t3 is the time period from one transfer end to the next transfer start (next access). For details of t1, t2, and t3, see **Section 17.4.9, Transfer Format**.

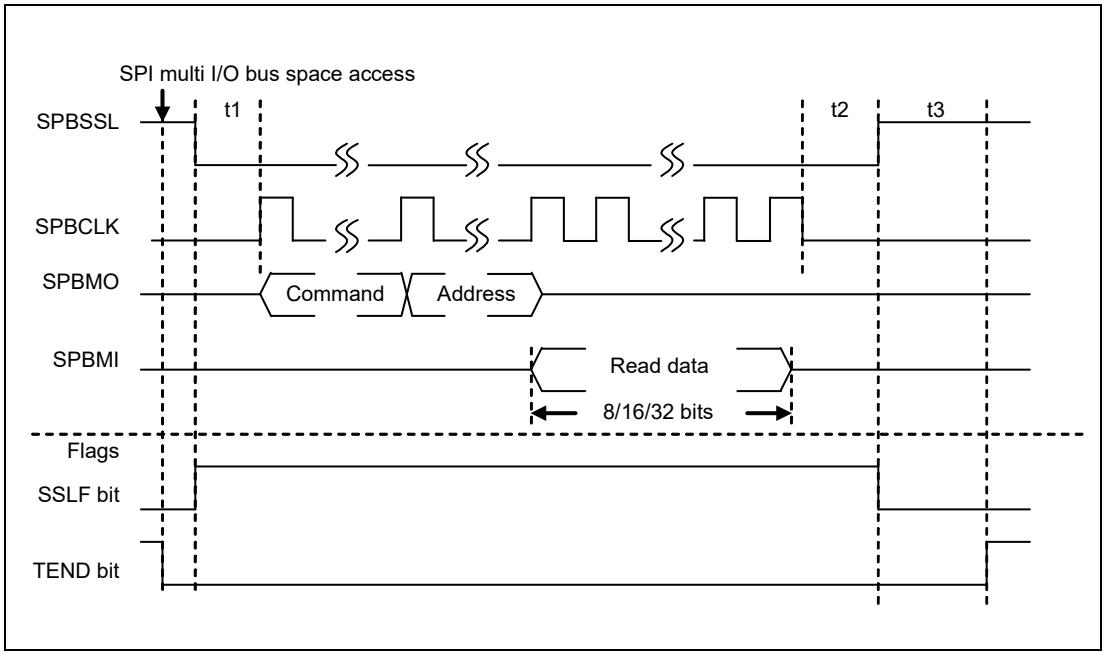


Figure 17.4 Normal Read Operation Timing

17.4.5.2 Burst Read Operation

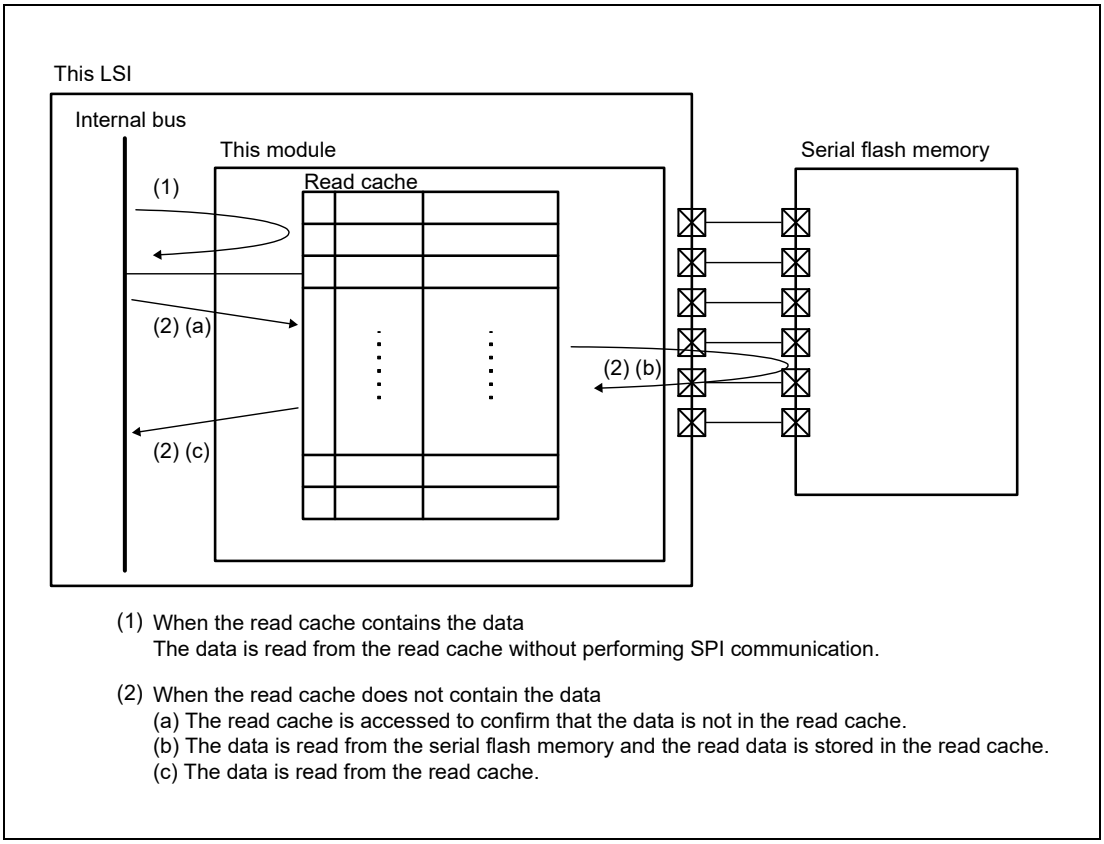
When the RBE bit in SFMAnDRCR is set to 1, burst read operation is performed.

Read cache is enabled in the burst read operation. For read cache operation, see **Section 17.4.7, Read Cache**.

For reading bytes, words, or longwords, the read cache is first referred to for the data. When the read cache contains the data, the data is read from the read cache without accessing the serial flash memory. When the read cache does not contain the data, burst read operation is performed in the serial flash memory and the read data is stored in the read cache. The data transfer length at that time is 64 bits × RBURST[3:0] bits and the data is always read from the 64-bit boundary.

The SPBSSL pin status after data transfer can be selected by using the SSLE bit in SFMAnDRCR. When the SSLE bit is set to 0, the SPBSSL pin is negated after data transfer. For an operation performed when the SSLE bit is set to 1, see **Section 17.4.5.3, Burst Read Operation with Automatic SPBSSL Negation**, just below.

A pattern diagram of this operation and a burst read operation timing diagram when SSLE bit is set to 0 are shown in **Figure 17.5** and **Figure 17.6**.



(1) When the read cache contains the data
The data is read from the read cache without performing SPI communication.

(2) When the read cache does not contain the data
(a) The read cache is accessed to confirm that the data is not in the read cache.
(b) The data is read from the serial flash memory and the read data is stored in the read cache.
(c) The data is read from the read cache.

Figure 17.5 Burst Read Operation

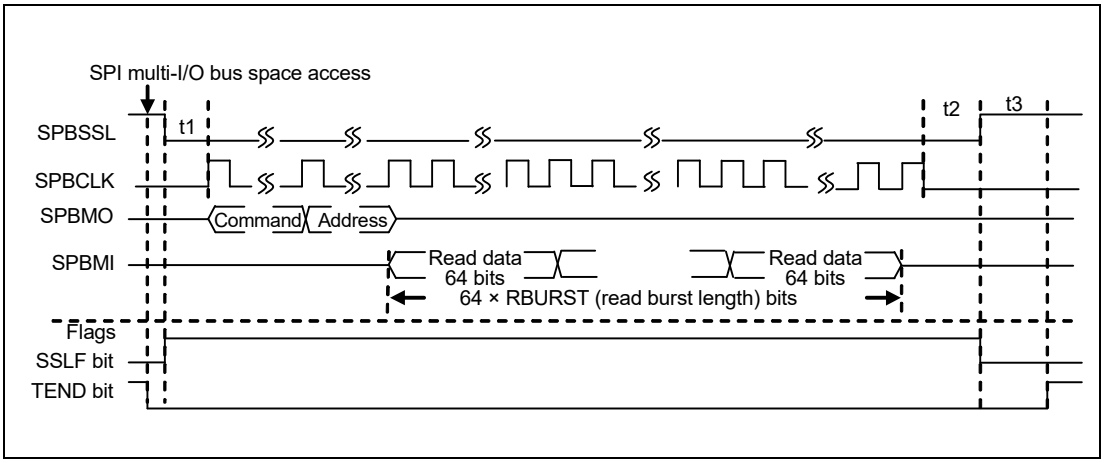


Figure 17.6 Burst Read Operation Timing (SSLE Bit = 0)

17.4.5.3 Burst Read Operation with Automatic SPBSSL Negation

When SSLE bit in SFMAnDRCR is set to 1, this module does not negate the SPBSSL pin after the burst read transfer. When accessing the next time, if the address is continuous with the previous read address, the burst read operation is performed without issuing the command, optional command, address, option data, or dummy cycle. If the address is not continuous with the previous read address, the SPBSSL pin is once negated and the burst read operation is performed after issuing the command, optional command, address, option data, or dummy cycle.

Burst read timing diagrams for continuous address and non-continuous address are shown in **Figure 17.7** and **Figure 17.8**.

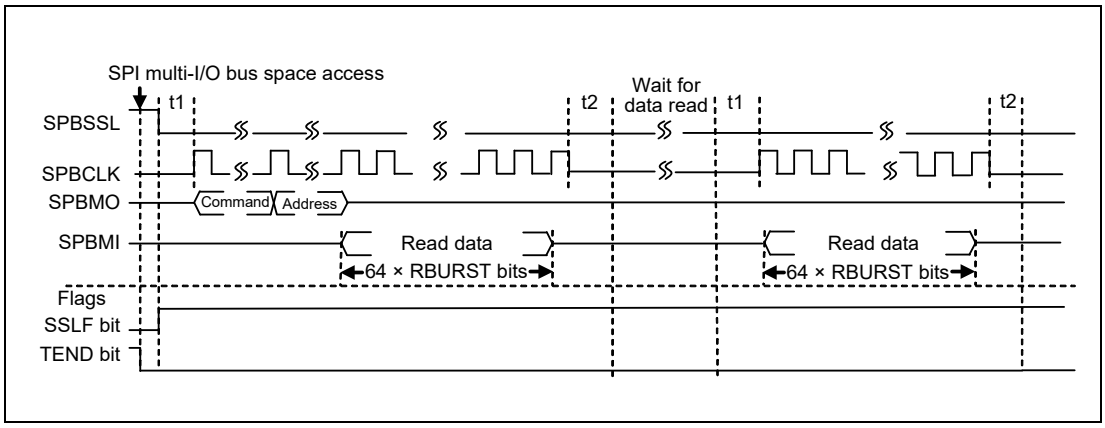


Figure 17.7 Burst Read Timing for Continuous Address (SSLE Bit = 1)

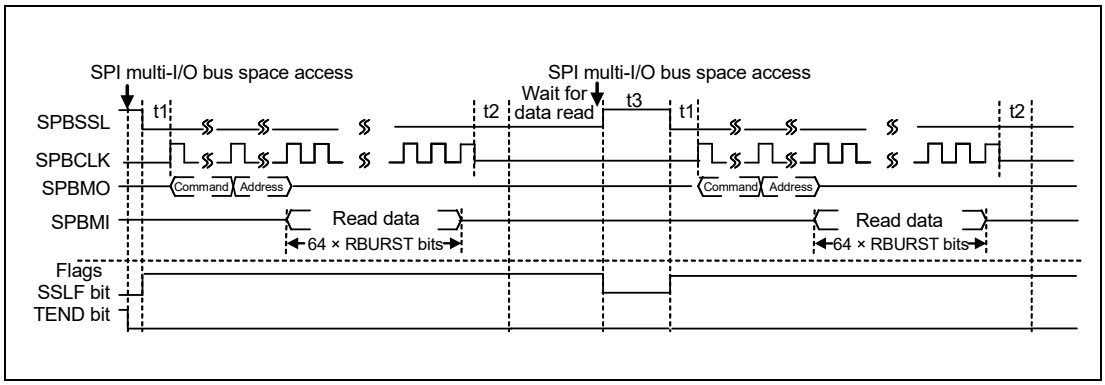


Figure 17.8 Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)

For the next access after negation of the SPBSSL with the SSLN bit in SFMAnDRCR with this operation, read SSLF = 0 in SFMAnCMNSR to confirm that the SPBSSL has been negated.

17.4.6 Initial Setting Flow

An example of an initial setting flow in external address space read mode is shown in **Figure 17.9**.

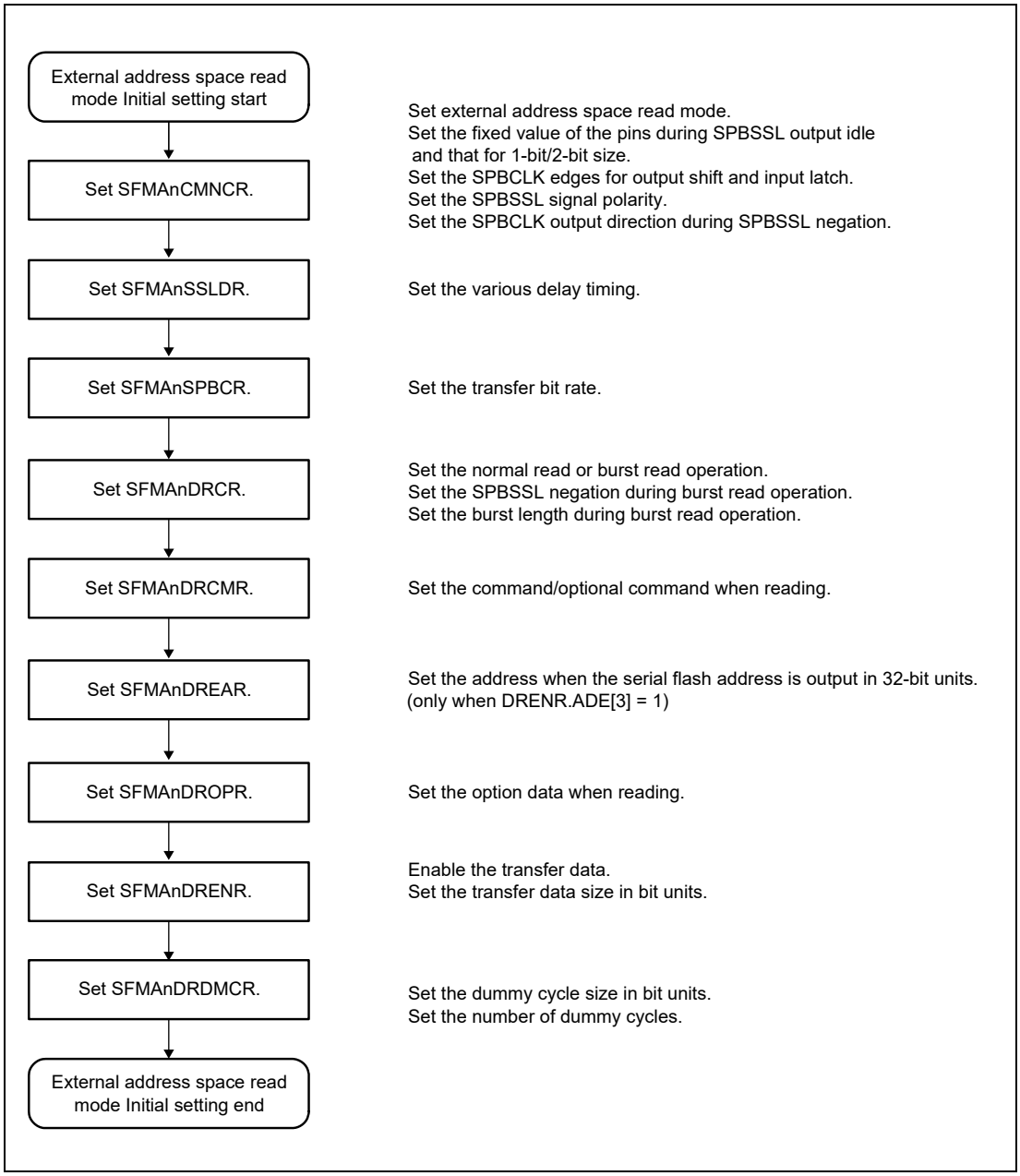


Figure 17.9 Example of Initial Setting Flow in External Address Space Read Mode

17.4.7 Read Cache

This module has a simple built-in read cache. The read cache can be used during external address space read mode and burst read operation. The read cache is configured with a line size of 64 bits and 16 entries.

Read cache configuration is shown in **Figure 17.10**.

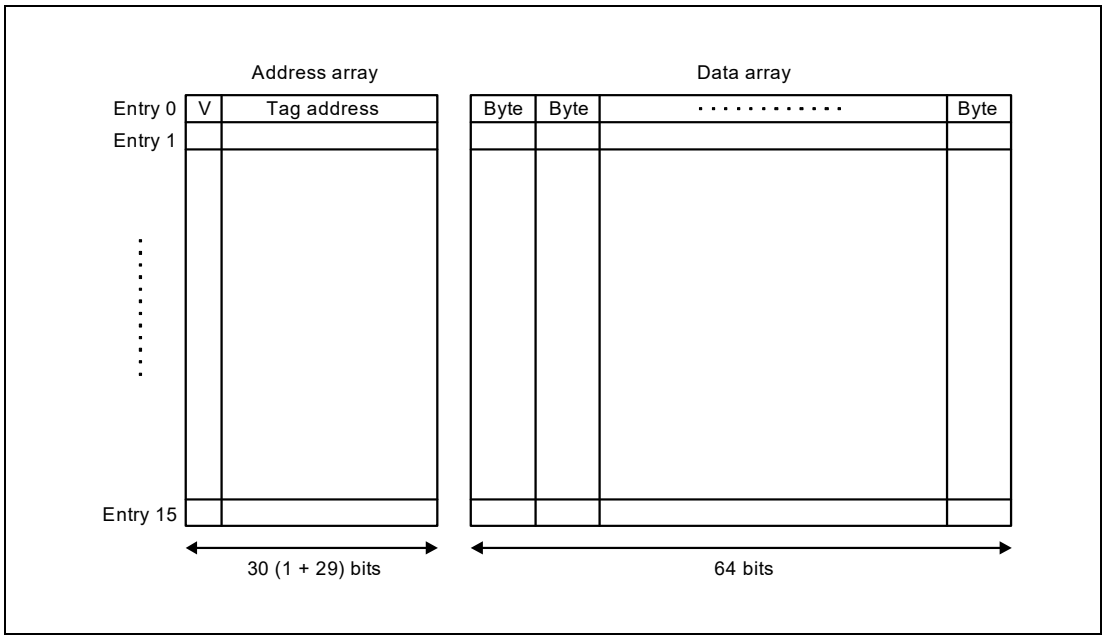


Figure 17.10 Read Cache Configuration

17.4.7.1 Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.

The tag address bits hold the address used for the serial flash memory. Address bits 31 to 3 are used for the tag address.

Address bits 23 to 3 are enabled when address output is 24 bits.
Address bits 31 to 3 are enabled when address output is 32 bits.

17.4.7.2 Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

17.4.7.3 Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 × RBURST (read burst length) data is read from the serial flash memory and the read cache is updated, the data is returned to the bus master.

17.4.7.4 Data Replacement

The write pointer is used to update data. In case of read-miss, the RBURST (read burst length) portion data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data. Whether data is referred to or not will not affect the replacement order of data.

17.4.8 SPI Operating Mode

This module can carry out an arbitrary SPI operation by using the register settings.

The transfer format is determined based on the common control register (SFMAnCMNCR), SSL delay register (SFMAnSSLDR), bit rate setting register (SFMAnSPBCR), SPI mode control register (SFMAnSMCR), SPI mode command setting register (SFMAnSMCMR), SPI mode address setting register (SFMAnSMADR), SPI mode option setting register (SFMAnSMOPR), and SPI mode enable setting register (SFMAnSMENR), SPI mode read data register (SFMAnSMRDR), SPI mode write data register (SFMAnSMWDR), and SPI mode dummy cycle setting register (SFMAnSMDMCR).

SPI operating mode can be used for reading the status of the serial flash memory and writing to the serial flash memory.

In this mode, one transfer refers to the operation from when the SPIE bit in SFMAnSMCR is set to 1 when the TEND bit in SFMAnCMNSR is set to 1.

17.4.8.1 Transfer Start

The transfer of data is started in the set transfer format by setting the SPIE bit in SFMAnSMCR to 1. When write operation is enabled, the SPI mode write data register is transmitted to the serial flash memory. When read operation is enabled, data read from the serial flash memory is stored into the SPI mode read data register.

The SPI operation timing is shown in **Figure 17.11**.

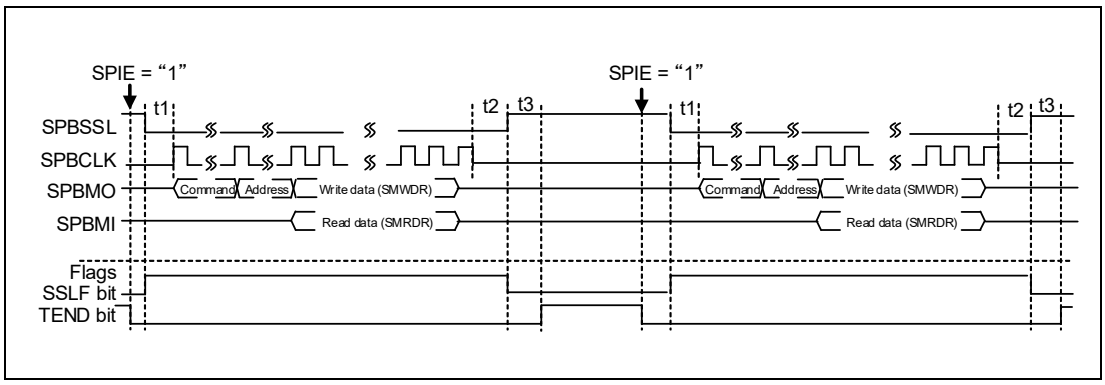


Figure 17.11 SPI Operation Timing

17.4.8.2 Read/Write Enable

- Read operation: Data can be read by setting the SPIRE bit in SFMAnSMCR to 1. The read data is stored into SFMAnSMRDR.
- Write operation: Data can be written by setting the SPIWE bit in SFMAnSMCR to 1. The data stored in SFMAnSMWDR is output.

When the data size is set to 1 bit using the SPIDB[1:0] bits in SFMAnSMENR, data can be transmitted and received by setting the SPIRE and SPIWE bits to 1. However, when the data size is set to 2 or 4 bits by using the SPIDB[1:0] bits, only one of the SPIRE and SPIWE bits should be enabled. The operation is not guaranteed if both the bits are enabled.

17.4.8.3 Retention of SPBSSL Pin Assertion

By setting the SSLKP bit in SFMAnSMCR to 1, assertion of the SPBSSL pin can be continued till the next transfer. With this function, the transfer can be carried out continuously with the SPBSSL kept in the asserted state.

The data transfer timing using the SSLKP bit is shown in **Figure 17.12**.

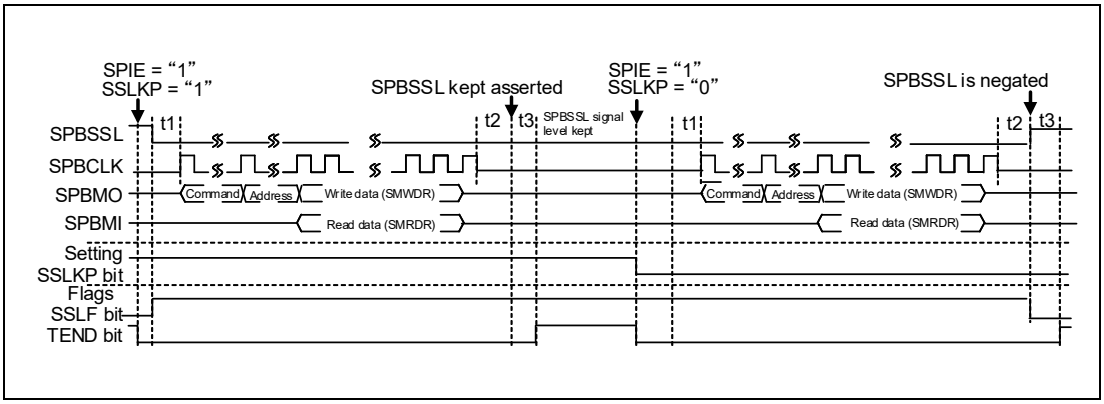


Figure 17.12 Data Transfer Timing using the SSLKP Bit

17.4.8.4 Initial Setting Flow

An example of an initial setting flow in SPI operating mode is shown in **Figure 17.13**.

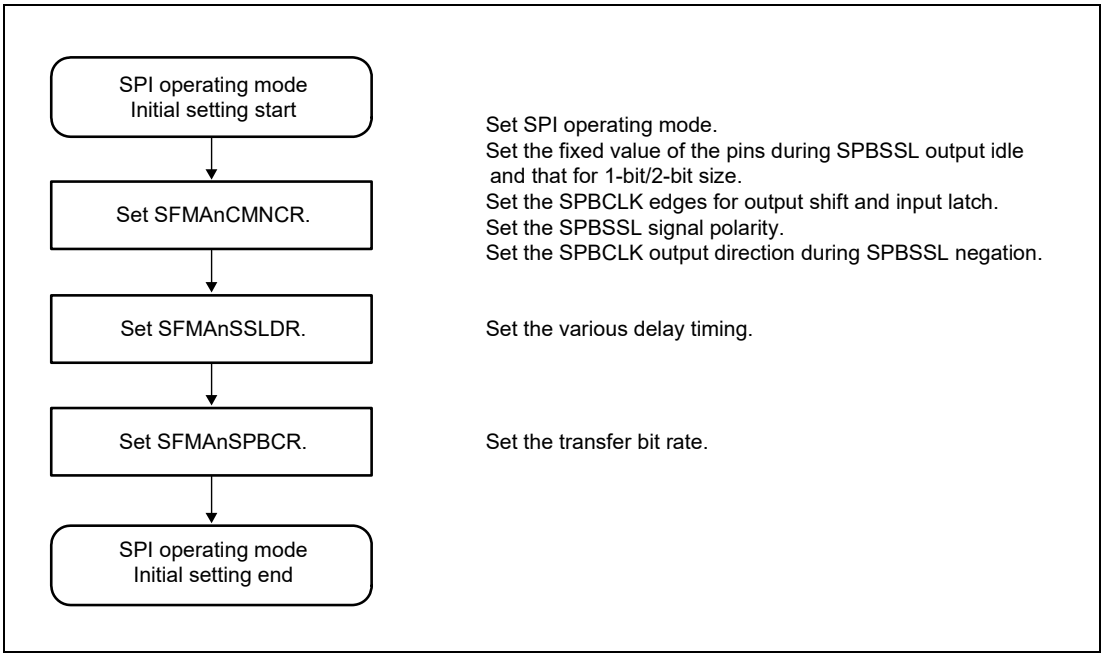


Figure 17.13 Example of Initial Setting Flow in SPI Operating Mode

17.4.8.5 Data Transfer Setting Flow

An example of a data transfer setting flow in SPI operating mode is shown in **Figure 17.14**.

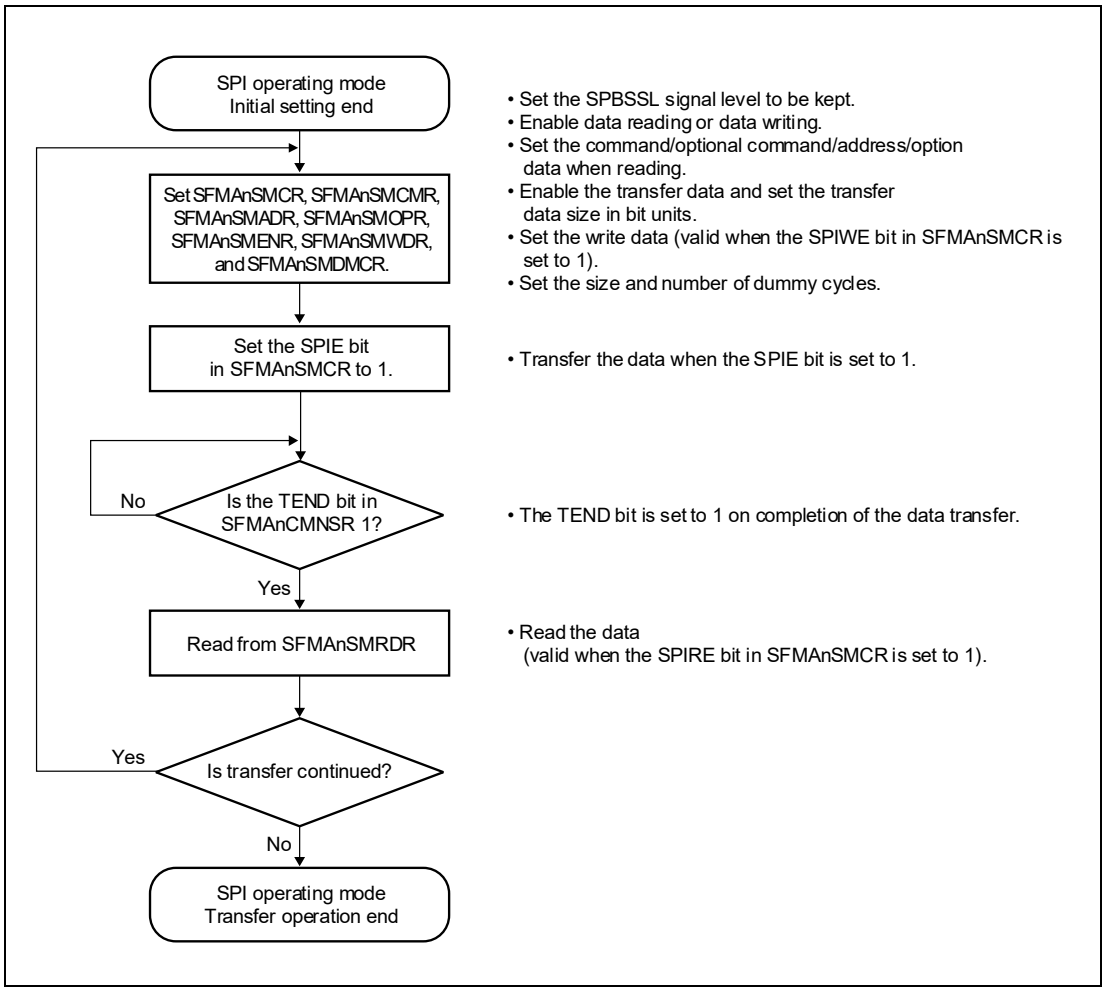


Figure 17.14 Example of a Data Transfer Setting Flow in SPI Operating Mode

17.4.9 Transfer Format

17.4.9.1 SPBSSL Pin Enable Polarity Control

The enable polarity of the SPBSSL pin can be changed with the SSLP bit in SFMAnCMNCR.

17.4.9.2 SPBCLK Output

The SPBCLK output direction during SPBSSL negation can be set with the CPOL bit in SFMAnCMNCR.

17.4.9.3 Data Transmission and Reception Timing

Data is transmitted and received at either the odd or even edges. The data transmission timing can be set to the odd or even edge with the CPHAT bit in SFMAnCMNCR. Similarly, the data reception timing can be set to the odd or even edge with the CPHAR bit in SFMAnCMNCR.

17.4.9.4 Delay Settings

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). It can be set with the SCKDL[2:0] bits in SFMAnSSLDR. t2 is the time period till the SPBSSL signal negation after the SPBCLK oscillation is stopped (SPBSSL negation delay). It can be set with the SLNDL[2:0] bits in SFMAnSSLDR. t3 is the time period required to prevent SPBSSL signal assertion for the next transfer after the end of the previous transfer (next access delay). It can be set with the SPNDL[2:0] bits in SFMAnSSLDR.

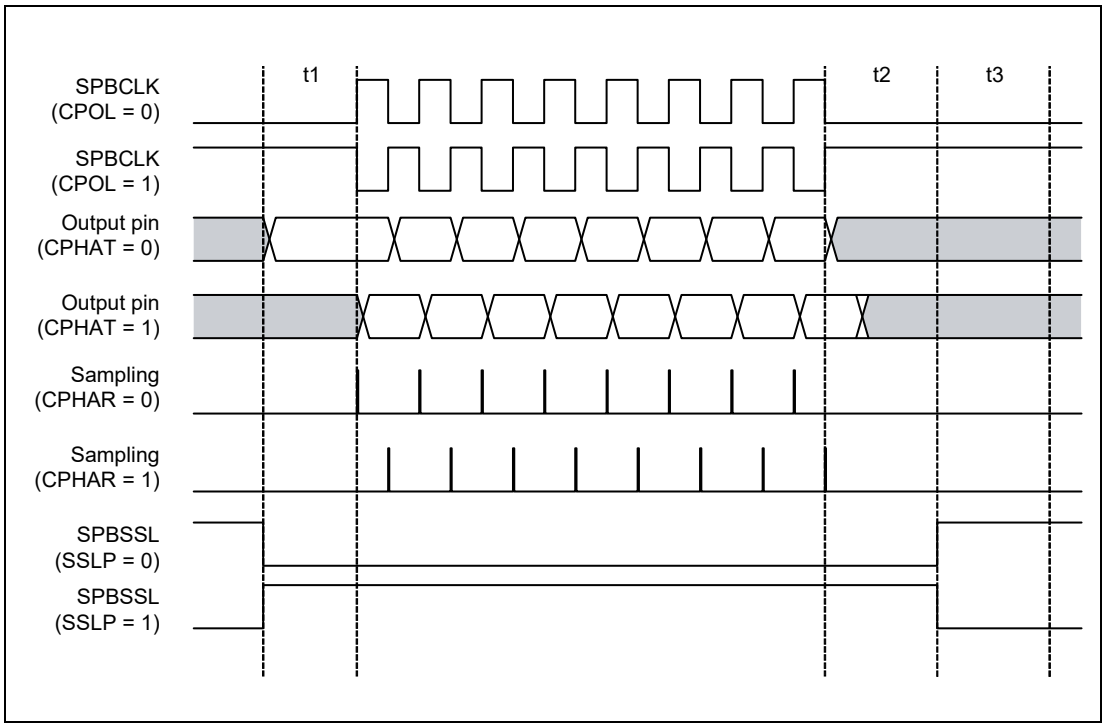


Figure 17.15 Transfer Format

17.4.10 Data Format

This module can input and output data in the order of command, optional command, address, option data, dummy cycle and data.

17.4.10.1 Data Registers

Table 17.27 shows the input and output data.

Table 17.27 Data Registers

Data	External Address Space Read Operation	SPI Operation
Command (8 bits)	CMD[7:0] bits in SFMAnDRCMR	CMD[7:0] bits in SFMAnSMCMR
Optional command (8 bits)	OCMD[7:0] bits in SFMAnDRCMR	OCMD[7:0] bits in SFMAnSMCMR
Address (32/24 bits)	32 bits: SFMAnDREAR.EAV[5:0] + lower [25:0] bits of the read address. 24 bits: Lower [23:0] bits of the read address	32 bits: ADR[31:0] bits in SFMAnSMADR 24 bits: ADR[23:0] bits in SFMAnSMADR
Option data (8 bits × 4)	SFMAnDROPR	SFMAnSMOPR
Dummy cycle (1 to 8 cycles)	SFMAnDRDMCR	SFMAnSMDMCR (only when read)
Transfer data	Normal read: 8, 16, and 32 bits Burst read: 64 × RBURST bits	Read: SFMAnSMRDR Write: SFMAnSMWDR

17.4.10.2 Data Enable

In external address space read mode, transfer enable or disable of the command, optional command, address, option data, and dummy cycle can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], and DME bits in SFMAnDRENR, respectively. The size and number of dummy cycles can be controlled with the data read dummy cycle setting register (SFMAnDRDMCR).

Similarly, in SPI operating mode, enable or disable of the command, optional command, address, option data, dummy cycle, and transfer data can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], DME, and SPIDE[3:0] bits in SFMAnSMENR, respectively. However, disabling all the above parameters is prohibited in SPI operating mode. At least one of them except dummy cycle must be enabled. The size and number of dummy cycles can be controlled with the SPI mode dummy cycle setting register (SFMAnSMDMCR).

For the address and option data in external address space read mode; and the address, option data, and transfer data in SPI operating mode, the enable bit setting allowed is determined according to the transfer data size. For the allowed setting combinations of the enable bits and transfer data size, refer to the description of the pertinent registers.

If data is disabled, that data is skipped, and input and output of the next data is carried out. The command, optional command, address, and option data are always output. During dummy cycles, the state of the used pins is Hi-Z. In external address space read mode, data is always input; and in SPI operating mode, input and output of data is determined based on the settings of the SPIRE and SPIWE bits in SFMAnSMCR.

There are some restrictions on dummy cycle insertion; refer to the description of the DME bits in SFMAnDRENR and SFMAnSMENR for details.

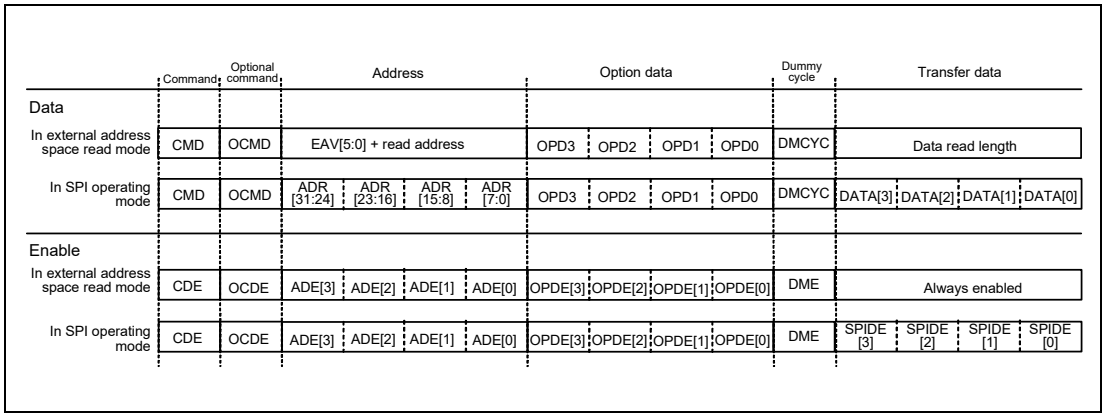


Figure 17.16 Data and Enable

17.4.10.3 Bit Size

In external address space read mode, the size of the command, optional command, address, option data, and the read data in bit units is respectively controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and DRDB[1:0] bits in SFMAnDRENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SFMAnDRDMCR.

Similarly, in SPI operating mode, the size of the command, optional command, address, option data, and read write data in bit units is controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and SPIDB[1:0] bits in SFMAnSMENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SFMAnSMDMCR.

(1) 1-bit Size

When the size is set to 1 bit, SPBMI pin will be the input pin and SPBMO pin will be the output pin. SPBIO2 and SPBIO3 pins are not used.

Figure 17.17 shows the transfer format example.

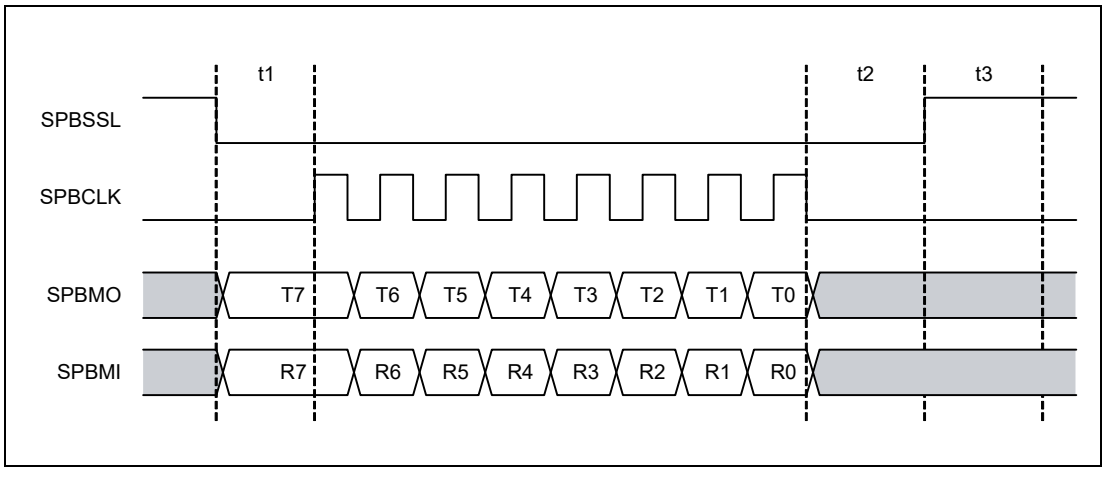


Figure 17.17 Transfer Format Example with 1-Bit Data Size

(2) 2-bit Size

When the size is set to 2 bits, SPBIO0 and SPBIO1 pins will be either the input pins or the output pins. SPBIO2 and SPBIO3 pins are not used.

Figure 17.18 shows the transfer format example.

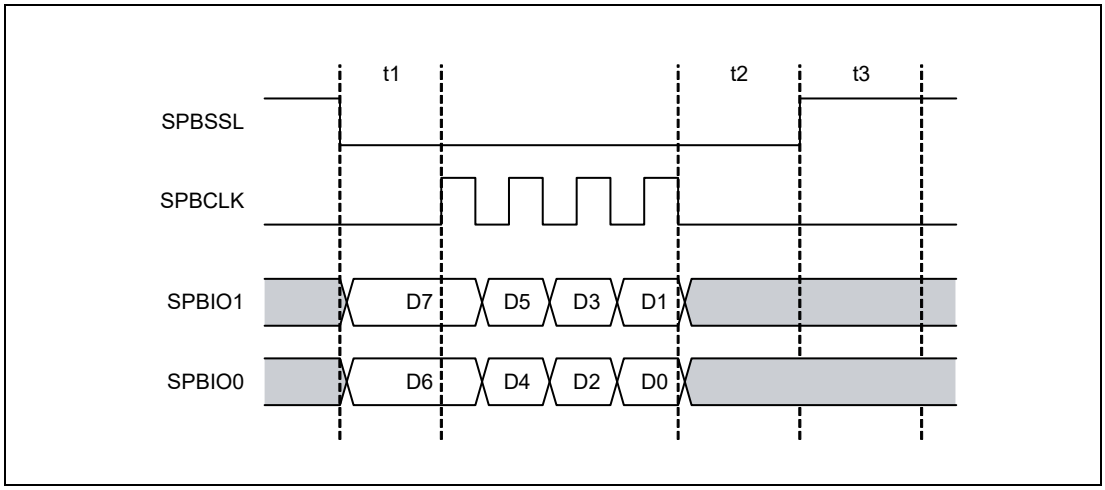


Figure 17.18 Transfer Format Example with 2-Bit Data Size

(3) 4-bit Size

When the size is set to 4 bits, SPBIO0, SPBIO1, SPBIO2, and SPBIO3 pins will be either the input pins or the output pins.

Figure 17.19 shows the transfer format example.

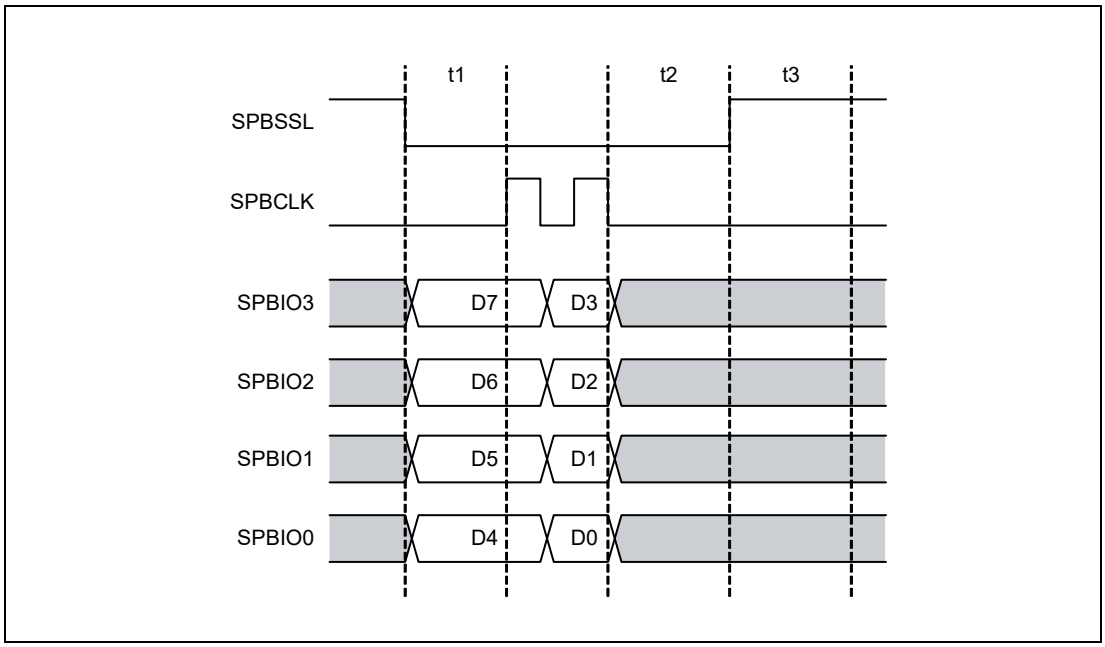


Figure 17.19 Transfer Format Example with 4-Bit Data Size

17.4.11 Data Pin Control

With this module, the status of pins can be automatically changed based on the data size to be used and the read/write settings. The pin status during the SPBSSL negation can be set with the MOIIO3, MOIIO2, MOIIO1, and MOIIO0 bits in SFMAnCMNCR. The SPBSSL and SPBCLK pins are always output pins. The status of respective pins is specified in **Table 17.28** to **Table 17.31**.

Table 17.28 Pin Status (1)

Pin	SPBSSL Negation	SPBSSL Assertion		
		Command, Optional Command, Address, Option Data		
		1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	MOIIO0 bit value	Output	Output	Output
SPBMI/SPBIO1	MOIIO1 bit value	Hi-Z	Output	Output
SPBIO2	MOIIO2 bit value	IO2FV bit value	IO2FV bit value	Output
SPBIO3	MOIIO3 bit value	IO3FV bit value	IO3FV bit value	Output

Table 17.29 Pin Status (2)

Pin	Transfer Data					
	External Address Space Read Operation			SPI Operation		
				SPIRE Bit = 1, SPIWE Bit = 0		
	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	IO0FV bit value	Input	Input	IO0FV bit value	Input	Input
SPBMI/SPBIO1	Input	Input	Input	Input	Input	Input
SPBIO2	IO2FV bit value	IO2FV bit value	Input	IO2FV bit value	IO2FV bit value	Input
SPBIO3	IO3FV bit value	IO3FV bit value	Input	IO3FV bit value	IO3FV bit value	Input

Table 17.30 Pin Status (3)

Pin	Transfer Data					
	SPI Operation					
	SPIRE Bit = 0, SPIWE Bit = 1			SPIRE Bit = 1, SPIWE Bit = 1		
	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	Output	Output	Output	Output	Setting prohibited	Setting prohibited
SPBMI/SPBIO1	Hi-Z	Output	Output	Input	Setting prohibited	Setting prohibited
SPBIO2	IO2FV bit value	IO2FV bit value	Output	IO2FV bit value	Setting prohibited	Setting prohibited
SPBIO3	IO3FV bit value	IO3FV bit value	Output	IO3FV bit value	Setting prohibited	Setting prohibited

Table 17.31 Pin Status (4)

Pin	Dummy Cycle		
	1-bit Size	2-bit Size	4-bit Size
SPBMO/SPBIO0	IO0FV bit value	Hi-Z	Hi-Z
SPBMI/SPBIO1	Hi-Z	Hi-Z	Hi-Z
SPBIO2	IO2FV bit value	IO2FV bit value	Hi-Z
SPBIO3	IO3FV bit value	IO3FV bit value	Hi-Z

17.4.12 SPBSSL Pin Control

Negation conditions of the SPBSSL pin are as follows.

17.4.12.1 External Address Space Read Mode

(1) Normal read operation (RBE bit in SFMAnDRCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(2) Burst read without automatic SPBSSL negation (RBE bit in SFMAnDRCR = 1, SSLE bit in SFMAnDRCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(3) Burst read with automatic SPBSSL negation (RBE bit in SFMAnDRCR = 1, SSLE bit in SFMAnDRCR = 1)

- SPBSSL negated after t2 cycle when the read address is not continuous with the previously read address
- SPBSSL negated after the SSLN bit in SFMAnDRCR is set to 1

17.4.12.2 SPI Operating Mode

(1) SPBSSL pin assertion not retained (SSLKP bit in SFMAnSMCR = 0)
SPBSSL negated after completing the data transfer and t2 cycle.

(2) SPBSSL pin assertion retained (SSLKP bit in SFMAnSMCR = 1)
SPBSSL not negated.
When to be negated, data should be transferred after setting the SSLKP bit to 0.

17.4.13 Flags

This module has two flag bits SSLF and TEND in SFMAnCMNSR. These bits are read-only bits.

17.4.13.1 SSLF Bit

This bit indicates the SPBSSL pin status. The status is 1 when the SPBSSL is asserted, and the status is 0 when the SPBSSL is negated.

17.4.13.2 TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.

During t1 time period, data transfer, t2 time period, t3 time period, and waiting for read access by burst read and SPBSSL automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.

When other than the above, the TEND bit is read as 1 to indicate that transfer of data has ended.

17.4.13.3 Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.

The registers which can be written to, except the SSLN bit in SFMAnDRCR, should be modified when TEND = 1. Read SFMAnSMRDR when TEND = 1. SFMAnCMNSR can always be read.

17.5 Usage Notes

17.5.1 Notes on Transfer to Read Data in SPI Operating Mode

In SPI operating mode, take note of the following points for caution when setting the SPI mode enable setting register (SFMAnSMENR) to enable transfer only for reading data.

"Transfer only for reading data" indicates transfer to read data while the CDE, OCDE, ADE[3:0], and OPDE[3:0] bits in SFMAnSMENR are all 0.

17.5.1.1 Transfer to Read Data While the Signal on the SPBSSL Pin is de-asserted

Set the SFMAnSMENR.SPIDE[3:0] bits to 1100 or 1111 when transfer only for reading data is to proceed.

Transfer will not proceed normally if the setting of the SFMAnSMENR.SPIDE[3:0] bits is 1000.

17.5.1.2 Transfer to Read Data While the Signal on the SPBSSL Pin is Asserted

When transfer only for reading data is to proceed, set the SFMAnSMENR.SPIDE[3:0] bits to 1100 or 1111, or end the immediately preceding transfer with reading data.

When the immediately preceding transfer is of a command, optional command, address, or option data, or is transfer for writing data, the subsequent transfer only for reading data will not proceed normally if the setting of the SFMAnSMENR.SPIDE[3:0] bits is 1000.

17.5.2 Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode

Be sure to set the SPIWE bit in the SFMAnSMCR register to 1 when the transfer of a command, optional command, address, or option data is started while the SPBSSL pin is being asserted in SPI operating mode.

Section 18 Multi Media Card Interface A (MMCA)

This section contains a generic description of Multi Media Card Interface A (MMCA).

The first section describes all properties specific to the RH850/U2A-EVA, such as units, register base addresses, input/output signal names, etc.

The remainder of the section describes the functions and registers of the MMCA.

18.1 Features MMCA for RH850/U2A-EVA

18.1.1 Number of Units and Channels

This microcontroller has the following number of MMCA units.

Table 18.1 Number of Units

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)	RH850/U2A6 (292 pins)	RH850/U2A6 (176 pins)	RH850/U2A6 (156 pins)	RH850/U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	—	—
Name	MMCAn									

Table 18.2 Index

Index	Description
n	Throughout this section, the individual MMCA units are identified by the index “n” (n = 0); for example, MMCAnCE_CMD_SET is the MMCAn command setting register.

18.1.2 Register Base Addresses

MMCAn base address is listed in the following table.

MMCAn register addresses are given as an offset from the base address.

Table 18.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<MMCAn_base>	FFD5 5000_H	P-Bus Group 3

18.1.3 Clock Supply

The MMCAn clock supply is shown in the following table.

Table 18.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
MMCAn	MMCA module clock	CLK_HSB	MMCA module clock
	Register access clock	CLK_HSB	Bus clock

18.1.4 Interrupt Requests and Error Notifications

The MMCAn interrupts are listed in the following table.

Table 18.5 Interrupt and DMA/DTS Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
MMCAn				
INTMMCA0	MMCA interrupt	696	—	—
DMAMMCA0	MMCA DMA request	—	Group0-191	Group1-45

This module has no error notifications.

18.1.5 Reset Sources

MMCAn reset sources are listed in the following table. MMCAn is initialized by these reset sources.

Table 18.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
MMCAn	All registers	√	√	√	√	√	√	—

18.1.6 External Input/Output Signals

External input/output signals of MMCAn are listed below.

Table 18.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
MMCAn		
MMCA0CLK	MMCA Clock	MMCA0CLK
MMCA0CMD	MMCA Command / Response	MMCA0CMD
MMCA0DAT[7:0]	MMCA Data[7:0]	MMCA0DAT[7:0]

18.2 Overview

18.2.1 Functional Overview

- Compliant with JEDEC STANDARD JESD84-A441 (neither DDR mode nor 1.8-V operation is supported).
- Supports 1-/4-/8-bit MMC bus widths.
- Supports the backward-compatible mode.
- High-speed mode is supported.
- Supports the single data rate.
- MMC Clock frequency = MMCA module clock frequency/2^k (k = 1 to 10).
- Supports block transfer.
  Stream transfer is not supported.
- Supports boot operation.
  The alternative boot operation is not supported.
- MMC clock frequency settings are adjustable in the boot mode.
- Supports high priority interrupts (HPI).
  HPIs in the sequence of CMD6, CMD24, CMD25 (pre-defined), and CMD38 are supported.
- Supports background operation.
- Interrupt requests: normal operation and error/timeout.
- DMA transfer requests: DMA or DTS can be used to read and write to the MMC interface buffer.

18.2.2 Block Diagram

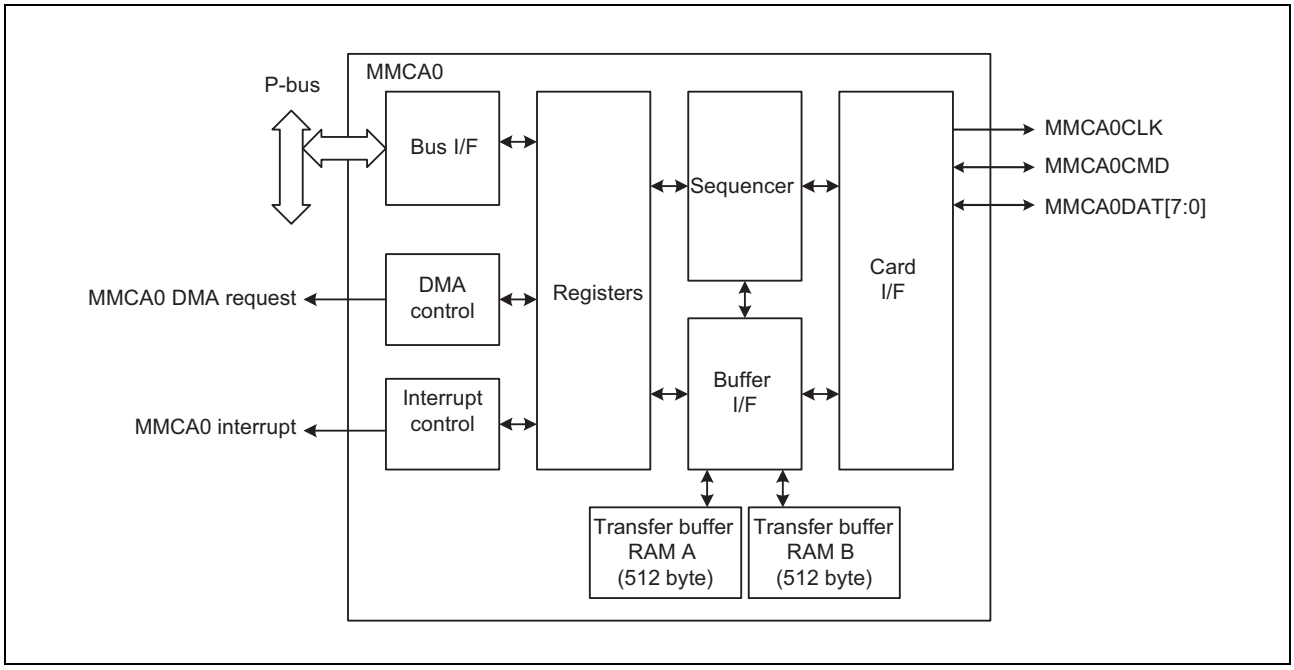


Figure 18.1 Block Diagram of MMCA

18.3 Registers

18.3.1 List of Registers

MMCA registers are listed in the following table.

For details on <MMCAn_base>, see **Section 18.1.2, Register Base Addresses**.

Table 18.8 List of Registers

Module	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MMCAn	MMCAn Command Setting Register	MMCAnCE_CMD_SET	<MMCAn_base> + 00_H	32	PBG32#0	—
	MMCAn Argument Register	MMCAnCE_ARG	<MMCAn_base> + 08_H	32	PBG32#0	—
	MMCAn Argument Register for Automatically-Issued CMD12	MMCAnCE_ARG_CMD12	<MMCAn_base> + $0C_H$	32	PBG32#0	—
	MMCAn Command Control Register	MMCAnCE_CMD_CTRL	<MMCAn_base> + 10_H	32	PBG32#0	—
	MMCAn Transfer Block Setting Register	MMCAnCE_BLOCK_SET	<MMCAn_base> + 14_H	32	PBG32#0	—
	MMCAn Clock Control Register	MMCAnCE_CLK_CTRL	<MMCAn_base> + 18_H	32	PBG32#0	—
	MMCAn Buffer Access Configuration Register	MMCAnCE_BUF_ACC	<MMCAn_base> + $1C_H$	32	PBG32#0	—
	MMCAn Response Register 3	MMCAnCE_RESP3	<MMCAn_base> + 20_H	32	PBG32#0	—
	MMCAn Response Register 2	MMCAnCE_RESP2	<MMCAn_base> + 24_H	32	PBG32#0	—
	MMCAn Response Register 1	MMCAnCE_RESP1	<MMCAn_base> + 28_H	32	PBG32#0	—
	MMCAn Response Register 0	MMCAnCE_RESP0	<MMCAn_base> + $2C_H$	32	PBG32#0	—
	MMCAn Response Register for Automatically-Issued CMD12	MMCAnCE_RESP_CMD12	<MMCAn_base> + 30_H	32	PBG32#0	—
	MMCAn Data Register	MMCAnCE_DATA	<MMCAn_base> + 34_H	32	PBG32#0	—
	MMCAn Boot Operation Setting Register	MMCAnCE_BOOT	<MMCAn_base> + $3C_H$	32	PBG32#0	—
	MMCAn Interrupt Flag Register	MMCAnCE_INT	<MMCAn_base> + 40_H	32	PBG32#0	—
	MMCAn Interrupt Enable Register	MMCAnCE_INT_EN	<MMCAn_base> + 44_H	32	PBG32#0	—
	MMCAn Status Register 1	MMCAnCE_HOST_STS1	<MMCAn_base> + 48_H	32	PBG32#0	—
	MMCAn Status Register 2	MMCAnCE_HOST_STS2	<MMCAn_base> + $4C_H$	32	PBG32#0	—
	MMCAn Software Reset Register	MMCAnCE_SWRESA	<MMCAn_base> + $7C_H$	32	PBG32#0	—

18.3.2 MMCAnCE_CMD_SET — MMCAn Command Setting Register

MMCAnCE_CMD_SET sets a command sequence. For the setting values of MMCAnCE_CMD_SET, see **Section 18.4.4.13, Setting Values of MMCAnCE_CMD_SET**. The command sequence starts when the settings have been made in bits 31 to 16. Note that writing to MMCAnCE_CMD_SET is disabled while a command sequence is proceeding (the CMDSEQ bit in MMCAnCE_HOST_STS1 is 1).

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BOOT	CMD[5:0]						RTYP[1:0]		RBSY	—	WDAT	DWEN	CMLTE	CMD12EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIDXC[1:0]		RCRC7C[1:0]		—	CRC16C	BOOTACK	CRCSTE	TBIT	OPDM	—	—	SBIT	—	DATW[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W

Table 18.9 MMCAnCE_CMD_SET Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	BOOT	Boot Operations 0: Command sequence other than for boot operations 1: Command sequence for boot operations
29 to 24	CMD[5:0]	Command Index Set a command index ([45:40]).
23, 22	RTYP[1:0]	Response Type 00: No response 01: 6-byte response (R1, R1b, R3, R4, R5) 10: 17-byte response (R2) 11: Setting prohibited
21	RBSY	Response Busy Select Selects whether "busy" is involved in response reception. 0: No response busy 1: Response busy involved (R1b)
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	WDAT	Presence/Absence of Data 0: No data 1: With data
18	DWEN	Read/Write (valid when “with data” is selected) 0: Read from card 1: Write to card
17	CMLTE	Single/Multi Block Transfer Select (valid when “with data” is selected) 0: Single-block transfer 1: Multi-block transfer

Table 18.9 MMCAnCE_CMD_SET Register Contents (2/2)

Bit Position	Bit Name	Function
16	CMD12EN	Automatic CMD12 Issuance (valid when multi-block transfer is selected) *1 0: Disables automatic CMD12 issuance 1: Enables automatic CMD12 issuance For details of automatic CMD12 issuance, see **Section 18.4.3.4, Automatic CMD12 Issuance**. Note : Set the transfer block size to 512 bytes. Set the RBSY bit to 0.
15, 14	RIDXC[1:0]	Response Index Check Specify the items to be checked in bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response. 00: Checks the response index (check whether matched with a command index) 01: Checks the check bits (check whether all the bits are set to 1) 10: No checking 11: Setting prohibited
13, 12	RCRC7C[1:0]	Response CRC7 Check Specify the items to be checked in bits [7:1] of a 6-byte response or of a 17-byte response. 00: Checks CRC7 (set the response type to 01). 01: Checks the check bits (check whether all the bits are set to 1) (set the response type to 01). 10: Checks internal CRC7 (R2 only) (set the response type to 10). 11: No checking
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	CRC16C	CRC16 Check in Reception (valid when “with data” and “read” are selected) 0: Checks CRC16 in reception. 1: Does not check CRC16 in reception (used when CMD14).
9	BOOTACK	Receive Boot Acknowledge (valid in boot mode) 0: Boot acknowledge is not received. 1: Boot acknowledge is received.
8	CRCSTE	CRC Status Reception (valid when “with data” and “write” are selected) 0: Receives CRC status. 1: Does not receive CRC status (used when CMD19).
7	TBIT	Transmission Bit Setting 0: Sets the transmission bit ([46]) to 1. 1: Sets the transmission bit ([46]) to 0.
6	OPDM	Open-Drain Output Mode 0: Normal output 1: Open-drain output This setting is only applied to the MMCAnCMD line.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SBIT	Read Data Start Bit Detection Setting (valid when “with data” and “read” are selected) 0: Detects a start bit when the valid MMCAnDAT signals specified by the DATW bits are all 0. 1: Detects a start bit when MMCAnDAT[0] is 0.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	DATW[1:0]	Data Bus Width Setting (valid when “with data” is selected) 00: 1 bit 01: 4 bits 10: 8 bits 11: Setting prohibited

Note 1. It is recommended to use the pre-defined multi-block transfer by setting this bit to 0 for a higher data transfer rate.

18.3.3 MMCAnCE_ARG — MMCAn Argument Register

MMCAnCE_ARG sets the argument for the command to be transmitted. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.10 MMCAnCE_ARG Register Contents

Bit Position	Bit Name	Function
31 to 0	ARG[31:0]	Set bits [39:8] of the command. Note : Set the argument of automatically-issued CMD12 by MMCAnCE_ARG_CMD12.

18.3.4 MMCAnCE_ARG_CMD12 — MMCAn Argument Register for Automatically-Issued CMD12

MMCAnCE_ARG_CMD12 is used to set the argument for the automatically-issued CMD12. This register is valid when issuing CMD12 automatically in multi-block transfer. For automatic issuance of CMD12, see **Section 18.4.3.4, Automatic CMD12 Issuance**. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	C12ARG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C12ARG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.11 MMCAnCE_ARG_CMD12 Register Contents

Bit Position	Bit Name	Function
31 to 0	C12ARG[31:0]	Set bits [39:8] of the automatically-issued CMD12.

18.3.5 MMCAnCE_CMD_CTRL — MMCAn Command Control Register

MMCAnCE_CMD_CTRL is used to terminate a command sequence forcibly.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BREAK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 18.12 MMCAnCE_CMD_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BREAK	Forcible Termination of Command Sequence Writing 1 to this bit while it is 0 and then writing 0 to it discontinues the current command sequence. After this bit is set as described above, check if the value of the CMDSEQ bit in MMCAnCE_HOST_STS1 has become 0. If this is the case, execute software reset. Note : A software reset initializes the value of this register, so the setting in this register needs to be remade.

18.3.6 MMCAnCE_BLOCK_SET — MMCAn Transfer Block Setting Register

MMCAnCE_BLOCK_SET specifies the size of the block and the number of blocks for the data to be transferred. Set this register before starting a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 14_H

Value after reset: 0000 0200_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLKCNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLKSIZ[15:0]															
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.13 MMCAnCE_BLOCK_SET Register Contents

Bit Position	Bit Name	Function
31 to 16	BLKCNT[15:0]	Number of Blocks for Transfer Note : This setting is valid for multi-block transfer.
15 to 0	BLKSIZ[15:0]	Transfer Block Size Note : Transfer block size should be set as follows. Single-block transfer: 1 to 512 bytes Multi-block transfer: 512 bytes

18.3.7 MMCAnCE_CLK_CTRL — MMCAn Clock Control Register

MMCAnCE_CLK_CTRL controls the MMC clock and sets timeout values. Do not change the setting of this register while a command sequence is in progress.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 18_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMCBUSBSY	—	—	—	—	—	—	CLKEN	—	—	—	—	CLKDIV[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SRSPTO[1:0]		SRBSYTO[3:0]				SRWDTO[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 18.14 MMCAnCE_CLK_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31	MMCBUSBSY	MMC Clock Output control / Division setting prohibition 0: MMC bus is not busy (During the command sequence + 10 cycle) 1: MMC bus is busy (During the command sequence + 10 cycle) When a command sequence is started with MMCAnCE_CMD_SET register setting, MMCBUSBSY bit is set to 1 at the same time as CMDSEQ bit is set to 1. After CMDSEQ bit is set to 0 at the command sequence end, MMCBUSBSY bit becomes 0 after 10 cycle of MMCAnCLK. Note: When MMC bus is busy, do not set CLKEN bit and the CLKDIV bit of the MMCAnCE_CLK_CTRL register.
30 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CLKEN	MMC Clock Output Control 0: Does not output the MMC clock (fixed to low level). 1: Outputs the MMC clock.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	CLKDIV[3:0]	MMC Clock Frequency Setting 0000: MMCA module clock/2^1 0001: MMCA module clock/2^2 : 0111: MMCA module clock/2^8 1000: MMCA module clock/2^9 1001: MMCA module clock/2^{10} 1010 to 1111: Setting prohibited For details of MMC clock frequency settings in boot operations, see **Section 18.4.3.5, MMC Clock Frequency in Boot Operations** and **Section 18.3.12, MMCAnCE_BOOT — MMCAn Boot Operation Setting Register**.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	SRSPTO[1:0]	Response Timeout Setting 00: 64 MMC clock cycles 01: 128 MMC clock cycles 10: 256 MMC clock cycles 11: Setting prohibited

Table 18.14 MMCAnCE_CLK_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	SRBSYTO[3:0]	Response Busy Timeout Setting 0000: MMC clock cycles × 2^{14} 0001: MMC clock cycles × 2^{15} 0010: MMC clock cycles × 2^{16} 0011: MMC clock cycles × 2^{17} 0100: MMC clock cycles × 2^{18} 0101: MMC clock cycles × 2^{19} 0110: MMC clock cycles × 2^{20} 0111: MMC clock cycles × 2^{21} 1000: MMC clock cycles × 2^{22} 1001: MMC clock cycles × 2^{23} 1010: MMC clock cycles × 2^{24} 1011: MMC clock cycles × 2^{25} 1100: MMC clock cycles × 2^{26} 1101: MMC clock cycles × 2^{27} 1110: MMC clock cycles × 2^{28} 1111: MMC clock cycles × 2^{29}
7 to 4	SRWDTO[3:0]	Write Data/Read Data Timeout Setting 0000: MMC clock cycles × 2^{14} 0001: MMC clock cycles × 2^{15} 0010: MMC clock cycles × 2^{16} 0011: MMC clock cycles × 2^{17} 0100: MMC clock cycles × 2^{18} 0101: MMC clock cycles × 2^{19} 0110: MMC clock cycles × 2^{20} 0111: MMC clock cycles × 2^{21} 1000: MMC clock cycles × 2^{22} 1001: MMC clock cycles × 2^{23} 1010: MMC clock cycles × 2^{24} 1011: MMC clock cycles × 2^{25} 1100: MMC clock cycles × 2^{26} 1101: MMC clock cycles × 2^{27} 1110: MMC clock cycles × 2^{28} 1111: MMC clock cycles × 2^{29}
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.8 MMCAnCE_BUF_ACC — MMCAn Buffer Access Configuration Register

MMCAnCE_BUF_ACC configures the method of accessing data registers and mode of DMA transfer. Do not set this register again during a command sequence. For explanation of the buffers, see **Section 18.4.3.3, Buffer Structure and Buffer Access**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $1C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DMATYP	DMAWEN	DMAREN	—	—	—	—	—	—	—	ATYP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.15 MMCAnCE_BUF_ACC Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	DMATYP	DMA Transfer setting When DMAWEN or DMAREN is set to 1, set to 1.
25	DMAWEN	Buffer Write DMA Transfer Request Enable 0: Disables DMA transfer request for buffer writing. 1: Enables DMA transfer request for buffer writing.
24	DMAREN	Buffer Read DMA Transfer Request Enable 0: Disables DMA transfer request for buffer reading. 1: Enables DMA transfer request for buffer reading.
23 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	ATYP	Buffer access selection 0: When not swapped byte-wise. 1: When swapped byte-wise. Note : For buffer access, refer to **18.4.3.3, Buffer Structure and Buffer Access**.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.9 MMCAnCE_RESP3 to MMCAnCE_RESP0 — MMCAn Response Registers 3 to 0

MMCAnCE_RESP3 to MMCAnCE_RESP0 are the registers in which the response that has been received is stored. For the formats of response values, see **Section 18.4.3.1, Command/Response Formats**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: MMCAnCE_RESP3: <MMCAn_base> + 20_H
MMCAnCE_RESP2: <MMCAn_base> + 24_H
MMCAnCE_RESP1: <MMCAn_base> + 28_H
MMCAnCE_RESP0: <MMCAn_base> + $2C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

MMCAnCE_RESP3

Table 18.16 MMCAnCE_RESP3 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[127:96]	Bits [127:96] of a 17-byte response are stored.

MMCAnCE_RESP2

Table 18.17 MMCAnCE_RESP2 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[95:64]	Bits [95:64] of a 17-byte response are stored.

MMCAnCE_RESP1

Table 18.18 MMCAnCE_RESP1 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[63:32]	Bits [63:32] of a 17-byte response are stored.

MMCAnCE_RESP0

Table 18.19 MMCAnCE_RESP0 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP[31:0]	Bits [39:8] of a 6-byte response or bits [31:0] of a 17-byte response are stored. Note : The response to an automatically issued CMD12 is stored in the MMCAnCE_RESP_CMD12 register.

18.3.10 MMCAnCE_RESP_CMD12 — MMCAn Response Register for Automatically-Issued CMD12

MMCAnCE_RESP_CMD12 is a register in which the response to the automatically-issued CMD12 is stored.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MMCAn_base> + 30_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP12[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSP12[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.20 MMCAnCE_RESP_CMD12 Register Contents

Bit Position	Bit Name	Function
31 to 0	RSP12[31:0]	Bits [39:8] of the response to automatically-issued CMD12 are stored.

18.3.11 MMCAnCE_DATA — MMCAn Data Register

MMCAnCE_DATA is used to access the buffers of this module. For the write/read data formats, see **18.4.3.2, Data Block Format**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 34_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.21 MMCAnCE_DATA Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Buffer write/read data [31:0]

18.3.12 MMCAnCE_BOOT — MMCAn Boot Operation Setting Register

MMCAnCE_BOOT controls the MMC clock and sets timeout values in boot mode. Do not set this register again during a command sequence.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $3C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BTCLKDIV[3:0]				SBTACKTO[3:0]				S1STBTDATTO[3:0]				SBTDATTO[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.22 MMCAnCE_BOOT Register Contents

Bit Position	Bit Name	Function
31 to 28	BTCLKDIV[3:0]	MMC Clock Frequency Setting in Boot Mode 0000: MMCA module clock/2^1 0001: MMCA module clock/2^2 0010: MMCA module clock/2^3 0011: MMCA module clock/2^4 0100 to 1111: Settings prohibited Set these bits to a value lower than that in the CLKDIV bits of MMCAnCE_CLK_CTRL. For MMC clock frequency in boot mode, see **Section 18.4.3.5, MMC Clock Frequency in Boot Operations**.
27 to 24	SBTACKTO[3:0]	Boot Acknowledge Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
23 to 20	S1STBTDATTO [3:0]	1st Boot Data Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
19 to 16	SBTDATTO[3:0]	Interval Between Boot Data Timeout Setting 0000: 2^{14} × MMC clock cycles 0001: 2^{15} × MMC clock cycles : 1110: 2^{28} × MMC clock cycles 1111: 2^{29} × MMC clock cycles
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.3.13 MMCAnCE_INT — MMCAn Interrupt Flag Register

MMCAnCE_INT indicates various statuses during execution of a command sequence. Each bit is set when its setting condition has been met. To clear flag(s), write 0 only to the bit(s) to be cleared and write 1 to the other bits. For the handling of this module in the case of an error or timeout, see **Section 18.4.3.8, Handling of This Module in the Case of Error/Timeout**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CMD12DRE	CMD12RBE	CMD12CRE	DTRANE	BUFRE	BUFWEN	BUFREN	—	—	RBSYE	CRSPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMDVIO	BUFVIO	—	—	WDATERR	RDATERR	RIDXERR	RSPERR	—	—	—	CRCSTO	WDATTO	RDATTO	RBSYTO	RSPTO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. Writing 0 initializes the bit. Writing 1 is ignored.

Table 18.23 MMCAnCE_INT Register Contents (1/5)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	CMD12DRE	Automatic CMD12 Issuance & Buffer Read Complete [Setting condition] Response busy for automatically-issued CMD12 and buffer reading have been completed. [Clearing condition] Writing a 0 to this bit. **Note:** When CMD12DRE has been set, CMD12RBE, CMD12CRE, and BUFRE have also been set. So, these bits should be cleared as well.
25	CMD12RBE	Automatic CMD12 Issuance Response Busy Complete [Setting condition] Reception of the response and response busy for an automatically-issued CMD12 have been completed. [Clearing condition] Writing a 0 to this bit. **Note:** When CMD12RBE has been set, CMD12CRE has also been set. So, this bit should be cleared as well. When CMD12RBE is set during a multi-block write, DTRANE is also set. So clear the bit as well.
24	CMD12CRE	Automatic CMD12 Response Complete [Setting condition] The response to an automatically-issued CMD12 has been received. [Clearing condition] Writing a 0 to this bit.

Table 18.23 MMCAnCE_INT Register Contents (2/5)

Bit Position	Bit Name	Function
23	DTRANE	Data Transmission Complete [Setting conditions] Transmission of all blocks of data has been completed. – When configured to receive CRC status: Completion of busy (data busy) after reception of CRC status – When configured not to receive CRC status: Completion of data transmission [Clearing condition] Writing a 0 to this bit.
22	BUFRE	Buffer Read Complete [Setting conditions] – Other than in boot operations All blocks of data have been received and the data have been read from the buffer. – In boot operations All blocks of data have been received and the data have been read from the buffer, MMCAnCMD has been modified from 0 to 1, and 48 MMC clock cycles have elapsed. [Clearing condition] Writing a 0 to this bit.
21	BUFWEN	Buffer Write Ready [Setting conditions] The buffer has become empty and ready for writing. [Clearing condition] Writing a 0 to this bit. **Note:** When writing data to MMCAnCE_DATA by the CPU, this bit should be cleared first and the data corresponding to the block size set in MMCAnCE_BLOCK_SET should be written. Note that this bit is not set when DMA transfer request for buffer writing is enabled.
20	BUFREN	Buffer Read Ready [Setting conditions] Transfer block size of data have been stored in the buffer and it has become ready for reading. [Clearing condition] Writing a 0 to this bit. **Note:** When reading data from MMCAnCE_DATA by the CPU, this bit should be cleared first and the data corresponding to the block size set in MMCAnCE_BLOCK_SET should be read. Note that this bit is not set when DMA transfer request for buffer reading is enabled.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RBSYE	Response Busy Complete [Setting condition] Reception of a response and response busy have been completed. [Clearing condition] Writing a 0 to this bit. **Note:** When RBSYE has been set, CRSPE has also been set. So, this bit should be cleared as well. Completion of reception of the response and response busy for automatically-issued CMD12 is reflected in CMD12RBE.

Table 18.23 MMCAnCE_INT Register Contents (3/5)

Bit Position	Bit Name	Function
16	CRSPE	Command/Response Complete [Setting conditions] – Other than in boot operations When configured not to receive response: A command has been transmitted. When configured to receive 6- or 17-byte response: A response has been received. – In boot operations When reception of boot acknowledge has been selected: The boot acknowledge pattern has been received. [Clearing condition] Writing a 0 to this bit. Note : Completion of reception of the response to automatically-issued CMD12 is reflected in CMD12CRE.
15	CMDVIO	Command Issuance Error [Setting conditions] Illegal setting has been made in MMCAnCE_CMD_SET or MMCAnCE_BLOCK_SET. – During execution of a command sequence: Writing to CMD[5:0] in MMCAnCE_CMD_SET. (The command sequence is not stopped automatically.) – At the start of command sequence: - Writing to CMD[5:0] in MMCAnCE_CMD_SET when the registers have been set for one of the following writing a combinations of selection. - No response + response busy - No response + with data + not during boot operations - No data + automatic CMD12 issuance - With data + single-block transfer + automatic CMD12 issuance - With data + response busy + automatic CMD12 issuance - With data + transfer block size = 0 - With data + transfer block size ≥ 513 - With data + multi-block transfer + number of blocks for transfer = 0 - Boot operations + no data - Boot operations + write - Boot operations + response busy - Boot operations + automatic CMD12 issuance - Boot acknowledge reception + not during boot operations [Clearing condition] Writing a 0 to this bit.
14	BUFVIO	Buffer Access Error [Setting conditions] Illegal buffer access has been attempted. – MMCAnCE_ DATA has been accessed exceeding the block size set in BLKSIZ[15:0] in MMCAnCE_BLOCK_SET. – While data is being read from the card: MMCAnCE_DATA has been accessed with BUFREN not set (when DMA is used, with no DMA transfer request asserted for buffer reading). – While data is being written to the card: MMCAnCE_DATA has been accessed with BUFWEN not set (when DMA is used, with no DMA transfer request asserted for buffer writing). [Clearing condition] Writing a 0 to this bit. Note : When BUFVIO has been set, the command sequence is not stopped automatically. If an error occurs, this bit will be set.
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 18.23 MMCAnCE_INT Register Contents (4/5)

Bit Position	Bit Name	Function
11	WDATERR	Write Data Error [Setting conditions] Error is found in the data that has been written. – Error is in the status of the CRC status. – Error is in the end bits of the CRC status. [Clearing condition] Writing a 0 to this bit. Note : When WDATERR has been set, the command sequence is stopped automatically.
10	RDATERR	Read Data Error [Setting conditions] Error is found in the read data. – Error is in CRC16 of the read data. – Error is in the end bits of the read data. [Clearing condition] Writing a 0 to this bit. Note : When RDATERR has been set, the command sequence is stopped automatically.
9	RIDXERR	Response Index Error [Setting conditions] Error has been found in the index value of the response. – When an error has been found in bits [45:40] of a 6-byte response (including automatically-issued CMD12) or bits [133:128] of a 17-byte response. (The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.) [Clearing condition] Writing a 0 to this bit. Note : When RIDXERR has been set, the command sequence is stopped automatically.
8	RSPERR	Response Error [Setting conditions] Error has been found in the response values of the response. – Transmission bit in the response is 1. – Error is in the end bits of the response. – When an error has been found in bits [7:1] of a 6-byte response (including automatically-issued CMD12) or a 17-byte response. (The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.) – Error in the boot acknowledge pattern. – Error in the end bits of the boot acknowledge. [Clearing condition] Writing a 0 to this bit. Note : When RSPERR has been set, the command sequence is stopped automatically.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CRCSTO	CRC Status Timeout [Setting condition] CRC status could not be received. [Clearing condition] Writing a 0 to this bit. Note : The command sequence is not stopped even if CRCSTO is set.

Table 18.23 MMCAnCE_INT Register Contents (5/5)

Bit Position	Bit Name	Function
3	WDATTO	Write Data Timeout [Setting condition] The busy status remains unchanged after the period set by SRWDTO in MMCAnCE_CLK_CTRL after the CRC status was received. [Clearing condition] Writing a 0 to this bit. Note : The command sequence is not stopped even if WDATTO is set.
2	RDATTO	Read Data Timeout [Setting conditions] – Other than in boot operations - Read data could not be received within the period set by SRWDTO in MMCAnCE_CLK_CTRL after the read command was transmitted. - Read data could not be received within the period set by SRWDTO in MMCAnCE_CLK_CTRL after the read data was received. – In boot operations - The first read data could not be received within the period set by S1STBTDATTO in MMCAnCE_BOOT. - Read data could not be received within the period set by SBTDATTO in MMCAnCE_BOOT after the read data was received. [Clearing condition] Writing a 0 to this bit. Note : The command sequence is not stopped even if RDATTO is set.
1	RBSYTO	Response Busy Timeout [Setting condition] The busy status remains unchanged after the period set by SRBSYTO in MMCAnCE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. [Clearing condition] Writing a 0 to this bit. Note : The command sequence is not stopped even if RBSYTO is set.
0	RSPTO	Response Timeout [Setting conditions] – Other than in boot operations Response could not be received within the period set by SRSPTO in MMCAnCE_CLK_CTRL after the command (including automatically-issued CMD12) was transmitted. – In boot operations When reception of boot acknowledge has been selected: The boot acknowledge could not be received within the period set by SBTACKTO in MMCAnCE_BOOT. [Clearing condition] Writing a 0 to this bit. Note : The command sequence is not stopped even if RSPTO is set.

18.3.14 MMCAnCE_INT_EN — MMCAn Interrupt Enable Register

MMCAnCE_INT_EN controls output of the MMCAnCE_INT-related interrupt signals. For details on interrupt requests, see **Section 18.4.1, Interrupt Requests**.

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + 44_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MCMD12DRE	MCMD12RBE	MCMD12CRE	MDTRANE	MBUFRE	MBUFWEN	MBUFREN	—	—	MRBSYE	MCRSPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCMDVIO	MBUFVIO	—	—	MWDATERR	MRDATERR	MRIDXERR	MRSPERR	—	—	—	MCRCSTO	MWDATTO	MRDATTO	MRBSYTO	MRSPTO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 18.24 MMCAnCE_INT_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	MCMD12DRE	CMD12DRE Interrupt Enable 0: Disables interrupt output by the CMD12DRE flag. 1: Enables interrupt output by the CMD12DRE flag.
25	MCMD12RBE	CMD12RBE Interrupt Enable 0: Disables interrupt output by the CMD12RBE flag. 1: Enables interrupt output by the CMD12RBE flag.
24	MCMD12CRE	CMD12CRE Interrupt Enable 0: Disables interrupt output by the CMD12CRE flag. 1: Enables interrupt output by the CMD12CRE flag.
23	MDTRANE	DTRANE Interrupt Enable 0: Disables interrupt output by the DTRANE flag. 1: Enables interrupt output by the DTRANE flag.
22	MBUFRE	BUFRE Interrupt Enable 0: Disables interrupt output by the BUFRE flag. 1: Enables interrupt output by the BUFRE flag.
21	MBUFWEN	BUFWEN Interrupt Enable 0: Disables interrupt output by the BUFWEN flag. 1: Enables interrupt output by the BUFWEN flag.
20	MBUFREN	BUFREN Interrupt Enable 0: Disables interrupt output by the BUFREN flag. 1: Enables interrupt output by the BUFREN flag.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MRBSYE	RBSYE Interrupt Enable 0: Disables interrupt output by the RBSYE flag. 1: Enables interrupt output by the RBSYE flag.
16	MCRSPE	CRSPE Interrupt Enable 0: Disables interrupt output by the CRSPE flag. 1: Enables interrupt output by the CRSPE flag.
15	MCMDVIO	CMDVIO Interrupt Enable 0: Disables interrupt output by the CMDVIO flag. 1: Enables interrupt output by the CMDVIO flag.

Table 18.24 MMCAnCE_INT_EN Register Contents (2/2)

Bit Position	Bit Name	Function
14	MBUFVIO	BUFVIO Interrupt Enable 0: Disables interrupt output by the BUFVIO flag. 1: Enables interrupt output by the BUFVIO flag.
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MWDATERR	WDATERR Interrupt Enable 0: Disables interrupt output by the WDATERR flag. 1: Enables interrupt output by the WDATERR flag.
10	MRDATERR	RDATERR Interrupt Enable 0: Disables interrupt output by the RDATERR flag. 1: Enables interrupt output by the RDATERR flag.
9	MRIDXERR	RIDXERR Interrupt Enable 0: Disables interrupt output by the RIDXERR flag. 1: Enables interrupt output by the RIDXERR flag.
8	MRSPERR	RSPERR Interrupt Enable 0: Disables interrupt output by the RSPERR flag. 1: Enables interrupt output by the RSPERR flag.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	MCRCSTO	CRCSTO Interrupt Enable 0: Disables interrupt output by the CRCSTO flag. 1: Enables interrupt output by the CRCSTO flag.
3	MWDATTO	WDATTO Interrupt Enable 0: Disables interrupt output by the WDATTO flag. 1: Enables interrupt output by the WDATTO flag.
2	MRDATTO	RDATTO Interrupt Enable 0: Disables interrupt output by the RDATTO flag. 1: Enables interrupt output by the RDATTO flag.
1	MRBSYTO	RBSYTO Interrupt Enable 0: Disables interrupt output by the RBSYTO flag. 1: Enables interrupt output by the RBSYTO flag.
0	MRSPTO	RSPTO Interrupt Enable 0: Disables interrupt output by the RSPTO flag. 1: Enables interrupt output by the RSPTO flag.

18.3.15 MMCAnCE_HOST_STS1 — MMCAn Status Register 1

MMCAnCE_HOST_STS1 indicates the number of blocks that have been transferred, the states of the MMCAnCMD line and MMCAnDAT lines, the index of the response that has been received, and whether a command sequence is in progress.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MMCAn_base> + 48_H

Value after reset: X0XX 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMDSEQ	CMDSIG	RSPIDX[5:0]						DATSIG[7:0]							
Value after reset	0	—	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCVBLK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.25 MMCAnCE_HOST_STS1 Register Contents

Bit Position	Bit Name	Function
31	CMDSEQ	Command Sequence in Progress 0: Command sequence is in the initial state. 1: Command sequence is being executed.
30	CMDSIG	MMCAnCMD State Indicates the state on the MMCAnCMD line.
29 to 24	RSPIDX[5:0]	Response Index Indicate bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response.
23 to 16	DATSIG[7:0]	MMCAnDAT[7:0] State Indicate the state on the MMCAnDAT[7:0] lines. **Note:** When a communication error or a timeout error occurs, MMCAnDAT[0] may remain 0.
15 to 0	RCVBLK[15:0]	Number of Transferred Blocks Indicate the number of blocks that have been transferred. When the DWEN bit in MMCAnCE_CMD_SET is 0: Number of blocks read from the card When the DWEN bit in MMCAnCE_CMD_SET is 1: Number of blocks written to the card

18.3.16 MMCAnCE_HOST_STS2 — MMCAn Status Register 2

MMCAnCE_HOST_STS2 indicates timeout and error statuses.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MMCAn_base> + $4C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRCSTE	CRC16E	AC12CRCE	RSPCRC7E	CRCSTEBE	RDATEBE	AC12REBE	RSPEBE	AC12IDXE	RSPIDXE	BTACKPATE	BTACKEBE	—	CRCST[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	STRDATTO	DATBSYTO	CRCSTTO	AC12BSYTO	RSPBSYTO	AC12RSPTO	STRSPTO	BTACKTO	1STBTDATTO	BTDATTO	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.26 MMCAnCE_HOST_STS2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	CRCSTE	CRC Status Error This bit is set to 1 when an error is found in the CRC status value.
30	CRC16E	Read Data CRC16 Error This bit is set to 1 when an error is found in CRC16 in the read data.
29	AC12CRCE	Automatic CMD12 Response CRC7 Error This bit is set to 1 when an error is found in bits [7:1] of the response to the automatically-issued CMD12. Note : The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.
28	RSPCRC7E	Command Response CRC7 Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in bits [7:1] of a 6-byte response or a 17-byte response. Note : The items to be checked are set by RCRC7C in MMCAnCE_CMD_SET.
27	CRCSTEBE	CRC Status End Bit Error This bit is set to 1 when an error is found in the end bits in CRC status.
26	RDATEBE	Read Data End Bit Error This bit is set to 1 when an error is found in the end bits in read data.
25	AC12REBE	Automatic CMD12 Response End Bit Error This bit is set to 1 when an error is found in the end bits of the response to the automatically-issued CMD12.
24	RSPEBE	Command Response End Bit Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in the end bits of the response.
23	AC12IDXE	Automatic CMD12 Response Index Error This bit is set to 1 when an error is found in bits [45:40] of the response to the automatically-issued CMD12. Note : The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.
22	RSPIDXE	Command Response Index Error (other than automatically-issued CMD12) This bit is set to 1 when an error is found in bits [45:40] of a 6-byte response or bits [133:128] of a 17-byte response. Note : The items to be checked are set by RIDXC in MMCAnCE_CMD_SET.
21	BTACKPATE	Boot Acknowledge Pattern Error This bit is set to 1 when an error is found in the boot acknowledge pattern.
20	BTACKEBE	Boot Acknowledge End Bit Error This bit is set to 1 when an error is found in the end bits of the boot-acknowledge.
19	Reserved	When read, the value after reset is returned.

Table 18.26 MMCAnCE_HOST_STS2 Register Contents (2/2)

Bit Position	Bit Name	Function
18 to 16	CRCST[2:0]	CRC Status/Boot Acknowledge Pattern Indication This bit indicates the value for CRC status that was received or the pattern value from the boot acknowledge.
15	Reserved	When read, the value after reset is returned.
14	STRDATTO	Read Data Timeout (valid other than in boot operations) This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after a read command was transmitted. This bit is set to 1 if read data is not received within the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after a read data was received.
13	DATBSYTO	Data Busy Timeout This bit is set to 1 if busy status remains unchanged after the period set by the SRWDTO bits in MMCAnCE_CLK_CTRL after the CRC status was received.
12	CRCSTTO	CRC Status Timeout This bit is set to 1 if CRC status could not be received.
11	AC12BSYTO	Automatic CMD12 Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
10	RSPBSYTO	Response Busy Timeout This bit is set to 1 if busy state remains unchanged after the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
9	AC12RSPTO	Automatic CMD12 Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in MMCAnCE_CLK_CTRL after the automatically-issued CMD12 was transmitted.
8	STRSPTO	Response Timeout This bit is set to 1 if the response is not received within the period set by the SRSPTO bits in MMCAnCE_CLK_CTRL after a command (other than automatically-issued CMD12) was transmitted.
7	BTACKTO	Boot Acknowledge Timeout In boot operations, this bit is set to 1 if boot acknowledge is not received within the period set by the SBTACKTO bits in MMCAnCE_BOOT.
6	1STBTDATTO	1st Boot Data Timeout In boot operations, this bit is set to 1 if the 1st read data is not received within the period set by the S1STBTDATTO bits in MMCAnCE_BOOT.
5	BTDATTO	Interval between Boot Data Timeout In boot operations, this bit is set to 1 if read data is not received within the period set by the SBTDATTO bits in MMCAnCE_BOOT after a read data was received.
4 to 0	Reserved	When read, the value after reset is returned.

18.3.17 MMCAnCE_SWRESA — MMCAn Software Reset Register

MMCAnCE_SWRESA controls software reset of this module

Access: This register can be read or written in 32-bit units.

Address: <MMCAn_base> + $7C_H$

Value after reset: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SWRST	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.27 MMCAnCE_SWRESA Register Contents

Bit Position	Bit Name	Function
31	SWRST	Software Reset 0: Software reset cleared (normal operation). 1: Executes software reset. When this bit is set to 1, the values of all registers are initialized. (SWRST is not initialized.)
30 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

18.4 Operations

18.4.1 Interrupt Requests

The table below shows the specification of the interrupt requests. This module generates two types of interrupt requests: normal operation and error/timeout. When a bit in the flag register is set to 1 and also the corresponding bit in the interrupt enable register is set to 1 (enabled), an interrupt request is asserted.

Table 18.28 Specification of Interrupt Requests

Flag Register	Bit	Enable Register	Bit	Interrupt Request
MMCAnCE_INT	CMD12DRE	MMCAnCE_INT_EN	MCMD12DRE	Normal operation interrupt
	CMD12RBE		MCMD12RBE	
	CMD12CRE		MCMD12CRE	
	DTRANE		MDTRANE	
	BUFRE		MBUFRE	
	BUFWEN		MBUFWEN	
	BUFREN		MBUFREN	
	RBSYE		MRBSYE	
	CRSPE		MCRSPE	
	CMDVIO		MCMDVIO	Error/timeout interrupt
	BUFVIO		MBUFVIO	
	WDATERR		MWDATERR	
	RDATERR		MRDATERR	
	RIDXERR		MRIDXERR	
	RSPERR		MRSPERR	
	CRCSTO		MCRCSTO	
	WDATTO		MWDATTO	
	RDATTO		MRDATTO	
	RBSYTO		MRBSYTO	
	RSPTO		MRSPTO	

18.4.2 DMA Specifications

This module has two types of DMA transfer requests: for buffer reading and for buffer writing.

The method of DMA transfer is configured by MMCAnCE_BUF_ACC.

18.4.2.1 DMA for Buffer Writing

The DMA transfer request is asserted for buffer writing when the buffer has become empty while the DMAWEN bit in MMCAnCE_BUF_ACC is set to 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in MMCAnCE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in MMCAnCE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFWEN bit in MMCAnCE_INT will not be asserted during DMA transfer.

If an error has occurred during DMA transfer or the command sequence is forcibly terminated by setting the BREAK bit in the MMCAnCE_CMD_CTRL register, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

18.4.2.2 DMA for Buffer Reading

The DMA transfer request is asserted for buffer reading when the buffer stores data of the block size specified in MMCAnCE_BLOCK_SET while the DMAREN bit in MMCAnCE_BUF_ACC is set to 1.

The DMA transfer request stays asserted for the amount of data specified by BLKSIZ (the block size set in MMCAnCE_BLOCK_SET) × BLKCNT (the number of blocks for transfer set in MMCAnCE_BLOCK_SET), and negated after the last block has been transferred. Note that the BUFREN bit in MMCAnCE_INT will not be asserted during DMA transfer.

If an error has occurred during DMA transfer or the command sequence is forcibly terminated by setting the BREAK bit in the MMCAnCE_CMD_CTRL register, the command sequence is stopped automatically, which causes the DMA transfer request to be negated.

18.4.3 Operations

18.4.3.1 Command/Response Formats

The figure below shows the format of the command to be transferred. The command index that is set in the CMD[5:0] bits in MMCAnCE_CMD_SET and the argument set in the ARG[31:0] bits in MMCAnCE_ARG are reflected in the command.

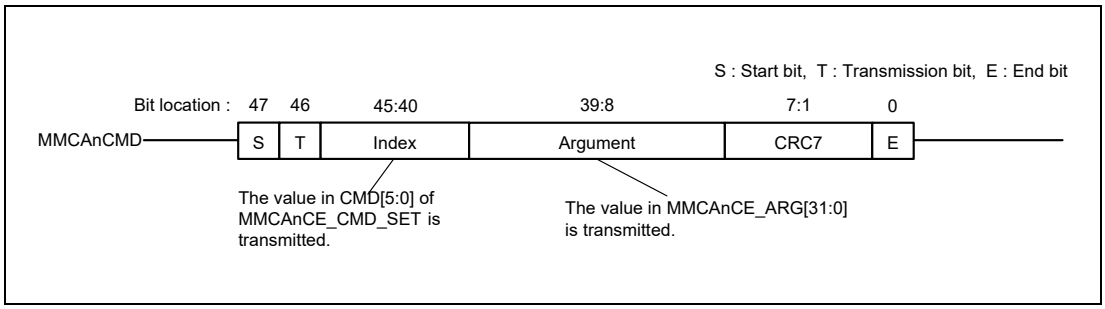


Figure 18.2 Command Format

Figure 18.3 and **Figure 18.4** show the formats when a 6-byte response and 17-byte response (R2) are received, respectively. The received response is stored in MMCAnCE_RESP0 or MMCAnCE_RESP3 to MMCAnCE_RESP0. The items to be checked are set by the RIDXC bits and the RCRC7C bits in MMCAnCE_CMD_SET.

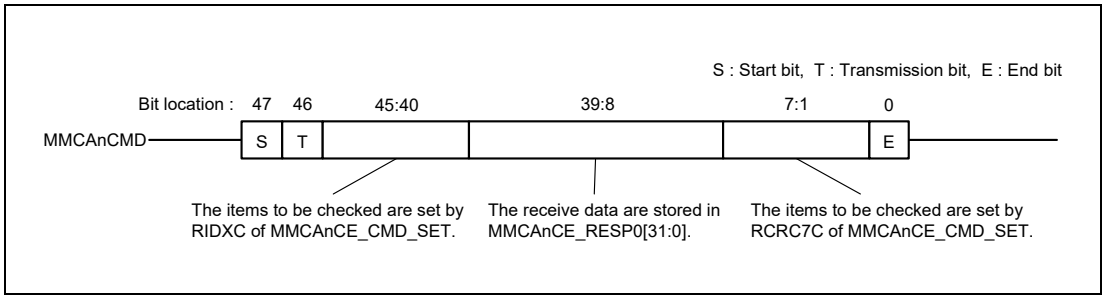


Figure 18.3 Format of 6-Byte Response

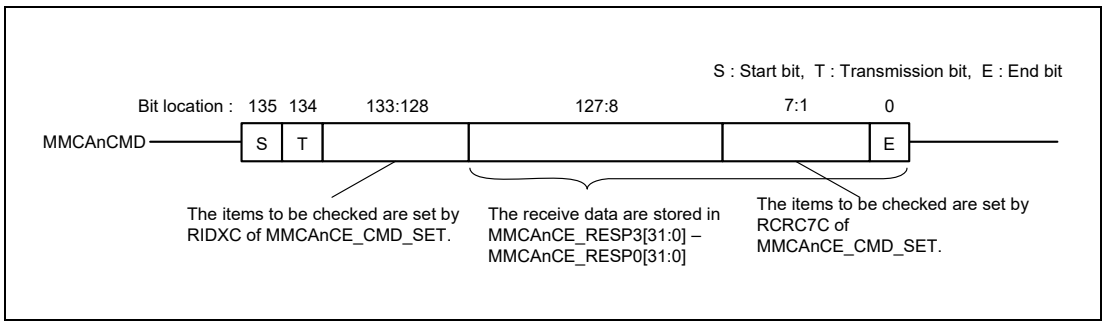


Figure 18.4 Format of 17-Byte Response (R2)

18.4.3.2 Data Block Format

The figure below shows the format of data blocks. For D0 to D3 in the figure, see **Section 18.4.3.3, Buffer Structure and Buffer Access**. When data are written to the card, the data stored in the buffer are transmitted. When data are read from the card, the received data are stored in the buffer.

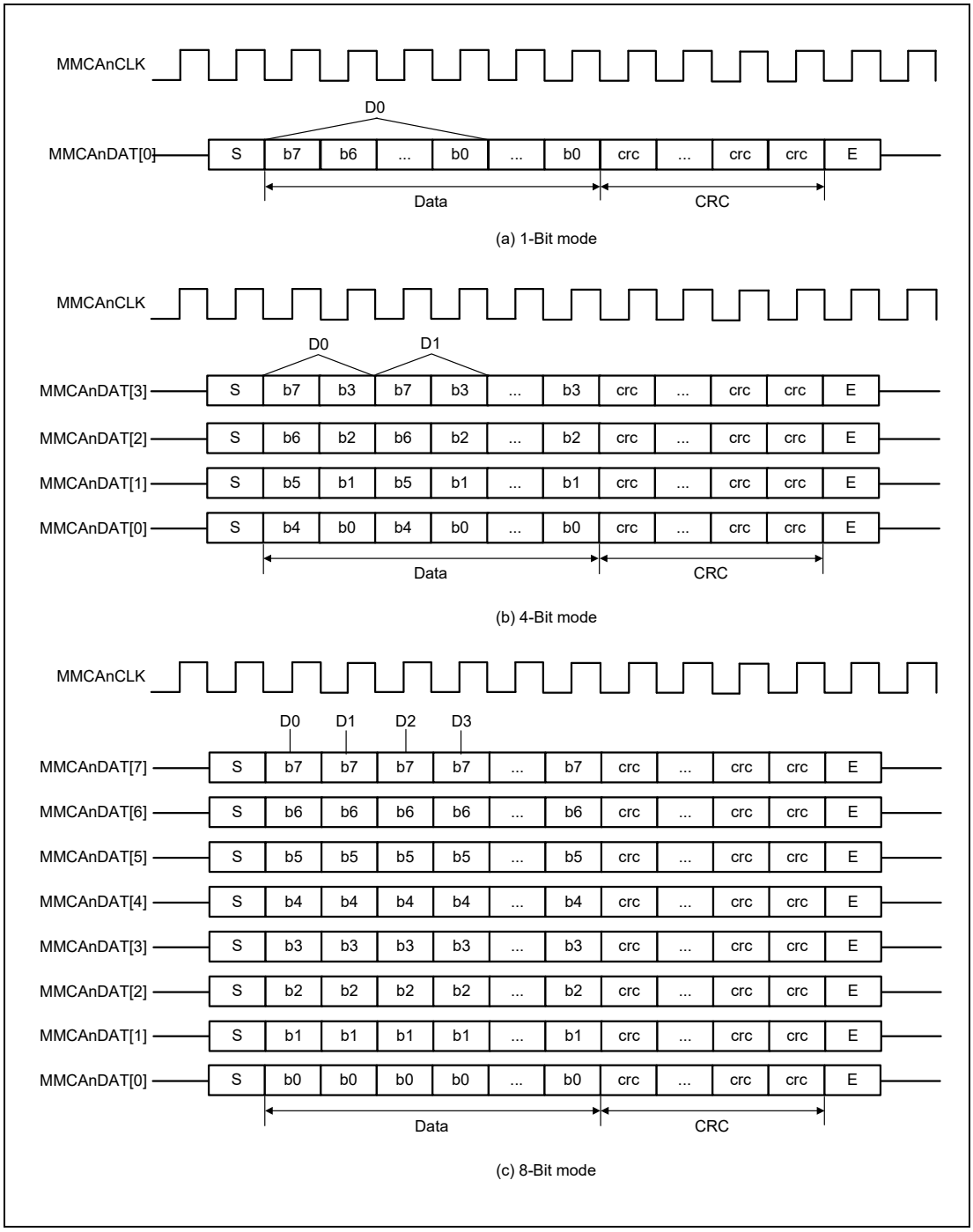


Figure 18.5 Data Block Format

18.4.3.3 Buffer Structure and Buffer Access

As shown in **Figure 18.6**, this module has two 512-byte RAM units for double buffering. If a block of data (512 bytes) stored in one buffer is transmitted during multi-block writing, the next block of data can be transmitted as soon as the other buffer is full. If a block of received data (512 bytes) is stored in one buffer during multi-block reading, the next block of received data can be stored in the other buffer once it is empty.

If neither buffer is empty during multi-block reading, the MMC clock is stopped, which in turn temporarily stops reception. Once either of the buffers becomes empty, supply of the MMC clock signal is re-started and data reception is resumed.

The buffers are accessed by MMCAnCE_DATA. If the transfer block size is set to 4 × n + 1 or 4 × n + 3, access should be made for 4 × (n + 1) bytes in 32-bit access. (n = 0, 1, 2, …, 127).

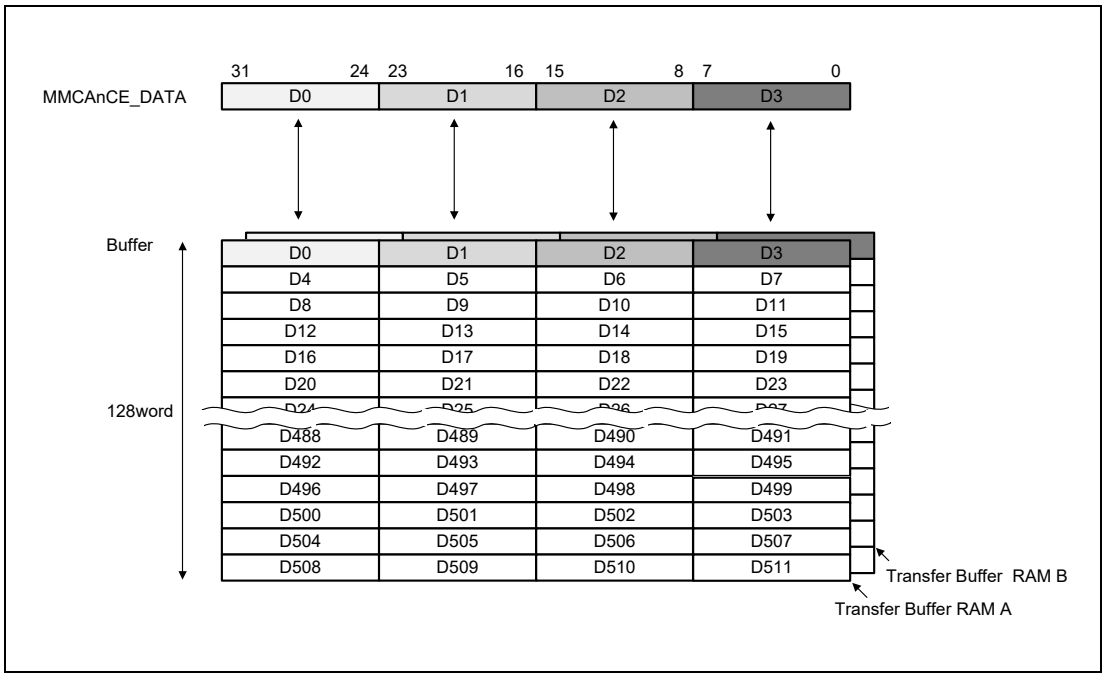


Figure 18.6 Double Buffer Structure

This module has the buffer access select function that allows byte-wise swapping of data when the buffer is accessed by writing to or reading from MMCAnCE_DATA. This function is enabled by the setting of MMCAnCE_BUF_ACC. The figure below shows the specification of 32-bit access.

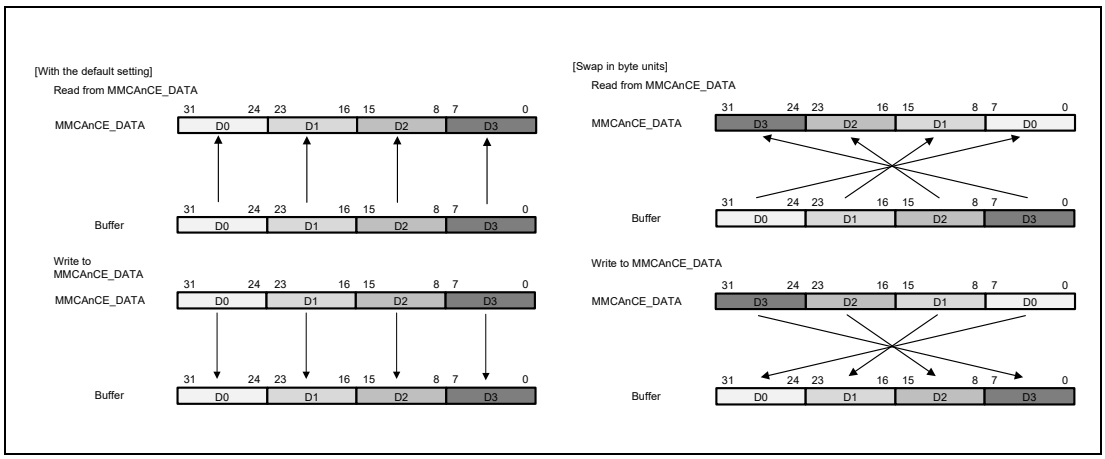


Figure 18.7 Specification of Byte-Swapping

18.4.3.4 Automatic CMD12 Issuance

This module has a function that automatically issues CMD12 when multi-block transfer is performed with the CMD12EN bit in MMCAnCE_CMD_SET set to 1.

The figure below shows the timing of automatic CMD12 issuance in multi-block read. CMD12 is issued so that the end bit of the command is sent two bits before the end bit of the data during reception of the last block.

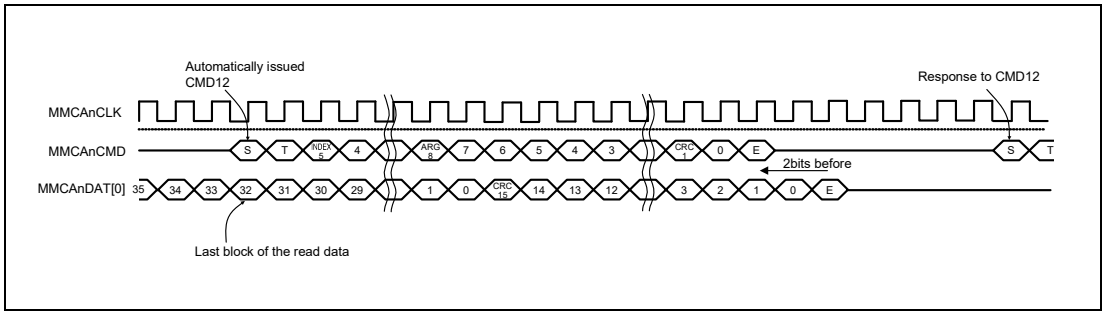


Figure 18.8 Timing of Automatically-Issued CMD12 in Multi-Block Read (1-Bit Mode)

The figure below shows the timing of automatic CMD12 issuance in multi-block write. CMD12 is issued after the data busy after transmission of the last block has ended.

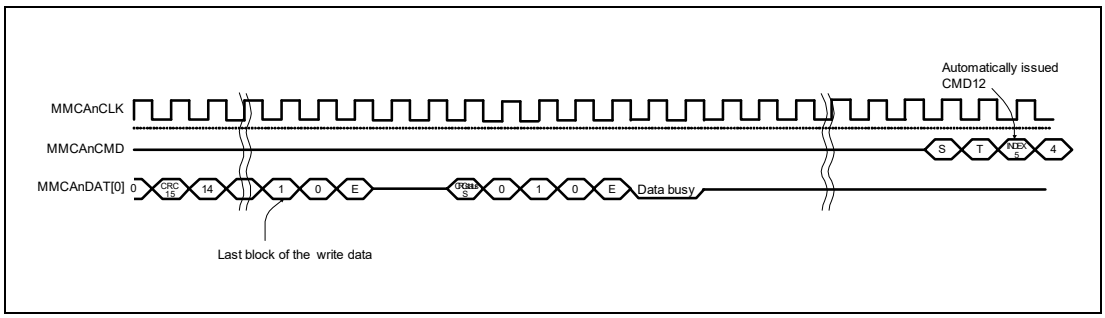


Figure 18.9 Timing of Automatically-Issued CMD12 in Multi-Block Write (1-Bit Mode)

The argument for the automatically-issued CMD12 is set in MMCAnCE_ARG_CMD12. Bits [39:8] of the response to the CMD12 is stored in MMCAnCE_RESP_CMD12. “Busy” is involved in response reception.

18.4.3.5 MMC Clock Frequency in Boot Operations

The figure below shows the timing for changing the MMC clock frequency in boot operations. In boot operations, the MMC clock frequency can be switched to the value corresponding to the setting of the BTCLKDIV bits of MMCAnCE_BOOT 74 MMC clock cycles after MMCAnCMD is modified from 1 to 0. Alternatively, it can be switched to the value corresponding to the setting of the CLKDIV bits of MMCAnCE_CLK_CTRL 48 MMC clock cycles after MMCAnCMD is modified from 0 to 1.

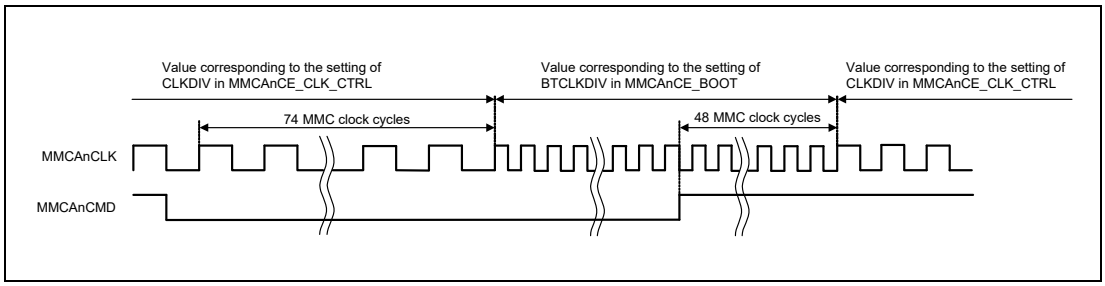


Figure 18.10 MMC Clock Frequency in Boot Operations

18.4.3.6 High Priority Interrupt (HPI)

The HPI should be processed according to the following procedure. Refer to **Section 18.3, Registers**, and **Section 18.4.4, Example of Setting**.

(a) To execute the HPI during a write to the device

- Terminate the command sequence forcibly.
- Wait until the CMDSEQ bit in MMCAnCE_HOST_STS1 becomes 0.
- Issue CMD12 (R1) to cause a device state transition from rcv to prg. If the device is already in the prg state here, the device does not output a response.
- Issue CMD13 (R1).
- Issue the HPI command.*1

(b) To execute the HPI in the response busy state not during a write to the device

- Terminate the command sequence forcibly.
- Wait until the CMDSEQ bit in MMCAnCE_HOST_STS1 turns 0.
- Issue CMD13 (R1).
- Issue the HPI command.*1

HPIs in the sequence of CMD6, CMD24, CMD25 (pre-defined), and CMD38 are supported.

Note 1. CMD12 (R1b) or CMD13 (R1b) differs depending on the e-MMC connected.

18.4.3.7 Background Operation

The background operation should be processed according to the following procedure. Refer to **Section 18.3, Registers**, and **Section 18.4.4, Example of Setting**.

To execute background operation, issue CMD6 (R1) to write to the BKOPS_START byte in the EXT_CSD register of the device.

To confirm that background operation has been completed, issue CMD6 (R1) and then CMD13 (R1) to check the device state, or poll the MMCAnDAT[0]. If the device state is tran after issuing CMD13 (R1) or MMCAnDAT[0] is high, background operation has been completed.

To suspend background operation, use the HPI (see **Section 18.4.3.6, High Priority Interrupt (HPI)**).

18.4.3.8 Handling of This Module in the Case of Error/Timeout

This module may not be stopped when an error has occurred. If an error has occurred and the CMDSEQ bit in MMCAnCE_HOST_STS1 is still indicating that the command sequence is in progress, terminate the sequence forcibly and execute software reset. Note that the data for transmission or received data that had been stored in the buffers at the time of error occurrence are not guaranteed.

This module is not stopped when a timeout has occurred. Before issuing the next command after the timeout has occurred, terminate the sequence forcibly and execute software reset.

18.4.4 Example of Setting

This section shows the procedures for executing typical command sequences.

18.4.4.1 Legends

The figure below shows the legends for the symbols used in the figures in the following subsections.

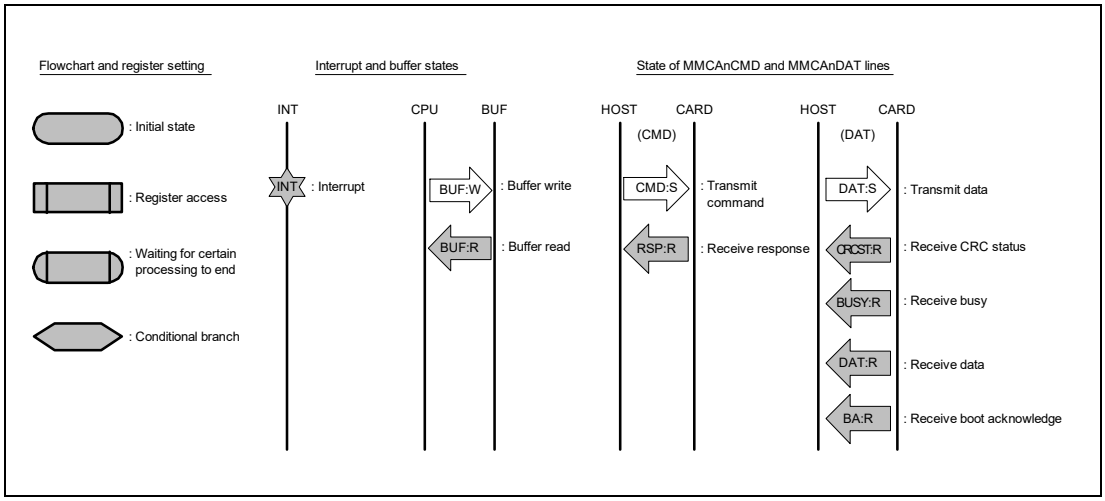


Figure 18.11 Legends for Symbols Used in Figures

18.4.4.2 Command Transmission

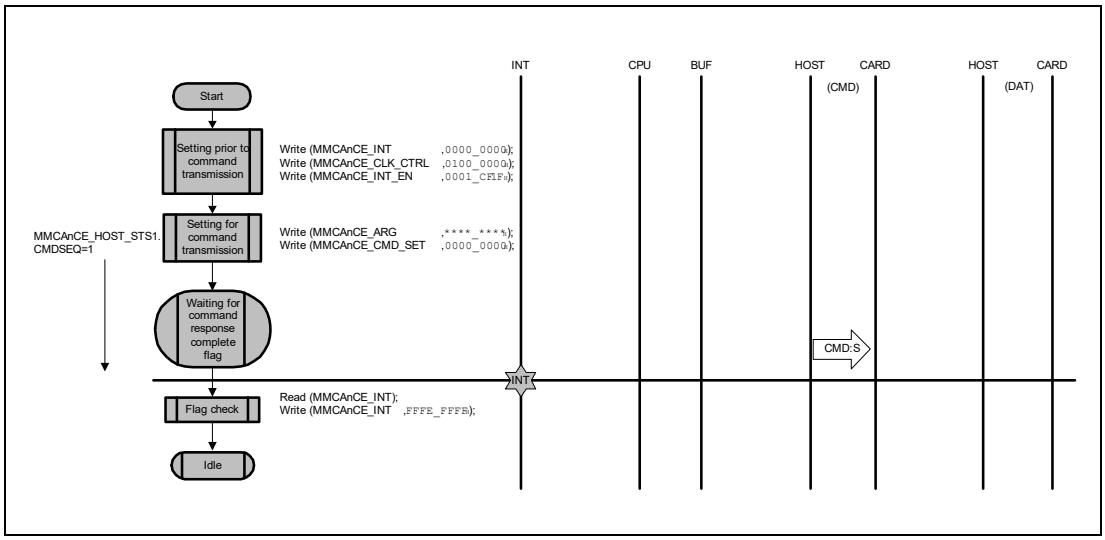


Figure 18.12 Command Transmission (CMD0)

18.4.4.3 Command Transmission → Response Reception

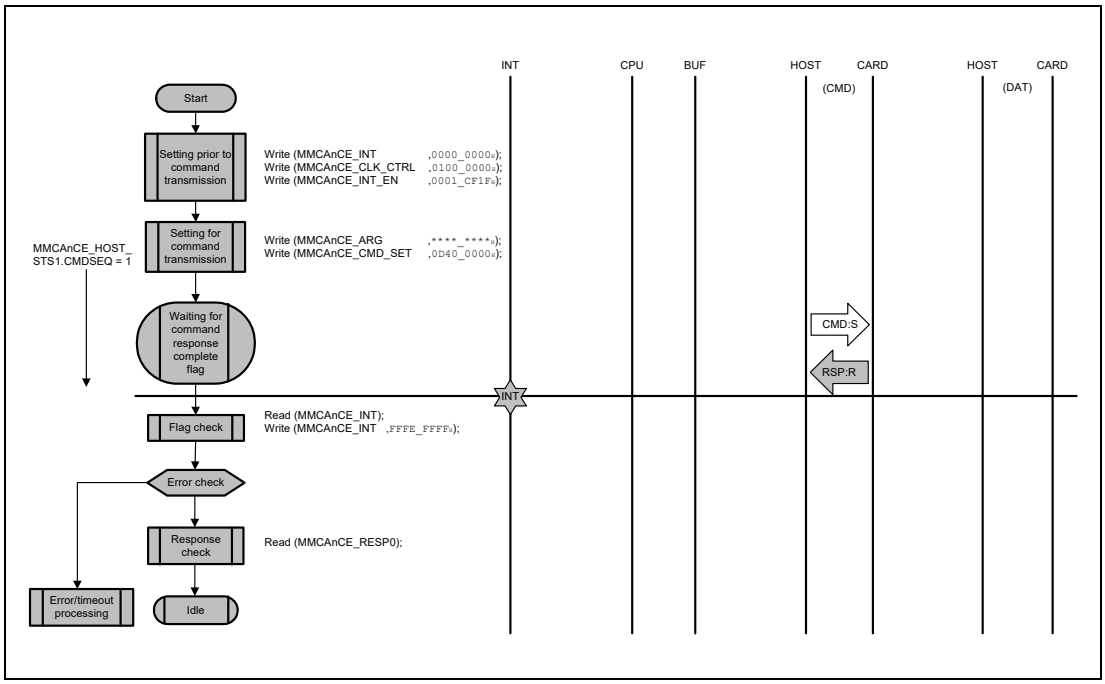


Figure 18.13 Command Transmission to Response Reception (CMD13)

18.4.4.4 Command Transmission → Response Reception (with Response Busy)

– When the busy time period is less than the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL

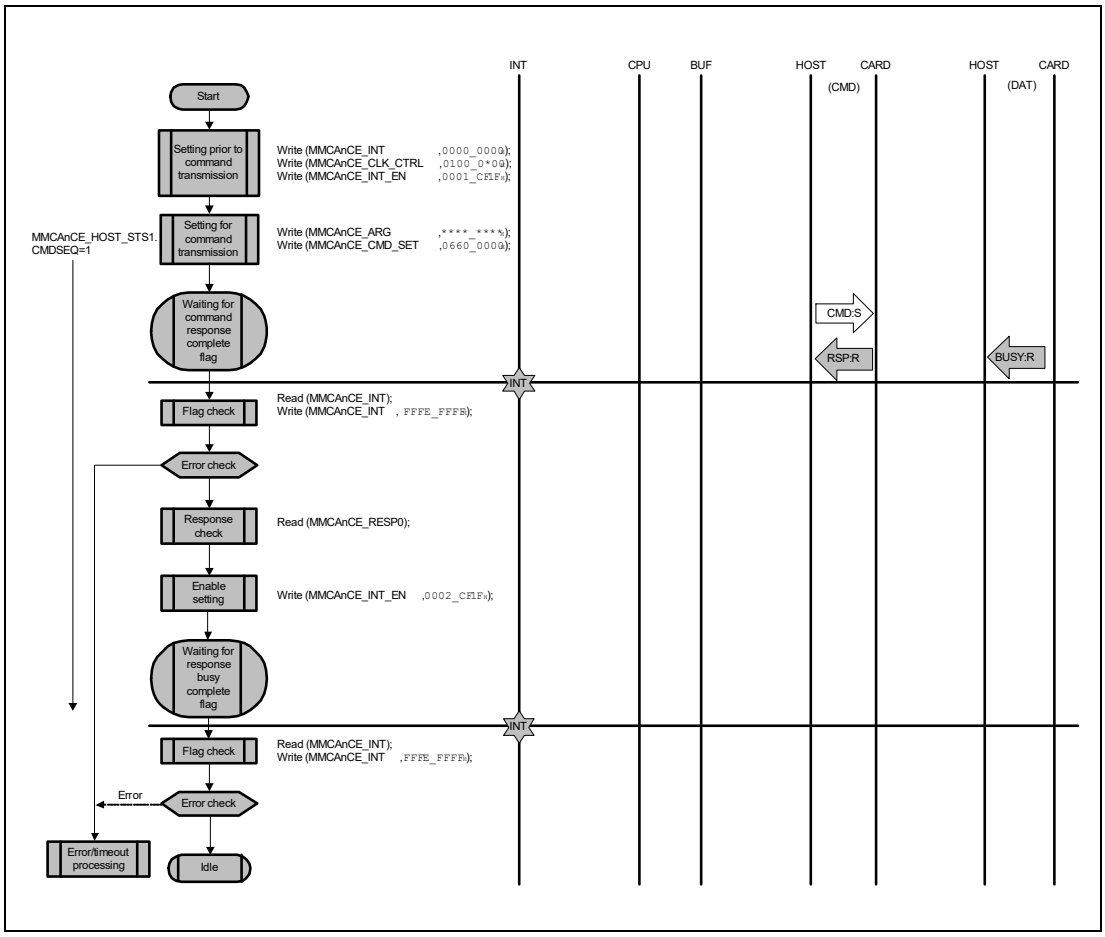


Figure 18.14 Command Transmission to Response Reception (with Response Busy) (CMD6)

– When the busy time period may be equal to or beyond the period set by the SRBSYTO bits in MMCAnCE_CLK_CTRL

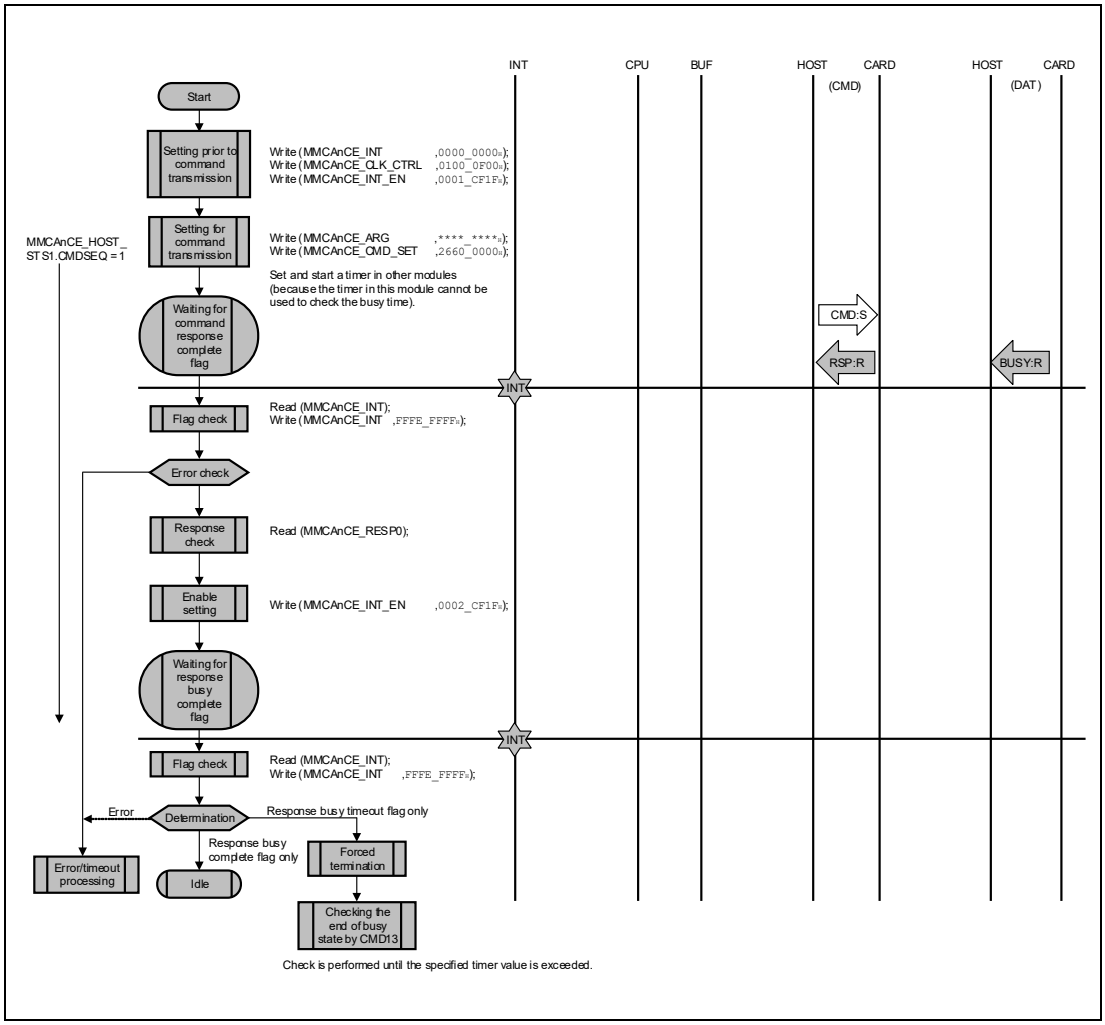


Figure 18.15 Command Transmission to Response Reception (with Response Busy) (CMD38)

18.4.4.5 Single-Block Read

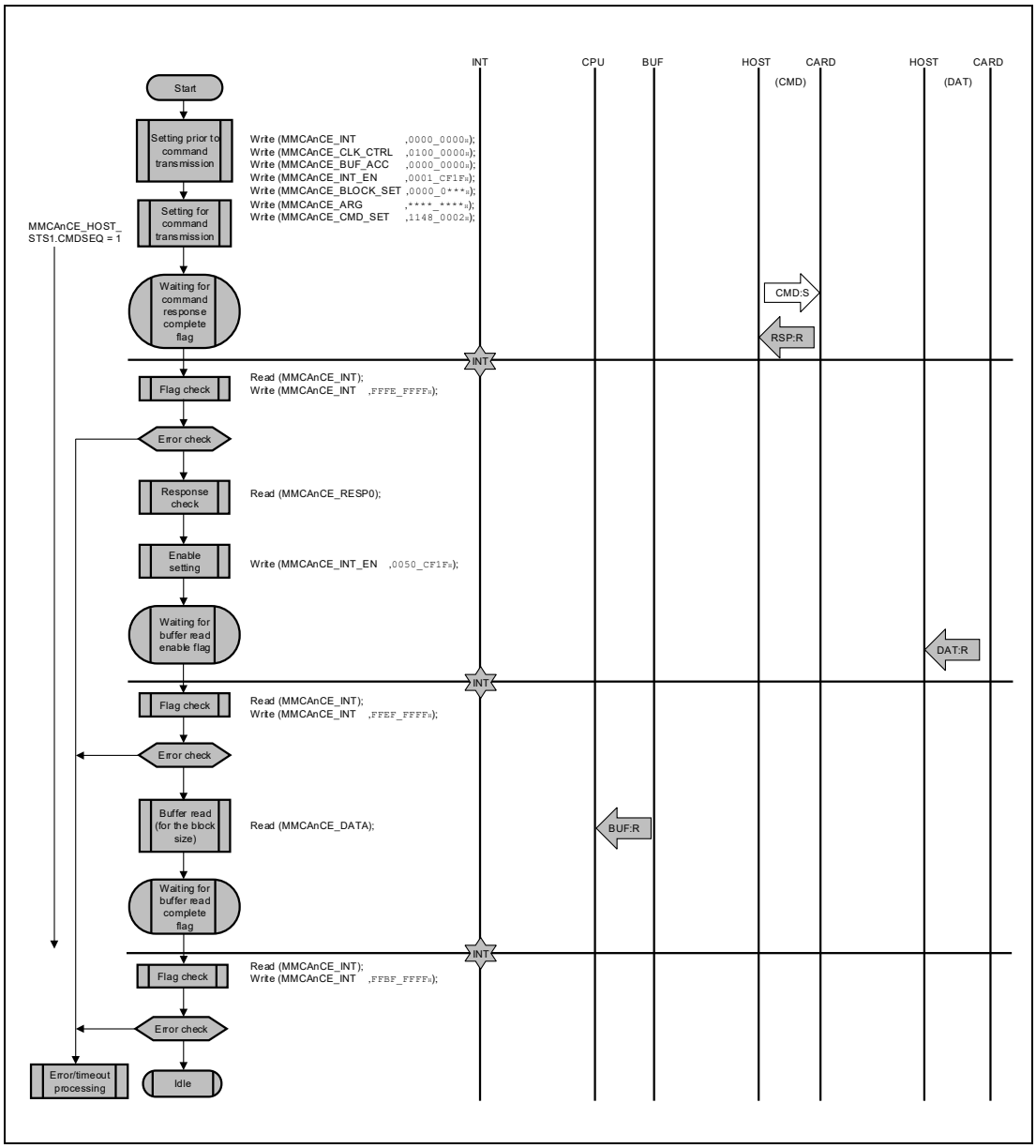


Figure 18.16 Single-Block Read (CMD17)

18.4.4.6 Multi-Block Read

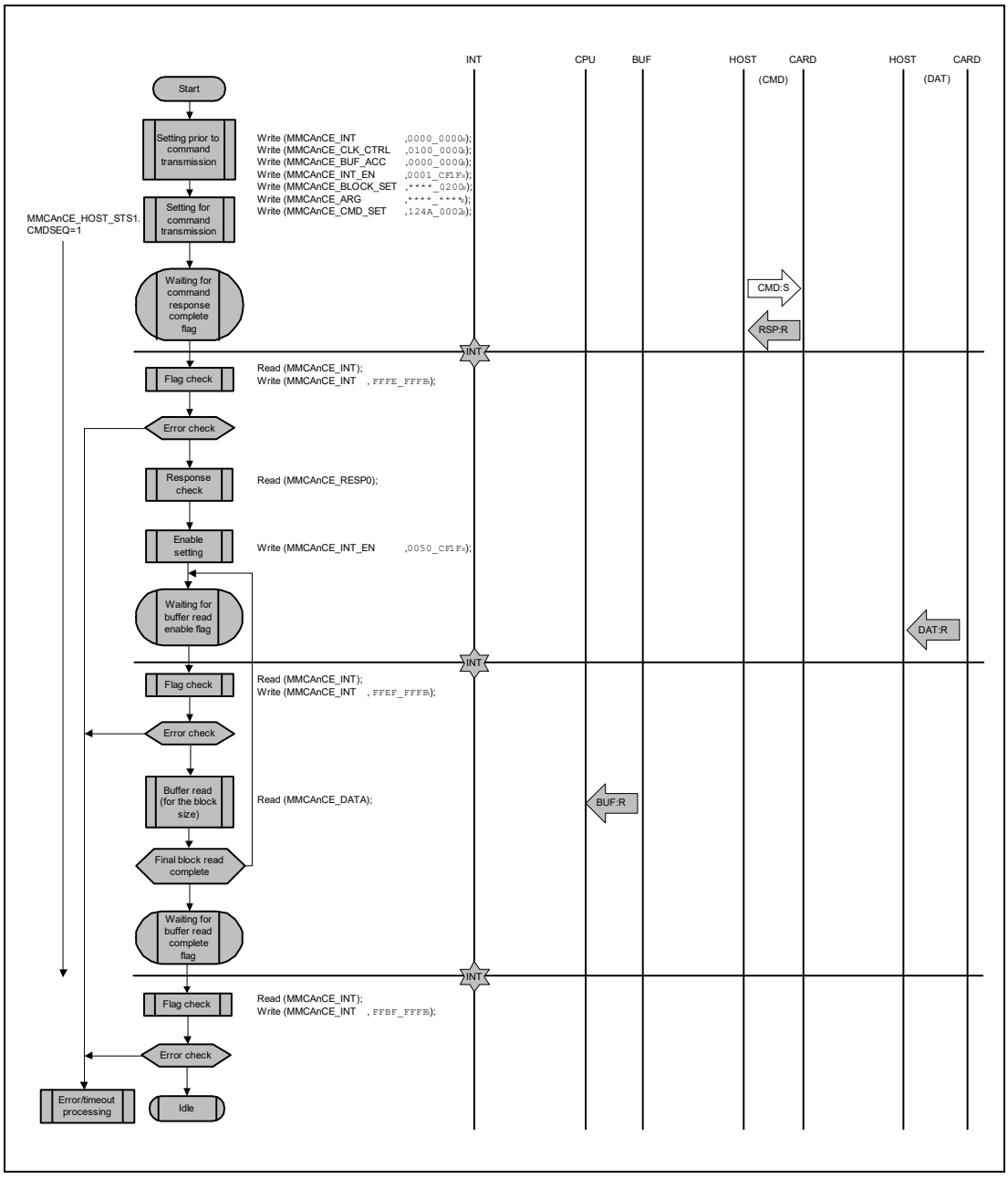


Figure 18.17 Multi-Block Read (CMD18 Pre-Defined)

18.4.4.7 Multi-Block Read (with Automatic CMD12 Issuance)

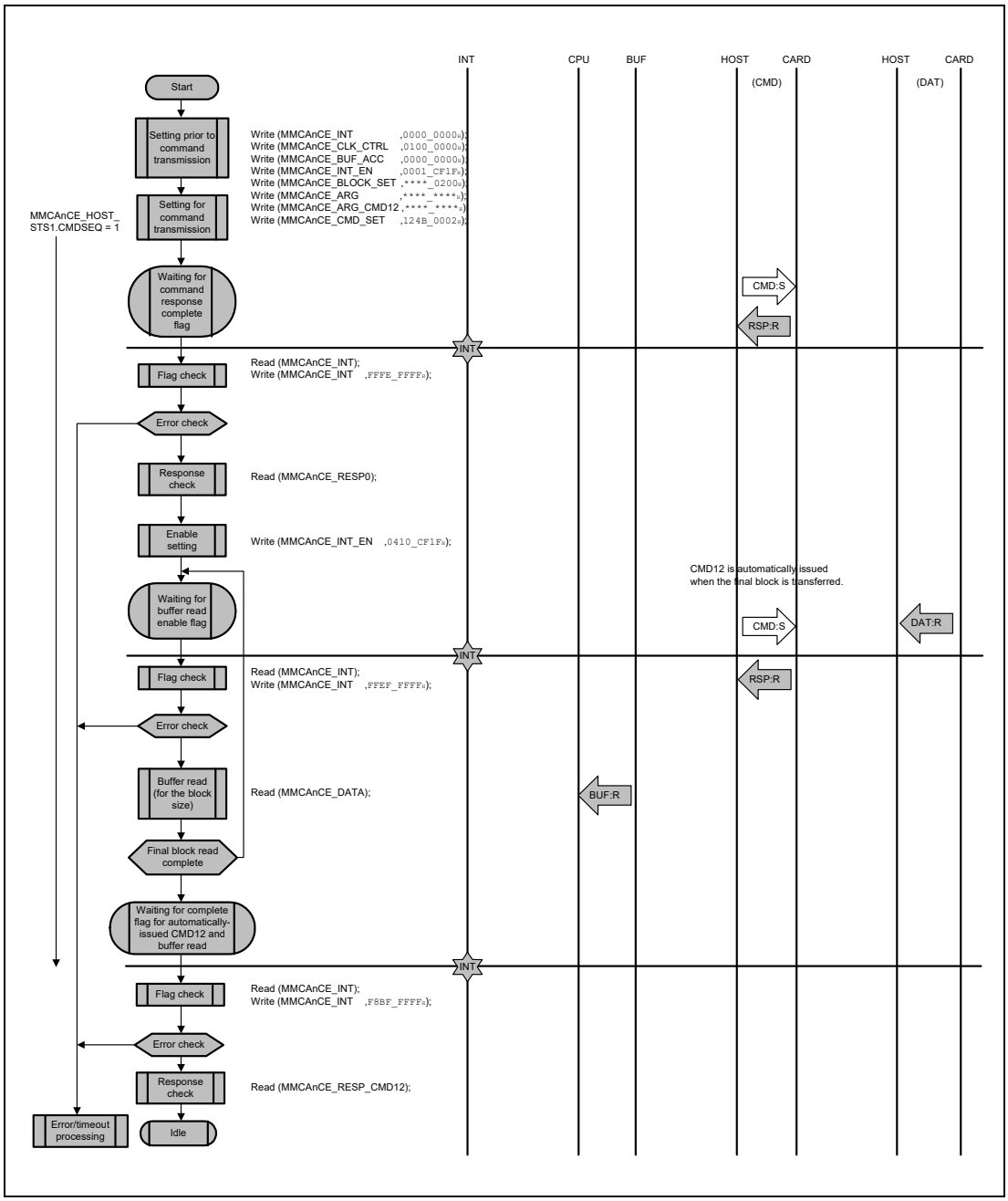


Figure 18.18 Multi-Block Read (with Automatic CMD12 Issuance) (CMD18 Open-Ended)

18.4.4.8 Single-Block Write

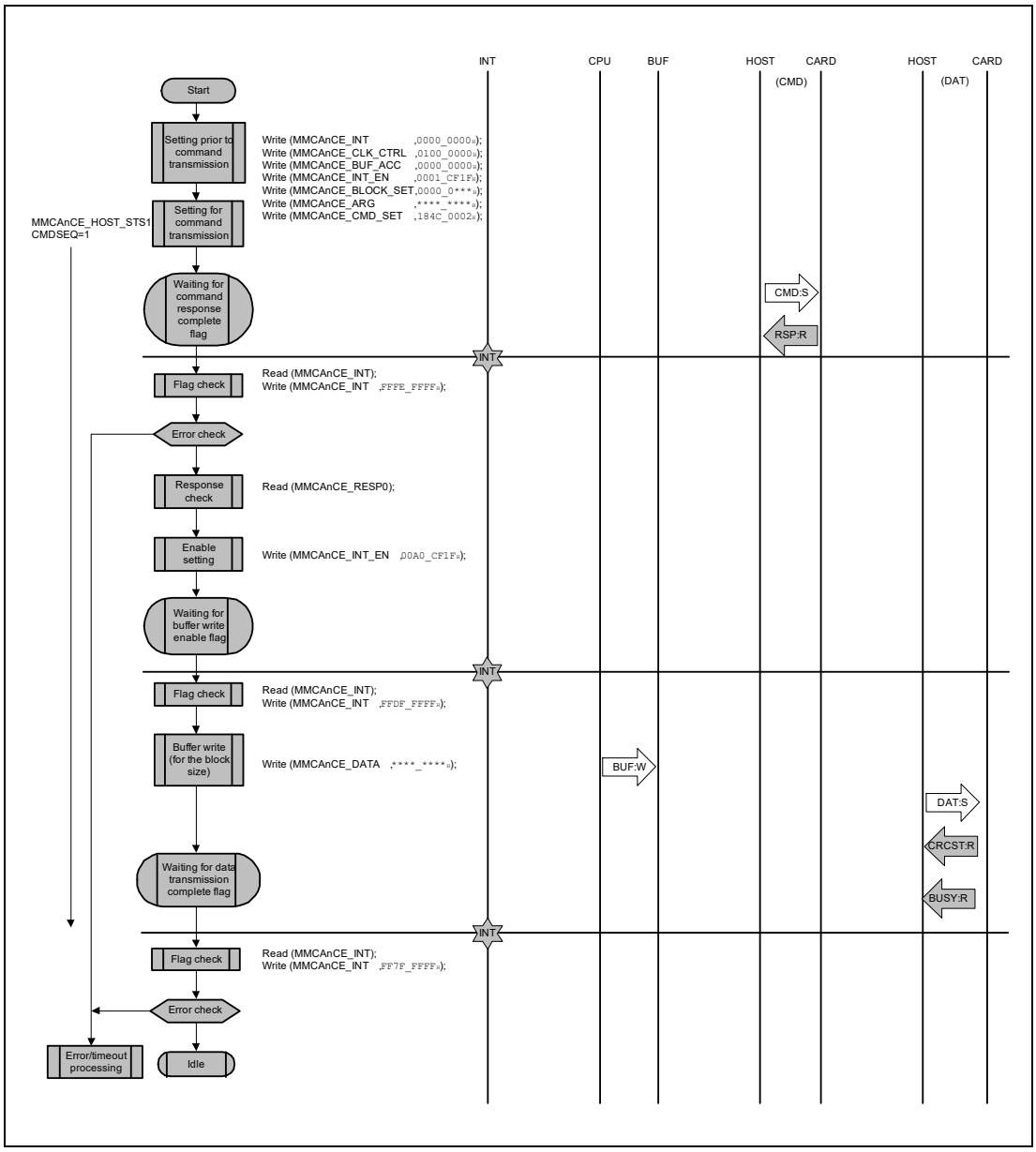


Figure 18.19 Single-Block Write (CMD24)

18.4.4.9 Multi-Block Write

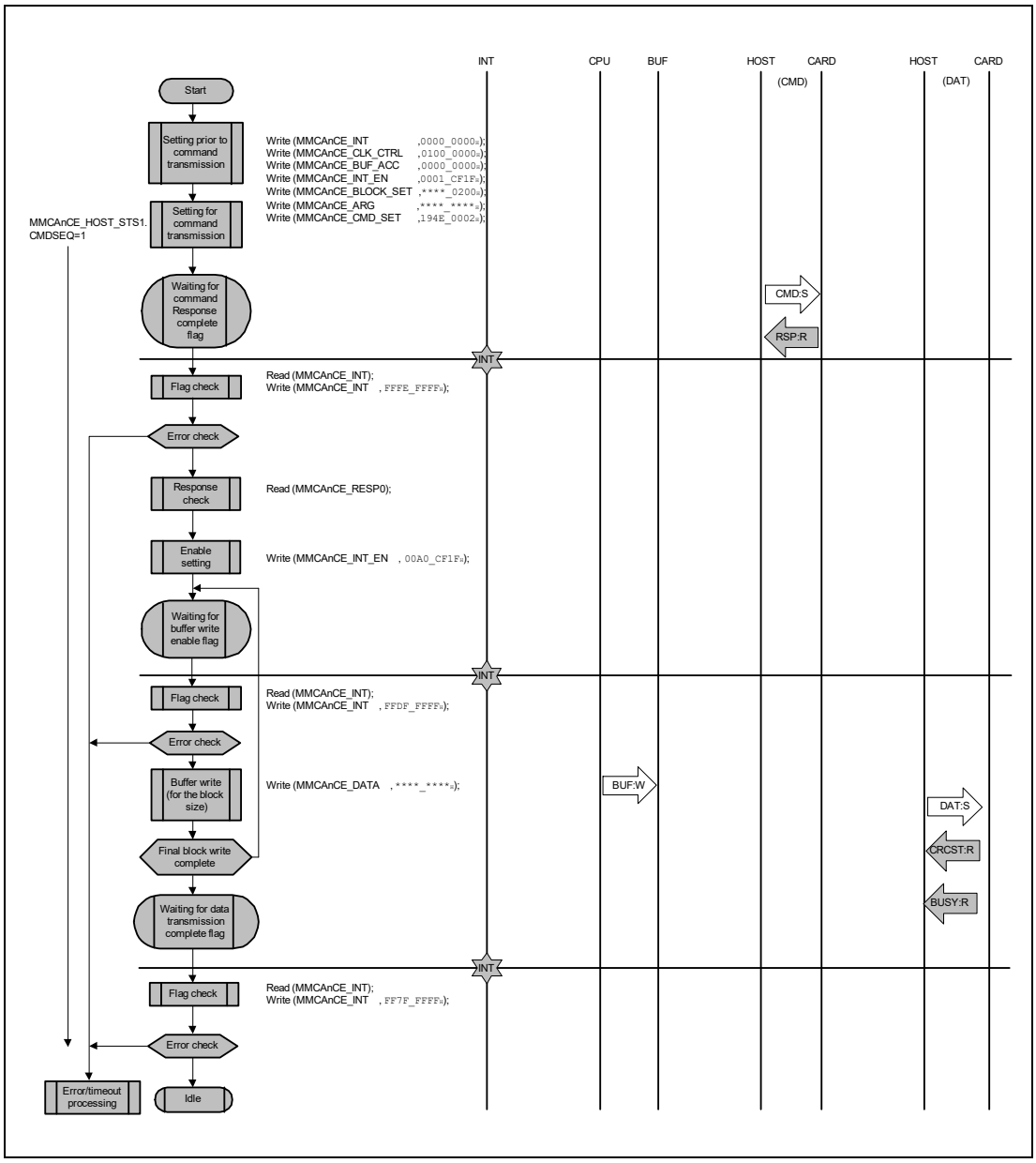


Figure 18.20 Multi-Block Write (CMD25 Pre-Defined)

18.4.4.10 Multi-Block Write (with Automatic CMD12 Issuance)

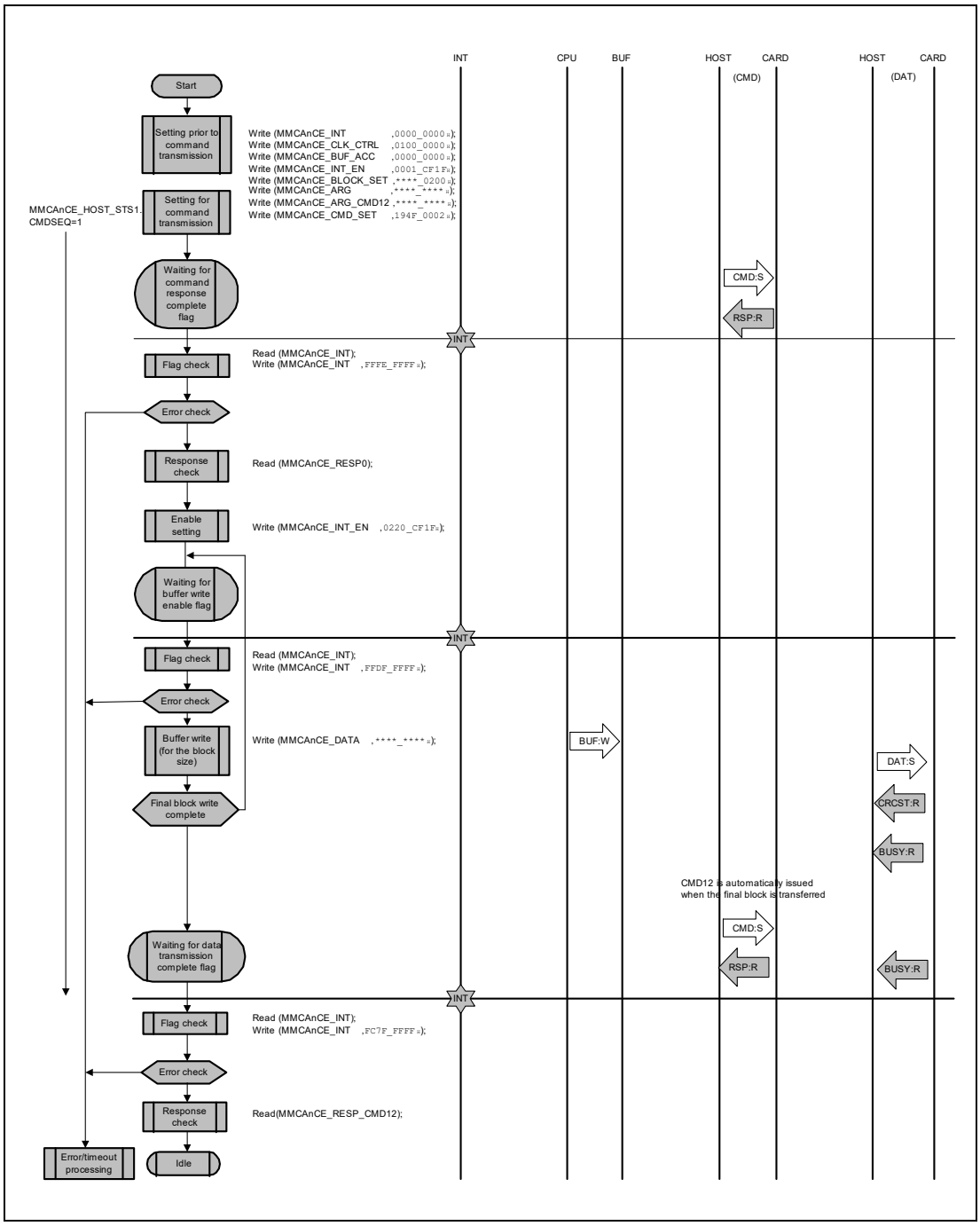


Figure 18.21 Multi-Block Write (with Automatic CMD12 Issuance) (CMD25 Open-Ended)

18.4.4.11 Boot Operations

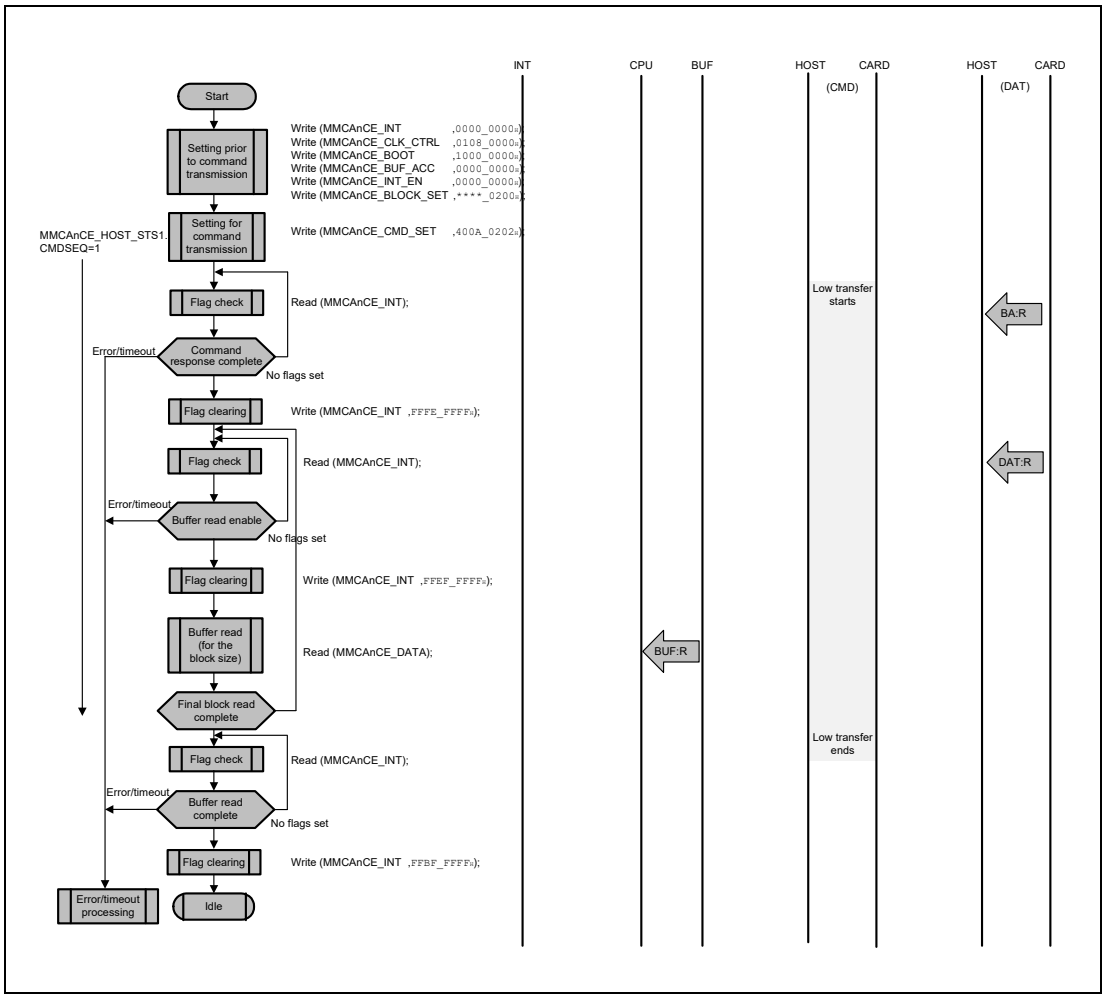


Figure 18.22 Boot Operations (with Boot Acknowledge)

18.4.4.12 Forcible Termination

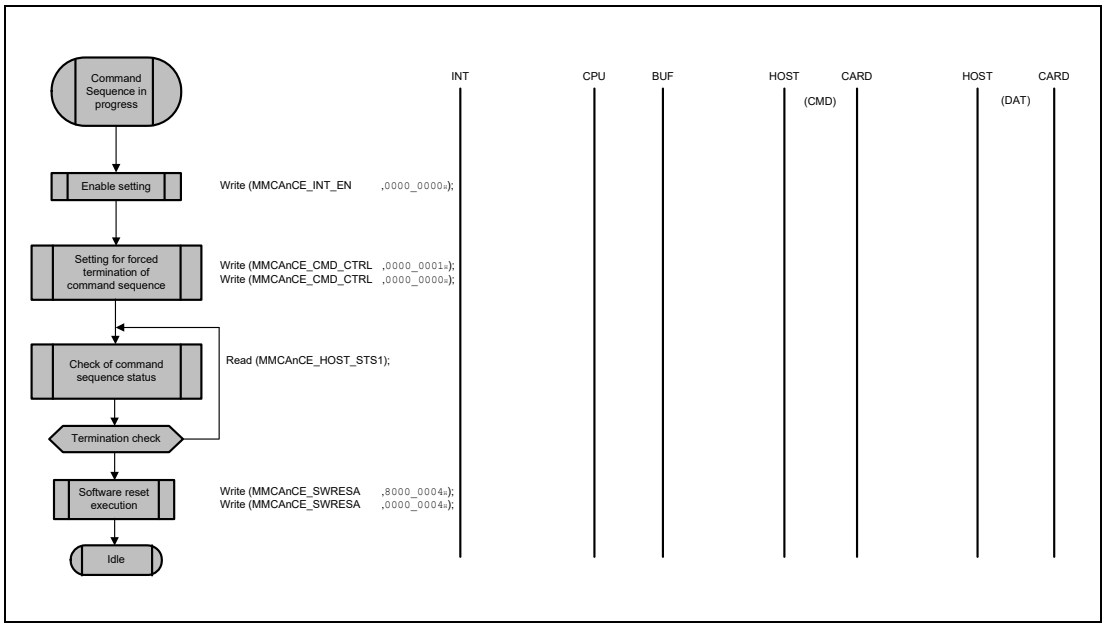


Figure 18.23 Forcible Termination

18.4.4.13 Setting Values of MMCAnCE_CMD_SET

The table below lists the setting values required to issue commands.

Table 18.29 Setting Values of MMCAnCE_CMD_SET

Command	Response	MMCAnCE_CMD_SET																							Remark
		Reserved	BOOT	CMD[5:0]	RTYP[1:0]	RBSY	Reserved	WDAT	DWEN	CMLTE	CMD12EN	RIDXC[1:0]	RCRC7C[1:0]	Reserved	CRC16C	BOOTACK	CRCSTE	TBIT	OPDM	Reserved	Reserved	SBIT	Reserved	DATW[1:0]	
CMD0	—	0	0	000000	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD1	R3	0	0	000001	01	0	0	0	0	0	0	01	01	0	0	0	0	0	0	0	0	0	0	00	
CMD2	R2	0	0	000010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD3	R1	0	0	000011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD4	—	0	0	000100	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD5	R1b	0	0	000101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD6	R1	0	0	000110	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Background operation
	R1b	0	0	000110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD7	R1	0	0	000111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	000111	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD8	R1	0	0	001000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD9	R2	0	0	001001	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD10	R2	0	0	001010	10	0	0	0	0	0	0	01	10	0	0	0	0	0	0	0	0	0	0	00	
CMD12	R1	0	0	001100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD13	R1	0	0	001101	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
	R1b	0	0	001101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD14	R1	0	0	001110	01	0	0	1	0	0	0	00	00	0	1	0	0	0	0	0	0	1	0	**	
CMD15	—	0	0	001111	00	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD16	R1	0	0	010000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD17	R1	0	0	010001	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD18	R1	0	0	010010	01	0	0	1	0	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	010010	01	0	0	1	0	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD19	R1	0	0	010011	01	0	0	1	1	0	0	00	00	0	0	0	1	0	0	0	0	0	0	**	
CMD23	R1	0	0	010111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD24	R1	0	0	011000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD25	R1	0	0	011001	01	0	0	1	1	1	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Pre-defined
	R1	0	0	011001	01	0	0	1	1	1	1	00	00	0	0	0	0	0	0	0	0	0	0	**	Open-ended
CMD26	R1	0	0	011010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD27	R1	0	0	011011	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD28	R1b	0	0	011100	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD29	R1b	0	0	011101	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD30	R1	0	0	011110	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD31	R1	0	0	011111	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD35	R1	0	0	100011	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD36	R1	0	0	100100	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD38	R1b	0	0	100110	01	1	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD39	R4	0	0	100111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD40	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	Send CMD
	R5	0	0	101000	01	0	0	0	0	0	0	00	00	0	0	0	0	1	1	0	0	0	0	00	Send RSP
CMD42	R1	0	0	101010	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	
CMD55	R1	0	0	110111	01	0	0	0	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	00	
CMD56	R1	0	0	111000	01	0	0	1	0	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Read
	R1	0	0	111000	01	0	0	1	1	0	0	00	00	0	0	0	0	0	0	0	0	0	0	**	Write
Boot operation		0	1	000000	00	0	0	1	0	1	0	00	00	0	0	*	0	0	0	0	0	0	0	**	

18.5 Detection and Correction of Errors in MMCA RAM

For details of this ECC function, see **Section 44.3.10, ECC for Peripheral RAM**.

Section 19 Multichannel Serial Peripheral Interface (MSPI)

This section contains a generic description of the Multi-channel Serial Peripheral Interface(MSPI).

19.1 Features of MSPI for RH850/U2A-EVA

19.1.1 Number of Units and Channels

This microcontroller has the following number of MSPI units.

Table 19.1 Number of Units

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)	RH850/U2A6 (292 pins)	RH850/U2A6 (176 pins)	RH850/U2A6 (156 pins)	RH850/U2A6 (144 pins)
Number of Units	10 (n = 0 to 9)	10 (n = 0 to 9)	9 (n = 0 to 8)	6 (n = 0 to 5)	9 (n = 0 to 8)	6 (n = 0 to 5)	6 (n = 0 to 5)	6 (n = 0 to 5)	6 (n = 0 to 5)	4 (n = 0 to 3
Name	MSPIn									

Table 19.2 Indices

Index	Description
n	Throughout this section, the individual MSPI units are identified by the index “n” (n = 0 to 9) For example, MSPI0CTL0 is the control register 0 for unit 0.
m	MSPIn has a maximum of 8 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 7). For example, MSPInCFG00 is the configuration register 0 for channel 0.
x	MSPIn has a maximum of 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index “x”: that is, CSx denotes a non-specified chip select signal.
k	Sets of registers where each has the same function are identified by the index “k” (k = 0 to 3). For example, the MSPInINTFk register (k = 0 to 3)

Table 19.3 Number of Transfer Control Channels and Chip Select Index

Unit Name	Number of Transfer Control Channels	Chip Select Index
MSPI0	8	CSx (x = 0 to 7)
MSPI1	8	CSx (x = 0 to 7)
MSPI2	8	CSx (x = 0 to 7)
MSPI3	8	CSx (x = 0 to 7)
MSPI4	8	CSx (x = 0 to 7)
MSPI5	4	CSx (x = 0 to 3)
MSPI6	4	CSx (x = 0 to 3)
MSPI7	4	CSx (x = 0 to 3)
MSPI8	4	CSx (x = 0 to 3)
MSPI9	4	CSx (x = 0 to 3)

19.1.2 Register Base Addresses

MSPIn register addresses are given as offsets from the base addresses.

MSPIn base addresses are listed in the following table.

Table 19.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
MSPI0_base	FFC7 6000$_H$	P-Bus Group 4
MSPI1_base	FFC7 E000$_H$	P-Bus Group 5
MSPI2_base	FFD8 0000$_H$	P-Bus Group 4
MSPI3_base	FFCC 0000$_H$	P-Bus Group 5
MSPI4_base	FFD8 2000$_H$	P-Bus Group 4
MSPI5_base	FFCC 4000$_H$	P-Bus Group 5
MSPI6_base	FFD8 4000$_H$	P-Bus Group 4
MSPI7_base	FFCC 8000$_H$	P-Bus Group 5
MSPI8_base	FFD8 6000$_H$	P-Bus Group 4
MSPI9_base	FFCC C000$_H$	P-Bus Group 5
MSPI0_INTF_base	FFC7 5740$_H$	P-Bus Group 4
MSPI1_INTF_base	FFC7 C840$_H$	P-Bus Group 5
MSPI2_INTF_base	FFC7 5840$_H$	P-Bus Group 4
MSPI3_INTF_base	FFC7 CA40$_H$	P-Bus Group 5
MSPI4_INTF_base	FFC7 5940$_H$	P-Bus Group 4
MSPI5_INTF_base	FFC7 CC40$_H$	P-Bus Group 5
MSPI6_INTF_base	FFC7 5A40$_H$	P-Bus Group 4
MSPI7_INTF_base	FFC7 CE40$_H$	P-Bus Group 5
MSPI8_INTF_base	FFC7 5B40$_H$	P-Bus Group 4
MSPI9_INTF_base	FFC7 D040$_H$	P-Bus Group 5
MSPITG_base	FFC7 5D00$_H$	P-Bus Group 4

19.1.3 Clock Supply

The MSPI clock supplies are shown in the following **Table 19.5**.

Table 19.5 MSPI Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
MSPIn	PCLK	CLK_HSB
	MSPInCLK(MSPI communication clock)	CLK_MSPI

19.1.4 Interrupt Requests and Error Notifications

MSPIn interrupt requests are listed in the following table.

Table 19.6 Interrupt Requests (1/6)

Unit Interrupt Name	Description	Interrupt Number
MSPI0		
INTMSPI0TX0	Transmit status interrupt for channel 0	244, 250
INTMSPI0TX1	Transmit status interrupt for channel 1	246, 250
INTMSPI0TX2	Transmit status interrupt for channel 2	248, 250
INTMSPI0TX3	Transmit status interrupt for channel 3	250
INTMSPI0TX4	Transmit status interrupt for channel 4	250
INTMSPI0TX5	Transmit status interrupt for channel 5	250
INTMSPI0TX6	Transmit status interrupt for channel 6	250
INTMSPI0TX7	Transmit status interrupt for channel 7	250
INTMSPI0RX0	Receive status interrupt for channel 0	245, 251
INTMSPI0RX1	Receive status interrupt for channel 1	247, 251
INTMSPI0RX2	Receive status interrupt for channel 2	249, 251
INTMSPI0RX3	Receive status interrupt for channel 3	251
INTMSPI0RX4	Receive status interrupt for channel 4	251
INTMSPI0RX5	Receive status interrupt for channel 5	251
INTMSPI0RX6	Receive status interrupt for channel 6	251
INTMSPI0RX7	Receive status interrupt for channel 7	251
INTMSPI0FE0	Frame count end interrupt for channel 0	252
INTMSPI0FE1	Frame count end interrupt for channel 1	252
INTMSPI0FE2	Frame count end interrupt for channel 2	252
INTMSPI0FE3	Frame count end interrupt for channel 3	252
INTMSPI0FE4	Frame count end interrupt for channel 4	252
INTMSPI0FE5	Frame count end interrupt for channel 5	252
INTMSPI0FE6	Frame count end interrupt for channel 6	252
INTMSPI0FE7	Frame count end interrupt for channel 7	252
INTMSPI0ERR0	Error interrupt for channel 0	253
INTMSPI0ERR1	Error interrupt for channel 1	253
INTMSPI0ERR2	Error interrupt for channel 2	253
INTMSPI0ERR3	Error interrupt for channel 3	253
INTMSPI0ERR4	Error interrupt for channel 4	253
INTMSPI0ERR5	Error interrupt for channel 5	253
INTMSPI0ERR6	Error interrupt for channel 6	253
INTMSPI0ERR7	Error interrupt for channel 7	253

Table 19.6 Interrupt Requests (2/6)

Unit Interrupt Name	Description	Interrupt Number
MSPI1		
INTMSPI1TX0	Transmit status interrupt for channel 0	254, 260
INTMSPI1TX1	Transmit status interrupt for channel 1	256, 260
INTMSPI1TX2	Transmit status interrupt for channel 2	258, 260
INTMSPI1TX3	Transmit status interrupt for channel 3	260
INTMSPI1TX4	Transmit status interrupt for channel 4	260
INTMSPI1TX5	Transmit status interrupt for channel 5	260
INTMSPI1TX6	Transmit status interrupt for channel 6	260
INTMSPI1TX7	Transmit status interrupt for channel 7	260
INTMSPI1RX0	Receive status interrupt for channel 0	255, 261
INTMSPI1RX1	Receive status interrupt for channel 1	257, 261
INTMSPI1RX2	Receive status interrupt for channel 2	259, 261
INTMSPI1RX3	Receive status interrupt for channel 3	261
INTMSPI1RX4	Receive status interrupt for channel 4	261
INTMSPI1RX5	Receive status interrupt for channel 5	261
INTMSPI1RX6	Receive status interrupt for channel 6	261
INTMSPI1RX7	Receive status interrupt for channel 7	261
INTMSPI1FE0	Frame count end interrupt for channel 0	262
INTMSPI1FE1	Frame count end interrupt for channel 1	262
INTMSPI1FE2	Frame count end interrupt for channel 2	262
INTMSPI1FE3	Frame count end interrupt for channel 3	262
INTMSPI1FE4	Frame count end interrupt for channel 4	262
INTMSPI1FE5	Frame count end interrupt for channel 5	262
INTMSPI1FE6	Frame count end interrupt for channel 6	262
INTMSPI1FE7	Frame count end interrupt for channel 7	262
INTMSPI1ERR0	Error interrupt for channel 0	263
INTMSPI1ERR1	Error interrupt for channel 1	263
INTMSPI1ERR2	Error interrupt for channel 2	263
INTMSPI1ERR3	Error interrupt for channel 3	263
INTMSPI1ERR4	Error interrupt for channel 4	263
INTMSPI1ERR5	Error interrupt for channel 5	263
INTMSPI1ERR6	Error interrupt for channel 6	263
INTMSPI1ERR7	Error interrupt for channel 7	263
MSPI2		
INTMSPI2TX0	Transmit status interrupt for channel 0	264
INTMSPI2TX1	Transmit status interrupt for channel 1	264
INTMSPI2TX2	Transmit status interrupt for channel 2	264
INTMSPI2TX3	Transmit status interrupt for channel 3	264
INTMSPI2TX4	Transmit status interrupt for channel 4	264
INTMSPI2TX5	Transmit status interrupt for channel 5	264
INTMSPI2TX6	Transmit status interrupt for channel 6	264
INTMSPI2TX7	Transmit status interrupt for channel 7	264
INTMSPI2RX0	Receive status interrupt for channel 0	265
INTMSPI2RX1	Receive status interrupt for channel 1	265

Table 19.6 Interrupt Requests (3/6)

Unit Interrupt Name	Description	Interrupt Number
INTMSPI2RX2	Receive status interrupt for channel 2	265
INTMSPI2RX3	Receive status interrupt for channel 3	265
INTMSPI2RX4	Receive status interrupt for channel 4	265
INTMSPI2RX5	Receive status interrupt for channel 5	265
INTMSPI2RX6	Receive status interrupt for channel 6	265
INTMSPI2RX7	Receive status interrupt for channel 7	265
INTMSPI2FE0	Frame count end interrupt for channel 0	266
INTMSPI2FE1	Frame count end interrupt for channel 1	266
INTMSPI2FE2	Frame count end interrupt for channel 2	266
INTMSPI2FE3	Frame count end interrupt for channel 3	266
INTMSPI2FE4	Frame count end interrupt for channel 4	266
INTMSPI2FE5	Frame count end interrupt for channel 5	266
INTMSPI2FE6	Frame count end interrupt for channel 6	266
INTMSPI2FE7	Frame count end interrupt for channel 7	266
INTMSPI2ERR0	Error interrupt for channel 0	267
INTMSPI2ERR1	Error interrupt for channel 1	267
INTMSPI2ERR2	Error interrupt for channel 2	267
INTMSPI2ERR3	Error interrupt for channel 3	267
INTMSPI2ERR4	Error interrupt for channel 4	267
INTMSPI2ERR5	Error interrupt for channel 5	267
INTMSPI2ERR6	Error interrupt for channel 6	267
INTMSPI2ERR7	Error interrupt for channel 7	267
MSPI3		
INTMSPI3TX0	Transmit status interrupt for channel 0	268
INTMSPI3TX1	Transmit status interrupt for channel 1	268
INTMSPI3TX2	Transmit status interrupt for channel 2	268
INTMSPI3TX3	Transmit status interrupt for channel 3	268
INTMSPI3TX4	Transmit status interrupt for channel 4	268
INTMSPI3TX5	Transmit status interrupt for channel 5	268
INTMSPI3TX6	Transmit status interrupt for channel 6	268
INTMSPI3TX7	Transmit status interrupt for channel 7	268
INTMSPI3RX0	Receive status interrupt for channel 0	269
INTMSPI3RX1	Receive status interrupt for channel 1	269
INTMSPI3RX2	Receive status interrupt for channel 2	269
INTMSPI3RX3	Receive status interrupt for channel 3	269
INTMSPI3RX4	Receive status interrupt for channel 4	269
INTMSPI3RX5	Receive status interrupt for channel 5	269
INTMSPI3RX6	Receive status interrupt for channel 6	269
INTMSPI3RX7	Receive status interrupt for channel 7	269
INTMSPI3FE0	Frame count end interrupt for channel 0	270
INTMSPI3FE1	Frame count end interrupt for channel 1	270
INTMSPI3FE2	Frame count end interrupt for channel 2	270
INTMSPI3FE3	Frame count end interrupt for channel 3	270
INTMSPI3FE4	Frame count end interrupt for channel 4	270

Table 19.6 Interrupt Requests (4/6)

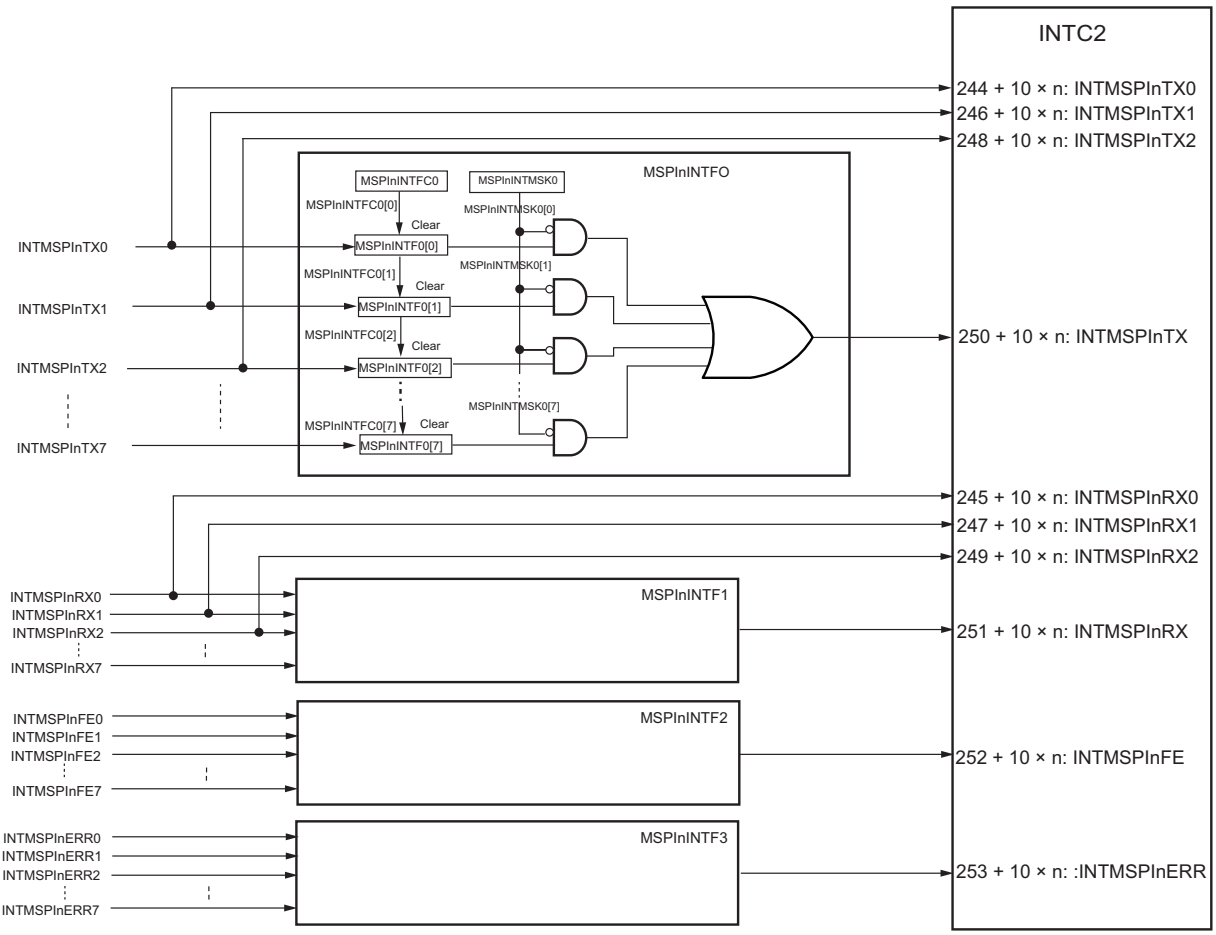
Unit Interrupt Name	Description	Interrupt Number
INTMSPI3FE5	Frame count end interrupt for channel 5	270
INTMSPI3FE6	Frame count end interrupt for channel 6	270
INTMSPI3FE7	Frame count end interrupt for channel 7	270
INTMSPI3ERR0	Error interrupt for channel 0	271
INTMSPI3ERR1	Error interrupt for channel 1	271
INTMSPI3ERR2	Error interrupt for channel 2	271
INTMSPI3ERR3	Error interrupt for channel 3	271
INTMSPI3ERR4	Error interrupt for channel 4	271
INTMSPI3ERR5	Error interrupt for channel 5	271
INTMSPI3ERR6	Error interrupt for channel 6	271
INTMSPI3ERR7	Error interrupt for channel 7	271
MSPI4		
INTMSPI4TX0	Transmit status interrupt for channel 0	272
INTMSPI4TX1	Transmit status interrupt for channel 1	272
INTMSPI4TX2	Transmit status interrupt for channel 2	272
INTMSPI4TX3	Transmit status interrupt for channel 3	272
INTMSPI4TX4	Transmit status interrupt for channel 4	272
INTMSPI4TX5	Transmit status interrupt for channel 5	272
INTMSPI4TX6	Transmit status interrupt for channel 6	272
INTMSPI4TX7	Transmit status interrupt for channel 7	272
INTMSPI4RX0	Receive status interrupt for channel 0	273
INTMSPI4RX1	Receive status interrupt for channel 1	273
INTMSPI4RX2	Receive status interrupt for channel 2	273
INTMSPI4RX3	Receive status interrupt for channel 3	273
INTMSPI4RX4	Receive status interrupt for channel 4	273
INTMSPI4RX5	Receive status interrupt for channel 5	273
INTMSPI4RX6	Receive status interrupt for channel 6	273
INTMSPI4RX7	Receive status interrupt for channel 7	273
INTMSPI4FE0	Frame count end interrupt for channel 0	274
INTMSPI4FE1	Frame count end interrupt for channel 1	274
INTMSPI4FE2	Frame count end interrupt for channel 2	274
INTMSPI4FE3	Frame count end interrupt for channel 3	274
INTMSPI4FE4	Frame count end interrupt for channel 4	274
INTMSPI4FE5	Frame count end interrupt for channel 5	274
INTMSPI4FE6	Frame count end interrupt for channel 6	274
INTMSPI4FE7	Frame count end interrupt for channel 7	274
INTMSPI4ERR0	Error interrupt for channel 0	275
INTMSPI4ERR1	Error interrupt for channel 1	275
INTMSPI4ERR2	Error interrupt for channel 2	275
INTMSPI4ERR3	Error interrupt for channel 3	275
INTMSPI4ERR4	Error interrupt for channel 4	275
INTMSPI4ERR5	Error interrupt for channel 5	275
INTMSPI4ERR6	Error interrupt for channel 6	275
INTMSPI4ERR7	Error interrupt for channel 7	275

Table 19.6 Interrupt Requests (5/6)

Unit Interrupt Name	Description	Interrupt Number
MSPI5		
INTMSPI5TX0	Transmit status interrupt for channel 0	276
INTMSPI5TX1	Transmit status interrupt for channel 1	276
INTMSPI5TX2	Transmit status interrupt for channel 2	276
INTMSPI5TX3	Transmit status interrupt for channel 3	276
INTMSPI5RX0	Receive status interrupt for channel 0	277
INTMSPI5RX1	Receive status interrupt for channel 1	277
INTMSPI5RX2	Receive status interrupt for channel 2	277
INTMSPI5RX3	Receive status interrupt for channel 3	277
INTMSPI5FE0	Frame count end interrupt for channel 0	278
INTMSPI5FE1	Frame count end interrupt for channel 1	278
INTMSPI5FE2	Frame count end interrupt for channel 2	278
INTMSPI5FE3	Frame count end interrupt for channel 3	278
INTMSPI5ERR0	Error interrupt for channel 0	279
INTMSPI5ERR1	Error interrupt for channel 1	279
INTMSPI5ERR2	Error interrupt for channel 2	279
INTMSPI5ERR3	Error interrupt for channel 3	279
MSPI6		
INTMSPI6TX0	Transmit status interrupt for channel 0	280
INTMSPI6TX1	Transmit status interrupt for channel 1	280
INTMSPI6TX2	Transmit status interrupt for channel 2	280
INTMSPI6TX3	Transmit status interrupt for channel 3	280
INTMSPI6RX0	Receive status interrupt for channel 0	281
INTMSPI6RX1	Receive status interrupt for channel 1	281
INTMSPI6RX2	Receive status interrupt for channel 2	281
INTMSPI6RX3	Receive status interrupt for channel 3	281
INTMSPI6FE0	Frame count end interrupt for channel 0	282
INTMSPI6FE1	Frame count end interrupt for channel 1	282
INTMSPI6FE2	Frame count end interrupt for channel 2	282
INTMSPI6FE3	Frame count end interrupt for channel 3	282
INTMSPI6ERR0	Error interrupt for channel 0	283
INTMSPI6ERR1	Error interrupt for channel 1	283
INTMSPI6ERR2	Error interrupt for channel 2	283
INTMSPI6ERR3	Error interrupt for channel 3	283
MSPI7		
INTMSPI7TX0	Transmit status interrupt for channel 0	284
INTMSPI7TX1	Transmit status interrupt for channel 1	284
INTMSPI7TX2	Transmit status interrupt for channel 2	284
INTMSPI7TX3	Transmit status interrupt for channel 3	284
INTMSPI7RX0	Receive status interrupt for channel 0	285
INTMSPI7RX1	Receive status interrupt for channel 1	285
INTMSPI7RX2	Receive status interrupt for channel 2	285
INTMSPI7RX3	Receive status interrupt for channel 3	285
INTMSPI7FE0	Frame count end interrupt for channel 0	286

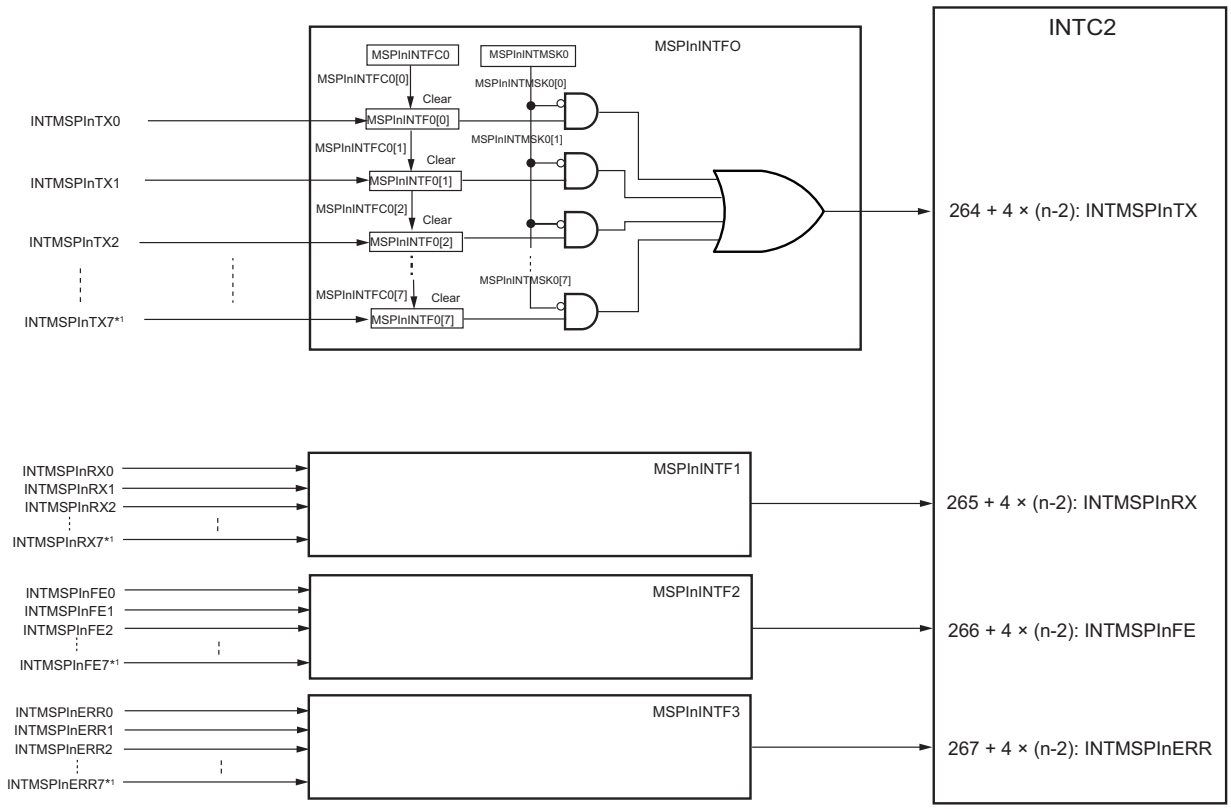
Table 19.6 Interrupt Requests (6/6)

Unit Interrupt Name	Description	Interrupt Number
INTMSPI7FE1	Frame count end interrupt for channel 1	286
INTMSPI7FE2	Frame count end interrupt for channel 2	286
INTMSPI7FE3	Frame count end interrupt for channel 3	286
INTMSPI7ERR0	Error interrupt for channel 0	287
INTMSPI7ERR1	Error interrupt for channel 1	287
INTMSPI7ERR2	Error interrupt for channel 2	287
INTMSPI7ERR3	Error interrupt for channel 3	287
MSPI8		
INTMSPI8TX0	Transmit status interrupt for channel 0	288
INTMSPI8TX1	Transmit status interrupt for channel 1	288
INTMSPI8TX2	Transmit status interrupt for channel 2	288
INTMSPI8TX3	Transmit status interrupt for channel 3	288
INTMSPI8RX0	Receive status interrupt for channel 0	289
INTMSPI8RX1	Receive status interrupt for channel 1	289
INTMSPI8RX2	Receive status interrupt for channel 2	289
INTMSPI8RX3	Receive status interrupt for channel 3	289
INTMSPI8FE0	Frame count end interrupt for channel 0	290
INTMSPI8FE1	Frame count end interrupt for channel 1	290
INTMSPI8FE2	Frame count end interrupt for channel 2	290
INTMSPI8FE3	Frame count end interrupt for channel 3	290
INTMSPI8ERR0	Error interrupt for channel 0	291
INTMSPI8ERR1	Error interrupt for channel 1	291
INTMSPI8ERR2	Error interrupt for channel 2	291
INTMSPI8ERR3	Error interrupt for channel 3	291
MSPI9		
INTMSPI9TX0	Transmit status interrupt for channel 0	292
INTMSPI9TX1	Transmit status interrupt for channel 1	292
INTMSPI9TX2	Transmit status interrupt for channel 2	292
INTMSPI9TX3	Transmit status interrupt for channel 3	292
INTMSPI9RX0	Receive status interrupt for channel 0	293
INTMSPI9RX1	Receive status interrupt for channel 1	293
INTMSPI9RX2	Receive status interrupt for channel 2	293
INTMSPI9RX3	Receive status interrupt for channel 3	293
INTMSPI9FE0	Frame count end interrupt for channel 0	294
INTMSPI9FE1	Frame count end interrupt for channel 1	294
INTMSPI9FE2	Frame count end interrupt for channel 2	294
INTMSPI9FE3	Frame count end interrupt for channel 3	294
INTMSPI9ERR0	Error interrupt for channel 0	295
INTMSPI9ERR1	Error interrupt for channel 1	295
INTMSPI9ERR2	Error interrupt for channel 2	295
INTMSPI9ERR3	Error interrupt for channel 3	295



Note: The number of interrupt signals is same as the number of channels of MSPIn. Refer **Section 19.1.1, Number of Units and Channels** for more details.

Figure 19.1 MSPI0-1 Interrupt Connection Image



Note: The number of interrupt signals is same as the number of channels of MSPIn. Refer **Section 19.1.1, Number of Units and Channels** for more details.

Figure 19.2 MSPI2-9 Interrupt Connection Image

This module has no error notification.

19.1.5 DMA/DTS Trigger

DMA/DTS trigger generator unit multiplexes INTMSPInTXm/INTMSPInRXm/INTMSPInFEm interrupt signals from all MSPI units and generates 30 DMA trigger signals (DMAMSPI0-29) and 64 DTS trigger signals (DTSMSPI0-63)The DMA/DTS trigger signals are listed in the following tables.

Table 19.7 sDMA Trigger number

DMA Trigger Signal	sDMA Trigger Number
DMAMSPI0	Group 0-109
DMAMSPI1	Group 0-110
DMAMSPI2	Group 0-111
DMAMSPI3	Group 0-112
DMAMSPI4	Group 0-113
DMAMSPI5	Group 0-114
DMAMSPI6	Group 0-115
DMAMSPI7	Group 0-116
DMAMSPI8	Group 0-117
DMAMSPI9	Group 0-118
DMAMSPI10	Group 0-119
DMAMSPI11	Group 0-120
DMAMSPI12	Group 0-121
DMAMSPI13	Group 0-122
DMAMSPI14	Group 0-123
DMAMSPI15	Group 0-124
DMAMSPI16	Group 0-125
DMAMSPI17	Group 0-126
DMAMSPI18	Group 0-127
DMAMSPI19	Group 0-128
DMAMSPI20	Group 0-129
DMAMSPI21	Group 0-130
DMAMSPI22	Group 0-131
DMAMSPI23	Group 0-132
DMAMSPI24	Group 0-133
DMAMSPI25	Group 0-134
DMAMSPI26	Group 0-135
DMAMSPI27	Group 0-136
DMAMSPI28	Group 0-137
DMAMSPI29	Group 0-138

Table 19.8 DTS trigger number (1/2)

DTS Trigger	DTS Trigger Number
DTSMSPI0	group2-0
DTSMSPI1	group2-1
DTSMSPI2	group2-2
DTSMSPI3	group2-3
DTSMSPI4	group2-4
DTSMSPI5	group2-5
DTSMSPI6	group2-6
DTSMSPI7	group2-7
DTSMSPI8	group2-8
DTSMSPI9	group2-9
DTSMSPI10	group2-10
DTSMSPI11	group2-11
DTSMSPI12	group2-12
DTSMSPI13	group2-13
DTSMSPI14	group2-14
DTSMSPI15	group2-15
DTSMSPI16	group2-16
DTSMSPI17	group2-17
DTSMSPI18	group2-18
DTSMSPI19	group2-19
DTSMSPI20	group2-20
DTSMSPI21	group2-21
DTSMSPI22	group2-22
DTSMSPI23	group2-23
DTSMSPI24	group2-24
DTSMSPI25	group2-25
DTSMSPI26	group2-26
DTSMSPI27	group2-27
DTSMSPI28	group2-28
DTSMSPI29	group2-29
DTSMSPI30	group2-30
DTSMSPI31	group2-31
DTSMSPI32	group2-32
DTSMSPI33	group2-33
DTSMSPI34	group2-34
DTSMSPI35	group2-35
DTSMSPI36	group2-36
DTSMSPI37	group2-37
DTSMSPI38	group2-38
DTSMSPI39	group2-39
DTSMSPI40	Group 0-98
DTSMSPI41	Group 0-99
DTSMSPI42	Group 0-100
DTSMSPI43	Group 0-101

Table 19.8 DTS trigger number (2/2)

DTS Trigger	DTS Trigger Number
DTSMSPI44	Group 0-102
DTSMSPI45	Group 0-103
DTSMSPI46	Group 0-104
DTSMSPI47	Group 0-105
DTSMSPI48	Group 0-106
DTSMSPI49	Group 0-107
DTSMSPI50	Group 0-108
DTSMSPI51	Group 0-109
DTSMSPI52	Group 0-110
DTSMSPI53	Group 0-111
DTSMSPI54	Group 0-112
DTSMSPI55	Group 0-113
DTSMSPI56	Group 0-114
DTSMSPI57	Group 0-115
DTSMSPI58	Group 0-116
DTSMSPI59	Group 0-117
DTSMSPI60	Group 0-118
DTSMSPI61	Group 0-119
DTSMSPI62	Group 0-120
DTSMSPI63	Group 0-121

19.1.6 Reset Sources

MSPI reset sources are listed in the following table. MSPI is initialized by these reset sources.

Table 19.9 Reset Sources

MSPIn	Reset Category						
Register Name	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
MSPIn registers*1	√	√	√	√*2	√	√*2	—
DMA/DTS Trigger Generator Registers	√	√	√	√	√	—	—

Note 1. All registers except DMA/DTS Trigger Registers.

Note 2. The execution of RAM initialization is configurable by a register setting.

19.1.7 External Input/Output Signals

External input/output signals of MSPIn are listed below.

Table 19.10 External Input/Output Signals (1/2)

Unit Signal Name	Description	Alternative port pin signal
MSPI0		
MSPI0SCK	Serial clock signal in Single end mode	MSPI0SC
	Serial clock signal in LVDS mode	MSPI0_SCKN MSPI0_SCKP
MSPI0SOUT	Output serial data signal in Single end mode	MSPI0SO
	Output serial data signal in LVDS mode	MSPI0_SON MSPI0_SOP
MSPI0SIN	Input serial data signal in Single end mode	MSPI0SI
	Input serial data signal in LVDS mode	MSPI0_SIN MSPI0_SIP
MSPI0CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI0SSI}}$
MSPI0DCSI	Data Consistency Check signal	MSPI0DCS
MSPI0CS[7:0]	Serial Peripheral Chip Select Output Signal	MSPI0CSS[7:0]
MSPI1		
MSPI1SCK	Serial clock signal in Single end mode	MSPI1SC
	Serial clock signal in LVDS mode	MSPI1_SCKN MSPI1_SCKP
MSPI1SOUT	Output serial data signal in Single end mode	MSPI1SO
	Output serial data signal in LVDS mode	MSPI1_SON MSPI1_SOP
MSPI1SIN	Input serial data signal in Single end mode	MSPI1SI
	Input serial data signal in LVDS mode	MSPI1_SIN MSPI1_SIP
MSPI1CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI1SSI}}$
MSPI1DCSI	Data Consistency Check signal	MSPI1DCS
MSPI1CS[7:0]	Serial Peripheral Chip Select Output Signal	MSPI1CSS[7:0]
MSPI2		
MSPI2SCK	Serial clock signal	MSPI2SC
MSPI2SOUT	Output serial data signal	MSPI2SO
MSPI2SIN	Input serial data signal	MSPI2SI
MSPI2CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI2SSI}}$
MSPI2SDCSI	Data Consistency Check signal	MSPI2DCS
MSPI2CS[7:0]	Serial Peripheral Chip Select Output Signal	MSPI2CSS[7:0]
MSPI3		
MSPI3SCK	Serial clock signal	MSPI3SC
MSPI3SOUT	Output serial data signal	MSPI3SO
MSPI3SIN	Input serial data signal	MSPI3SI
MSPI3CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI3SSI}}$
MSPI3DCSI	Data Consistency Check signal	MSPI3DCS
MSPI3CS[7:0]	Serial Peripheral Chip Select Output Signal	MSPI3CSS[7:0]
MSPI4		
MSPI4SCK	Serial clock signal	MSPI4SC

Table 19.10 External Input/Output Signals (2/2)

Unit Signal Name	Description	Alternative port pin signal
MSPI4SOUT	Output serial data signal	MSPI4SO
MSPI4SIN	Input serial data signal	MSPI4SI
MSPI4CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI4SSI}}$
MSPI4DCSI	Data Consistency Check signal	MSPI4DCS
MSPI4CS[7:0]	Serial Peripheral Chip Select Output signal	MSPI4CSS[7:0]
MSPI5		
MSPI5SCK	Serial clock signal	MSPI5SC
MSPI5SOUT	Output serial data signal	MSPI5SO
MSPI5SIN	Input serial data signal	MSPI5SI
MSPI5CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI5SSI}}$
MSPI5DCSI	Data Consistency Check signal	MSPI5DCS
MSPI5CS[3:0]	Serial Peripheral Chip Select Output signal	MSPI5CSS[3:0]
MSPI6		
MSPI6SCK	Serial clock signal	MSPI6SC
MSPI6SOUT	Output serial data signal	MSPI6SO
MSPI6SIN	Input serial data signal	MSPI6SI
MSPI6CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI6SSI}}$
MSPI6DCSI	Data Consistency Check signal	MSPI6DCS
MSPI6CS[3:0]	Serial Peripheral Chip Select Output signal	MSPI6CSS[3:0]
MSPI7		
MSPI7SCK	Serial clock signal	MSPI7SC
MSPI7SOUT	Output serial data signal	MSPI7SO
MSPI7SIN	Input serial data signal	MSPI7SI
MSPI7CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI7SSI}}$
MSPI7DCSI	Data Consistency Check signal	MSPI7DCS
MSPI7CS[3:0]	Serial Peripheral Chip Select Output signal	MSPI7CSS[3:0]
MSPI8		
MSPI8SCK	Serial clock signal	MSPI8SC
MSPI8SOUT	Output serial data signal	MSPI8SO
MSPI8SIN	Input serial data signal	MSPI8SI
MSPI8CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI8SSI}}$
MSPI8DCSI	Data Consistency Check signal	MSPI8DCS
MSPI8CS[3:0]	Serial Peripheral Chip Select Output signal	MSPI8CSS[3:0]
MSPI9		
MSPI9SCK	Serial clock signal	MSPI9SC
MSPI9SOUT	Output serial data signal	MSPI9SO
MSPI9SIN	Input serial data signal	MSPI9SI
MSPI9CSI	Serial Peripheral Chip Select Input Signal	$\overline{\text{MSPI9SSI}}$
MSPI9DCSI	Data Consistency Check signal	MSPI9DCS
MSPI9CS[3:0]	Serial Peripheral Chip Select Output signal	MSPI9CSS[3:0]

19.1.8 Data Consistency Check

The following table lists the port pins on which MSPInSOUT (MSPInSO) pin functions are multiplexed and data consistency checking.

Refer to **Section 19.6.8.2, Data Consistency Check** details on data consistency checking.

Table 19.11 Port Pins for Data Consistency Checking

Unit Signal Name	Port Pin Name	Alternative Function
MSPI0		
MSPI0SO	P2_13	AF3
	P4_5	AF2
	P6_9	AF3
MSPI1		
MSPI1SO	P5_2	AF3
	P6_7	AF4
	P10_2	AF3
MSPI2		
MSPI2SO	P2_5	AF4
	P4_6	AF4
	P10_10	AF3
MSPI3		
MSPI3SO	P5_4	AF2
	P22_0	AF4
	P24_13	AF2
MSPI4		
MSPI4SO	P21_2	AF4
	P23_2	AF1
MSPI5		
MSPI5SO	P12_2	AF1
	P24_6	AF2
MSPI6		
MSPI6SO	P9_2	AF1
	P11_2	AF1
MSPI7		
MSPI7SO	P11_10	AF1
	P19_2	AF1
MSPI8		
MSPI8SO	P0_2	AF5
	P18_3	AF1
MSPI9		
MSPI9SO	P1_2	AF5
	P18_10	AF1

CAUTION

This function is not supported with LVDS mode.

19.1.9 Combination of Pin and Port

MSPI uses multiple pins for one unit and the pins should be used in groups as given in the following table.

Table 19.12 Combination of Pin and Port table (1/2)

	Group	MSPIn SCK	MSPIn SIN	MSPIn SO	MSPIn DCSI	MSPIn SSI	MSPIn CSS0	MSPIn CSS1	MSPIn CSS2	MSPIn CSS3	MSPIn CSS4	MSPIn CSS5	MSPIn CSS6	MSPIn CSS7
MSPI0	1	P4_6	P4_7	P4_5	P4_5	P4_4	P2_3 P2_14 P4_4	P3_4 P4_1 P20_6	P4_13 P20_7	P2_8 P6_3 P10_10	P2_9 P4_15 P6_2	P3_4 P10_13 P20_0	P3_5 P10_14 P20_1	P2_14 P6_11 P17_1
	2	P6_8	P6_7	P6_9	P6_9	P4_4	P2_3 P2_14 P4_4	P3_4 P4_1 P20_6	P4_13 P20_7	P2_8 P6_3 P10_10	P2_9 P4_15 P6_2	P3_4 P10_3 P20_0	P3_5 P10_14 P20_1	P2_14 P6_11 P17_1
	3	P2_12	P2_11	P2_13	P2_13	P4_4	P2_3 P2_14 P4_4	P3_4 P4_1 P20_6	P4_13 P20_7	P2_8 P6_3 P10_10	P2_9 P4_15 P6_2	P3_4 P10_3 P20_0	P3_5 P10_14 P20_1	P2_14 P6_11 P17_1
	4	P2_14, P2_15 (LVDS)	P2_10, P2_11 (LVDS)	P2_12, P2_13 (LVDS)	-	P4_4	P2_3 P2_14 P4_4	P3_4 P4_1 P20_6	P4_13 P20_7	P2_8 P6_3 P10_10	P2_9 P4_15 P6_2	P3_4 P10_3 P20_0	P3_5 P10_14 P20_1	P2_14 P6_11 P17_1
MSPI1	1	P2_2	P2_1	P5_2	P5_2	P10_1	P2_0 P6_6 P10_1	P2_6 P6_5 P20_8	P2_4 P6_4 P20_12	P4_9 P20_4 P20_13	P10_11 P20_9 P17_2	P3_6 P10_12 P20_10	P3_5 P17_0 P17_3	P3_2 P5_6 P6_11
	2	P6_8	P6_9	P6_7	P6_7	P10_1	P2_0 P6_6 P10_1	P2_6 P6_5 P20_8	P2_4 P6_4 P20_12	P4_9 P20_4 P20_13	P10_11 P20_9 P17_2	P3_6 P10_12 P20_10	P3_5 P17_0 P17_3	P3_2 P5_6 P6_11
	3	P10_3	P10_4	P10_2	P10_2	P10_1	P2_0 P6_6 P10_1	P2_6 P6_5 P20_8	P2_4 P6_4 P20_12	P4_9 P20_4 P20_13	P10_11 P20_9 P17_2	P3_6 P10_12 P20_10	P3_5 P17_0 P17_3	P3_2 P5_6 P6_11
	4	P4_9, P4_10 (LVDS)	P4_5, P4_6 (LVDS)	P4_7, P4_8 (LVDS)	-	P10_1	P2_0 P6_6 P10_1	P2_6 P6_5 P20_8	P2_4 P6_4 P20_12	P4_9 P20_4 P20_13	P10_11 P20_9 P17_2	P3_6 P10_12 P20_10	P3_5 P17_0 P17_3	P3_2 P5_6 P6_11
MSPI2	1	P2_0	P2_3	P2_5	P2_5	P10_9	P4_14 P10_9	P2_2 P2_4	P2_5 P4_0 P6_2	P2_6 P4_1 P4_12	P2_11 P6_3	P2_12 P6_4	P2_13 P6_5	P2_14 P6_6 P10_13
	2	P4_7	P4_10	P4_6	P4_6	P10_9	P4_14 P10_9	P2_2 P2_4	P2_5 P4_0 P6_2	P2_6 P4_1 P4_12	P2_11 P6_3	P2_12 P6_4	P2_13 P6_5	P2_14 P6_6 P10_13
	3	P10_11	P10_12	P10_10	P10_10	P10_9	P4_14 P10_9	P2_2 P2_4	P2_5 P4_0 P6_2	P2_6 P4_1 P4_12	P2_11 P6_3	P2_12 P6_4	P2_13 P6_5	P2_14 P6_6 P10_13
MSPI3	1	P22_4	P22_1	P22_0	P22_0	P22_2	P22_2 P24_11 P17_2	P2_1 P20_2 P17_6	P5_6 P20_3 P17_5	P2_10 P17_3	P21_7	P21_6	P21_5	P21_4
	2	P24_10	P24_12	P24_13	P24_13	P22_2	P22_2 P24_11 P17_2	P2_1 P20_2 P17_6	P5_6 P20_3 P17_5	P2_10 P17_3	P21_7	P21_6	P21_5	P21_4
	3	P5_2	P5_3	P5_4	P5_4	P22_2	P22_2 P24_11 P17_2	P2_1 P20_2 P17_6	P5_6 P20_3 P17_5	P2_10 P17_3	P21_7	P21_6	P21_5	P21_4
MSPI4	1	P23_0	P23_1	P23_2	P23_2	P21_3 P23_3	P21_3 P23_3	P10_5 P23_4	P10_6 P23_5	P10_7 P23_6	P10_8 P20_4 P23_7	P20_14 P23_8	P10_0 P23_9	P23_10 P17_4
	2	P21_0	P21_1	P21_2	P21_2	P21_3 P23_3	P21_3 P23_3	P10_5 P23_4	P10_6 P23_5	P10_7 P23_6	P10_8 P20_4 P23_7	P20_14 P23_8	P10_0 P23_9	P23_10 P17_4
MSPI5	1	P12_0	P12_1	P12_2	P12_2	P12_3, P24_7	P12_3 P24_7	P12_4 P24_8	P12_5 P24_9	P4_8 P18_0	-	-	-	-
	2	P24_4	P24_5	P24_6	P24_6	P12_3, P24_7	P12_3 P24_7	P12_4 P24_8	P12_5 P24_9	P4_8 P18_0	-	-	-	-

Table 19.12 Combination of Pin and Port table (2/2)

	Group	MSPIn SCK	MSPIn SIN	MSPIn SO	MSPIn DCSI	MSPIn SSI	MSPIn CSS0	MSPIn CSS1	MSPIn CSS2	MSPIn CSS3	MSPIn CSS4	MSPIn CSS5	MSPIn CSS6	MSPIn CSS7
MSPI6	1	P9_0	P9_1	P9_2	P9_2	P9_3 P11_3	P9_3 P11_3	P9_4 P11_4	P9_5 P11_5	P9_6 P11_6	-	-	-	-
	2	P11_0	P11_1	P11_2	P11_2	P9_3 P11_3	P9_3 P11_3	P9_4 P11_4	P9_5 P11_5	P9_6 P11_6	-	-	-	-
MSPI7	1	P11_8	P11_9	P11_10	P11_10	P11_11 P19_3	P11_11 P19_3	P11_12 P19_4	P11_13 P19_5	P11_14 P18_15	-	-	-	-
	2	P19_0	P19_1	P19_2	P19_2	P11_11 P19_3	P11_11 P19_3	P11_12 P19_4	P11_13 P19_5	P11_14 P18_15	-	-	-	-
MSPI8	1	P0_0	P0_1	P0_2	P0_2	P0_3 P18_4	P0_3 P18_4	P0_4 P18_5	P0_5 P18_6	P0_6 P18_7	-	-	-	-
	2	P18_1	P18_2	P18_3	P18_3	P0_3 P18_4	P0_3 P18_4	P0_4 P18_5	P0_5 P18_6	P0_6 P18_7	-	-	-	-
MSPI9	1	P1_0	P1_1	P1_2	P1_2	P1_3 P18_11	P1_3 P18_11	P1_4 P18_12	P1_5 P18_13	P1_6 P18_14	-	-	-	-
	2	P18_8	P18_9	P18_10	P18_10	P1_3 P18_11	P1_3 P18_11	P1_4 P18_12	P1_5 P18_13	P1_6 P18_14	-	-	-	-

Note: Refer to **Table 2.3, Pin Function Name Definition** for available Port for each Product.
Refer to **Figure 2.9, Port setting flow example (Alternative mode with IP control)** for Port setting for MSPInSO and MSPInDCS.
Refer to **Figure 2.10, Port setting flow example (Alternative mode without IP control)** for Port setting for other pins.

19.2 Overview

19.2.1 Functional Overview

Multichannel Serial Peripheral Interface (MSPI) has the function as the following:

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Multiple slaves configuration thanks to up to eight configurable chip select output signals
- Maximum transmission speed in master/slave mode:
  - In LVDS mode: up to 40 MHz (MSPI0-1)
  - In Single end mode: up to 20 MHz (MSPI0-5), up to 10 MHz (MSPI6-9)

CAUTIONS

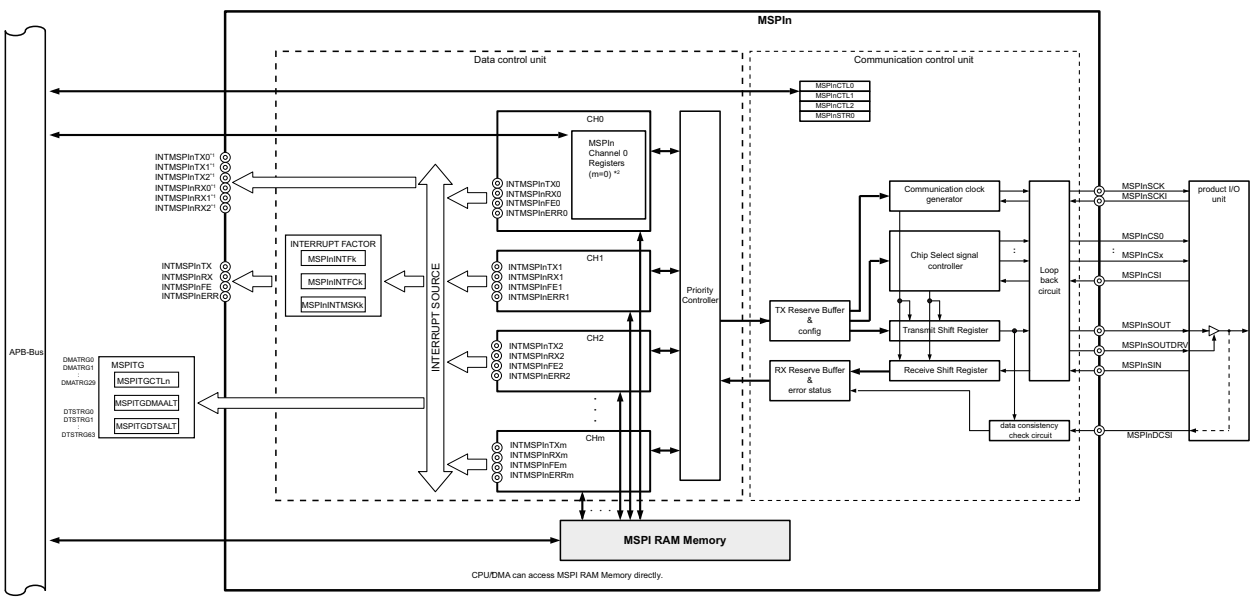
1. **Actual timings will depend on the characteristics of master & slave.**
2. **MSPI0 does not support LVDS mode in U2A6.**

- Phase of clock and data selectable for each channel
- Data transfer with MSB or LSB first selectable for each channel
- Transfer control channel: Up to 8 for each unit
- Transfer data length selectable from 2 to 128 incremented in 1-bit units for channel
- Three selectable transfer modes:
  - Transmit-only mode
  - Receive-only mode
  - Transmit/receive mode
- Error detection
  - Parity error
  - Data consistency error
  - Over-write error
  - Over-read error
  - Overrun error
- Support of JOB concept for AUTOSAR
- JOB enable control bit for AUTOSAR is provided
- LBM (Loop Back Mode) function for self-test
- Enforced chip select idle setting
- Support Safe-SPI ver1.00 only in master mode
- MSPI has four different interrupt requests.

  - Communication status
  - Reception status
  - Communication error
  - Job completion
- MSPI has three DMA requests.
  - Communication status
  - Reception status
  - Job completion
- Receive sample point
  - The sample point for the RX can be shifted to the next SCK edge

19.2.2 Block Diagram

The following block diagram shows the main components of the MSPI.



Note 1. These interrupts are only for MSPI0 and MSPI1. For more details, refer to **Figure 19.1, MSPI0-1 Interrupt Connection Image**.

Note 2. Refer to **Table 19.17, List of Channel m Registers**.

Figure 19.3 Block Diagram of MSPI

19.3 Interrupt Sources

MSPIn can generate the following interrupt requests:

- INTMSPInTXm (transmit status interrupt)
- INTMSPInRXm (receive status interrupt)
- INTMSPInFEm (frame count end interrupt)
- INTMSPInERRm (error interrupt)

m = 0 to 7

Table 19.13 Interrupt Generation

Mode	Interrupt	Cause of Interrupt
Direct memory mode	INTMSPInTXm	TX data empty *1
	INTMSPInRXm	RX data received *2
	INTMSPInFEm	Last frame end
	INTMSPInERRm	Error detected
Fixed buffer memory mode	INTMSPInTXm	Not output
	INTMSPInRXm	Not output
	INTMSPInFEm	Last frame end
	INTMSPInERRm	Error detected
Fixed FIFO memory mode	INTMSPInTXm	Half of FIFO buffer empty. *1
	INTMSPInRXm	Half of FIFO buffer full. *2
	INTMSPInFEm	Last frame end
	INTMSPInERRm	Error detected

Note 1. Only if data transmission is enabled (MSPInCFGm0.MSPInTXE=1).
Note 2. Only if data reception is enabled (MSPInCFGm0.MSPInRXEm=1.

19.4 DMA/DTS Trigger Generation

DMA/DTS trigger generator unit multiplexes INTMSPInTXm/INTMSPInRXm/INTMSPInFEm interrupt signals from all MSPI units and generates 30 DMA trigger signals (DMAMSPI0-29) and 64 DTS trigger signals (DTSMSPI0-63).

Two DMA trigger signals (Trigger 1/Trigger 2) and two DTS trigger signals (Trigger 1/Trigger 2) are assigned to each unit/channel, and the source interrupt signals for each trigger signal are selected by register settings. For more detailed information, refer to **Section 19.5.3, DMA/DTS Trigger Generator Registers**.

Each DMA/DTS trigger signal is shared by multiplex channels. In addition, a part of DMA/DTS trigger signals is shared by two MSPI units (Ex: MSPI0 and MSPI9 shared DMAMSPI4 Trigger 1 and DMAMSPI5 Trigger 2).

The assignments and the sharing are shown in the following table.

Table 19.14 DMA/DTS Trigger signals (1/2)

Unit	Channel	DMA Trigger Signal				DTS Trigger Signal			
		Trigger 1	Trigger 2	Trigger 1 (alternative)	Trigger 2 (alternative)	Trigger 1	Trigger 2	Trigger 1 (alternative)	Trigger 2 (alternative)
MSPI0	0	DMAMSPI0	DMAMSPI1	-	-	DTSMSPI0	DTSMSPI1	-	-
	1	DMAMSPI2	DMAMSPI3	-	-	DTSMSPI2	DTSMSPI3	-	-
	2	DMAMSPI4*1	DMAMSPI5*1	DMAMSPI0	DMAMSPI1	DTSMSPI4	DTSMSPI5	-	-
	3	DMAMSPI6*2	DMAMSPI7*2	DMAMSPI2	DMAMSPI3	DTSMSPI6	DTSMSPI7	-	-
	4	DMAMSPI0	DMAMSPI1	-	-	DTSMSPI8	DTSMSPI9	-	-
	5	DMAMSPI2	DMAMSPI3	-	-	DTSMSPI10	DTSMSPI11	-	-
	6	DMAMSPI4*1	DMAMSPI5*1	DMAMSPI0	DMAMSPI1	DTSMSPI12*7	DTSMSPI13*7	DTSMSPI8	DTSMSPI9
	7	DMAMSPI6*2	DMAMSPI7*2	DMAMSPI2	DMAMSPI3	DTSMSPI14*7	DTSMSPI15*7	DTSMSPI10	DTSMSPI11
MSPI1	0	DMAMSPI8	DMAMSPI9	-	-	DTSMSPI16	DTSMSPI17	-	-
	1	DMAMSPI10	DMAMSPI11	-	-	DTSMSPI18	DTSMSPI19	-	-
	2	DMAMSPI12*3	DMAMSPI13*3	DMAMSPI8	DMAMSPI9	DTSMSPI20	DTSMSPI21	-	-
	3	DMAMSPI14*4	DMAMSPI15*4	DMAMSPI10	DMAMSPI11	DTSMSPI22	DTSMSPI23	-	-
	4	DMAMSPI8	DMAMSPI9	-	-	DTSMSPI24*8	DTSMSPI25*8	DTSMSPI16	DTSMSPI17
	5	DMAMSPI10	DMAMSPI11	-	-	DTSMSPI26*8	DTSMSPI27*8	DTSMSPI18	DTSMSPI19
	6	DMAMSPI12*3	DMAMSPI13*3	DMAMSPI8	DMAMSPI9	DTSMSPI28*9	DTSMSPI29*9	DTSMSPI20	DTSMSPI21
	7	DMAMSPI14*4	DMAMSPI15*4	DMAMSPI10	DMAMSPI11	DTSMSPI30*9	DTSMSPI31*9	DTSMSPI22	DTSMSPI23
MSPI2	0	DMAMSPI16	DMAMSPI17	-	-	DTSMSPI32	DTSMSPI33	-	-
	1	DMAMSPI18	DMAMSPI19	-	-	DTSMSPI34	DTSMSPI35	-	-
	2	DMAMSPI16	DMAMSPI17	-	-	DTSMSPI36	DTSMSPI37	-	-
	3	DMAMSPI18	DMAMSPI19	-	-	DTSMSPI38	DTSMSPI39	-	-
	4	DMAMSPI16	DMAMSPI17	-	-	DTSMSPI40*10	DTSMSPI41*10	DTSMSPI32	DTSMSPI33
	5	DMAMSPI18	DMAMSPI19	-	-	DTSMSPI42*10	DTSMSPI43*10	DTSMSPI34	DTSMSPI35
	6	DMAMSPI16	DMAMSPI17	-	-	DTSMSPI44*11	DTSMSPI45*11	DTSMSPI36	DTSMSPI37
	7	DMAMSPI18	DMAMSPI19	-	-	DTSMSPI46*11	DTSMSPI47*11	DTSMSPI38	DTSMSPI39

Table 19.14 DMA/DTS Trigger signals (2/2)

Unit	Channel	DMA Trigger Signal				DTS Trigger Signal			
		Trigger 1	Trigger 2	Trigger 1 (alternative)	Trigger 2 (alternative)	Trigger 1	Trigger 2	Trigger 1 (alternative)	Trigger 2 (alternative)
MSPI3	0	DMAMSPI20	DMAMSPI21	-	-	DTSMSPI48	DTSMSPI49	-	-
	1	DMAMSPI22	DMAMSPI23	-	-	DTSMSPI50	DTSMSPI51	-	-
	2	DMAMSPI20	DMAMSPI21	-	-	DTSMSPI52	DTSMSPI53	-	-
	3	DMAMSPI22	DMAMSPI23	-	-	DTSMSPI54	DTSMSPI55	-	-
	4	DMAMSPI20	DMAMSPI21	-	-	DTSMSPI56*12	DTSMSPI57*12	DTSMSPI48	DTSMSPI49
	5	DMAMSPI22	DMAMSPI23	-	-	DTSMSPI58*12	DTSMSPI59*12	DTSMSPI50	DTSMSPI51
	6	DMAMSPI20	DMAMSPI21	-	-	DTSMSPI60*13	DTSMSPI61*13	DTSMSPI52	DTSMSPI53
	7	DMAMSPI22	DMAMSPI23	-	-	DTSMSPI62*13	DTSMSPI63*13	DTSMSPI54	DTSMSPI55
MSPI4	0	DMAMSPI24	DMAMSPI25	-	-	DTSMSPI60*14	DTSMSPI61*14	-	-
	1	DMAMSPI26	DMAMSPI27	-	-	DTSMSPI62*14	DTSMSPI63*14	-	-
	2	DMAMSPI24	DMAMSPI25	-	-	DTSMSPI60*14	DTSMSPI61*14	-	-
	3	DMAMSPI26	DMAMSPI27	-	-	DTSMSPI62*14	DTSMSPI63*14	-	-
	4	DMAMSPI24	DMAMSPI25	-	-	DTSMSPI44*15	DTSMSPI45*15	-	-
	5	DMAMSPI26	DMAMSPI27	-	-	DTSMSPI46*15	DTSMSPI47*15	-	-
	6	DMAMSPI24	DMAMSPI25	-	-	DTSMSPI44*15	DTSMSPI45*15	-	-
	7	DMAMSPI26	DMAMSPI27	-	-	DTSMSPI46*15	DTSMSPI47*15	-	-
MSPI5	0	DMAMSPI28	DMAMSPI29	-	-	DTSMSPI28*16	DTSMSPI29*16	-	-
	1	DMAMSPI28	DMAMSPI29	-	-	DTSMSPI30*16	DTSMSPI31*16	-	-
	2	DMAMSPI28	DMAMSPI29	-	-	DTSMSPI28*16	DTSMSPI29*16	-	-
	3	DMAMSPI28	DMAMSPI29	-	-	DTSMSPI30*16	DTSMSPI31*16	-	-
MSPI6	0	DMAMSPI14*5	DMAMSPI15*5	-	-	DTSMSPI12*17	DTSMSPI13*17	-	-
	1	DMAMSPI14*5	DMAMSPI15*5	-	-	DTSMSPI14*17	DTSMSPI15*17	-	-
	2	DMAMSPI14*5	DMAMSPI15*5	-	-	DTSMSPI12*17	DTSMSPI13*17	-	-
	3	DMAMSPI14*5	DMAMSPI15*5	-	-	DTSMSPI14*17	DTSMSPI15*17	-	-
MSPI7	0	DMAMSPI12*5	DMAMSPI13*5	-	-	DTSMSPI56*14	DTSMSPI57*14	-	-
	1	DMAMSPI12*5	DMAMSPI13*5	-	-	DTSMSPI58*14	DTSMSPI59*14	-	-
	2	DMAMSPI12*5	DMAMSPI13*5	-	-	DTSMSPI56*14	DTSMSPI57*14	-	-
	3	DMAMSPI12*5	DMAMSPI13*5	-	-	DTSMSPI58*14	DTSMSPI59*14	-	-
MSPI8	0	DMAMSPI6*6	DMAMSPI7*6	-	-	DTSMSPI40*15	DTSMSPI41*15	-	-
	1	DMAMSPI6*6	DMAMSPI7*6	-	-	DTSMSPI42*15	DTSMSPI43*15	-	-
	2	DMAMSPI6*6	DMAMSPI7*6	-	-	DTSMSPI40*15	DTSMSPI41*15	-	-
	3	DMAMSPI6*6	DMAMSPI7*6	-	-	DTSMSPI42*15	DTSMSPI43*15	-	-
MSPI9	0	DMAMSPI4*6	DMAMSPI5*6	-	-	DTSMSPI24*16	DTSMSPI25*16	-	-
	1	DMAMSPI4*6	DMAMSPI5*6	-	-	DTSMSPI26*16	DTSMSPI27*16	-	-
	2	DMAMSPI4*6	DMAMSPI5*6	-	-	DTSMSPI24*16	DTSMSPI25*16	-	-
	3	DMAMSPI4*6	DMAMSPI5*6	-	-	DTSMSPI26*16	DTSMSPI27*16	-	-

Note 1. Shared with MSPI9
Note 2. Shared with MSPI8
Note 3. Shared with MSPI7
Note 4. Shared with MSPI6
Note 5. Shared with MSPI1
Note 6. Shared with MSPI0
Note 7. Shared with MSPI6
Note 8. Shared with MSPI9
Note 9. Shared with MSPI5
Note 10. Shared with MSPI8
Note 11. Shared with MSPI4

Note 12. Shared with MSPI7
Note 13. Shared with MSPI4
Note 14. Shared with MSPI3
Note 15. Shared with MSPI2
Note 16. Shared with MSPI1
Note 17. Shared with MSPI0

Some examples of DMA trigger generation setting are shown in the following. (DTS trigger generation setting can also be done in a similar way.)

(1) [Example 1] MSPI0-Ch0: Use DMAMSPI0 for INTMSPI0TX0

As shown in **Table 19.14, DMA/DTS Trigger signals**, MSPI0-Ch0 has two DMA trigger signals, DMAMSPI0(Trigger 1) and DMAMSPI1(Trigger 2).

Set as follows to use DMAMSPI0(Trigger 1) for INTMSPI0TX0.

Setting	Explanation
MSPITGCTL0 = 0000 0002_H	• DMA trigger is enabled (DTS trigger is disabled) for MSPI0-Ch0 when MSPITGCTL0.TRGSEL0 = 0. • INTMSPI0TX0 interrupt signal is selected as source of Trigger 1 for MSPI0-Ch0 when MSPITGCTL0.SRCSEL0 = 001.
MSPITGDMAALT = Don't care	MSPI0-Ch0 can not use alternative trigger signals, and does not need MSPITGDMAALT register to be set. DMAMSPI0 is always used as DMA Trigger 1 for MSPI0-Ch0.

(2) [Example 2] MSPI0-Ch2: Use DMAMSPI1 or DMAMSPI5 for INTMSPI0RX2

As shown in **Table 19.14, DMA/DTS Trigger signals**, MSPI0-Ch2 has two DMA trigger signals, DMAMSPI4(Trigger 1) and DMAMSPI5(Trigger 2), or two alternative DMA trigger signals, DMAMSPI0(Trigger 1) and DMAMSPI1(Trigger 2).

Set as follows when using the alternative DMA trigger signal DMAMSPI1(Trigger 2) for INTMSPI0RX2.

Setting	Explanation
MSPITGCTL0 = 0000 $0A00_H$	• DMA trigger is enabled (DTS trigger is disabled) for MSPI0-Ch2 when MSPITGCTL0.TRGSEL2 = 0. • INTMSPI0RX2 interrupt signal is selected as source of Trigger 2 for MSPI0-Ch2 when MSPITGCTL0.SRCSEL2 = 101.
MSPITGDMAALT = 0000 0001_H	MSPI0-Ch2 can use alternative DMA trigger signals, and needs MSPITGDMAALT register to be set. DMAMSPI1 is used as DMA Trigger 2 for MSPI0-Ch2 when MSPITGDMAALT.DMAAS02 = 1.

On the other hand, set as follows when using the usual DMA trigger signal DMAMSPI5(Trigger 2) for INTMSPI0RX2.

Setting	Explanation
MSPITGCTL0 = 0000 $0A00_H$	• DMA trigger is enabled (DTS trigger is disabled) for MSPI0-Ch2 when MSPITGCTL0.TRGSEL2 = 0. • INTMSPI0RX2 interrupt signal is selected as source of Trigger 2 for MSPI0-Ch2 when MSPITGCTL0.SRCSEL2 = 101.
MSPITGDMAALT = 0000 0000_H	MSPI0-Ch2 can use alternative DMA trigger signals, and needs MSPITGDMAALT register to be set. DMAMSPI5 is used as DMA Trigger 2 for MSPI0-Ch2 when MSPITGDMAALT.DMAAS02 = 0.

(3) [Example 3] MSPI9-Ch0 use DMAMSPI4 for INTMSPI9TX0 and DMAMSPI5 for INTMSPI9RX0

As shown in **Table 19.14, DMA/DTS Trigger signals**, MSPI9-Ch0 has two DMA trigger signals, DMAMSPI4(Trigger 1) and DMAMSPI5(Trigger 2).

Set as follows to use DMAMSPI4(Trigger 1) for INTMSPI9TX0, and DMAMSPI5 for INTMSPI9RX0.

Setting	Explanation
MSPITGCTL9 = 0000 000E$_H$	• DMA trigger is enabled (DTS trigger is disabled) for MSPI9-Ch0 when MSPITGCTL9.TRGSEL0 = 0. • INTMSPI9TX0/INTMSPI9RX0 interrupt signal is selected as source of Trigger 1/2 for MSPI9-Ch0 when MSPITGCTL9.SRCSEL0 = 111.
MSPITGDMAALT = Don't care	MSPI9-Ch0 can not use alternative trigger signals, and does not need MSPITGDMAALT register to be set. DMAMSPI4/DMAMSPI5 is always used as DMA Trigger 1/2 for MSPI9-Ch0.

(4) [Example 4] Combination of settings for multiple units/channels

The assignments of DMA trigger signals and the source interrupt signals for each units/channels in this example are shown in the following table.

Table 19.15 Description of Example 4

Unit	Channel	Trigger 1		Trigger 2		Trigger 1 (alternative)		Trigger 2 (alternative)	
		DMA Trigger Signal	Source Signal	DMA Trigger Signal	Source Signal	DMA Trigger Signal	Source Signal	DMA Trigger Signal	Source Signal
MSPI0	0	DMAMSPI0	INTMSPI0TX0	DMAMSPI1	-	-	-	-	-
	1	DMAMSPI2	INTMSPI0TX1	DMAMSPI3	-	-	-	-	-
	2	DMAMSPI4	-	DMAMSPI5	-	DMAMSPI0	-	DMAMSPI1	INTMSPI0RX2
	3	DMAMSPI6	-	DMAMSPI7	-	DMAMSPI2	-	DMAMSPI3	INTMSPI0RX3
MSPI8	0	DMAMSPI6	INTMSPI8TX0	DMAMSPI7	-	-	-	-	-
	1	DMAMSPI6	-	DMAMSPI7	INTMSPI8TX1	-	-	-	-
MSPI9	0	DMAMSPI4	INTMSPI9TX0	DMAMSPI5	INTMSPI9RX0	-	-	-	-

The following assignments are necessary for each unit/channel.

- MSPI0-Ch0: Use DMAMSPI0 for INTMSPI0TX0
- MSPI0-Ch1: Use DMAMSPI2 for INTMSPI0TX1
- MSPI0-Ch2: Use DMAMSPI1 for INTMSPI0RX2
- MSPI0-Ch3: Use DMAMSPI3 for INTMSPI0RX3
- MSPI8-Ch0: Use DMAMSPI6 for INTMSPI8TX0
- MSPI8-Ch1: Use DMAMSPI7 for INTMSPI8TX1
- MSPI9-Ch0: Use DMAMSPI4/DMAMSPI5 for INTMSPI9TX0/INTMSPI9RX0

Set as follows for these assignments.

Setting	Explanation
MSPITGCTL0 = 0000 $AA22_H$	• DMA trigger is enabled (DTS trigger is disabled) for MSPI0-Ch0/1/2/3 when MSPITGCTL0.TRGSEL0/1/2/3 = 0. • INTMSPI0TX0 interrupt signal is selected as source of Trigger 1 for MSPI0-Ch0 when MSPITGCTL0.SRCSEL0 = 001. • INTMSPI0TX1 interrupt signal is selected as source of Trigger 1 for MSPI0-Ch1 when MSPITGCTL0.SRCSEL1 = 001. • INTMSPI0RX2 interrupt signal is selected as source of Trigger 2 for MSPI0-Ch2 when MSPITGCTL0.SRCSEL2 = 101. • INTMSPI0RX3 interrupt signal is selected as source of Trigger 2 for MSPI0-Ch3 when MSPITGCTL0.SRCSEL3 = 101.
MSPITGCTL8 = 0000 0082_H	• DMA trigger is enabled (DTS trigger is disabled) for MSPI8-Ch0/1 when MSPITGCTL8.TRGSEL0/1 = 0. • INTMSPI8TX0 interrupt signal is selected as source of Trigger 1 for MSPI8-Ch0 when MSPITGCTL8.SRCSEL0 = 001. • INTMSPI8TX1 interrupt signal is selected as source of Trigger 2 for MSPI8-Ch1 when MSPITGCTL8.SRCSEL1 = 100.
MSPITGCTL9 = 0000 $000E_H$	• DMA trigger is enabled (DTS trigger is disabled) for MSPI9-Ch0 when MSPITGCTL9.TRGSEL0 = 0. • INTMSPI9TX0/INTMSPI9RX0 interrupt signal is selected as source of Trigger 1/2 for MSPI9-Ch0 when MSPITGCTL9.SRCSEL0 = 111.
MSPITGDMAALT = 0000 0003_H	• MSPI0-Ch2 can use alternative DMA trigger signals, and needs MSPITGDMAALT register to be set. DMAMSPI1 is used as DMA Trigger 2 for MSPI0-Ch2 when MSPITGDMAALT.DMAAS02 = 1. • MSPI0-Ch3 can use alternative DMA trigger signals, and needs MSPITGDMAALT register to be set. DMAMSPI3 is used as DMA Trigger 2 for MSPI0-Ch3 when MSPITGDMAALT.DMAAS03 = 1.

19.5 Registers of MSPI

19.5.1 List of Registers

MSPI registers are listed in the following table.

Table 19.16 List of Common Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MSPIn	MSPIn Control Register 0	MSPInCTL0	<MSPIn_base> + 0000_H	8	*1	—
	MSPIn Control Register 1	MSPInCTL1	<MSPIn_base> + 0004_H	8,16,32	*1	—
	MSPIn Control Register 2	MSPInCTL2	<MSPIn_base> + 0008_H	8	*1	—
	MSPIn Status Register 0	MSPInSTR0	<MSPIn_base> + 0010_H	8	*1	—
MSPInINTF	MSPIn Interrupt Factor Register	MSPInINTFk	<MSPIn_INTF_base> + k x 30_H + 00_H	32	*1	—
	MSPIn Interrupt Mask Register	MSPInINTMSKk	<MSPIn_INTF_base> + k x 30_H + 04_H	32	*1	—
	MSPIn Interrupt Factor Clear Register	MSPInINTFCk	<MSPIn_INTF_base> + k x 30_H + 08_H	32	*1	—

Note 1. n = 0: PBG40#3, n = 1: PBG52#6, n = 2: PBG40#4, n = 3: PBG52#7, n = 4: PBG40#5, n = 5: PBG52#8, n = 6: PBG40#6, n = 7: PBG52#9, n = 8: PBG40#7, n = 9: PBG52#10

Table 19.17 List of Channel m Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MSPIn	MSPIn channel Configuration Register m0	MSPInCFGm0	<MSPIn_base> + 00_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn channel Configuration Register m1	MSPInCFGm1	<MSPIn_base> + 04_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn channel Configuration Register m2	MSPInCFGm2	<MSPIn_base> + 08_H + 80_H x (m+1)	8,16	*1	—
	MSPIn channel Configuration Register m3	MSPInCFGm3	<MSPIn_base> + $0C_H$ + 80_H x (m+1)	8,16	*1	—
	MSPIn channel Configuration Register m4	MSPInCFGm4	<MSPIn_base> + 10_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn SCK Delay Time Setting Register m	MSPInSEUPm	<MSPIn_base> + 18_H + 80_H x (m+1)	8,16	*1	—
	MSPIn CS Negation Delay Time Setting Register m	MSPInHOLDm	<MSPIn_base> + $1C_H$ + 80_H x (m+1)	8,16	*1	—
	MSPIn CS Next Frame Delay Time Setting Register m	MSPInIDLEm	<MSPIn_base> + 20_H + 80_H x (m+1)	8,16	*1	—
	MSPIn Inter Data Time Setting Register m	MSPInINDAm	<MSPIn_base> + 24_H + 80_H x (m+1)	8,16	*1	—
	MSPI RAM Start Address Setting Register m	MSPInRASTADm	<MSPIn_base> + 28_H + 80_H x (m+1)	8,16	*1	—
	MSPIn Communication Frame Count Setting Register m	MSPInCFSETm	<MSPIn_base> + $2C_H$ + 80_H x (m+1)	8,16	*1	—
	MSPIn Communication Transmit Data Register m	MSPInSSTXm	<MSPIn_base> + 30_H + 80_H x (m+1)	8,16,32	*1	—

Table 19.17 List of Channel m Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MSPIn	MSPIn Channel Status Set Register m	MSPInCSTSm	<MSPIn_base> + 34_H + 80_H x (m+1)	8	*1	—
	MSPIn Channel Status Clear Register m	MSPInCSTCm	<MSPIn_base> + 38_H + 80_H x (m+1)	8	*1	—
	MSPIn CS signal Selection Register m	MSPInSSELm	<MSPIn_base> + $3C_H$ + 80_H x (m+1)	8,16	*1	—
	MSPIn Communication Transmit Data Register m0	MSPInTXDAm0	<MSPIn_base> + 40_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn Frame Error Status Register m	MSPInFRERSTm	<MSPIn_base> + $4C_H$ + 80_H x (m+1)	8	*1	—
	MSPIn Communication Receive Data Register m0	MSPInRXDAm0	<MSPIn_base> + 50_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn Channel Status Register m	MSPInCSTRm	<MSPIn_base> + 58_H + 80_H x (m+1)	8,16	*1	—
	MSPIn Channel Frame Count Register m	MSPInCFCNTm	<MSPIn_base> + $5C_H$ + 80_H x (m+1)	8,16,32	*1	—
	MSPIn Frame Error Status and Communication Receive Data Register m	MSPInFSRXm	<MSPIn_base> + 60_H + 80_H x (m+1)	8,16,32	*1	—
	MSPIn Frame Error Status Clear Register m	MSPInFRERSTCm	<MSPIn_base> + 64_H + 80_H x (m+1)	8	*1	—
	MSPIn Channel Error Status Register m	MSPInCESTm	<MSPIn_base> + 68_H + 80_H x (m+1)	8	*1	—
	MSPIn Channel Status Clear Register m	MSPInCESTCm	<MSPIn_base> + $6C_H$ + 80_H x (m+1)	8	*1	—

Note 1. n = 0: PBG40#3, n = 1: PBG52#6, n = 2: PBG40#4, n = 3: PBG52#7, n = 4: PBG40#5, n = 5: PBG52#8, n = 6: PBG40#6, n = 7: PBG52#9, n = 8: PBG40#7, n = 9: PBG52#10

Table 19.18 List of DMA/DTS Trigger Generator Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MSPITG	MSPI Trigger Control Register for MSPIn	MSPITGCTLn	<MSPITG_base > + n x 04_H	8,16,32	PBG40#2	—
	MSPI DMA Trigger alternative Select Register	MSPITGDMAALT	<MSPITG_base > + 28_H	8,16,32	PBG40#2	—
	MSPI DTS Trigger alternative Select Register	MSPITGDTSALT	<MSPITG_base > + $2C_H$	8,16,32	PBG40#2	—

Table 19.19 Areas of MSPI RAM

Module Name	Address	Product
MSPIn	<MSPIn_base> + 1000_H to <MSPIn_base> + $107F_H$	U2A-EVA, U2A16, U2A8
	<MSPIn_base> + 1000_H to <MSPIn_base> + $11FF_H$	U2A6

19.5.2 MSPIn Registers

19.5.2.1 MSPInCTL0 — MSPIn Control Register 0

This register is used to control the enable/disable of the MSPI function.

Access: This register can be read or written in 8-bit units.

Address: <MSPIn_base> + 0000_H

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MSPInEN
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Table 19.20 MSPInCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MSPInEN	This bit enables or disables the MSPI Function. 0: Disables the MSPIn function. 1: Enables the MSPIn function.

CAUTION

Four or more PCLK clock cycles plus four or more MSPInCLK clock cycles are necessary if MSPInEN has been cleared and then if MSPInEN has set again.

19.5.2.2 MSPInCTL1 — MSPIn Control Register 1

This register is used to control the MSPI function.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 0004_H

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSPIn MSSEL	—	MSPIn CSIE	MSPIn SAMP	MSPIn CKR	—	MSPIn SOLS[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MSPInCSP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W	R/W	R/W	R/W

Note 1. Writable only for MSPI0 to 4

Table 19.21 MSPInCTL1 Register Contents

Bit Position	Bit Name	Function
31	MSPInMSSEL	This bit selects Master/Slave Mode 0: Master mode 1: Slave mode **Note:** Setting MSPInMSSEL before setting Port Alternative function for MSPI operation.
30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MSPInCSIE	This bit enables or disables the slave select signal(MSPInCSI) in slave mode. 0:Input CS signal(MSPInCSI) is ignored in slave mode. 1:Input CS signal(MSPInCSI) is recognized in slave mode.
28	MSPInSAMP	This bit controls the internal sampling timing for receive data in master mode. 0: The sampling timing of Master receive is standard sampling point of SPI protocol. 1: The sampling timing of Master receive is next edge sampling point of SPI protocol.
27	MSPInCKR	This bit selects MSPIn Clock default level 0:The default level of MSPInSCK is low. 1:The default level of MSPInSCK is high.
26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	MSPInSOLS [1:0]	These bits select the MSPInSOUT function in Master mode. 00: Set MSPInSOUT to low after macro enable, and holds the level after each transfer. 01: Set MSPInSOUT to low after macro enable and low during idle time. 10: Set MSPInSOUT to high after macro enable, and holds the level after each transfer 11: Set MSPInSOUT to high after macro enable and high during idle time.
23 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	MSPInCSP [7:0]	These bits control the polarity of MSPInCS. 0: The MSPInCS signal is active low. 1: The MSPInCS signal is active high.

CAUTIONS

1. This register should be written when MSPInCTL0.MSPInEN = 0.
2. In the master mode(MSPInMSSEL=0), MSPInCSIE must be set to 0.
3. In the transmission only setting, MSPInSAMP must be set to 0.
4. In the slave mode(MSPInMSSEL=1), MSPInSAMP must be set to 0.
5. In the slave mode(MSPInMSSEL=1), MSPInSOLS[1:0] must be set to 00.
6. In the slave mode(MSPInMSSEL=1), MSPInCSP[7:0] must be set to 00_H.
7. In the master mode (MSPInMSSEL=0), MSPInSAMP must be set to 1 for 20 MHz/40 MHz mode.

19.5.2.3 MSPInCTL2 — MSPI Control Register 2

This register is used to enable or disable the data consistency check and the loop-back mode.

Access: This register can be read or written in 8-bit units.

Address: <MSPIn_base> + 0008_H

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MSPInDCS	MSPInLBM
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Table 19.22 MSPInCTL2 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MSPInDCS	This bit enables or disables the data consistency check. 0: Disables the data consistency check. 1: Enables the data consistency check.
0	MSPInLBM	This bit enables or disables the loop-back mode. 0: Disables the loop-back mode. 1: Enables the loop-back mode.

CAUTIONS

1. This register can be written only if MSPInCTL0.MSPInEN = 0.
2. In slave mode (MSPInMSSEL=1), MSPInLBM must be set to 0.

19.5.2.4 MSPInSTR0 — MSPIn Status Register 0

This register indicates MSPI communication status.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <MSPIn_base> + 0010_H

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	MSPInCNUMF[2:0]			—	—	—	MSPInCSF
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Table 19.23 MSPInSTR0 Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 4	MSPInCNUMF [2:0]	These bits indicate the number of the communicating channel. 0: The communicating channel is channel 0 (or communication stops). 1: The communicating channel is channel 1. 2: The communicating channel is channel 2. 3: The communicating channel is channel 3. 4: The communicating channel is channel 4. 5: The communicating channel is channel 5 6: The communicating channel is channel 6. 7: The communicating channel is channel 7.
3 to 1	Reserved	When read, the value after reset is returned.
0	MSPInCSF	This bit indicates the communication status of master mode. (When MSPInCSF is 0, all channels stop, and MSPI does not communicate.) • Set condition When 1 is written to MSPInCSTSm.MSPInCHENSm • Clear condition – Master Mode: When communication is completed and MSPInCHENm of ALL channels are cleared and insertion of hold time is completed. – Slave mode: When communication is completed and MSPInCHEN0 is cleared.

19.5.2.5 MSPInCSTRm — MSPIn Channel Status Register m (m = 0 to 7)

This register indicates the status of the MSPIn channel.

Access: This register is a read-only register that can be read in 8-, 16-bit units.

Address: <MSPIn_base> + 58_H + 80_H x (m+1)

Value after reset: 0000_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInFIRXNm[7:0]								—	—	MSPIn TXRQFm	MSPIn RXRQFm	—	—	MSPIn ACTFm	MSPIn CHENm
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.24 MSPInCSTRm Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 8	MSPInFIRXNm [7:0]	These bits indicate the number of received data not being read in fixed FIFO memory mode. (These bits are fixed to 00_H in Direct memory mode or Fixed Buffer memory mode.) 0: The number of received data not read is 0. 1: The number of received data not read is 1. 2: The number of received data not read is 2. : 32: The number of received data not read is 32. These bits are used in the fixed FIFO memory mode when communication is stopped using the clear trigger (MSPInCSTCm.MSPInCHENCm). In the case of 1 frame 8 bits, if there is one unread data, this register indicates 1. In the case of 1 frame 32 bits, if there is one unread data, this register indicates 1. In the case of 1 frame 128 bits, if there is one unread data, this register indicates 4.
7, 6	Reserved	When read, the value after reset is returned.
5	MSPInTXRQFm	This bit indicates a transmission data write request in Direct memory mode or Fixed FIFO memory mode. (This bit is fixed to 0 in Fixed Buffer memory mode.) 0: There is no request to write transmission data. 1: There is a request to write transmission data.
4	MSPInRXRQFm	This bit indicates a reception data read request in Direct memory mode or Fixed FIFO memory mode. (This bit is fixed to 0 in Fixed Buffer memory mode.) 0: There is no request to read reception data. 1: There is a request to read reception data.
3, 2	Reserved	When read, the value after reset is returned.
1	MSPInACTFm	This bit indicates whether the channel is active or inactive. 0: Channel m is inactive. 1: Channel m is active. • Set condition 1. If 1 is written to MSPInCSTSm.MSPInACTFSm when MSPInCHENm is 1. 2. When 1 is written to MSPInCSTSm.MSPInACTFSm and MSPInCSTSm.MSPInCHENSm simultaneously. 3. If trigger selected by MSPInCFGm4.MSPInHWTSm[5:0] is input, when MSPInCHENm is 1. • Clear condition 1. If MSPInCFGm0.MSPInFCCE is 0 and the transmission of the last frame is finished. 2. When 1 is written to MSPInCSTCm.MSPInCHENCm. 3. Write MSPInCTL0.MSPInEN to 0. (MSPInEN 1→0)

Table 19.24 MSPInCSTRm Register Contents (2/2)

Bit Position	Bit Name	Function
0	MSPInCHENm	This bit indicates whether the channel is disabled or enabled. When 1 is written to MSPInCSTSm.MSPInCHENSm, this bit is set to 1 And when the channel is frame count end, this bit is cleared to 0 by HW. 0: Channel m is disabled. 1: Channel m is enabled. • Set condition 1. When 1 is written to MSPInCSTSm.MSPInCHENSm. • Clear condition 1. If MSPInCFGm0.MSPInFCCE is 0 and the transmission of the last frame is finished. 2. When 1 is written to MSPInCSTCm.MSPInCHENCm. 3. When 0 is written to MSPInCTL0.MSPInEN to 0. (MSPInEN 1→0)

19.5.2.6 MSPInCSTSm — MSPIn Channel Status Set Register m (m = 0 to 7)

This register is used to set the MSPInCSTRm register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <MSPIn_base> + 34_H + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MSPIn ACTFSm	MSPIn CHENSm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	W	W

Table 19.25 MSPInCSTSm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	MSPInACTFSm	This bit sets the channel active flag (MSPInCSTRm.MSPInACTFm). 0: No function. 1: Sets MSPInCSTRm.MSPInACTFm to 1.
0	MSPInCHENSm	This bit sets the channel enable flag (MSPInCSTRm.MSPInCHENm). 0: No function. 1: Sets MSPInCSTRm.MSPInCHENm to 1.

19.5.2.7 MSPInCSTCm — MSPIn Channel Status Clear Register m (m = 0 to 7)

This register is used to clear the MSPInCSTRm register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <MSPIn_base> + 38_H + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MSPIn CHENCm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	W

Table 19.26 MSPInCSTCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	MSPInCHENCm	This bit clears the channel enable flag. (This bit is the stop trigger of channel operation.) 0: No function. 1: Clears MSPInCSTRm.MSPInCHENm to 0.

19.5.2.8 MSPInCFGm0 — MSPIn channel Configuration Register m0 (m = 0 to 7)

This register is used to configure the channel operation.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 00_H + 80_H x (m+1)

Value after reset: 0070 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MSPIn TXEm	MSPIn RXEm	—	—	MSPIn MDm[1:0]		—	MSPInPRIOm[2:0]			—	—	—	MSPIn LOCKm
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MSPIn FCCEm	—	—	—	—	—	—	—	—	MSPIn IEREm	MSPIn IFEEm	MSPIn IRXEm	MSPIn ITXEm
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 19.27 MSPInCFGm0 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MSPInTXEm	This bit enables transmission. 0: Transmission disabled. 1: Transmission enabled.
28	MSPInRXEm	This bit enables reception. 0: Reception disabled. 1: Reception enabled.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	MSPInMDm[1:0]	This bits selects channel mode 00: Direct memory mode 01: Fixed buffer memory mode 10: Fixed FIFO memory mode 11: Setting prohibited
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	MSPInPRIOm [2:0]	This bits select channel priority. 000: Channel priority level 1(Highest priority) 001: Channel priority level 2 010: Channel priority level 3 : 111: Channel priority level 8(Lowest priority)
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	MSPInLOCKm	This bit enables channel lock operation. 0: Disables the channel m lock operation. 1: Enables the channel m lock operation.
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 19.27 MSPInCFGm0 Register Contents (2/2)

Bit Position	Bit Name	Function
12	MSPInFCCEm	This bit selects the operation of channel frame count end in the direct memory mode. 0:When a last frame ends, MSPInCHENm is cleared and the channel operation ends. 1:When a last frame ends, MSPInCHENm is not cleared and the channel operation continues.
11 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MSPInIEREm	This bit enables or disables the output of INTMSPInERRm interrupt. 0: Disables the interrupt output. 1: Enables the interrupt output.
2	MSPInIFEEm	This bit enables or disables the output of INTMSPInFEm interrupt. 0: Disables the interrupt output. 1: Enables the interrupt output.
1	MSPInIRXEm	This bit enables or disables the output of INTMSPInRXm interrupt. 0: Disables the interrupt output. 1: Enables the interrupt output.
0	MSPInITXEm	This bit enables or disables the output of INTMSPInTXm interrupt. 0: Disables the interrupt output. 1: Enables the interrupt output.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInPRIOm to 111 in the slave mode.**
3. **Set MSPInLOCKm to 0 in the slave mode.**
4. **Set MSPInFCCEm to 0 in the fixed buffer memory mode and fixed FIFO memory mode.**
5. **Set MSPInFCCEm to 0 in mode reception only setting of the direct memory mode.**
6. **When MSPInFCCEm is 1, set MSPInIFEEm to 0.**

19.5.2.9 MSPInCFGm1 — MSPIn channel Configuration Register m1 (m = 0 to 7)

This register is used to configure the channel operation.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 04_H + 80_H x (m+1)

Value after reset: 0100 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MSPIn CPOLm	MSPIn CPHAm	—	—	—	MSPIn DIRm	—	MSPIn ICLSm	MSPIn FIDLm	MSPIn CSRIm
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MSPIn SAFCMm	—	MSPIn SAFSm	MSPIn SAFEm	—	—	MSPInPSm[1:0]		—	—	—	MSPIn DECHKm
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Table 19.28 MSPInCFGm1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MSPInCPOLm	This bit selects the MSPInSCK polarity. 0: MSPInSCK is low during idle time 1: MSPInSCK is high during idle time
24	MSPInCPHAm	This bit selects the MSPInSCK phase. 0: Shifting bits out for transmission takes place on even-numbered edges, and sampling for reception takes place on odd-numbered edges. 1: Shifting bits out for transmission takes place on odd-numbered edges, and sampling for reception takes place on even-numbered edges.
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	MSPInDIRm	This bit selects the communication direction. 0: Data is transmitted/received with MSB first. 1: Data is transmitted/received with LSB first.
19	Reserved.	When read, the value after reset is returned. When writing, write the value after reset.
18	MSPInICLSm	This bit controls the MSPInCS level for idle time in the Direct memory mode 0: MSPInCS level is inactive for idle time 1: MSPInCS level holds active level for idle time.
17	MSPInFIDLm	This bit controls whether to insert an idle period after each end of a frame. 0: The idle time is not inserted each end of a frame. 1: The idle time is forcibly inserted after each end of a frame.
16	MSPInCSRIm	This bit controls whether to insert an idle period after the last frame end. 0: Holds MSPInCS at the active level. 1: MSPInCS returns to the inactive level.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MSPInSAFCMm	This bit masks the CRC error of the first frame in Safe-SPI protocol function. 0: Does not mask the CRC error of the first frame. 1: Masks the CRC error of the first frame.
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 19.28 MSPInCFGm1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>9</td><td>MSPInSAFSm</td><td>This bit selects the format of the Safe-SPI protocol function.
0:In-frame format
1:Out-of-frame format</td></tr>
<tr><td>8</td><td>MSPInSAFEm</td><td>This bit enables Safe-SPI protocol function.
0: Disables Safe SPI protocol function
1: Enables Safe SPI protocol function</td></tr>
<tr><td>7, 6</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>5, 4</td><td>MSPInPSm[1:0]</td><td>These bits select odd parity check, even parity check or 0 parity operation, when the parity check function is enabled.
<table>
<tr><th>MSPIn PSm[1]</th><th>MSPIn PSm[0]</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Adds odd parity</td><td>Odd parity bit is expected</td></tr>
<tr><td>0</td><td>1</td><td>Adds even parity</td><td>Even parity bit is expected</td></tr>
<tr><td>1</td><td>0</td><td>0 parity</td><td>No check</td></tr>
<tr><td>1</td><td>1</td><td>Prohibited</td><td>Prohibited</td></tr>
</table></td></tr>
<tr><td>3 to 1</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>0</td><td>MSPInDECHKm</td><td>This bit enables the parity check.
0: No check
1: parity check</td></tr>
</table>

CAUTIONS

1. This register should be written when MSPInCSTRm.MSPInCHENm = 0.
2. Set MSPInCPOLm to 0 in the slave mode.
   Set MSPInICLSm to 0 in the slave mode.
   Set MSPInICLSm to 0 in the fixed buffer memory mode and fixed FIFO memory mode.
   Set MSPInICLSm to 0 in the direct memory mode, in case only data reception is configured.
3. Set MSPInFIDLm to 0 in the slave mode.
4. Set MSPInCSRIm to 0 in the slave mode.
5. Set MSPInSAFEm to 0 in the slave mode.

19.5.2.10 MSPInCFGm2 — MSPIn channel Configuration Register m2 (m = 0 to 7)

This register is used to configure the channel operation.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + 08_H + 80_H x (m+1)

Value after reset: 0020_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MSPInFLENm[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.29 MSPInCFGm2 Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	MSPInFLENm [7:0]	These bits select frame length. 00_H: This setting is prohibited 01_H: This setting is prohibited 02_H: 2 bits 03_H: 3 bits 04_H: 4 bits 05_H: 5 bits 06_H: 6 bits 07_H: 7 bits 08_H: 8 bits : $1E_H$: 30 bits $1F_H$: 31 bits 20_H: 32 bits(default) : 80_H: 128 bits Settings other than above are prohibited.

CAUTION

This register should be written when MSPInCSTRm.MSPInCHENm = 0.

19.5.2.11 MSPInCFGm3 — MSPIn channel Configuration Register m3 (m = 0 to 7)

This register is used to configure the channel operation.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + $0C_H$ + 80_H x (m+1)

Value after reset: 0001_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MSPIn PRCSm[1:0]		—	—	—	MSPInCDIVm[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.30 MSPInCFGm3 Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	MSPInPRCSm [1:0]	These bits select communications clock frequency division in the master mode 00: MSPInCLK 01: MSPInCLK/4 10: MSPInCLK/16 11: MSPInCLK/64
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	MSPInCDIVm [4:0]	These bits select communications clock frequency division selection in master mode 00000: Setting prohibited 00001: MSPInCLK/2 00010: MSPInCLK/4 00011: MSPInCLK/6 00100: MSPInCLK/8 00101: MSPInCLK/10 00110: MSPInCLK/12 00111: MSPInCLK/14 01000: MSPInCLK/16 : 11101: MSPInCLK/58 11110: MSPInCLK/60 11111: MSPInCLK/62 $MSPInSCK \quad baud\ rate = \frac{MSPInCLK}{4^{MSPInPRCSm} \times MSPInCDIVm \times 2}$

CAUTIONS

1. **This register can be written only if MSPInCSTRm.MSPInCHENm = 0.**
2. **In the slave mode(MSPInMSSEL=1), MSPInCFGm3 must be set to 0001_H(default value).**

NOTE

Use by “PCLK/2 ≥ MSPInSCK”.

19.5.2.12 MSPInCFGm4 — MSPIn channel Configuration Register m4 (m = 0 to 7)

This register is used to configure the channel operation.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 10_H + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	MSPInHWTSm[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSPIn SIZEm[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 19.31 MSPInCFGm4 Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	MSPInHWTSm [5:0]	These bits enable HW trigger and select the HW trigger source. 000000: HW trigger disabled.(Only SW trigger is effective) 000001 to 11111111:HW trigger enabled.(SW trigger is also enabled) For details about HW trigger sources, see **Table 19.32, MSPI HW trigger sources**.
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MSPInSIZEm [1:0]	These bits select the stage size of FIFO Buffer in Fixed FIFO memory mode. 00: The stage size of Buffer is 8 in Fixed FIFO memory mode. 01: The stage size of Buffer is 16 in Fixed FIFO memory mode. 10: The stage size of Buffer is 32 in Fixed FIFO memory mode. 11: Setting prohibited

CAUTIONS

1. **This register can be written only if MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInSIZEm[1:0] to 00 in the direct memory mode or the fixed buffer memory mode.**

Table 19.32 MSPI HW trigger sources (1/2)

MSPInHWTSm[5:0]	Unit	Trigger source
000000	—	HW trigger disabled. The channel can be started only with the software trigger bit (MSPInCSTSm.MSPInACTFSm).
000001	Pin	External Interrupt 0 (INTP0)
000010	Pin	External Interrupt 1 (INTP1)
000011	Pin	External Interrupt 2 (INTP2)
000100	Pin	External Interrupt 3 (INTP3)
000101	TPTM0	TPTM0 an up timer interrupt for comparison value 0 (INTTPTMU00)
000110	TPTM0	TPTM0 an up timer interrupt for comparison value 1 (INTTPTMU01)
000111	TPTM0	TPTM0 an up timer interrupt for comparison value 2 (INTTPTMU02)

Table 19.32 MSPI HW trigger sources (2/2)

MSPInHWTSm[5:0]	Unit	Trigger source
001000	TPTM0	TPTM0 an up timer interrupt for comparison value 3 (INTTPTMU03)
001001	TPTM1	TPTM1 an up timer interrupt for comparison value 0 (INTTPTMU10)
001010	TPTM1	TPTM1 an up timer interrupt for comparison value 1 (INTTPTMU11)
001011	TPTM1	TPTM1 an up timer interrupt for comparison value 2 (INTTPTMU12)
001100	TPTM1	TPTM1 an up timer interrupt for comparison value 3 (INTTPTMU13)
001101	TPTM2	TPTM2 an up timer interrupt for comparison value 0 (INTTPTMU20)
001110	TPTM2	TPTM2 an up timer interrupt for comparison value 1 (INTTPTMU21)
001111	TPTM2	TPTM2 an up timer interrupt for comparison value 2 (INTTPTMU22)
010000	TPTM2	TPTM2 an up timer interrupt for comparison value 3 (INTTPTMU23)
010001	TPTM3	TPTM3 an up timer interrupt for comparison value 0 (INTTPTMU30)
010010	TPTM3	TPTM3 an up timer interrupt for comparison value 1 (INTTPTMU31)
010011	TPTM3	TPTM3 an up timer interrupt for comparison value 2 (INTTPTMU32)
010100	TPTM3	TPTM3 an up timer interrupt for comparison value 3 (INTTPTMU33)
010101	GTM	MCS0 Interrupt for channel (MCS0_IRQ0)
010110	GTM	MCS0 Interrupt for channel (MCS0_IRQ1)
010111	GTM	MCS0 Interrupt for channel (MCS0_IRQ2)
011000	GTM	MCS0 Interrupt for channel (MCS0_IRQ3)
011001	GTM	MCS0 Interrupt for channel (MCS0_IRQ4)
011010	GTM	MCS0 Interrupt for channel (MCS0_IRQ5)
011011	GTM	MCS0 Interrupt for channel (MCS0_IRQ6)
011100	GTM	MCS0 Interrupt for channel (MCS0_IRQ7)
011101	GTM	MCS2 Interrupt for channel (MCS2_IRQ0)
011110	GTM	MCS2 Interrupt for channel (MCS2_IRQ1)
011111	GTM	MCS2 Interrupt for channel (MCS2_IRQ2)
100000	GTM	MCS2 Interrupt for channel (MCS2_IRQ3)
100001 to 110000	—	Setting is prohibited
110001	TSG	TSG30 valley interrupt (INTTSG30IVLY)
110010	TSG	TSG30 peak interrupt (INTTSG30IPEK)
110011	TSG	TSG30 compare match interrupt 12 (INTTSG30I12)
110100	TSG	TSG30 compare match interrupt 11 (INTTSG30I11)
110101	TSG	TSG31 valley interrupt (INTTSG31IVLY)
110110	TSG	TSG31 peak interrupt (INTTSG31IPEK)
110111	TSG	TSG31 compare match interrupt 12 (INTTSG31I12)
111000	TSG	TSG31 compare match interrupt 11 (INTTSG31I11)
Other than above	—	Setting is prohibited

CAUTIONS

1. **When setting all except for 000000_B to 111000_B for MSPInHWTSm[5:0], a trigger becomes invalid.**
2. **U2A-EVA does not support the MCS0 and MCS2 interrupt as MSPI HW trigger.**

19.5.2.13 MSPInRASTADm — MSPI RAM Start Address Setting Register (m = 0 to 7)

This register is used to set start address of MSPI RAM in the fixed FIFO memory mode and fixed buffer memory mode.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + 28_H + 80_H x (m+1)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MSPInRASTAD[8:2]							—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 19.33 MSPInRASTADm Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 2	MSPInRASTAD [8:2]	These bits set start Address of MSPI RAM in Fixed FIFO memory mode or fixed buffer memory mode.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInRASTADm[8:2] to 0000_H in direct memory mode.**

19.5.2.14 MSPInSEUPm — MSPInSCK Setup Time Setting Register m (m = 0 to 7)

This register is used to control the operation of the set delay time from MSPInCS assertion to MSPInSCK operation start.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + 18_H + 80_H x (m+1)

Value after reset: 0001_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MSPInSEUPm [11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.34 MSPInSEUPm Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	MSPInSEUPm [11:0]	These bits set the delay between MSPInCS assertion and MSPInSCK operation start. 000_H: Prohibited 001_H to FFF_H : MSPInSCK delay time = MSPInSEUPm [11:0] × MSPInCLK.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInSEUPm to 0001_H in the slave mode.**

NOTE

When using the MSPI at master mode with the setting MSPInCFGm1.MSPInCPHAm = 0, set the period from CS active to the first edge of SCK to 1/2 or more of the communication rate by MSPInSEUPm.

19.5.2.15 MSPInHOLDm — Hold Time Setting Register m (m = 0 to 7)

This register is used to control the operation of the MSPInCS negation delay time.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + $1C_H$ + 80_H x (m+1)

Value after reset: 0001_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MSPInHOLDm [11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.35 MSPInHOLDm (m = 0 to 7) Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	MSPInHOLDm [11:0]	These bits set the delay time between MSPInSCK operation stop and MSPInCS negation delay time. 000_H: Prohibited 001_H to FFF_H: MSPInCS negation delay time = MSPInHOLDm [11:0] × MSPInCLK

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInHOLDm to 0001_H in Slave mode.**

19.5.2.16 MSPInIDLEm — Idle Time Setting Register m (m = 0 to 7)

This register is used to control the operation of the MSPInCS next frame delay time.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + 20_H + 80_H x (m+1)

Value after reset: 0001_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MSPInIDLEm [11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.36 MSPInIDLEm(m = 0 to 7) Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	MSPInIDLEm [11:0]	These bits set the delay between MSPInCS negation and MSPInCS assertion between two frames. 000_H: Prohibited 001_H to FFF_H: MSPInCS[7:0] next frame time = MSPInIDLEm [11:0] × MSPInCLK.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInIDLEm to 0001_H in slave mode.**

19.5.2.17 MSPInINDAm — Inter-Data Time Setting Register m (m = 0 to 7)

This register is used to control the operation of the inter-data time.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + 24_H + 80_H x (m+1)

Value after reset: 0000_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MSPInINDAm [11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.37 MSPInINDAm (m = 0 to 7) Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	MSPInINDAm [11:0]	These bits set the delay between MSPInSCK operation stop and MSPInSCK operation start between two frames with same MSPInCS configuration. 000_H-FFF_H : MSPInCS[7:0] next frame time = MSPInINDAm [11:0] × MSPInCLK.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0.**
2. **Set MSPInINDAm to 0000_H in slave mode.**

19.5.2.18 MSPInCFSETm — MSPIn Communication Frame Count Setting Register m (m = 0 to 7)

This register is used to set the number of frame count.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + $2C_H$ + 80_H x (m+1)

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInCFSETm[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.38 MSPInCFSETm Register Contents

Bit Position	Bit Name	Function
15 to 0	MSPInCFSETm [15:0]	These bits set the number of frame count. 0000_H: Setting prohibited 0001_H: The frame count number is 1 0002_H: The frame count number is 2 : $FFFE_H$: The frame count number is 65534 $FFFF_H$: The frame count number is 65535

CAUTION

This register should be written when MSPInCSTRm.MSPInCHENm = 0

19.5.2.19 MSPInSSELm —MSPInCS signal Selection Register m (m = 0 to 7)

This register is used to control the operation of the chip select activation and JOB function of Direct memory mode.

Access: This register can be read or written in 8-, 16-bit units.

Address: <MSPIn_base> + $3C_H$ + 80_H x (m+1)

Value after reset: 0000_H

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInJOBENm	—	—	—	—	—	—	—	MSPInCSRm [7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W	R/W	R/W	R/W

Note 1. Writable only for MSPI0 to 4

Table 19.39 MSPInSSELm Register Contents

Bit Position	Bit Name	Function
15	MSPInJOBENm	This bit activates the Job function for the next transfer. 0: Job ends with this frame. After this frame the channel with a higher priority can transfer data. 1: Job continuous after this frame. This and the next frame of channel m cannot be interrupted.
14 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	MSPInCSRm [7:0]	These bits set whether to activate or deactivate the MSPInCS[7:0]. 0: Deactivates MSPInCS[7:0] 1: Activates MSPInCS[7:0] for the communication.

CAUTIONS

1. **This register should be written when MSPInCSTRm.MSPInCHENm = 0 in the fixed buffer memory mode or fixed FIFO memory mode.**
2. **This register should be written when MSPInCSTRm.MSPInCHENm = 0 in reception only setting of the direct memory mode.**
3. **Set MSPInJOBENm to 0 in the fixed buffer memory mode, fixed FIFO memory mode, or slave mode.**
4. **Set MSPInSSELm to 0000_H in the slave mode.**

19.5.2.20 MSPInTXDAm0 — MSPIn Communication Transmit Data Register m0 (m = 0 to 7)

This register is used to write the transmit data.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 40_H + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSPInTXDAm0[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInTXDAm0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.40 MSPInTXDAm0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MSPInTXDAm0 [31:0]	These bits hold the transmit data (transmit data bits 31 to 0).

CAUTION

In the Fixed FIFO mode, it is necessary to access this register according to the frame length.

Frame length 2 to 8: Writing in units of 8 bits.

Frame length 9 to 16: Writing in units of 16 bits.

Frame length 32 to 128: Writing in units of 32 bits.

19.5.2.21 MSPInFRERSTm — MSPIn Frame Error Status Register m(m = 0 to 7)

This register indicates the error of status in communication frame.

This register is used to monitor communication error in each communication frame in the Direct memory mode.

This register is updated when the receive data is stored in the MSPInRXDAm0 register, and can be used to confirm the error occurrence status for each communication frame.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <MSPIn_base> + $4C_H$ + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	MSPIn DCEFSTm	—	—	MSPIn CEFSTm	MSPIn PEFSTm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Table 19.41 MSPInFRERSTm Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4	MSPInDCEFSTm	This bit indicates whether a transmission data consistency error was detected in communication frame. 0: No consistency error was detected on the associated transmission data in communication frame 1: A consistency error was detected on the associated transmission data in communication frame.
3, 2	Reserved	When read, the value after reset is returned.
1	MSPInCEFSTm	This bit indicates whether a reception data CRC error was detected in communication frame. 0: No CRC error was detected on the associated reception data in communication frame. 1: A CRC error was detected on the associated reception data in communication frame.
0	MSPInPEFSTm	This bit indicates whether a reception data parity error was detected in communication frame. 0: No parity error was detected on the associated reception data in communication frame. 1: A parity error was detected on the associated reception data in communication frame.

19.5.2.22 MSPInFRERSTCm — MSPIn Frame Error Status Clear Register m(m = 0 to 7)

This register is used to clear the MSPInFRERSTm register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <MSPIn_base> + 64_H + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	—	—	—	MSPIn DCEFSCm	—	—	MSPIn CEFSCm	MSPIn PEFSCm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	W	R	R	W	W

Table 19.42 MSPInFRERSTCm Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When writing, write the value after reset.
4	MSPInDCEFSCm	This bit clears the status flag of data consistency error. 0: No function. 1: Clears MSPInFRERSTm.MSPInDCEFSTm to 0.
3, 2	Reserved	When writing, write the value after reset.
1	MSPInCEFSCm	This bit clears the status flag of CRC error. 0: No function. 1: Clears MSPInFRERSTm.MSPInCEFSTm to 0.
0	MSPInPEFSCm	This bit clears the status flag of Parity error. 0: No function. 1: Clears MSPInFRERSTm.MSPInPEFSTm to 0.

19.5.2.23 MSPInRXDAm0 — MSPIn Communication Receive Data Register m0 (m = 0 to 7)

This register is used to control the receive data.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 50_H + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSPInRXDAm0[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInRXDAm0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.43 MSPInRXDAm0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MSPInRXDAm0 [31:0]	These bits store the received data (received data bits 31 to 0).

CAUTION

In the Fixed FIFO mode, it is necessary to access this register according to the frame length.

Frame length 2 to 8: Reading in units of 8 bits.

Frame length 9 to 16: Reading in units of 16 bits.

Frame length 32 to 128: Reading in units of 32 bits.

19.5.2.24 MSPInSSTXm — MSPIn SSL Selection and Communication Transmit Data Register m (m = 0 to 7)

This register is used to write the 16bit transmit data, chip select activation and JOB information with one access.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 30_H + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSPInJOBENm	—	—	—	—	—	—	—	MSPInCSRm[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInTXDAm0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Writable only for MSPI0 to 4

Table 19.44 MSPInSSTXm Register Contents

Bit Position	Bit Name	Function
31	MSPInJOBENm	This bit activates the Job function for the next transfer. 0: Job ends with this frame. After this frame the channel with a higher priority can transfer data. 1: Job continuous after this frame. This and the next frame of channel m cannot be interrupted. This bit is the mirror bit of MSPInSSELm.MSPInJOBENm.
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MSPInCSRm [7:0]	These bits set whether to activate or deactivate the MSPInCS[7:0]. 0: Deactivates MSPInCS[7:0]. 1: Activates MSPInCS[7:0] for the communication. These bits are the mirror bits of MSPInSSELm.MSPInCSRm[7:0].
15 to 0	MSPInTXDAm0 [15:0]	These bits hold the transmission data (transmission data bits 15 to 0). These bits are the mirror bits of MSPInTXDAm0[15:0].

CAUTION

This register is used to write MSPInJOBENm, MSPInCSRm[7:0], and MSPInTXDAm0[15:0] at the same time when communicating in direct memory mode with a frame length of 16 bits or less.

19.5.2.25 MSPInFSRXm — MSPIn Frame Error Status and Communication Receive Data Register m (m = 0 to 7)

This register is used to read the receive data and error status of communication frame.

This register reading will be affect same as MSPInRXDAm0 reading. (Clearing MSPInRXRQFm)

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + 60_H + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MSPIn DEFSTm	—	—	MSPIn CEFSTm	MSPIn PEFSTm
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInRXDAm0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.45 MSPInFSRXm Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20	MSPInDEFSTm	This bit indicates whether a transmission data consistency error was detected in communication frame. 0: No consistency error was detected on the associated transmission data in communication frame 1: A consistency error was detected on the associated transmission data in communication frame. This bit is the mirror bit of MSPInFRERSTm.MSPInDCEFSTm.
19 to 18	Reserved	When read, the value after reset is returned.
17	MSPInCEFSTm	This bit indicates whether a reception data CRC error was detected in communication frame. 0: No CRC error was detected on the associated reception data in communication frame. 1: A CRC error was detected on the associated reception data in communication frame. This bit is the mirror bit of MSPInFRERSTm.MSPInCEFSTm.
16	MSPInPEFSTm	This bit indicates whether a reception data parity error was detected in communication frame. 0: No parity error was detected on the associated reception data in communication frame. 1: A parity error was detected on the associated reception data in communication frame. This bit is the mirror bit of MSPInFRERSTm.MSPInPEFSTm.
15 to 0	MSPInRXDAm0 [15:0]	These bits hold the receive data (receive data bit 15 to 0). These bits are the mirror bits of MSPInRXDAm0[15:0].

19.5.2.26 MSPInCESTm — MSPIn Channel Error Status Register m (m = 0 to 7)

This register indicates the channel error status of MSPIn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <MSPIn_base> + 68_H + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	MSPIn OVREEm	MSPIn OVWREm	MSPIn OVRUEm	MSPIn DCEm	—	—	MSPIn CEm	MSPIn PEm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Table 19.46 MSPInCESTm Register Contents

Bit Position	Bit Name	Function
7	MSPInOVREEm	This bit indicates whether an over-read error was detected in Fixed FIFO memory mode. 0: No over-read Error was detected. 1: An over-read error was detected.
6	MSPInOVWREm	This bit indicates whether an over-write error was detected in Fixed FIFO memory mode. 0: No over-write error was detected. 1: An over-write error was detected.
5	MSPInOVRUEm	This bit indicates whether an over-run error was detected in Direct memory mode or Fixed FIFO memory mode. 0: No overrun error was detected. 1: An overrun error was detected.
4	MSPInDCEm	This bit indicates whether a transmission data consistency error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
3, 2	Reserved	When read, the value after reset is returned.
1	MSPInCEm	Indicates whether a reception data CRC error was detected in communication frame. 0: No CRC error was detected on the associated reception data in communication frame. 1: A CRC error was detected on the associated reception data in communication frame.
0	MSPInPEm	Indicates whether a reception data parity error was detected in communication frame. 0: No parity error was detected on the associated reception data in communication frame. 1: A parity error was detected on the associated reception data in communication frame.

19.5.2.27 MSPInCESTCm — MSPIn Channel Status Clear Register m (m = 0 to 7)

This register is used to clear the Error flag of MSPInCESTm register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <MSPIn_base> + $6C_H$ + 80_H x (m+1)

Value after reset: 00_H

Bit:	7	6	5	4	3	2	1	0
	MSPIn OVREECm	MSPIn OVWRECm	MSPIn OVRUECm	MSPIn DCECm	—	—	MSPIn CECm	MSPIn PECm
Value after reset:	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	R	R	W	W

Table 19.47 MSPInCESTCm Register Contents

Bit Position	Bit Name	Function
7	MSPInOVREECm	This bit clears the status flag of over-read error. 0: No function. 1: Clear MSPInCESTm. MSPInOVREEm to 0.
6	MSPInOVWRECm	This bit clears the status flag of over-write error. 0: No function. 1: Clear MSPInCESTm.MSPInOVWREm to 0.
5	MSPInOVRUECm	This bit clears the status flag of overrun error. 0: No function. 1: Clear MSPInCESTm.MSPInOVRUEm to 0.
4	MSPInDCECm	This bit clears the status flag of data consistency error. 0: No function. 1: Clear MSPInCESTm.MSPInDCEm to 0.
3, 2	Reserved	When writing, write the value after reset.
1	MSPInCECm	This bit clears the status flag of CRC error. 0: No function. 1: Clear MSPInCESTm.MSPInCEm to 0.
0	MSPInPECm	This bit clears the status flag of reception data parity error. 0: No function. 1: Clear MSPInCESTm.MSPInPEm to 0.

19.5.2.28 MSPInCFCNTm — MSPIn Channel Frame Count Register m (m = 0 to 7)

This register is used to confirm the remaining frame count number.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <MSPIn_base> + $5C_H$ + 80_H x (m+1)

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSPInCFCNTm[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.48 MSPInCFCNTm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	MSPInCFCNTm [15:0]	These bits indicate the remaining frame count number.

19.5.2.29 MSPInINTFk — MSPIn Interrupt Factor Register (k = 0 to 3)

These registers contain information about which MSPIn interrupt (INTMSPInTXm, INTMSPInRXm, INTMSPInFEm, INTMSPInERRm) has been generated without depending on MSPInINTMSK setting.

Regarding block diagram image, see **Figure 19.1, MSPI0-1 Interrupt Connection Image** and **Figure 19.2, MSPI2-9 Interrupt Connection Image**.

Access: These registers are read-only register that can be read in 32-bit units.

Address: <MSPIn_INTF_base> + k × 30_H + 00_H

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MSPInINTFk[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.49 MSPInINTFk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	MSPInINTFk [7:0]	MSPIn interrupts occurrence 0: No interrupt occurred 1: interrupt has occurred The bits correspond to the following interrupts. MSPInINTF0[7:0] : INTMSPInTX MSPInINTF1[7:0] : INTMSPInRX MSPInINTF2[7:0] : INTMSPInFE MSPInINTF3[7:0] : INTMSPInERR The number of bits of MSPInINTFk is same as the number of channels of MSPIn. Refer to **Table 19.3, Number of Transfer Control Channels and Chip Select Index**.

19.5.2.30 MSPInINTMSKk — MSPIn Interrupt Mask Register (k = 0 to 3)

These registers mask MSPIn interrupts output to INTC2 by each channels.

Access: These registers can be read or written in 32-bit units.

Address: <MSPIn_INTF_base> + k × 30_H + 04_H

Value after reset: 0000 $00FF_H$ (MSPI0-4), 0000 $000F_H$ (MSPI5-9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MSPInINTMSKk[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0/1*1	0/1*1	0/1*1	0/1*1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W*2	R/W*2	R/W*2	R/W*2	R/W	R/W	R/W	R/W

Note 1. MSPI0-4: 1_B
MSPI5-9: 0_B

Note 2. Writable only for MSPI0-4

Table 19.50 MSPInINTMSKk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	MSPInINTMSKk [7:0]	MSPIn interrupts output to INTC2 mask 0: Not masked 1: Masked The bits correspond to the following interrupts. MSPInINTMSK0[7:0] : INTMSPInTX MSPInINTMSK1[7:0] : INTMSPInRX MSPInINTMSK2[7:0] : INTMSPInFE MSPInINTMSK3[7:0] : INTMSPInERR The number of bits of MSPInINTMSKk is same as the number of channels of MSPIn. Refer to **Table 19.3, Number of Transfer Control Channels and Chip Select Index**.

19.5.2.31 MSPInINTFCk — MSPIn Interrupt Factor Clear (k = 0 to 3)

These registers clear MSPIn interrupt factor register (MSPInINTFk) to INTC2 by each channels.

Access: These registers are write-only register that can be written in 32-bit units.

Address: <MSPIn_INTF_base> + k × 30_H + 08_H

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MSPInINTFCk[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	W*1	W*1	W*1	W*1	W	W	W	W

Note 1. Writable only for MSPI0-4

Table 19.51 MSPInINTFCk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	MSPInINTFCk [7:0]	MSPInINTFCk flag clear 0: — 1: Clear The bits correspond to the following interrupts. MSPInINTFC0[7:0] : INTMSPInTX MSPInINTFC1[7:0] : INTMSPInRX MSPInINTFC2[7:0] : INTMSPInFE MSPInINTFC3[7:0] : INTMSPInERR The number of bits of MSPInINTFCk is same as the number of channels of MSPIn. Refer to **Table 19.3, Number of Transfer Control Channels and Chip Select Index**.

19.5.3 DMA/DTS Trigger Generator Registers

19.5.3.1 MSPITGCTLn — MSPI Trigger Control Register for MSPIn (n = 0 to 9)

This register is used to control DMA/DTS trigger generation

For details about DMA/DTS trigger generation, see **Section 19.4, DMA/DTS Trigger Generation**.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPITG_base> + n × 04_H

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRCSEL7[2:0]			TRGSEL7	SRCSEL6[2:0]			TRGSEL6	SRCSEL5[2:0]			TRGSEL5	SRCSEL4[2:0]			TRGSEL4
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRCSEL3[2:0]			TRGSEL3	SRCSEL2[2:0]			TRGSEL2	SRCSEL1[2:0]			TRGSEL1	SRCSEL0[2:0]			TRGSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. Writable only for MSPITGCTL0-4 registers

Table 19.52 MSPITGCTLn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 29	SRCSEL7[2:0]	Trigger Source Select For Channel 7 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX7, Trigger 2=None 010: Trigger 1=INTMSPInRX7, Trigger 2=None 011: Trigger 1=INTMSPInFE7 Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX7 101: Trigger 1=None, Trigger 2=INTMSPInRX7 110: Trigger 1=None, Trigger 2=INTMSPInFE7 111: Trigger 1=INTMSPInTX7, Trigger 2=INTMSPInRX7 These bits are valid for MSPITGCTL0-4 registers only
28	TRGSEL7	Trigger Select for Channel 7 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled. This bit is valid for MSPITGCTL0-4 registers only
27 to 25	SRCSEL6[2:0]	Trigger Source Select For Channel 6 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX6, Trigger 2=None 010: Trigger 1=INTMSPInRX6, Trigger 2=None 011: Trigger 1=INTMSPInFE6, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX6 101: Trigger 1=None, Trigger 2=INTMSPInRX6 110: Trigger 1=None, Trigger 2=INTMSPInFE6 111: Trigger 1=INTMSPInTX6, Trigger 2=INTMSPInRX6 These bits are valid for MSPITGCTL0-4 registers only
24	TRGSEL6	Trigger Select for Channel 6 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled. This bit is valid for MSPITGCTL0-4 registers only

Table 19.52 MSPITGCTLn Register Contents (2/3)

Bit Position	Bit Name	Function
23 to 21	SRCSEL5[2:0]	Trigger Source Select For Channel 5 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX5, Trigger 2=None 010: Trigger 1=INTMSPInRX5, Trigger 2=None 011: Trigger 1=INTMSPInFE5, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX5 101: Trigger 1=None, Trigger 2=INTMSPInRX5 110: Trigger 1=None, Trigger 2=INTMSPInFE5 111: Trigger 1=INTMSPInTX5, Trigger 2=INTMSPInRX5 These bits are valid for MSPITGCTL0-4 registers only
20	TRGSEL5	Trigger Select for Channel 5 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled. This bit is valid for MSPITGCTL0-4 registers only
19 to 17	SRCSEL4[2:0]	Trigger Source Select For Channel 4 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX4, Trigger 2=None 010: Trigger 1=INTMSPInRX4, Trigger 2=None 011: Trigger 1=INTMSPInFE4, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX4 101: Trigger 1=None, Trigger 2=INTMSPInRX4 110: Trigger 1=None, Trigger 2=INTMSPInFE4 111: Trigger 1=INTMSPInTX4, Trigger 2=INTMSPInRX4 These bits are valid for MSPITGCTL0-4 registers only
16	TRGSEL4	Trigger Select for Channel 4 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled. This bit is valid for MSPITGCTL0-4 registers only
15 to 13	SRCSEL3[2:0]	Trigger Source Select For Channel 3 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX3, Trigger 2=None 010: Trigger 1=INTMSPInRX3, Trigger 2=None 011: Trigger 1=INTMSPInFE3, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX3 101: Trigger 1=None, Trigger 2=INTMSPInRX3 110: Trigger 1=None, Trigger 2=INTMSPInFE3 111: Trigger 1=INTMSPInTX3, Trigger 2=INTMSPInRX3
12	TRGSEL3	Trigger Select for Channel 3 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled.
11 to 9	SRCSEL2[2:0]	Trigger Source Select For Channel 2 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX2, Trigger 2=None 010: Trigger 1=INTMSPInRX2, Trigger 2=None 011: Trigger 1=INTMSPInFE2, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX2 101: Trigger 1=None, Trigger 2=INTMSPInRX2 110: Trigger 1=None, Trigger 2=INTMSPInFE2 111: Trigger 1=INTMSPInTX2, Trigger 2=INTMSPInRX2
8	TRGSEL2	Trigger Select for Channel 2 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled.
7 to 5	SRCSEL1[2:0]	Trigger Source Select For Channel 1 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX1, Trigger 2=None 010: Trigger 1=INTMSPInRX1, Trigger 2=None 011: Trigger 1=INTMSPInFE1, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX1 101: Trigger 1=None, Trigger 2=INTMSPInRX1 110: Trigger 1=None, Trigger 2=INTMSPInFE1 111: Trigger 1=INTMSPInTX1, Trigger 2=INTMSPInRX1

Table 19.52 MSPITGCTLn Register Contents (3/3)

Bit Position	Bit Name	Function
4	TRGSEL1	Trigger Select for Channel 1 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled.
3 to 1	SRCSEL0[2:0]	Trigger Source Select For Channel 0 000: Trigger 1=None, Trigger 2=None 001: Trigger 1=INTMSPInTX0, Trigger 2=None 010: Trigger 1=INTMSPInRX0, Trigger 2=None 011: Trigger 1=INTMSPInFE0, Trigger 2=None 100: Trigger 1=None, Trigger 2=INTMSPInTX0 101: Trigger 1=None, Trigger 2=INTMSPInRX0 110: Trigger 1=None, Trigger 2=INTMSPInFE0 111: Trigger 1=INTMSPInTX0, Trigger 2=INTMSPInRX0
0	TRGSEL0	Trigger Select for Channel 0 0: DMA Trigger is enabled, and DTS Trigger is disabled. 1: DTS Trigger is enabled, and DMA Trigger is disabled.

19.5.3.2 MSPITGDMAALT — MSPI DMA Trigger Alternative Select Register

This register is used to select alternative DMA trigger signals

For details about DMA/DTS trigger generation, see **Section 19.4, DMA/DTS Trigger Generation**

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPITG_base> + 28_H

Value after reset: 0000 0000_H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DMAAS17	DMAAS16	DMAAS13	DMAAS12	DMAAS07	DMAAS06	DMAAS03	DMAAS02
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.53 MSPITGDMAALT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DMAAS17	DMA Trigger Alternative Select for MSPI1-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
6	DMAAS16	DMA Trigger Alternative Select for MSPI1-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
5	DMAAS13	DMA Trigger Alternative Select for MSPI1-Ch3 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
4	DMAAS12	DMA Trigger Alternative Select for MSPI1-Ch2 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
3	DMAAS07	DMA Trigger Alternative Select for MSPI0-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
2	DMAAS06	DMA Trigger Alternative Select for MSPI0-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
1	DMAAS03	DMA Trigger Alternative Select for MSPI0-Ch3 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
0	DMAAS02	DMA Trigger Alternative Select for MSPI0-Ch2 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals

19.5.3.3 MSPITGDTSALT — MSPI DTS Trigger Alternative Select Register

This register is used to select alternative DTS trigger signals

For details about DMA/DTS trigger generation, see **Section 19.4, DMA/DTS Trigger Generation**

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <MSPITG_base> + $2C_H$

Value after reset: $0000\ 0000_H$

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DTSAS37	DTSAS36	DTSAS35	DTSAS34	DTSAS27	DTSAS26	DTSAS25	DTSAS24	DTSAS17	DTSAS16	DTSAS15	DTSAS14	DTSAS07	DTSAS06
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.54 MSPITGDTSALT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	DTSAS37	DTS Trigger Alternative Select for MSPI3-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
12	DTSAS36	DTS Trigger Alternative Select for MSPI3-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
11	DTSAS35	DTS Trigger Alternative Select for MSPI3-Ch5 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
10	DTSAS34	DTS Trigger Alternative Select for MSPI3-Ch4 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
9	DTSAS27	DTS Trigger Alternative Select for MSPI2-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
8	DTSAS26	DTS Trigger Alternative Select for MSPI2-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
7	DTSAS25	DTS Trigger Alternative Select for MSPI2-Ch5 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
6	DTSAS24	DTS Trigger Alternative Select for MSPI2-Ch4 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
5	DTSAS17	DTS Trigger Alternative Select for MSPI1-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
4	DTSAS16	DTS Trigger Alternative Select for MSPI1-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals

Table 19.54 MSPITGDTSALT Register Contents (2/2)

Bit Position	Bit Name	Function
3	DTSAS15	DTS Trigger Alternative Select for MSPI1-Ch5 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
2	DTSAS14	DTS Trigger Alternative Select for MSPI1-Ch4 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
1	DTSAS07	DTS Trigger Alternative Select for MSPI0-Ch7 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals
0	DTSAS06	DTS Trigger Alternative Select for MSPI0-Ch6 0: Not use Alternative Trigger Signals 1: Use Alternative Trigger Signals

19.6 Basic Operation

19.6.1 Master/Slave Mode Operation

19.6.1.1 Master Mode Operation

The master mode is enabled by setting MSPInCTL1.MSPInMSSEL to 0.

In the master mode, all channels can be used.

The serial communication clock frequency and the active chip select can be specified individually for each channels.

The serial communication clock(MSPInSCK) is generated by the internal baud rate generator and the frequency division of MSPInSCK is set by MSPInCFGm3.MSPInPRCSm[1:0] and MSPInCFGm3.MSPInCDIVm[4:0].

One or more chip select signals can be selected MSPInSSELm.MSPInCSRm[7:0].

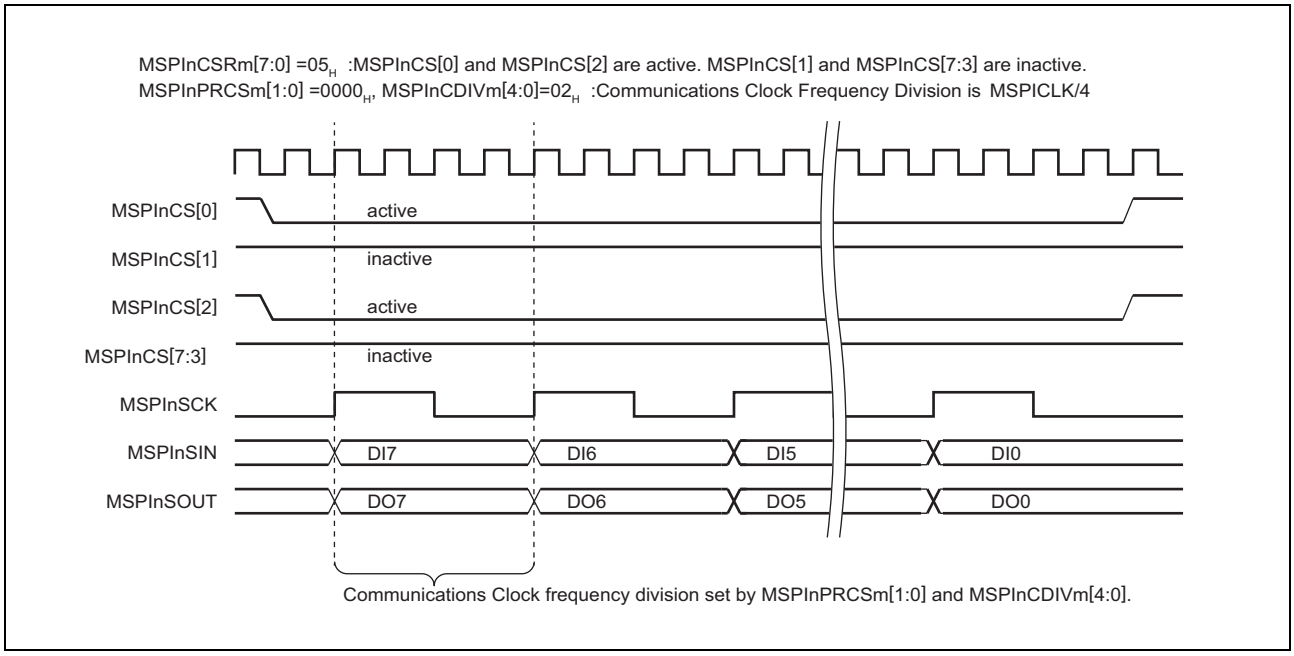


Figure 19.4 Communication in the Master mode

19.6.1.2 Slave Mode Operation

The slave mode is enabled by setting MSPInCTL1.MSPInMSSEL to 1.

In the slave mode, another device fact as the communication master and this supplies the serial communication clock(MSPInSCKI).

In the slave mode, the slave select signal(MSPInCSI) can be enabled/disabled by MSPInCTL1.MSInCSIE.

When MSPInCSI is disabled (MSPInCSIE = 0), normal transmit/receive operation is started as soon as the MSPInSCKI is detected.

When MSPInCSI is enabled (MSPInCSIE = 1), normal transmit/receive operation is started as soon as the MSPInSCKI is detected with the MSPInCSI in an active state.

In the slave mode, only channel 0 can be used.

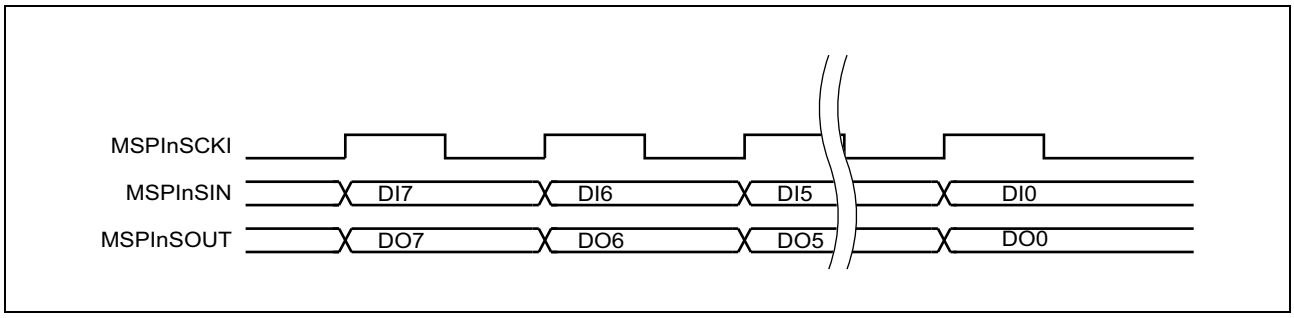


Figure 19.5 Communication in the Slave Mode (MSPInCSIE = 0)

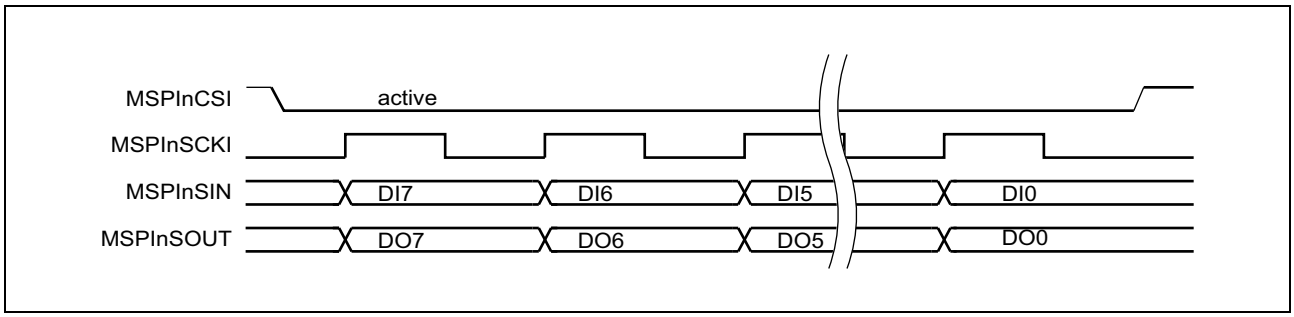


Figure 19.6 Communication in the Slave Mode (MSPInCSIE = 1)

19.6.2 Frame Length and Frame Count

Communication of MSPIn is executed by a preset frame length and frame count.

The frame length is set by MSPInCFGm2.MSPInFLENm[7:0], and the frame count is set by MSPInCFSETm.

The formats of frame length and frame count are shown in the figure below.

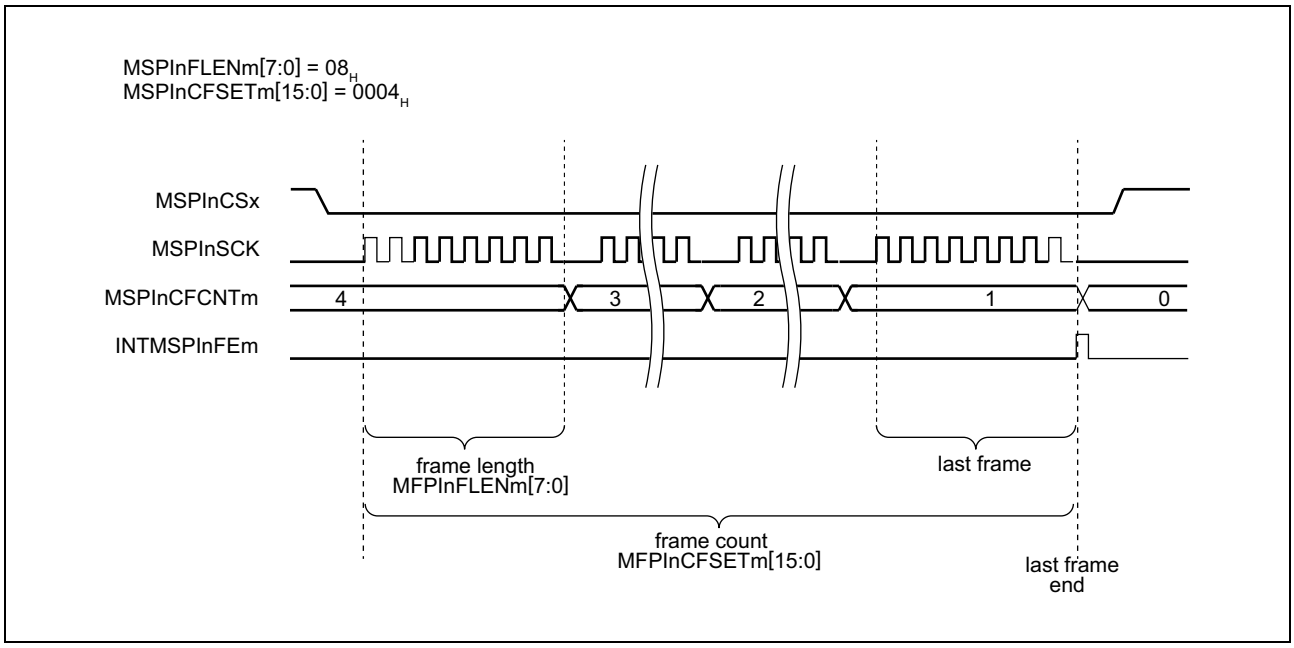


Figure 19.7 Formats of Frame Length and Frame Count

19.6.3 Chip Select Signal Function

19.6.3.1 Active or Inactive Settings of Chip Select Signal

In the master mode, MSPI can communicate with one or more slaves can be performed by using the chip select signals (MSPInCS[7:0]).

The active or inactive state of MSPInCS[7:0] is controlled by MSPInCSRm[7:0] bits.

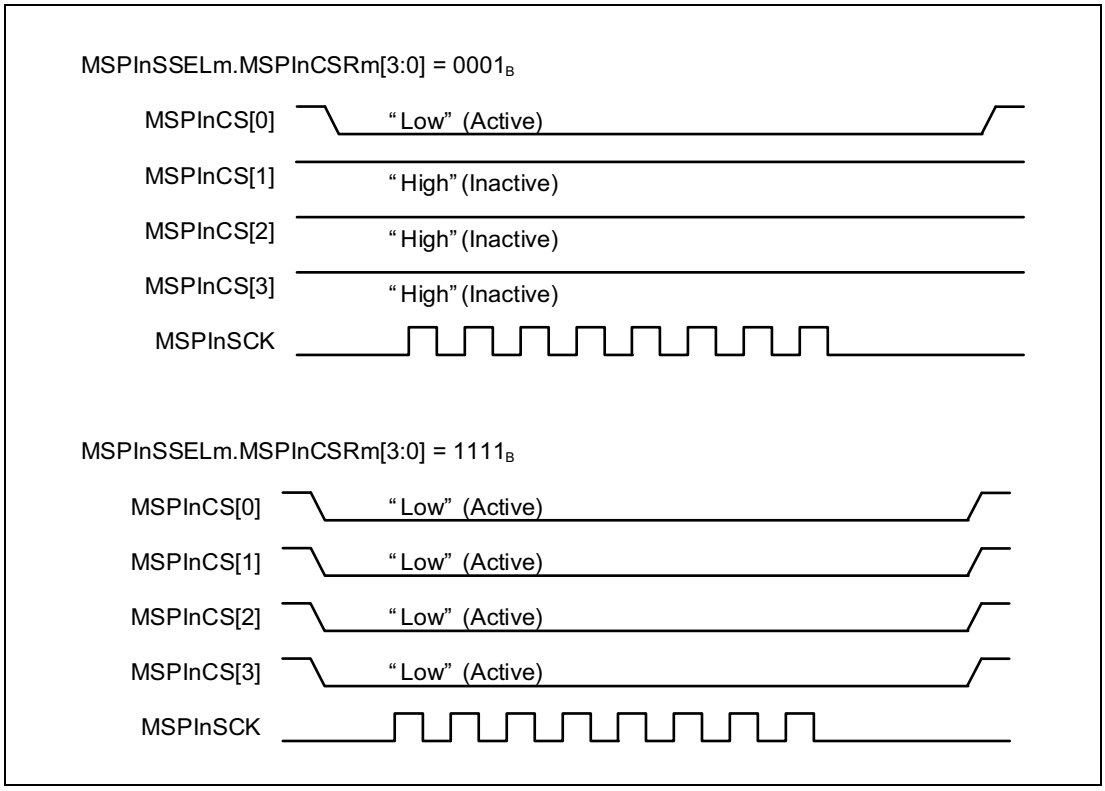


Figure 19.8 Chip Select Signal Control by MSPInCSRm Setting

CAUTION

If MSPInCSRm of multiple channels are the exact same, following parameters have to be set the same value.

- **MSPInCPOLm: Setting of MSPInSCK clock polarity**
- **MSPInCPHAm Data Phase: Setting of MSPInSCK clock phase**
- **MSPInDIRm: Setting of communication direction**
- **MSPInFIDLm: Setting of forced idle insertion.**
- **MSPInSAFCMm: Setting of Safe-SPI protocol format**
- **MSPInSAFSm: Setting of Safe-SPI protocol format**
- **MSPInSAFEm: Enable of Safe-SPI protocol function**
- **MSPInPSm[1:0]: Setting of parity check format**
- **MSPInDECHKm: Enable of parity check**
- **MSPInFLENm[7:0]: Setting of communication frame length**
- **MSPInPRCSm[1:0]: Setting of MSPInSCK baud rate**
- **MSPInCDIVm[4:0]: Setting of MSPInSCK baud rate**
- **MSPInSEUPm[11:0]: Setting of setup time**
- **MSPInHOLDm[11:0]: Setting of hold time**
- **MSPInIDLEm[11:0]: Setting of idle time**
- **MSPInINDAm[11:0]: Setting of inter-data time**

19.6.3.2 Active Output Level Settings of Chip Select Signal

The active or inactive level of MSPInCS[7:0] is set by MSPInCSP [7:0] bits.

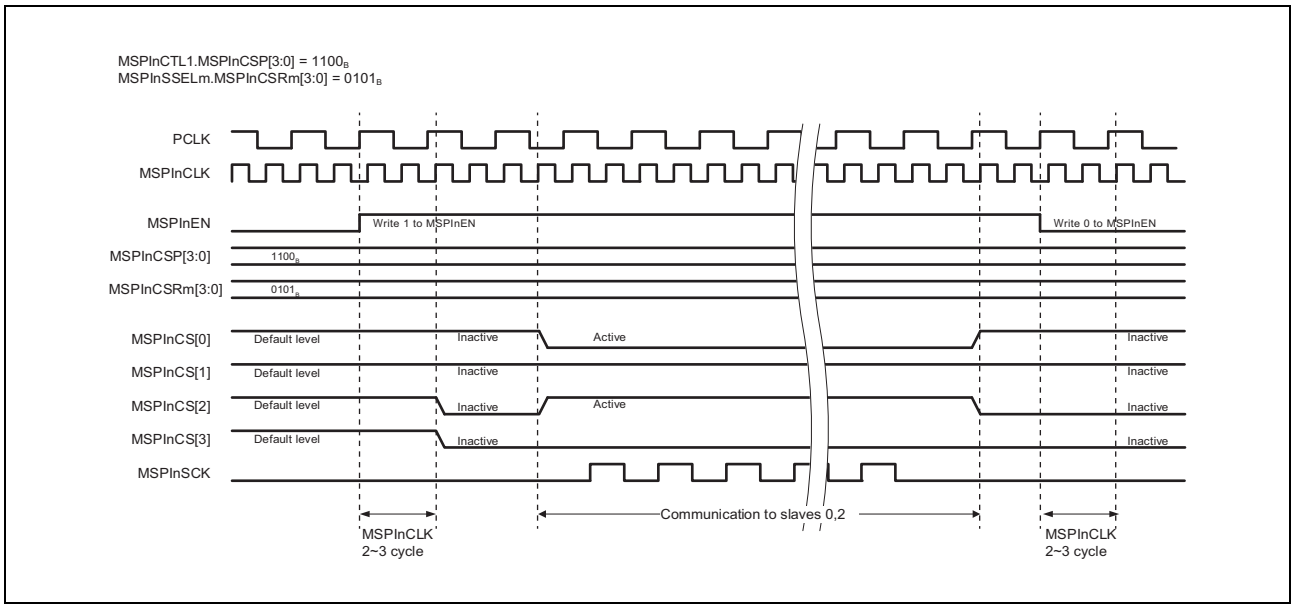
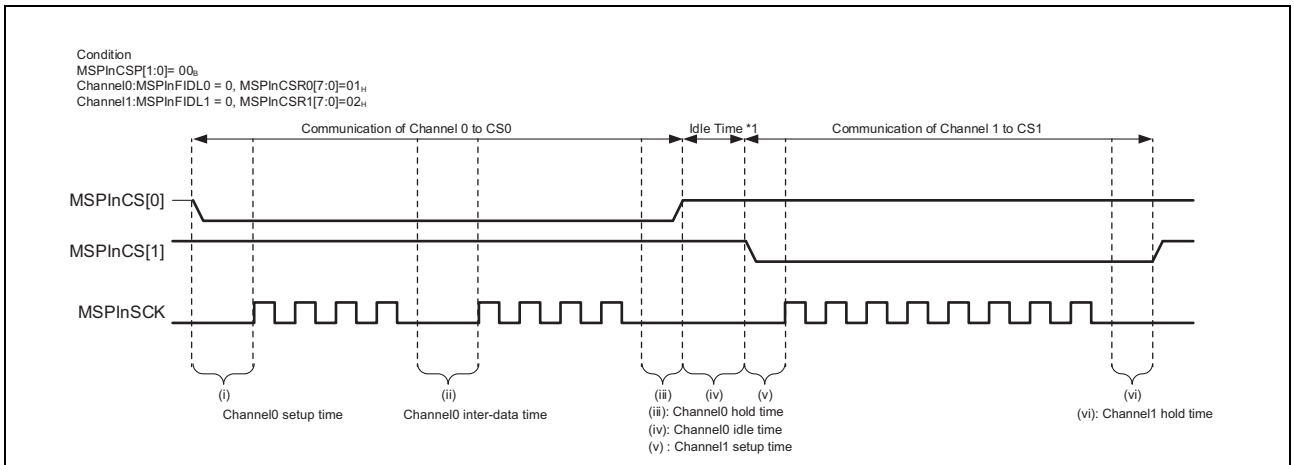


Figure 19.9 Chip Select Signal Control by MSPInCSP[7:0] Setting

19.6.3.3 Setup/Hold/Idle/Inter-Data Time Settings of Chip Select Signal

The setup time, the hold time, the idle time, and the inter-data time of MSPInCS can be set independently for each channel by MSPInSEUPm, MSPInHOLDm, MSPInIDLEm, and MSPInINDAm, respectively.

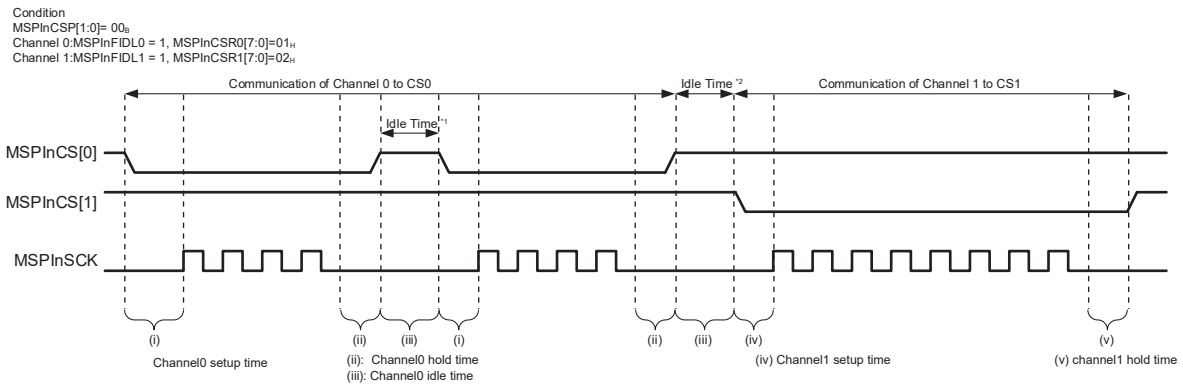
And an idle time can be inserted between two consecutive communications by setting MSPInFIDLm to 1.



(i): Channel0 setup time = MSPInSEUP0[11:0] x MSPInCLK
(ii): Channel0 inter-date time = MSPInINDA0[11:0] x MSPInCLK
(iii): Channel0 hold time = MSPInHOLD0 x MSPInCLK
(iv): Channel0 idle time = MSPInIDLE0 x MSPInCLK
(v): Channel1 setup time = MSPInSEUP1[11:0] x MSPInCLK
(vi): Channel1 hold time = MSPInHOLD1[11:0] x MSPInCLK

Note1. This idle time is inserted because the CS signal has changed(from CS0 to CS1).

Figure 19.10 Chip Select Signal Control by MSPInCSRm Setting (MSPInFIDLm = 0)



(i): Channel0 setup time = MSPInSEUP0[11:0] x MSPInCLK
(ii): Channel0 hold time = MSPInHOLD0[11:0] x MSPInCLK
(iii): Channel0 idle time = MSPInIDLE0 x MSPInCLK
(iv): Channel1 setup time = MSPInSEUP1[11:0] x MSPInCLK
(v): Channel1 hold time = MSPInHOLD1[11:0] x MSPInCLK

Note 1. This idle time is inserted because the MSPInFIDL0 is 1.
Note 2. This idle time is inserted because the CS signal has changed(from CS0 to CS1).

Figure 19.11 Chip Select Signal Control by MSPInCSRm Setting(MSPInFIDLm=1)

19.6.3.4 Level of chip select signal after last frame end

The level of MSPInCS[7:0] after the last frame end can be set independently for each channel by MSPInCSRIm.

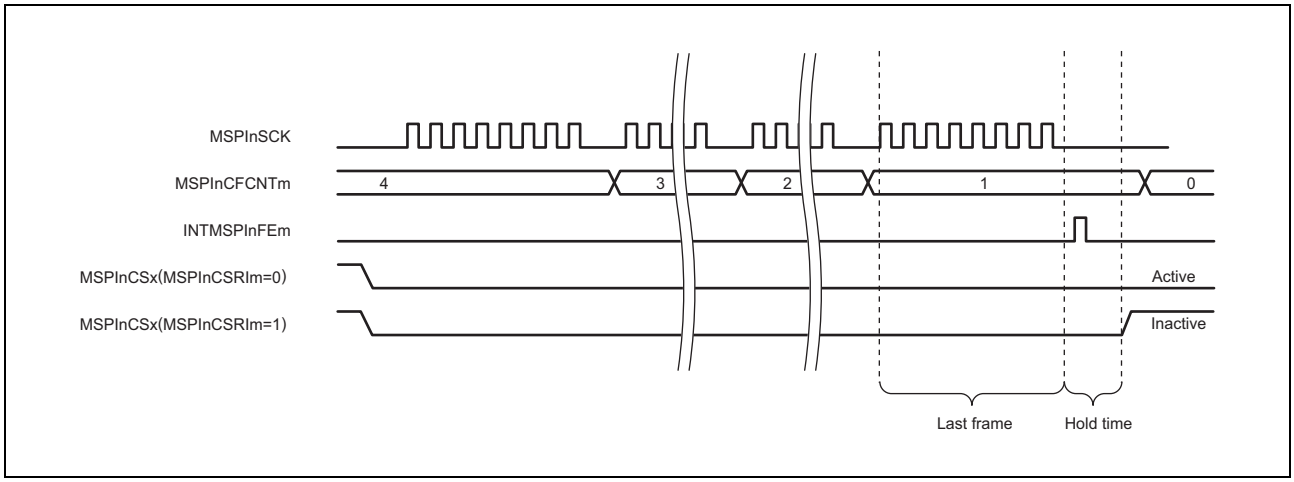
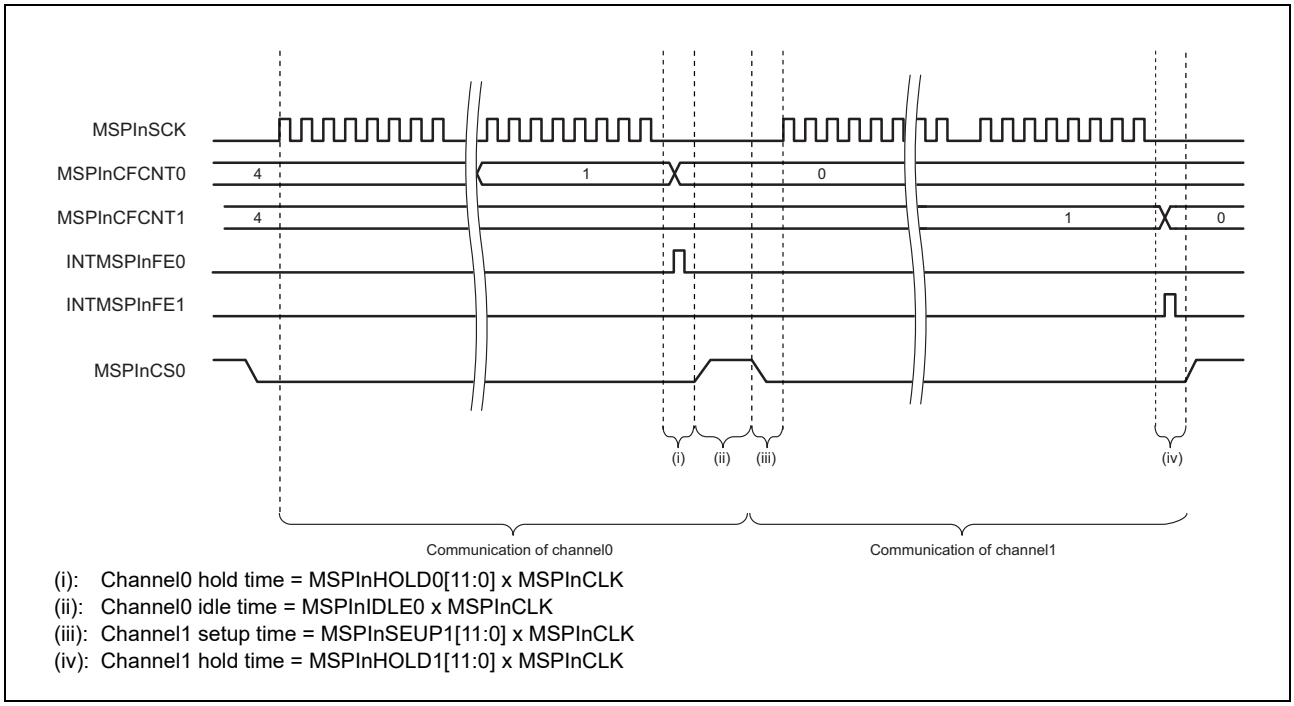


Figure 19.12 Level of Chip Select Signal After Last Frame End

NOTE

When MSPInFCCEm is 1 in the direct memory mode, even if MSPInCSRIm is 1, the MSPInCS signal holds the active level after the last frame.

Even if communication of other channels continues while the same CS signal active settings, the CS signal becomes the inactive level when MSPInCSRIm is set to 1.



(i): Channel0 hold time = MSPInHOLD0[11:0] x MSPInCLK
(ii): Channel0 idle time = MSPInIDLE0 x MSPInCLK
(iii): Channel1 setup time = MSPInSEUP1[11:0] x MSPInCLK
(iv): Channel1 hold time = MSPInHOLD1[11:0] x MSPInCLK

Figure 19.13 End of Last Frame and Continuous Communication of Other Channels (MSPInCSRIm=1)

19.6.3.5 CS Level Control of the Idle Time in the Direct Memory Mode

The level of the CS signal in idle time can be controlled with MSPInCFGm1.MSPInICLSm.

When the MSPInICLSm is set to 1, the CS signal in the idle time keeps the active level.

This function can be used only in the transmission or transmission/reception setting of Direct memory mode.

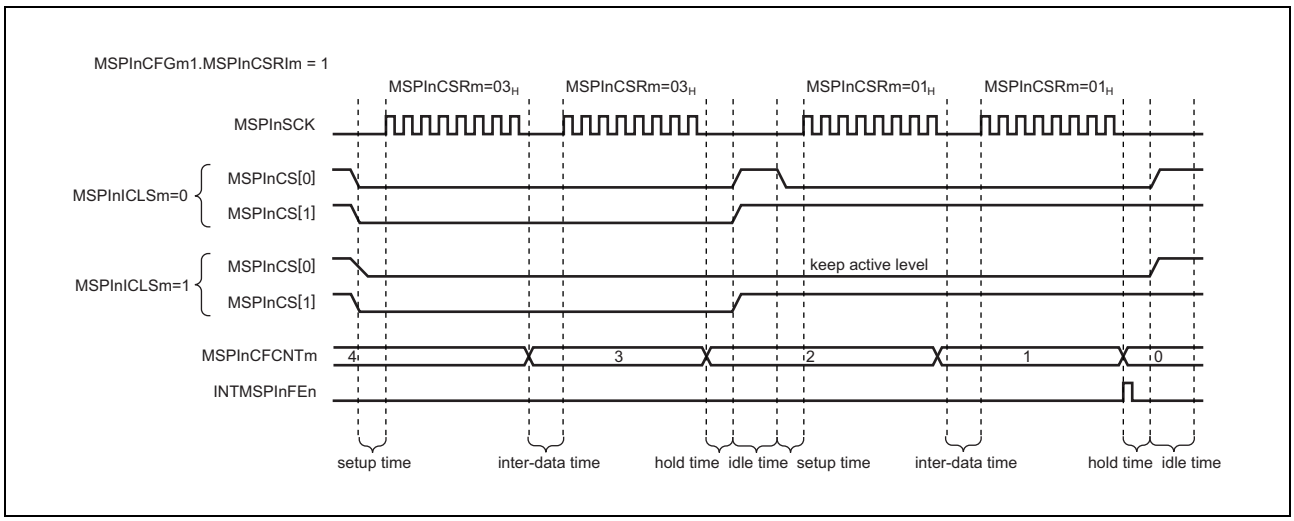


Figure 19.14 CS Level Control of MSPInICLSm (MSPInFIDLm = 0)

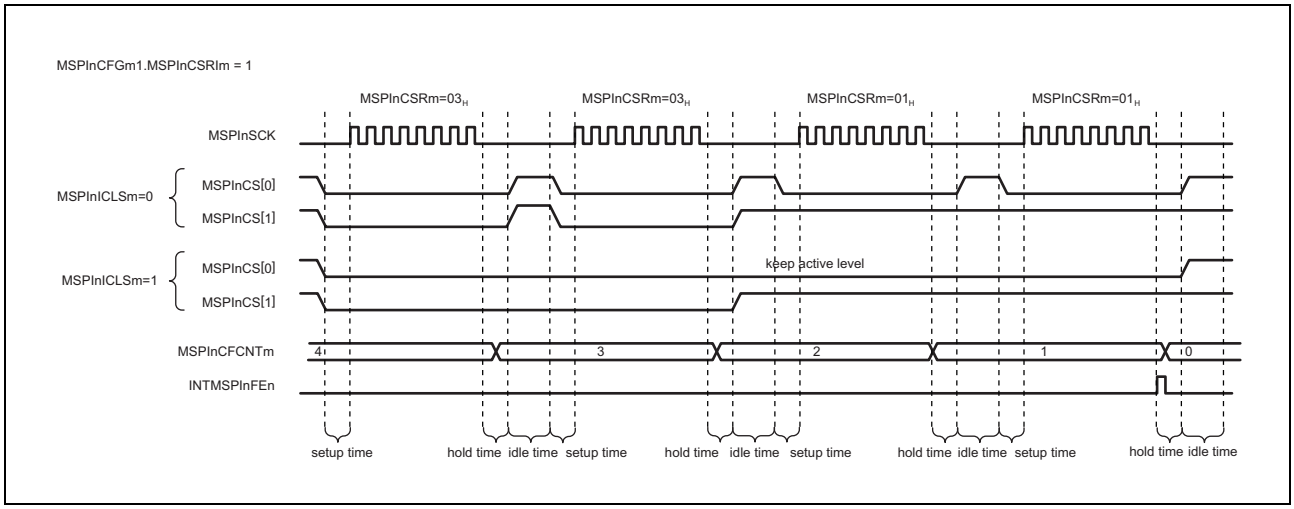


Figure 19.15 CS Level Control of MSPInICLSm (MSPInFIDLm = 1)

CAUTION

When MSPInICLSm = 1, MSPInCSRIm must be set to 1.

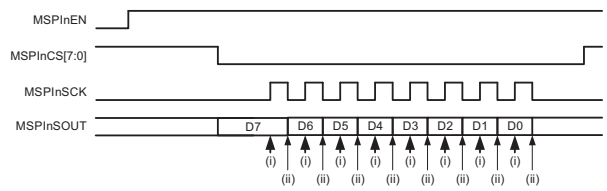
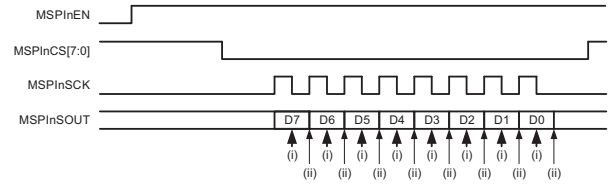
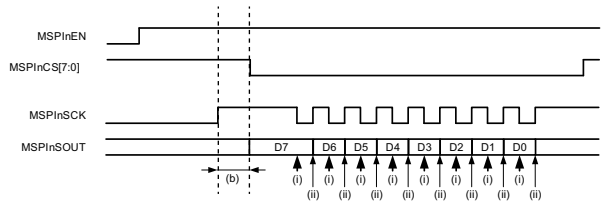
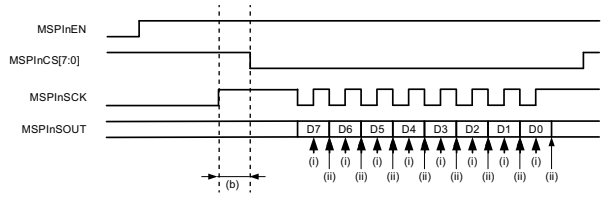
19.6.4 Clock Polarity and Data Phase

19.6.4.1 Clock default level, Clock polarity, Data phase and sampling point in the master mode

In the master mode, the default level of MSPInSCK can be set by MSPInCTL1.MSPInCKR, and the clock phase and data phase can be set independently for each channel by MSPInCFGm1. MSPInCPOLm and MSPInCFGm1.MSPInCPHAm, respectively.

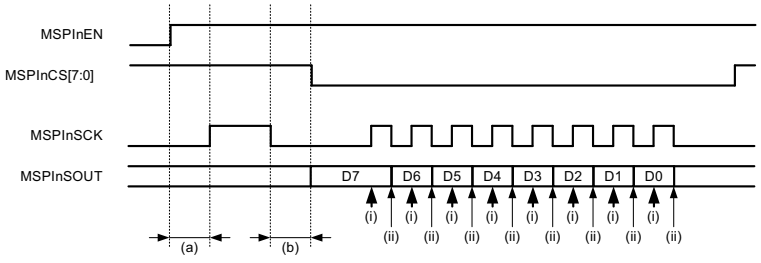
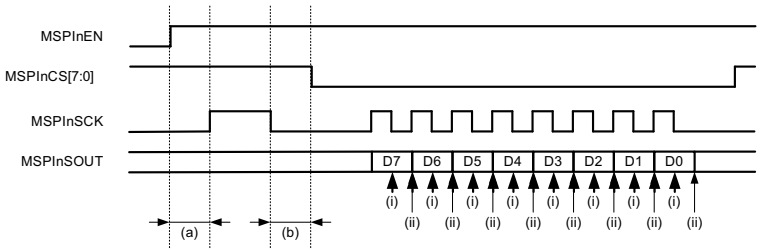
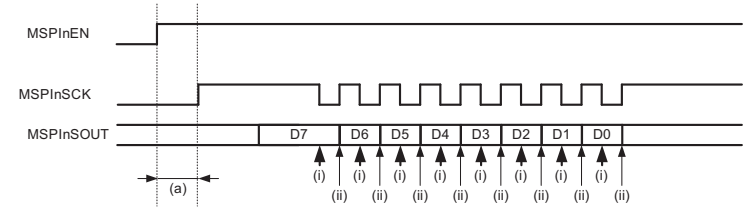
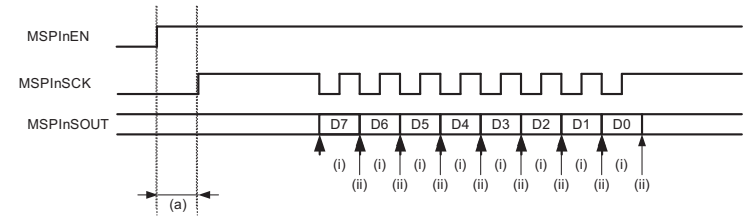
The sampling point can be set by MSPInCTL1.MSPInSAMP.

Table 19.55 Clock Polarity and Data Phase Selection in the Master Mode(MSPInCKR = 0)

MSPInCKR	MSPInCPOLm	MSPInCPHAm	
0	0	0	
0	0	1	
0	1	0	
0	1	1	

Note: (b) Polarity change by MSPInCPOLm:
When MSPInSCK switches polarity, MSPInCS activates after 2 clock of MSPInCLK
(i) Reception data sampling point for MSPInSAMP = 0
(ii) Reception data sampling point for MSPInSAMP = 1

Table 19.56 Clock Polarity and Data Phase Selection in the Master Mode(MSPInCKR = 1)

MSPInCKR	MSPInCPOLm	MSPInCPHAm	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

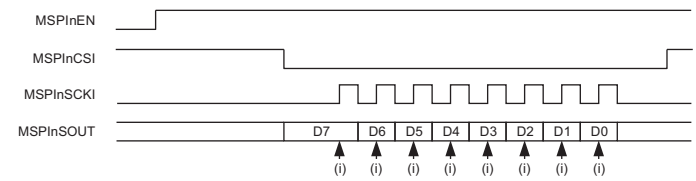
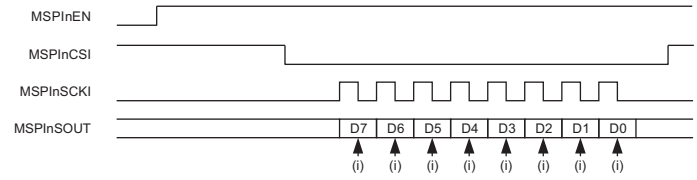
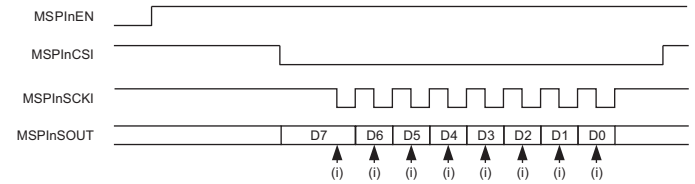
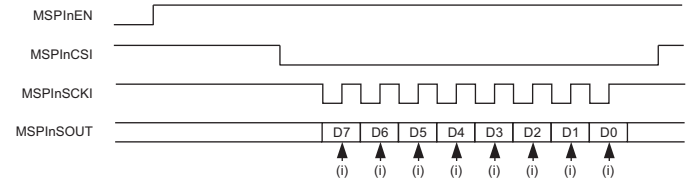
Note: (a) Polarity change by MSPInCKR: When MSPInEN is set to 1, MSPInSCK switches polarity after 2-3 clocks of MSPInCLK.
(b) Polarity change by MSPInCPOLm :When MSPInSCK switches polarity, MSPInCS activates after 3 clock of MSPInCLK.
(i) Reception data sampling point for MSPInSAMP = 0
(ii) Reception data sampling point for MSPInSAMP = 1

19.6.4.2 Clock Default Level and Data Phase in the Slave Mode

In the slave mode, the default level and clock phase of MSPInSCKI can be set by MSPInCTL1.MSPInCKR. And the data phase can be set by MSPInCFG01.MSPInCPHA0.

Set MSPInCFG01.MSPInCPOL0 to 0 and set MSPInCTL1.MSPInSAMP to 0 in the slave mode (MSPInCTL1.MSPInMSSEL = 1).

Table 19.57 Clock Polarity and Data Phase Selection in the Slave Mode

MSPInCKR	MSPInCPHA0	
0	0	
0	1	
1	0	
1	1	

Note: (i) Reception data sampling point in slave mode

19.6.4.3 Changing the Clock Phase

The polarity of the MSPInSCK can be set for each channel by the MSPInCFGm1.MSPInCPOLm.

When the idle time is set to 5 or less and two consecutive data packets are sent with different MSPInCFGm1.MSPInCPOLm configuration, the idle time is automatically extended to six period of MSPInCLK.

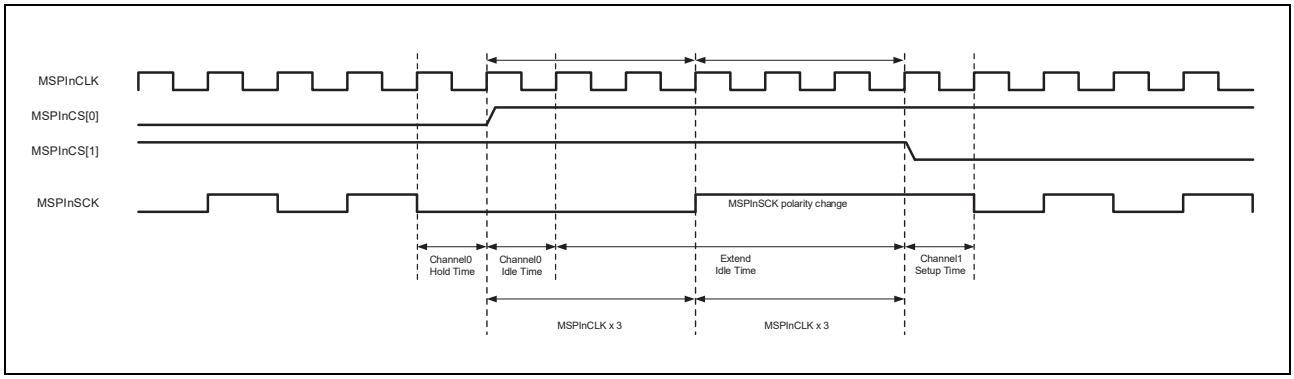


Figure 19.16 Change of the Clock Phase timing (idle time is set to 5 or less)

When the idle time is set to 6 or more and two consecutive data packets are sent with different MSPInCFGm1.MSPInCPOLm configuration, it changes the clock polarity of MSPInCLK three cycles before setup time.

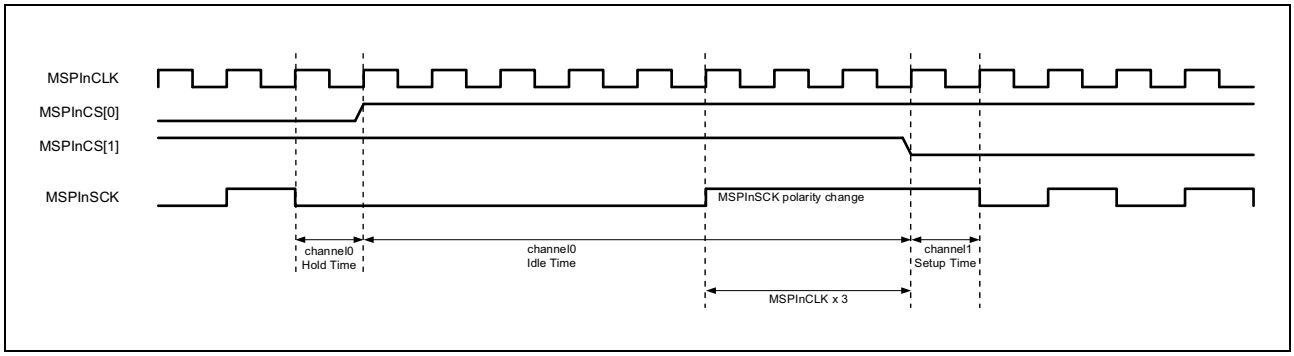


Figure 19.17 Change of the Clock Phase (idle time is set to 6 or more)

19.6.4.4 Clock Output During Communication Stop

When all channels communication ends, MSPI communication stops and MSPInCSx are inactive level (MSPInCSRIm=1), MSPInSCK returns the default level set by MSPInCTL1.MSPInCKR.

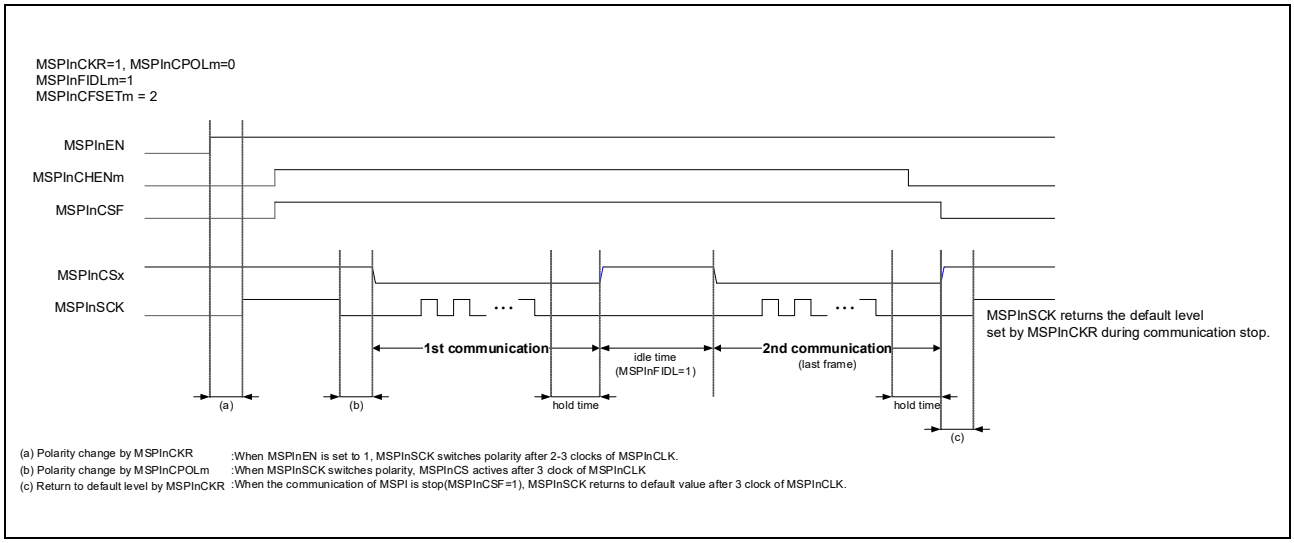


Figure 19.18 Clock Output During Communication Stop

19.6.5 SOUT Default Level and Idle Time Level

The default level and idle time level of MSPInSOUT can be set by MSPInCTL1.MSPInSOLS[1:0].

When MSPInEN is set to 1, MSPInSOUT becomes the inactive level set by MSPInSOLS[1].

The MSPInSOUT is held at the same level or becomes inactive level after communication is completed. This operation is set by the MSPInSOLS[0].

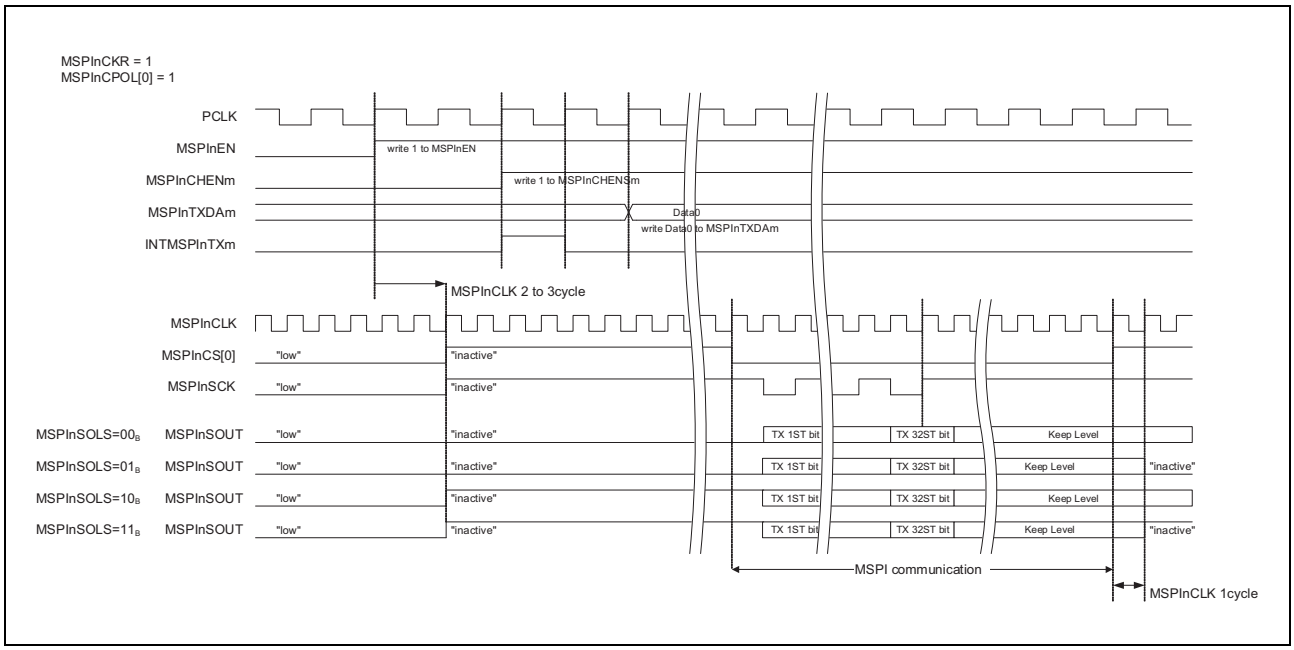


Figure 19.19 SOUT Operation

19.6.6 Serial Data Direction Selection

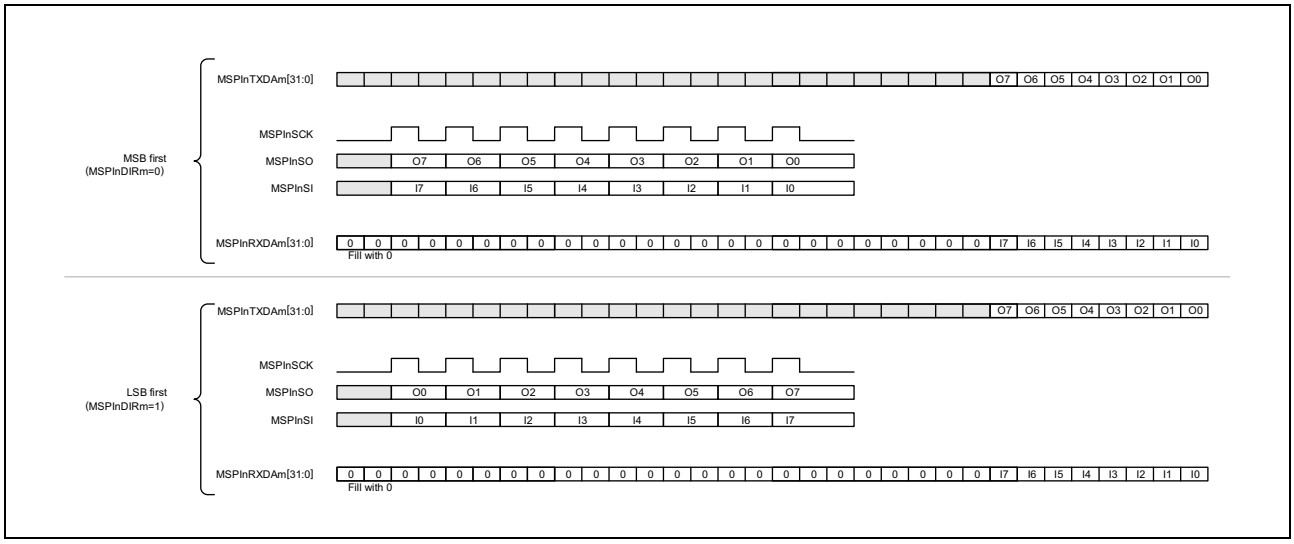


Figure 19.20 Serial Data Direction Select Function for 8-bit Communication

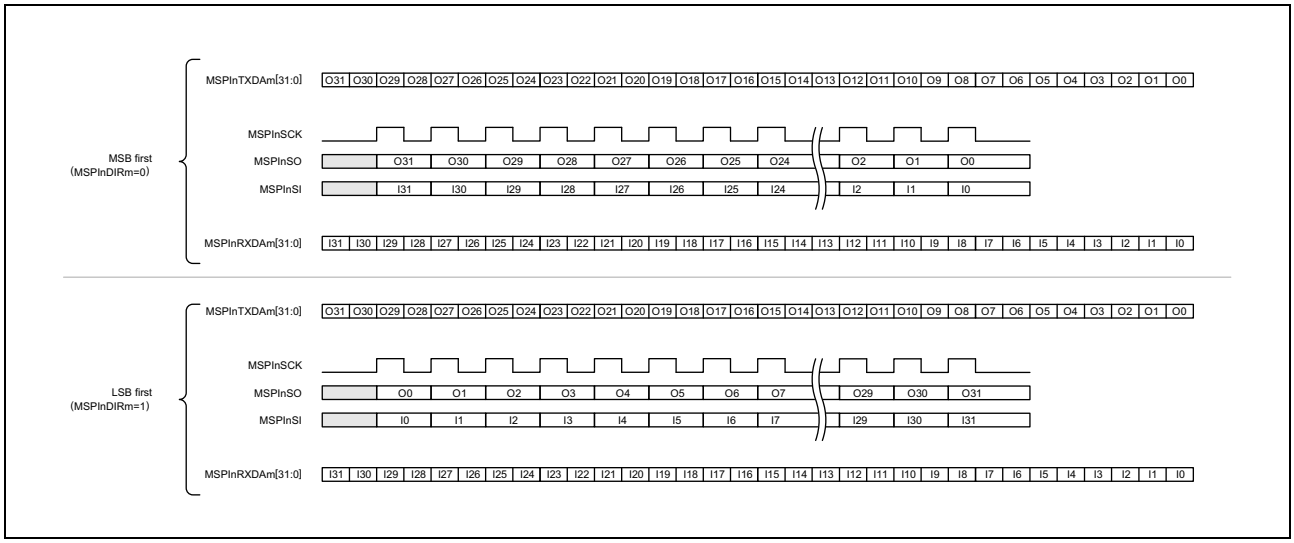


Figure 19.21 Serial Data Direction Select Function for 32-bit Communication

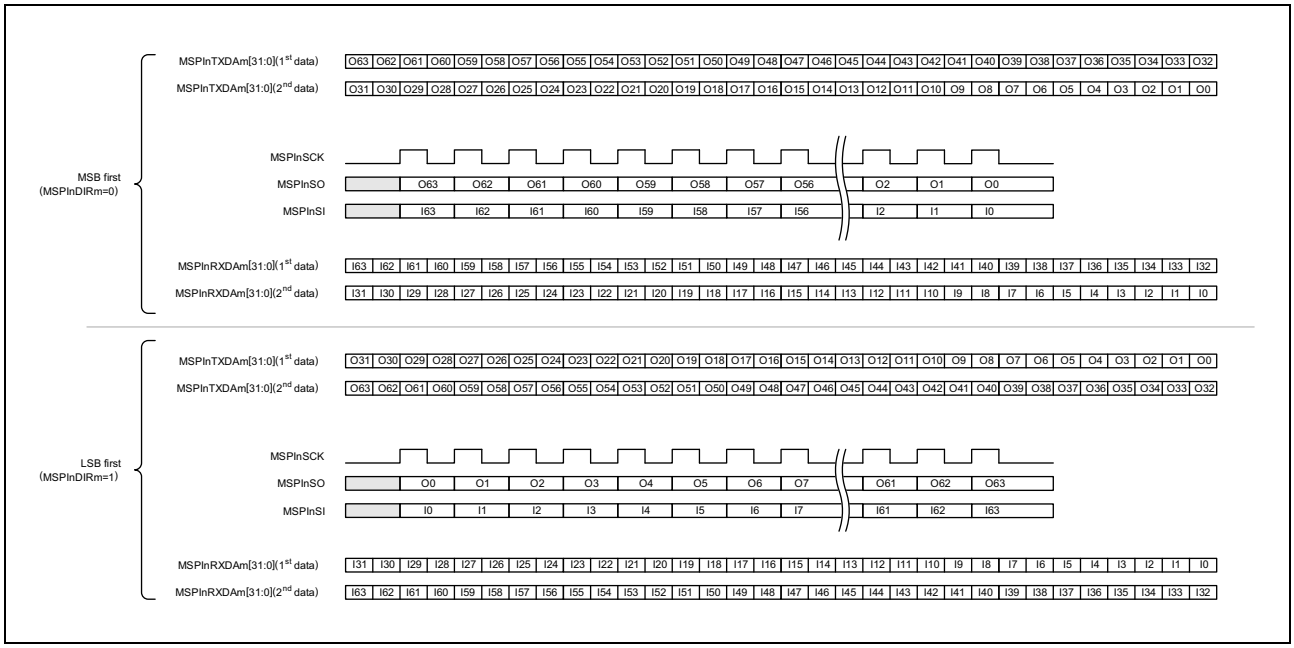


Figure 19.22 Serial Data Direction Select Function for the 64-bit Communication

NOTE

In the MSB communication of 33 bits or more, transmission data must be written from the upper address, and reception data has been read from the upper address.

19.6.7 Channel Priority Control

The priority level can be set independently for each channel by MSPInCFGm0.MSPInPRIOm[2:0]. The channel priority levels can be set from 0 to 7, with 0 the highest and 7 the lowest level.

If the priority settings of several channels are the same, the lowest interrupt channel number has priority.

MSPI has a shift register for communication and a reserve buffer for burst CSI/SPI communication.

The priority of the channel is judged at the timing when the reserve buffer can accept the request of the channel.

The following figure explains the timing of priority judgement when MSPI executes communication using channels 0, 1, 2, 3 in master mode/direct memory mode/transfer only.

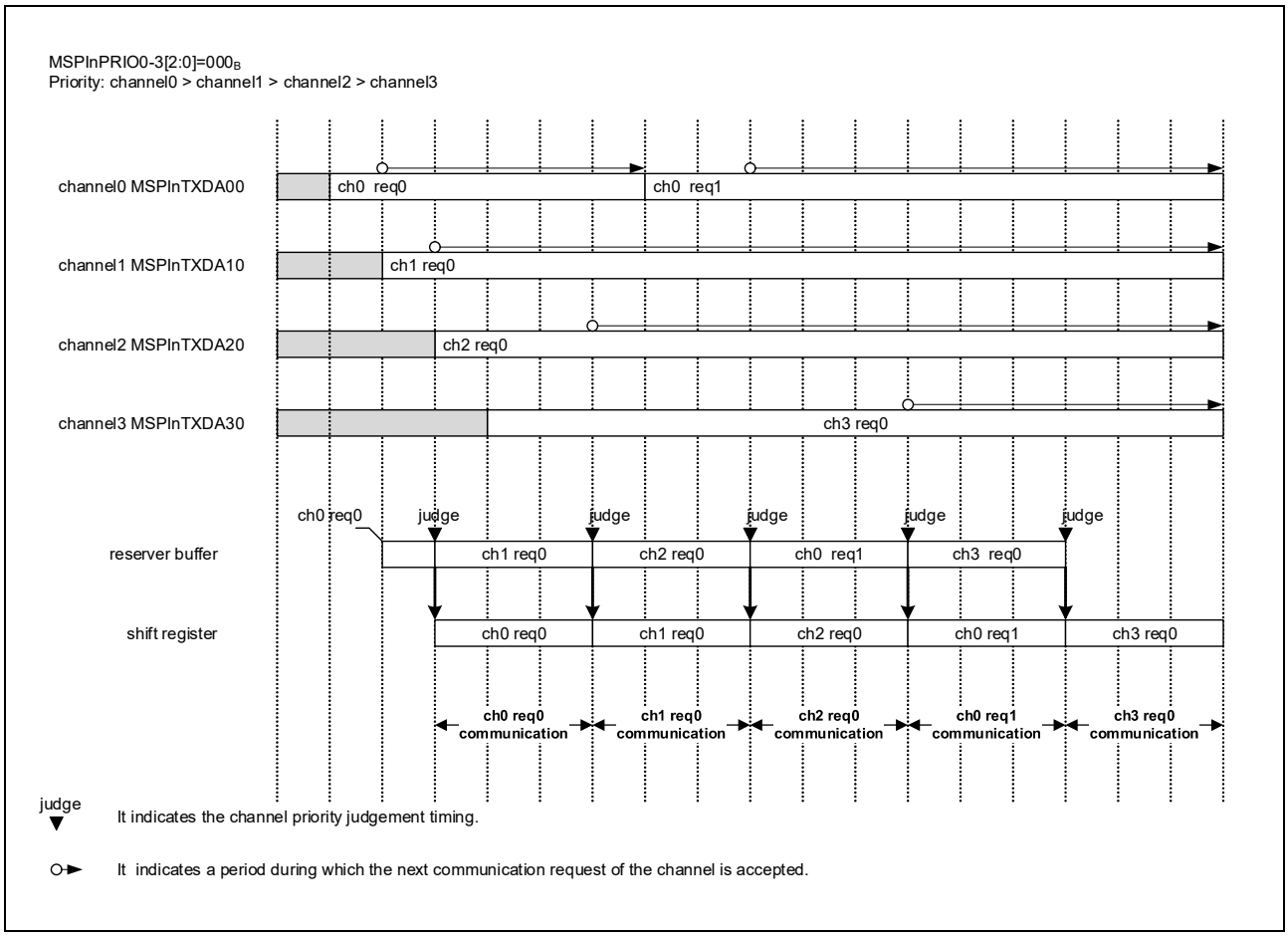


Figure 19.23 Timing Diagram of the Channel Priority Judgement

19.6.7.1 Channel lock function

The channel lock function can be used independently for each channel by MSPInCFGm0.MSPInLOCKm.

If the channel lock function is set, it is possible to reject interrupts of other channels from the acceptance of the first communication frame until the frame counter ends.

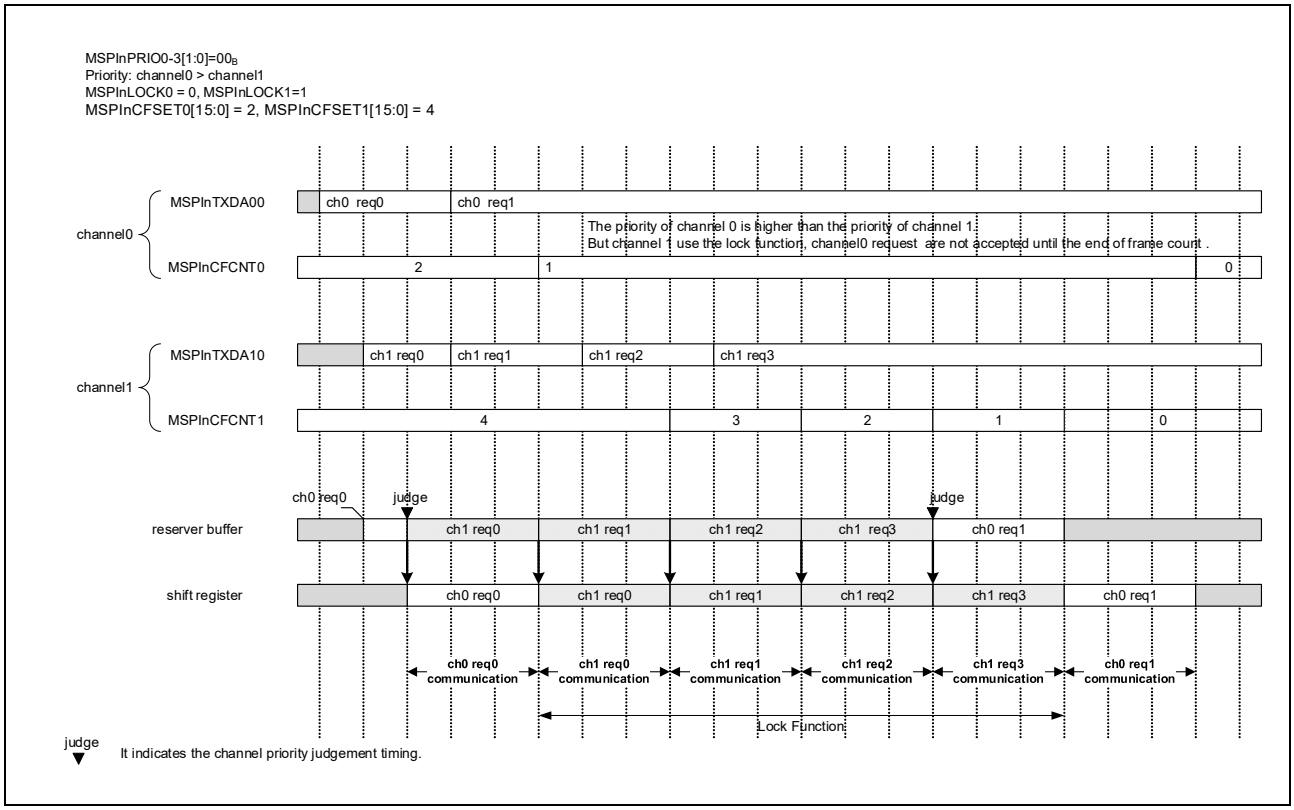


Figure 19.24 Timing Diagram of the Channel Priority Judgement in the Channel Lock Function

19.6.7.2 Job Enable Function in the Direct Memory Mode

The Job Enable function can be used independently for each channel by MSPInSSELm.MSPInJOBENm in the direct memory mode.

If MSPInJOBENm is set to 1, it is possible to reject interrupts from other channels until the MSPInJOBENm is cleared to 0.

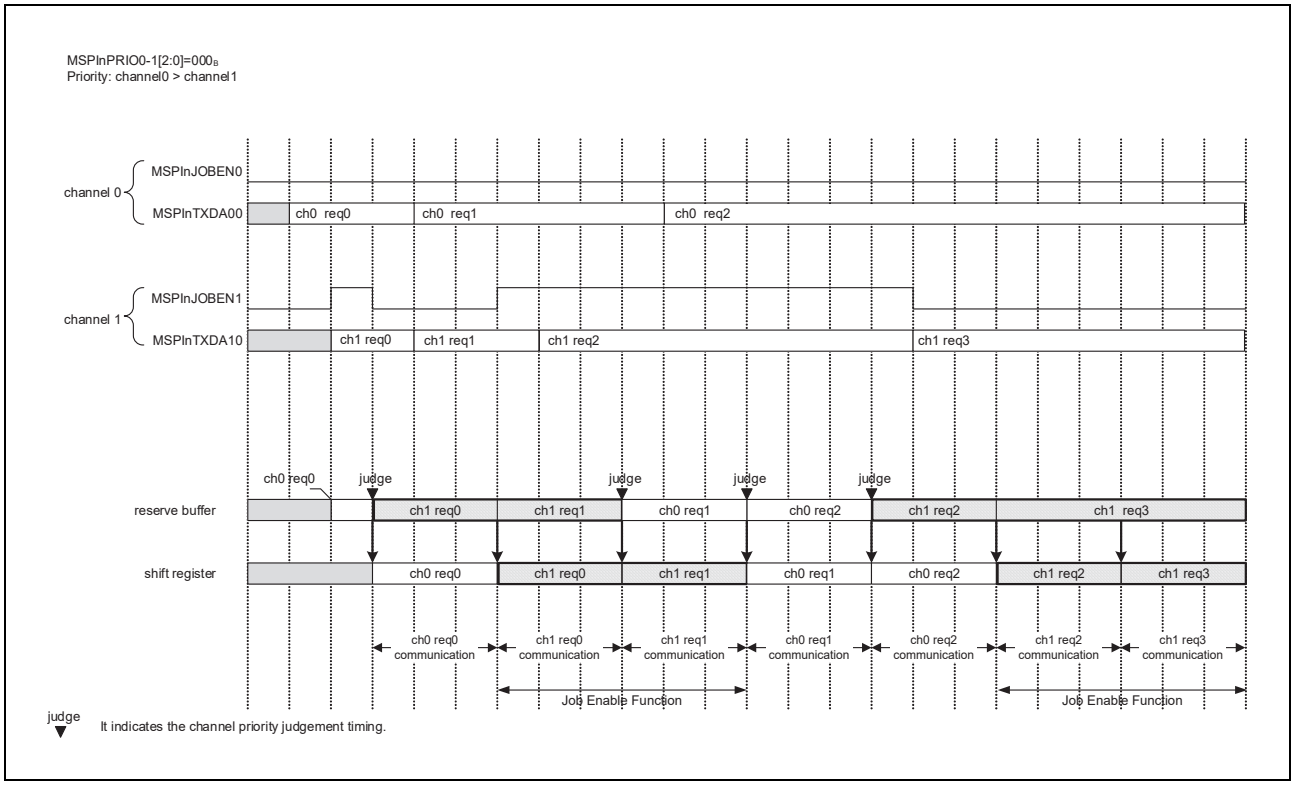


Figure 19.25 Timing Diagram of the Channel Priority Judgement in the Job Enable Function

NOTE

Even if MSPInJOBENm is set to 1 in the last frame data, MSPInJOBENm setting is ignored.

19.6.8 Error Detection

MSPI can detect 6 error types.

1. Parity error
2. CRC error
3. Data consistency error
4. Over-write error
5. Over-read error
6. Overrun error

Check for parity error and data consistency error can be enabled/disabled individually.

The table below shows the error types that can be detected in each mode for transmission/reception operation.

NOTE

The CRC check can use only in Safe-SPI protocol function.(see **Section 19.6.11**)

Table 19.58 Error Detection in Each Memory Mode for Transmission/Reception Operation

Error type	Direct Memory Mode			Fixed Buffer Memory Mode			Fixed FIFO Memory Mode		
	TX/RX*1	TX*2	RX*3	TX/RX*1	TX*2	RX*3	TX/RX*1	TX*2	RX*3
Parity	Detected	Not detected	Detected	Detected	Not detected	Detected	Detected	Not detected	Detected
CRC	Detected	Not detect	Detected	Detected	Not detected	Detected	Detected	Not detected	Detected
Data consistency	Detected	Detected	Not detected	Detected	Detected	Not detected	Detected	Detected	Not detected
Over-write	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected	Detected	Detected	Not detected
Over-read	Not detected	Not detected	Not detected	Not detected	Not detected	Not detected	Detected	Not detected	Detected
Overrun*4	Detected	Not detected	Detected	Not detected	Not detected	Not detected	Detected	Not detected	Detected

Note 1. TX/RX: Transmission/Reception(MSPInTXEm=1 and MSPInRXEm=1)
Note 2. TX: Transmission only (MSPInTXEm=1 and MSPInRXEm=0)
Note 3. RX: Reception only (MSPInTXEm=0 and MSPInRXEm=1)
Note 4. The Overrun error occurs in the slave mode. It does not occur in the master mode.

19.6.8.1 Parity check

The parity check is enabled by setting MSPInCFGm1.MSPInDECHKm to 01_B. The parity type (odd or even) is specified by MSPInCFGm1.MSPInPSm. When parity check is enabled, the 1 bit of the parity bit is added in the communication.

If reception is enabled (MSPInCFGm0.MSPInRXEm =1) and the parity check is enabled (MSPInCFGm1.MSPInDECHKm = 01_B), the parity bit is checked after the communication completion.

When a parity check error occurs, the MSPInPEm flag is set to 1 and INTMSPInERR becomes active.

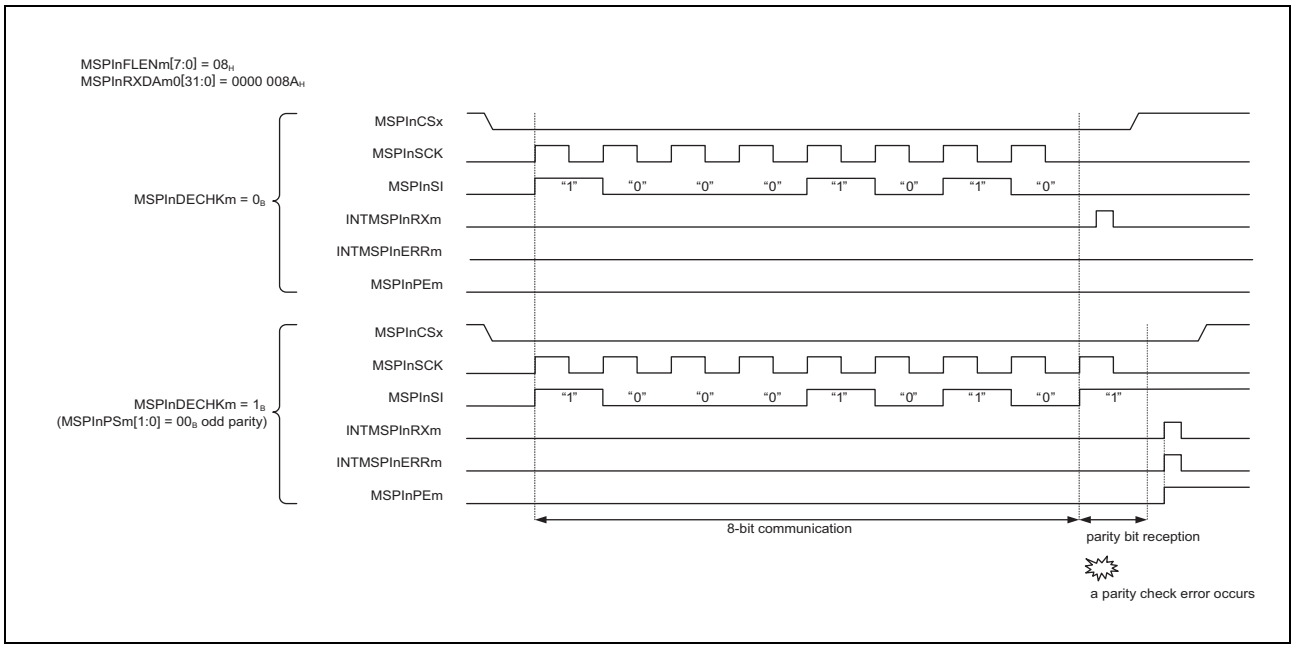


Figure 19.26 Timing Diagram of the Parity Check in the Reception

If transmission is enabled (MSPInCFGm0.MSPInTXEm = 1) and parity check is enabled(MSPInCFGm1.MSPInDECHKm = 01_B), the parity bit is added at the end of the communication frame.

When the transmission only operation is set (MSPInTXEm = 1, MSPInRXEm = 0), a parity check error never occurs.

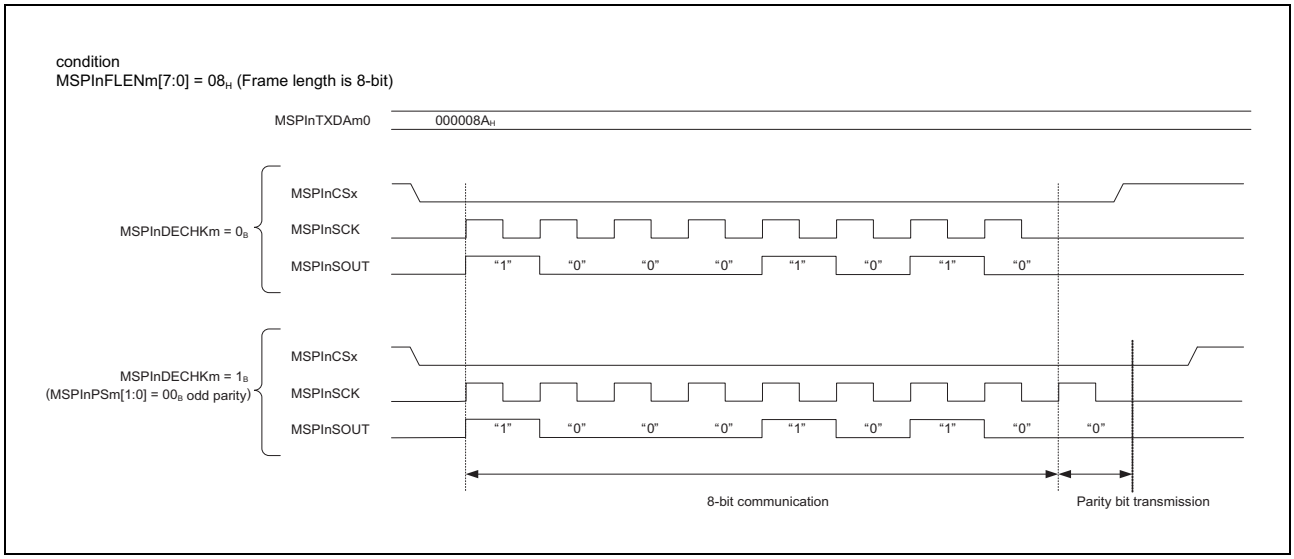


Figure 19.27 Timing Diagram of the Parity Check in the Transmission

19.6.8.2 Data Consistency Check

The data consistency check is enabled by setting MSPInCTL2.MSPInDCS to 1. It is not active if data transmission is disabled (MSPInCFGm0.MSPInTXEm = 0).

When the data consistency check is enabled, MSPIn compares the level of the transmission data (MSPInSOUT) and the physical level of the transmission pin (MSPInDCSI) at the reception sampling point.

When a data consistency check error occurs, the MSPInDCEm flag is set to 1 and INTMSPInERR becomes active at communication completion. The data consistency check error detection is common to all channels. When MSPInCTL2.MSPInDCS is set to 1, it is valid for all transmission setting channels.

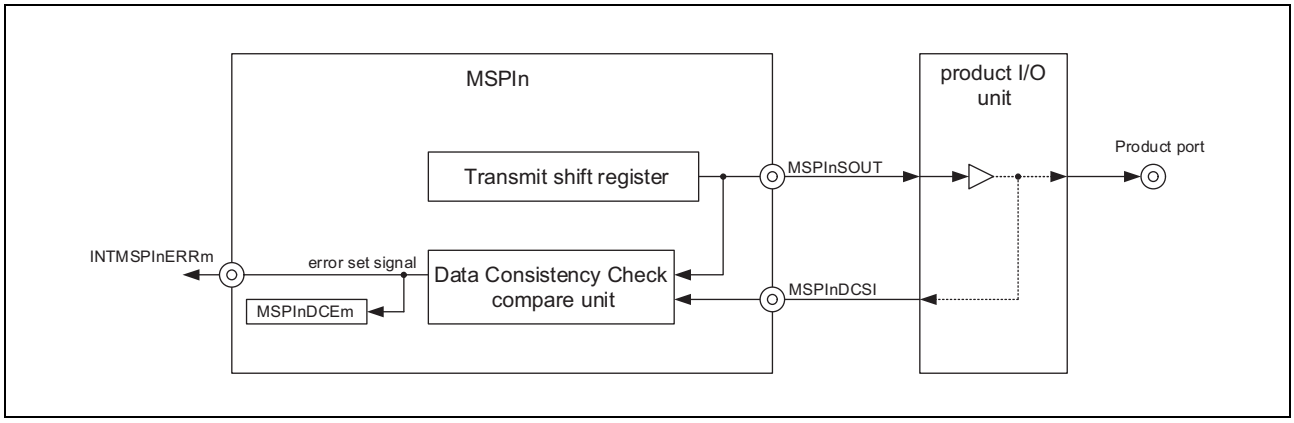


Figure 19.28 Functional Bock Diagram of the Data Consistency Check

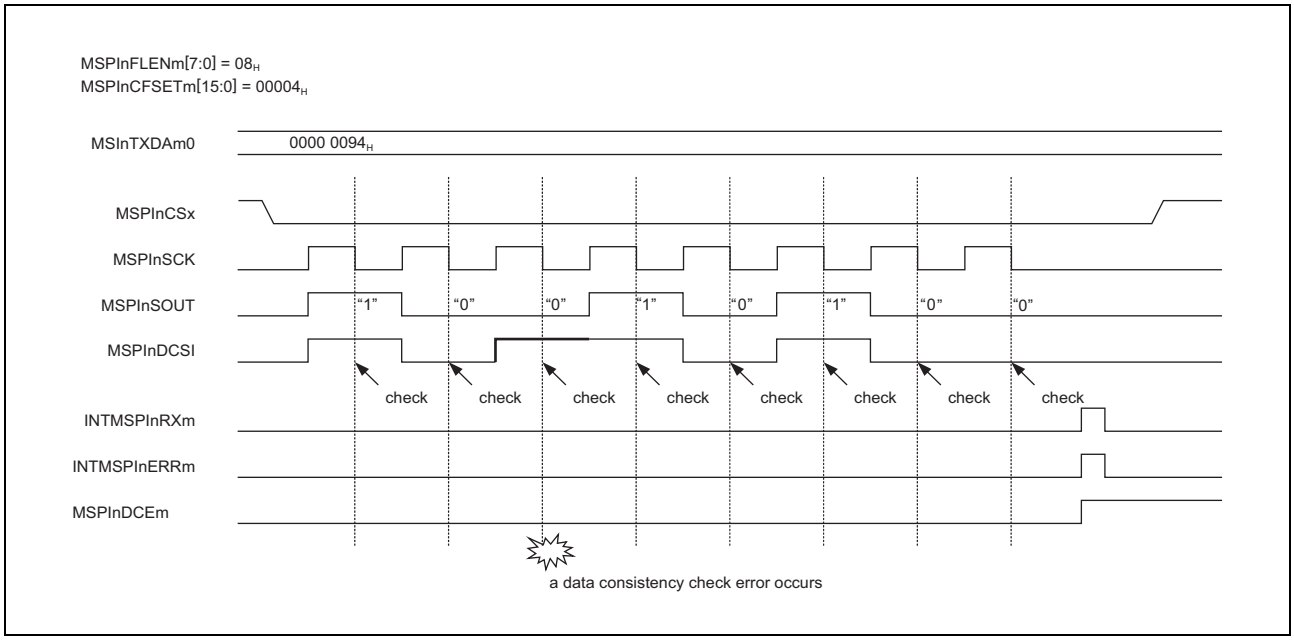


Figure 19.29 Timing Diagram of the Data Consistency Check

CAUTION

This function is not supported with LVDS mode.

19.6.8.3 Over-write/Over-read Error

An over-write and over-read errors may occur in the fixed FIFO memory mode and they never occur in the direct memory mode or fixed buffer memory mode.

An over-write error occurs when transmission data is written to the MSPInTXDAm0 register while the FIFO buffer is full.

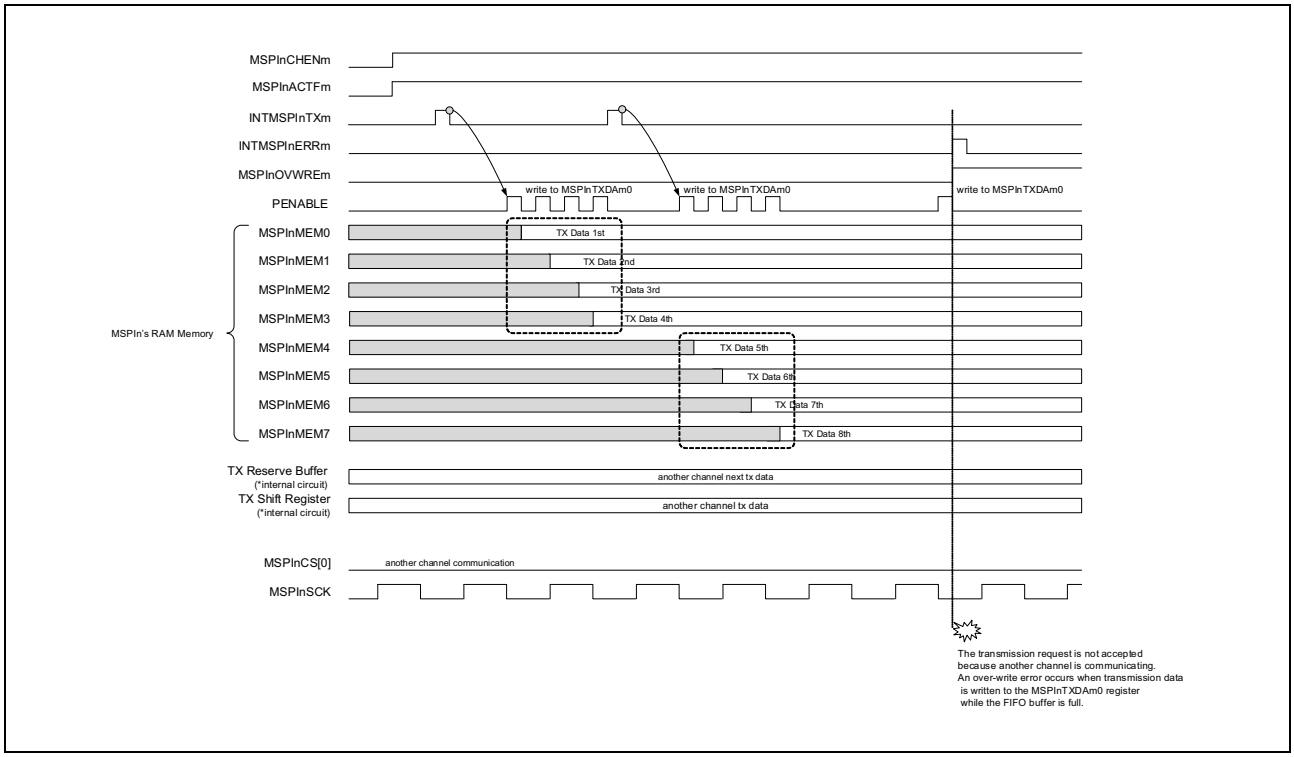


Figure 19.30 Timing Diagram of Over-write Error

An over-read error occurs when MSPInRXDAm0 register is read while the FIFO buffer is empty.

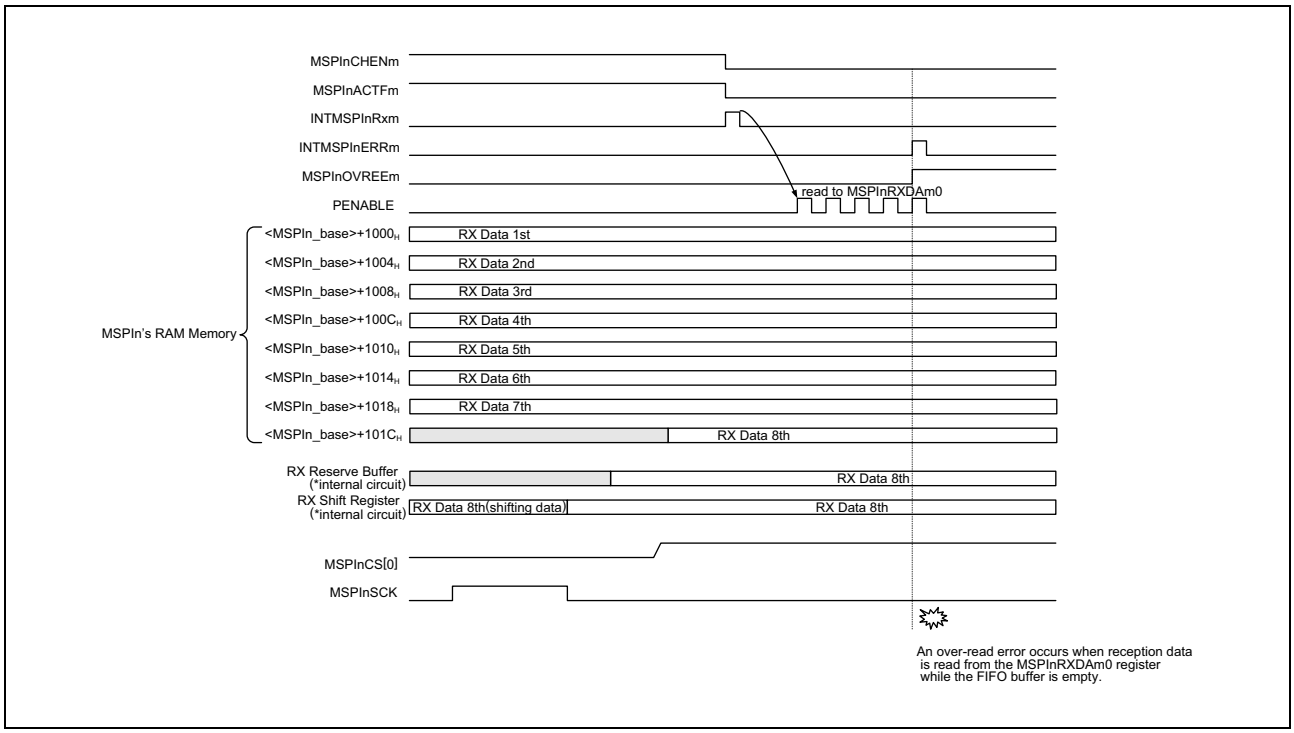


Figure 19.31 Timing Diagram of Over-read Error

19.6.8.4 Overrun Error Check

An overrun error can occur in the slave mode/direct memory mode and slave mode/fixed FIFO memory mode, and cannot occur in the fixed buffer memory mode.

The overrun error is not detected if data reception is disabled (MSPInCFGm0.MSPInRXEm = 0).

In the slave mode/direct memory mode and the slave mode/fixed FIFO memory mode, this error occurs when newly received data cannot be transferred from the shift register(internal reserve buffer) to the receive data register MSPInRXDAm0.

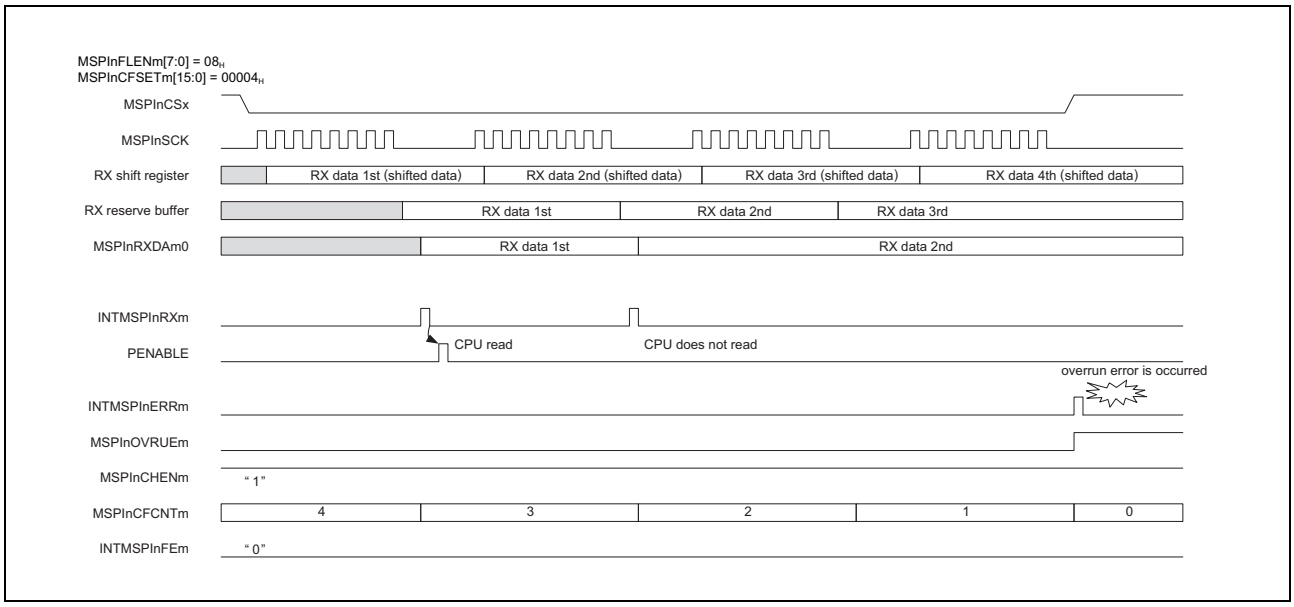


Figure 19.32 Timing Diagram of the Overrun Error

When an overrun error occurs, set MSPInEN = 0, and read received data as necessary.

If reading receive data, follow the procedure below.

Direct memory mode: Read MSPInRXDAm0 until MSPInRXRQFm = 0.

Fixed FIFO memory mode:Check the value of MSPInFIRXNm[7:0] and read the unread reception data.

NOTE

In the master mode, communication is stopped until the reception register MSPInRXDAm0 is read.

19.6.9 Communication Stop or IP Initialize

19.6.9.1 Communication Stop by the Clear Trigger

The channel operation can be stopped in synchronization with communication by using the channel enable clear trigger bit (MSPInCSTCm.MSPInCHENCm).

When the channel is stopped by the MSPInCSTCm.MSPInCHENCm, the communication held in the internal reserve-buffer is executed and then MSPInCHENm is cleared.

Therefore if channel is stopped by the MSPInCHENCm bit, when the current communication or the next communication stops, MSPInCHENm is cleared to 0.(If the channel is not communicating or communication of the channel is not held in the reserve buffer, MSPInCHENm is immediately cleared to 0.)

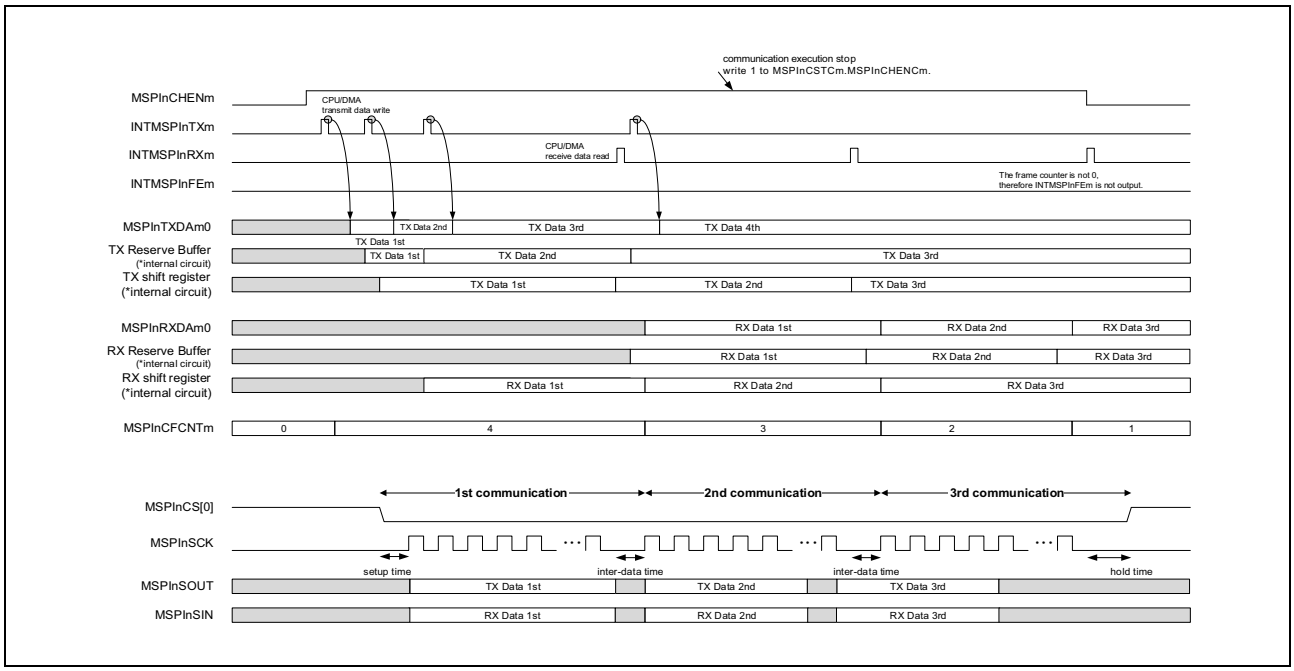


Figure 19.33 Communication Stop by the MSPInCHENCm

NOTE

If MSPInCFGm1.MSPInCSRIm is 1, MSPInCS becomes the inactive level, when the all channels are stopped.

If MSPInCFGm1.MSPInCSRIm is 0, MSPInCS keeps the active level, when the all channels are stopped.

If 1 is written to MSPInCHENCm in the Job Enable function, The communication is continued until the Job Enable function ends, and after receiving the last frame of the Job Enable function, MSPInCHENm is cleared to 0.

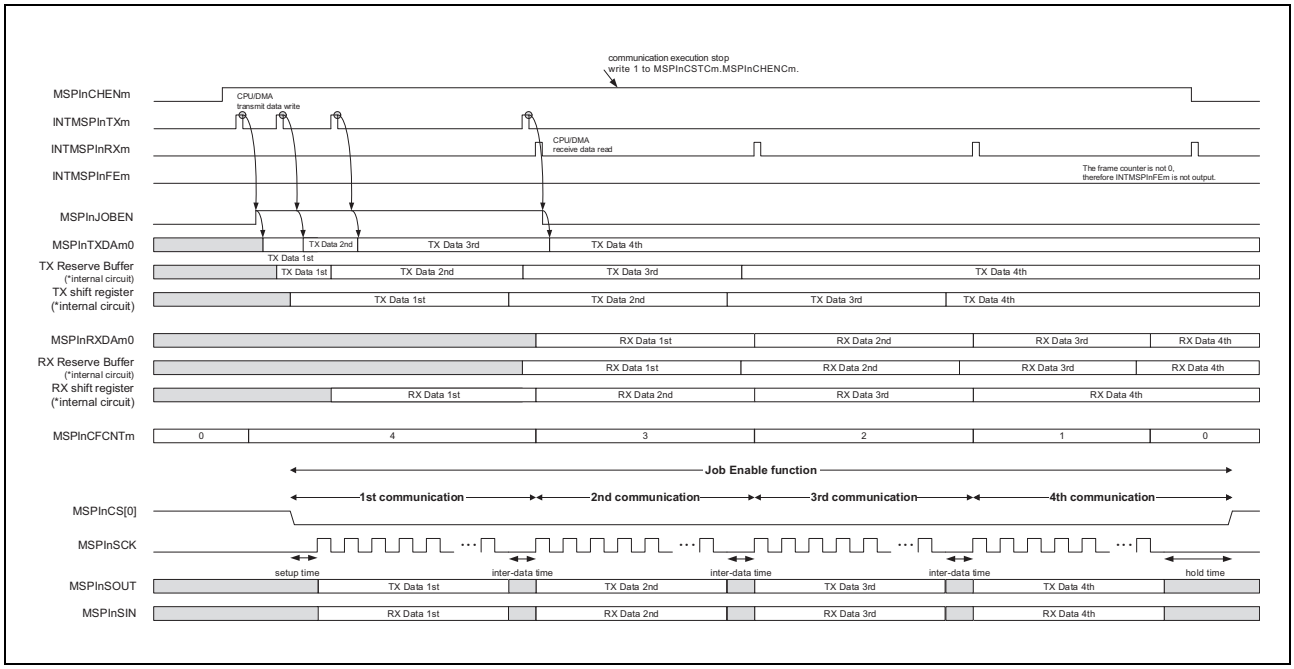


Figure 19.34 Communication Stop by the MSPInCHENCm in Job Enable Function of Direct Memory Mode

To stop communication by using MSPInCSTCm.MSPInCHENCm, follow the procedure below.

(1) Master mode

Direct Memory Mode:

1. Write MSPInCSTCm.MSPInCHENCm to 1.
2. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
   *The remaining frame transmission data and job transmission data must be written.
   If MSPInCSTRm.MSPInRXRQFm is set to 1(INTMSPInRX is output), read the received data.

Fixed buffer memory mode:

1. Write MSPInCSTCm.MSPInCHENCm to 1.
2. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
3. Read received data as necessary.

Fixed FIFO memory mode:

1. Check the value of MSPInCSTRm.MSPInRXRQFm flag and read received data until MSPInCSTRm.MSPInRXRQFm flag becomes 0.
2. Write MSPInCSTCm.MSPInCHENCm to 1.
   *Write MSPInCSTCm.MSPInCHENCm to 1 within one communication frame after Software executes "1".
3. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
   *The remaining frame transmission data must be written.

(2) Slave mode

Direct Memory Mode:

1. Write MSPInCSTCm.MSPInCHENCm to 1.
2. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
   *The remaining frame transmission data must be written.
   If MSPInCSTRm.MSPInRXRQFm is set to 1(INTMSPInRX is output), read the received data.
3. Write MSPInCTL0.MSPInEN to 0.(Initialize IP)

Fixed buffer memory mode:

1. Write MSPInCSTCm.MSPInCHENCm to 1.
2. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
3. Read received data as necessary.
4. Write MSPInCTL0.MSPInEN to 0.(Initialize IP)

Fixed FIFO memory mode:

1. Check the value of MSPInCSTRm.MSPInRXRQFm flag and read received data until MSPInCSTRm.MSPInRXRQFm flag becomes 0.
2. Write MSPInCSTCm.MSPInCHENCm to 1.
   *Write MSPInCSTCm.MSPInCHENCm to 1 within one communication frame after Software executes "1".
3. Wait until MSPInCSTRm.MSPInCHENm becomes 0.
4. Write MSPInCTL0.MSPInEN to 0.(Initialize IP)

Table 19.59 List of Bits Initialized when MSPInCHENm is Cleared from 1 to 0

Bit name	Initialization value
MSPInCSTRm.MSPInTXRQFm	0
MSPInCSTRm.MSPInACTFm	0

Table 19.60 List of Bits Initialized when MSPInCHENm is Set from 0 to 1

Bit name	Initialization value
MSPInCESTm.MSPInOVREEm	0
MSPInCESTm.MSPInOVWREm	0
MSPInCESTm.MSPInOVRUEm	0
MSPInCESTm.MSPInDCEm	0
MSPInCESTm.MSPInCEm	0
MSPInCESTm.MSPInPEm	0
MSPInFRERSTm.MSPInDCEFSTm	0
MSPInFRERSTm.MSPInCEFSTm	0
MSPInFRERSTm.MSPInPEFSTm	0

19.6.9.2 IP initialize by the IP enable

All channel operation can be stopped immediately by using the IP enable bit (MSPInCTL0.MSPInEN).

When MSPInEN is cleared to 0, the MSPInCSTRm.MSPInCHENm bit is cleared to 0, and the communication stops immediately.

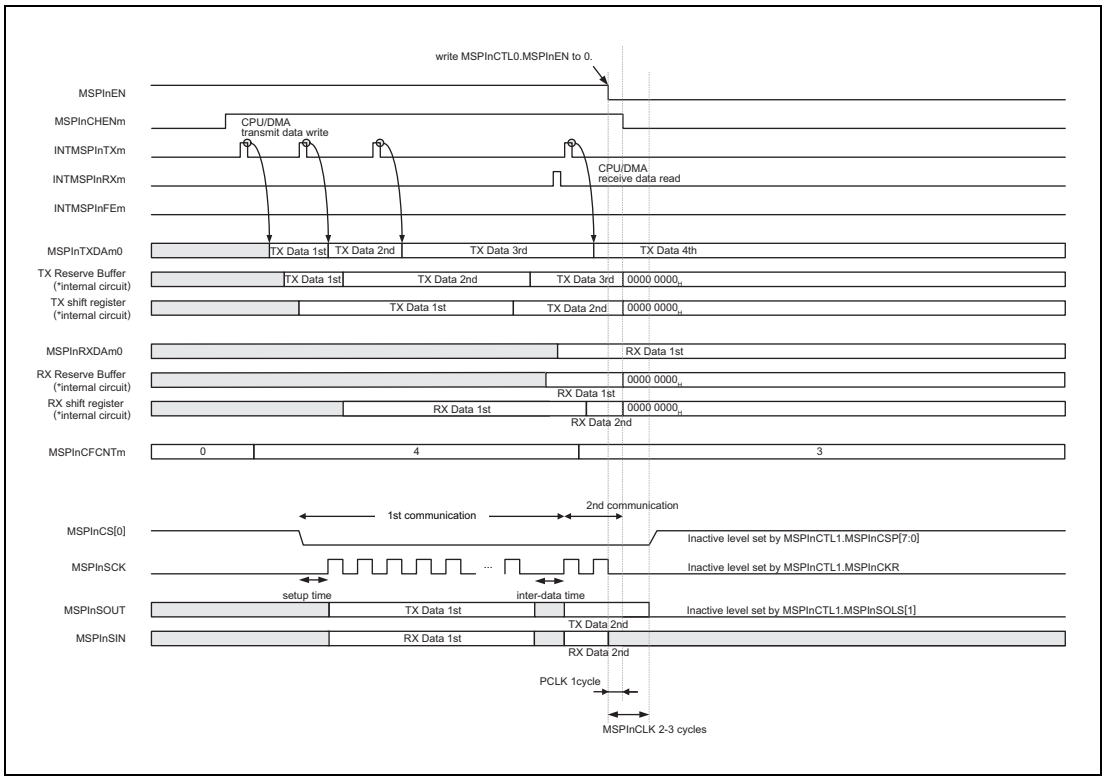


Figure 19.35 IP Initialize by the MSPInEN

NOTE

If MSPInCFGm1.MSPInCSRIm is 1, MSPInCS becomes the inactive level, when the channel is stopped.

Table 19.61 List of Ports or Bits Initialized when MSPInEN is Cleared from 1 to 0

Port name or bit name	Initialization value
MSPInCSx	Inactive level set by MSPInCTL1.MSPInCSP[7:0]
MSPInSCK	Inactive level set by MSPInCTL1.MSPInCKR
MSPInSOUT	Inactive level set by MSPInCTL1.MSPInSOLS[1]
MSPInSTR0.MSPInCNUMF[2:0]	00_H
MSPInSTR0.MSPInCSF	0
MSPInCSTRm.MSPInTXRQFm(all channel)	0
MSPInCSTRm.MSPInACTFm(all channel)	0
MSPInCSTRm.MSPInCHENm(all channel)	0

Table 19.62 List of Ports or Bits Initialized when MSPInEN is Set from 0 to 1

Port name or bit name	Initialization value
MSPInCSx	Inactive level set by MSPInCTL1.MSPInCSP[7:0]
MSPInSCK	Inactive level set by MSPInCTL1.MSPInCKR
MSPInSOUT	Inactive level set by MSPInCTL1.MSPInSOLS[1]
MSPInCESTm.MSPInOVREEm(all channel)	0
MSPInCESTm.MSPInOVWREm(all channel)	0
MSPInCESTm.MSPInOVRUEm(all channel)	0
MSPInCESTm.MSPInDCEm(all channel)	0
MSPInCESTm.MSPInCEm(all channel)	0
MSPInCESTm.MSPInPEm(all channel)	0
MSPInFRERSTm.MSPInDCEFSTm	0
MSPInFRERSTm.MSPInCEFSTm	0
MSPInFRERSTm.MSPInPEFSTm	0
MSPInCFCNTm(all channel)	0000_H

19.6.10 Loop-back mode

Loop-back mode is a special mode for self-test. This mode is only available in the master mode.

In Loop-back mode, MSPInSCK, MSPInCS[7:0] and MSPInSOUT are fixed to the inactive level. The transmit and receive signals are internally connected, as shown in the figures below.

In the self-test of the Loop-back mode, it performs transmission/reception and It checks whether the received data is the same as the transmitted data.

Table 19.63 Port Output Level in Loop-Back Mode

Port Name	Output level
MSPInSCK	Inactive(inactive level set by MSPInCKR)
MSPInCS[7:0]	Inactive(inactive level set by MSPInCSP[7:0])
MSPInSOUT	Inactive(inactive level set by MSPInSOLS[1])

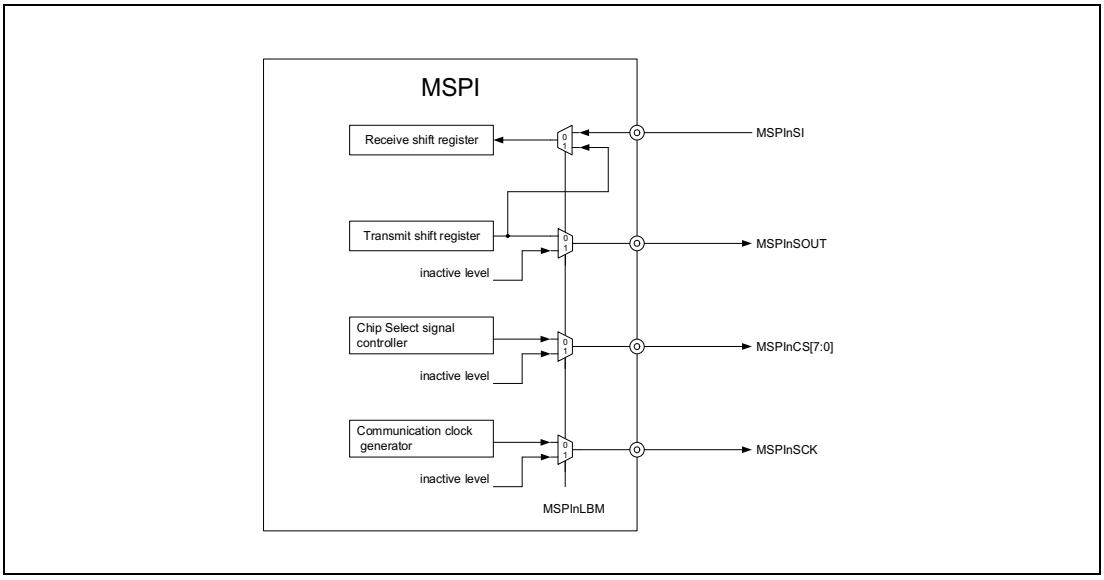


Figure 19.36 Transmission/reception in the Master-direct Memory Mode

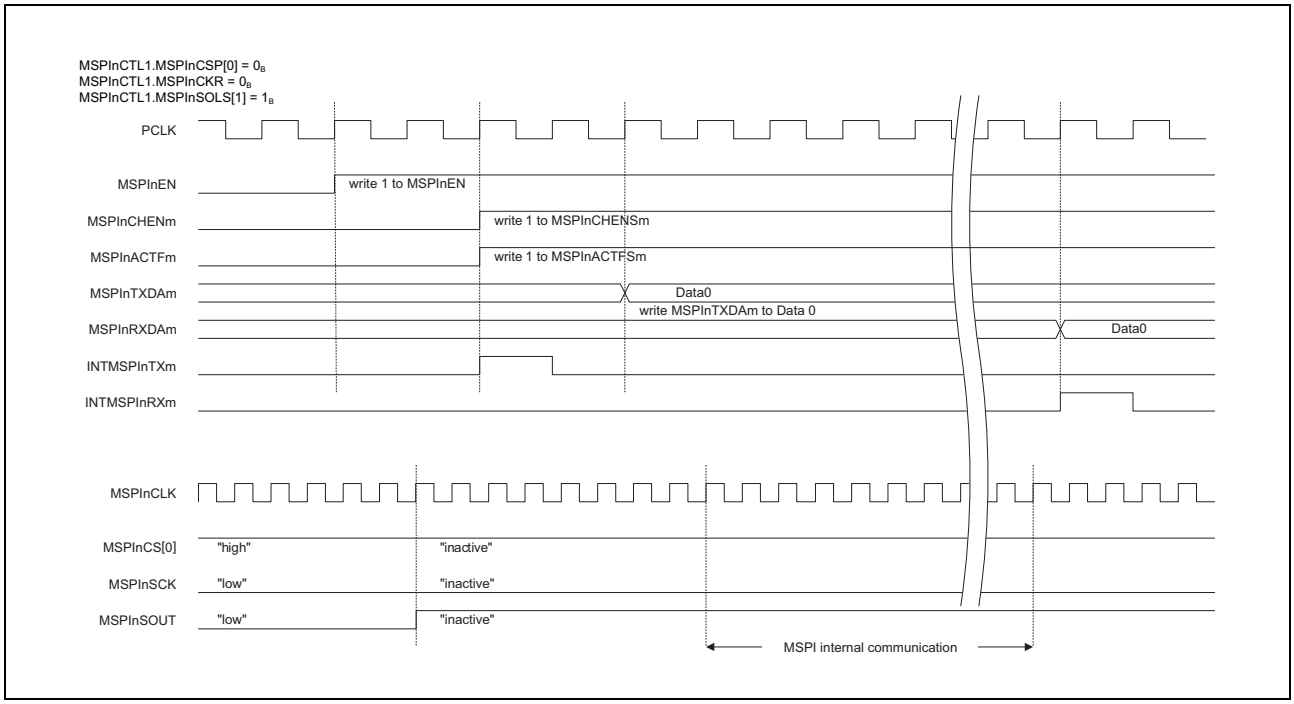


Figure 19.37 Communication of Loop-back Mode in Master Direct Memory Mode

19.6.11 Safe-SPI Protocol Function

In the master mode, MSPI can operates the Safe-SPI protocol function by setting MSPInCFGm1.MSPInSAFEm to 1.

Safe-SPI protocol has two frame formats, in-frame and out-of-frame.

When MSPInCFGm1.MSPInSAFS is 0, MSPI operates in in-frame format. When MSPInCFGm1.MSPInSAFS is 1, MSPI operates in out-of-frame format.

When using the Safe-SPI protocol function, The following configuration must be set.

- Transmission/Reception
- Frame length: 32 bits
- Communication direction : MSB
- The parity check is invalid

CAUTION

The CRC error always occurs when MSPI operates Loop-Back mode in Safe-SPI protocol function.

19.6.11.1 Safe-SPI protocol function in in-frame format

In transmission of in-frame format, MSPI calculates CRC using 31 to 5 bits of setting data, and transmits the CRC code in 4 to 2 bits of the communication transmission data.

In reception of in-frame MSPI internally calculates CRC using 26 to 3 bits of reception data, and compares bit 2 to 0 with internally calculated CRC.

If the comparison result is different, MSPI outputs an INTMSPInERR and sets the MSPInCEm flag.

CRC protection

For the in-frame format a 3 bit CRC with the polynomial 0x5 (x3 + x1 + x0) is used with a start value of 111_B and a target value of 000_B.

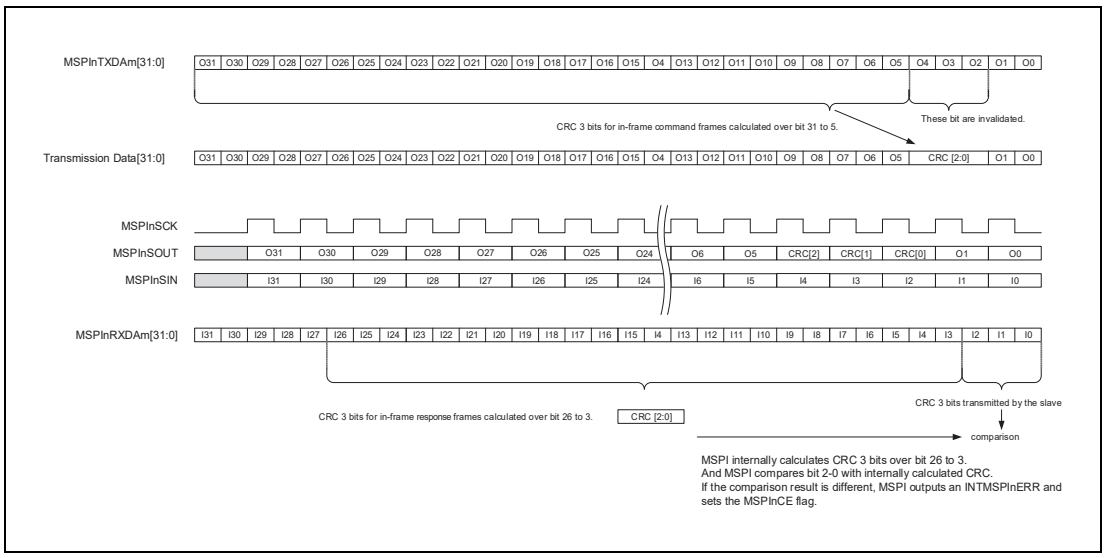


Figure 19.38 MSPI communication of Safe-SPI protocol in in-frame format

19.6.11.2 Safe-SPI protocol function in out-of-frame format

In transmission of out-of-frame format, MSPI calculates CRC using 31 to 3 bit of setting data, and transmits the CRC code in 2 to 0 bits of the communication transmission data.

In reception of out-of-frame MSPI internally calculates CRC using 31 to 3 bits of reception data, and compares bit 2 to 0 with internally calculated CRC.

If the comparison result is different, MSPI outputs an INTMSPInERR and sets the MSPInCE flag.

CRC protection

For the out-of-frame format a 3bit CRC with the polynomial 0x5 (x3 + x1 + x0) is used with a start value of 101_B and a target value of 000_B.

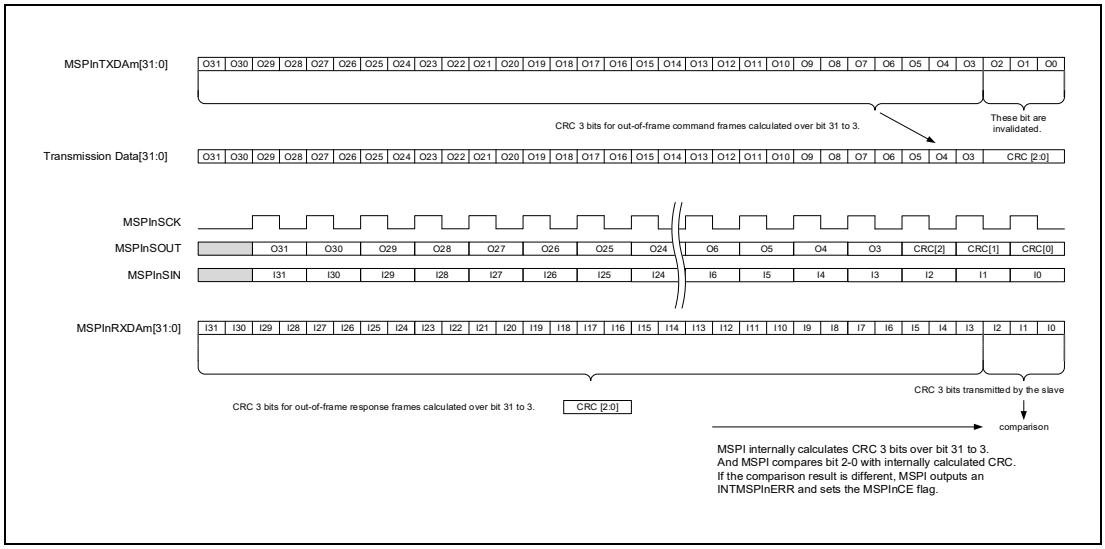


Figure 19.39 MSPI communication of Safe-SPI protocol in out-of-frame format

19.6.11.3 The CRC error mask function in Safe-SPI protocol function

In the out-of-frame format, the slave may not be able to send valid data and CRC on the first communication. Therefore, MSPI may detect a CRC error.

It is possible to mask the CRC error the first frame using the MSPInSAFCMm bit function.

When the MSPInSAFCMm =1, it does not check the CRC error of the first frame(MSPInCFSETm[15:0] = MSPInCFCNTm[15:0])

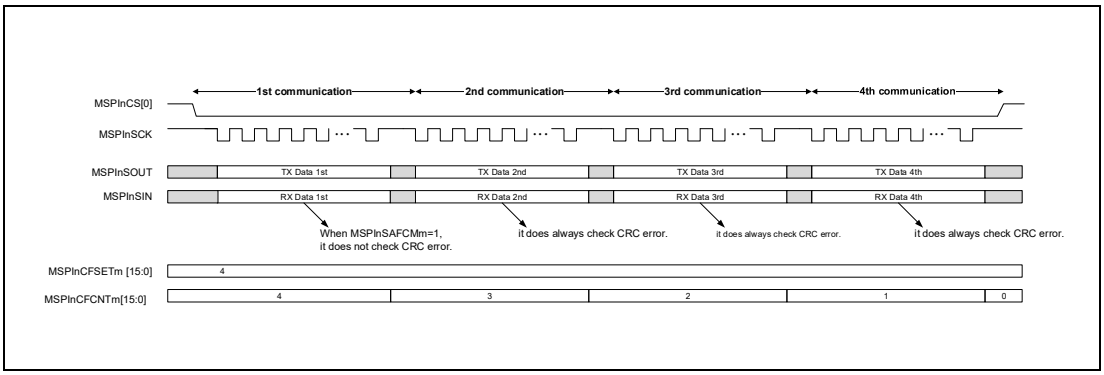


Figure 19.40 The MSPInSAFCMm bit function

NOTE

When the MSPInSAFCMm is set to 1, the CRC error of the first frame is masked in both in-frame format and out-of-frame format.

19.7 Memory Modes

The MSPI has a configurable RAM that can be used for buffered I/O.

The MSPI RAM is shared by each channel. Each channel has a register that sets the RAM area to be used.

The MSPI has three memory modes, and MSPI's RAM is used in two memory modes.

The memory modes can be set for each channels.

The MSPI RAM can be directly accessed by CPU/DMA.

19.7.1 Direct Memory Mode

In direct access memory mode, the channel does not use RAM, communication is using the MSPInTXDAm0 register for transmit data and the MSPInRXDAm0 register for receive data.

Interrupts are output for each transmission and reception, and communication is controlled by CPU/DMA.

19.7.1.1 Operation

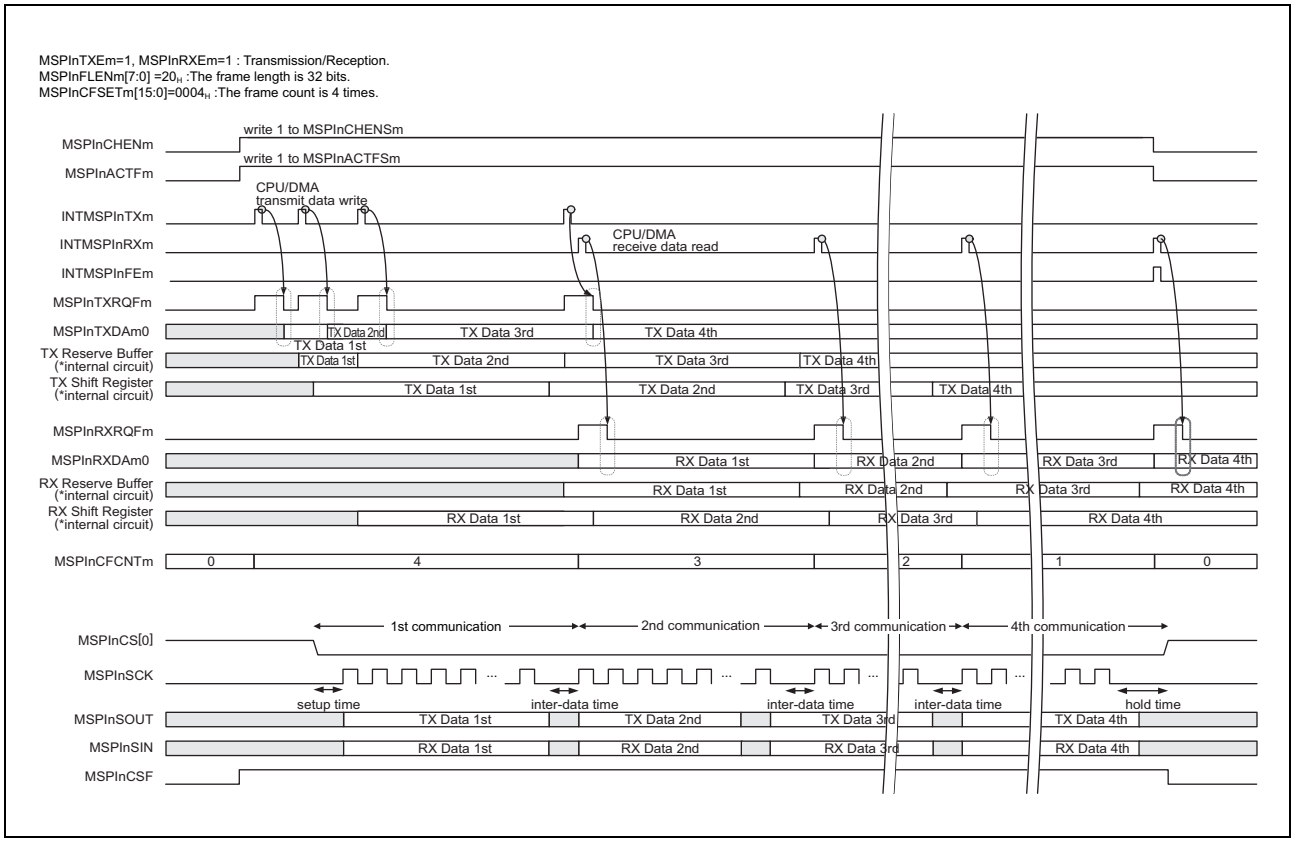


Figure 19.41 Transmission/Reception in the Master-Direct Memory Mode

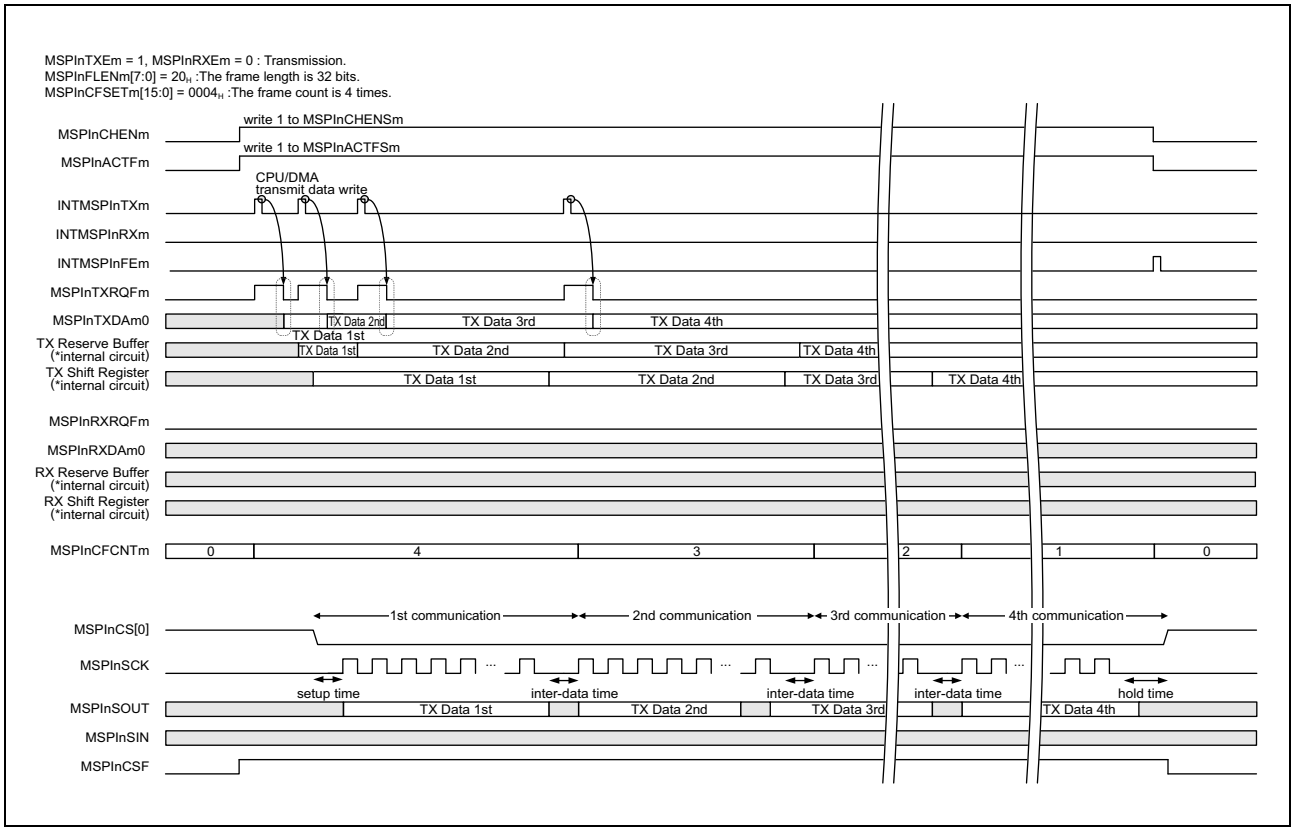


Figure 19.42 Transmission in the Master-Direct Memory Mode

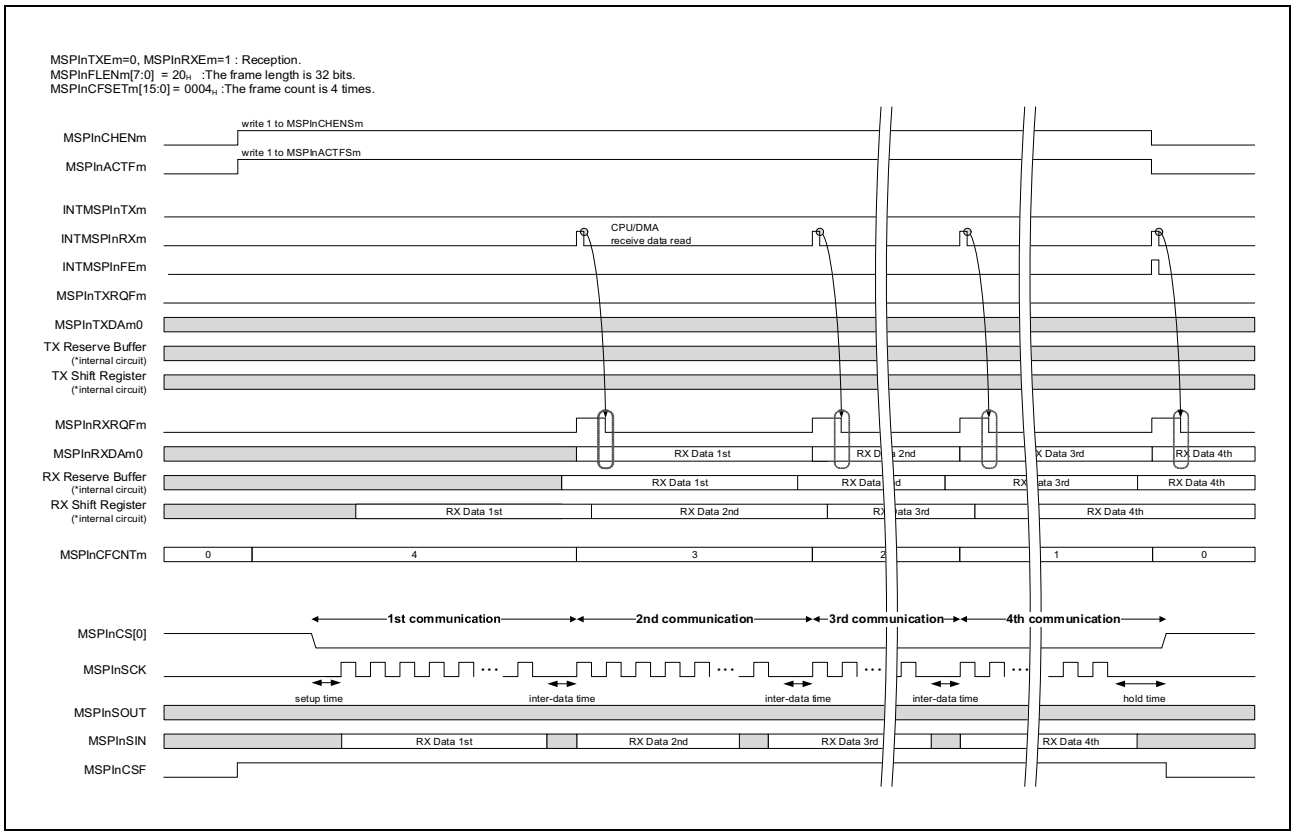


Figure 19.43 Reception in the Master-Direct Memory Mode

When it executes communication of 33 bits or more in the direct mode, an interrupt is output in units of 32 bits.

For each INTMSPInTX interrupt output, write MSPInTXDAm0 to transmission data. And for each INTMSPInRX interrupt output, read reception data of MSPInRXDAm0.

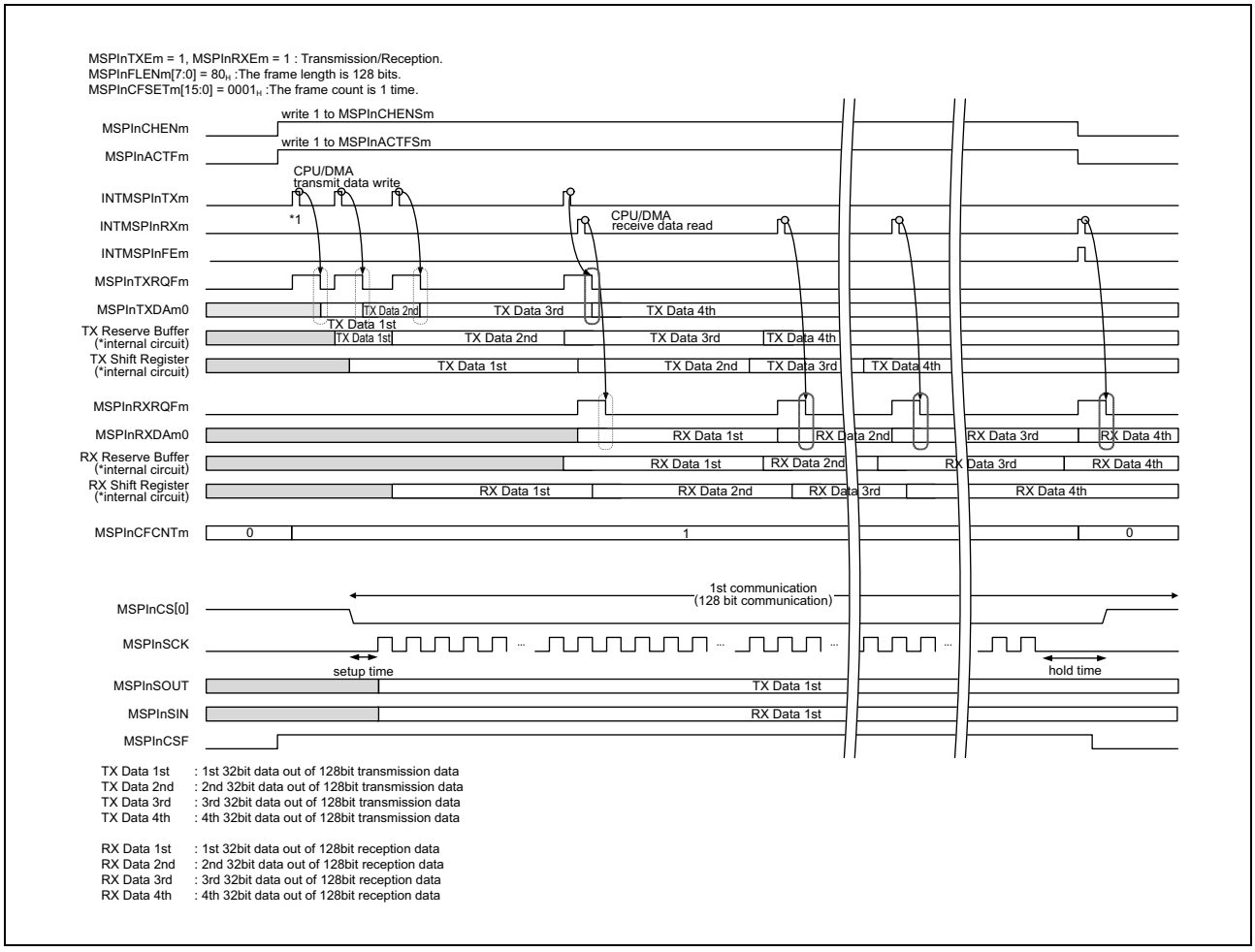


Figure 19.44 Transmission/Reception in the Master-Direct Memory Mode (The Frame Length is 128 bits)

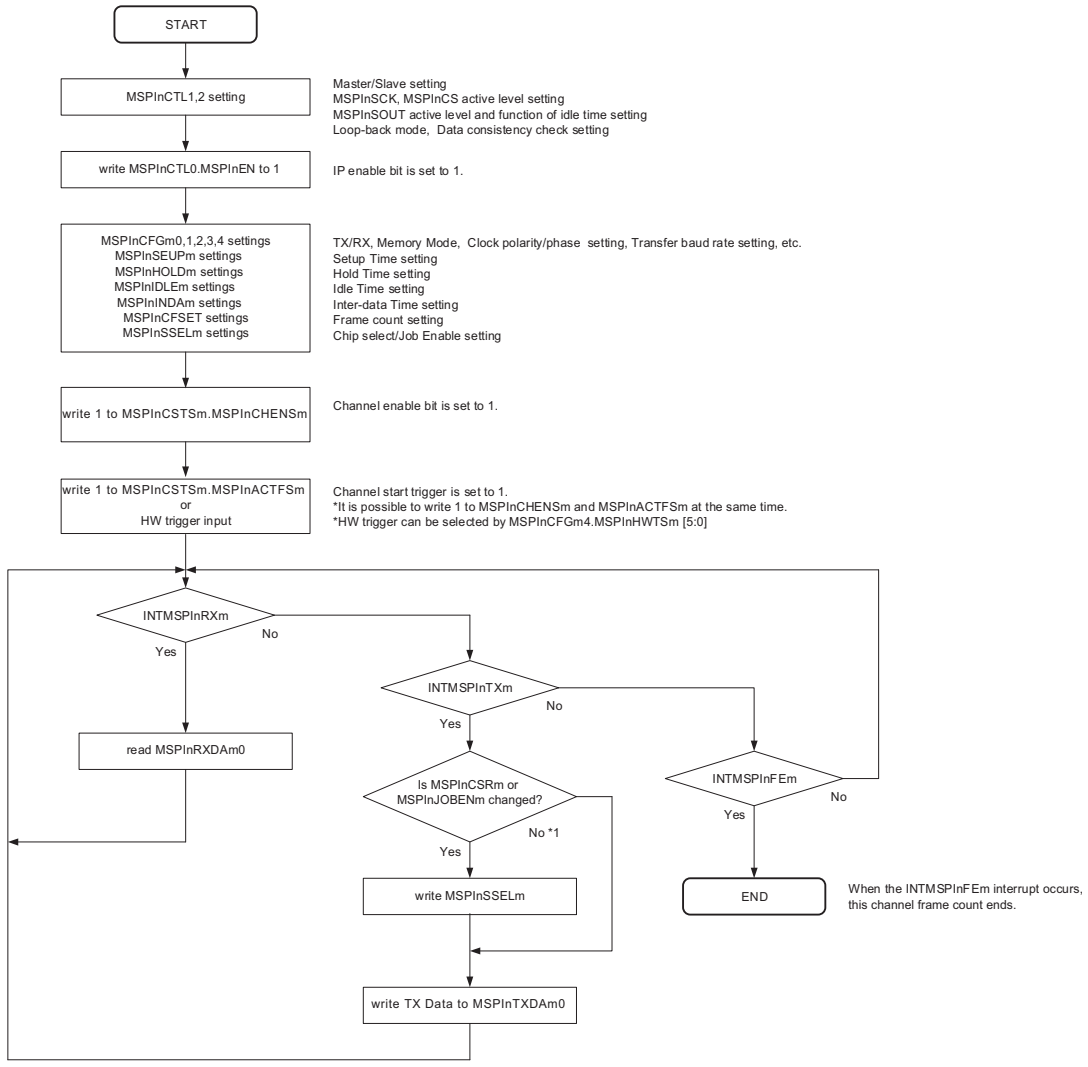
CAUTION

For the MSPInCS [7:0] and MSPInJOBENm settings, the setting value when the first data of the frame is written becomes effective (*1 in the Figure).

19.7.1.2 Operating Procedure

Follow the procedure of this chapter to operate in direct memory mode.

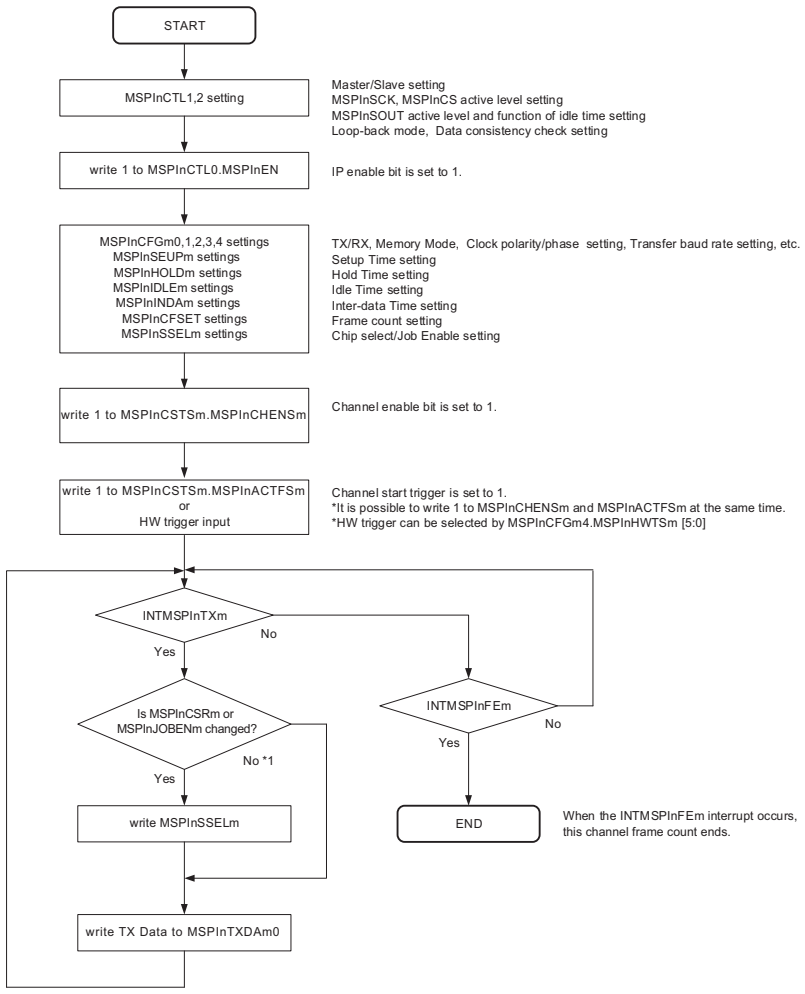
In the direct memory mode, interrupts are output at every communication frame, and communication data is controlled by CPU/DMA.



Note 1. If MSPInCSRm or MSPInJOBENm are not changed, the same value can be written in MSPInSSELm.

Note: Set MSPInJOBENm of the last frame to 0.

Figure 19.45 Master Transmission/Reception Operating Procedure in Direct Memory Mode



Note 1. If MSPInCSRm or MSPInJOBENm are not changed, the same value can be written in MSPInSSELm.

Note: Set MSPInJOBENm of the last frame to 0.

Figure 19.46 Master Transmission Operating Procedure in the Direct Memory Mode

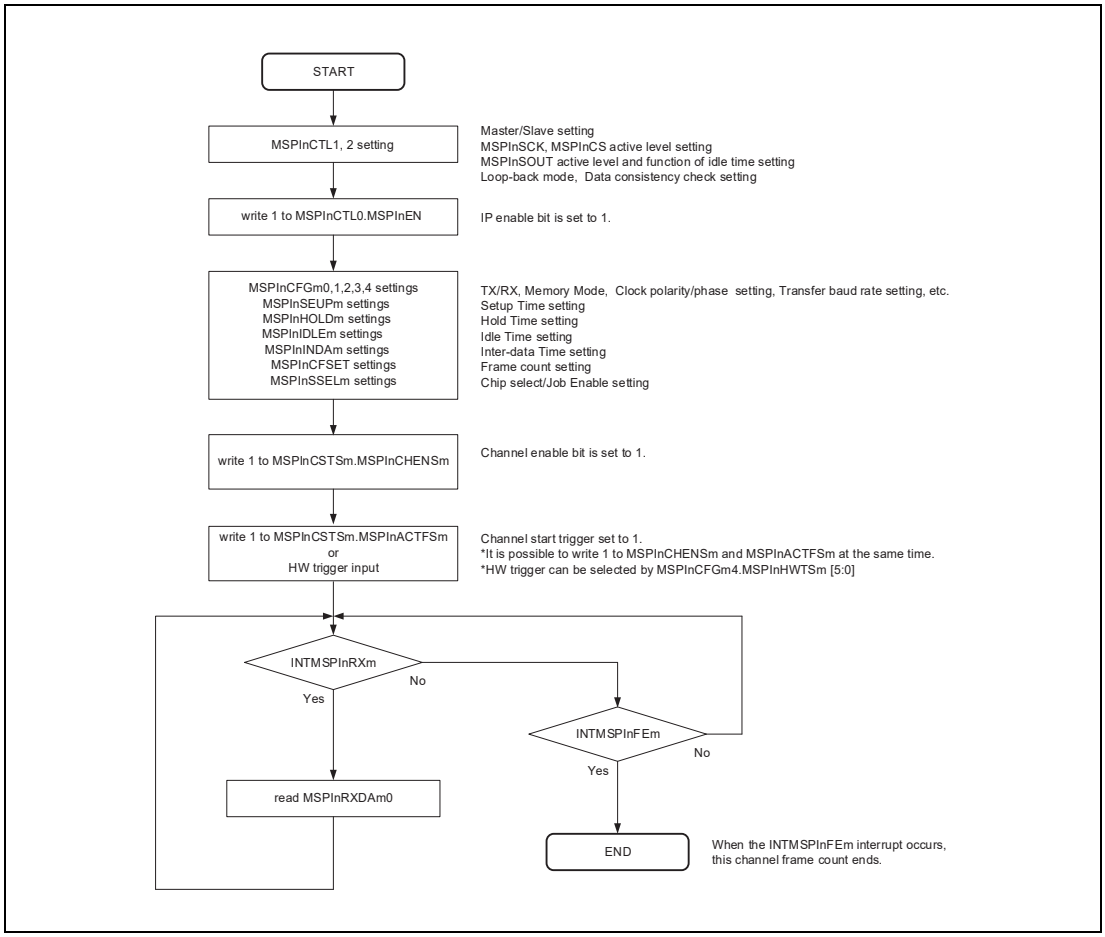
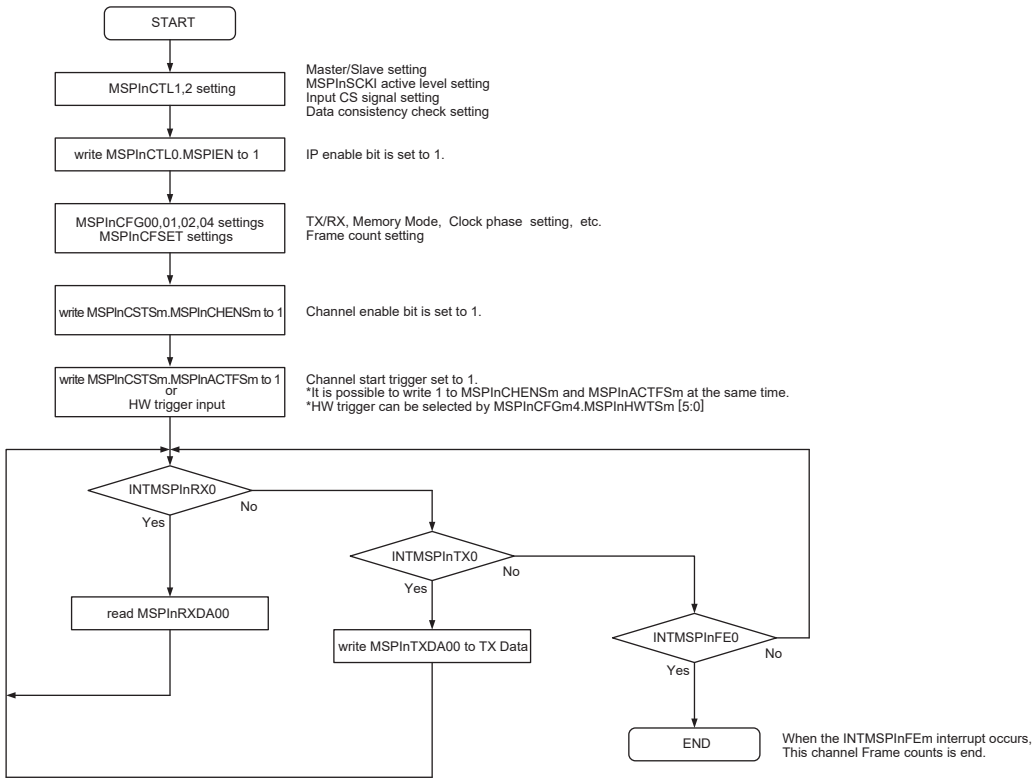
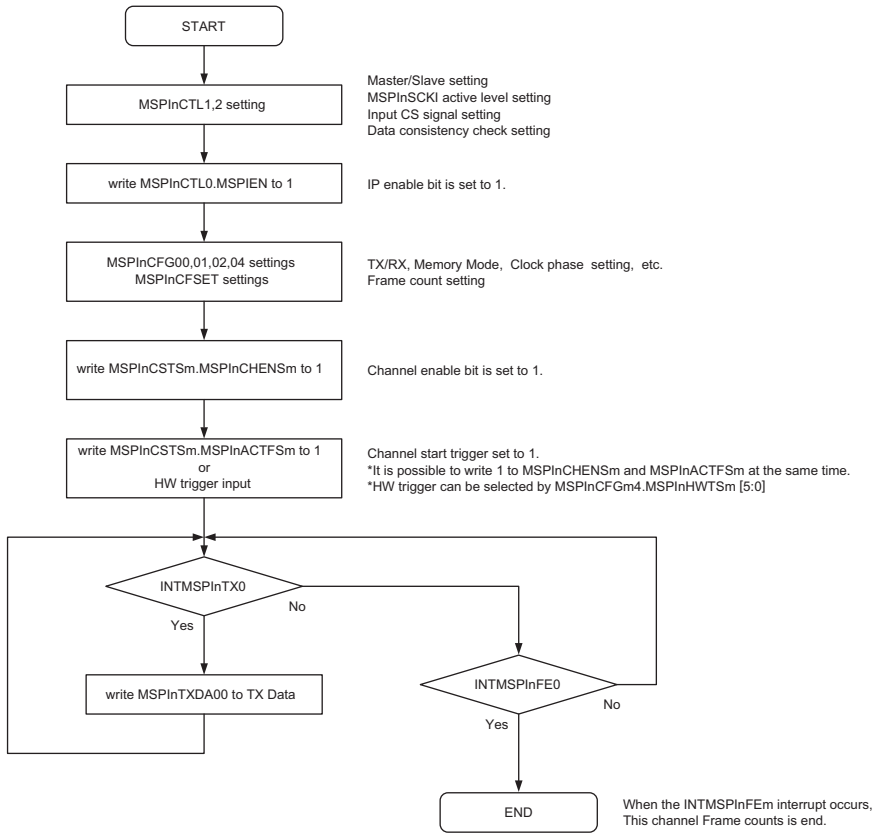


Figure 19.47 Master Reception Operating Procedure in the Direct Memory Mode



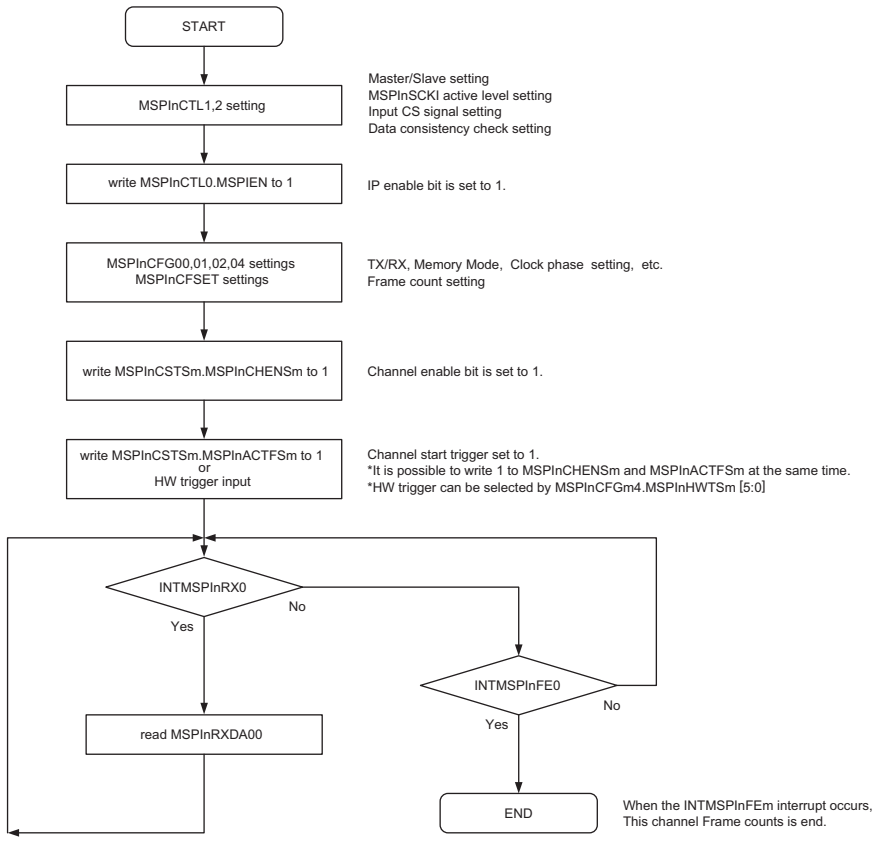
Note 1. In slave mode, communication is controlled by Channel 0.

Figure 19.48 Slave Transmission/Reception Operating Procedure in Direct Memory Mode



Note 1. In slave mode, communication is controlled by Channel 0.

Figure 19.49 Slave Transmission Operating Procedure in the Direct Memory Mode



Note 1. In slave mode, communication is controlled by Channel 0.

Figure 19.50 Slave Reception Operating Procedure in Direct Memory Mode

19.7.2 Fixed Buffer Memory Mode

In the fixed buffer memory mode, the channels use MSPI RAM. Data directly written in the MSPI RAM is used as transmission data, and the received data is stored in the MSPI RAM.

All transmission data must be written before starting communication. All receive data must be read after all communications are completed.

19.7.2.1 Operation

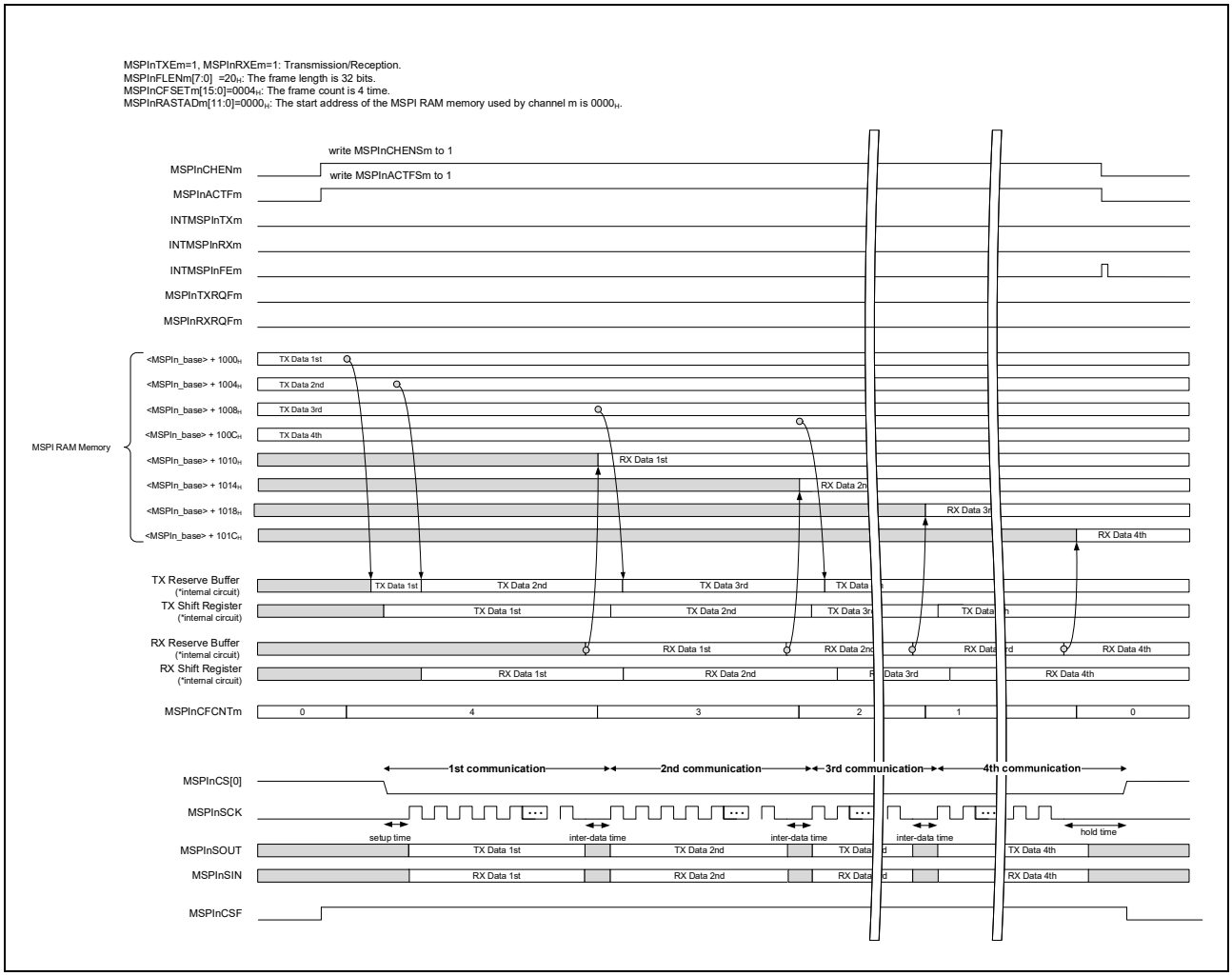


Figure 19.51 Master Transmission/Reception in the Fixed Buffer Memory Mode

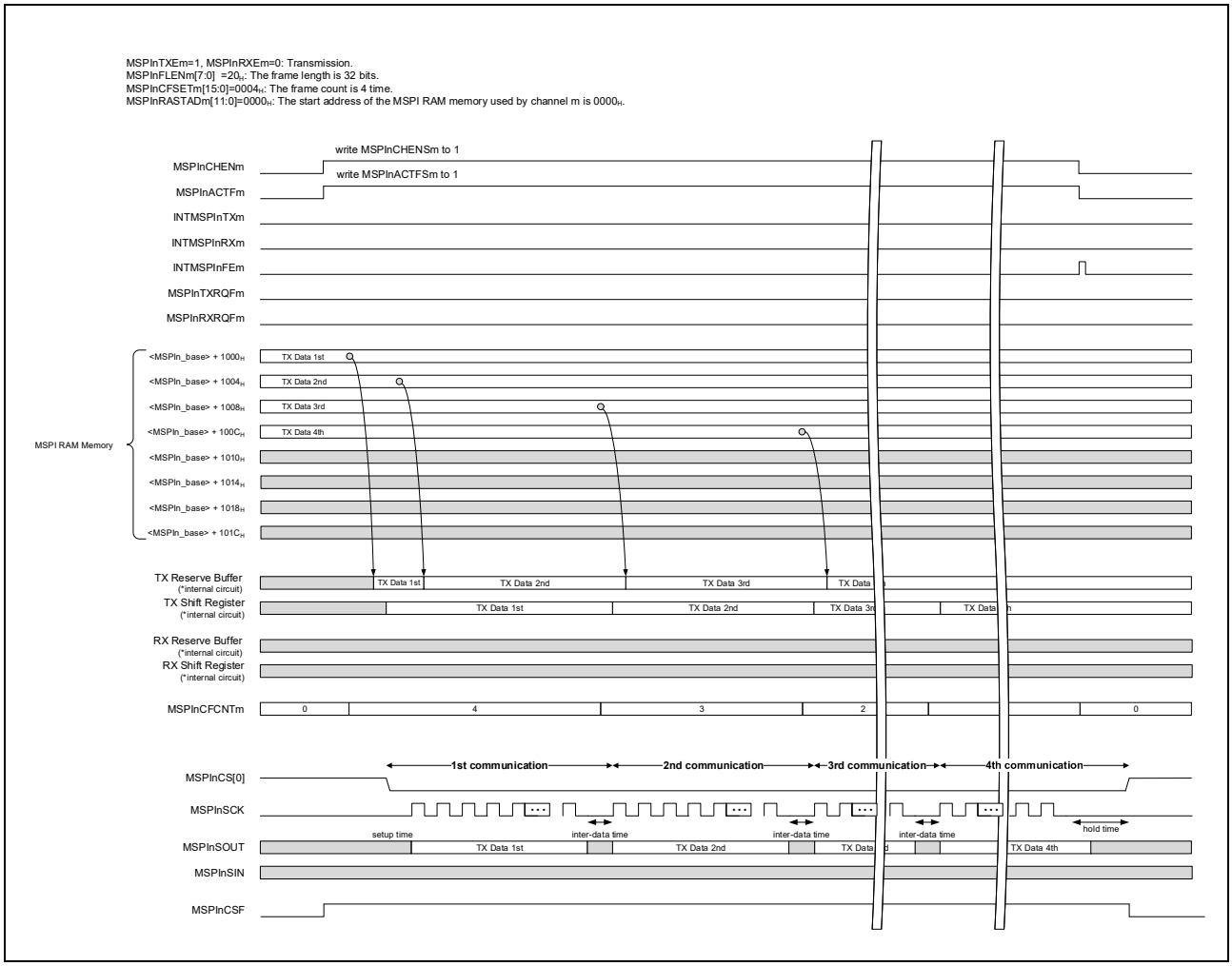


Figure 19.52 Master Transmission in the Fixed Buffer Memory Mode

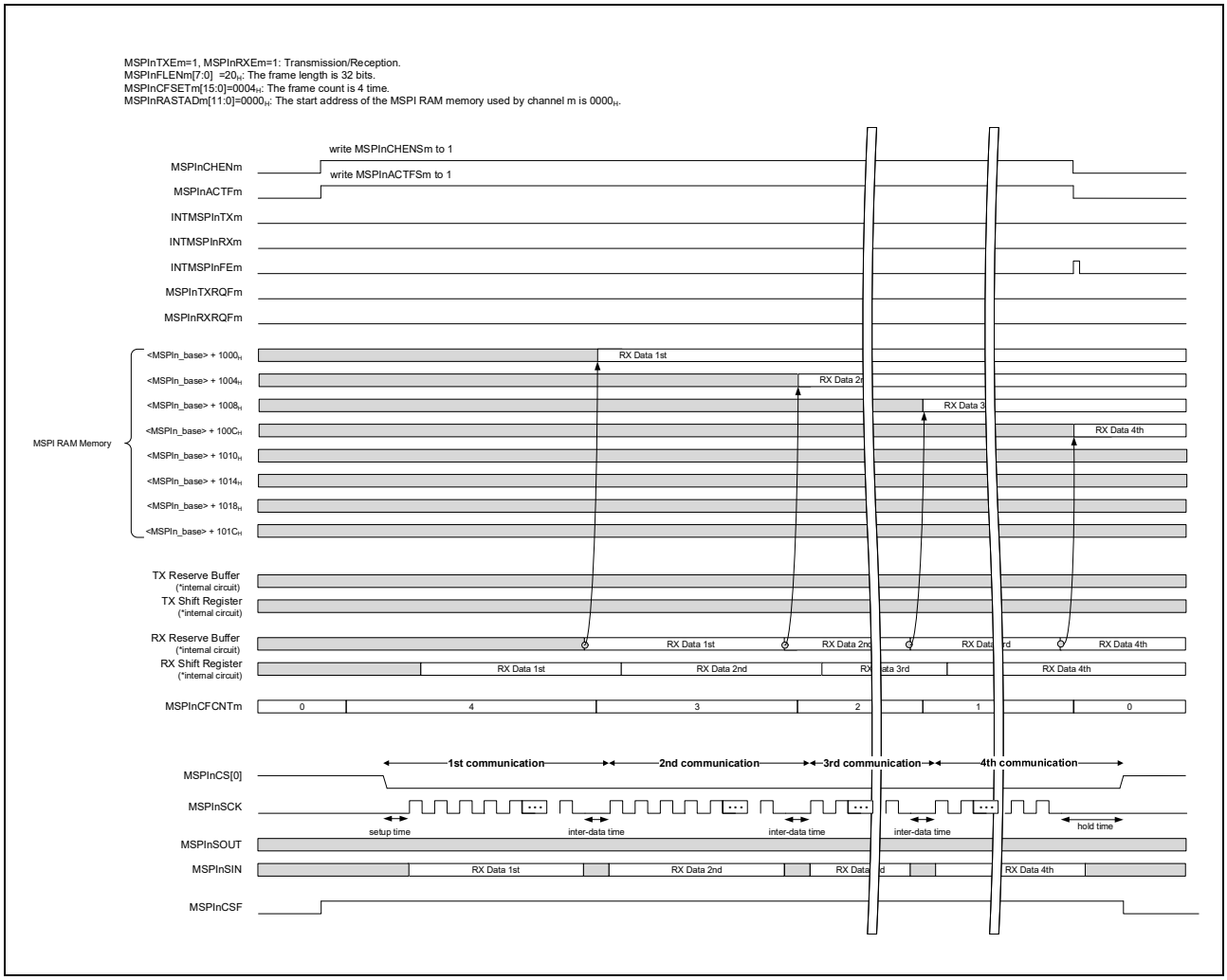
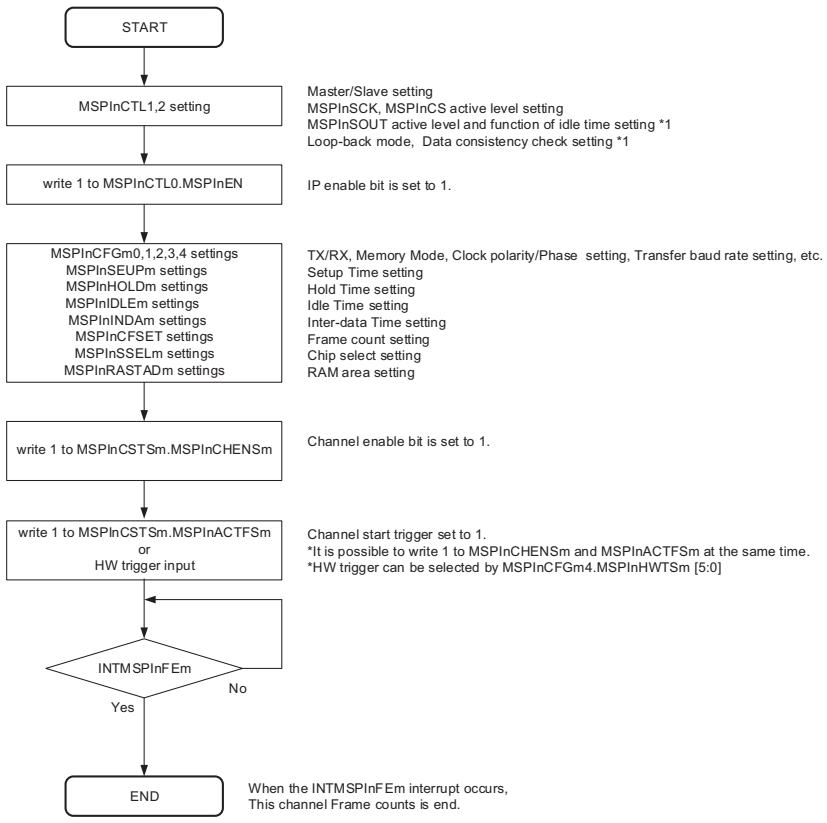


Figure 19.53 Master Reception in the Fixed Buffer Memory Mode

19.7.2.2 Operating Procedure

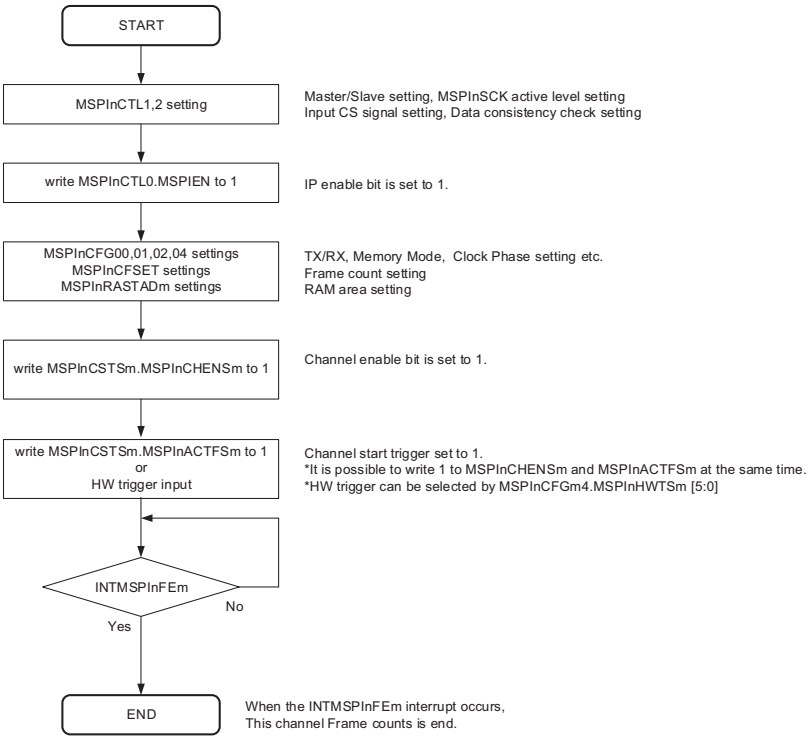
Follow the procedure of this chapter to operate in fixed buffer memory mode.

In the fixed buffer memory mode, an interrupt is output when the Frame count of the communication ends.



Note 1. All transmission data must be written before starting communication, and receive data must be read after all communications are completed.

Figure 19.54 Master Transmission/Reception or Transmission or Reception Operating Procedure in the Fixed Buffer Memory Mode



Note 1. All transmission data must be written before starting communication, and receive data must be read after all communications are completed.

Figure 19.55 Slave Transmission/Reception or Transmission or Reception Operating Procedure in the Fixed Buffer Memory Mode

19.7.2.3 RAM Area

MSPI has its own MSPI RAM, and transmission memory and a reception memory are provided respectively. The MSPI RAM is shared by each channel.

In the fixed buffer memory mode, the RAM area to be used is determined by MSPInCFGm0.MSPInTXEm, MSPInCFGm0.MSPInRXEm, MSPInCFGm2.MSPInFLENm[7:0], MSPInCFSETm and MSPInRASTADm.

Do not configure RAM area exceed the range which is defined in **Table 19.19, Areas of MSPI RAM**.

Table 19.64 Setting Registers of MSPI RAM in the Fixed Buffer Memory Mode

Register Name	Setting
MSPInTXEm	Transmission setting
MSPInRXEm	Reception setting
MSPInFLENm[7:0]	Frame length setting
MSPInCFSETm[15:0]	Frame count number setting
MSPInRASTADm	MSPI RAM start address setting

[Example 1]

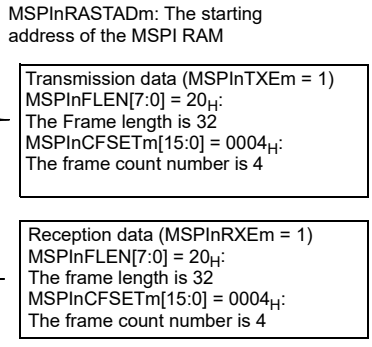
MSPInTXEm, MSPInRXEm = 11

MSPInFLENm[7:0] = 20_H

MSPInCFSETm[15:0] = 0004_H

MSPInRASTADm = 0008_H

Address	
<MSPIn_base> + 1000_H	Not used by Channel m.
<MSPIn_base> + 1004_H	Not used by Channel m.
<MSPIn_base> + 1008_H	Transmission data 1
<MSPIn_base> + $100C_H$	Transmission data 2
<MSPIn_base> + 1010_H	Transmission data 3
<MSPIn_base> + 1014_H	Transmission data 4
<MSPIn_base> + 1018_H	Reception data 1
<MSPIn_base> + $101C_H$	Reception data 2
<MSPIn_base> + 1020_H	Reception data 3
<MSPIn_base> + 1024_H	Reception data 4
<MSPIn_base> + 1028_H	Not used by Channel m.
<MSPIn_base> + $102C_H$	Not used by Channel m.
<MSPIn_base> + 1030_H	Not used by Channel m.
<MSPIn_base> + 1034_H	Not used by Channel m.
<MSPIn_base> + 1038_H	Not used by Channel m.
<MSPIn_base> + $103C_H$	Not used by Channel m.



[Example 2]

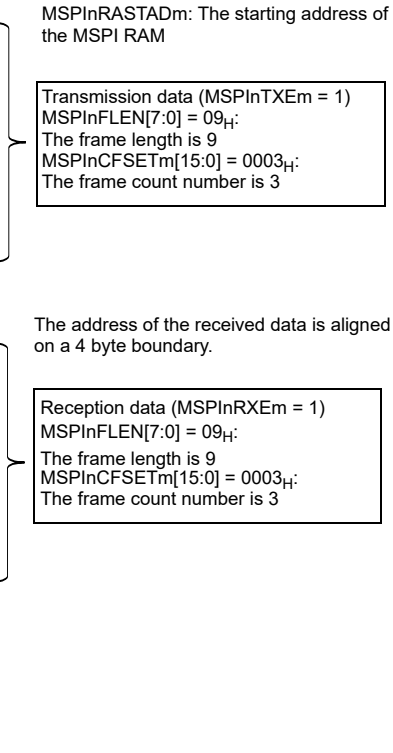
MSPInTXEm, MSPInRXEm = 11

MSPInFLENm[7:0] = 09_H

MSPInCFSETm[15:0] = 0003_H

MSPInRASTADm = 0010_H

Address	
<MSPIn_base> + 1000_H	Not used by Channel m.
<MSPIn_base> + 1001_H	Not used by Channel m.
<MSPIn_base> + 1002_H	Not used by Channel m.
<MSPIn_base> + 1003_H	Not used by Channel m.
<MSPIn_base> + 1004_H	Not used by Channel m.
<MSPIn_base> + 1005_H	Not used by Channel m.
<MSPIn_base> + 1006_H	Not used by Channel m.
<MSPIn_base> + 1007_H	Not used by Channel m.
<MSPIn_base> + 1008_H	Not used by Channel m.
<MSPIn_base> + 1009_H	Not used by Channel m.
<MSPIn_base> + $100A_H$	Not used by Channel m.
<MSPIn_base> + $100B_H$	Not used by Channel m.
<MSPIn_base> + $100C_H$	Not used by Channel m.
<MSPIn_base> + $100D_H$	Not used by Channel m.
<MSPIn_base> + $100E_H$	Not used by Channel m.
<MSPIn_base> + $100F_H$	Not used by Channel m.
<MSPIn_base> + 1010_H	Transmission data 1
<MSPIn_base> + 1011_H	
<MSPIn_base> + 1012_H	Transmission data 2
<MSPIn_base> + 1013_H	
<MSPIn_base> + 1014_H	Transmission data 3
<MSPIn_base> + 1015_H	
<MSPIn_base> + 1016_H	Not used by Channel m.
<MSPIn_base> + 1017_H	Not used by Channel m.
<MSPIn_base> + 1018_H	Reception data 1
<MSPIn_base> + 1019_H	
<MSPIn_base> + $101A_H$	Reception data 2
<MSPIn_base> + $101B_H$	
<MSPIn_base> + $101C_H$	Reception data 3
<MSPIn_base> + $101D_H$	
<MSPIn_base> + $101E_H$	Not used by Channel m.
<MSPIn_base> + $101F_H$	Not used by Channel m.
<MSPIn_base> + 1020_H	Not used by Channel m.
<MSPIn_base> + 1021_H	Not used by Channel m.



[Example 3]

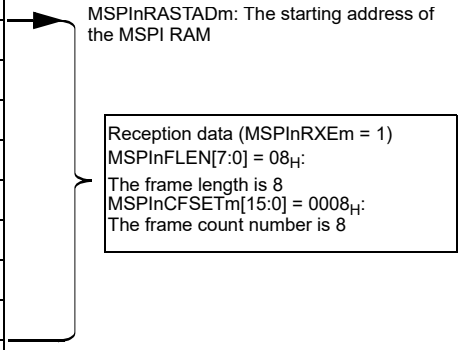
MSPInTXEm, MSPInRXEm = 01

MSPInFLENm[7:0] = 08_H

MSPInCFSETm[15:0] = 0008_H

MSPInRASTADm = 0004_H

Address	
<MSPIn_base> + 1000_H	Not used by Channel m.
<MSPIn_base> + 1001_H	Not used by Channel m.
<MSPIn_base> + 1002_H	Not used by Channel m.
<MSPIn_base> + 1003_H	Not used by Channel m.
<MSPIn_base> + 1004_H	Reception data 1
<MSPIn_base> + 1005_H	Reception data 2
<MSPIn_base> + 1006_H	Reception data 3
<MSPIn_base> + 1007_H	Reception data 4
<MSPIn_base> + 1008_H	Reception data 5
<MSPIn_base> + 1009_H	Reception data 6
<MSPIn_base> + $100A_H$	Reception data 7
<MSPIn_base> + $100B_H$	Reception data 8
<MSPIn_base> + $100C_H$	Not used by Channel m.
<MSPIn_base> + $100D_H$	Not used by Channel m.
<MSPIn_base> + $100E_H$	Not used by Channel m.
<MSPIn_base> + $100F_H$	Not used by Channel m.
<MSPIn_base> + 1010_H	Not used by Channel m.
<MSPIn_base> + 1011_H	Not used by Channel m.
<MSPIn_base> + 1012_H	Not used by Channel m.
<MSPIn_base> + 1013_H	Not used by Channel m.
<MSPIn_base> + 1014_H	Not used by Channel m.
<MSPIn_base> + 1015_H	Not used by Channel m.
<MSPIn_base> + 1016_H	Not used by Channel m.
<MSPIn_base> + 1017_H	Not used by Channel m.
<MSPIn_base> + 1018_H	Not used by Channel m.
<MSPIn_base> + 1019_H	Not used by Channel m.
<MSPIn_base> + $101A_H$	Not used by Channel m.
<MSPIn_base> + $101B_H$	Not used by Channel m.
<MSPIn_base> + $101C_H$	Not used by Channel m.
<MSPIn_base> + $101D_H$	Not used by Channel m.
<MSPIn_base> + $101E_H$	Not used by Channel m.
<MSPIn_base> + $101F_H$	Not used by Channel m.
<MSPIn_base> + 1020_H	Not used by Channel m.
<MSPIn_base> + 1021_H	Not used by Channel m.



*When MSPInTXEm is 0, the channel does not reserve memory for transmission.

19.7.3 Fixed FIFO memory mode

In the fixed FIFO memory mode, the channels use MSPI RAM.

Transmission data written from CPU/DMA to the MSPInTXDAm0 register is automatically stored in MSPI RAM. The reception data is automatically stored in the MSPI RAM, and the CPU/DMA can read received data using MSPInRXDAm0.

A transmission interrupt is output when the buffers of half the number of FIFO stages set by MSPInSIZEm[1:0] become empty, and a reception interrupt is output when the buffers of half the number of FIFO stages set by MSPInSIZEm [1: 0] are filled with receive data.

19.7.3.1 Operation

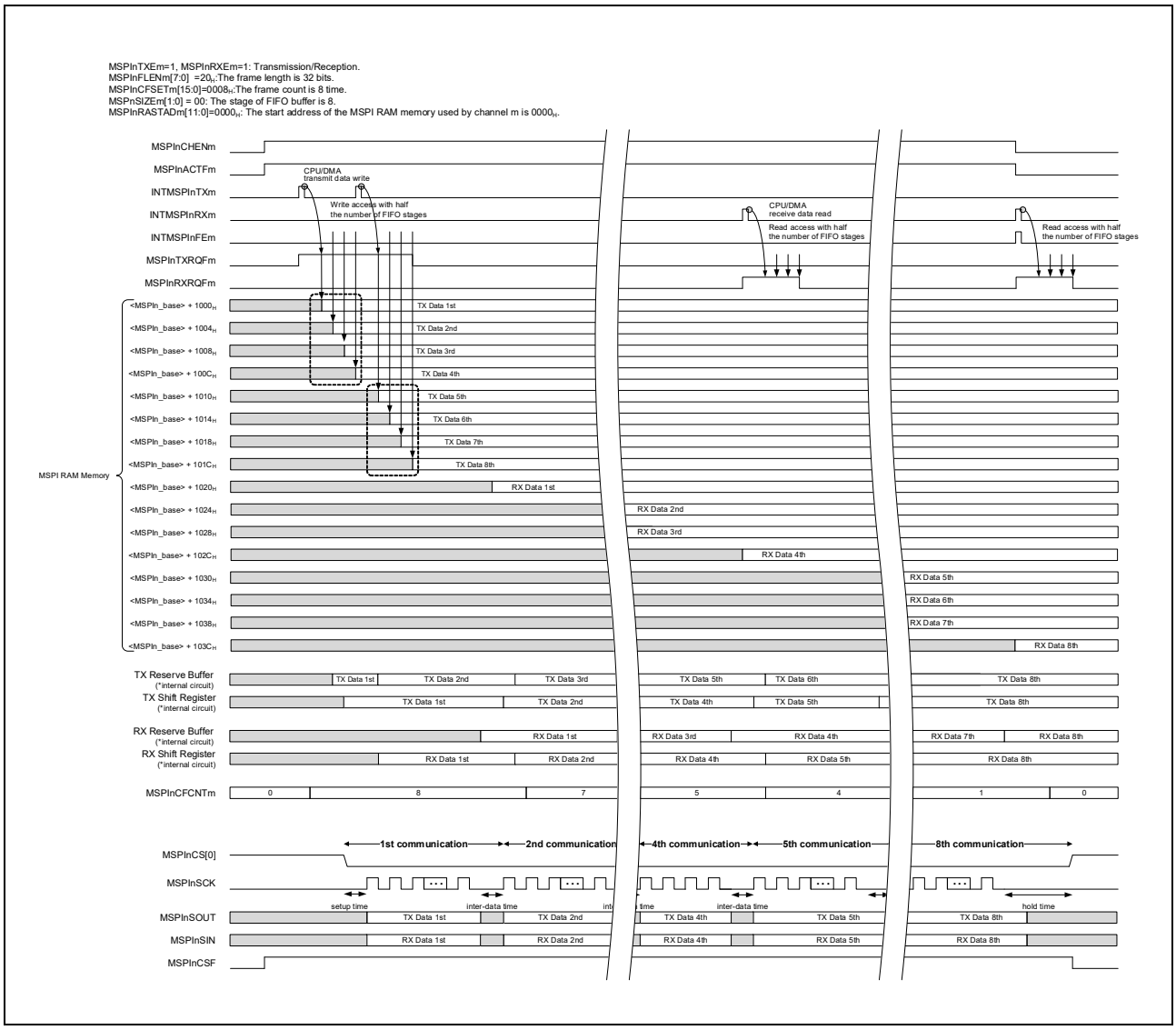


Figure 19.56 Master Transmission/Reception in the Fixed FIFO Memory Mode

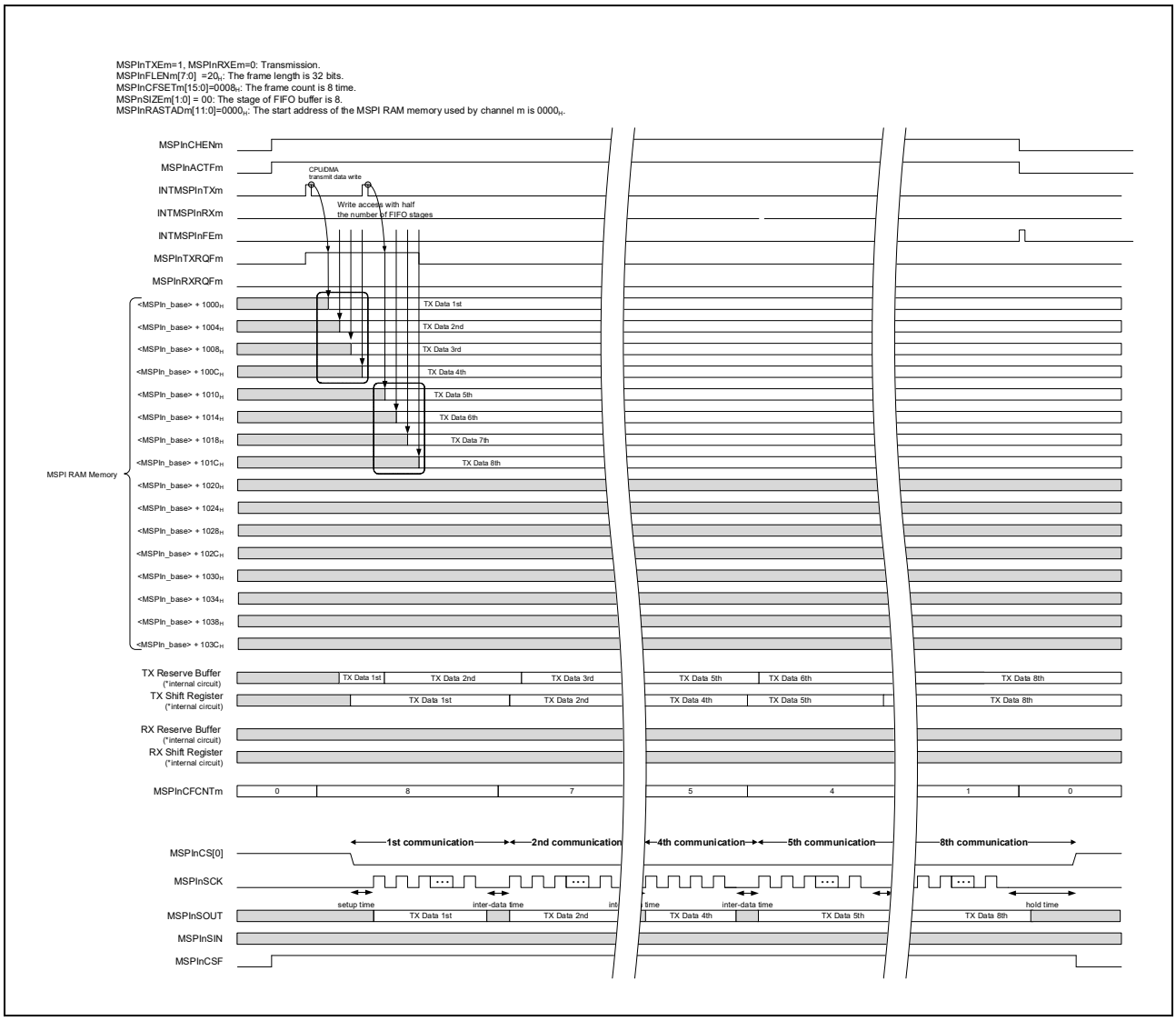


Figure 19.57 Master Transmission in the Fixed FIFO Memory Mode

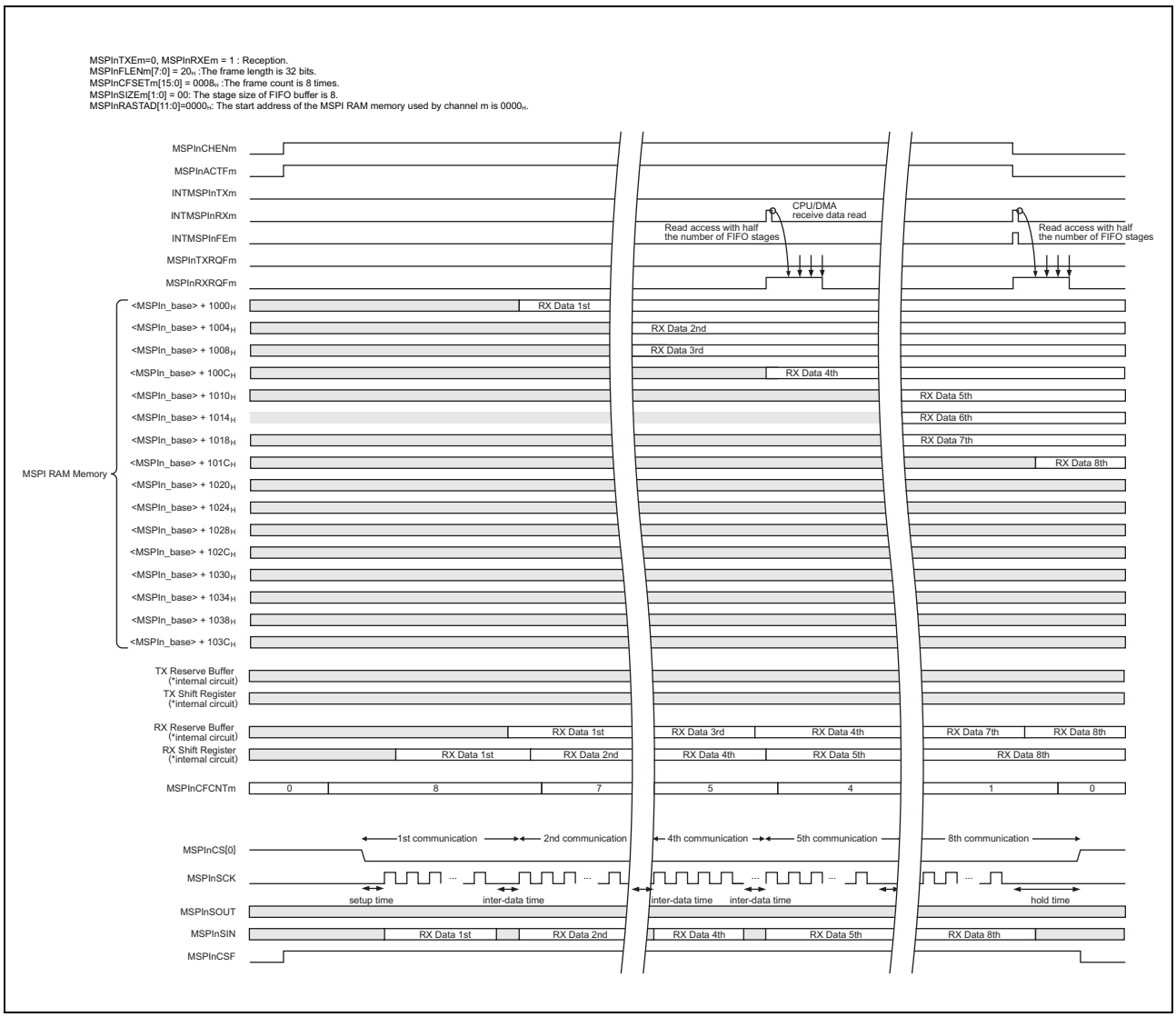
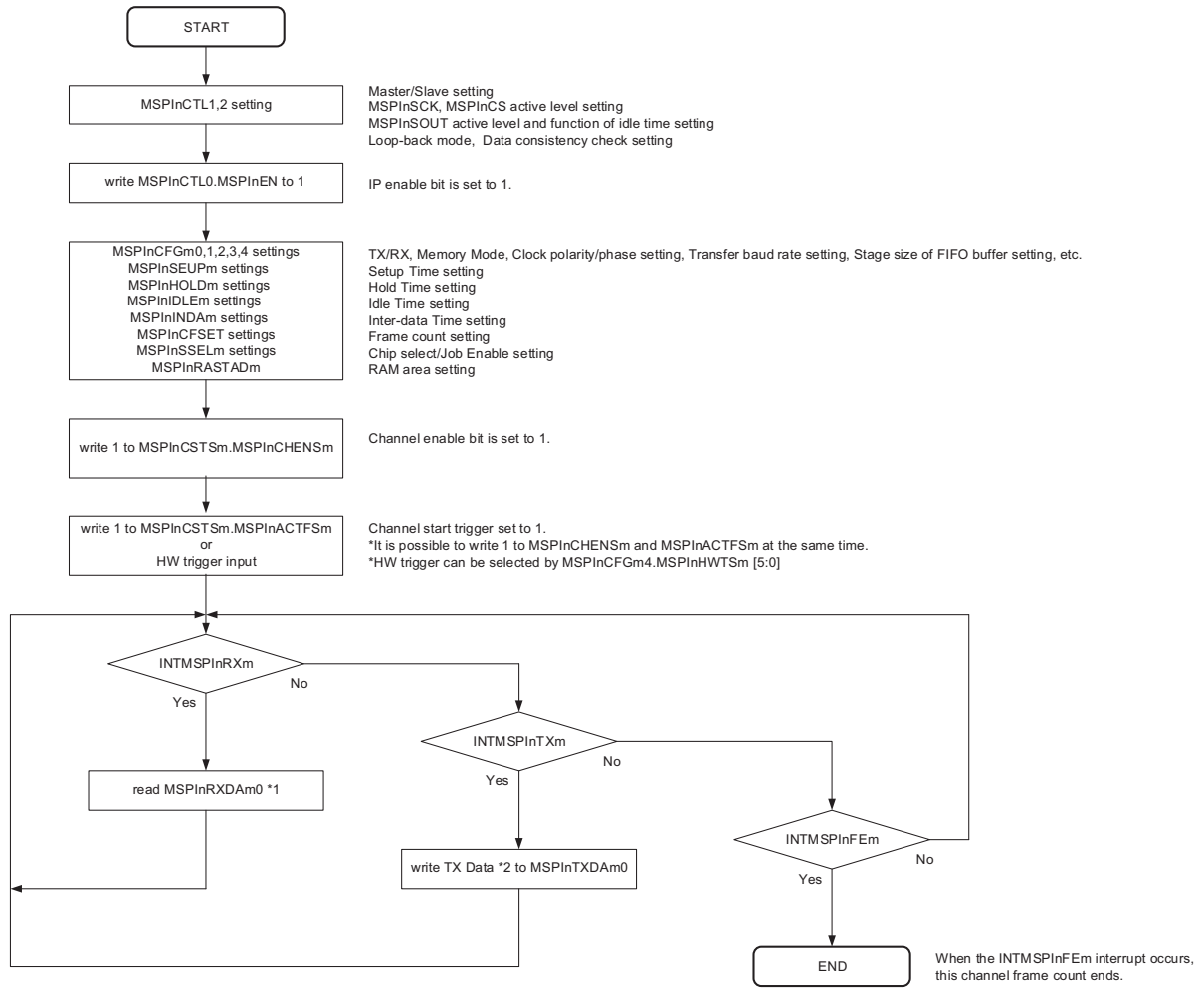


Figure 19.58 Master Reception in the Fixed FIFO Memory Mode

19.7.3.2 Operating Procedure

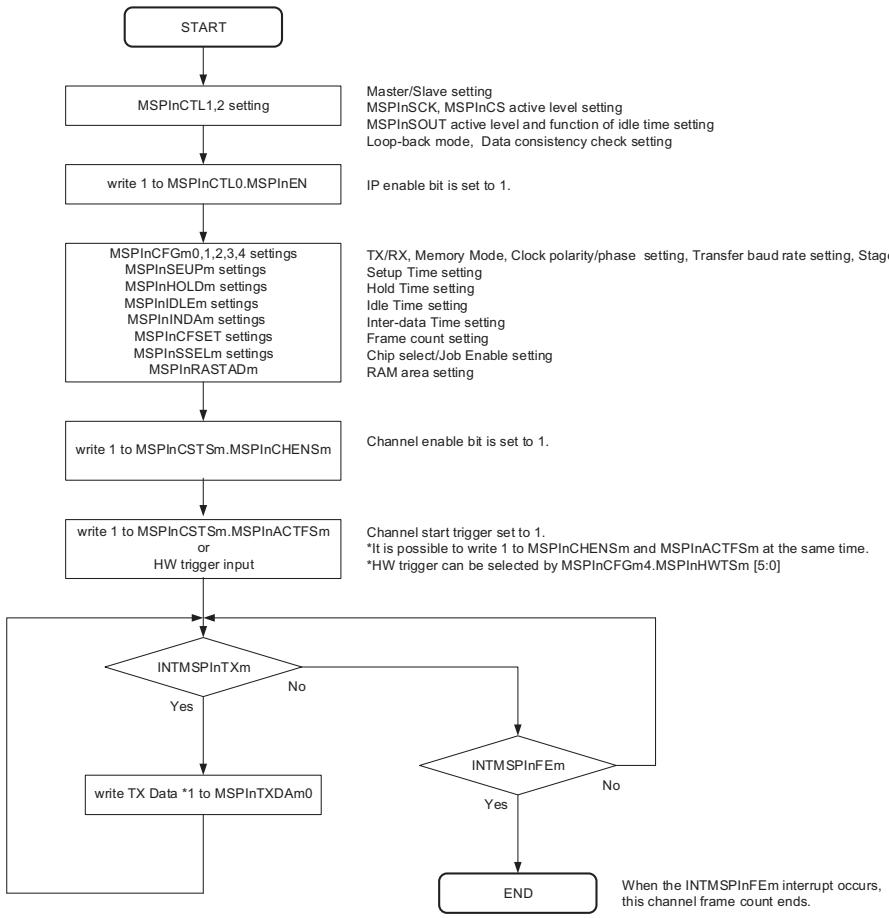
Follow the procedure of this chapter to operate in fixed FIFO memory mode.

In the fixed FIFO memory mode, an interrupt is output when a communication request that is half of the set buffer size is accepted.



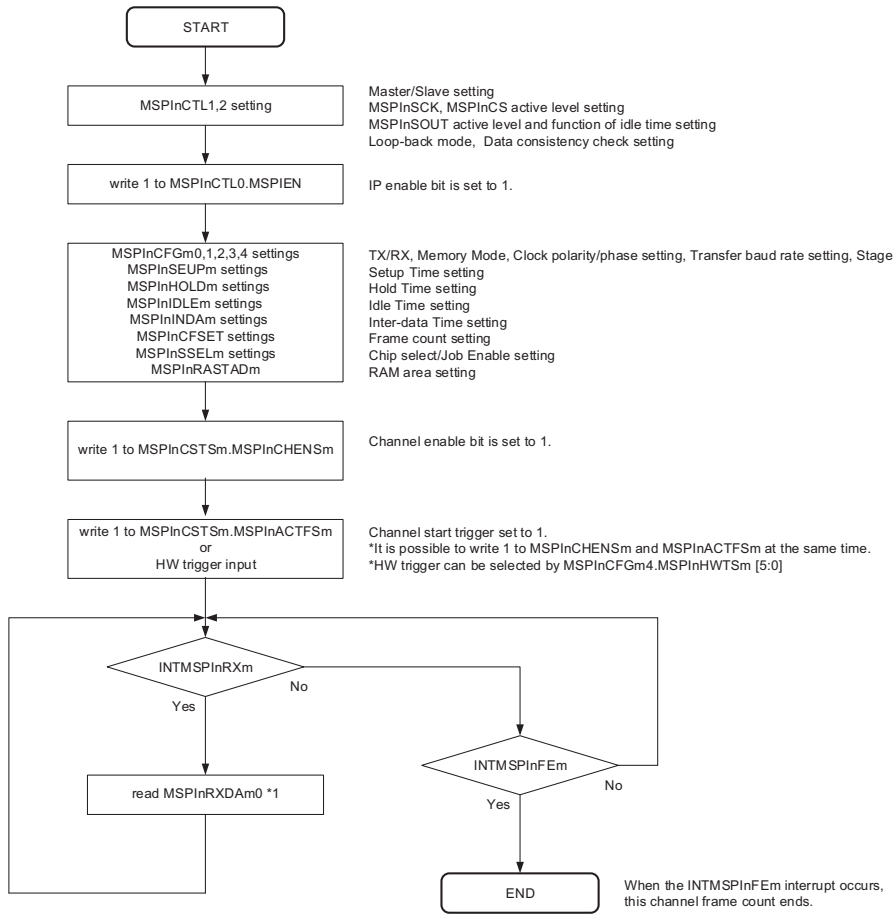
Note 1. It is necessary to read half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]
Note 2. It is necessary to write half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

Figure 19.59 Master Transmission/Reception Operating Procedure in the Fixed FIFO Memory Mode



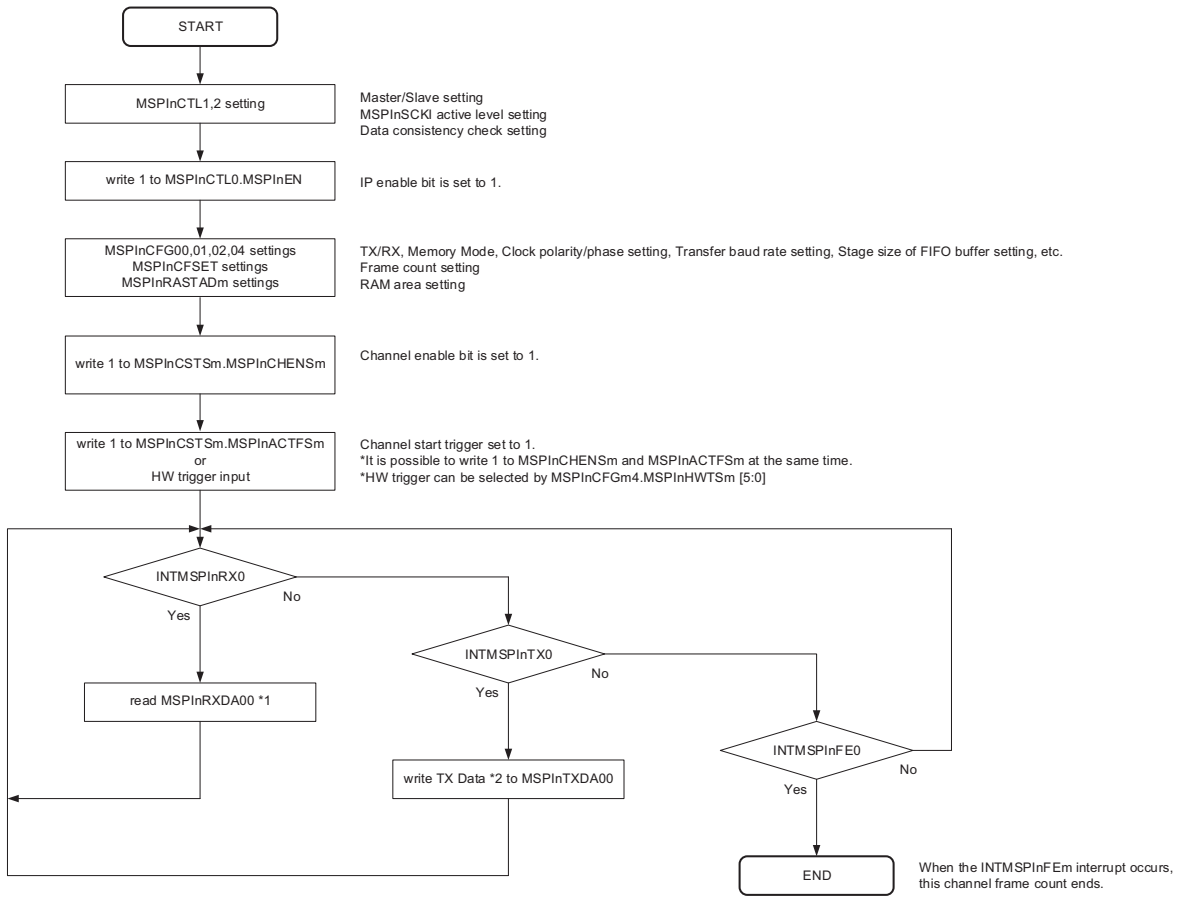
Note 1. It is necessary to write half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

Figure 19.60 Master Transmission Operating Procedure in the Fixed FIFO Memory Mode



Note 1. It is necessary to read half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

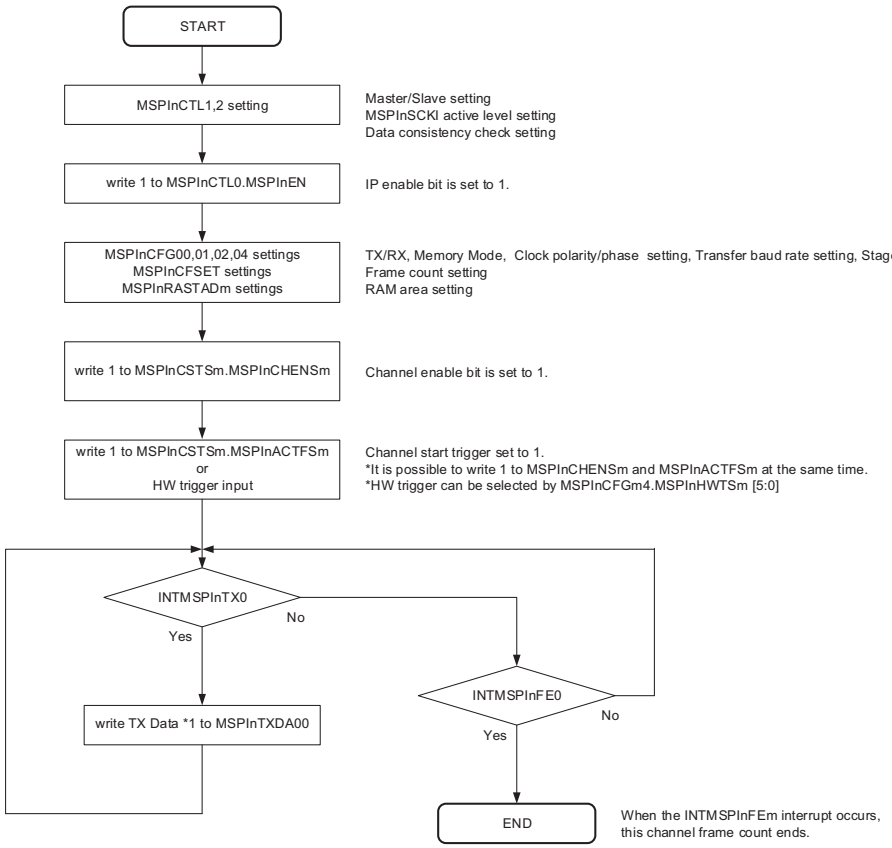
Figure 19.61 Master Reception Operating Procedure in Fixed FIFO Memory Mode



Note 1. It is necessary to read half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

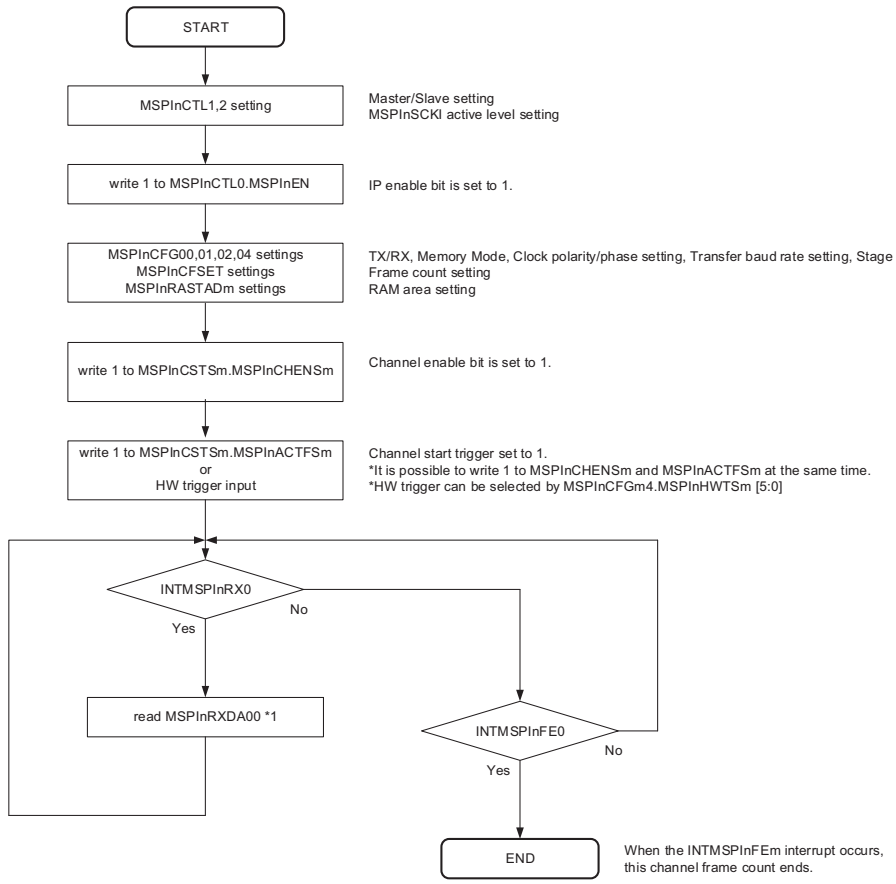
Note 2. It is necessary to write half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

Figure 19.62 Slave Transmission/Reception Operating Procedure in the Fixed FIFO Memory Mode



Note 1. It is necessary to write half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

Figure 19.63 Slave Transmission Operating Procedure in the Fixed FIFO Memory Mode



Note 1. It is necessary to read half the size of FIFO stage set by MSPInCFGm4.MSPInSIZEm [1:0]

Figure 19.64 Slave Reception Operating Procedure in Fixed FIFO Memory Mode

19.7.3.3 RAM Area

MSPI IP has its own MSPI RAM, and transmission memory and a reception memory are provided respectively. The MSPI RAM is shared by each channel.

In the fixed FIFO memory mode, the RAM area to be used is determined by MSPInCFGm0.MSPInTXEm, MSPInCFGm0.MSPInRXEm, MSPInCFGm2.MSPInFLENm[7:0], MSPInCFGm4.MSPInSIZEm[1:0] and MSPInRASTADm.

Do not configure RAM area exceed the range which is defined in **Table 19.19, Areas of MSPI RAM**.

Table 19.65 Setting Registers of MSPI RAM in the Fixed FIFO Memory Mode

Register Name	Setting
MSPInTXEm	Transmission setting
MSPInRXEm	Reception setting
MSPInFLENm[7:0]	Frame length setting
MSPInSIZEm[1:0]	FIFO stage size setting
MSPInRASTADm	MSPI RAM start address setting

[Example 1]

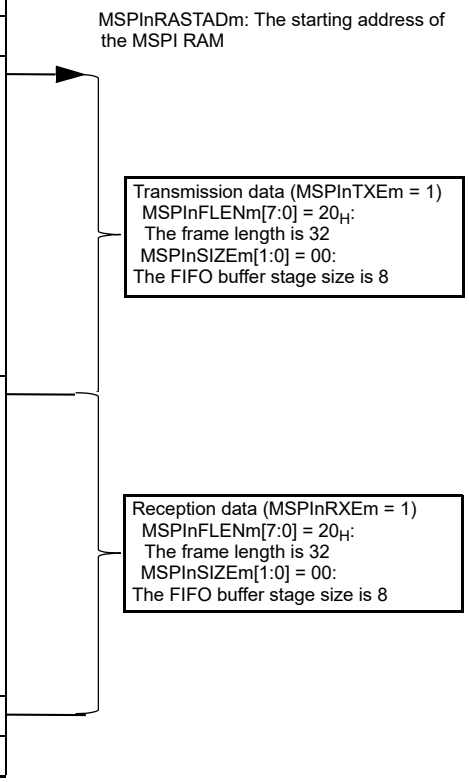
MSPInTXEm, MSPInRXEm = 11

MSPInFLENm[7:0] = 20_H

MSPInSIZEm[1:0] = 00

MSPInRASTADm = 0008_H

Address	
<MSPIn_base> + 1000_H	Not used by Channel m.
<MSPIn_base> + 1004_H	Not used by Channel m.
<MSPIn_base> + 1008_H	The area to be used as the transmission FIFO buffer.
<MSPIn_base> + $100C_H$	
<MSPIn_base> + 1010_H	
<MSPIn_base> + 1014_H	
<MSPIn_base> + 1018_H	
<MSPIn_base> + $101C_H$	
<MSPIn_base> + 1020_H	
<MSPIn_base> + 1024_H	
<MSPIn_base> + 1028_H	The area to be used as the reception FIFO buffer.
<MSPIn_base> + $102C_H$	
<MSPIn_base> + 1020_H	
<MSPIn_base> + 1024_H	
<MSPIn_base> + 1028_H	
<MSPIn_base> + $102C_H$	
<MSPIn_base> + 1030_H	
<MSPIn_base> + 1034_H	
<MSPIn_base> + 1038_H	Not used by Channel m.
:	:



[Example 2]

MSPInTXEm, MSPInRXEm = 11
MSPInFLENm[7:0] = 09_H
MSPInSIZEm[1:0] = 10
MSPInRASTADm = 0000_H

Address	
<MSPIn_base> + 1000_H	The area to be used as the transmission FIFO buffer.
<MSPIn_base> + 1004_H	
:	
<MSPIn_base> + $103C_H$	
<MSPIn_base> + 1040_H	The area to be used as the reception FIFO buffer.
<MSPIn_base> + 1044_H	
:	
<MSPIn_base> + $107C_H$	

MSPInRASTADm: The starting address of the MSPI RAM

Transmission data (MSPInTXEm = 1)
MSPInFLENm[7:0] = 09_H:
The frame length is 9
MSPInSIZEm[1:0] = 10:
The FIFO buffer stage size is 32

Reception data(MSPInRXEm = 1)
MSPInFLENm[7:0] = 09_H:
The frame length is 9
MSPInSIZEm[1:0] = 10:
The FIFO buffer stage size is 32

[Example 3]

MSPInTXEm, MSPInRXEm = 10
MSPInFLENm[7:0] = 08_H
MSPInSIZEm[1:0] = 01
MSPInRASTADm = 0004_H

Address	
<MSPIn_base> + 1000_H	Not used by Channel m.
<MSPIn_base> + 1004_H	The area to be used as the transmission FIFO buffer.
<MSPIn_base> + 1008_H	
<MSPIn_base> + $100C_H$	
<MSPIn_base> + 1010_H	
<MSPIn_base> + 1014_H	Not used by Channel m.
<MSPIn_base> + 1018_H	Not used by Channel m.
<MSPIn_base> + $101C_H$	Not used by Channel m.
:	

MSPInRASTADm: The starting address of the MSPI RAM

Transmission data (MSPInTXEm = 1)
MSPInFLENm[7:0] = 08_H:
The frame length is 8
MSPInSIZEm[1:0] = 01:
The FIFO buffer stage size is 16

19.8 MSPI restrictions

This section shows issues and workarounds of MSPIn.

For products in **Table 19.66, Affected products** follow each workarounds.

For products in **Table 19.67, Not affected products**, it is no need to follow them.

Table 19.66 Affected products

Product Name	Note
R7F702Z19AEDBG	U2A-EVA BGA516
R7F702300EBBG-C	U2A16 BGA516
R7F702300EBBB-C	U2A16 BGA373
R7F702300EABA-C	U2A16 BGA292
R7F702301EBBA-C	U2A8 BGA373
R7F702301EABG-C	U2A8 BGA292

Table 19.67 Not affected products

Product Name	Note
R7F702Z19BFDBG	U2A-EVA BGA516
R7F702300AEBBC-C	U2A16 BGA516
R7F702300AEBBB-C	U2A16 BGA373
R7F702300AFABA-C	U2A16 BGA292
R7F702300BEBBC-C	U2A16 BGA516
R7F702300BEBBB-C	U2A16 BGA373
R7F702300BFABA-C	U2A16 BGA292
R7F702301AEBBA-C	U2A8 BGA373
R7F702301AFABG-C	U2A8 BGA292
R7F702301BEBBA-C	U2A8 BGA373
R7F702301BFABG-C	U2A8 BGA292
R7F702302FABB-C	U2A6 BGA292
R7F702302FABD-C	U2A6 BGA156
R7F702302FAFK-C	U2A6 QFP176
R7F702302FAFM-C	U2A6 QFP144

19.8.1 Issue #1: Misjudgment of Overrun

19.8.1.1 Phenomenon

Unexpected behaviors occur under certain conditions as follows:

(1) Issue #1-(1)

If there is the case that the reception data buffer would be full*1 when MSPInCFGm0.MSPInRXEm = 1 at master mode and direct memory mode, an extra reception interrupt is generated. The flags of MSPInRXDAm0.MSPInRXDAm0[31:0] and MSPInCSTRm.MSPInRXRQFm are influenced by this issue.

(2) Issue #1-(2)

If there is the case that the reception data buffer would be full*1 when MSPInCFGm0.MSPInRXEm = 1 and MSPInCFGm0.MSPInFCCEm = 0 at master mode and direct memory mode, the necessary reception interrupt is not generated at the frame end. The flags of MSPInRXDAm0.MSPInRXDAm0[31:0] and MSPInCSTRm.MSPInRXRQFm are influenced by this issue.

(3) Issue #1-(3)

If the number of reception frames is larger than the number of FIFOs when MSPInCFGm0.MSPInRXEm = 1 at master mode and fixed FIFO memory mode, the reception interrupt is generated before reception completion. The flags of MSPInRXDAm0.MSPInRXDAm0[31:0], MSPInCSTRm.MSPInFIRXNm[7:0] and MSPInRXRQFm are influenced by this issue.

(4) Issue #1-(4)

If there is the case that reception data buffer would be full*1 when MSPInCFGm0.MSPInRXEm = 1 at slave mode and direct memory mode, an over run error cannot be detected. Additionally, if the number of reception frame is larger than the number of FIFO when MSPInCFGm0.MSPInRXEm = 1 at slave mode and fixed FIFO memory mode, an over run error cannot be detected. The flags of MSPInRXDAm0.MSPInRXDAm0[31:0], MSPInCSTRm.MSPInFIRXNm[7:0], MSPInRXRQFm, MSPInACTFm, MSPInCHENm, MSPInCFCNTm.MSPInCFCNTm[15:0], MSPInCESTm.MSPInOVREEm and MSPInOVRUEm are influenced by this issue.

Note 1. It means that after the reception completion interrupt of the 1st data, CPU does not read the following 2 data until their reception is completed. (Refer to **Section 19.6.8.4, Overrun Error Check**.)

19.8.1.2 Workaround

Issue #1-(1) to (4) can be avoided by the following workarounds.

(1) Issue #1-(1) to (3)

These issues occur by a conflict between the reception completion and reception data read when the reception buffer is full at master mode. Therefore, it is necessary to control the reception buffer so that it does not become full. Generally, data is written by INTMSPInTX, but it should be used only for the first 2 data. For the third and later data, write the transmission data by INTMSPInRX after the reception data read. (Refer to **Figure 19.65**.) The reception buffer can be controlled so that it does not become full by controlling the transmission clock. Therefore, these issues can be avoided by this workaround.

(2) Issue #1-(4)

This issue occurs by a conflict between the reception completion and reception data read when the reception buffer is full at slave mode. Therefore, it is necessary to control the reception buffer so that it does not become full. Read the reception data within the transmission time for the next frame data*1 after INTMSPInRX occurs. The reception buffer can be controlled so that it does not become full by this workaround and this issue can be avoided.

Note 1. “Transmission time for the next frame data” means the following.

Direct memory mode:

The transmission time for the frame length defined by MSPInCFGm2.MSPInFLENm[7:0]. But in case of a setting larger than 32bit, since both transmission times for 32bit and for fraction bits are generated, “transmission time for the next frame data” means the transmission time for the fraction bits.

Fixed FIFO memory mode:

The time for storing data which is the half of the located FIFO memory size.

The circuit will be revised for issue #1-(1) to (4).

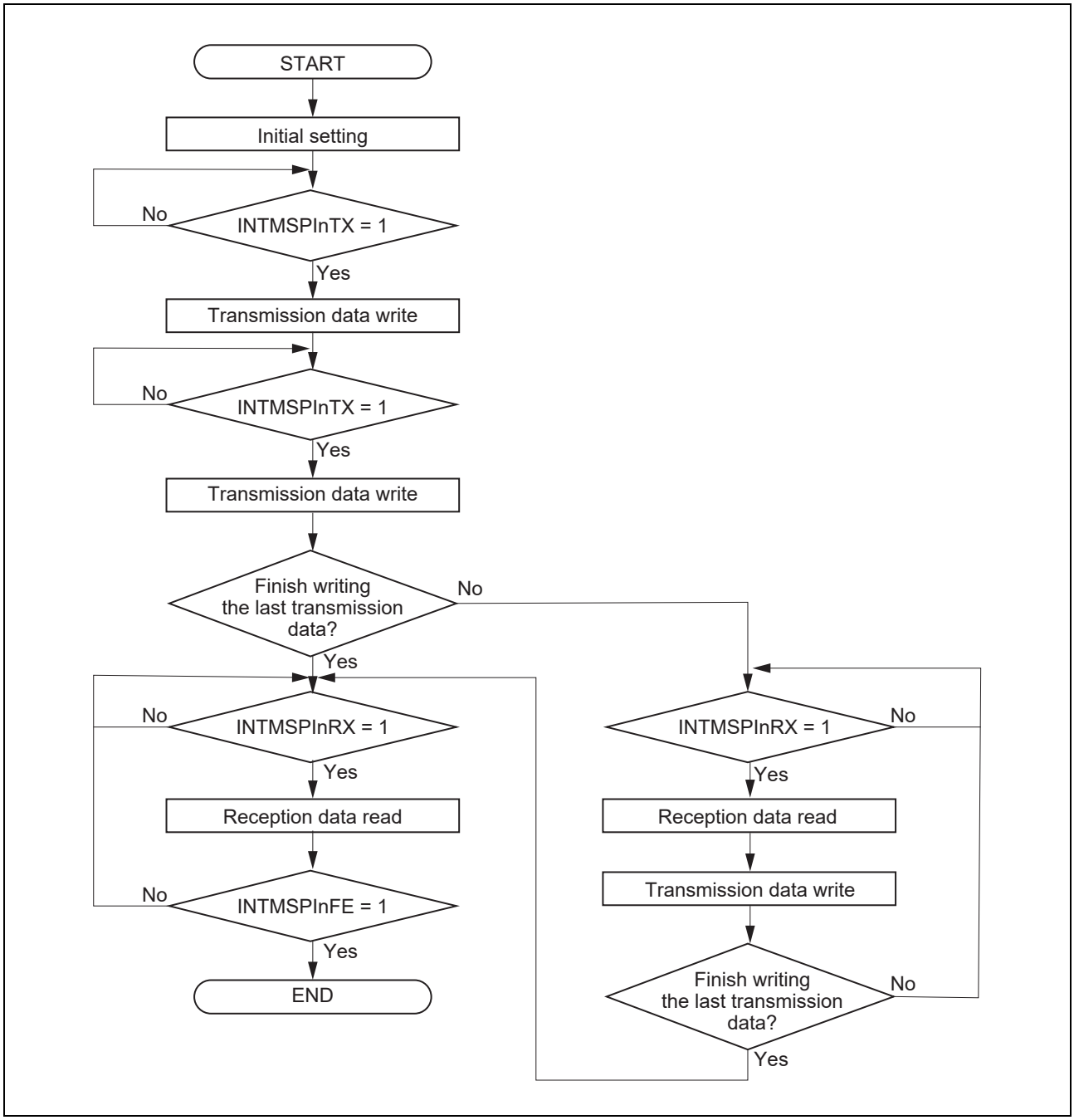


Figure 19.65 Workaround for issue #1-(1) to (3)

Refer to following figures.

Figure 19.45, Master Transmission/Reception Operating Procedure in Direct Memory Mode

Figure 19.46, Master Transmission Operating Procedure in the Direct Memory Mode

Figure 19.59, Master Transmission/Reception Operating Procedure in the Fixed FIFO Memory Mode

Figure 19.61, Master Reception Operating Procedure in Fixed FIFO Memory Mode

19.8.2 Issue #2: Misalignment of Bit at Slave Transmission

19.8.2.1 Phenomenon

When MSPInCFGm0.MSPInTXEm = 1, MSPInCFGm1.MSPInCPHAm = 0 and MSPInCFGm0.MSPInFCCEm = 0, the next transmission data after communication completion is misaligned.

19.8.2.2 Workaround

Issue #2 can be avoided by one of the following workarounds.

(1) Workaround 1

Set MSPInCFGm1.MSPInCPHAm = 1.

(2) Workaround 2

Set MSPInCFGm0.MSPInFCCEm = 1.
This workaround is available only at direct memory mode.
The transmission frame is not counted by MSPI in case of this workaround. Therefore, it is necessary to be counted by DMA etc.

(3) Workaround 3

At each transmission completion, after setting MSPInCTL0.MSPInEN = 0 set MSPInCTL0.MSPInEN = 1 again, before use it.

Section 20 Serial Communication Interface 3 (SCI3)

The serial communication interface 3 (SCI3: Serial Communication Interface 3) can handle two methods of serial communications: asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communication LSIs such as Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). Asynchronous mode is equipped with a serial communications function between multiple processors (multi-processor communications function).

20.1 Features SCI3 for RH850/U2A-EVA

20.1.1 Number of Units and Channels

This microcontroller has the following number of SCI3 units.

Each unit has one channel SCI3. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 20.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	—	—	—	—
Name	SCI3n									

Table 20.2 Unit Configurations and Channels

Unit Name SCI3n	Channels per Unit	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
SCI30	1	√	√	√	√	√	√	—	—	—	—
SCI31	1	√	√	√	√	√	√	—	—	—	—
SCI32	1	√	√	√	√	√	√	—	—	—	—

Note: The channel name is the same as the unit name.

Table 20.3 Indices

Index	Description
n	Throughout this section, the individual SCI3 units are identified by the index “n” (n = 0 to 2): for example, SCI3nSMR is the serial mode register.

20.1.2 Register Base Addresses

SCI3 base addresses are listed in the following table.

SCI3 register addresses are given as offsets from these base addresses.

Table 20.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<SCI30_base>	FFD3 1000_H	P-Bus Group 3
<SCI31_base>	FFD3 2000_H	P-Bus Group 3
<SCI32_base>	FFD3 3000_H	P-Bus Group 3

20.1.3 Clock Supply

Clock supply by and to SCI3 is listed in the following table.

Table 20.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
SCI3n	PCLK	High-speed peripheral clock CLK_HSB

20.1.4 Interrupt Requests and Error Notification

SCI3 interrupt requests are listed in the following table.

Table 20.6 Interrupt and DMA/DTS Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
SCI30				
INTSCI30ERI	Receive error	348	—	—
INTSCI30RXI	Receive data full	349	Group0-140	Group3-10
INTSCI30TXI	Transmit data empty	350	Group0-141	Group3-11
INTSCI30TEI	End of transmission	351	—	—
SCI31				
INTSCI31ERI	Receive error	352	—	—
INTSCI31RXI	Receive data full	353	Group0-142	Group3-12
INTSCI31TXI	Transmit data empty	354	Group0-143	Group3-13
INTSCI31TEI	End of transmission	355	—	—
SCI32				
INTSCI32ERI	Receive error	356	—	—
INTSCI32RXI	Receive data full	357	Group0-144	Group3-14
INTSCI32TXI	Transmit data empty	358	Group0-145	Group3-15
INTSCI32TEI	End of transmission	359	—	—

This module have no error notifications.

20.1.5 Reset Sources

For details, refer to **Section 9, Reset Controller**.

Table 20.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
SCI3n	All registers	√	√	√	√	√	√	—

20.1.6 External Input/Output Signals

External input/output signals of SCI3 are listed below.

Table 20.8 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal
SCI30			
SCI0SCK	I	SCI30 serial clock input	SCI30SCK
	O	SCI30 serial clock output	SCI30SCK
SCI0RxD	I	SCI30 data input signal	SCI30RXD
SCI0TxD	O	SCI30 data output signal	SCI30TXD
SCI31			
SCI1SCK	I	SCI31 serial clock input	SCI31SCK
	O	SCI31 serial clock output	SCI31SCK
SCI1RxD	I	SCI31 data input signal	SCI31RXD
SCI1TxD	O	SCI31 data output signal	SCI31TXD
SCI32			
SCI2SCK	I	SCI32 serial clock input	SCI32SCK
	O	SCI32 serial clock output	SCI32SCK
SCI2RxD	I	SCI32 data input signal	SCI32RXD
SCI2TxD	O	SCI32 data output signal	SCI32TXD

20.1.7 Combination of Pins and Ports

Combinations of SCI3 pins and ports are listed in the following table.

Table 20.9 Combinations of Pins and Ports

Function	Pin Name	Port Name
SCI30	SCI30RXD	P6_2
	SCI30TXD	P6_3
	SCI30SCK	P6_4
SCI31	SCI31RXD	P6_5
	SCI31TXD	P6_6
	SCI31SCK	P6_7
SCI32	SCI32RXD	P20_0
	SCI32TXD	P20_1
	SCI32SCK	P20_3

20.2 Overview

- The serial data communication mode can be configured for asynchronous communications or clock synchronous communications.
- Full-duplex communications are available. The independent transmitter unit and receiver unit allow simultaneous transmission and reception. Both the transmitter and the receiver have a double-buffered structure, enabling continuous data transmission and reception.
- An arbitrary bit rate is selectable with the on-chip baud rate generator. An external clock is also selectable as a transmission/reception clock source.
- LSB-first or MSB-first transfer is selectable (except for asynchronous 7-bit data.)
- Interrupt sources: 4 types consisting of transmit-end, transmit-data-empty, receive-data-full, and receive-error. Transmit-data-empty and receive-data-full interrupt sources can activate the DMAC.
- Module standby can be set for each channel (for details of module standby, see **Section 15, Standby Controller (STBC)**).
- The bit rate modulation function can reduce errors averagely (even in a high bit rate) by correcting the output of the on-chip baud rate generator (except for the maximum speed in clock synchronous mode).
- The pin level of serial input data can be checked.

20.2.1 Serial Communication Modes

Asynchronous mode

- Data length: 7 bits, 8 bits or 16 bits selectable
- Stop bit length: 1 bit or 2 bits selectable
- Parity: Even parity, odd parity, or none selectable
- Receive error detection: Parity error, overrun error, and framing error
- Break detection: A break can be detected by reading a register when a framing error occurs

Clock synchronous mode

- Data length: 8 bits or 16 bits
- Receive error detection: Overrun error

20.2.2 Block Diagram

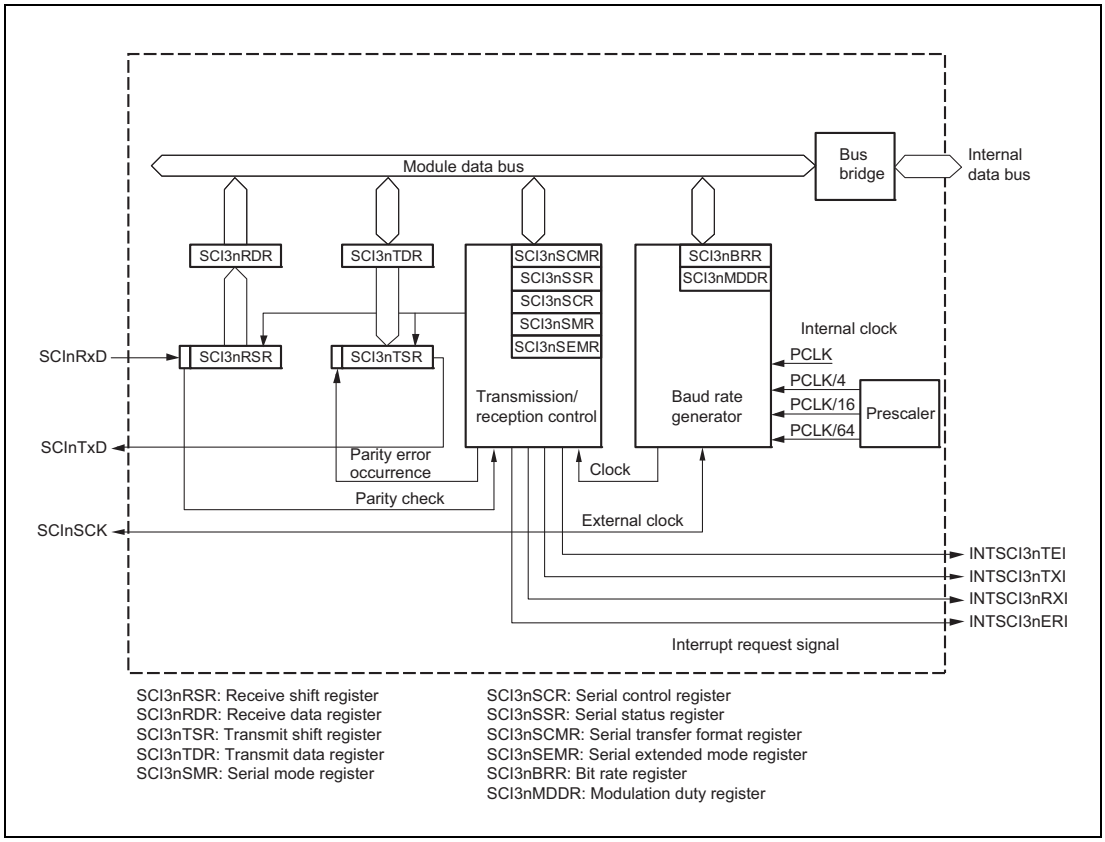


Figure 20.1 Block Diagram

20.3 Registers

20.3.1 List of Registers

The SCI3 has the following registers. Some registers have limitations in read/write by the CPU.
For <SCI3n_base>, refer to **Section 20.1.2, Register Base Addresses**.

CAUTION

SCI3nBRR and SCI3nMDDR are allocated to the same address (relative address 4). The MDDRS bit in SCI3nSEMR is used to switch these registers.

Table 20.10 List of Registers

Register Name	Symbol*1	Value after Reset	Address	Access Protection	
				PBG	Other
Receive shift register	SCI3nRSR	—	—	—	—
Serial mode register	SCI3nSMR	00_H	<SCI3n_base> + 0000_H	*2	—
Bit rate register / Modulation duty register	SCI3nBRR/ SCI3nMDDR	FF_H	<SCI3n_base> + 0004_H	*2	—
Serial control register	SCI3nSCR	00_H	<SCI3n_base> + 0008_H	*2	—
Transmit data register	SCI3nTDR	$FFFF_H$	<SCI3n_base> + $000C_H$	*2	—
Transmit shift register	SCI3nTSR	—	—	—	—
Serial status register	SCI3nSSR	84_H	<SCI3n_base> + 0010_H	*2	—
Receive data register	SCI3nRDR	0000_H	<SCI3n_base> + 0014_H	*2	—
Serial transfer format register	SCI3nSCMR	$F2_H$	<SCI3n_base> + 0018_H	*2	—
Serial extended mode register	SCI3nSEMR	04_H	<SCI3n_base> + $001C_H$	*2	—

Note 1. n = 0 to 2
Note 2. n = 0: PBG30#9
n = 1: PBG30#10
n = 2: PBG30#11

Relative addresses 4k + 1, 4k + 2, and 4k + 3 (k = 0 to 7) are reserved areas. These areas are always read as 0. Writing is invalid.

20.3.2 SCI3nRSR — Receive Shift Register

SCI3nRSR is a shift register which is used to receive serial data input from the SCInRxD pin and convert it to parallel data. When one frame of data has been received, it is automatically transferred to SCI3nRDR. SCI3nRSR cannot be directly accessed by the CPU.

20.3.3 SCI3nRDR — Receive Data Register

SCI3nRDR is an 16-bit register to store receive data. The value of SCI3nRDR after a reset is 0000_H. When one frame of data has been received, it is transferred from SCI3nRSR to this register allowing SCI3nRSR to receive the next data. SCI3nRSR and SCI3nRDR function as a double-buffer, allowing continuous receive operations. Be sure to check that the RDRF flag in SCI3nSSR is set to 1 before reading SCI3nRDR. SCI3nRDR cannot be written from the CPU.

When the data length is 7 bits, receive data is stored in bits 6 to 0 and bit 7 is fixed to 0 regardless of the SINV bit in SCI3nSCMR. When the data length is 8 bits, the received data are stored in bits 7 to 0.

20.3.4 SCI3nTDR — Transmit Data Register

SCI3nTDR is an 16-bit register to store transmit data. The value of SCI3nTDR after a reset is $FFFF_H$.

When SCI3nTSR empty is detected, the transmit data written to SCI3nTDR is transferred to SCI3nTSR and transmission starts. The double-buffered structure of SCI3nTDR and SCI3nTSR allows continuous serial transmission. If the next transmit data has been written to SCI3nTDR when one frame of data is sent, the transmit data is transferred to SCI3nTSR to continue transmission. SCI3nTDR can always be read and written by the CPU. Be sure to check that the TDRE flag in SCI3nSSR is set to 1 before writing transmit data to SCI3nTDR.

Write to bits 0 to 7, if you select the 8-bit data length. Write to bits 0 to 7 at the end when you write the transmit data to SCI3nTDR with setting 16-bit data length and dividing into bytes state of TE = 1.

20.3.5 SCI3nTSR — Transmit Shift Register

SCI3nTSR is a shift register to transmit serial data. To perform serial data transmission, transmit data written to SCI3nTDR is automatically transferred SCI3nTSR, and is then sent to the SCInTxD pin. SCI3nTSR cannot be directly accessed by the CPU.

20.3.6 SCI3nSMR — Serial Mode Register

SCI3nSMR is a register used to select the communication format and the clock source for the on-chip baud rate generator.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CM	CHR	PE	PM	STOP	MP	CKS1	CKS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. Writable only when TE = RE = 0.

Table 20.11 SCI3nSMR Register Contents (1/2)

Bit Position	Bit Name	Function
7	CM	Communication Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	Character Length (Valid only in asynchronous mode) This bit is used in combination with CHR1 bit in SCI3nSCMR. CHR1 CHR 0 0: Selects 16 bits as the data length. 0 1: Reserved 1 0: Selects 8 bits as the data length. 1 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) in SCI3nTDR is not transmitted in transmission. Data length of clocked synchronous mode is selected by CHR1 bit in SCI3nSCMR.
5	PE	Parity Enable (Valid only in asynchronous mode) When this bit is set to 1, a parity bit is added to transmit data and the parity bit is checked in reception. Regardless of the setting of this bit, no parity bit is added or checked in the multi-processor format.
4	PM	Parity Mode (Valid only when PE = 1 in asynchronous mode) 0: Selects even parity. 1: Selects odd parity. When even parity is set, parity bit is added so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit is added so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.
3	STOP	Stop Bit Length (Valid only in asynchronous mode) 0: 1 stop bit for transmission 1: 2 stop bits for transmission In reception, only the first stop bit is checked regardless of the setting of this bit. If the second stop bit is 0, it is treated as the start bit of the next transmission frame.
2	MP	Multi-Processor Mode (Valid only in asynchronous mode and CHR1 bit in SCI3nSCMR is 1) When this bit is set to 1, the multi-processor communication function is enabled. The PE and PM bits settings are invalid in multi-processor mode.

Table 20.11 SCI3nSMR Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	CKS1, CKS0	Clock Select 1, 0 These bits select the clock source for the on-chip baud rate generator. 00: PCLK clock (α = 0) 01: PCLK/4 clock (α = 1) 10: PCLK/16 clock (α = 2) 11: PCLK/64 clock (α = 3) For the relation between the setting of these bits and the baud rate, see **Section 20.3.11, SCI3nBRR — Bit Rate Register**. The character α is the decimal notation of the value of α in **Section 20.3.11, SCI3nBRR — Bit Rate Register**.

20.3.7 SCI3nSCR — Serial Control Register

SCI3nSCR is a register used for the transmission/reception control, interrupt control, and transmission/reception clock source selection listed below. For interrupt requests, see **Section 20.4.5, Interrupt Sources**.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W*1	R/W*1	R/W	R/W	R/W*2	R/W*2

Note 1. While the CM bit in SCI3nSMR is 1, a value of 1 can be written only when TE = 0 and RE = 0. After the TE or RE bit is set to 1, only 0 can be written to TE and RE. While the CM bit in SCI3nSMR is 0, writing is enabled at any timing.

Note 2. Writable only when TE = 0 and RE = 0. Also, writable at the same time when TE = 0 and RE = 0 are written.

Table 20.12 SCI3nSCR Register Contents (1/2)

Bit Position	Bit Name	Function
7	TIE	Transmit Interrupt Enable When this bit is set to 1, INTSCI3nTXI interrupt request is enabled. INTSCI3nTXI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0, or clearing the TIE bit.
6	RIE	Receive Interrupt Enable When this bit is set to 1, INTSCI3nRXI and INTSCI3nERI interrupt requests are enabled. INTSCI3nRXI and INTSCI3nERI interrupt requests can be cancelled by reading 1 from the RDRF, FER, PER, or ORER flag and then clearing the flag to 0, or clearing the RIE bit.
5	TE	Transmit Enable When this bit is set to 1, transmission is enabled. In this state, serial transmission is started by writing transmit data to SCI3nTDR and clearing the TDRE flag in SCI3nSSR to 0. Be sure to configure SCI3nSMR before setting the TE bit to 1 to determine the transmission format. When this bit is set to 0 to disable transmission, the TDRE flag in SCI3nSSR is fixed to 1.
4	RE	Receive Enable When this bit is set to 1, reception is enabled. In this state, serial reception is started by detecting a start bit in asynchronous mode or the input of synchronous clock in clock synchronous mode. Be sure to configure SCI3nSMR before setting the RE bit to 1 to determine the reception format. Even if reception is disabled by clearing this bit, the RDRF, FER, PER, or ORER flags are not affected and the previous value is retained.

Table 20.12 SCI3nSCR Register Contents (2/2)

Bit Position	Bit Name	Function
3	MPIE	Multiprocessor Interrupt Enable (Valid only when MP in SCI3nSMR = 1 in asynchronous mode) When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, this bit is automatically cleared to 0, and normal reception is resumed. For details, see **Section 20.4.2, Multi-Processor Communication Function**. When the receive data includes MPB = 0 in SCI3nSSR, the receive data is not transferred from SCI3nRSR to SCI3nRDR, a receive error is not detected, and setting the RDRF, FER, and ORER flags in SCI3nSSR to 1 is disabled. When the receive data includes MPB = 1 in SCI3nSSR, the MPB bit in SCI3nSSR is set to 1, the MPIE bit is automatically cleared to 0, the INTSCI3nRXI and INTSCI3nERI interrupt requests are enabled (if the RIE bit in SCI3nSCR is set to 1), and setting the FER and ORER flags to 1 is enabled.
2	TEIE	Transmit End Interrupt Enable When this bit is set to 1, INTSCI3nTEI interrupt request is enabled. INTSCI3nTEI interrupt request can be cancelled by reading 1 from the TDRE flag and then clearing the flag to 0 to clear the TEND flag to 0, or clearing the TEIE bit.
1, 0	CKE1, CKE0	Clock Enable 1, 0 These bits select the clock source and the SCInSCK pin function. For asynchronous mode 00: On-chip baud rate generator (The SCInSCK pin functions as an input/output port.) 01: On-chip baud rate generator (The clock with the same frequency as the bit rate is output from the SCInSCK pin.) 1X: Setting prohibited For clock synchronous mode 0X: Internal clock (The SCInSCK pin functions as a clock output pin.) 1X: External clock (The SCInSCK pin functions as a clock input pin.)

Note: X: Don’t care

Note: When writing to any bit other than the MPIE bit of this register, use a store instruction such that the value of the MPIE bit becomes 0.
Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the MPIE bit to 1.

20.3.8 SCI3nSSR — Serial Status Register

SCI3nSSR consists of SCI3 status flags and multi-processor transmit/receive bits.
The TDRE, RDRF, ORER, PER, and FER flags can be cleared only.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0010_H

Value after reset: 84_H

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset	1	0	0	0	0	1	0	0
R/W	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R/W

Note 1. Only 0 can be written to clear the flag.

Table 20.13 SCI3nSSR Register Contents (1/2)

Bit Position	Bit Name	Function
7	TDRE	Transmit Data Register Empty Indicates whether or not transmit data exists in SCI3nTDR. [Setting conditions] • When the TE bit in SCI3nSCR is 0 • When the data in SCI3nTDR has been transferred to SCI3nTSR so that new data can be written to SCI3nTDR [Clearing condition] • Writing 0 to TDRE after reading TDRE = 1 • When transmit data is written to SCI3nTDR while TE = 1
6	RDRF	Receive Data Register Full Indicates whether or not receive data exists in SCI3nRDR [Setting condition] • When reception finishes successfully and the receive data is transferred from SCI3nRSR to SCI3nRDR [Clearing conditions] • Writing 0 to RDRF after reading RDRF = 1 • When data is read from SCI3nRDR Even when the RE bit in SCI3nSCR is cleared, the RDRF flag is not affected and the previous value is retained. • Note that completing the reception of the next data with the RDRF flag set to 1 will cause an overrun error, resulting in the loss of the receive data.
5	ORER	Overrun Error Indicates that an overrun error has occurred during reception and the reception ended abnormally. [Setting condition] • When the next data is received while RDRF = 1 In SCI3nRDR, data received prior to the overrun error is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be continued. In clock synchronous mode, serial transmission also cannot be continued. [Clearing condition] • Writing 0 to ORER after reading ORER = 1 Even when the RE bit in SCI3nSCR is cleared, the ORER flag is not affected and the previous value is retained.

Table 20.13 SCI3nSSR Register Contents (2/2)

Bit Position	Bit Name	Function
4	FER	Framing Error Indicates that a framing error has occurred during reception in asynchronous mode and the reception ended abnormally. [Setting condition] • When the stop bit is 0 In 2-stop-bit mode, only whether the first stop bit is 1 is checked but the second stop bit is not checked. Although the receive data that existed when a framing error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the FER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. [Clearing condition] • Writing 0 to FER after reading FER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the FER flag is not affected and the previous value is retained.
3	PER	Parity Error Indicates that a parity error has occurred during reception in asynchronous mode and the reception ended abnormally. [Setting condition] • Although the receive data that existed when a parity error has occurred is transferred to SCI3nRDR, the RDRF flag is not set. In addition, while the PER flag is set to 1, the subsequent receive data is not transferred to SCI3nRDR. [Clearing condition] • Writing 0 to PER after reading PER = 1 Even when the RE bit in SCI3nSCR is cleared to 0, the PER flag is not affected and the previous value is retained.
2	TEND	Transmit End Indicates that a transmission is completed. [Setting conditions] • When the TE bit in SCI3nSCR is 0 • When the TDRE flag is 1 while the last bit of a transmit character is being transmitted [Clearing condition] • Writing 0 to the TDRE flag after reading TDRE = 1 • When transmit data is written to SCI3nTDR while TE = 1
1	MPB	Multi-processor Bit Holds the value of the multi-processor bit in the receive frame.
0	MPBT	Multi-processor Bit Transfer Sets the value of the multi-processor bit to be added to the transmit frame.

20.3.9 SCI3nSCMR — Serial Transfer Format Register

SCI3nSCMR is a register to select the communication format which can be commonly configured for both asynchronous mode and clock synchronous mode.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0018_H

Value after reset: $F2_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	CHR1	SDIR	SINV	ASTLS	—
Value after reset	1	1	1	1	0	0	1	0
R/W	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R

Note 1. Writable only when TE = 0 and RE = 0.

Table 20.14 SCI3nSCMR Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is read. When writing, write the value after reset.
4	CHR1	Character Length 1 (Valid in asynchronous mode and clocked synchronous mode) • Asynchronous mode CHR1 is used in combination with CHR bit in SCI3nSMR. Refer to the **Section 20.3.6, SCI3nSMR — Serial Mode Register** value. • Clocked synchronous mode 1: Selects 8 bits as the data length 0: Selects 16 bits as the data length
3	SDIR	Serial Data Transfer Direction (Valid in asynchronous mode and clock synchronous mode) This bit is used to select the direction of serial/parallel conversion. 0: Transfer with LSB-first 1: Transfer with MSB-first This bit is valid only when the transfer format is 8-bit and 16-bit data. For 7-bit data, LSB-first transfer is used.
2	SINV	Serial Data Invert (Valid in asynchronous mode and clock synchronous mode) This bit is used to invert the transfer data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SCI3nSMR. 0: SCI3nTDR data is transmitted as it is, and receive data is stored in SCI3nRDR as it is. 1: SCI3nTDR data is inverted and transmitted, and receive data is inverted and stored in SCI3nRDR.
1	ASTLS	Asynchronous Mode Start Bit Level Detection Select (Valid only in asynchronous mode) 0: Falling edge of RxD pin input 1: Low level of RxD pin input
0	Reserved	When read, the value after reset is read. When writing, write the value after reset.

20.3.10 SCI3nSEMR — Serial Extended Mode Register

SCI3nSEMR is a register to select a 1-bit period.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + $001C_H$

Value after reset: 04_H

Bit	7	6	5	4	3	2	1	0
	BRME	MDDRS	—	BGDM	ABCS	RXDMON	—	—
Value after reset	0	0	0	0	0	1	0	0
R/W	R/W*1	R/W*1	R	R/W*1	R/W*1	R	R	R

Note 1. Writable only when TE = 0 and RE = 0.

Table 20.15 SCI3nSEMR Register Contents

Bit Position	Bit Name	Function
7	BRME	Bit Rate Modulation Enable When this bit is set to 1, the bit rate modulation function is enabled.
6	MDDRS	Modulation Duty Register Select This bit is used to select an accessible register. 0: SCI3nBRR is accessible. 1: SCI3nMDDR is accessible.
5	Reserved	When read, the value after reset is read. When writing, write the value after reset.
4	BGDM	Baud Rate Generator Double-Speed Mode Select (Valid only when asynchronous mode and CKE1 bit in SCI3nSCR is 0) This bit selects output clock period of baud rate generator. 0: Output clock is normal frequency from baud rate generator 1: Output clock is twice frequency from baud rate generator (Double speed operation)
3	ABCS	Asynchronous Reference Clock Select (Valid only in asynchronous mode) This bit is used to select the reference clock for 1-bit period. 0: Operates on the reference clock with a frequency of 16 times the transfer rate. 1: Operates on the reference clock with a frequency of 8 times the transfer rate (double-speed operation).
2	RXDMON	Serial Input Data Monitor This bit indicates the SCInRxD pin state. 0: SCInRxD pin state is low level. 1: SCInRxD pin state is high level.
1, 0	Reserved	When read, the value after reset is read. When writing, write the value after reset.

20.3.11 SCI3nBRR — Bit Rate Register

SCI3nBRR is an 8-bit register to adjust the bit rate. Since each SCI3 channel has an independent baud rate generator, different bit rates can be set for each channel. **Table 20.17** shows the relationship between the setting (N) in SCI3nBRR and the bit rate (B) in normal asynchronous mode, clock synchronous mode . The value of SCI3nBRR after reset is FF_H. SCI3nBRR is allocated at the same address as SCI3nMDDR and is selected when MDDRS in SCI3nSEMR is 0. This register is writable only when TE = 0 and RE = 0.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0004_H

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	BRR							
Value after reset	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.16 SCI3nBRR Register Contents

Bit Position	Bit Name	Function
7 to 0	BRR	Baud rate generator setting (0 ≤ N ≤ 255)

Table 20.17 Relationship between Setting N in SCI3nBRR and Bit Rate B

	SEMR Setting			
Mode	ABCS bit	BGDM bit	Bit Rate	Mean Error
Asynchronous	0	0	$B = \frac{PCLK \times 10^6}{64 \times 2^{2\alpha-1} \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2\alpha-1} \times (N+1)} - 1\right\} \times 100$
	0	1	$B = \frac{PCLK \times 10^6}{32 \times 2^{2\alpha-1} \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2\alpha-1} \times (N+1)} - 1\right\} \times 100$
	1	0		
	1	1	$B = \frac{PCLK \times 10^6}{16 \times 2^{2\alpha-1} \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2\alpha-1} \times (N+1)} - 1\right\} \times 100$
Clock synchronous	—		$B = \frac{PCLK \times 10^6}{8 \times 2^{2\alpha-1} \times (N+1)}$	—

Note: B: Bit rate (bps)
N: SCI3nBRR setting value for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
α: Determined by the SCI3nSMR setting value as shown in the following table.

SCI3nSMR Setting Value		
CKS1	CKS0	α
0	0	0
0	1	1
1	0	2
1	1	3

Table 20.18 lists sample N settings of the SCI3nBRR register in asynchronous mode. **Table 20.19** shows the maximum configurable bit rates.

Table 20.18 Examples of BRR Settings for Bit Rates (Asynchronous Mode) (80 MHz)

PCLK (MHz)	α	N	Baud Rate		
			SCI3nSEMR.ABCS = 0, SCI3nSEMR.BGDM = 0	SCI3nSEMR.ABCS = 0, SCI3nSEMR.BGDM = 1 or SCI3nSEMR.ABCS = 1, SCI3nSEMR.BGDM = 0	SCI3nSEMR.ABCS = 1, SCI3nSEMR.BGDM = 1
80	0	0	2.500 Mbps	5.000 Mbps	10.000 Mbps
80	0	1	1.250 Mbps	2.500 Mbps	5.000 Mbps
80	0	2	0.833 Mbps	1.667 Mbps	3.334 Mbps
80	0	3	0.625 Mbps	1.250 Mbps	2.500 Mbps
80	0	4	0.500 Mbps	1.000 Mbps	2.000 Mbps
80	0	5	0.417 Mbps	0.833 Mbps	1.667 Mbps
80	0	6	0.357 Mbps	0.714 Mbps	1.428 Mbps
80	0	7	0.313 Mbps	0.625 Mbps	1.250 Mbps
:					
80	1	0	0.625 Mbps	1.250 Mbps	2.500 Mbps
80	1	1	0.313 Mbps	0.625 Mbps	1.250 Mbps
80	1	2	0.208 Mbps	0.417 Mbps	0.834 Mbps
80	1	3	0.156 Mbps	0.313 Mbps	0.626 Mbps
80	1	4	0.125 Mbps	0.250 Mbps	0.500 Mbps
80	1	5	0.104 Mbps	0.208 Mbps	0.416 Mbps
80	1	6	0.089 Mbps	0.179 Mbps	0.358 Mbps
80	1	7	0.078 Mbps	0.156 Mbps	0.312 Mbps
:					
80	3	242	160.751 bps	321.502 bps	643.004 bps
80	3	243	160.092 bps	320.184 bps	640.368 bps
80	3	244	159.439 bps	318.878 bps	637.756 bps
80	3	245	158.791 bps	317.581 bps	635.162 bps
80	3	246	158.148 bps	316.296 bps	632.592 bps
80	3	247	157.510 bps	315.020 bps	630.040 bps
80	3	248	156.878 bps	313.755 bps	627.510 bps
80	3	249	156.250 bps	312.500 bps	625.000 bps
80	3	250	155.627 bps	311.255 bps	622.510 bps
80	3	251	155.010 bps	310.020 bps	620.040 bps
80	3	252	154.397 bps	308.794 bps	617.588 bps
80	3	253	153.789 bps	307.579 bps	615.158 bps
80	3	254	153.186 bps	306.373 bps	612.746 bps
80	3	255	152.588 bps	305.176 bps	610.352 bps

Table 20.19 Maximum Bit Rate (Asynchronous Mode)

PCLK (MHz)	Setting				Maximum Serial Clock Frequency
	ABCS Setting	BGDM setting	α	N	
80	0	0	0	0	2.5 Mbps
80	1	0	0	0	5 Mbps
80	0	1	0	0	5 Mbps
80	1	1	0	0	10 Mbps

Table 20.20 lists sample N settings of the SCI3nBRR register in clock synchronous mode.
Table 20.21 shows the maximum configurable bit rates.

Table 20.20 Examples of Bit Rate Settings for Clock Synchronous Mode (Master Mode)

α	N	Baud Rate PCLK = 80 MHz
0	0	Setting prohibited
0	1	Setting prohibited
0	2	Setting prohibited
0	3	5.000 Mbps
0	4	4.000 Mbps
0	5	3.333 Mbps
0	6	2.857 Mbps
0	7	2.500 Mbps
	:	
1	0	5.000 Mbps
1	1	2.500 Mbps
1	2	1.667 Mbps
1	3	1.250 Mbps
1	4	1.000 Mbps
1	5	0.833 Mbps
1	6	0.714 Mbps
1	7	0.625 Mbps
	:	
3	242	1286.008 bps
3	243	1280.738 bps
3	244	1275.510 bps
3	245	1270.325 bps
3	246	1265.182 bps
3	247	1260.081 bps
3	248	1255.020 bps
3	249	1250.000 bps
3	250	1245.020 bps
3	251	1240.079 bps
3	252	1235.178 bps
3	253	1230.315 bps
3	254	1225.490 bps
3	255	1220.703 bps

Table 20.21 Maximum Bit Rate (Clock Synchronous Mode) (Master Mode)

PCLK (MHz)	α	N	Maximum Serial Clock Frequency
80	0	3	5 Mbps

20.3.12 SCI3nMDDR — Modulation Duty Register

SCI3nMDDR is a register to correct the bit rate adjusted by SCI3nBRR. The value of SCI3nMDDR after a reset is FF_H. When the BRME bit in SCI3nSEMR is set to 1, the bit rate generated by the on-chip baud rate generator is corrected to SCI3nMDDR/256 on average. **Table 20.23** shows the relationship between the SCI3nMDDR setting and the bit rate B. SCI3nMDDR is allocated at the same address as SCI3nBRR and is selected when MDDRS in SCI3nSEMR is 1. This register is writable only when TE = 0 and RE = 0. Bit 7 is fixed to 1.

Access: This register can be read or written in 8-bit units.

Address: <SCI3n_base> + 0004_H

Value after reset: FF_H

Bit	7	6	5	4	3	2	1	0
	MDDR							
Value after reset	1	1	1	1	1	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.22 SCI3nMDDR Register Contents

Bit Position	Bit Name	Function
7 to 0	MDDR	Baud rate generator setting value (128 ≤ MDDR ≤ 255)

Table 20.23 Relationship between SCI3nMDDR Setting Value and Bit Rate B when Bit Rate Modulation Function Is Used

Mode	SEMR Setting		Bit Rate	Mean Error
	ABCS bit	BGDM bit		
Asynchronous	0	0	$B = \frac{PCLK \times 10^6}{64 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)} - 1\right\} \times 100$
	0	1	$B = \frac{PCLK \times 10^6}{32 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)} - 1\right\} \times 100$
	1	0		
	1	1	$B = \frac{PCLK \times 10^6}{16 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)} - 1\right\} \times 100$
Clock synchronous	—	—	$B = \frac{PCLK \times 10^6}{8 \times 2^{2\alpha-1} \times (256/MDDR) \times (N+1)}$	—

Note: B: Bit rate (bps)
N: SCI3nBRR setting value of baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
α: See **Table 20.17, Relationship between Setting N in SCI3nBRR and Bit Rate B**.
SCI3nMDDR: SCI3nMDDR setting (128 ≤ SCI3nMDDR ≤ 255)

20.4 Operation

20.4.1 Operation in Asynchronous Mode

Figure 20.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communication line is usually held in the mark state (high level). The SCI3 monitors the communication line, and when the SCI3 detects the space state (low level), it recognizes a start bit and starts serial communications. Inside the SCI3, the transmitter and the receiver are independent, enabling full-duplex communications. Both the transmitter and the receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

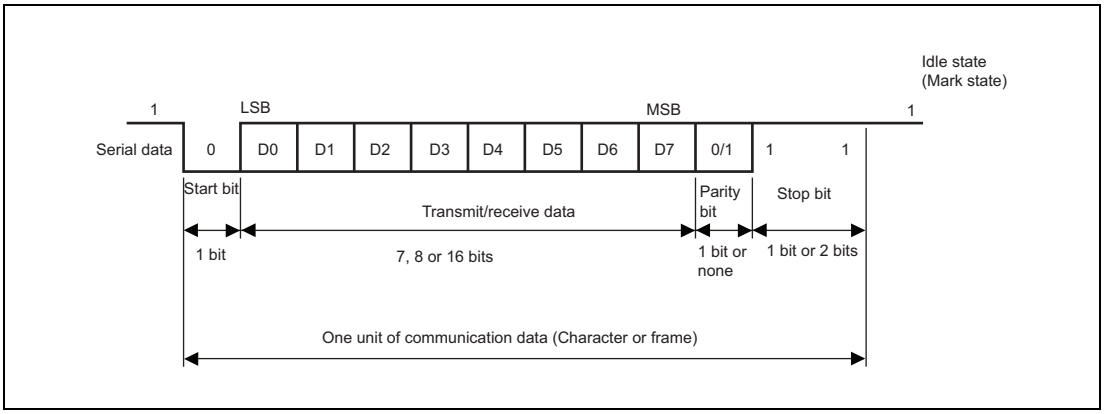


Figure 20.2 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, Two Stop Bits)

20.4.1.1 Transmission/Reception Format

Table 20.24 lists the transmission/reception formats that can be configured in asynchronous mode. Any of 16 transmission/reception formats can be selected according to the SCI3nSMR setting. For details of the multi-processor bit, see **Section 20.4.2, Multi-Processor Communication Function**.

Table 20.24 Serial Transmission/Reception Formats (Asynchronous Mode) (1/2)

SCI3nSCMR Setting	SCI3nSMR Setting				Serial Transmission/Reception Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
0	0	0	0	0	S 16-bit data STOP
0	0	0	0	1	S 16-bit data STOP STOP
0	0	1	0	0	S 16-bit data P STOP
0	0	1	0	1	S 16-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP

Table 20.24 Serial Transmission/Reception Formats (Asynchronous Mode) (2/2)

SCI3nSCMR Setting	SCI3nSMR Setting				Serial Transmission/Reception Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1	1	—	1	1	S | 7-bit data | MPB | STOP | STOP

Note: S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

20.4.1.2 Receive Data Sampling Timing and Reception Margin

In asynchronous mode, the SCI3 operates on a reference clock with a frequency of 16 times (8 times when ABCS bit in SCI3nSEMR = 1) the bit rate. In reception, the SCI3 samples the falling edge of the beginning of the start bit (low level) using the reference clock and performs internal synchronization.

As shown in **Figure 20.3**, data is latched at the middle of each bit by sampling receive data at the rising edge of the eighth pulse (fourth pulse when ABCS bit in SCI3nSEMR = 1) of the reference clock.

Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100[\%] \ldots \text{formula(1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SCI3nSEMR = 0, N = 8 when ABCS = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 20)
F: Absolute value of clock frequency deviation

Assuming that F (absolute value of clock frequency deviation) = 0, D (duty cycle of clock) = 0.5, and N = 16 in formula (1), the reception margin is obtained by the formula below.

$$M = \left\{0.5 - \frac{1}{(2 \times 16)}\right\} \times 100[\%] = 46.875\%$$

However, this is only the calculated value, and a margin of 20% to 30% should be allowed in system design.

When the bit rate modulation function is used, the reference clock frequency is corrected on average.

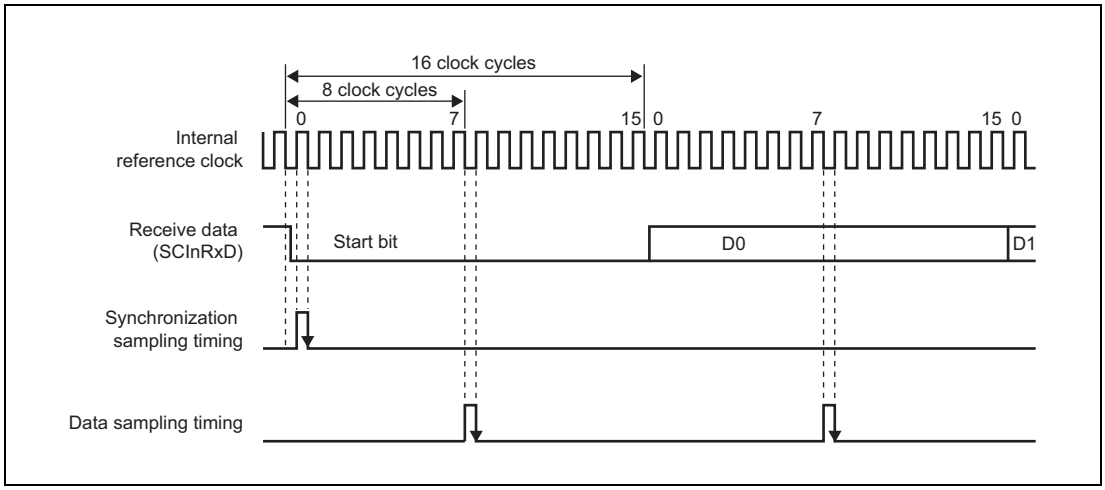


Figure 20.3 Receive Data Sampling Timing in Asynchronous Mode (when ABCS bit in SCI3nSEMR = 0)

20.4.1.3 Clock

An internal clock generated by the on-chip baud rate generator can be selected as the SCI3's transmission/reception clock according to the settings of the CM bit in SCI3nSMR and the CKE1 and CKE0 bits in SCI3nSCR. When the SCI3 is operated on an internal clock, the clock can be output from the SCInSCK pin.

For details of clock synchronous mode, see **Section 20.4.3, Operation in Clock Synchronous Mode**.

In asynchronous mode, the frequency of the clock output is equal to the bit rate and the phase is such that the rising edge of the clock comes at the center of the transmit data, as shown in **Figure 20.4**.

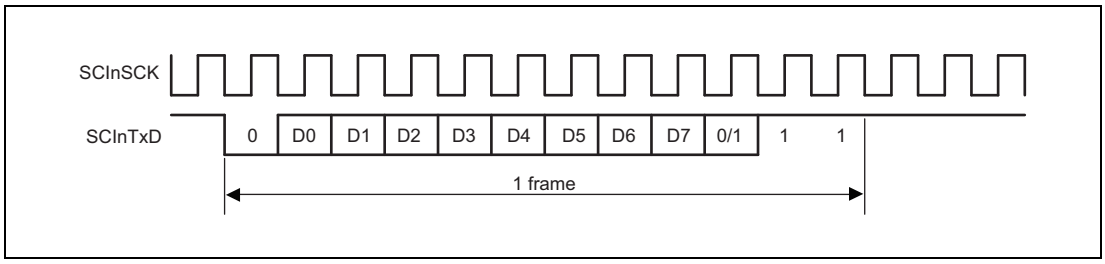


Figure 20.4 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode)

20.4.1.4 Double-Speed Operation

In addition to the operation described in **Section 20.4.1.3, Clock**, double-speed operation is enabled by the setting of the ABCS bit in SCI3nSEMR.

In double-speed operation, the same operation on a clock with a frequency of 16 times the bit rate in normal operation can be conducted on a clock with a frequency of 8 times the bit rate, meaning that the SCI3 operates on a double transfer rate using the same reference clock.

In addition, the basic clock cycle is halved if BGDM bit in SCI3nSEMR is set to 1, therefore SCI3 operates at a bit rate twice as fast as when BGDM bit is set to 0.

Double-speed operation by the ABCS bit and double-speed operation by the BGDM bit can be used in combination when CKE1 bit in SCR is set to 0 and on-chip baud rate generator is selected. By setting the ABCS bit = 1 and BGDM bit = 1, SCI3 operates at a bit rate four times faster than when ABCS bit = 0 and BGDM bit = 0 .

20.4.1.5 SCI3 Initialization (Asynchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR and then initialize the SCI3 according to the sample flowchart in **Figure 20.5**. Before changing the operating mode or transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

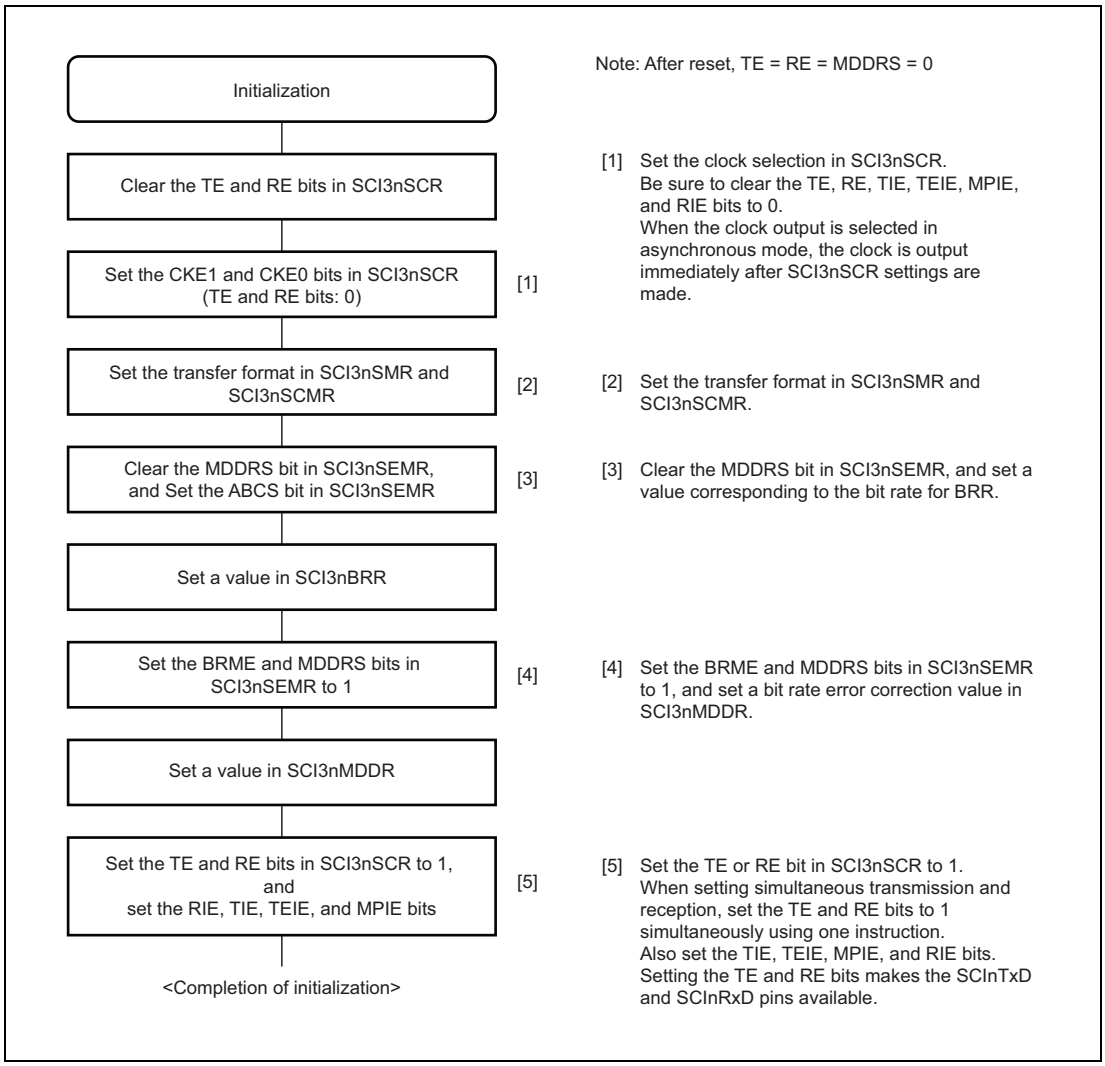


Figure 20.5 Sample Flowchart for SCI3 Initialization

20.4.1.6 Serial Data Transmission (Asynchronous Mode)

Figure 20.6 shows an example of operation for data transmission in asynchronous mode. In data transmission, the SCI3 operates as described below.

1. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR. When writing transmit data to SCI3nTDR at a trigger of INTSCI3nTXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a INTSCI3nTXI interrupt request for starting data transfer.
2. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR and the TDRE flag is set to 1. If the TIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this INTSCI3nTXI interrupt processing routine before transmission of the previously transferred data is completed. When an INTSCI3nTEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
3. Data is sent from the SCInTxD pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The TDRE flag is checked when the stop bit is output.
5. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR. After the stop bit has been sent, transmission of the next frame starts.
6. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1, the stop bit is sent, and then 1 is output to enter the mark state. If the TEIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTEI interrupt request is generated.

Figure 20.7 shows a sample flowchart for data transmission. **Figure 20.8** shows a sample flowchart for stopping the SCI3 after data transmission.

Note about the operation when transmission is enabled in asynchronous mode:

When the setting of the TE bit is changed from 0 to 1, a high level (preamble) is output for one frame. If transmit data is written to SCI3nTDR while the preamble is being output, that data will be transferred from SCI3nTDR to SCI3nTSR after preamble output is complete.

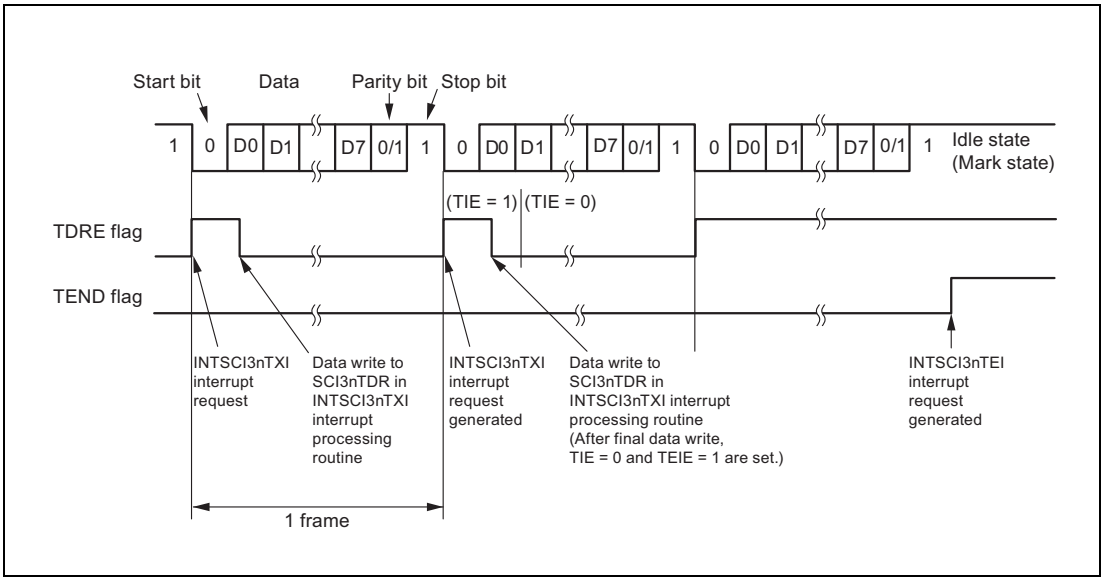


Figure 20.6 Example of Operation for Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

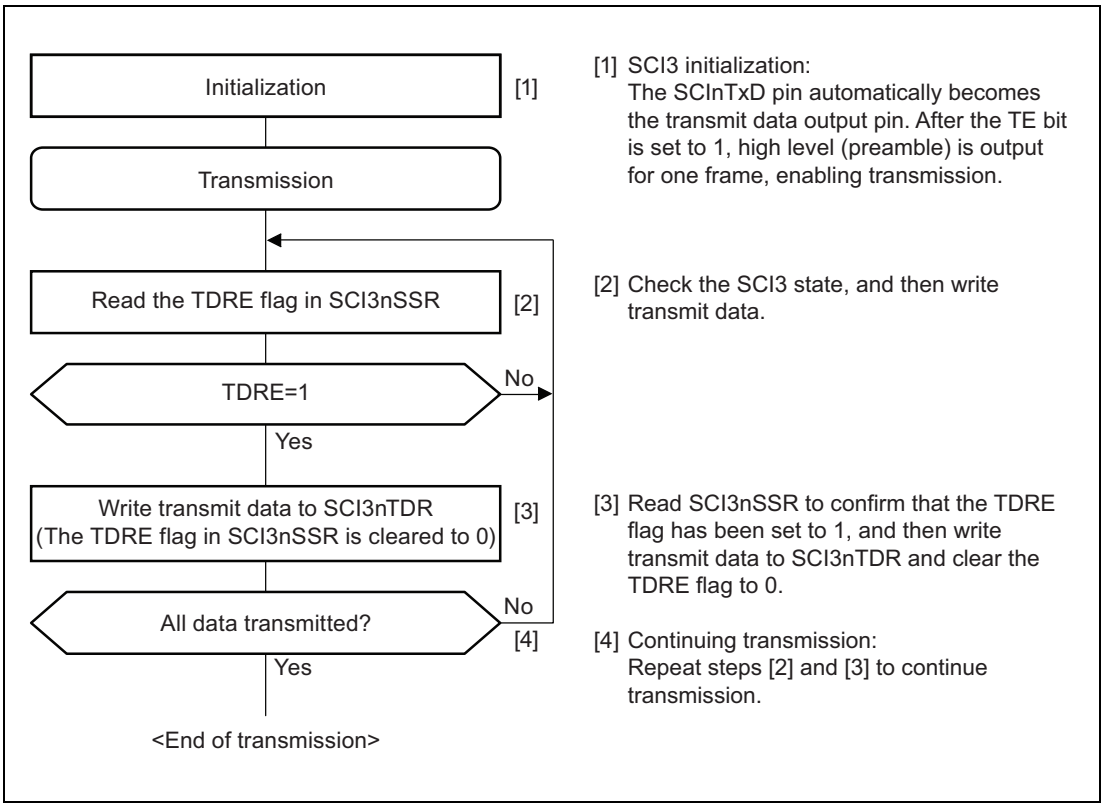


Figure 20.7 Example of Serial Transmission Flowchart

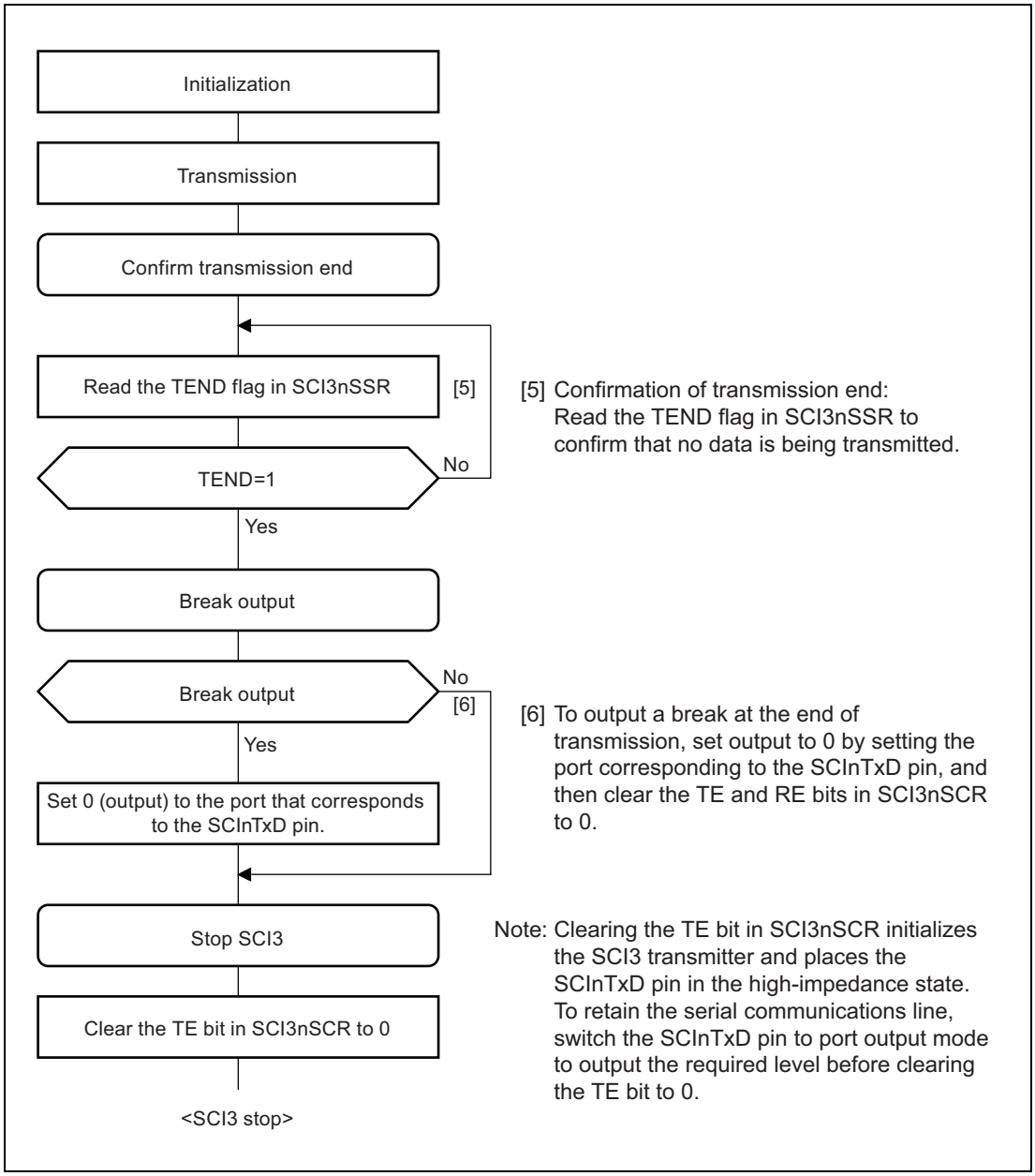


Figure 20.8 Example Flowchart for Stopping the SCI3 after Serial Transmission

20.4.1.7 Serial Data Reception (Asynchronous Mode)

Figure 20.9 shows an example of the operation for data reception in asynchronous mode. In data reception, the SCI3 operates as described below.

1. SCI3 monitors the communications line and upon detection of a start bit, it performs internal synchronization, stores receive data in SCI3nRSR, and checks the parity bit and the stop bit.
2. When an overrun error occurs (the next data has been received with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When a parity error is detected, the PER flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated.
4. When a framing error (when the stop bit is 0) is detected, the FER flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated.
5. When reception finishes successfully, the RDRF flag in SCI3nSSR is set to 1 and receive data is transferred to SCI3nRDR. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this INTSCI3nRXI interrupt processing routine before reception of the next receive data is completed. Reading SCI3nRDR automatically clears the RDRF flag to 0.

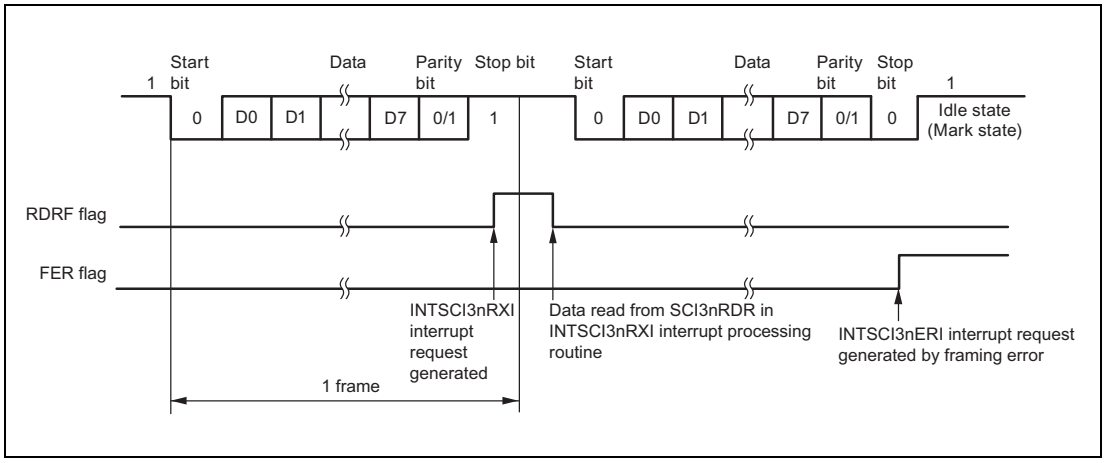


Figure 20.9 Example of Operation for Reception in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Table 20.25 lists the states of the SCI3nSSR status flags and receive data handling when a receive error is detected. When a receive error is detected, the RDRF flag retains the status before receiving the data. Subsequent data reception is disabled while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF flags before continuing data reception. **Figure 20.10** shows a sample flowchart for data reception.

Table 20.25 SCI3nSSR Status Flags and Receive Data Handling

SCI3nSSR Status Flags					
RDRF*1	ORER	FER	PER	Receive Data	Receive Status
1	0	0	0	Transferred to SCI3nRDR	Successful reception
0	0	1	0	Transferred to SCI3nRDR	Framing error
0	0	0	1	Transferred to SCI3nRDR	Parity error
0	0	1	1	Transferred to SCI3nRDR	Framing error + parity error
1*	1	0	0	Lost	Overrun error
1*	1	1	0	Lost	Overrun error + framing error
1*	1	0	1	Lost	Overrun error + parity error
1*	1	1	1	Lost	Overrun error + framing error + parity error

Note 1. In the case of an overrun error, the RDRF flag retains the state before the data reception.

Note: "+" indicates that two or more receive statuses occur simultaneously in a single reception operation.

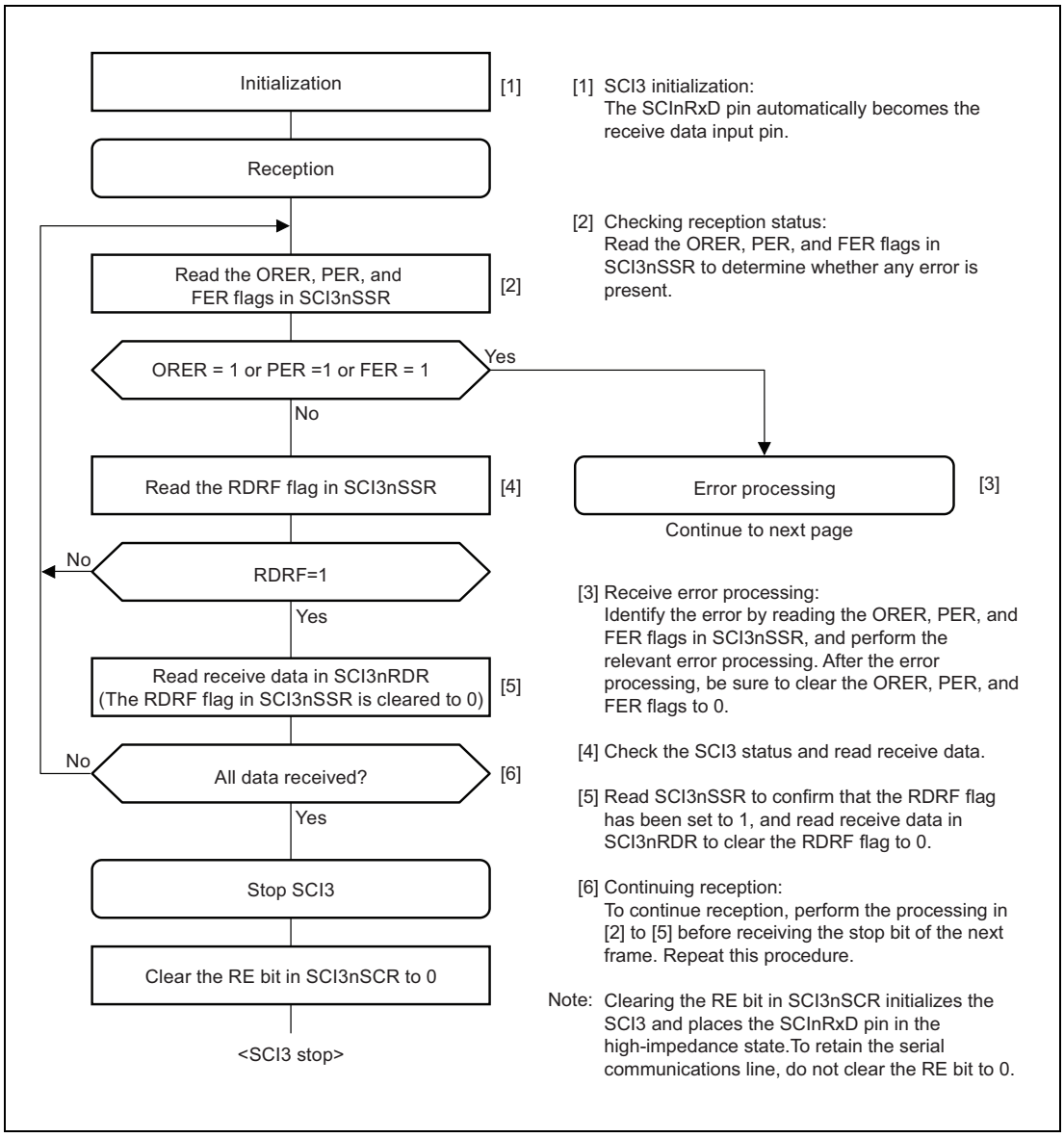


Figure 20.10 Example of Serial Reception Flowchart (1)

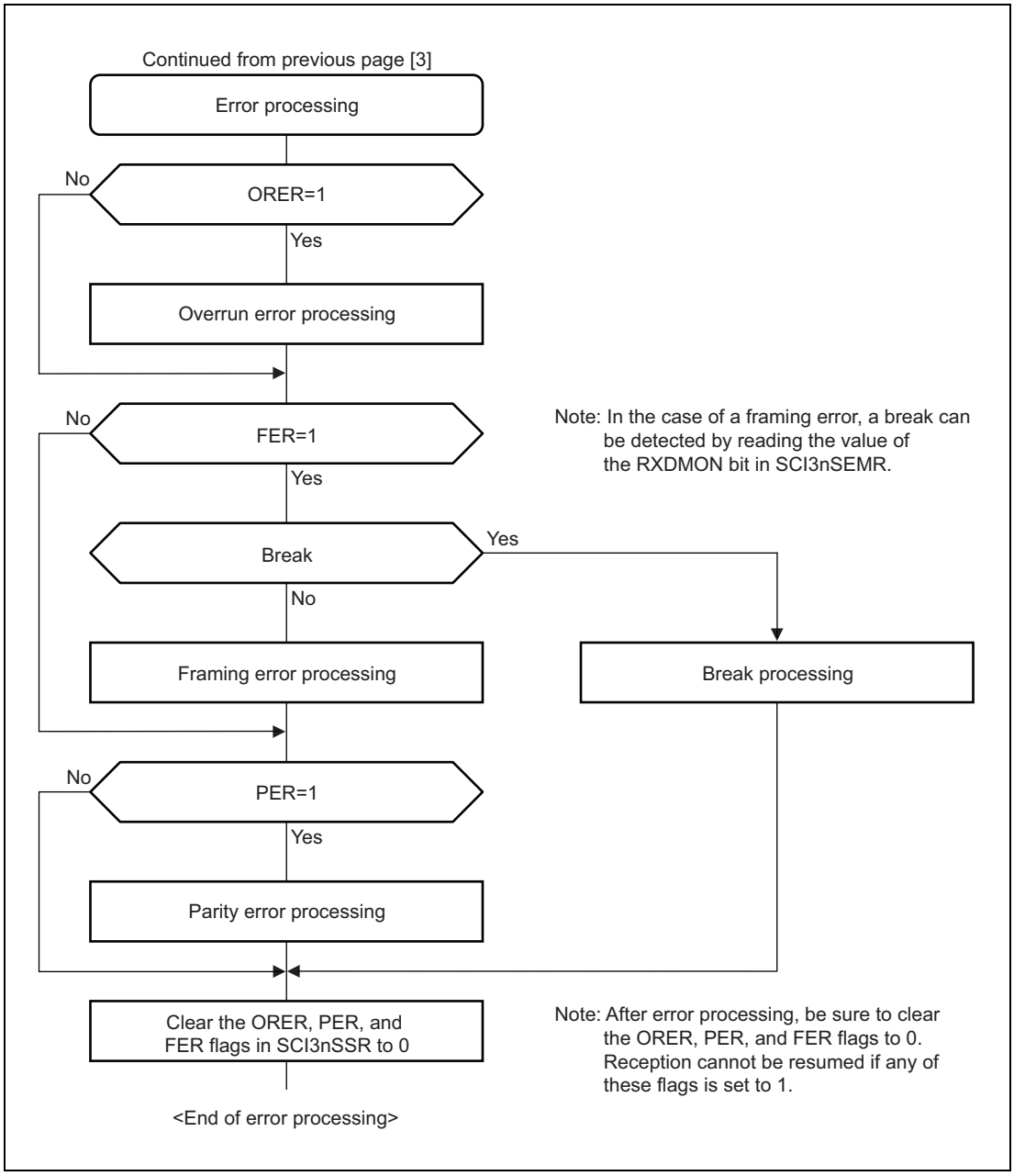


Figure 20.11 Example of Serial Reception Flowchart (2)

20.4.2 Multi-Processor Communication Function

20.4.2.1 Overview and Sample Connection

Using the multi-processor communication function allows data transmission and reception by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish the ID transmission cycle from the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle. When the multi-processor bit is set to 0, it indicates the data transmission cycle. **Figure 20.12** shows an example of communication between processors by using the multi-processor format. First, a transmitting station sends communication data in which the multi-processor bit (= 1) is added to the ID code of the receiving station. Next, the transmitting station sends communication data in which the multi-processor bit (= 0) is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself. When these IDs match, the receiving station receives communication data that is subsequently transmitted. If these IDs do not match, the receiving station skips communication data until it receives communication data in which the multi-processor bit is set to 1.

To support this function, the SCI3 provides the MPIE bit in SCI3nSCR. When the MPIE bit is set to 1, transfer of receive data from SCI3nRSR to SCI3nRDR, detection of a receive error, and setting the RDRF, FER, and ORER flags in SCI3nSSR are disabled until data in which the multi-processor bit is set to 1 is received. Upon receiving a character in which the multi-processor bit is set to 1, the MPB bit in SCI3nSSR is set to 1 and the MPIE bit is automatically cleared to 0, thus returning to a normal reception operation. When the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. While the MPIE bit is cleared to 0, reception operation is conducted regardless of the multi-processor bit value. The multi-processor bit is stored in the MPB bit in SCI3nSSR.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock that is used for multi-processor communications is also the same as the clock used in the normal asynchronous mode.

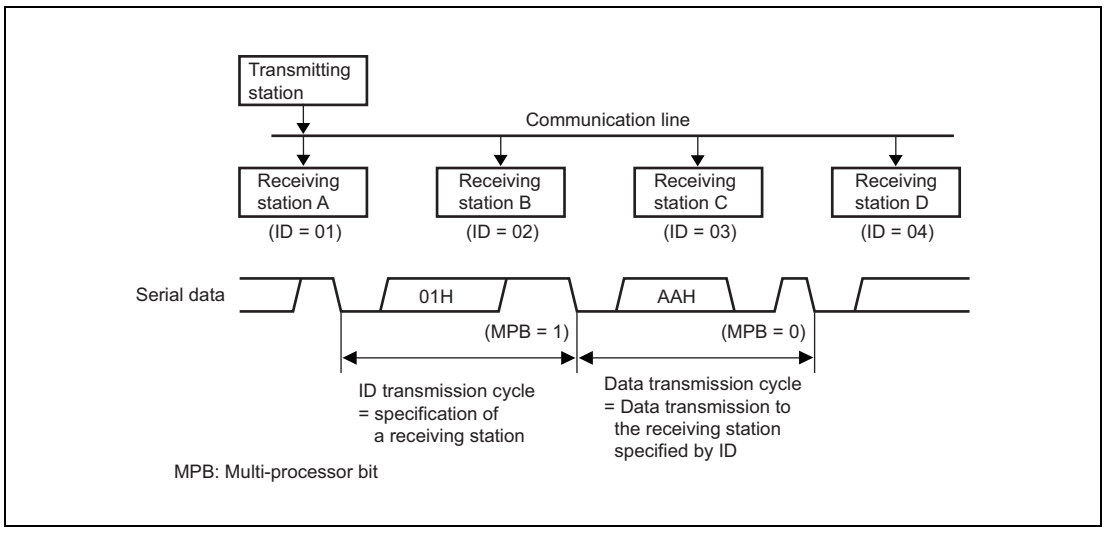


Figure 20.12 Example of Communication Using the Multi-Processor Format (Example of Transmission of Data AA_H to Receiving Station A)

20.4.2.2 Multi-Processor Serial Data Transmission

Figure 20.13 shows a sample flowchart of multi-processor data processing. In the ID transmission cycle, send the ID with the MPBT bit in SCI3nSSR set to 1. In the data transmission cycle, send data with the MPBT bit in SCI3nSSR cleared to 0. Other operations are the same as operations in asynchronous mode.

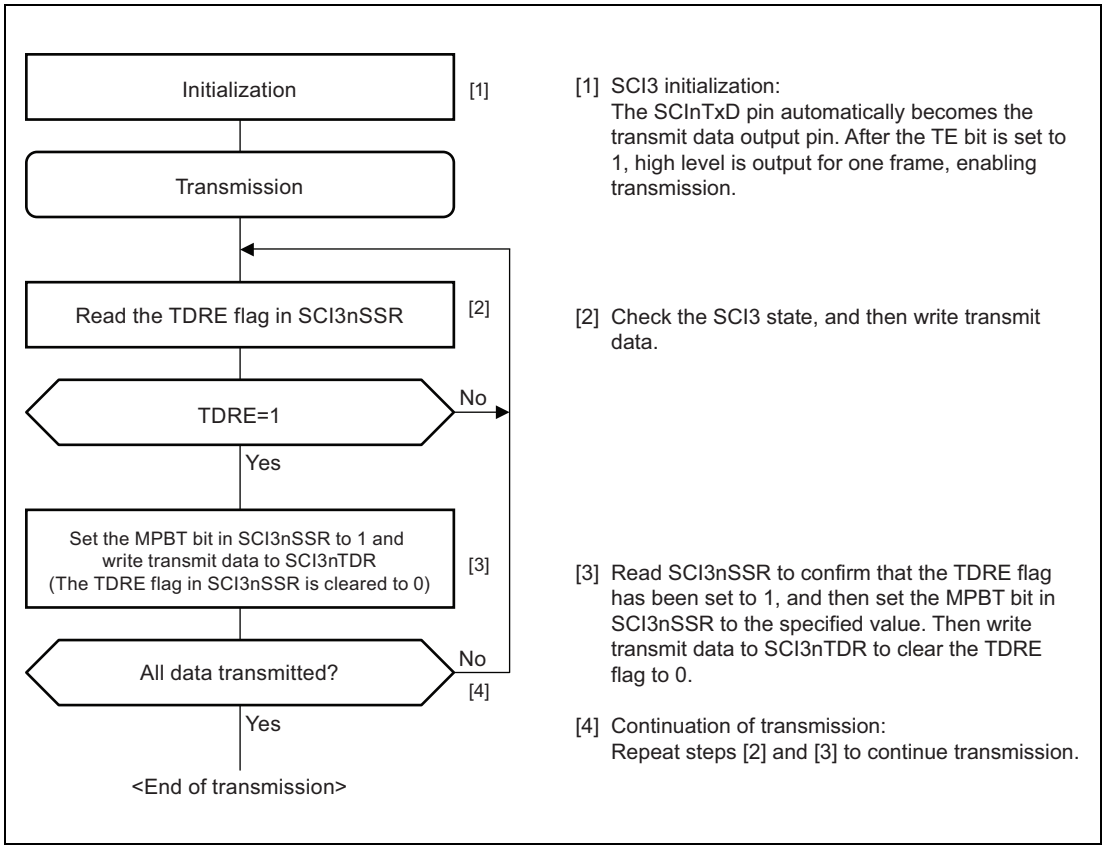


Figure 20.13 Example of Multi-Processor Serial Transmission Flowchart

20.4.2.3 Multi-Processor Serial Data Reception

Figure 20.15 shows a sample flowchart of multi-processor data reception. When the MPIE bit in SCI3nSCR is set to 1, reading the communication data is skipped until communication data in which the multi-processor bit is set to 1 is received. When communication data in which the multi-processor bit is set to 1 is received, the receive data is transferred to SCI3nRDR. At this time, an INTSCI3nRXI interrupt request is generated. Other operations are the same as operations in asynchronous mode. **Figure 20.14** shows an example of operation for reception.

CAUTION

Do not write data to SCI3nSCR when communication data in which the multi-processor bit is set to 1 is received. The MPIE bit may not become the desired state.

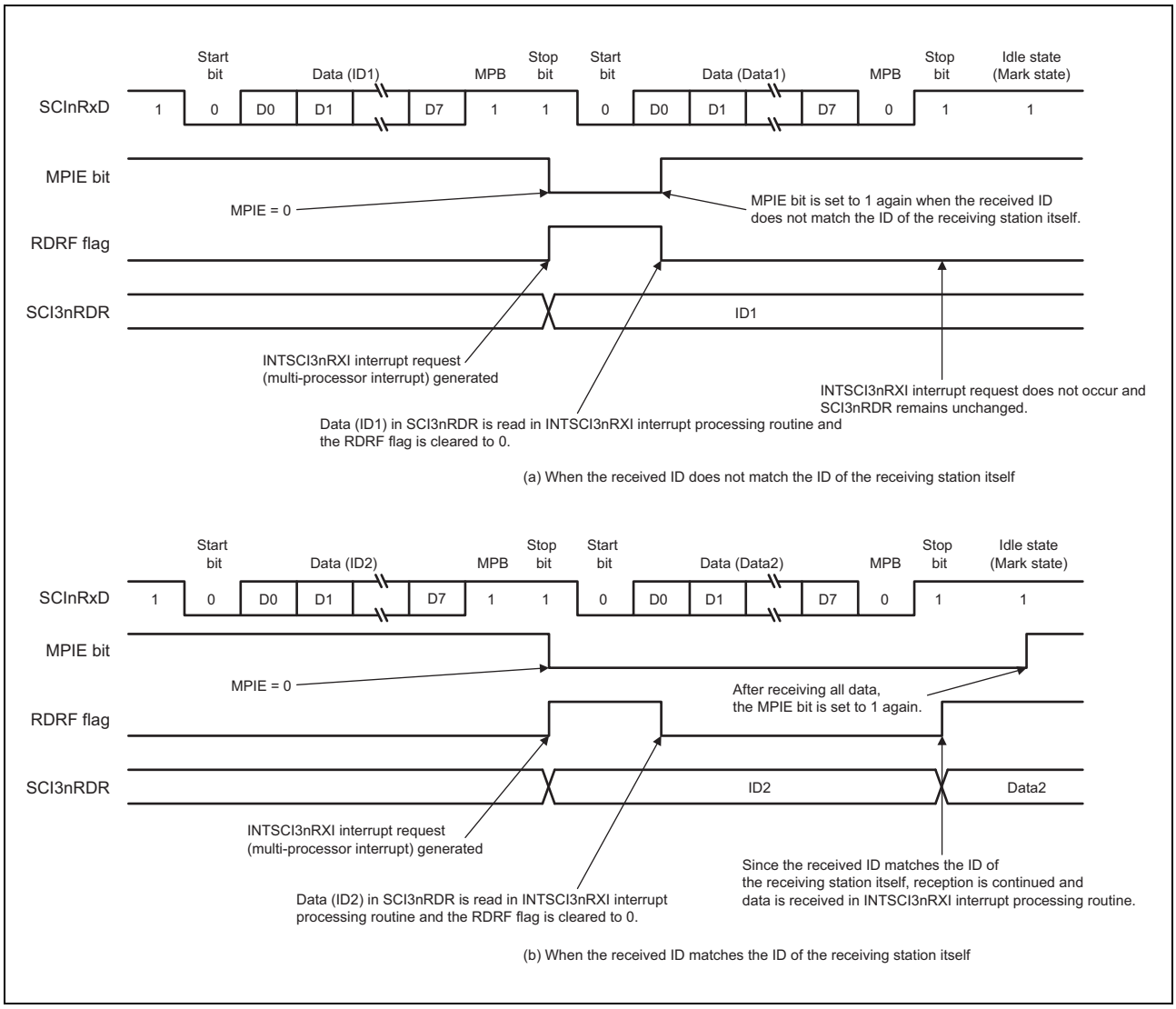


Figure 20.14 Example of SCI3 Reception (8-Bit Data, Multi-Processor Bit, One Stop Bit)

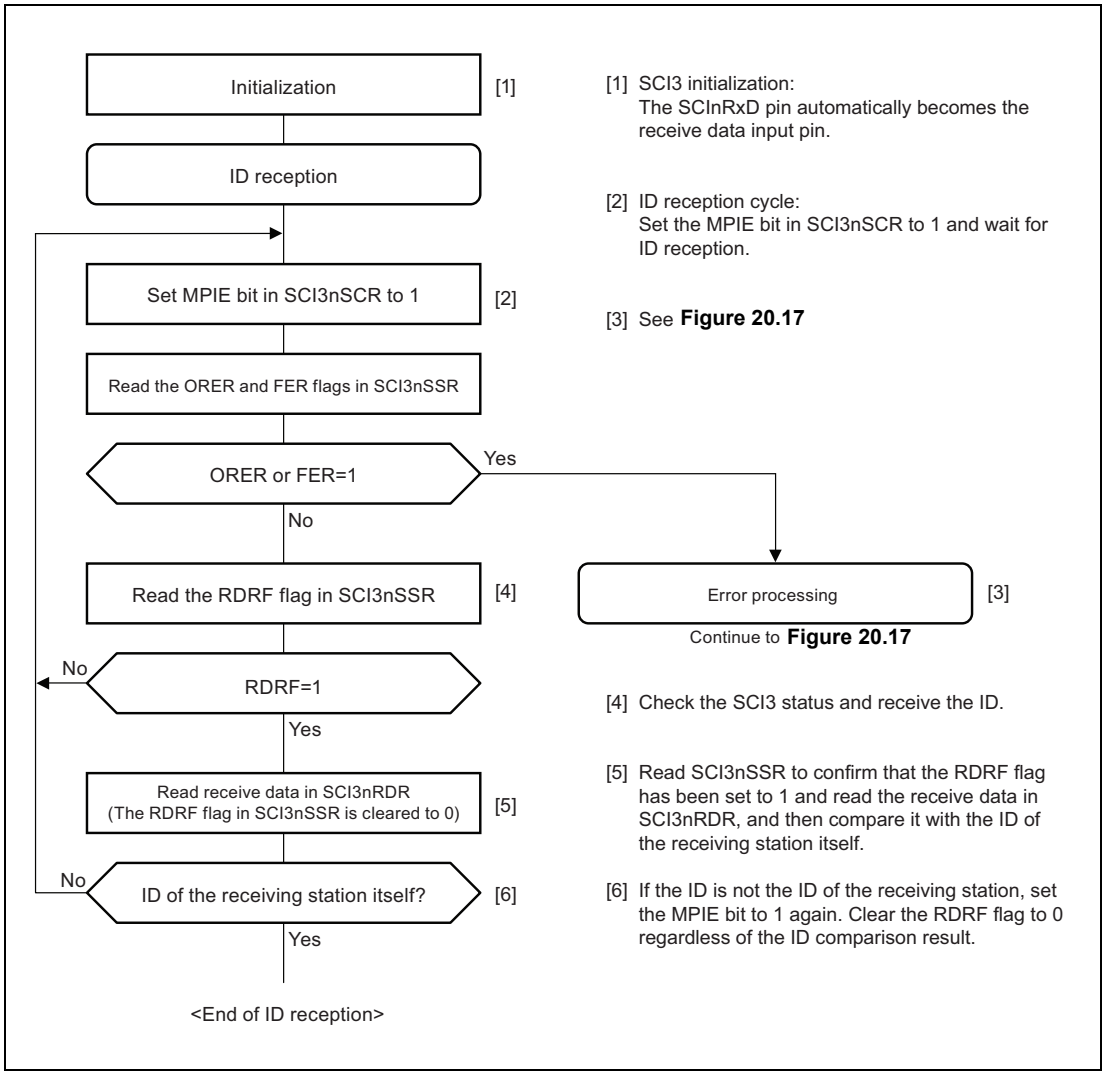


Figure 20.15 Example of Multi-Processor Serial Reception Flowchart (1)

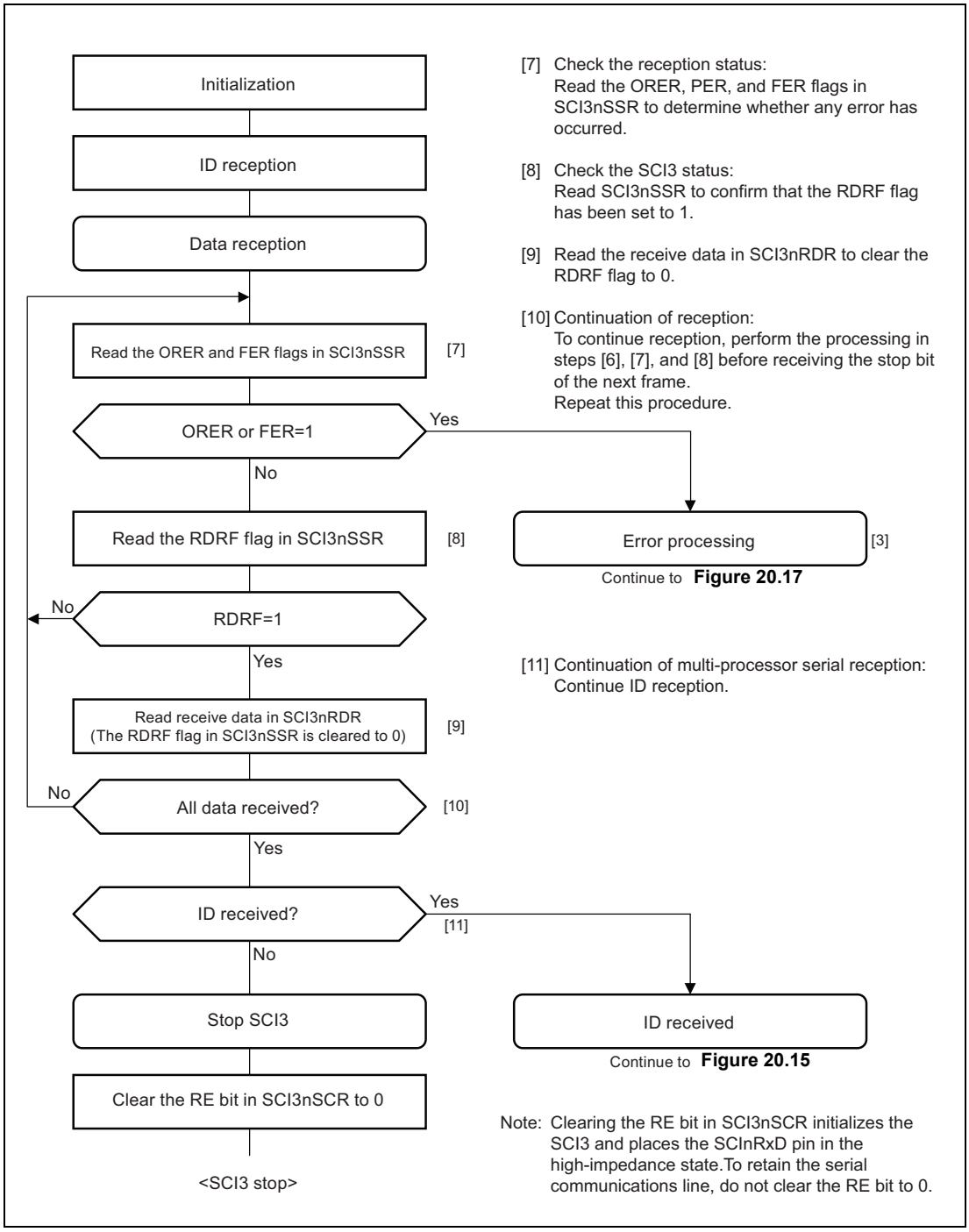


Figure 20.16 Example of Multi-Processor Serial Reception Flowchart (2)

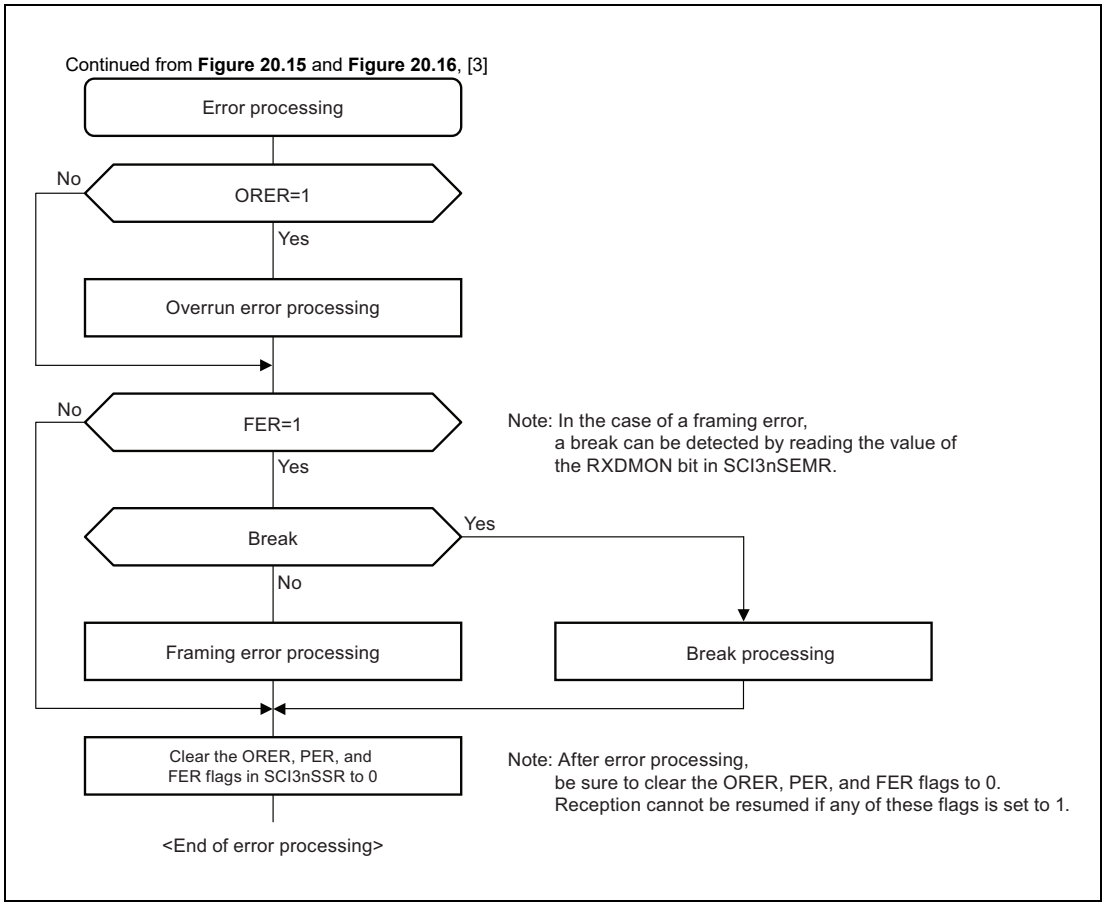
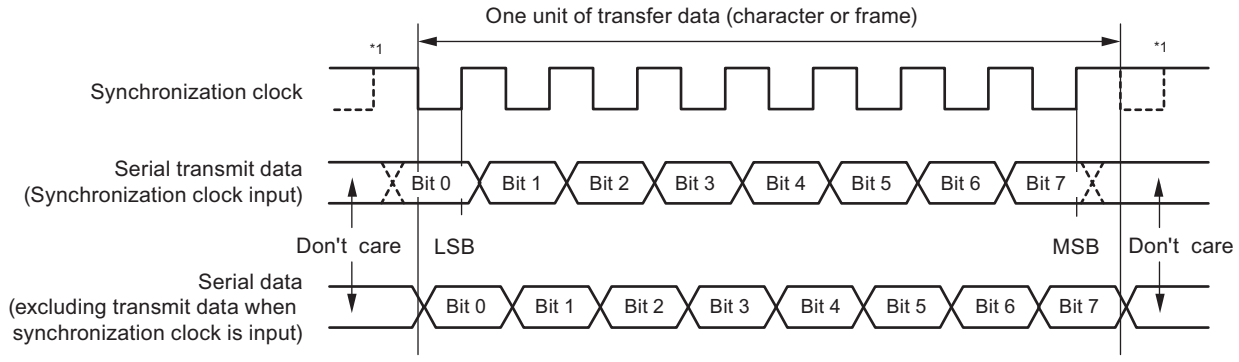


Figure 20.17 Example of Multi-Processor Serial Reception Flowchart (3)

20.4.3 Operation in Clock Synchronous Mode

Figure 20.18 shows the data format for clock synchronous serial data communication. In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8- or 16-bit data. In data transmission with the synchronization clock output, the SCI3 outputs data from one falling edge to the next falling edge of the synchronization clock. In data transmission with the synchronization clock input, the SCI3 outputs the first data (bit 0) after starting transfer immediately after clearing the TDRE bit in SCI3nSSR to 0, and then outputs the next bit data after 2 to 3 PCLK clock cycles from the rising edge of the synchronization clock. In data reception, the SCI3 receives data in synchronization with the rising edge of the synchronization clock. After 8- or 16-bit data is output, the communication line holds the last-bit output state. In clock synchronous mode, neither parity bit nor multi-processor bit can be added. The transmitter and the receiver are independent in the SCI3, enabling full-duplex communication by using a common clock. Both the transmitter and the receiver have a double-buffered structure so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.



Note 1: Holds a high level except during continuous transmission/reception.

Figure 20.18 Data Format in Clock Synchronous Mode (LSB-First)

20.4.3.1 Clock

An internal clock generated by the on-chip baud rate generator or an external synchronous clock that is input from the SCInSCK pin can be selected by the setting of the CKE1 and CKE0 bits in SCI3nSCR. When operating the SCI3 on the internal clock, a synchronous clock is output from the SCInSCK pin. Eight or sixteen pulses of the synchronous clock are output during transfer of one character, and the clock is held high while no data is transferred.

20.4.3.2 SCI3 Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, clear the TE and RE bits in SCI3nSCR to 0 and then initialize the SCI3 according to the sample flowchart in **Figure 20.19**. To switch the operation between transmission, reception, and transmission/reception, clear the TE and RE bits to 0 and then set these bits to the desired value. Before changing the transfer format, be sure to clear the TE and RE bits to 0. Note that clearing the TE bit to 0 sets the TDRE flag to 1, but clearing the RE bit to 0 initializes neither the RDRF, PER, FER, and ORER flags nor SCI3nRDR.

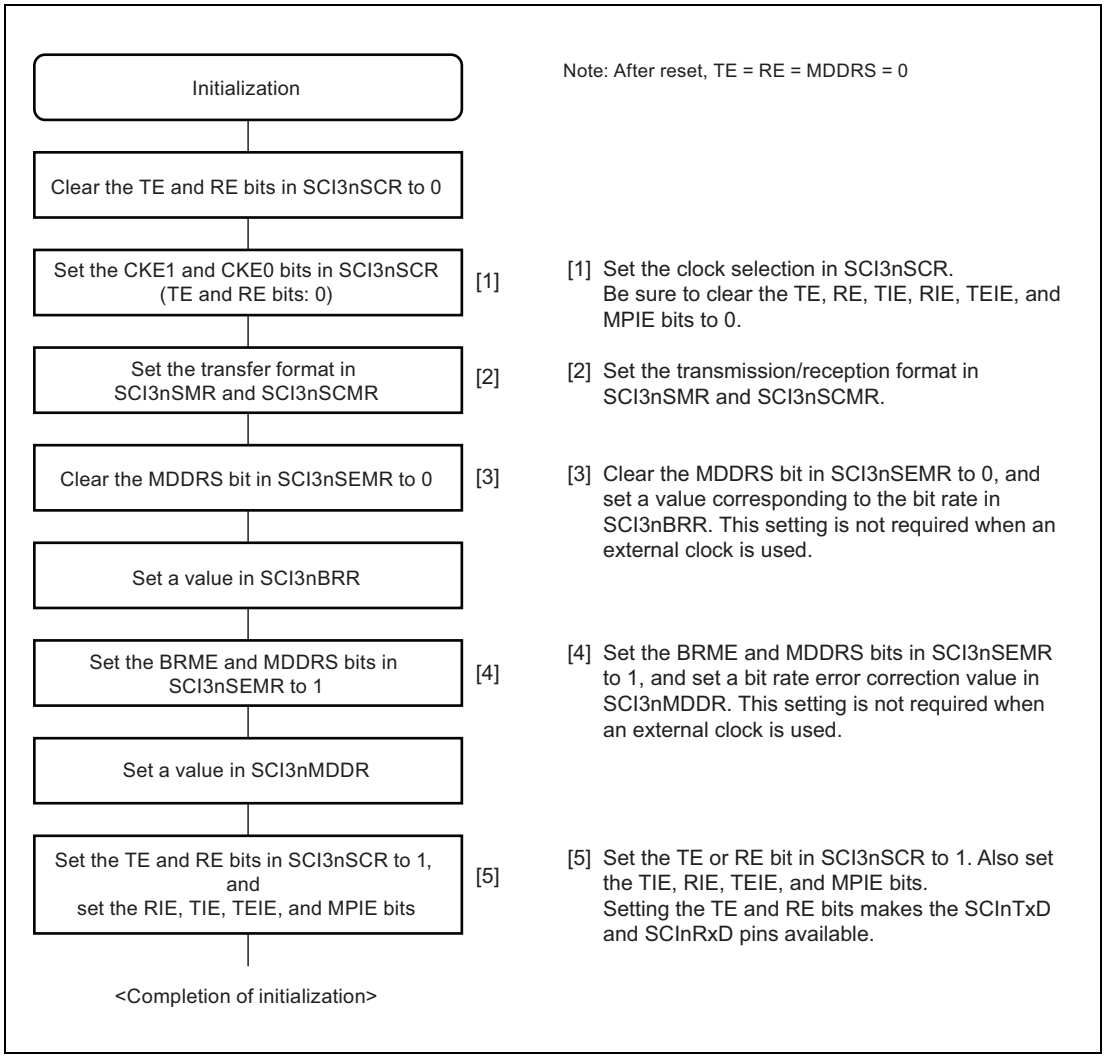


Figure 20.19 Example of SCI3 Initialization Flowchart

20.4.3.3 Serial Data Transmission (Clock Synchronous Mode)

Figure 20.20 shows an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the SCI3 operates as described below.

1. When transmit data is written to SCI3nTDR, the TDRE flag is automatically cleared to 0. The SCI3 monitors the TDRE flag in SCI3nSSR. When the flag is cleared, the SCI3 recognizes that data has been written to SCI3nTDR and transfers data from SCI3nTDR to SCI3nTSR, starting output of the first bit when the synchronization clock is input. To write transmit data to SCI3nTDR at a trigger of INTSCI3nTXI interrupt request, set the TIE bit to 1 and then set the TE bit to 1 or set both TIE and TE bits simultaneously with one instruction to generate a INTSCI3nTXI interrupt request for starting data transfer.
2. Transmission starts after data is transferred from SCI3nTDR to SCI3nTSR, and the TDRE flag is set to 1. When the TIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to SCI3nTDR in this INTSCI3nTXI interrupt processing routine before the transmission of the previously transferred data is completed. When a INTSCI3nTEI interrupt request is used, the TIE bit is cleared to 0 after the last transmit data has been written to SCI3nTDR, and the TEIE bit is set to 1.
3. 8-bit or 16-bit data is output from the SCInTxD pin in synchronization with the output clock (when clock output mode has been specified) or in synchronization with the input clock (when external clock has been specified).
4. The SCI3 checks the TDRE flag when the last bit is output.
5. When the TDRE flag is 0, the next transmit data is transferred from SCI3nTDR to SCI3nTSR, and serial transmission of the next frame starts.
6. When the TDRE flag is 1, the TEND flag in SCI3nSSR is set to 1 and the SCInTxD pin retains the output state of the last bit. If the TEIE bit in SCI3nSCR is set to 1 at this time, a INTSCI3nTEI interrupt request is generated. The SCInSCK pin is held high.

Figure 20.21 shows a sample flowchart of serial data transmission. Also, **Figure 20.22** shows a sample flowchart for stopping the SCI3 after data transmission. Transmission will not start even if the TDRE flag is cleared while a receive error flag (ORER) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

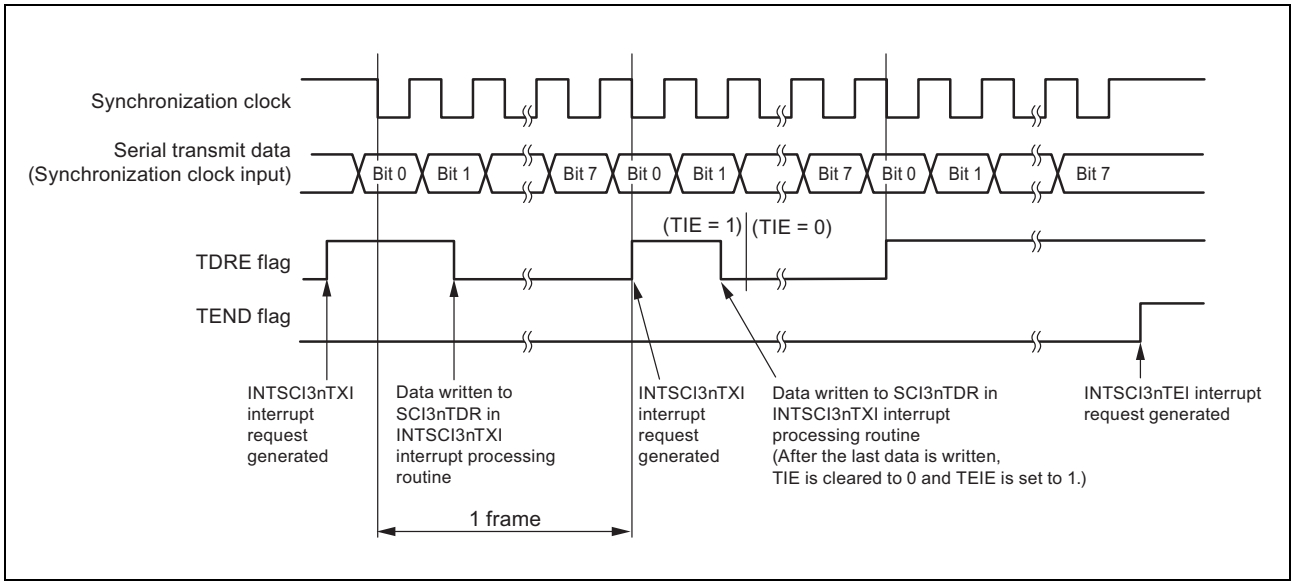


Figure 20.20 Example of Operation for Transmission in Clock Synchronous Mode

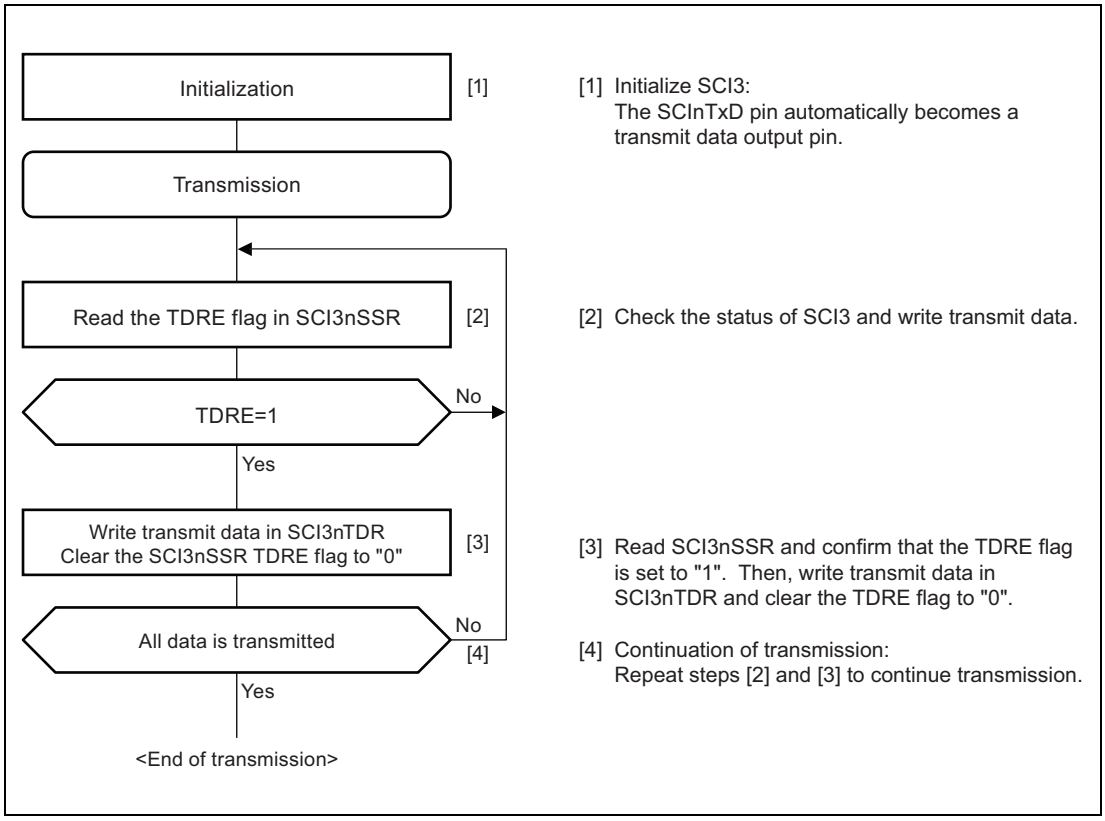


Figure 20.21 Example of Serial Transmission Flowchart

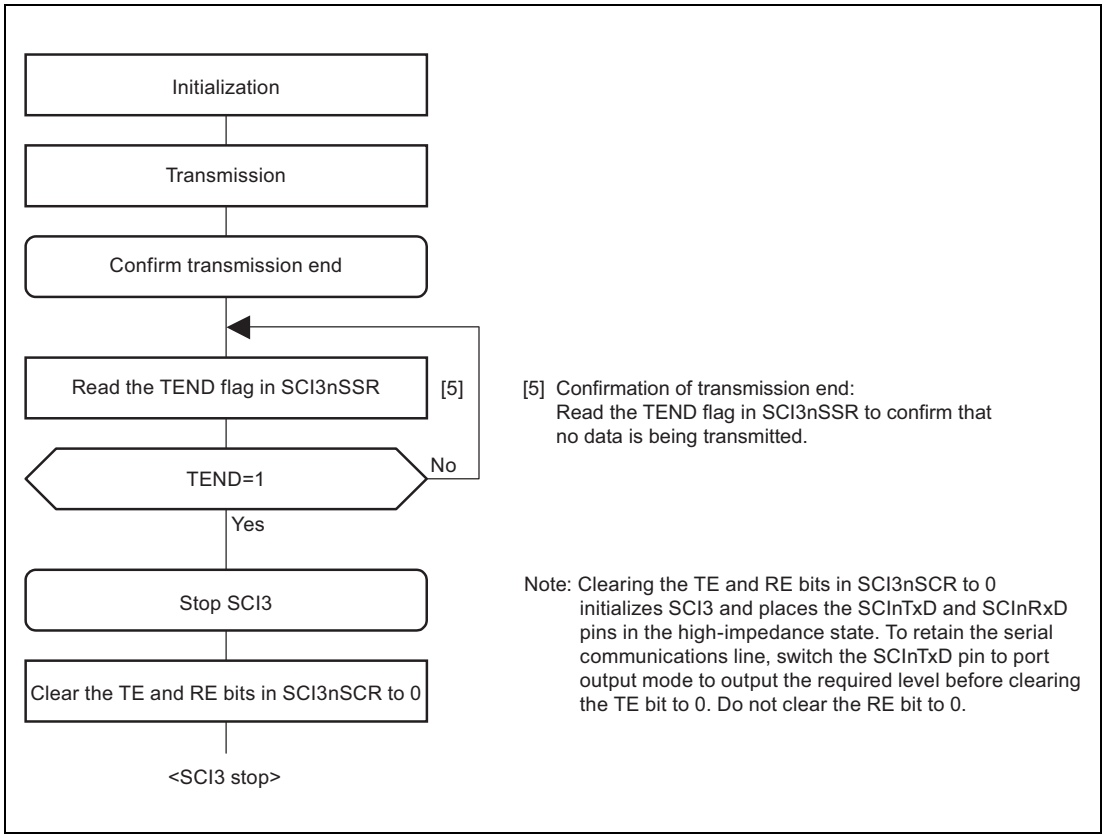


Figure 20.22 Example Flowchart for Stopping the SCI3 after Serial Transmission

20.4.3.4 Serial Data Reception (Clock Synchronous Mode)

Figure 20.23 shows an example of SCI3 operation for serial reception in clock synchronous mode. In serial data reception, the SCI3 operates as described below.

1. The SCI3 performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores receive data in SCI3nRSR.
2. When an overrun error occurs (the reception of the next data is completed with the RDRF flag in SCI3nSSR set to 1), the ORER flag in SCI3nSSR is set to 1. If the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nERI interrupt request is generated. Receive data is not transferred to SCI3nRDR. The RDRF flag retains the state of being set to 1.
3. When data has been successfully received, the RDRF flag in SCI3nSSR is set to 1 and the receive data is transferred to SCI3nRDR. When the RIE bit in SCI3nSCR is set to 1 at this time, an INTSCI3nRXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to SCI3nRDR in this INTSCI3nRXI interrupt processing routine before reception of the next data is completed. Reading SCI3nRDR automatically clears the RDRF flag to 0.

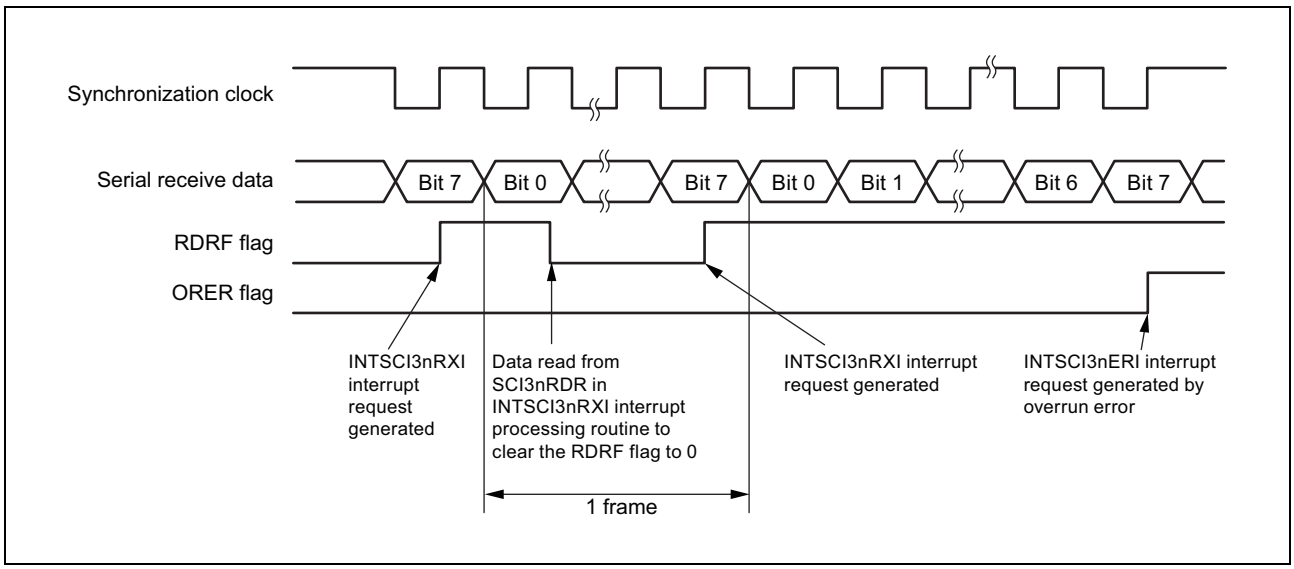


Figure 20.23 Example of SCI3 Operation for Reception

Subsequent transmission and reception are disabled with a receive error flag set to 1. Therefore, be sure to clear the ORER, FER, PER, and RDRF flags to 0 before continuing reception. **Figure 20.24** shows an example of flowchart for data reception.

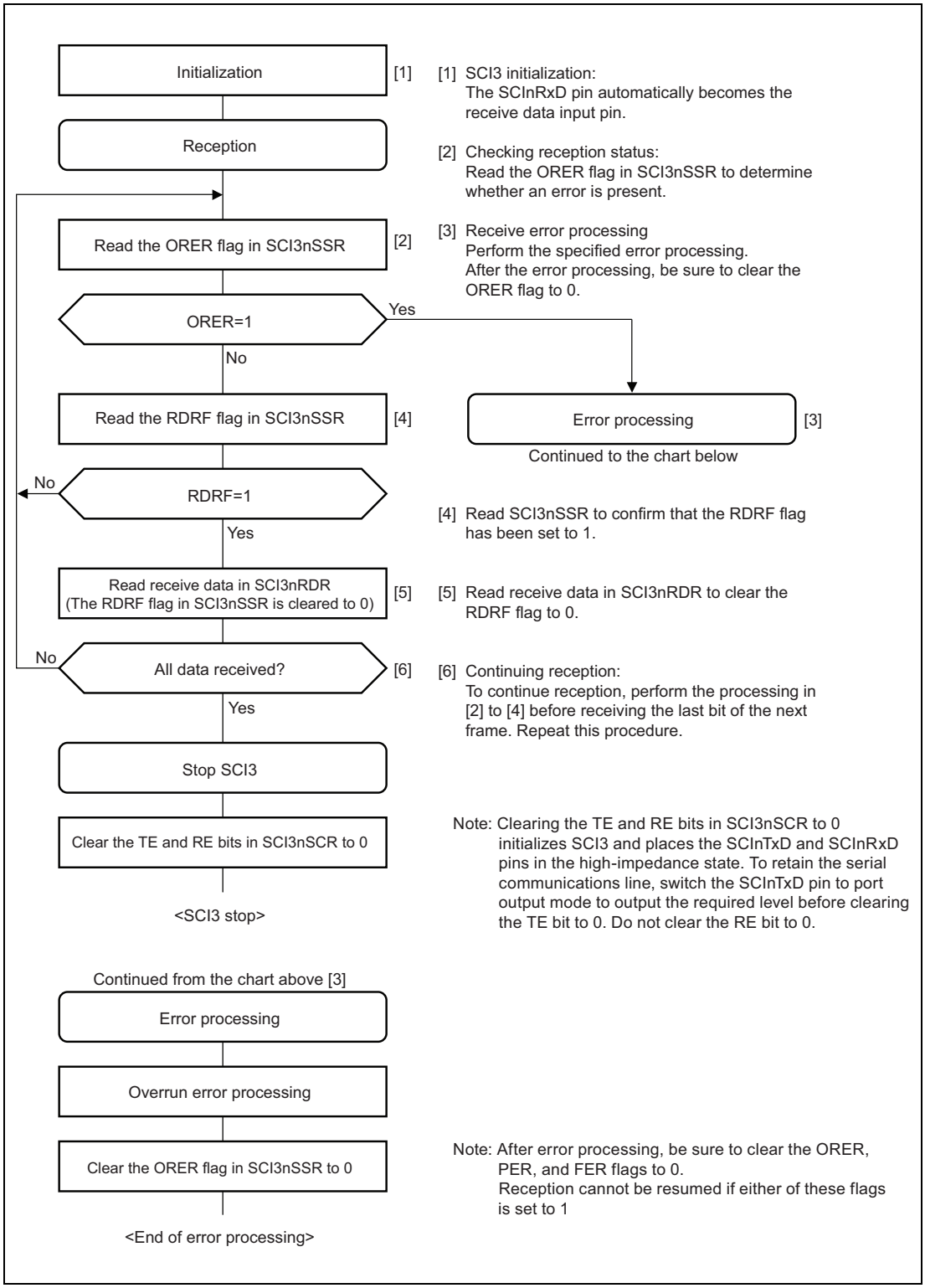


Figure 20.24 Example of Serial Reception Flowchart

20.4.3.5 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 20.25 shows a sample flowchart for simultaneous data transmit and receive operations. After the SCI3 is initialized, perform the following procedure for simultaneous data transmit and receive operations.

1. To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI3 has finished transmission and the TDRE and TEND flags are set to 1. Then clear the TE bit to 0 and then set the TE and RE bits to 1 with a single instruction.
2. To switch from receive mode to simultaneous transmit and receive mode, check that the SCI3 has finished reception and clear the RE bit to 0. Then check that the RDRF and error flags (ORER, FER, and PER) are cleared to 0 and then set the TE and RE bits to 1 with a single instruction.

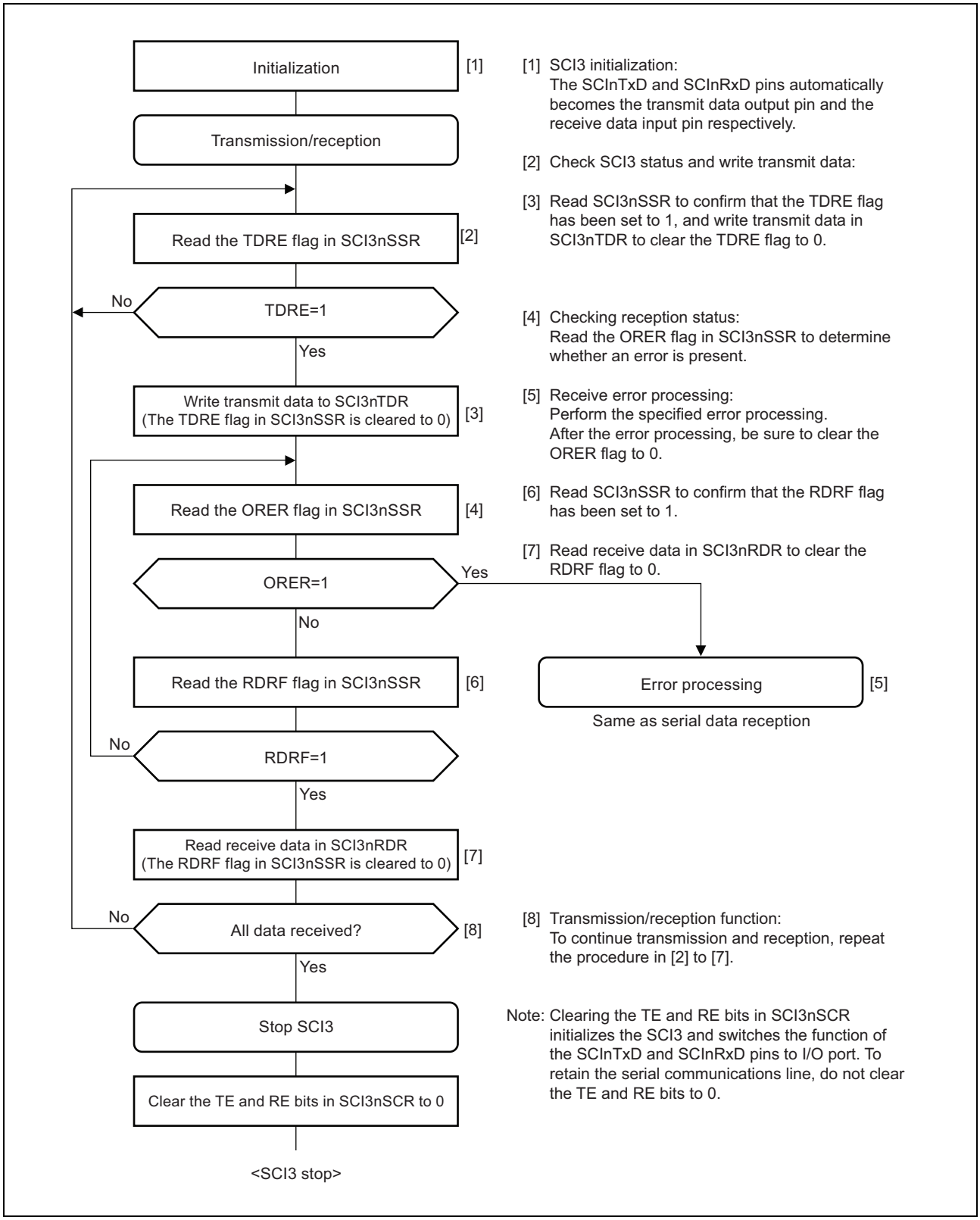


Figure 20.25 Example of Simultaneous Serial Transmission and Reception Flowchart

20.4.4 Bit Rate Modulation Function

The bit rate modulation function corrects a bit rate by averagely enabling the internal clocks specified by the CKS1 and CKS0 bits in SCI3nSMR for the number specified by SCI3nMDDR out of 256 clocks.

Figure 20.26 shows an example of asynchronous mode in which PCLK clock is selected with the CKS1 and CKS0 bits, SCI3nBRR is set to 0, and SCI3nMDDR is set to 160. In this example, the reference clock cycle is corrected to 256/160 on average and the bit rate is corrected to 160/256. Note that the pulse widths of the internal reference clock expand or contract for the amount of the selected internal clocks because there is a deviation in the enabling of the internal clocks.

Do not use this function with the maximum speed settings (CKS1 and CKS0 bits = 0 in SCI3nSMR, CKE1 bit = 0 in SCI3nSCR, and SCI3nBRR = 0) for clock synchronous mode.

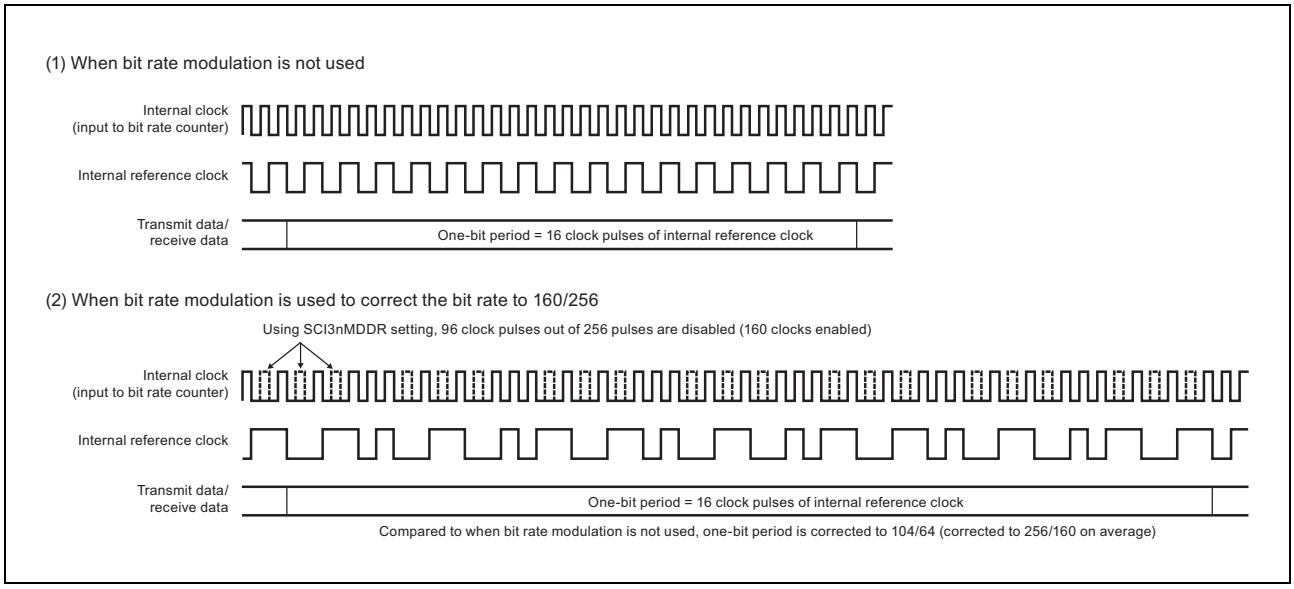


Figure 20.26 Example of Internal Reference Clock when the Bit Rate Modulation Function is Used

20.4.5 Interrupt Sources

20.4.5.1 The Normal Mode of SCI

Table 20.26 lists interrupt sources. Each interrupt source outputs an individual interrupt request signal. These interrupt sources can be enabled independently with the enable bit in SCI3nSCR.

An INTSCI3nTXI interrupt request is generated when the TDRE flag in SCI3nSSR is set to 1. An INTSCI3nTEI interrupt request is generated when the TEND flag in SCI3nSSR is set to 1. An INTSCI3nTXI interrupt request can activate the DMAC to handle data transfer. The TDRE flag is automatically cleared to 0 when data is transferred using the DMAC.

CAUTION

The TDRE flag and the TEND flag cannot be cleared to 0 when the TE bit in SCI3nSCR is 0. Since the TEND flag is the level interrupt request flag of the TEI interrupt, do not set the TEIE bit of SCI3nSCR to 1 when the TE bit is 0.

An INTSCI3nRXI interrupt request is generated when the RDRF flag in SCI3nSSR is set to 1. An INTSCI3nERI interrupt is generated when any of the ORER, PER, and FER flags in SCI3nSSR is set to 1. An INTSCI3nRXI interrupt request can activate the DMAC to handle data transfer. The RDRF flag is automatically cleared to 0 when data is transferred using the DMAC.

A INTSCI3nTEI interrupt request is generated when the TEND flag is set to 1 with the TEIE bit set to 1.

CAUTION

If an INTSCI3nTEI interrupt request and an INTSCI3nTXI interrupt request are generated at the same time, the INTSCI3nTXI interrupt request is accepted first. Note that clearing the TDRE flag to 0 at this time in the INTSCI3nTXI interrupt processing routine also clears the TEND flag to 0 automatically, disabling the branch to the INTSCI3nTEI interrupt processing routine.

Table 20.26 SCI3 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DMAC Activation	DTS Activation
INTSCI3nERI	Receive error	ORER, FER, PER	Not possible	Not possible
INTSCI3nRXI	Receive data full	RDRF	Possible	Possible
INTSCI3nTXI	Transmit data empty	TDRE	Possible	Possible
INTSCI3nTEI	Transmit end	TEND	Not possible	Not possible

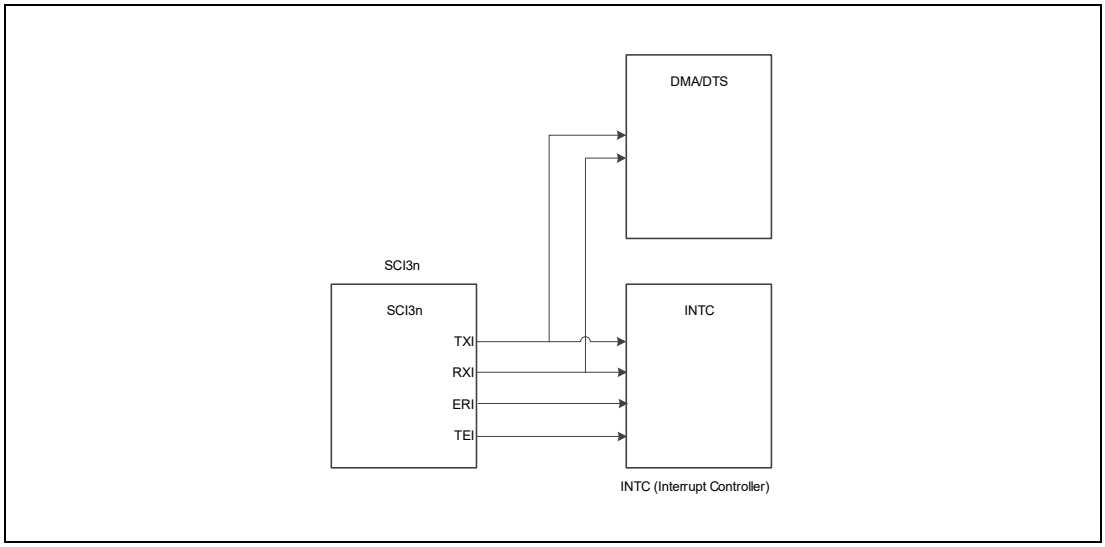


Figure 20.27 Relationship between the SCI3 Interrupt Signals, Interrupt Controller (INTC) and DMA/DTS

20.5 Notes on Use

20.5.1 Break Detection and Processing

A break can be detected by reading the RXDMON bit in SCI3nSEMR when detecting a framing error. Since all inputs from the SCInRxD pin are 0 in the break state, the FER flag is set to 1 and the PER flag may also be set to 1. When ASTLS bit in SCI3nSCMR is set to 1, the SCI3 continues data reception even after it receives a break. For this reason, note that the FER flag is set to 1 again even after the FER flag is cleared to 0. After the break is completed, hold the SCInRxD pin input High level for at least one frame period to avoid bit shift between the transmitter and receiver. On the other hand, when the ASTLS bit in SCI3nSCMR is 0, the FER flag is set to 1 and then the reception operation is stopped in the state of waiting for the start bit detection of the next frame. At this time, if the FER flag is cleared to 0, the FER flag is held at 0 during break. After the input from the SCInRxD pin becomes 1 and the break is completed, the start of the start bit is detected at the first trailing edge of the SCInRxD pin input, and reception operation is started.

20.5.2 Mark State and Break Output

While the TE bit is 0 (transmission/reception disabled), the SCInTxD pin can output any level by switching the SCInTxD pin to a general output port. This allows the SCInTxD pin to enter the mark state or to output a break during data transmission. When the TE bit is cleared to 0, the transmitter unit is initialized regardless of the current transmission state.

20.5.3 Receive Error Flags and Transmit Operations in Clock Synchronous Mode

During the clock synchronous simultaneous data transmit/receive operation, transmission cannot be started when a receive error flag (ORER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note that the receive error flags cannot be cleared to 0 even by clearing the RE bit to 0.

20.5.4 Relationship between Writing to SCI3nTDR and the TDRE Flag

The TDRE flag in SCI3nSSR is a status flag indicating that transmit data in SCI3nTDR has been transferred to SCI3nTSR. The TDRE flag is set to 1 when the SCI3 transfers data from SCI3nTDR to SCI3nTSR.

Data can be written to SCI3nTDR regardless of the status of the TDRE flag. However, writing new data to SCI3nTDR while the TDRE flag is 0 will cause the data stored in SCI3nTDR to be lost because it has not been transferred to SCI3nTSR. Therefore, make sure to check that the TDRE flag is set to 1 before writing transmit data to SCI3nTDR.

20.5.5 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

When using an external synchronization clock, clear the TDRE flag to 0 and then input a transmission clock (See **Figure 20.28**). Also in continuous transmission mode, clear the TDRE flag to 0 and then input a transmission clock for the next frame.

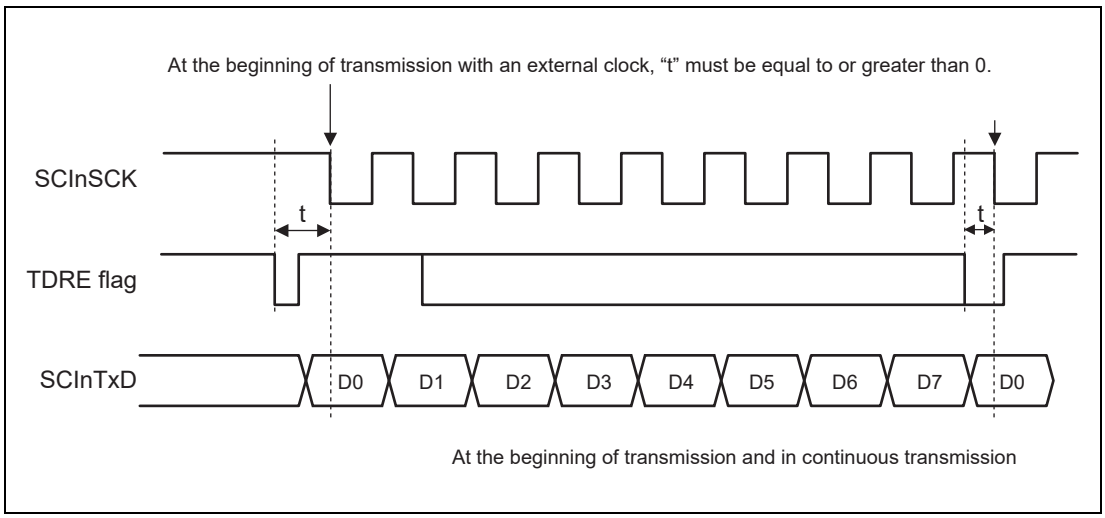


Figure 20.28 Restrictions on Using an External Clock for Transmission in Clock Synchronous Mode

20.5.6 External Clock Input in Clock Synchronous Mode

For the external clock SCInSCK input in clock synchronous mode, see **Section 55.3.11, SCI3 Timing**.

Section 21 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

21.1 Features RLIN3 for RH850/U2A-EVA

21.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has a single channel interface. “Number of channels” therefore has the same meaning as “number of units” in this section.

Table 21.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	24 (n = 0 to 23)	24 (n = 0 to 23)	24 (n = 0 to 23)	12 (n = 0 to 11)	24 (n = 0 to 23)	12 (n = 0 to 11)	12 (n = 0 to 11)	12 (n = 0 to 11)	8 (n = 0, 3 to 5, 8 to 11)	10 (n = 0, 2 to 5, 7 to 11)
Name	RLIN3n									

Table 21.2 Unit Configurations and Channels (1/2)

Unit Name (Channel Name) RLIN3n	Channels per Unit	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
RLIN30	1	√	√	√	√	√	√	√	√	√	√
RLIN31	1	√	√	√	√	√	√	√	√	—	—
RLIN32	1	√	√	√	√	√	√	√	√	—	√
RLIN33	1	√	√	√	√	√	√	√	√	√	√
RLIN34	1	√	√	√	√	√	√	√	√	√	√
RLIN35	1	√	√	√	√	√	√	√	√	√	√
RLIN36	1	√	√	√	√	√	√	√	√	—	—
RLIN37	1	√	√	√	√	√	√	√	√	—	√
RLIN38	1	√	√	√	√	√	√	√	√	√	√
RLIN39	1	√	√	√	√	√	√	√	√	√	√
RLIN310	1	√	√	√	√	√	√	√	√	√	√
RLIN311	1	√	√	√	√	√	√	√	√	√	√
RLIN312	1	√	√	√	—	√	—	—	—	—	—
RLIN313	1	√	√	√	—	√	—	—	—	—	—
RLIN314	1	√	√	√	—	√	—	—	—	—	—
RLIN315	1	√	√	√	—	√	—	—	—	—	—
RLIN316	1	√	√	√	—	√	—	—	—	—	—
RLIN317	1	√	√	√	—	√	—	—	—	—	—
RLIN318	1	√	√	√	—	√	—	—	—	—	—
RLIN319	1	√	√	√	—	√	—	—	—	—	—

Table 21.2 Unit Configurations and Channels (2/2)

Unit Name (Channel Name) RLIN3n	Channels per Unit	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
RLIN320	1	√	√	√	—	√	—	—	—	—	—
RLIN321	1	√	√	√	—	√	—	—	—	—	—
RLIN322	1	√	√	√	—	√	—	—	—	—	—
RLIN323	1	√	√	√	—	√	—	—	—	—	—

Note: The channel names are same as those of the corresponding units.

Table 21.3 Indices

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index “n” (n = 0 to 23); for example, RLN3nLCUC is the LIN control register.
b	Throughout this section, the individual transmit/receive data buffers of RLIN3n are identified by the index “b” (b = 1 to 8); for example, RLN3nLDBRb is the data buffer register.

The following lists the index value corresponding to each product.

Table 21.4 Index Correspondence of Each Product

Index Corresponding to Product
All products
b = 1 to 8

21.1.2 Register Base Addresses

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses.

Table 21.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RLIN30_base>	FFD2 8000$_H$	P-Bus Group 3
<RLIN31_base>	FFC7 C000$_H$	P-Bus Group 5
<RLIN32_base>	FFD2 8200$_H$	P-Bus Group 3
<RLIN33_base>	FFC7 C080$_H$	P-Bus Group 5
<RLIN34_base>	FFD2 8400$_H$	P-Bus Group 3
<RLIN35_base>	FFC7 C100$_H$	P-Bus Group 5
<RLIN36_base>	FFD2 8600$_H$	P-Bus Group 3
<RLIN37_base>	FFC7 C180$_H$	P-Bus Group 5
<RLIN38_base>	FFD2 8800$_H$	P-Bus Group 3
<RLIN39_base>	FFC7 C200$_H$	P-Bus Group 5
<RLIN310_base>	FFD2 8A00$_H$	P-Bus Group 3
<RLIN311_base>	FFC7 C280$_H$	P-Bus Group 5
<RLIN312_base>	FFD2 8C00$_H$	P-Bus Group 3
<RLIN313_base>	FFC7 C300$_H$	P-Bus Group 5
<RLIN314_base>	FFD2 8E00$_H$	P-Bus Group 3
<RLIN315_base>	FFC7 C380$_H$	P-Bus Group 5
<RLIN316_base>	FFD2 9000$_H$	P-Bus Group 3
<RLIN317_base>	FFC7 C400$_H$	P-Bus Group 5
<RLIN318_base>	FFD2 9200$_H$	P-Bus Group 3
<RLIN319_base>	FFC7 C480$_H$	P-Bus Group 5
<RLIN320_base>	FFD2 9400$_H$	P-Bus Group 3
<RLIN321_base>	FFC7 C500$_H$	P-Bus Group 5
<RLIN322_base>	FFD2 9600$_H$	P-Bus Group 3
<RLIN323_base>	FFC7 C580$_H$	P-Bus Group 5

21.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 21.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RLIN3n	LIN communication clock sources	CLK_RLIN
	Register access clock	CLK_HSB

21.1.4 Interrupt Requests and Error Notifications

RLIN3 interrupt requests are listed in the following table.

Table 21.7 Interrupt and DMA/DTS Requests (1/3)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number	Wake up
RLIN30					
INTRLIN3n (n = 0)	RLIN30 interrupt	416	—	—	√
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	417	Group0-61	Group0-50	—
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	418	Group0-62	Group0-51	—
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	419	—	—	—
RLIN31					
INTRLIN3n (n = 1)	RLIN31 interrupt	420	—	—	√
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	421	Group0-63	Group0-52	—
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	422	Group0-64	Group0-53	—
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	423	—	—	—
RLIN32					
INTRLIN3n (n = 2)	RLIN32 interrupt	424	—	—	√
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	425	Group0-65	Group0-54	—
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	426	Group0-66	Group0-55	—
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	427	—	—	—
RLIN33					
INTRLIN3n (n = 3)	RLIN33 interrupt	428	—	—	√
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	429	Group0-67	Group0-56	—
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	430	Group0-68	Group0-57	—
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	431	—	—	—
RLIN34					
INTRLIN3n (n = 4)	RLIN34 interrupt	432	—	—	√
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	433	Group0-69	Group0-58	—
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	434	Group0-70	Group0-59	—
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	435	—	—	—
RLIN35					
INTRLIN3n (n = 5)	RLIN35 interrupt	436	—	—	√
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	437	Group0-71	Group0-60	—
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	438	Group0-72	Group0-61	—
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	439	—	—	—
RLIN36					
INTRLIN3n (n = 6)	RLIN36 interrupt	440	—	—	√
INTRLIN3nUR0 (n = 6)	RLIN36 transmit interrupt	441	Group0-73	Group0-62	—
INTRLIN3nUR1 (n = 6)	RLIN36 receive completion interrupt	442	Group0-74	Group0-63	—
INTRLIN3nUR2 (n = 6)	RLIN36 status interrupt	443	—	—	—
RLIN37					
INTRLIN3n (n = 7)	RLIN37 interrupt	444	—	—	√
INTRLIN3nUR0 (n = 7)	RLIN37 transmit interrupt	445	Group0-75	Group0-64	—
INTRLIN3nUR1 (n = 7)	RLIN37 receive completion interrupt	446	Group0-76	Group0-65	—
INTRLIN3nUR2 (n = 7)	RLIN37 status interrupt	447	—	—	—

Table 21.7 Interrupt and DMA/DTS Requests (2/3)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number	Wake up
RLIN38					
INTRLIN3n (n = 8)	RLIN38 interrupt	448	—	—	√
INTRLIN3nUR0 (n = 8)	RLIN38 transmit interrupt	449	Group0-77	Group0-66	—
INTRLIN3nUR1 (n = 8)	RLIN38 receive completion interrupt	450	Group0-78	Group0-67	—
INTRLIN3nUR2 (n = 8)	RLIN38 status interrupt	451	—	—	—
RLIN39					
INTRLIN3n (n = 9)	RLIN39 interrupt	452	—	—	√
INTRLIN3nUR0 (n = 9)	RLIN39 transmit interrupt	453	Group0-79	Group0-68	—
INTRLIN3nUR1 (n = 9)	RLIN39 receive completion interrupt	454	Group0-80	Group0-69	—
INTRLIN3nUR2 (n = 9)	RLIN39 status interrupt	455	—	—	—
RLIN310					
INTRLIN3n (n = 10)	RLIN310 interrupt	456	—	—	√
INTRLIN3nUR0 (n = 10)	RLIN310 transmit interrupt	457	Group0-81	Group0-70	—
INTRLIN3nUR1 (n = 10)	RLIN310 receive completion interrupt	458	Group0-82	Group0-71	—
INTRLIN3nUR2 (n = 10)	RLIN310 status interrupt	459	—	—	—
RLIN311					
INTRLIN3n (n = 11)	RLIN311 interrupt	460	—	—	√
INTRLIN3nUR0 (n = 11)	RLIN311 transmit interrupt	461	Group0-83	Group0-72	—
INTRLIN3nUR1 (n = 11)	RLIN311 receive completion interrupt	462	Group0-84	Group0-73	—
INTRLIN3nUR2 (n = 11)	RLIN311 status interrupt	463	—	—	—
RLIN312					
INTRLIN3n (n = 12)	RLIN312 interrupt	464	—	—	√
INTRLIN3nUR0 (n = 12)	RLIN312 transmit interrupt	465	Group0-85	Group0-74	—
INTRLIN3nUR1 (n = 12)	RLIN312 receive completion interrupt	466	Group0-86	Group0-75	—
INTRLIN3nUR2 (n = 12)	RLIN312 status interrupt	467	—	—	—
RLIN313					
INTRLIN3n (n = 13)	RLIN313 interrupt	468	—	—	√
INTRLIN3nUR0 (n = 13)	RLIN313 transmit interrupt	469	Group0-87	Group0-76	—
INTRLIN3nUR1 (n = 13)	RLIN313 receive completion interrupt	470	Group0-88	Group0-77	—
INTRLIN3nUR2 (n = 13)	RLIN313 status interrupt	471	—	—	—
RLIN314					
INTRLIN3n (n = 14)	RLIN314 interrupt	472	—	—	√
INTRLIN3nUR0 (n = 14)	RLIN314 transmit interrupt	473	Group0-89	Group0-78	—
INTRLIN3nUR1 (n = 14)	RLIN314 receive completion interrupt	474	Group0-90	Group0-79	—
INTRLIN3nUR2 (n = 14)	RLIN314 status interrupt	475	—	—	—
RLIN315					
INTRLIN3n (n = 15)	RLIN315 interrupt	476	—	—	√
INTRLIN3nUR0 (n = 15)	RLIN315 transmit interrupt	477	Group0-91	Group0-80	—
INTRLIN3nUR1 (n = 15)	RLIN315 receive completion interrupt	478	Group0-92	Group0-81	—
INTRLIN3nUR2 (n = 15)	RLIN315 status interrupt	479	—	—	—
RLIN316					
INTRLIN3n (n = 16)	RLIN316 interrupt	480	—	—	√
INTRLIN3nUR0 (n = 16)	RLIN316 transmit interrupt	481	Group0-93	Group0-82	—

Table 21.7 Interrupt and DMA/DTS Requests (3/3)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number	Wake up
INTRLIN3nUR1 (n = 16)	RLIN316 receive completion interrupt	482	Group0-94	Group0-83	—
INTRLIN3nUR2 (n = 16)	RLIN316 status interrupt	483	—	—	—
RLIN317					
INTRLIN3n (n = 17)	RLIN317 interrupt	484	—	—	√
INTRLIN3nUR0 (n = 17)	RLIN317 transmit interrupt	485	Group0-95	Group0-84	—
INTRLIN3nUR1 (n = 17)	RLIN317 receive completion interrupt	486	Group0-96	Group0-85	—
INTRLIN3nUR2 (n = 17)	RLIN317 status interrupt	487	—	—	—
RLIN318					
INTRLIN3n (n = 18)	RLIN318 interrupt	488	—	—	√
INTRLIN3nUR0 (n = 18)	RLIN318 transmit interrupt	489	Group0-97	Group0-86	—
INTRLIN3nUR1 (n = 18)	RLIN318 receive completion interrupt	490	Group0-98	Group0-87	—
INTRLIN3nUR2 (n = 18)	RLIN318 status interrupt	491	—	—	—
RLIN319					
INTRLIN3n (n = 19)	RLIN319 interrupt	492	—	—	√
INTRLIN3nUR0 (n = 19)	RLIN319 transmit interrupt	493	Group0-99	Group0-88	—
INTRLIN3nUR1 (n = 19)	RLIN319 receive completion interrupt	494	Group0-100	Group0-89	—
INTRLIN3nUR2 (n = 19)	RLIN319 status interrupt	495	—	—	—
RLIN320					
INTRLIN3n (n = 20)	RLIN320 interrupt	496	—	—	√
INTRLIN3nUR0 (n = 20)	RLIN320 transmit interrupt	497	Group0-101	Group0-90	—
INTRLIN3nUR1 (n = 20)	RLIN320 receive completion interrupt	498	Group0-102	Group0-91	—
INTRLIN3nUR2 (n = 20)	RLIN320 status interrupt	499	—	—	—
RLIN321					
INTRLIN3n (n = 21)	RLIN321 interrupt	500	—	—	√
INTRLIN3nUR0 (n = 21)	RLIN321 transmit interrupt	501	Group0-103	Group0-92	—
INTRLIN3nUR1 (n = 21)	RLIN321 receive completion interrupt	502	Group0-104	Group0-93	—
INTRLIN3nUR2 (n = 21)	RLIN321 status interrupt	503	—	—	—
RLIN322					
INTRLIN3n (n = 22)	RLIN322 interrupt	504	—	—	√
INTRLIN3nUR0 (n = 22)	RLIN322 transmit interrupt	505	Group0-105	Group0-94	—
INTRLIN3nUR1 (n = 22)	RLIN322 receive completion interrupt	506	Group0-106	Group0-95	—
INTRLIN3nUR2 (n = 22)	RLIN322 status interrupt	507	—	—	—
RLIN323					
INTRLIN3n (n = 23)	RLIN323 interrupt	508	—	—	√
INTRLIN3nUR0 (n = 23)	RLIN323 transmit interrupt	509	Group0-107	Group0-96	—
INTRLIN3nUR1 (n = 23)	RLIN323 receive completion interrupt	510	Group0-108	Group0-97	—
INTRLIN3nUR2 (n = 23)	RLIN323 status interrupt	511	—	—	—

This module has no error notifications.

21.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 21.8 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RLIN3n	All registers	√	√	√	√	√	√	—

21.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 21.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN3n		
RLIN3nRX (n = 0 to 23)	RLIN3n receive data input	RLIN3nRX
RLIN3nTX (n = 0 to 23)	RLIN3n transmit data output	RLIN3nTX

21.2 Overview

21.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, LIN slave, or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

Table 21.10 shows the LIN/UART interface specifications.

Table 21.10 LIN/UART Interface Specifications (1/3)

Item		Specifications	
	Channel count	Up to 24 channels	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602	
	Variable frame structure	Master	• Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Transmission response space width: 0 to 7 Tbits*1 • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
		Slave	• Break reception width : 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate] • Transmission response space width: 0 to 7 Tbits • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible	
	Frame communication modes	Master	• Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
		Slave	• Mode in which header is automatically received with fixed baud rate • Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected
	Wake-up transmission and reception	LIN wake-up mode provided • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
		Slave	• Successful response/wake-up transmission • Successful response/wake-up reception*2 • Successful header reception • Break field reception • Sync field reception • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)

Table 21.10 LIN/UART Interface Specifications (2/3)

Item		Specifications	
LIN communication function	Error status	Master	• Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
		Slave	• Bit error • Checksum error • Frame timeout error/response timeout error • Sync field error • ID parity error • Framing error • Response preparation error
	Baud rate selection	Baud rates conforming to the LIN specifications generated using baud rate generator	
	Test mode	Self-test mode for user evaluation	
	Interrupt function	Master	• Successful header/frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection
		Slave	• Successful response/wake-up transmission • Successful Header/response/wake-up reception*2 • Error detection
UART communication function	Data buffer	• Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9. Character length of 7 and 8 bits supported) • Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)	
	Data format	Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit.	
		Transmission stop bit: 1 or 2 bits	
		Parity function: odd, even, 0, or none	
		LSB- or MSB-first transfer selectable	
		Reverse input/output of transmission/reception data possible	
	Status	• Transmission status • Reception status • Successful UART buffer transmission • Error detection • Expansion bit detection • ID match • Reset mode status	
	Error status	• Bit error • Framing error • Parity error • Overrun error	
	Baud rate selection	With the built-in baud rate generator, any baud rate can be set.	
	When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register.		

Table 21.10 LIN/UART Interface Specifications (3/3)

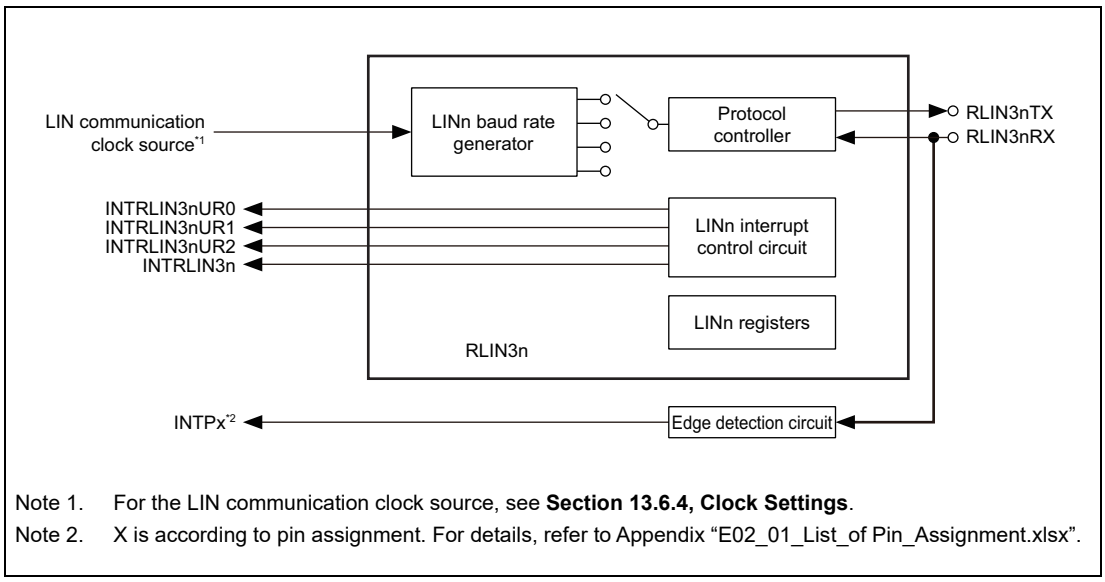
Item		Specifications
UART communication function	Reception of the stop bit is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
	Interrupt function	• Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

21.2.2 Block Diagram

Figure 21.1 shows a block diagram of the LIN/UART interface.



Note 1. For the LIN communication clock source, see **Section 13.6.4, Clock Settings**.
Note 2. X is according to pin assignment. For details, refer to Appendix “E02_01_List_of Pin_Assignment.xlsx”.

Figure 21.1 LIN/UART Interface Block Diagram

21.2.3 Terms used in block diagram

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock.
- LINn registers: LIN/UART interface registers
- LINn interrupt control circuit: Controls interrupt requests generated by the LIN/UART interface

21.3 Registers

21.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n_base>, see **Section 21.1.2, Register Base Addresses**.

Table 21.11 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART	Access Protection	
							PBG	Other
RLN3n	LIN wake-up baud rate select register	RLN3nLWBR	<RLIN3n_base> + 01_H	√	√	√	*1	—
RLN3n	LIN / UART baud rate prescaler 01 register	RLN3nLBRP01	<RLIN3n_base> + 02_H	—	√	√	*1	—
RLN3n	LIN / UART baud rate prescaler 0 register	RLN3nLBRP0	<RLIN3n_base> + 02_H	√	√	√	*1	—
RLN3n	LIN / UART baud rate prescaler 1 register	RLN3nLBRP1	<RLIN3n_base> + 03_H	√	√	√	*1	—
RLN3n	LIN self-test control register	RLN3nLSTC	<RLIN3n_base> + 04_H	√	√	—	*1	—
RLN3n	LIN / UART mode register	RLN3nLMD	<RLIN3n_base> + 08_H	√	√	√	*1	—
RLN3n	LIN break field configuration register/ UART configuration register	RLN3nLBFC	<RLIN3n_base> + 09_H	√	√	√	*1	—
RLN3n	LIN / UART space configuration register	RLN3nLSC	<RLIN3n_base> + $0A_H$	√	√	√	*1	—
RLN3n	LIN wake-up configuration register	RLN3nLWUP	<RLIN3n_base> + $0B_H$	√	√	—	*1	—
RLN3n	LIN interrupt enable register	RLN3nLIE	<RLIN3n_base> + $0C_H$	√	√	—	*1	—
RLN3n	LIN / UART error detection enable register	RLN3nLEDE	<RLIN3n_base> + $0D_H$	√	√	√	*1	—
RLN3n	LIN / UART control register	RLN3nLCUC	<RLIN3n_base> + $0E_H$	√	√	√	*1	—
RLN3n	LIN / UART transmission control register	RLN3nLTRC	<RLIN3n_base> + 10_H	√	√	√	*1	—
RLN3n	LIN / UART mode status register	RLN3nLMST	<RLIN3n_base> + 11_H	√	√	√	*1	—
RLN3n	LIN / UART status register	RLN3nLST	<RLIN3n_base> + 12_H	√	√	√	*1	—
RLN3n	LIN / UART error status register	RLN3nLEST	<RLIN3n_base> + 13_H	√	√	√	*1	—
RLN3n	LIN data field configuration register	RLN3nLDFC	<RLIN3n_base> + 14_H	√	√	√	*1	—
RLN3n	LIN / UART ID buffer register	RLN3nLIDB	<RLIN3n_base> + 15_H	√	√	√	*1	—
RLN3n	LIN checksum buffer register	RLN3nLCBR	<RLIN3n_base> + 16_H	√	√	—	*1	—
RLN3n	UART data buffer 0 register	RLN3nLUDB0	<RLIN3n_base> + 17_H	—	—	√	*1	—
RLN3n	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLIN3n_base> + 18_H	√	√	√	*1	—
RLN3n	LIN / UART data buffer 2 register	RLN3nLDBR2	<RLIN3n_base> + 19_H	√	√	√	*1	—
RLN3n	LIN / UART data buffer 3 register	RLN3nLDBR3	<RLIN3n_base> + $1A_H$	√	√	√	*1	—
RLN3n	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLIN3n_base> + $1B_H$	√	√	√	*1	—
RLN3n	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLIN3n_base> + $1C_H$	√	√	√	*1	—
RLN3n	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLIN3n_base> + $1D_H$	√	√	√	*1	—
RLN3n	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLIN3n_base> + $1E_H$	√	√	√	*1	—
RLN3n	LIN / UART data buffer 8 register	RLN3nLDBR8	<RLIN3n_base> + $1F_H$	√	√	√	*1	—
RLN3n	UART operation enable register	RLN3nLUOER	<RLIN3n_base> + 20_H	—	—	√	*1	—
RLN3n	UART option register 1	RLN3nLUOR1	<RLIN3n_base> + 21_H	—	—	√	*1	—
RLN3n	UART transmission data register	RLN3nLUTDR	<RLIN3n_base> + 24_H	—	—	√	*1	—
RLN3n	UART transmission data register L	RLN3nLUTDRL	<RLIN3n_base> + 24_H	—	—	√	*1	—
RLN3n	UART transmission data register H	RLN3nLUTDRH	<RLIN3n_base> + 25_H	—	—	√	*1	—
RLN3n	UART reception data register	RLN3nLURDR	<RLIN3n_base> + 26_H	—	—	√	*1	—
RLN3n	UART reception data register L	RLN3nLURDRL	<RLIN3n_base> + 26_H	—	—	√	*1	—

Table 21.11 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART	Access Protection	
							PBG	Other
RLN3n	UART reception data register H	RLN3nLURDRH	<RLIN3n_base> + 27H	—	—	√	*1	—
RLN3n	UART wait transmission data register	RLN3nLUWTDR	<RLIN3n_base> + 28_H	—	—	√	*1	—
RLN3n	UART wait transmission data register L	RLN3nLUWTDRL	<RLIN3n_base> + 28_H	—	—	√	*1	—
RLN3n	UART wait transmission data register H	RLN3nLUWTDRH	<RLIN3n_base> + 29_H	—	—	√	*1	—
RLN3n	LIN slave break/sync field status register	RLN3nLBSS	<RLIN3n_base> + 30_H	—	√	—	*1	—
RLN3n	LIN slave response space status register	RLN3nLRSS	<RLIN3n_base> + 34_H	—	√	—	*1	—

Note: √: Used, —: Not used
When writing to an unused register, write the value after reset.

Note 1. n=0: PBG32#1, n=1: PBG52#11, n=2: PBG32#2, n=3: PBG52#12,
n=4: PBG32#3, n=5: PBG52#13, n=6: PBG32#4, n=7: PBG52#14,
n=8: PBG32#5, n=9: PBG52#15, n=10: PBG32#6, n=11: PBG53#0,
n=12: PBG32#7, n=13: PBG53#1, n=14: PBG32#8, n=15: PBG53#2,
n=16: PBG32#9, n=17: PBG53#3, n=18: PBG32#10, n=19: PBG53#4,
n=20: PBG32#11, n=21: PBG53#5, n=22: PBG32#12, n=23: PBG53#6

21.3.2 LIN Master Related Registers

21.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.12 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	These bits configure the value for number of samples in 1 Bit time period. 0000: 16 samples per bit 1111: 16 samples per bit Others: Prohibited
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x)

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B or 1111_B (16 samplings).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.
When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 us or longer to be detected in the LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

21.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.13 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00_H to FF_H

Configure the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

21.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.14 RLN3nLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00_H to FF_H

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

21.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	LSTME[6:1]						LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.15 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 1	LSTME	The test mode key values for configuring the RLIN3 module in Test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

When read, bits 6 to 1 return "000000_B", and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

LSTME Bit (Self test mode entry key bits)

Users are expected to write the Test mode Key sequence values to these bits. Refer to **Section 21.4.6, LIN Self-Test Mode** for details how to enter the Test mode Key.

21.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base>+ 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.16 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 0: LIN master mode

Configure the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 21.4.1, Interrupt Sources**.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is supplied with fa regardless of the setting of this bit (the LCKS bit is not changed).

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an LIN master, set these bits to 00_B.

21.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.17 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.
1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.
13 Tbits to 28 Tbits can be set.

21.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base>+ $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 21.18 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

21.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base>+ $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 21.19 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.
1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

21.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.20 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Request Enable)

The SHIE bit enables or disables interrupt requests upon successful transmission of a header.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.
Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).

With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

21.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 21.21 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable*1 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable *1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode)

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function is used for frame timeout.
With 1 set, the timeout function is used for response timeout.
For details on the timeout error, see **Section 21.4.4.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 21.4.4.7, Error Status**.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.

When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 21.4.4.7, Error Status**.

If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.FTERE = 1), users should take the procedure shown in **Figure 21.4**.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.
With 0 set, the physical bus error is not detected.
With 1 set, the physical bus error is detected.
When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.
For details on the physical bus error, see **Section 21.4.4.7, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result of the bit error is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 21.4.4.7, Error Status**.

21.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.22 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode.
With 1 set, LIN operation mode.
This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode.
With 1 set, LIN reset mode is canceled.

If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.FTERE = 1), users should take the procedure shown in **Figure 21.4**.

21.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.23 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission reception is started.

RTS Bit (Response Transmission/Reception Start)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.
Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

21.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.24 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

21.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 21.25 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value it had before 1 is written.
The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication is started (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit while in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not

generated. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Successful Frame/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n reception complete is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Frame/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

21.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 21.26 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/ reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

21.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.27 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum 1: Enhanced checksum
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.
With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.
With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response communication mode.
With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.
When causing a transition to LIN self-test mode, set this bit to 0 before transition.
For details on frame separate mode, see **Section 21.4.4.4(1), Transmission of LIN Frames**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 21.4.4.7, Error Status**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).
To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).
Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

21.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.28 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

21.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.29 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum data transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for other data groups.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

Set the this register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

21.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.30 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the received data. Setting range: 00_H to FF_H

- For response transmission:
  These registers set the data to be transmitted in the response field.
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 0 (not frame separate mode)
  - FTS in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)

  or
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For response reception:
  These registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)
- For transmission of response data of 9 bytes or more:
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For reception of response data of 9 bytes or more:
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

21.3.3 LIN Slave Related Registers

21.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 21.31 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 0 1 1: 4 samplings 0 1 1 1: 8 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
When the communication is performed in LIN slave mode (fixed baud rate) (LMD[1:0] bits in the RLN3nLMD register = 11_B), set these bits to "0000_B" or "1111_B" (16 samplings).
When the LIN frame communication is performed in LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits to "0011_B" (4 samplings) or "0111_B" (8 samplings).

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

In LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits so that the prescaler clock becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz*1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB bits set to "0011_B" (four samplings).

21.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 can be read or written in 16-bit units.
RLN3nLBRP0 can be read or written in 8-bit units.
RLN3nLBRP1 can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.32 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by registers RLN3nLBRP0 and RLN3nLBRP1.

NOTE

In LIN slave mode [auto baud rate], the system automatically sets the result of baud rate correction to the RLN3nLBRP01 register on successful reception of the sync field.

21.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	LSTME[6:1]						LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.33 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 1	LSTME	The test mode key values for configuring the RLIN3 module in Test mode.
0	LSTM	LIN Self-Test Mode 0: LIN self test mode is not set. 1: LIN self test mode is set.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Configure the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

When reading bits 6 to 1, they return "000000_B", and reading bit 7, it returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

LSTME Bit (Self test mode entry key bits)

Users are expected to write the Test mode Key sequence values to these bits. Refer to **Section 21.4.6, LIN Self-Test Mode** for details how to enter the Test mode Key.

21.3.3.4 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 21.34 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3 interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 1 0: LIN slave mode with auto baud rate 1 1: LIN slave mode with fixed baud rate

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 21.4.1, Interrupt Sources**.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use this module as an LIN slave, set these bits to "10_B" (auto baud rate) or "11_B" (fixed baud rate).

21.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LBLT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 21.35 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LBLT	Reception Break (Low-Level) Detection Width Setting 0: A break (low-level) is detected in 9.5 or 10 Tbits 1: A break (low-level) is detected in 10.5 or 11 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is "10_B" (LIN slave mode (auto baud rate))
  - 0: Low-level width of 10 Tbits or longer is detected.
  - 1: Low-level width of 11 Tbits or longer is detected.
- When RLN3nLMD.LMD is "11_B" (LIN slave mode (fixed baud rate))
  - 0: Low-level width of 9.5 Tbits or longer is detected.
  - 1: Low-level width of 10.5 Tbits or longer is detected.

21.3.3.6 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 21.36 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Response Space Setting b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This register is enabled only for response transmission, and disabled for response reception.

Some combinations of the setting values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the response transmission.
0 Tbit to 3 Tbits can be set.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the transmission width of the response space.
0 Tbit to 7 Tbits can be set.

21.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 21.37 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low-level width of the wake-up frame transmission.
1 Tbit to 16 Tbits can be set.

21.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.38 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Request Enable 0: Disables successful header reception interrupt request. 1: Enables successful header reception interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Request Enable 0: Disables successful Response/wake-up reception interrupt request. 1: Enables successful Response/wake-up reception interrupt request.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Request Enable 0: Disables successful Response/wake-up transmission interrupt request. 1: Enables successful Response/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Reception Interrupt Request Enable)

The SHIE bit enables or disables an interrupt request upon successful reception of a header.
With 0 set, the interrupt request for RLIN3n reception complete is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n reception complete is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Error types that are interrupt sources are the bit error, frame/response timeout error, framing error, sync filed error, ID parity error, checksum error, and response preparation error.
Detection of the bit error, frame/response timeout error, sync filed error, ID parity error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Response/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (input signal low-level width count).

With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Response/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a response or a wake-up frame.
With 0 set, the interrupt request for successful RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

21.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	IPERE	—	SFERE	FERE	TERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Table 21.39 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6	IPERE	ID Parity Error Detection Enable 0: Disables ID Parity error detection. 1: Enables ID Parity error detection.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SFERE	Sync Field Error Detection Enable 0: Disables Sync Field error detection. 1: Enables Sync Field error detection.
3	FERE	Framing Error Detection Enable*1 0: Disables framing error detection. 1: Enables framing error detection.
2	TERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable*1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function operates as frame timeout.
With 1 set, the timeout function operates as response timeout.
For details on the timeout error, see **Section 21.4.4.7, Error Status**.

IPERE Bit (ID Parity Error Detection Enable)

This bit enables or disables detection of the ID parity error.
With 0 set, the ID parity error is not detected.
With 1 set, the ID parity error is detected.
When this bit is set to 1, the detection result is reflected in the IPER flag in the RLN3nLEST register.
For details on the ID parity error, see **Section 21.4.4.7, Error Status**.

SFERE Bit (Sync Field Error Detection Enable)

This bit enables or disables detection of the sync field error.
With 0 set, the sync field error is not detected.
With 1 set, the sync field error is detected.
Regardless of the setting of this bit, when a sync field error is detected, this module waits for the next header.
When this bit is set to 1, the detection result is reflected in the SFER flag in the RLN3nLEST register.
For details on the sync filed error, see **Section 21.4.4.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 21.4.4.7, Error Status**.

TERE Bit (Timeout Error Detection Enable)

The TERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is reflected in the TER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
The timeout error should not be used in LIN slave mode [auto baud rate] (when the LMD[1:0] bits in the RLN3nLMD register are "10_B").
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 21.4.4.7, Error Status**.

If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.TERE = 1), users should take the procedure shown in **Figure 21.4**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result of the bit error is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 21.4.4.7, Error Status**.

21.3.3.10 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.40 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific LIN operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN wake-up mode.
With 1 set, LIN/UART interface enters LIN operation mode.
This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN reset mode.
With 1 set, LIN reset mode of LIN/UART interface is canceled.

If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.TERE = 1), users should take the procedure shown in **Figure 21.4**.

21.3.3.11 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	LNRR	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 21.41 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LNRR	No LIN Response Request 0: Response for the reception ID 1: No response for the reception ID
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped. 1: Response transmission/reception is started.
0	FTS	LIN Communication Start 0: Header reception/wake-up transmission/reception is stopped. 1: Header reception/wake-up transmission/reception is started.

LNRR Bit (No LIN Response Request)

Set this bit to 1 if no response is to be transmitted/received after receiving the header and checking the received ID.
Once set, this bit is automatically cleared to 0 upon detection of new sync field or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 04_H using the store instruction.
Do not set this bit and the RTS bit to 1 simultaneously.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, use this bit only after the completion of the header. (Do not use this bit for the second or later data group.)

RTS Bit (Response Transmission/Reception Start)

Set this bit to 1 to start response transmission or reception after receiving the header and checking the received ID.
Once set, this bit is automatically cleared to 0 upon completion of response transmission or reception (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Do not set this bit and the LNRR bit to 1 simultaneously
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

FTS Bit (LIN Communication Start)

Set this bit to 1 to start header reception or wake-up transmission/reception.
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of wake-up communication and transition to LIN reset mode.

21.3.3.12 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> +11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.42 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

21.3.3.13 RLN3nLST — LIN Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 21.43 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header reception has not been completed. 1: Header reception has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Response/Wake-up Reception Flag 0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.
0	FTC	Successful Response/Wake-up Transmission Flag 0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, writing a value to this register is disabled. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Reception Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value before 1 is written. The HTRC flag is set to 1 upon completion of header reception. Here, an interrupt request for RLIN3n successful reception is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if header reception is completed while this bit is 1, an interrupt is not generated. To clear this bit to 0, write 0 to the bit.
To detect a new header in the response field upon completion of header reception, clear this bit after it is set to 1.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if an error is detected while this bit is 1, an interrupt is not generated. To clear the ERR flag to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLN3nLEST register. This clears the ERR flag to 0.

FRC Flag (Successful Response/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt request for successful RLIN3n reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response reception or wake-up reception is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Response/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response transmission or wake-up transmission is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

21.3.3.14 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	IPER	CSER	SFER	FER	TER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 21.44 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	IPER	ID Parity Error Flag 0: ID parity error has not been detected. 1: ID parity error has been detected.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	SFER	Sync Field Error Flag 0: Sync field error has not been detected. 1: Sync field error has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	TER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. Write 0 to clear this bit.

IPER Flag (ID Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the IPERE bit of the RLN3nLEDE register is 1 (ID parity error detection enabled), this bit is set to 1 upon ID parity error detection. Write 0 to clear this bit.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The CSER flag is set to 1 upon checksum error detection. Write 0 to clear this bit.

SFER Flag (Sync Field Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the SFERE bit of the RLN3nLEDE register is 1 (sync field error detection enabled), this bit is set to 1 upon sync field error detection. Write 0 to clear this bit.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. Write 0 to clear this bit.

TER Flag (Timeout Error Flag)

Only 0 can be written to the TER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the TERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), this flag is set to 1 upon frame timeout error or response timeout error detection. Write 0 to clear this bit.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. Write 0 to clear this bit.

21.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	—	LCS	RCDS	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.45 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception continues without waiting for reception of the next header.)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LCS	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RCDS	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one. With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
When multi-byte response transmission/reception function is not used, set this bit to “0”.
Set this bit when the RTS bit is 0 (response transmission/reception stopped).

LCS Bit (Checksum Select)

The LCS bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the TERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 21.4.4.7, Error Status**.
When the length of the response field data is 0 bytes (the RFDL bit is 0), do not set this bit to “1”

(enhanced).
When response data of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.
During transmission or reception of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.
Set this bit when the RTS bit is 0 (response transmission/reception stopped).

RCDS Bit (Response Field Communication Direction Select)

This bit selects the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the RTS bit in the RLN3nLTRC register is 0 in LIN operation mode (response transmission/reception stopped) or when the FTS bit is 0 in LIN wake-up mode (header reception or wake-up transmission/reception stopped).
When response data of 9 bytes or more is to be transmitted or received, do not change this bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
Set these bits when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped).
When response data of 9 bytes or more is to be transmitted or received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit in the RLN3nLDFC register is 1) include the checksum.

21.3.3.16 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.46 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value received in the ID field.

The value in the RLN3nLIDB register is enabled after the completion of header reception. In LIN mode (LIN operation mode, LIN wake-up mode), writing to this register is disabled.

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) received in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

When the IPERE bit in the RLN3nLEDE register is 1 (ID parity detection enabled), the received value and the value calculated internally are checked. If they do not match, IPER (ID parity error flag) is set.

ID[5:0] Bits (ID Setting)

The ID bits store the 6-bit ID value received in the ID field of the LIN frame.

21.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. However, in LIN self-test mode, this register can be read and written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.47 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the transmitted or received checksum data.

In LIN operation mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

21.3.3.18 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

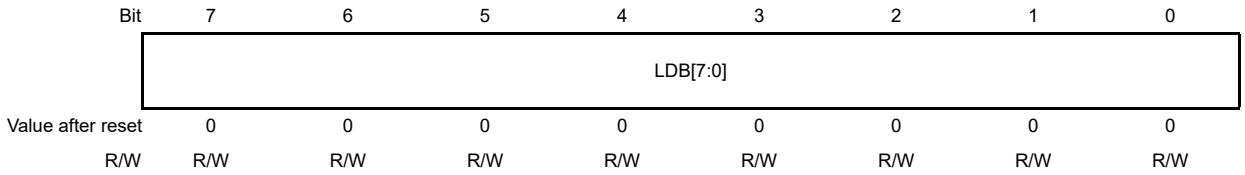


Table 21.48 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00_H to FF_H

- For response transmission:
  The RLN3nLDBRb registers set the data to be transmitted in the response field.
  Set these registers when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception is halted).
- For response reception:
  The RLN3nLDBRb registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started)

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

21.3.3.19 RLN3nLBSS — LIN Slave Break/Sync Field Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 30_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SYCC	BRKC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.49 RLN3nLBSS Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SYCC	LIN Sync Field Reception Flag 0: LIN Sync Field Rx not completed 1: LIN Sync Field Rx completed successfully
0	BRKC	LIN Break Field Reception Flag 0: LIN Break Field Rx not completed 1: LIN Break Field Rx completed successfully

The RLN3nLBSS register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

BRKC Bit (LIN Break Field Reception Flag)

Only 0 can be written to the BRKC Bit; when 1 is written, the bit retains the value that has been retained before 1 is written.

Users can not write to this bit if the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This bit is cleared by writing “0” to it.

The BRKC bit is cleared when RLIN3 module is transitioning to the reset mode.

This bit is set when Break field is successfully reception in LIN Slave mode.

Users should use this bit only for the re-transmission of a wake-up signal. (Refer to **Section 21.4.4.9, Wake-up Processing for Retransmission in LIN Slave Mode**)

SYCC Bit (LIN Sync Field Reception Flag)

Only 0 can be written to the SYCC Bit; when 1 is written, the bit retains the value that has been retained before 1 is written.

Users can not write to this bit if the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This bit is cleared by writing “0” to it.

The SYCC bit is cleared when RLIN3 module is transitioning to the reset mode.

This bit is set when Sync field is successfully reception in LIN Slave mode.

Users should use this bit only for the re-transmission of a wake-up signal. (Refer to **Section 21.4.4.9, Wake-up Processing for Retransmission in LIN Slave Mode**)

21.3.3.20 RLN3nLRSS — LIN Slave Response Space Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 34_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RSDD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.50 RLN3nLRSS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	RSDD	Response Space Dominant Detection Flag 0: No dominant detected. 1: Dominant detected.

RSDD Bit (Response Space Dominant Detection Flag)

Users can not write to this bit.

This bit is valid only for LIN response transmission in LIN slave mode.

This bit is invalid when a bit error is occurred in the transmission of the response space.

This bit is cleared when Sync field is received.

This bit is cleared when RLIN3 module is transitioning to the reset mode.

This bit is set when a dominant level of 0.5 Tbit or more is detected from the completion of the header reception (the stop bit of the ID field) to the start of transmission*1 in LIN slave mode, the operation is continued.

Note 1. When RLN3nLSC.IBHS = 0_H, it is the transmission of the start bit of the 1st Data byte.
When RLN3nLSC.IBHS = 1 to 7_H, it is the transmission of the response space.

21.3.4 UART Related Registers

21.3.4.1 RLN3nLWBR — UART Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 21.51 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 1 0 1: 6 samplings 0 1 1 0: 7 samplings 0 1 1 1: 8 samplings 1 0 0 0: 9 samplings 1 0 0 1: 10 samplings 1 0 1 0: 11 samplings 1 0 1 1: 12 samplings 1 1 0 0: 13 samplings 1 1 0 1: 14 samplings 1 1 1 0: 15 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In UART mode, it is possible to set the NSPB bits from 6 samplings to 16 samplings.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

21.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read or written in 16-bit units.
RLN3nLBRP0 register can be read or written in 8-bit units.
RLN3nLBRP1 register can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.52 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by the registers RLN3nLBRP0 and RLN3nLBRP1.

21.3.4.3 RLN3nLMD — UART Mode Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	—	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R/W	R/W

Table 21.53 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 1: UART mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an UART, set these bits to 01_B.

21.3.4.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS	UPS[1:0]		USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.54 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data with inverted output
5	URPS	UART Input Polarity Switch 0: Reception data normal input 1: Reception data inverted input
4, 3	UPS[1:0]	UART Parity Select 00: Parity disabled 01: Even parity 10: 0 Parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit: 1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB First 1: MSB First
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.
With 0 set, transmit data is output without inversion.
With 1 set, inverted transmit data is output.
The setting of this bit is valid for all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

Sets the input polarity for UART communication.
With 0 set, receive data is input without inversion.
With 1 set, receive data is input with inversion.
The setting of this bit is valid for all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of UTPS bit.
When setting this bit to “1” and expansion bit reception ((with expansion bit comparison) or (with data comparison)) is performed, set the inverse of the expected value to the UEBDL bit in the

RLN3nLUOR1 register and RLN3nLIDB register to enable comparison of the inverted values of the received values.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to "00_B", data is transmitted/received without the parity.

  [Transmission]

  A parity bit is not added to transmit data.

  [Reception]

  Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to "01_B", data is transmitted/received with the even parity.

  [Transmission]

  If the number of 1s in transmit data is odd, "1" is added to the parity bit. If the number of 1s in transmit data is even, "0" is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to "10_B", data is transmitted/received with 0 parity.

  [Transmission]

  Regardless of the number of 1s in transmit data, "0" is added to the parity bit.

  [Reception]

  The value of the parity bit is not judged. Therefore, no parity error occurs.

- When these bits are set to "11_B", data is transmitted/received with the odd parity.

  [Transmission]

  If the number of 1s in transmit data is odd, "0" is added to the parity bit. If the number of 1s in transmit data is even, "1" is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.
With 0 set, stop bit length of 1 bit is selected.
With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.
With 0 set, LSB first is selected.
With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.
With 0 set, the character length is 8 bits.
With 1 set, the character length is 7 bits.
When the character length of one frame is 9 bits (the UEBE bit in the RLN3nLUOR1 register is 1), the setting of this bit is invalid.

21.3.4.5 RLN3nLSC — UART Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 21.55 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frames when transmitting data from the UART buffer. 0 to 3 Tbits can be set.
Set IBS[1:0] bits to "00_B" when UART buffer is not used.

When data is transferred from the UART transmission data register (RLN3nLUTDR) and the UART wait transmission data register (RLN3nLUWTDR), the setting of these bits is ignored. Set these bits to "00_B".

21.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Table 21.56 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 21.4.5.5, Error Status**.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.
With 0 set, the overrun error is not detected.
With 1 set, the overrun error is detected.
When this bit is set to 1, the detection result is reflected in the OER flag in the RLN3nLEST register.
For details on the overrun error, see **Section 21.4.5.5, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
In full-duplex communication, do not set this bit to “1”.
Do not set this register when the NSPB bits in the RLN3nLWBR register are 0101_B (6 samplings) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filtering is enabled).

For details on the bit error, see **Section 21.4.5.5, Error Status**.

21.3.4.7 RLN3nLCUC — UART Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 21.57 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to reset mode or canceling reset mode.
With 0 set, reset mode.
With 1 set, reset mode is canceled.

21.3.4.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R

Table 21.58 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.
Only 1 can be written to this bit; 0 cannot be written.
Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enabled) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).
Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. Moreover, this bit is automatically cleared to 0 upon transition to reset mode.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
When writing 1 to this bit while the UTSW bit in the RLN3nLDFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the stop bit of reception data is completed), write only during the reception of stop bit.

21.3.4.9 RLN3nLMST — UART Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.59 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

21.3.4.10 RLN3nLST — UART Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 21.60 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	URS	Reception Status Flag 0: Reception is stopped. 1: Reception is started.
4	UTS	Transmission Status Flag 0: Transmission is stopped. 1: Transmission is started.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to "00_H" upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains "00_H". To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

URS Flag (Reception Status Flag)

At the start of the reception, this flag is set to 1.
The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.
The reception is ended under the following conditions.

- Sampling point of the first bit of the stop bits

UTS Flag (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.
The transmission is started under the following conditions.

- When transmission data is set to the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the end of transmission.
The transmission is ended under the following conditions.

- When transmission of data set in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not set
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or upon ID matching (when at least one of the flags of the RLN3nLEST register is 1). At this time, an interrupt request for RLIN3n status is generated. However, when this bit is 1, an interrupt is not generated upon detection of an error, detection of an expansion bit, or upon ID matching. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

FTC Flag (Successful UART Buffer Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written.
Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated. Write 0 to clear this flag.

21.3.4.11 RLN3nLEST — UART Error Status Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 21.61 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Match Flag 0: The receive data does not match the ID value. 1: The receive data matches the ID value.
4	EXBT	Expansion Bit Detection Flag 0: Expansion bit has not been detected. 1: Expansion bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is retained. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written. This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

IDMT Flag (ID Match Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.

The IDMT flag is set to 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)
- The UEBDCE bit in the RLN3nLUOR1 register is 1 (expansion bit/data comparison enabled)

– The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register match.
– The 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

EXBT Flag (Expansion Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches with the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.
Write 0 to clear this flag.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled).
Write 0 to clear this flag.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value it had before 1 is written.
The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled).
Write 0 to clear this flag.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value it had before 1 is written.
The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).
Write 0 to clear this flag.

21.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	UTSW	—	MDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 21.62 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MDL[3:0]	UART Buffer Data Length Select b3 b0 0 0 0 0: 9 data 0 0 0 1: 1 data 0 0 1 0: 2 data 0 0 1 1: 3 data 0 1 0 0: 4 data 0 1 0 1: 5 data 0 1 1 0: 6 data 0 1 1 1: 7 data 1 0 0 0: 8 data 1 0 0 1: 9 data Settings other than the above are prohibited.

UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.
With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.
With 1 set, transmission is started after the completion of the stop bit reception.
Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.
This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).
Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.
Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

21.3.4.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.63 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	ID value that is referred for the expansion bit data comparison is set

ID Bits (ID)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

21.3.4.14 RLN3nLUDB0 — UART Data Buffer 0 Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 17_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	UDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.64 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H

When transmitting 9-byte data from the UART buffer (the RLN3nLDFC.MDL bit is "0_H" or "9_H"), set the first data to be transmitted.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 21.65, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format, shows the bit arrangement according to the settings for communication format.

For details about the UART buffer, see **Section 21.4.5.1(2), UART Buffer Transmission**.

Table 21.65 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—*1	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

Note 1. In the case of 7-bit data length, write the value after reset.

21.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.66 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H

This register specifies the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 21.67, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 21.4.5.1(2), UART Buffer Transmission**.

Table 21.67 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

	RLN3nLDBRb							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—*1	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

Note 1. In the case of 7-bit data length, write the value after reset.

21.3.4.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UROE	UTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 21.68 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	UROE	Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.
In LIN reset mode, this register cannot be written to.
In LIN reset mode, the register retains 00_H.

UROE Bit (Reception Enable)

The UROE bit enables or disables reception.
With 0 set, reception is disabled.
With 1 set, reception is enabled.
Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. However, the transmit operation is also suspended at this time.

Do not set this bit to 1 when data transmission from the UART buffer is in progress.

UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.
With 0 set, transmission is disabled.
With 1 set, transmission is enabled.
Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. However, the receive operation is also suspended at this time.

21.3.4.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 21_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.69 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison.
3	UTIGTS	Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).
With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.
With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.
Do not set this bit to 1 when the UEBDCE bit is 1 (data comparison after expansion bit is detected).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit sets the generation timing of the transmission interrupt.
With 0 set, the transmission interrupt is generated at the start of transmission.
With 1 set, the transmission interrupt is generated at the completion of transmission.
When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.
When transmission from the UART buffer is performed with 1 set, the transmission interrupt is

generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.
With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.
With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).
Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).
Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).
With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.
With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.
With 0 set, expansion bit operation is disabled.
With 1 set, expansion bit operation is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

21.3.4.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read or written in 16-bit units.
RLN3nLUTDRL register can be read or written in 8-bit units.
RLN3nLUTDRH register can be read or written in 8-bit units.

Address: RLN3nLUTDR: <RLIN3n_base> + 24_H
RLN3nLUTDRL: <RLIN3n_base> + 24_H
RLN3nLUTDRH: <RLIN3n_base> + 25_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.70 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UTD[8:0]	Sets the data to be transmitted. Setting range: 000_H to $1FF_H$

The RLN3nLUTDR register sets the data to be transmitted from the transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 21.71 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	—*1	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—*1	—*1	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—*1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—*1	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

Note 1. In the case of 7-bit or 8-bit data length, write the value after reset.

21.3.4.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register is a read-only register that can be read in 16-bit units.
RLN3nLURDRL register is a read-only register that can be read in 8-bit units.
RLN3nLURDRH register is a read-only register that can be read in 8-bit units.

Address: RLN3nLURDR: <RLIN3n_base> + 26_H
RLN3nLURDRL: <RLIN3n_base> + 26_H
RLN3nLURDRH: <RLIN3n_base> + 27_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	URD [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.72 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8 to 0	URD [8:0]	Stores the received data Setting range: 000_H to $1FF_H$

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the reception data is stored in this register and can be read out.

This register is updated when the stop bit of the receive data is received.

This register is also updated when an error is caused by the parity or stop bit.

However, the value of this register is not updated upon occurrence of an overrun error when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun error detection disabled).

Read this register upon occurrence of a receive error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8 bits.

However, during expansion bit use (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 21.73 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR								
	b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

21.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access: RLN3nLUWTDR register can be read or written in 16-bit units.
RLN3nLUWTDRL register can be read or written in 8-bit units.
RLN3nLUWTDRH register can be read or written in 8-bit units.

Address: RLN3nLUWTDR: <RLIN3n_base> + 28_H
RLN3nLUWTDRL: <RLIN3n_base> + 28_H
RLN3nLUWTDRH: <RLIN3n_base> + 29_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UWTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.74 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000_H to $1FF_H$

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

The user should write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the RLN3nLUTDR register value is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 21.75 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—*1	—*1	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—*1	—*1	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—*1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—*1	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

Note 1. In the case of 7-bit or 8-bit data length, write the value after reset.

21.4 Operation

21.4.1 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt
- RLIN3n interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the RLIN3n interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt source.

Table 21.76 lists the sources for each interrupt.

Table 21.76 Interrupt Sources

		LIOS bit in RLN3nLMD register is 0	LIOS bit in RLN3nLMD register is 1*1		
		RLIN3n Interrupt	RLIN3n Transmission Interrupt	RLIN3n Successful Reception Interrupt	RLIN3n Status Interrupt
LIN mode	LIN master mode	• Successful frame transmission • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful response reception • Successful wake-up reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
	LIN slave mode	• Successful response transmission • Successful response reception • Successful wake-up transmission • Successful wake-up reception • Successful header reception • Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error	• Successful response transmission • Successful wake-up transmission	• Successful response reception • Successful wake-up reception • Successful header reception	• Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error
UART mode		—	• Transmission start/ successful transmission	• Successful reception • Expansion bit mismatch	• Bit error • Overrun error • Framing error • Expansion bit match • ID match • Parity error

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is 1.

21.4.2 Modes

The LIN/UART interface provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode
  - LIN master mode
  - LIN slave mode [auto baud rate]
  - LIN slave mode [fixed baud rate]
- UART mode
- LIN self-test mode

Figure 21.2 shows mode transitions. **Table 21.77** describes mode transition conditions. **Table 21.78** lists operations available in each mode.

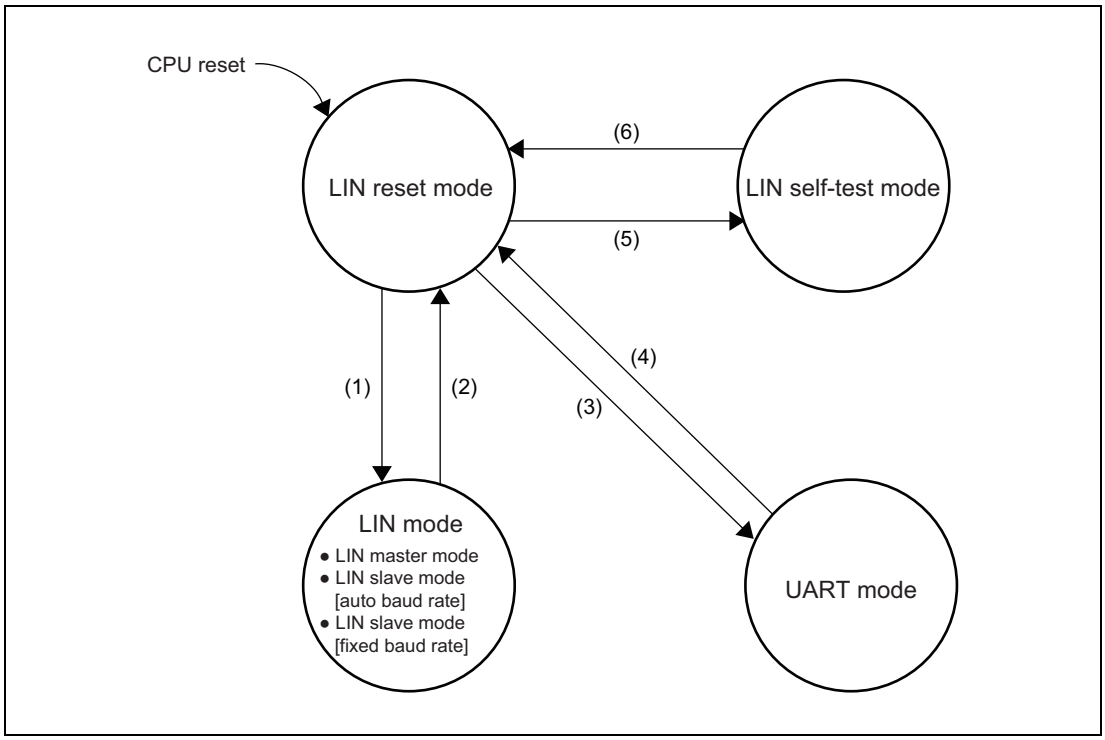


Figure 21.2 Mode Transitions

Table 21.77 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode	
			• LIN master mode	• LMD bits in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			• LIN slave mode [auto baud rate]	• LMD bits in RLN3nLMD register = 10_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			• LIN slave mode [fixed baud rate]	• LMD bits in RLN3nLMD register = 11_B and OM1 and OM0 bits of RLN3nLCUC register = 01_B or 11_B
(2)	LIN mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(3)	LIN reset mode	→	UART mode	LMD bits in RLN3nLMD register = 01_B and OM0 bit in RLN3nLCUC register = 1_B
(4)	UART mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(5)	LIN reset mode	→	LIN self-test mode	See **Section 21.4.6, LIN Self-Test Mode**.
(6)	LIN self-test mode	→	LIN reset mode	See **Section 21.4.6, LIN Self-Test Mode**.

Table 21.78 Operations Available in Each Mode

LIN Mode			
LIN Master Mode	**LIN Slave Mode [auto baud rate] LIN Slave Mode [fixed baud rate]**	**UART Mode**	**LIN Self-Test Mode**
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Header reception Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception Error detection	Self test

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 21.4.6, LIN Self-Test Mode**.

21.4.3 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

21.4.4 LIN Mode

LIN mode can operate in the following submodes: LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.

The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. The baud rate can be set to 1 kbps to 20 kbps.

Set the LPRS[2:0] bits in the RLN3nLWBR register so that the prescaler clock (with the frequency of the LIN communication clock source divided by the prescaler) becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz*1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011_B" (four samplings).

LIN slave mode [fixed baud rate] allows automatic detection of the break field, sync field, and ID field at a baud rate that is set in advance by the baud rate generator.

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 10_B (LIN slave mode [auto baud rate]) and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [auto baud rate]; and setting the LMD bits in the RLN3nLMD register to 11_B (LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B or 11_B.

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 21.3 shows the transition of operation modes. **Table 21.79** describes the transition conditions of operation modes.

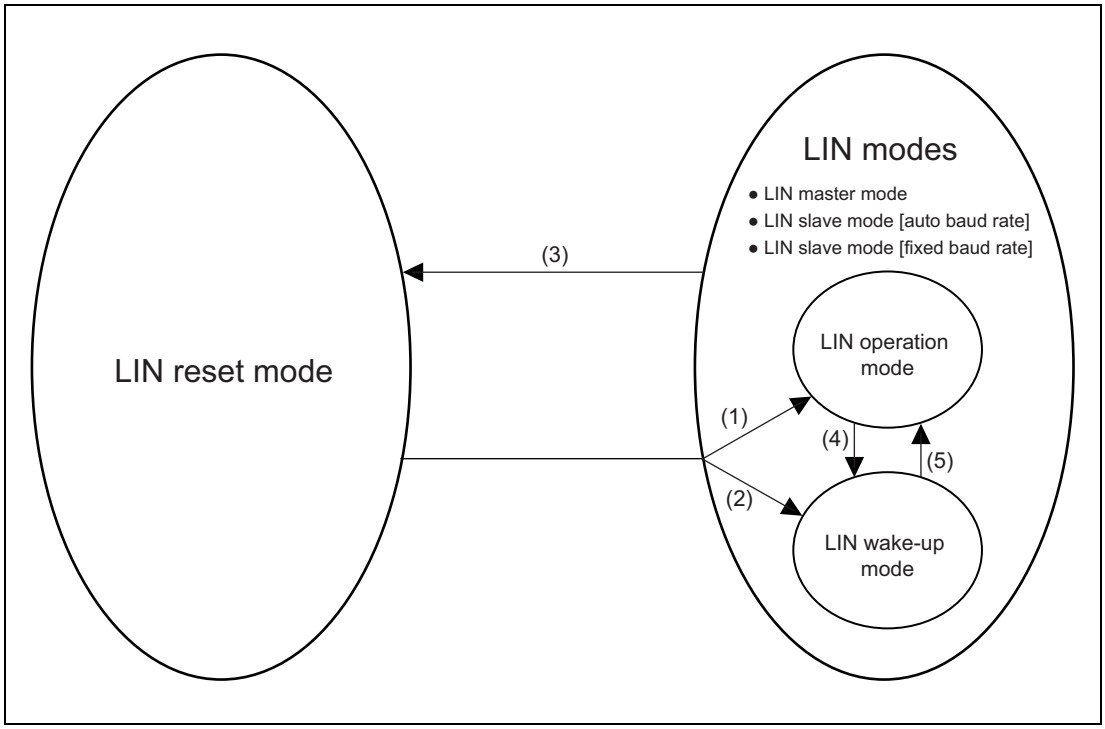


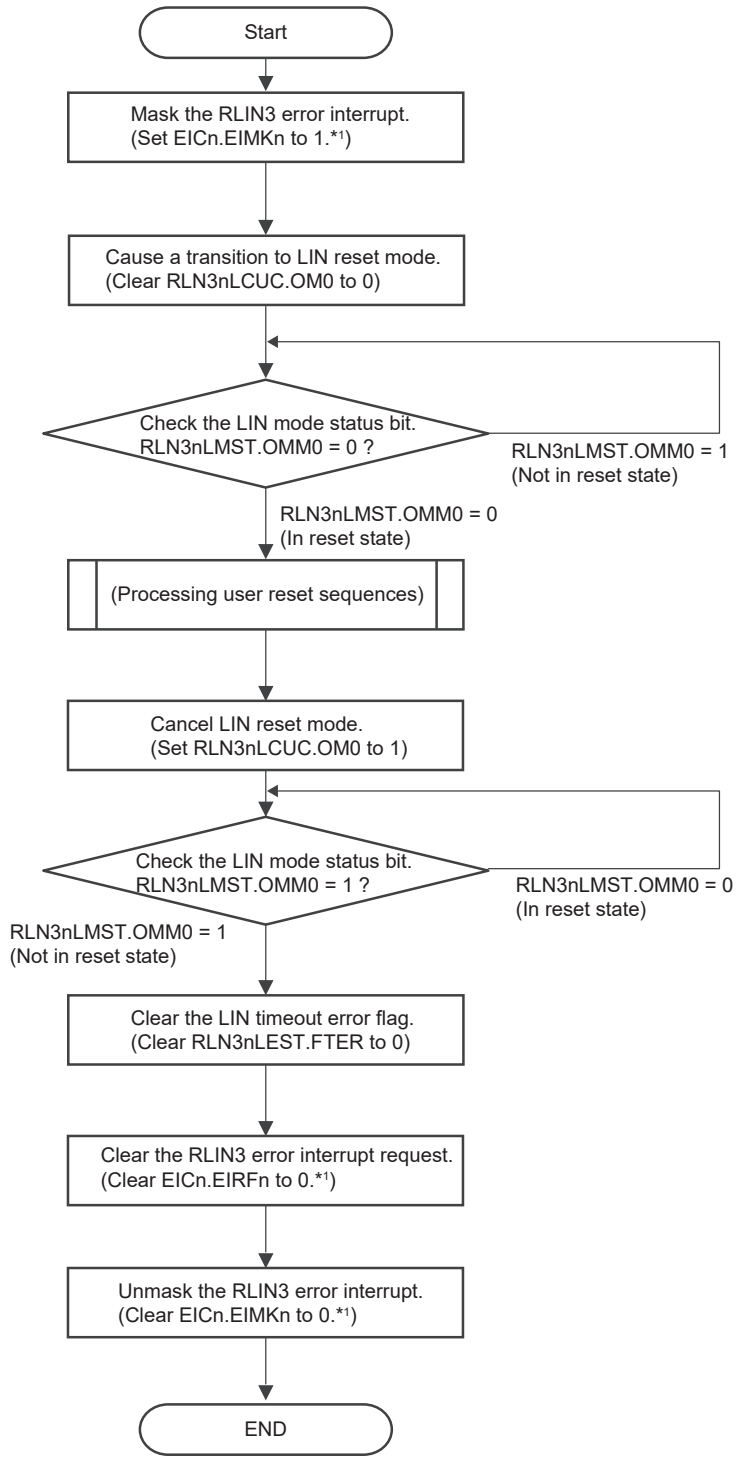
Figure 21.3 Transition of Operation Modes

Table 21.79 Transition Condition for Operation Mode

	Operation Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode • LIN operation mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 11_B
(2)	LIN reset mode	→	LIN mode • LIN wake-up mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B
(3) *2	LIN mode • LIN operation mode • LIN wake-up mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(4) *1	LIN mode • LIN operation mode	→	LIN mode • LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = 01_B
(5) *1	LIN mode • LIN wake-up mode	→	LIN mode • LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = 11_B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

Note 2. In LIN mode, if RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.FTERE = 1 or RLN3nLEDE.TERE = 1), users should take the procedure shown in **Figure 21.4**. The procedure is to protect a RLIN3 error interrupt and clear a RLIN3 error interrupt request.



Note 1. This “n” value means interrupt number. See **Section 21.1.4, Interrupt Requests and Error Notifications** and confirm interrupt numbers of RLIN3n interrupt or RLIN3n status interrupt (n = 0 to 23) in **Table 21.7** in accordance with the value of RLN3nLMD.LIOS. For EICn, refer to **Section 6, Interrupts**.

Figure 21.4 LIN reset sequence by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 01_B.

21.4.4.1 LIN Master Mode

(1) Header Transmission

Figure 21.5 shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 21.80** describes the processing in header transmission.

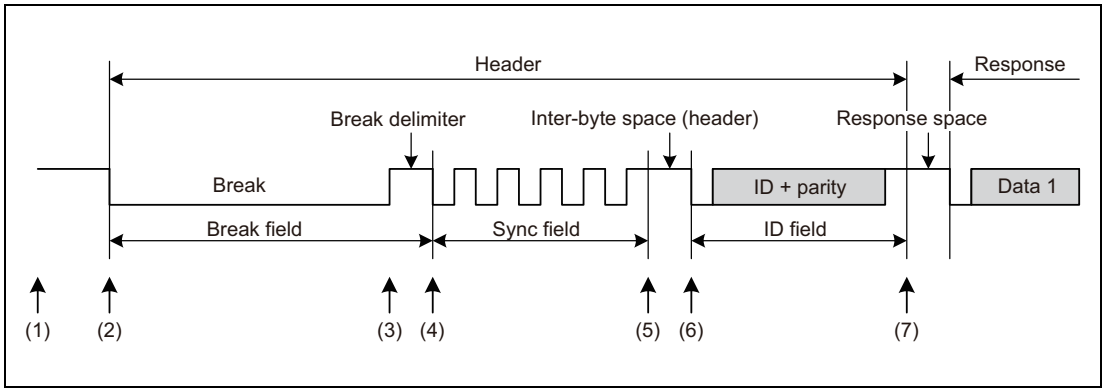


Figure 21.5 Operation in Header Transmission

Table 21.80 Processing in Header Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software (idle)
(2)	Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55_H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets a successful header transmission flag.

NOTE

For information about error detection conditions, see **Section 21.4.4.7, Error Status**.

(2) Response Transmission

Figure 21.6 shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 21.81** describes the processing in response transmission.

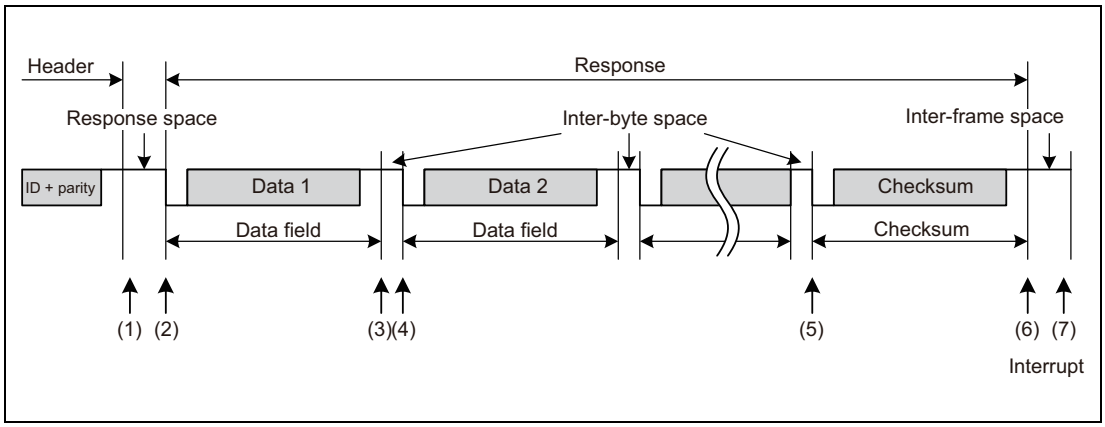


Figure 21.6 Operation in Response Transmission

Table 21.81 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	[When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/ reception started) [When not in frame separate mode] • Waits for an interrupt request	[When in frame separate mode] • Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. • When the bit is set to 1, sends a response space. [When not in frame separate mode] • Sends a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception is halted).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection conditions, see **Section 21.4.4.7, Error Status**.

(3) Response Reception

Figure 21.7 shows the operation of the LIN/UART interface (LIN master mode) in response reception. **Table 21.82** describes the processing in response reception.

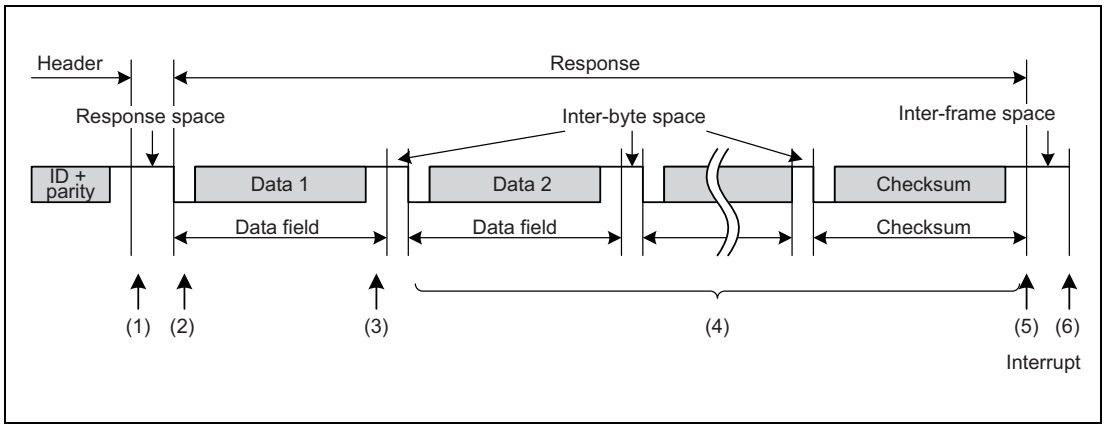


Figure 21.7 Operation in Response Reception

Table 21.82 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	Waits for an interrupt request (no processing)	Waits for detection of a start bit.
(2)		Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful frame/wake-up reception flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 21.4.4.7, Error Status**.

21.4.4.2 LIN Slave Mode

(1) Header Reception

Figure 21.8 shows the operation of the LIN/UART interface (LIN slave mode) in header reception.
Table 21.83 provides processing in header reception.

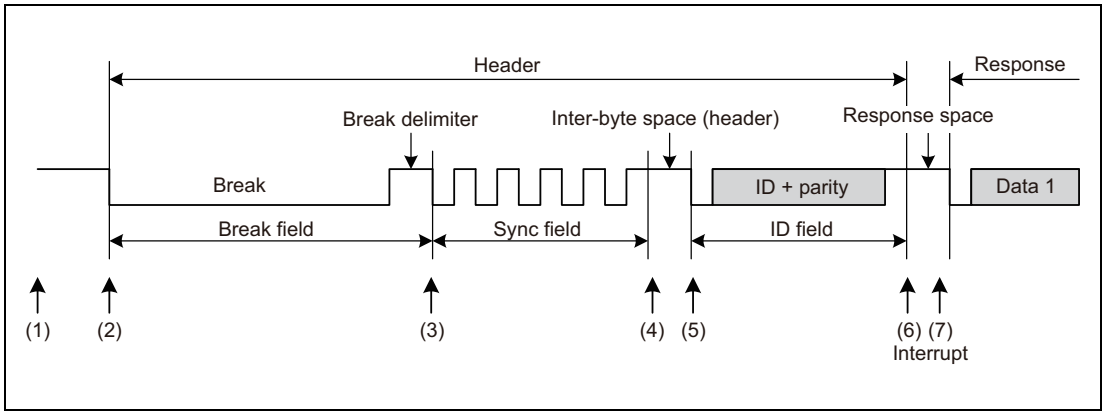


Figure 21.8 Operation in Header Reception

Table 21.83 Processing in Header Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN slave mode: LIN operation mode • Sets the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software.
(2)	Waits for an interrupt request.	Waits for detection of break field
(3)		Detects a break field. (LIN slave mode [fixed baud rate]). For details about the break field detection timing in the case of LIN slave mode [auto baud rate], see [Auto Baud Rate Correction Function].)
(4)		• Detects a sync field (55_H) • Sets the baud rate generator (in the case of LIN slave mode [auto baud rate]) • Clears the no-response request bit (LNRR bit).
(5)		• Receives an ID field. • Checks an ID parity bit
(6)		Sets a header reception complete flag.
(7)	• Checks the RLN3nLST register, and clears flags. • Checks the RLN3nLIDB register, and prepares a response.	• Completes a header reception process. • Waits for a response request.

NOTE

The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred. For information about error detection conditions, see **Section 21.4.4.7, Error Status**.

[Auto Baud Rate Correction Function]

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first "Low level" width is 10 times (if the BLT bit of the RLN3nLBFC register is "0") or 11 times (if the BLT bit of the RLN3nLBFC register is "1") or greater than the bit width calculated from the average of the starting 2 bits (the period of the consecutive falling edges from the beginning of the sync field) of the sync field, the system concludes that the detection of break field was successful, the system verifies that the data in the sync field is 55_H. If the data in the sync field is indeed 55_H and the system judges that sync field reception was successful, the system automatically sets the baud rate correction result to the RLN3nLBRP01 register.

If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.

On the other hand, if the data in the sync field is not 55_H and the system judges that sync field reception failed, the system sets the sync field error flag and an error interrupt is generated. In that case, baud rate correction is not performed and the LIN/UART interface waits for the detection of the next break field (low level).

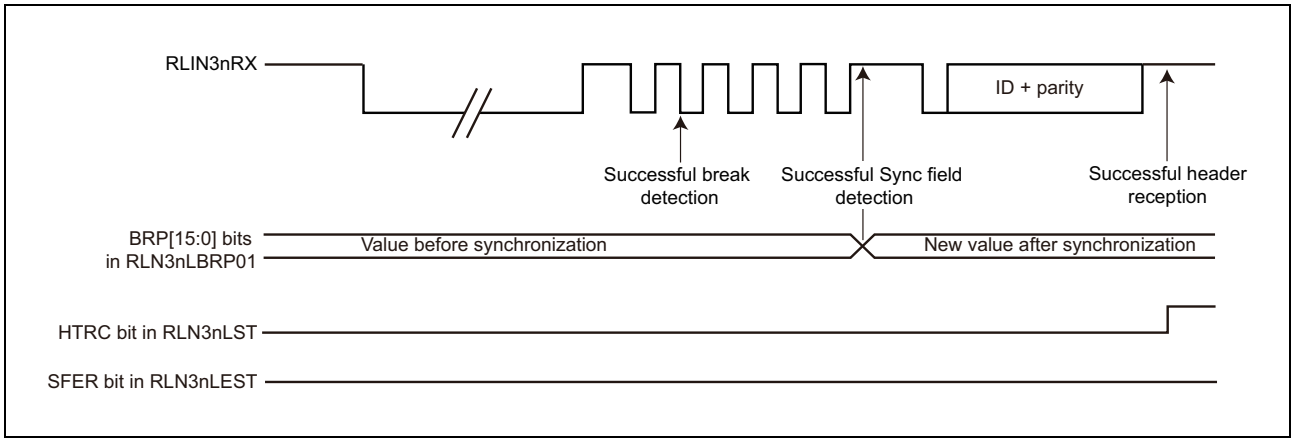


Figure 21.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

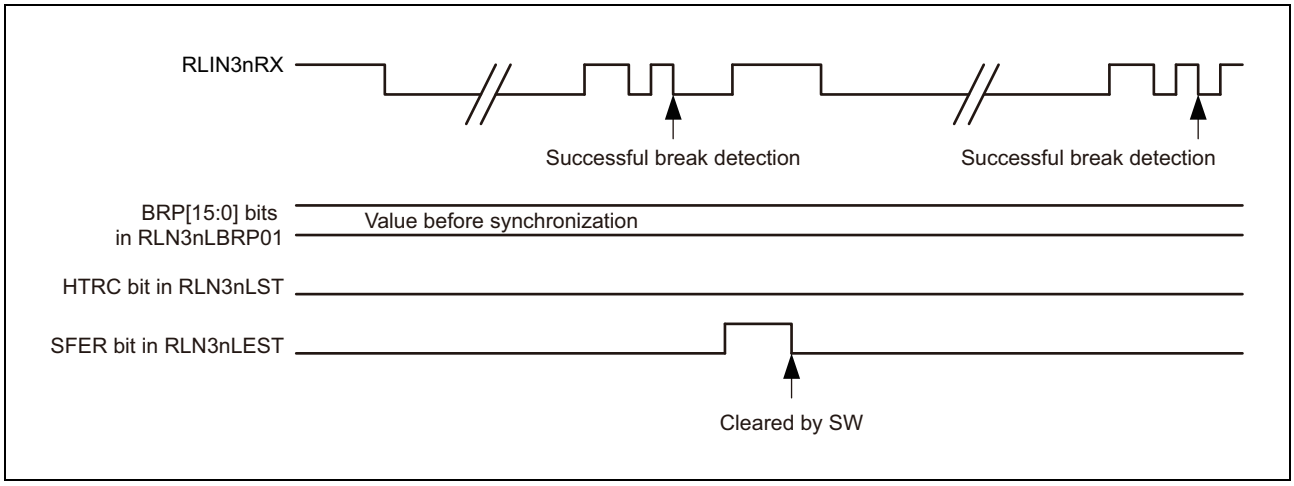


Figure 21.10 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

(2) Response Transmission

Figure 21.11 shows the operation of the LIN/UART interface (LIN slave mode) in response transmission. **Table 21.84** describes the processing in response transmission.

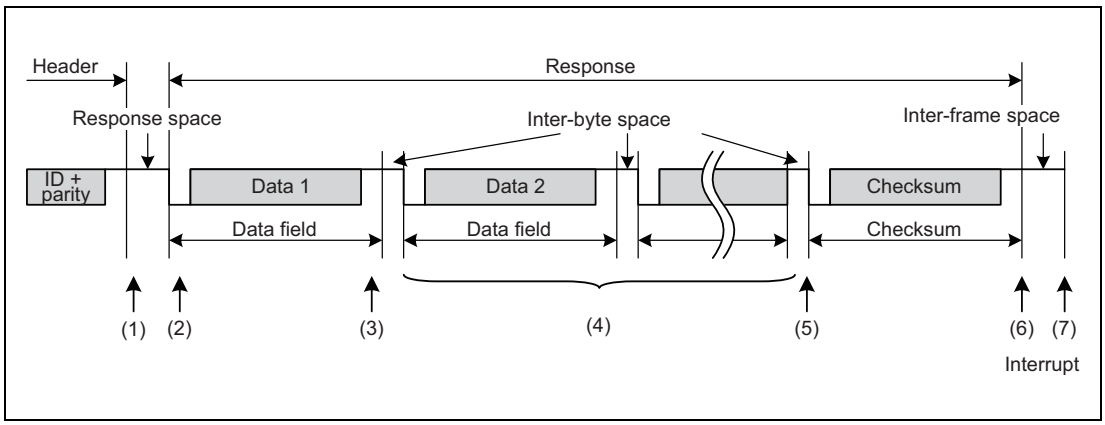


Figure 21.11 Operation in Response Transmission

Table 21.84 Processing in Response Transmission (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the RLN3nLDFC register. • Sets the RLN3nLDBRb registers.(b = 1 to 8) • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/ reception started)	• Waits for the setting of the RTS or LNRR bit of the RLN3nLTRC register by software • Transmits the response space after the RTS bit of the RLN3nLTRC register is set to 1

Table 21.84 Processing in Response Transmission (2/2)

	Software Processing	LIN/UART Interface Processing
(2)	Waits for an interrupt request.	Transmits data 1.
(3)		Transmits the inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets the successful response/wake-up transmission flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7)	• Processing after communication Checks the RLN3nLST, RLIN3nLRSS registers, and clears flags.	• Completes the response transmission process. • Waits for a new break.

NOTE

- For information about error detection, see **Section 21.4.4.7, Error Status**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.

(3) Response Reception

Figure 21.12 shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 21.85** describes the processing in response reception.

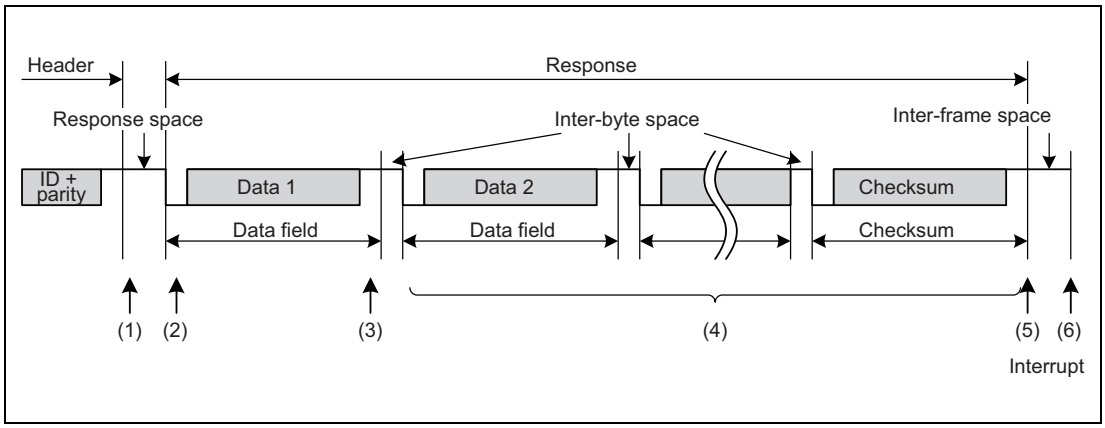


Figure 21.12 Operation in Response Reception

Table 21.85 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the RLN3nLDFC register. • Sets the response transmission/ reception start bit (RTS bit) to 1.	• Waits for the setting by software of the response transmission/ reception start bit (RTS bit) or the no-response request bit (LNRR bit). • Waits for detection of the start bit.
(2)	Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets a successful response/wake-up reception flag or error flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	• Completes the response process. • Waits for a new break.

NOTE

- For information about error detection conditions, see **Section 21.4.4.7, Error Status**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.

(4) No-response Request

Figure 21.13 shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 21.86** shows the processing that occurs when no response is requested.

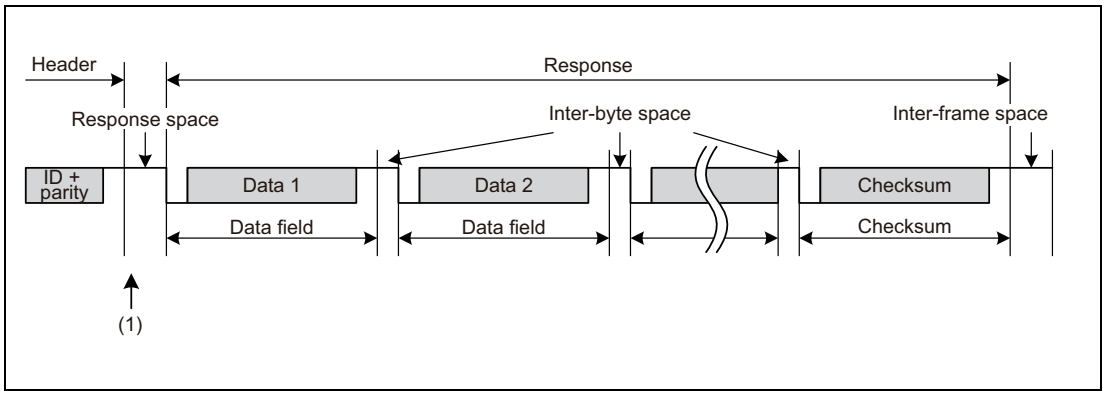


Figure 21.13 Operation when No Response is Requested

Table 21.86 Processing when No Response is Requested

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the no-response request bit (LNRR bit) to 1.	• Waits for setting of the no-response request bit (LNRR bit) by software • Completes the frame reception process • Waits for a new break

21.4.4.3 Data Transmission/Reception

(1) Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 21.4.4.7, Error Status**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be 16/fLIN, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

In LIN slave mode [auto baud rate], if 1 Tbit is generated to be 4/fLIN, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be 8/fLIN, the sampling point for received data is at the 7th clock cycle (87.5% position).

Figure 21.14 shows an example of data transmission timing.

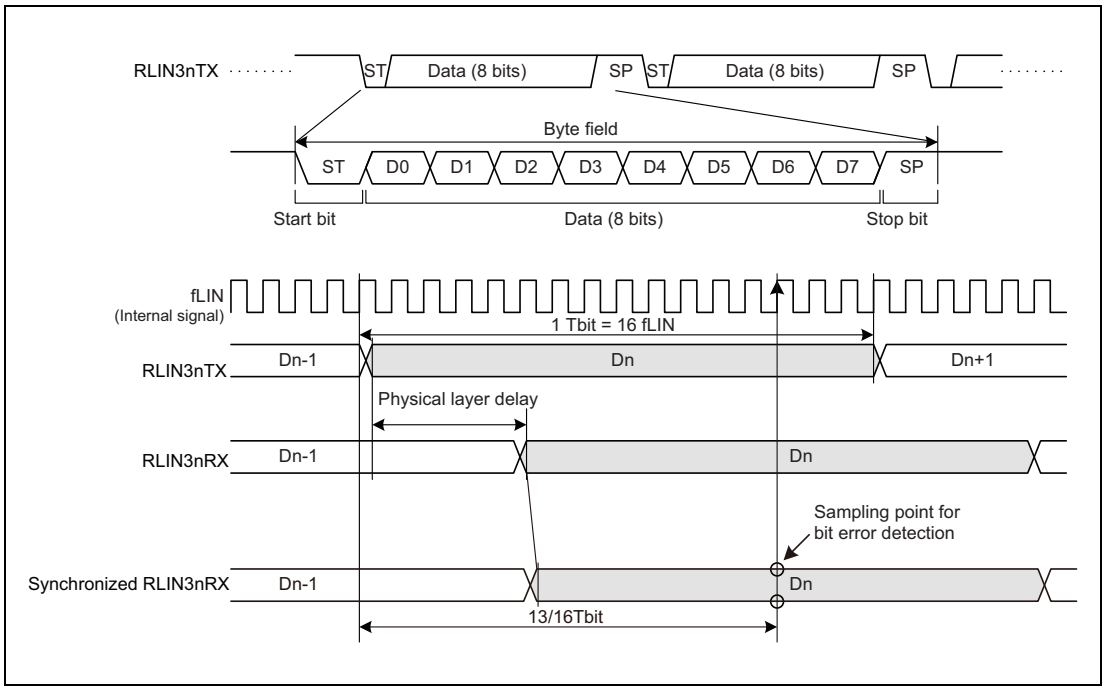


Figure 21.14 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

(2) Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function for reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

Figure 21.15 shows an example of data reception timing.

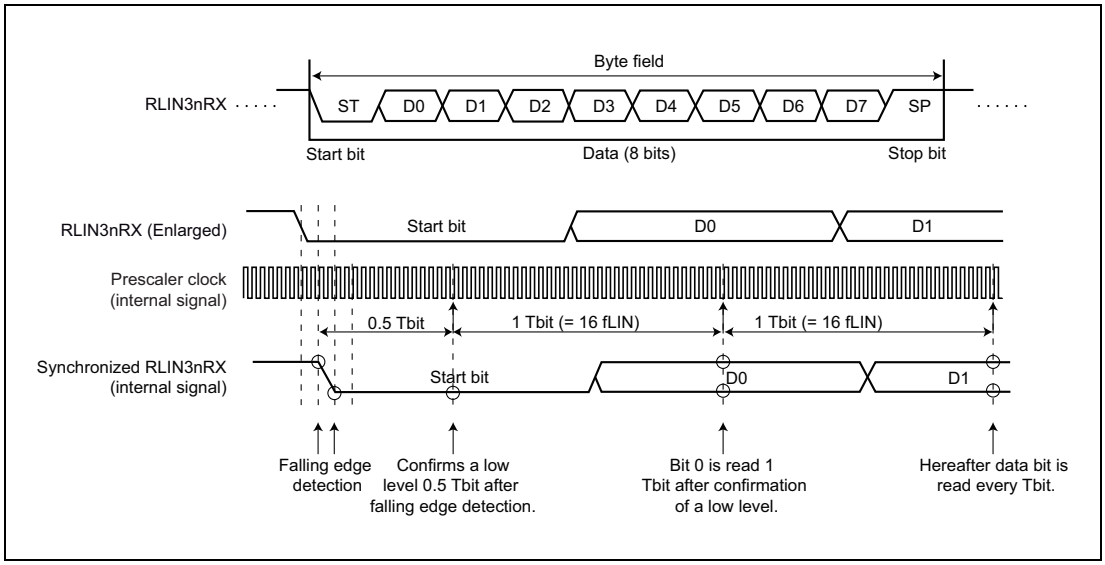


Figure 21.15 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

21.4.4.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

(1) Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 21.16 shows the LIN transmission processing and the required buffers.

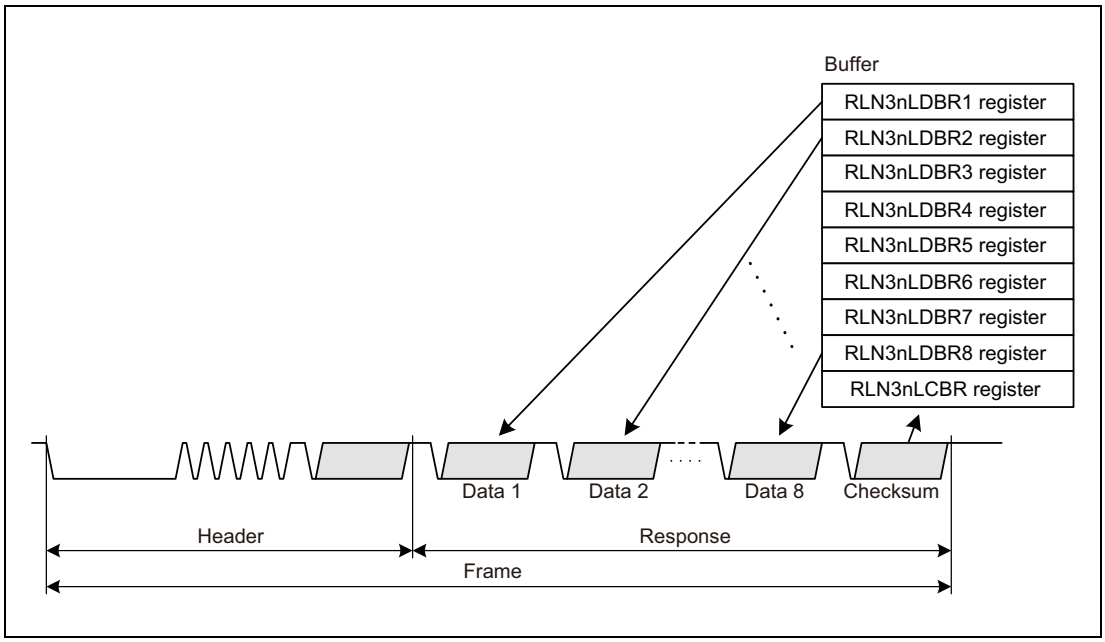


Figure 21.16 LIN Transmission Processing and Required Buffer

[Frame Separate Mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or more in LIN master mode.

(2) Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 21.17 shows the LIN reception processing and the required buffers.

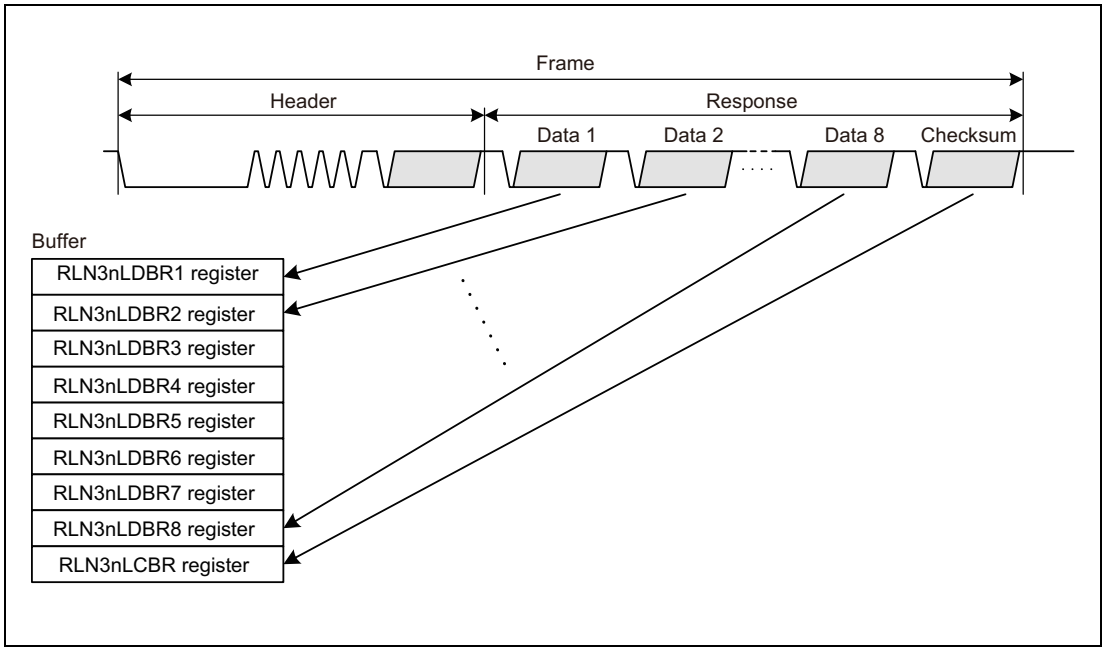


Figure 21.17 LIN Reception Processing and Required Buffer

[Reception of Data 1]

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

(3) Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or more can also be transmitted and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is longer than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the last data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the last data group. If it is the last data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the last data group), and a checksum should be appended to the last data group.

By changing the RFDL bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register to 1 (frame separate mode).

NOTE

In LIN slave mode, the LIN/UART interface can detect a new break field during the transmission or reception of a response.

21.4.4.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

(1) Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission), or setting the RCDS bit in the RLN3nLDFC register to 1 (LIN slave mode: response transmission), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level width is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame response or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wakeup transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).
When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 21.18 shows the wake-up transmission timing.

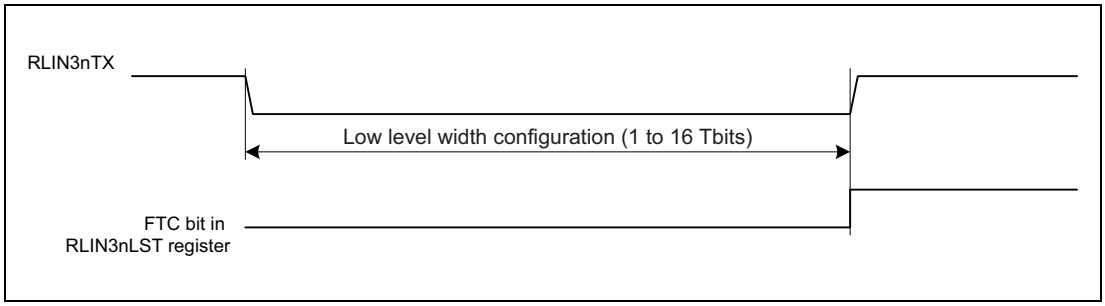


Figure 21.18 Wake-up Transmission Timing

For wake-up retransmission in LIN slave mode, refer to **Section 21.4.4.9, Wake-up Processing for Retransmission in LIN Slave Mode**.

(2) Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation can be executed without changing the baud rate generator setting when switching between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 sets the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200bps while fa is selected, an input signal with a low-level width of 130 us or longer can be measured in LIN wake-up mode regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using the wake-up reception function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), or the RCDS bit in the RLN3nLDFC register to 0 (LIN slave mode: response reception), and then set the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame response/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for successful RLIN3n reception is generated.

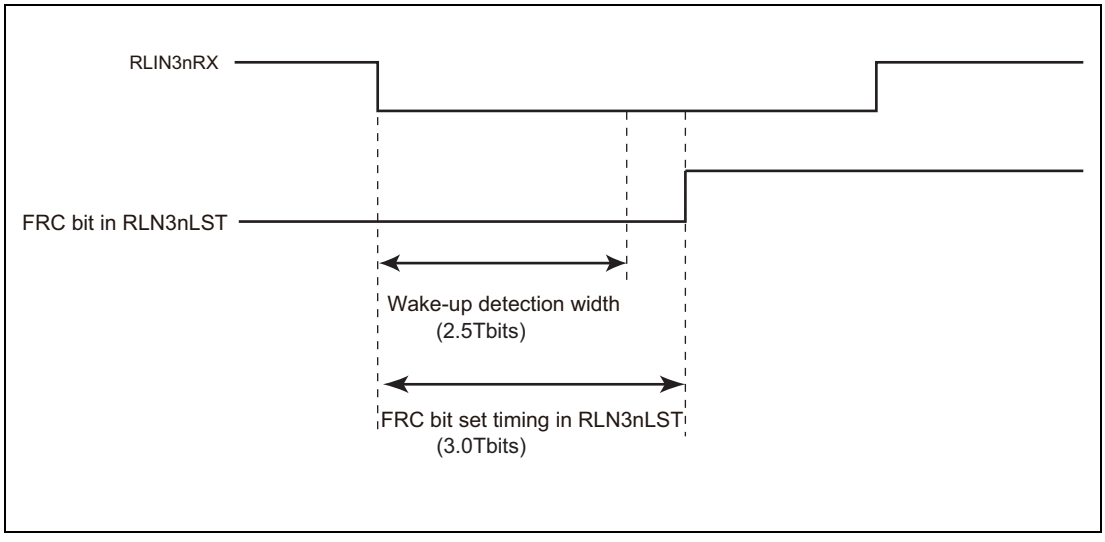


Figure 21.19 Input Signal Low level Count Function

(3) Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, though a collision of wakeup signals is not detected by the LIN/UART interface.

21.4.4.6 Status

In LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, and successful header transmission/reception, can generate interrupt requests.

Table 21.87 shows the types of statuses available in LIN master mode. **Table 21.88** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 21.87 Types of Statuses in LIN Master Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Frame/wake-up reception end	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software*1 • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header transmission end	When a header field is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

Table 21.88 Types of Statuses in LIN Slave Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN–reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit of the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	• After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Response/ wake-up transmission end	When a response field, a wake-up signal, or a data group is transmitted successfully.	• When cleared by software • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Response/ wake-up reception end	When a response field, a wake-up signal, or a data group is received successfully.	• When cleared by software • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When cleared by software*1 • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RCDS bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte for each data group is received successfully.*2	• When cleared by software • Transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header reception end	When a header field is received successfully.	• When cleared by software • Transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√
Break field reception	When break field is received successfully	• When cleared by software • Transition to LIN reset mode	LIN wake-up mode	BRKC flag in RLN3nLBSS register	—
Sync field reception	When sync field is received successfully	• When cleared by software • Transition to LIN reset mode	LIN wake-up mode	SYCC flag in RLN3nLBSS register	—
Response space dominant detection	When a dominant level of 0.5 Tbit or more is detected from the completion of the header reception to the start of transmission	• When received a sync field • Transition to LIN reset mode	LIN operation mode	RSDD flag in RLN3nLRSS register	—

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

21.4.4.7 Error Status

(1) LIN Master Mode

(a) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

All error statuses are interrupt factors.

Table 21.89 shows the types of error statuses.

Table 21.89 Types of Error Statuses in LIN Master Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1*2	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	BER flag in RLN3nLEST register
Physical bus error	• LIN bus detected a high level when sending a break • LIN bus detected a low level when sending a break delimiter • LIN bus detected a high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time*3,*4	LIN operation mode	Cancel	Enabled	FTER flag in RLN3nLEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	—	Disabled	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: • After header transmission is complete, the first byte of receive data is received before a response transmission/reception request is set. • After the previous data group reception is complete, the first byte of receive data is received before a transmission/reception request for the next data group is set.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the CSM bit in the RLN3nLDFC register), and can be calculated from the following formula.
When the FSM bit in the RLN3nLDFC register is 1 (frame separation mode), the timeout time is that for eight bytes until the RTS bit of the RLN3nLTRC register is set. Once the RTS bit is set, the timeout time is re-set to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDFC register).

Note 4. If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.FTERE = 1), users should take the procedure shown in **Figure 21.4**.

[Frame timeout]
When classic checksum is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]
The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.
[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]

When an error is detected, the timeout error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set) when it is cleared by software, or at a transition to LIN reset mode.

(b) Target Time Area for LIN Error Detection

Figure 21.20 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

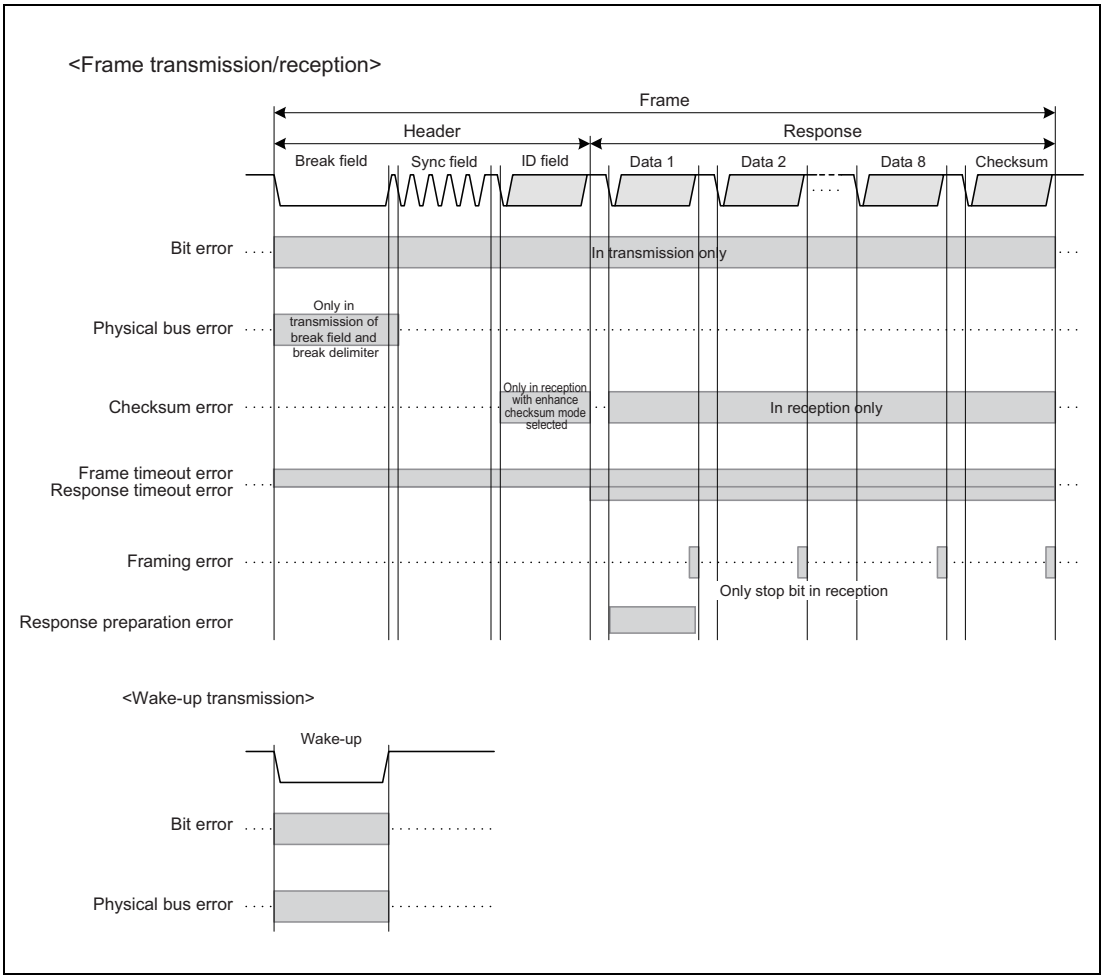


Figure 21.20 Target Time Area for LIN Error Detection (LIN Master Mode)

(2) LIN Slave Mode

(a) Types of Error Statuses

The LIN/UART interface can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

Table 21.90 shows the types of error statuses.

Table 21.90 Types of Error Statuses in LIN Slave Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match*1 *2	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	BER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time*3, *6	LIN operation mode	Cancel	Enabled	TER flag in RLN3nLEST register
Framing error	In frame reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Sync field error	If the width of the break low level is greater than the width set by the LBLT bit in the RLN3nLBFC register and the sync field is not 55_H	LIN operation mode	Cancel	Enabled*4	SFER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	—*5	Disabled	CSER flag in RLN3nLEST register
ID parity error	If the received ID parity bit does not match the value that is automatically calculated by the LIN/ UART interface	LIN operation mode	Cancel	Enabled	IPER flag in RLN3nLEST register
Response preparation error	• After the reception of a header, response preparation is not completed in time before the first byte of reception data is received. • In multi-byte response reception, the reception preparation for the next data group is not completed in time before the first byte of the next data group reception data is received.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the LCS bit in the RLN3nLDFC register), and this can be calculated according to the following formula. The timeout period until the RTS or LNRR bit of the RLN3nLTRC register is set is 8 data bytes. When the RTS bit is set, the timeout time is reset to the time based on the response field data length (RFDL[3:0] bits of the RLN3nLDFC register). When the LNRR bit is set, the timeout function stops.

[Frame timeout]
When classic checksum is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]
When an error is detected, timeout error detection function stops.

Note 4. Only reflection of the result to the SFER flag can be enabled/disabled. Error detection cannot be enabled/disabled.

Note 5. Checksum judgment is performed upon completion of response frame reception. In case of an error, the successful reception flag is not set to 1.

Note 6. If RLIN3 transitions to LIN reset mode by clearing RLN3nLCUC.OM0 to 0 when the timeout function is used (RLN3nLEDE.TERE = 1), users should take the procedure shown in **Figure 21.4**.

The error status is cleared by software or at a transition to LIN reset mode.

(b) Target Time Area for LIN Error Detection

Figure 21.21 shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

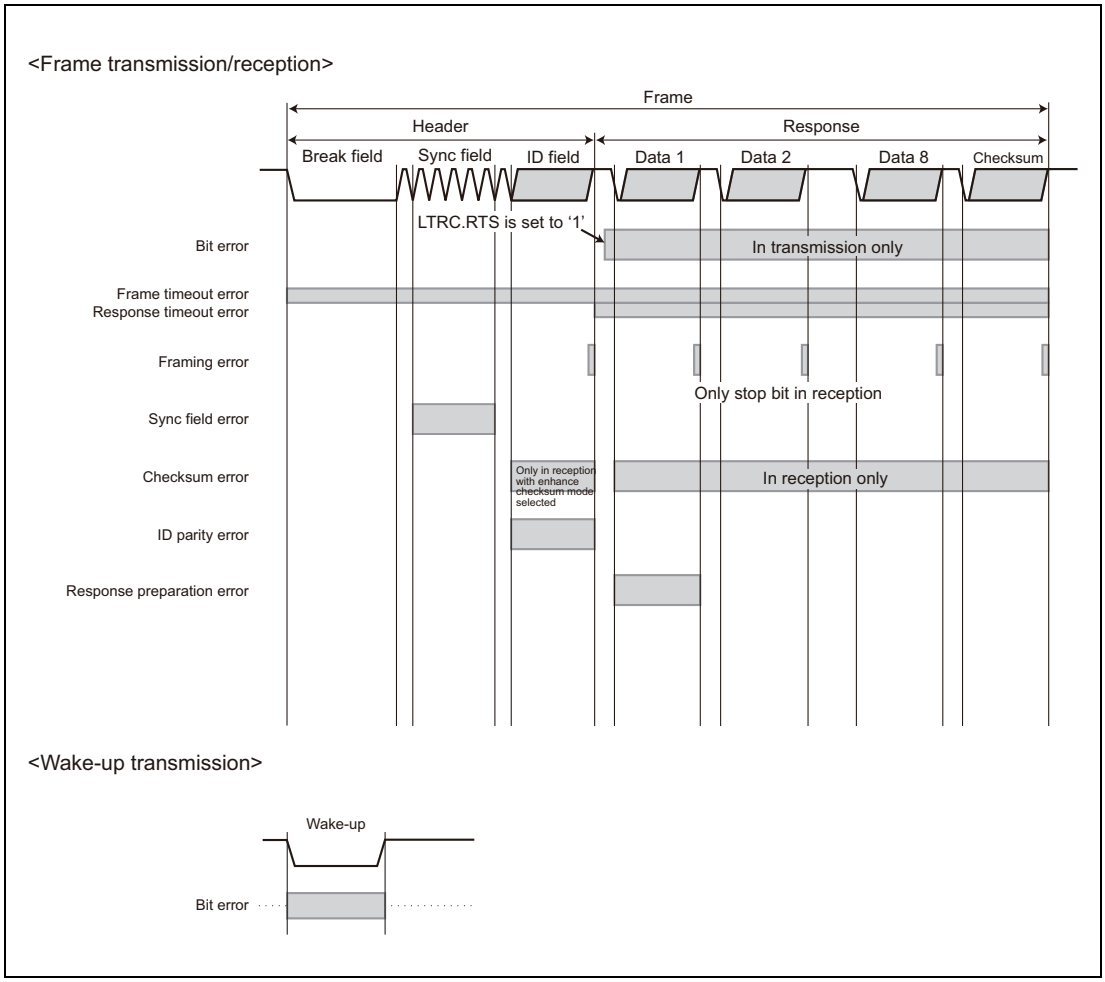


Figure 21.21 Target Time Area for LIN Error Detection (LIN Slave Mode)

21.4.4.8 Conflicts of Error Status

The table below shows the condition of error status bits that occurs at the end of single frame in LIN Master Mode.

The combination not listed in the table, condition does not occur.

Table 21.91 Conflicts of Error Status (LIN Master Mode)

Number of Conflict	kinds of Errors					
	BER	PBER	TER	FER	CSER	RPER
1	1	0	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0
	0	0	0	1	0	0
	0	0	0	0	1	0
	0	0	0	0	0	1
2	1	1	0	0	0	0
	1	0	1	0	0	0
	0	0	1	1	0	0
	0	0	1	0	1	0
	0	0	1	0	0	1
	0	0	0	1	1	0
	0	0	0	1	0	1
	0	0	0	0	1	1
3	0	0	1	1	1	0
	0	0	1	1	0	1
	0	0	1	0	1	1
	0	0	0	1	1	1
4	0	0	1	1	1	1
5	—	—	—	—	—	—
6	—	—	—	—	—	—

Note: 1: set error
0: no error

Table 21.92 Conflicts of Error Status (LIN Slave Mode)

Number of Conflict	kinds of Errors						
	BER	TER	FER	SFER	CSER	IPER	RPER
1	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	1	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	1	0	0
	0	0	0	0	0	1	0
	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0
	0	1	1	0	0	0	0
	0	1	0	1	0	0	0
	0	1	0	0	1	0	0
	0	1	0	0	0	1	0
	0	1	0	0	0	0	1
	0	0	1	0	1	0	0
	0	0	1	0	0	1	0
	0	0	1	0	0	0	1
	0	0	0	0	1	0	1
3	0	1	1	0	1	0	0
	0	1	1	0	0	1	0
	0	1	1	0	0	0	1
	0	1	0	0	1	0	1
	0	0	1	0	1	0	1
4	0	1	1	0	1	0	1
5	—	—	—	—	—	—	—
6	—	—	—	—	—	—	—
7	—	—	—	—	—	—	—

Note: 1: set error
0: no error

21.4.4.9 Wake-up Processing for Retransmission in LIN Slave Mode

The figure below illustrates flow of the wake-up processing in LIN slave mode.

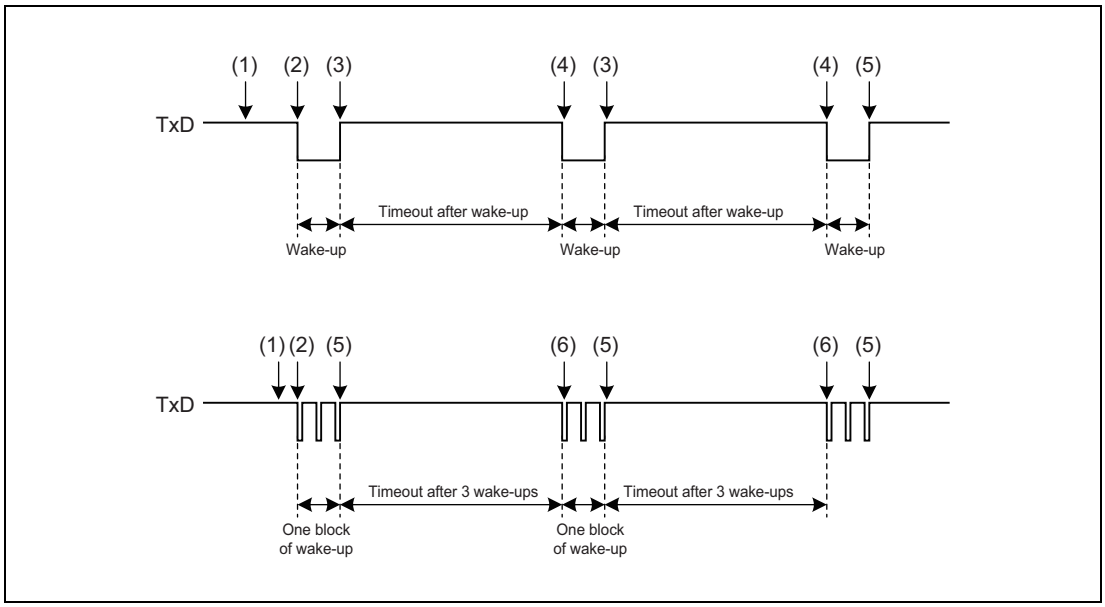


Figure 21.22 The Wake-up Processing with Retransmission in LIN Slave Mode

(1) LIN Break Field Reception flag and LIN Sync Field Reception flag are cleared by writing “0” to them.

(2) Transmitting a wake-up signal.

(3) Wait for wake-up signal transmission complete. When the wake-up transmission is aborted, LIN controller handles the error. When the wake-up transmission is successful, a timer for “timeout after wake-up” is started.

(4) When a master node does not transmit a header (the LIN Break Field Reception flag or the LIN Sync Field Reception flag is not set) within the time specified by LIN standard, a SW driver should retransmit a wake-up signal.

(5) When three wake-up transmissions are completed, a timer for “timeout after 3 wake-ups” is started.

(6) When a master node does not transmit a header (the LIN Break Field Reception flag or the LIN Sync Field Reception flag is not set) within the time specified (after three wake-up signals) by LIN standard, a SW driver should retransmit a wake-up signal.

21.4.4.10 Detail of Response Space Dominant Detection

Flow of operation on response space dominant detection is illustrated below.

The detection period of the Response Space Dominant Detection Flag is (2) to (4) in the figure below.

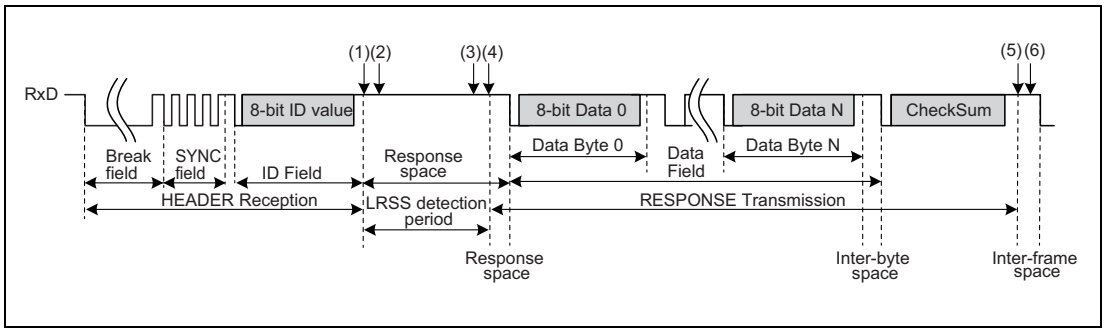


Figure 21.23 Flow of Operation of the LIN Controller on Response Space Dominant Detection

(1) Output LIN Header Reception flag.

(2) Header Reception process completed and SW drivers should read the received ID to check if Response preparation is required.

(3) Check the LIN Response Space Dominant Detection flag and set Response Transmission or Reception start bit or the LIN No Response request bit to “1”.

(4) When Response Transmission or Reception start bit is set and response Transmission is selected, LIN Controller transmits Response Space.

(5) Output Frame Transmission complete flag. The Response Transmission or Reception start bit is cleared by HW.

(6) LIN Response Transmission is completed and SW drivers should read the Response Space Dominant Detection Flag to check status of response transmission.

The figure below illustrates a flow of response transmission in LIN slave mode.

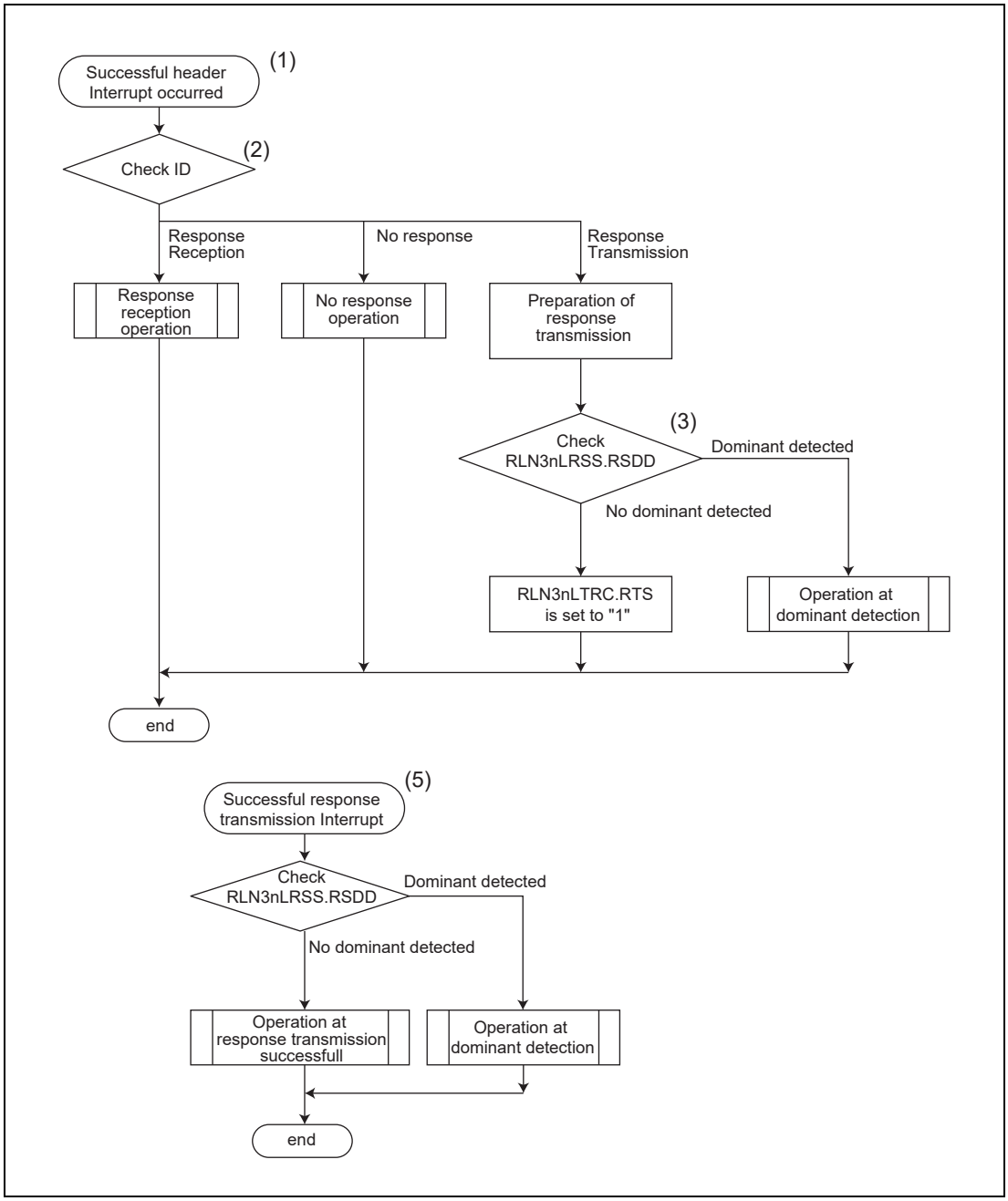


Figure 21.24 Flow of Response Transmission in LIN Slave Mode

21.4.5 UART Mode

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

21.4.5.1 Transmission

Figure 21.25 shows LIN/UART interface (in UART mode) transmission operations; **Table 21.93** shows LIN/UART interface (in UART mode) transmission processing.

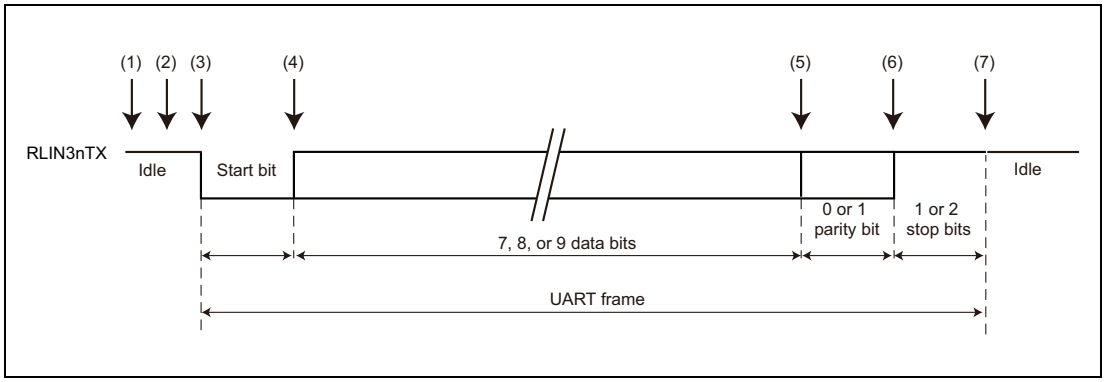


Figure 21.25 LIN/UART Interface (in UART Mode) Transmission Operation

Table 21.93 LIN/UART Interface (UART Mode) Transmission Processing (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Sets an interrupt generation timing. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1.	• Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2)	• Sets the transmit data to the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDR).	• Sets the transmit status flag.
(3)	• Waits an interrupt request. [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • When transmitting data continuously, sets another piece of transmission data in the UART transmit data register (RLN3nLUTDR register), waits for the generation of an interrupt request.	• Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see **Section 21.4.5.1(4), Transmission Start Wait Function**. [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • Outputs a transmission interrupt.
(4)		Transmits the data set in the UART (for wait) transmit data register.
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits.

Table 21.93 LIN/UART Interface (UART Mode) Transmission Processing (2/2)

	Software Processing	LIN/UART Interface Processing
(7)	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another item of transmission data is set, goes to step (3). [When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)] • When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmit status flag. [When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)] • Generates RLIN3n transmission interrupt request. • Clears the transmission status flag.

(1) Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 21.26** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the end of transmission.

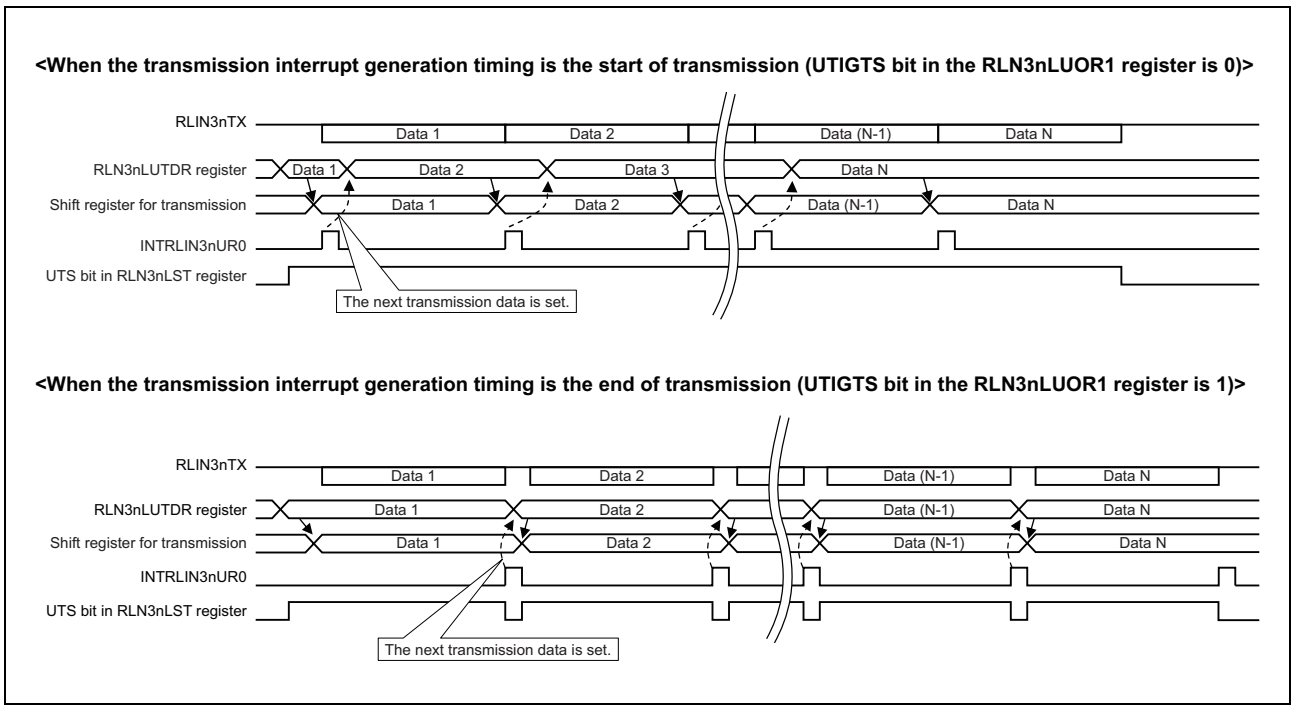


Figure 21.26 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the RLN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

(2) UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.

Figure 21.27 shows the UART buffer transmission operation of the LIN/UART interface (in UART mode). **Table 21.94** describes the UART buffer transmission processing.

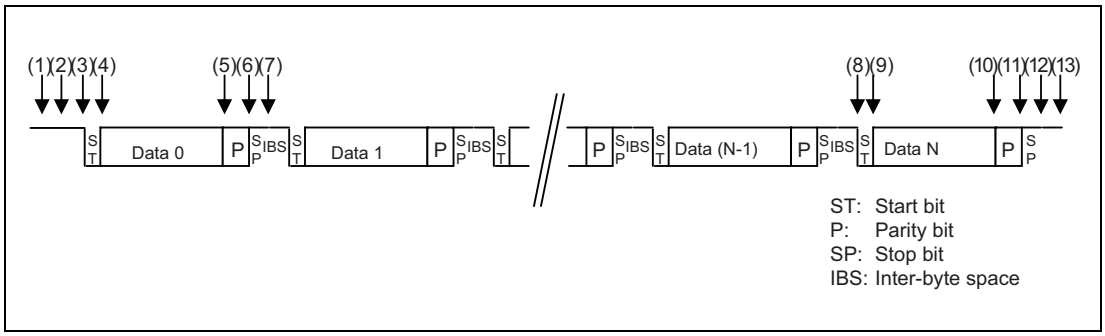


Figure 21.27 UART Buffer Transmission of LIN/UART Interface (in UART Mode)

Table 21.94 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables error detection • Sets data format • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1	• Waits for a transmission trigger (RTS bit) by software
(2)	• Sets the UART buffer data length and whether the system must wait for the start of transmission. • Sets the transmission data in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b =1 to 8) • Sets the UART buffer transmission start bit (RTS).	• Sets the transmit status flag.
(3)	Waits for an interrupt request.	Transmits a start bit. (When switching from reception to transmission during half-duplex communication, transmits the start bit upon completion of the stop bit for reception. For details about this function, see **Section 21.4.5.1(4), Transmission Start Wait Function**.)
(4)		Transmits the data set in the UART data buffer 0 register (RLN3nLUDB0) and the LIN/UART data buffer b register (RLN3nLDBRb).
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).)
(7)		Transmits an inter-byte space (idle).
		Repeats steps (3) to (7) until number of data set in the UART buffer data length select bits -1 is reached.

Table 21.94 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (2/2)

	Software Processing	LIN/UART Interface Processing
(8)		Transmits a start bit.
(9)		Transmits the data set in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)		Transmits a parity bit when parity is used.
(11)		Transmits 1 or 2 stop bits.
(12)		• Sets the successful buffer transmission flag. • Clears the UART buffer transmit start bit (RTS). • A transmission interrupt request signal. • Clears the transmission status flag.
(13)	• Checks the RLN3nLST register, and clears flags • In the case of continuous data transmission, goes to step (2).	

(a) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data areas 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is set. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data areas 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between transmission data items can be set in the IBS bit in the RLN3nLSC register.

Figure 21.28 shows a 9-byte UART buffer and the transmission processing.

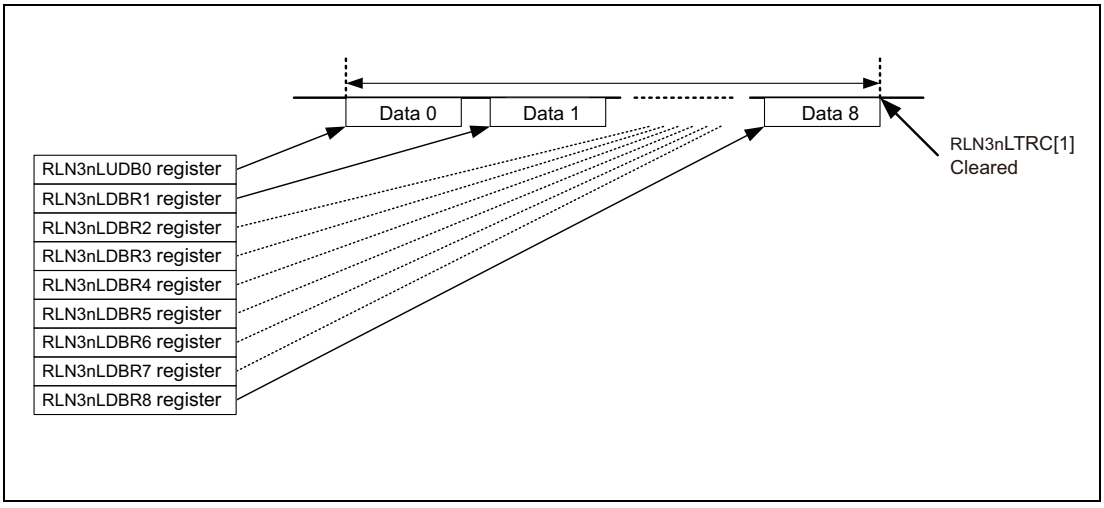


Figure 21.28 UART Buffer and Transmission Processing (for 9-Byte Transmission)

(3) Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 21.4.5.5, Error Status**). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 21.95**.

Table 21.95 Error Detection Timing in UART Mode

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 samplings) is shown in **Figure 21.29**

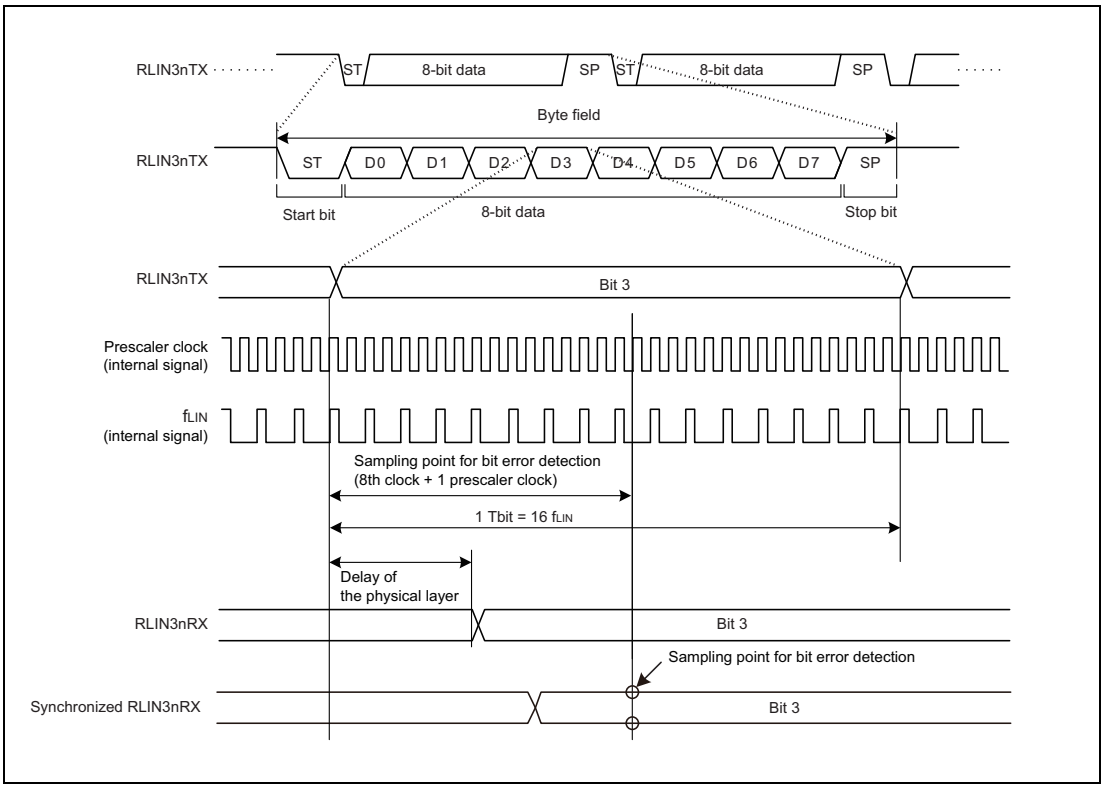


Figure 21.29 Example of Data Transmission Timing (When 1 Tbit = 16 samplings)

(4) Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the RLN3nLUWTDR register, which is used only for the wait function, instead of setting transmission data in the RLN3nLUTDR register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission started) in the RTS bit in the RLN3nLTRC register with 1 set in the UTSW bit in the RLN3nLDFC register.

In such a case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

Note that even if the UART stop bit length selection bit (USBLS) in RLN3nLBFC register is 1 (stop bits = 2 bits), there is only a 1-bit delay.

Figure 21.30 shows the operation of transmission wait function.

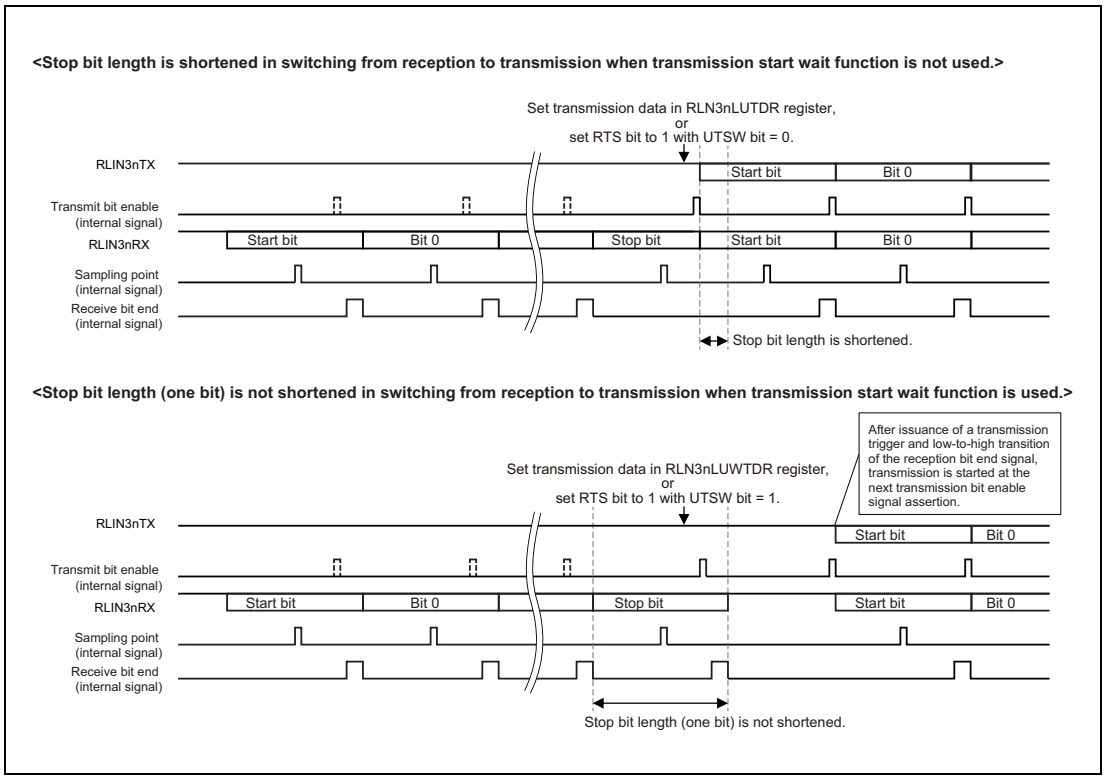


Figure 21.30 Case When Transmit Data Is Set While Stop Bits Are Being Received

21.4.5.2 Reception

Figure 21.31 shows the LIN/UART interface (in UART mode) reception operation. **Table 21.96** shows the LIN/UART interface (in UART mode) reception processing.

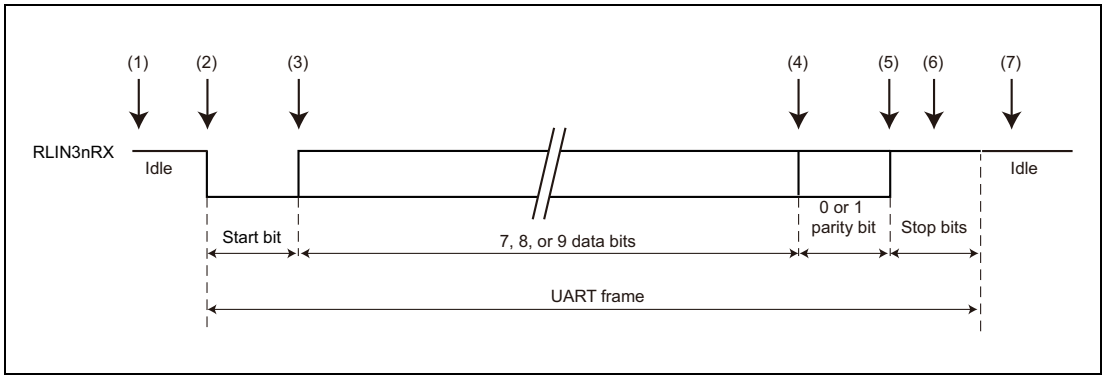


Figure 21.31 LIN/UART Interface (in UART Mode) Reception Operation

Table 21.96 LIN/UART Interface (in UART Mode) Reception Processing

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1.	• Waits for the reception to be enabled by software. • Waits for detection of a start bit.
(2)	Waits for an interrupt request.	• Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag.
(3)		Receives data.
(4)		Receives a parity bit when parity is used.
(5)		Receives only 1 stop bit.
(6)		• Generates a successful RLIN3n reception interrupt request. • Clears the reception status flag.
(7)	Checks the RLN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

(1) Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and {(the number of sampling + 1) / 2} / (the number of sampling) Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filtering function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-samplings majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 21.32 shows an example of data reception timing.

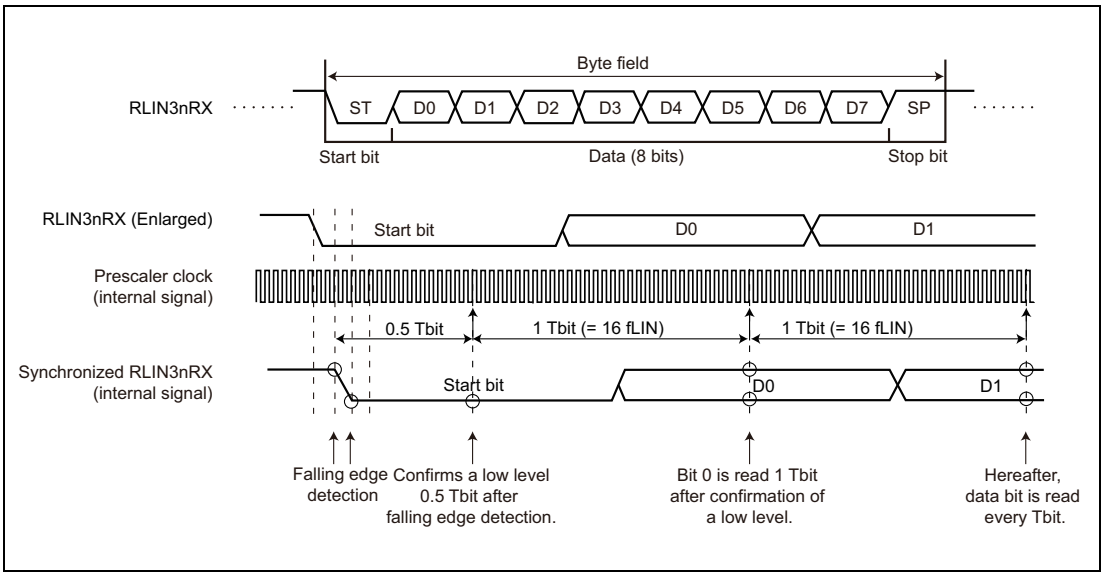


Figure 21.32 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

21.4.5.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

(1) Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR).

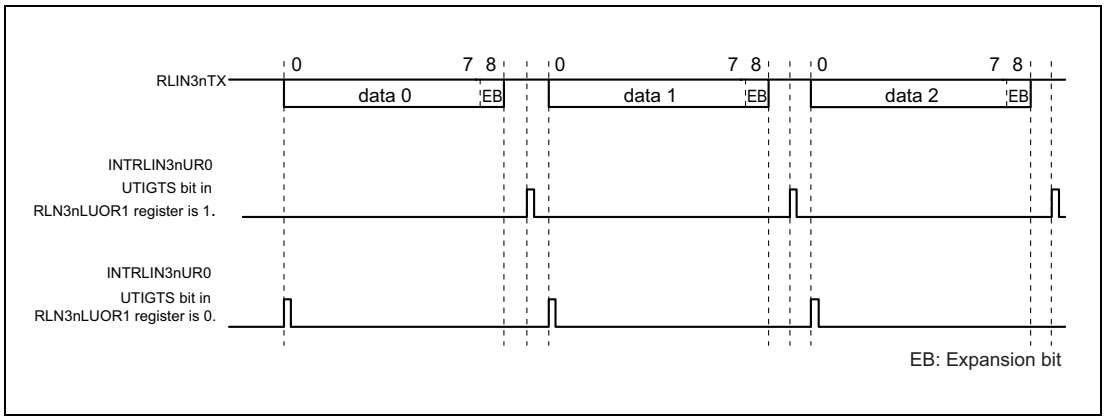


Figure 21.33 Transmission Example When Expansion Bit is Enabled (LSB First)

(2) Expansion Bit Reception

With the LIN/UART interface (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level selection bit (UEBDL) in the UART option register 1 (RLN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

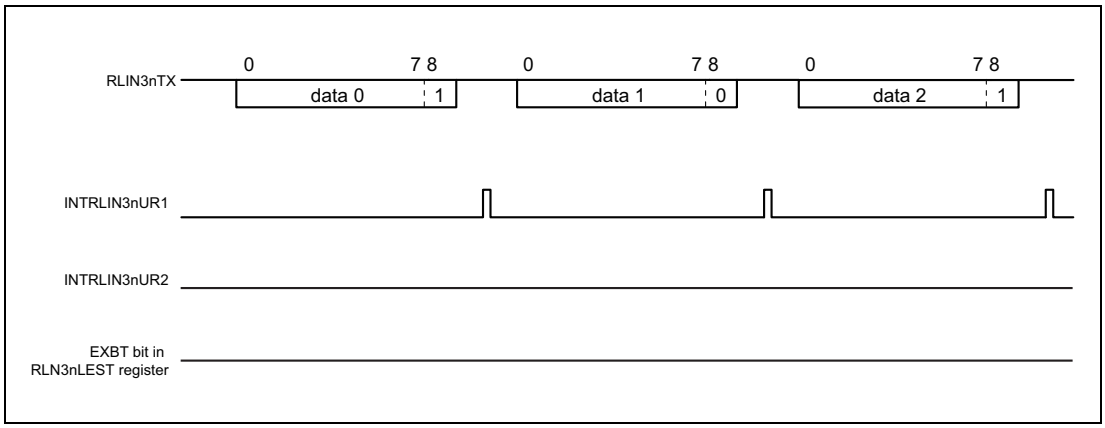


Figure 21.34 Expansion Bit Reception Example (LSB First)

(3) Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level selection bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, successful RLIN3n reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 21.35 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.

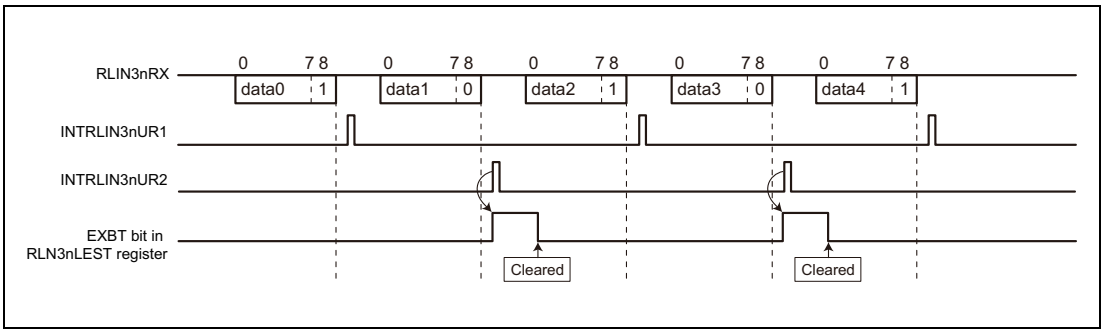


Figure 21.35 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a successful RLIN3n reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

(4) Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, and if the level that was set by the expansion bit detection level selection bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits, excluding the expansion bit in the received data, with the a pre-set RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

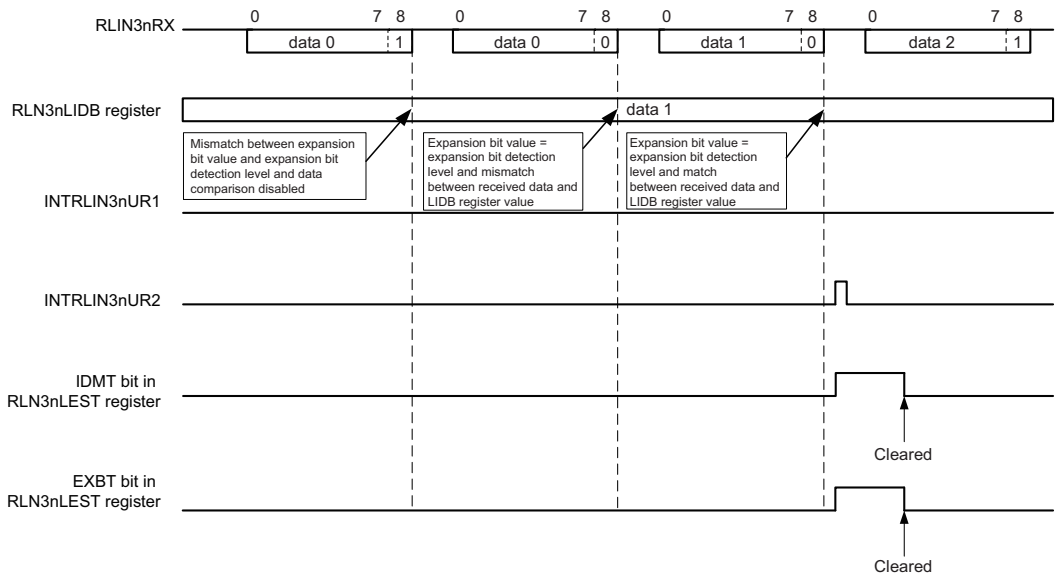
- Generates an RLIN3n status interrupt
- Sets an expansion bit detection flag (EXBT)
- Sets an ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n reception interrupt is not generated.

If the result of the comparison is not a match, no successful RLIN3n reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of another set of data is finished.

Figure 21.36 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.



Note: If a reception error (parity error, framing error, or overrun error) occurs, an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs and the result of comparison is a match, the EXBT and IDMT are also set to 1.

Figure 21.36 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

21.4.5.4 Status

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 21.97 shows the types of statuses available in UART mode.

Table 21.97 Types of Statuses in UART Mode

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	• When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), the transmission of the last data of the data length set by the MDL bit in the RLN3nLDFC register is started. • When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon end of transmission), the transmission of the data length set by the MDL bit in the RLN3nLDFC register is ended.	• When cleared by software • After transition to LIN reset mode	FTC flag in RLN3nLST register	√
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When cleared by software*1 • After transition to LIN reset mode	ERR flag in RLN3nLST register	√
Transmission status	• When data is written to the RLN3nLUTDR or RLN3nLUWTDR register. • When 1 is written to the RTS bit in the RLN3nLTRC register.	• When the transmission of the data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed, but the next transmission data is not specified. • When the transmission of the data in the UART buffer is completed and the RTS bit in the RLN3nLTRC register is cleared. • After transition to LIN reset mode	UTS flag in RLN3nLST register	—
Reception status	• When a start bit is detected.	• When a sampling point for stop bits is detected • After transition to LIN reset mode	URS flag in RLN3nLST register	—

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when the LIN reset mode is being canceled sets the ERR flag in the RLN3nLST register to 0.

21.4.5.5 Error Status

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be verified by using the corresponding bits in the RLN3nLEST register.

Table 21.98 shows available status types.

Table 21.98 Types of Statuses in UART Mode

Status	Error Detection Condition	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the receive pin do not match*1	Continues until the transmission of the set transmission data is finished.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, another data item is received before the data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is finished by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is finished by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is finished.	Disabled *2	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit receive data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

21.4.5.6 Conflicts of Error Status

The table below shows the condition of error status bits that occurs at the end of single frame in UART Mode.

The combination not listed in the table, condition does not occur.

Table 21.99 Conflicts of Error Status (UART Mode)

Number of Conflict	kinds of Errors					
	BER	OER	FER	EXBT	IDMT	UPER
1	1	0	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0
	0	0	0	1	0	0
	0	0	0	0	0	1
2	1	1	0	0	0	0
	1	0	1	0	0	0
	1	0	0	1	0	0
	1	0	0	0	0	1
	0	1	1	0	0	0
	0	1	0	1	0	0
	0	1	0	0	0	1
	0	0	1	0	0	1
	0	0	0	1	1	0
3	1	1	1	0	0	0
	1	1	0	1	0	0
	1	1	0	0	0	1
	1	0	1	0	0	1
	1	0	0	1	1	0
	0	1	1	0	0	1
	0	1	0	1	1	0
4	1	1	0	1	1	0
	1	1	1	0	0	1
5	—	—	—	—	—	—
6	—	—	—	—	—	—

Note: 1: set error
0: no error

21.4.6 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode.

When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and RLIN3nTX and RLIN3nRX are connected to the LIN/UART interface internally. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following four modes:

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception
- LIN slave self-test mode (transmission): Header reception and response transmission
- LIN slave self-test mode (reception): Header reception and response reception

In LIN self-test mode, operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps].
(The NSPB bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)
(The LPRS bits in the RLN3nLWBR register should be set to 000_B.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- LIN slave mode (Auto baud rate)
- Frame/response timeout error

Do not use these functions.

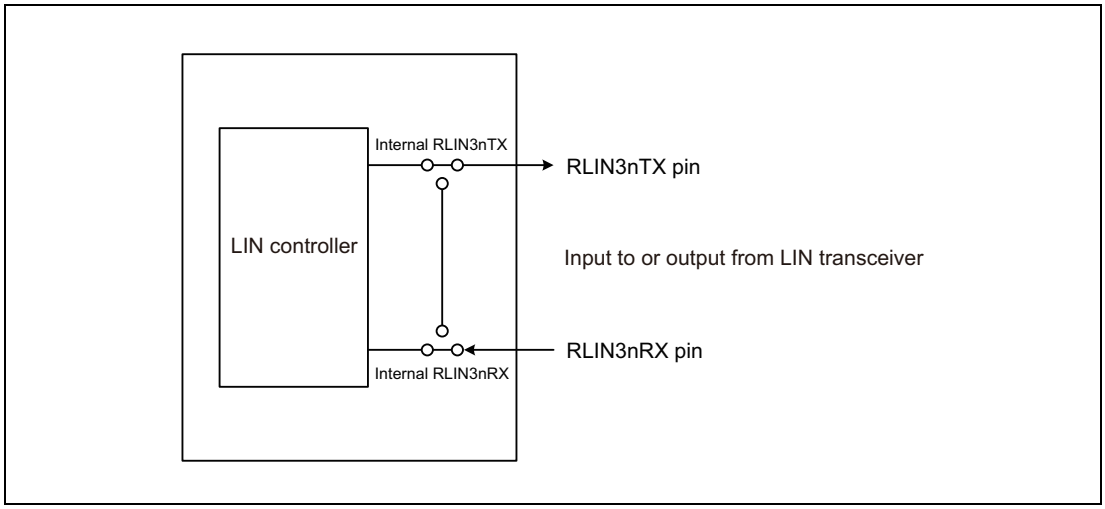


Figure 21.37 Connection in LIN Reset Mode, LIN Mode, and UART Mode

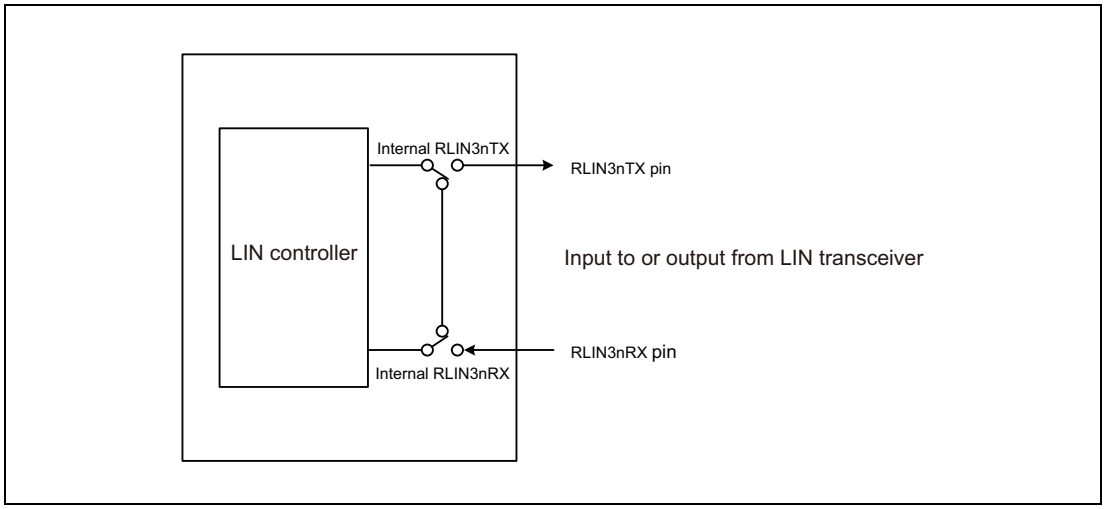


Figure 21.38 Connection in LIN Self-Test Mode

21.4.6.1 Transitioning to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN self-test mode.

To transition to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register as shown below:

- Transition to LIN reset mode
  Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD bits in RLN3nLMD = 00_B (LIN master mode) or 11_B (LIN slave mode [fixed baud rate])
- 1st write: RLN3nLSTC register = 1010 0111_B ($A7_H$)
- 2nd write: RLN3nLSTC register = 0101 1000_B (58_H)
- 3rd write: RLN3nLSTC register = 0000 0001_B (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

21.4.6.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $000x_B$
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 $x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = 00x1 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
- Header transmission → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (transmission) is executed. In this mode, interrupt is generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/ UART interface. To suspend the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled). The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up transmission flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

21.4.6.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $000x_B$
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 $x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$*1
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
  RLN3nLDFC register = 00x0 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
  RLN3nLCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header transmission → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (reception) is executed. In this mode, interrupt is generated, and status and error status are also updated. To suspend the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The RLN3nLBRP0 register, the RLN3nLBRP1 register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).

The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

21.4.6.4 Transmission in LIN Slave Self-Test Mode

To execute a self-test on LIN slave transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = $0000\ 0000_B$
  RLN3nLBRP0 register = $xxxx\ xxxx_B$*1
  RLN3nLBRP1 register = $xxxx\ xxxx_B$*1
  RLN3nLMD register = $00x\ x0011_B$
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = $0000\ xxxx_B$*2
  RLN3nLEDE register = $xx0x\ x00x_B$
- Set the break field and space related registers.
  RLN3nLBFC register = $0000\ 000x_B$*3
  RLN3nLSC register = $00xx\ 0001_B$
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = $00x1\ xxxx_B$
  RLN3nLIDB register = $xxxx\ xxxx_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = $xxxx\ xxxx_B$
- Header reception → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started).
  (The header reception and the response transmission are executed in this order, without manipulating the RTS bit in the RLN3nLTRC register.)
  The LIN slave self-test mode (transmission) is executed. In this mode, interrupt is generated, and status and error status are also updated.
  The checksum is automatically calculated by the LIN/UART interface. To suspend the LIN slave self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The RLN3nLBRP0 register, and the RLN3nLBRP1 register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response transmission interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).
The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up transmission flag is calculated by using the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

21.4.6.5 Reception in LIN Slave Self-Test Mode

To execute a self-test on LIN slave reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = $0000\ 0000_B$
  RLN3nLBRP0 register = $xxxx\ xxxx_B$*1
  RLN3nLBRP1 register = $xxxx\ xxxx_B$*1
  RLN3nLMD register = $00xx\ 0011_B$
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = $0000\ xxxx_B$*2
  RLN3nLEDE register = $xx0x\ x00x_B$
- Set the break field and space related registers.
  RLN3nLBFC register = $0000\ 000x_B$*3
  RLN3nLSC register = $00xx\ 0001_B$*1
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
  RLN3nLDFC register = $00x0\ xxxx_B$
  RLN3nLIDB register = $xxxx\ xxxx_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = $xxxx\ xxxx_B$
  RLN3nLCBR register = $xxxx\ xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header reception → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started).
  (Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response reception are executed in this order.)
  The LIN slave self-test mode (reception) is executed. In this mode, interrupt is generated, and status and error status are also updated. To suspend the LIN slave self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode.
The RLN3nLBRP0 register, the RLN3nLBRP1 register, and the IBS bit in the RLN3nLSC register.
Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Interrupts**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response reception interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).
The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

21.4.6.6 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
  If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B, transition to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; confirm that it is 0 (not in LIN self-test mode).
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

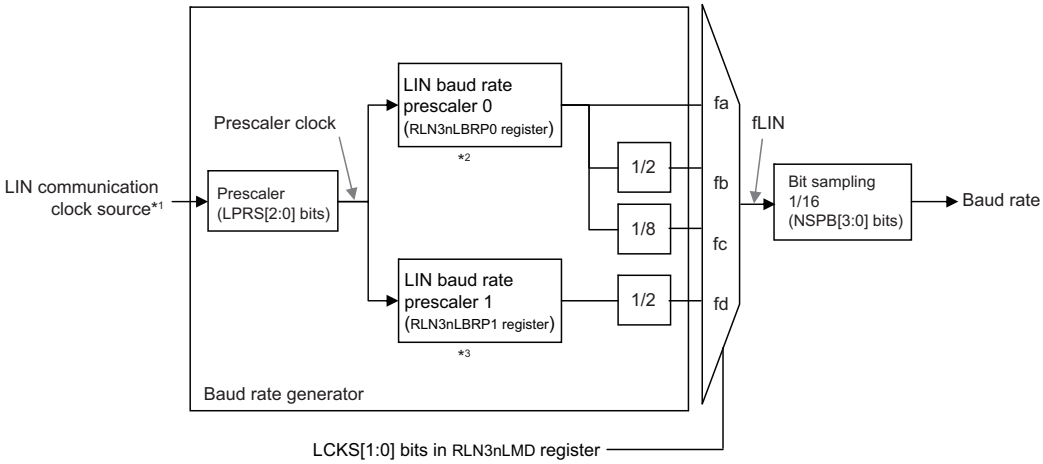
21.4.7 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (fLIN) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (fLIN) by the number of samplings is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generator to be used is switched according to the mode.

21.4.7.1 LIN Master Mode

Figure 21.39 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see **Section 13.6.4, Clock Settings**.
Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.
Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 21.39 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting system clock frequencies are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These system clock frequencies are divided by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by setting the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting system clock frequency is fd = 10417 × 16. This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for calculating the baud rate is described below.

Baud rate of LIN master

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0+ 1) ÷ 16 [bps] (When fa is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

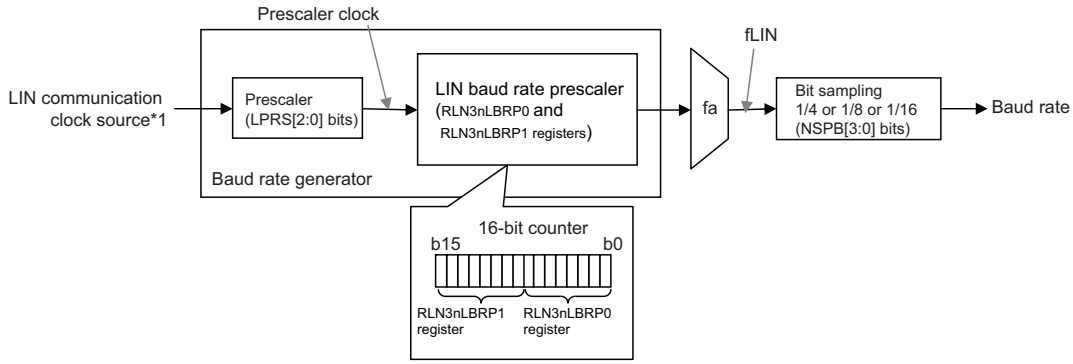
÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for f_{LIN})

21.4.7.2 LIN Slave Mode

Figure 21.40 shows a block diagram of baud rate generation in LIN slave mode.



Note 1. For the LIN communication clock source, see **Section 13.6.4, Clock Settings**.

Figure 21.40 Block Diagram of Baud Rate Generation in LIN Slave Mode

In LIN slave mode (auto baud rate), the baud rate can be set to 1 kbps to 20 kbps. Set the prescaler clock as follows according to the target baud rate:

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz *1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011_B" (four samplings).

The formula for baud rate is described below.
Baud rate of LIN slave

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP01 + 1) ÷ 16 [bps] ([Fixed baud rate])

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

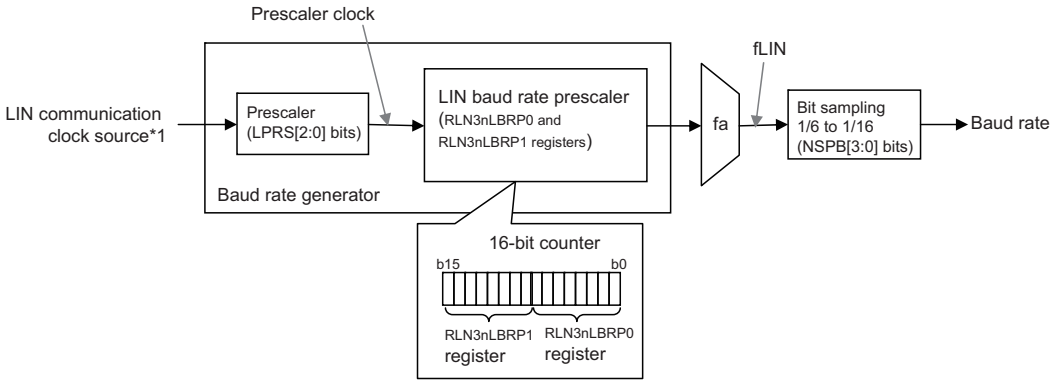
÷ (RLN3nLBRP01 + 1) ÷ 4 or 8 [bps] ([Auto baud rate])

NOTE

For a LIN slave with fixed baud rate, set the NSPB[3:0] bits to "0000_B" (16 samples) or "1111_B" (16 samples). For a LIN slave with auto baud rate, set the NSPB[3:0] bits to "0011_B" (4 samples) or "0100_B" (8 samples).

21.4.7.3 UART Mode

Figure 21.41 shows a block diagram of baud rate generation in UART mode.



Note 1. For the LIN communication clock source, see **Section 13.6.4, Clock Settings**.

Figure 21.41 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

UART baud rate
= {LIN communication clock source frequency} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP01 + 1) ÷ {RLN3nLWBR.NSPB[3:0] selection count} [bps]

21.4.8 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (use the noise filter), the noise filter is enabled. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-samplings majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 21.42 shows the configuration of the noise filter, **Figure 21.43** shows an example of a noise filter circuit, and **Figure 21.44** shows the determination of the received data when the noise filter is used.

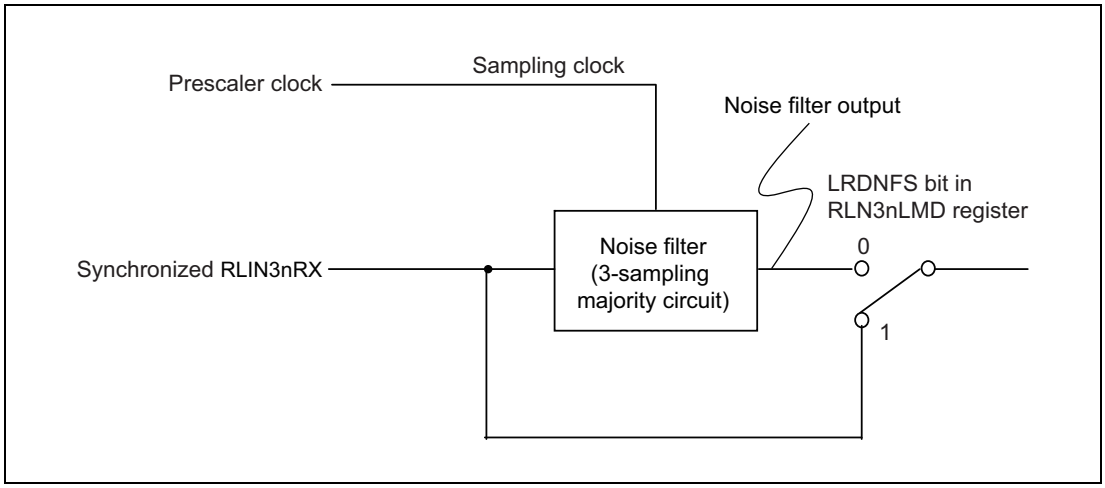


Figure 21.42 Configuration of Noise Filter

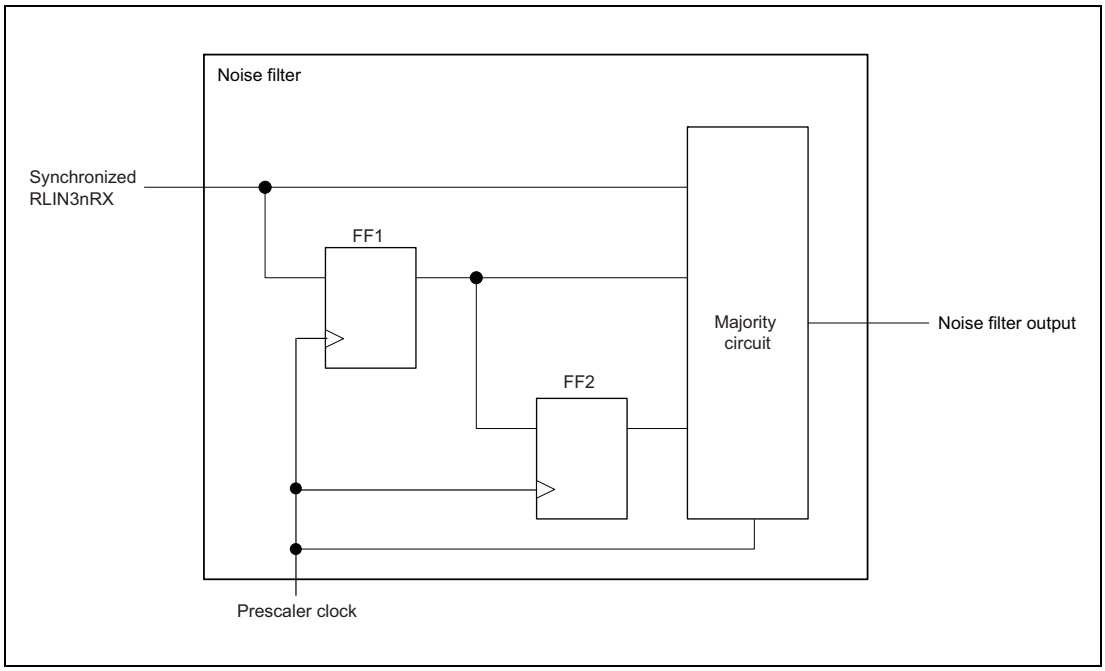


Figure 21.43 Example of Noise Filter Circuit

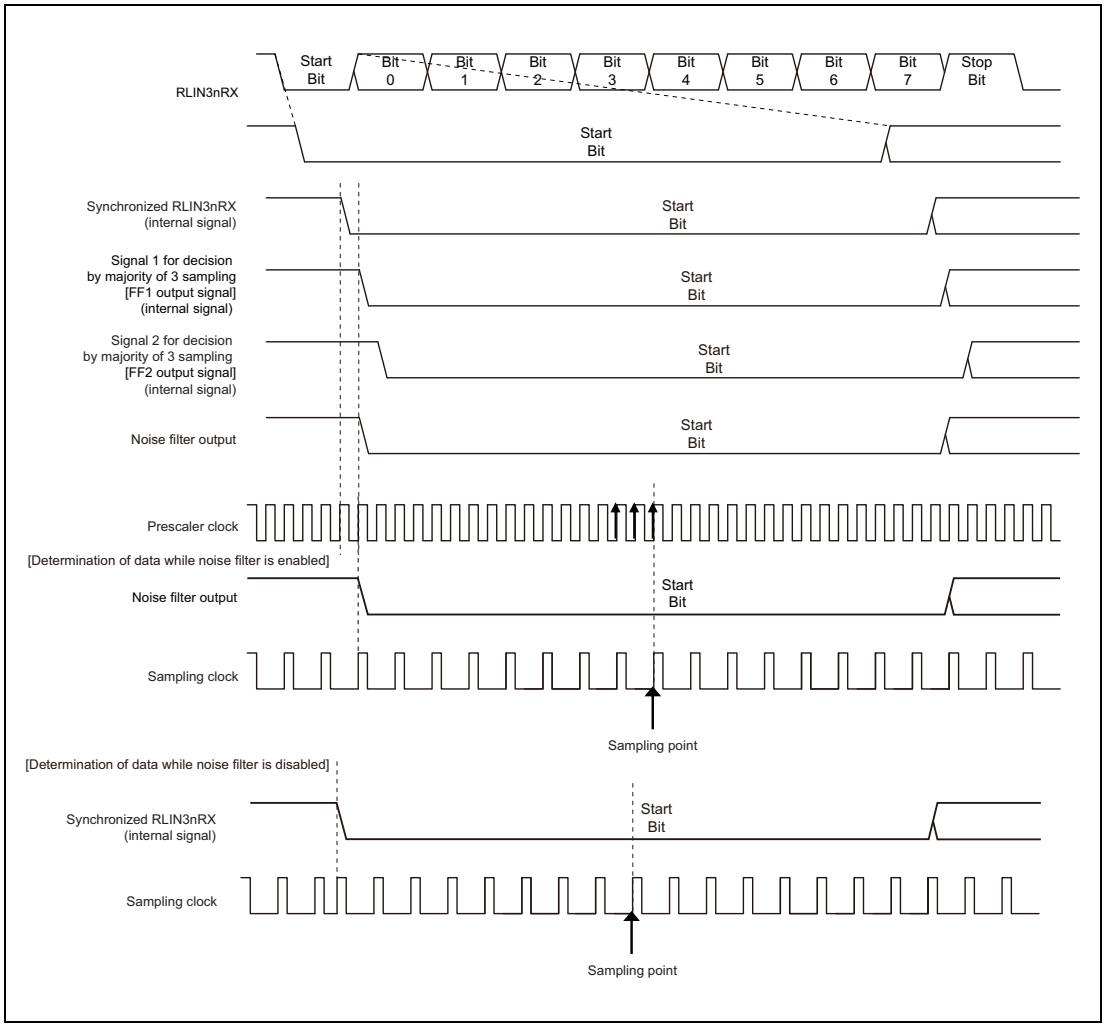


Figure 21.44 Determination of Received Data when Noise Filter is Used

21.5 Notes on Use

21.5.1 Note on LIN Master mode

If in frame separate mode (RLN3nLDFC.FSM=1) the start of the response transmission (Writing 1 to RLN3nLTRC.RTS) coincides with an error*1 occurrence, the response transmission will not be started.

Additionally, after receiving the header of next frame, unexpected data may be sent on the response transmission, even though "1" is not written to RLN3nLTRC.RTS. In case of an error occurrence, re-initialize the LIN/UART module (RLIN3) by transiting to LIN reset mode.

Note 1. Bit error, Physical bus error, Frame/Response timeout error, Framing error, Response preparation error.

21.5.2 Note on LIN Slave mode (Fixed baud rate)

If the low-level width, set by Reception Break (Low-Level) Detection Width Setting bit (RLN3nLBFC.LBLT), is detected during the response transmission (RLN3nLDFC.RCDS=1 and RLN3nLTRC.RTS=1), the response transmission will be interrupted and the RLIN3nTX pin will keep the same status as at the low-level width detection.

If in LIN Slave mode with fixed baud rate, set RLN3nLEDE.BERE=1 (Bit error detection is enable) and RLN3nLSC.IBHS [2:0] = 001_B to 111_B (Except 0 Tbit as response space).

If the Response Space Setting with 0 Tbit (RLN3nLSC.IBHS [2:0] = 000_B), set RLN3nLEDE.BERE=1 and RLN3nLEDE.TERE=1 (Timeout error detection is enable). And transit to LIN reset mode after a timeout error occurrence to initialize the LIN/UART module (RLIN3).

21.5.3 Note on LIN Slave mode (Auto baud rate)

If the conditions (1.) and (2.) are met subsequently, the header might not be received due to false recognition of the ID field. However, the next header will be received correctly.

1. The falling edge is detected from the sampling point of the Stop-bit and Response/Inter-Byte-Space which Bit error is occurred to the end of its bit.
2. The falling edge of the start bit of the ID field is detected from the sampling point of the sync field of the next reception to the end of its bit.

Section 22 I2C Bus Interface (RIIC)

This section contains a generic description of the I2C Bus Interface (RIIC).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RIIC.

22.1 Features RIIC for RH850/U2A-EVA

22.1.1 Number of Units and Channels

This microcontroller has the following number of RIIC units.

Each RIIC unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 22.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	2 (n = 0 to 1)	1 (n = 0)
Name	RIICn									

Table 22.2 Index

Index	Description
n	Throughout this section, the individual RIIC units are identified by the index “n” (n = 0 to 1): for example, RIICnCR1 is the I2C bus control register 1.

22.1.2 Register Base Addresses

RIICn base addresses are listed in the following table.

RIICn register addresses are given as offsets from the base address.

Table 22.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RIIC0_base>	FFF2 2000_H	P-Bus Group 7
<RIIC1_base>	FFF2 3000_H	P-Bus Group 7

22.1.3 Clock Supply

The RIICn clock supply is shown in the following table.

Table 22.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
RIICn	PCLK*1	CLK_LSB	Bus clock

Note 1. Set PCLK to a value that is less than 1/2 the SCL clock (high level width).
For information about SCL clock, refer to **Table 22.7, External Input/Output Signals**.
For details of clock supply, see **Section 13, Clock Controller**.

22.1.4 Interrupt Requests and Error Notifications

RIICn interrupt requests are listed in the following table.

Table 22.5 Interrupt and DMA/DTS Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
RIIC0				
INTRIIC0EE	RIIC communication error/event generation interrupt	685	—	—
INTRIIC0RI	RIIC receive end interrupt	686	Group0-150	Group0-122
INTRIIC0TI	RIIC transmit data empty interrupt	687	Group0-151	Group0-123
INTRIIC0TEI	RIIC transmit end interrupt	688	—	—
RIIC1				
INTRIIC1EE	RIIC communication error/event generation interrupt	689	—	—
INTRIIC1RI	RIIC receive end interrupt	690	Group0-152	Group0-124
INTRIIC1TI	RIIC transmit data empty interrupt	691	Group0-153	Group0-125
INTRIIC1TEI	RIIC transmit end interrupt	692	—	—

This module has no error notifications.

22.1.5 Reset Sources

RIICn reset sources are listed in the following table. RIICn is initialized by these reset sources.

Table 22.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RIICn	All registers	√	√	√	√	√	√	—

For details of reset source, see **Section 9, Reset Controller**.

22.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

Table 22.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
RIIC0		
RIIC0SCL	Serial clock I/O pin	RIIC0SCL
RIIC0SDA	Serial data I/O pin	RIIC0SDA
RIIC1		
RIIC1SCL	Serial clock I/O pin	RIIC1SCL
RIIC1SDA	Serial data I/O pin	RIIC1SDA

When using these ports, the PBDCn register for the corresponding port and the corresponding bit in the PODCn register must be set to 1.

For details about PBDCn/PODCn registers, refer to **Section 2.5, Port Register Description**.

22.2 Overview

22.2.1 Functional Overview

Communications format

- I2C bus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times according to the specified transfer rate.

Transfer rate

Up to 400 kbps

SCL clock

- For master operation, the duty cycle of the SCL clock is selectable in the following range:
  - 0% < Duty < 100%

Issuing and detecting conditions

Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.

Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses and device ID addresses are detectable.

Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create a conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching of internal and line levels for transmit data.
- Loss of arbitration due to detection of the start condition while the bus is busy can be detected (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching can be detected.
- Loss of arbitration due to non-matching of internal and line levels for data can be detected in slave transmission.

Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

Noise removal

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.

Interrupt sources

- Four sources:
  - Error in transfer or occurrence of events (detection of arbitration loss, NACK, time-out, a start condition including a restart condition, or a stop condition)
  - Reception complete (including matching with a slave address)
  - Transmit-data-empty (including matching with a slave address)
  - Transmission complete

22.2.2 Block Diagram

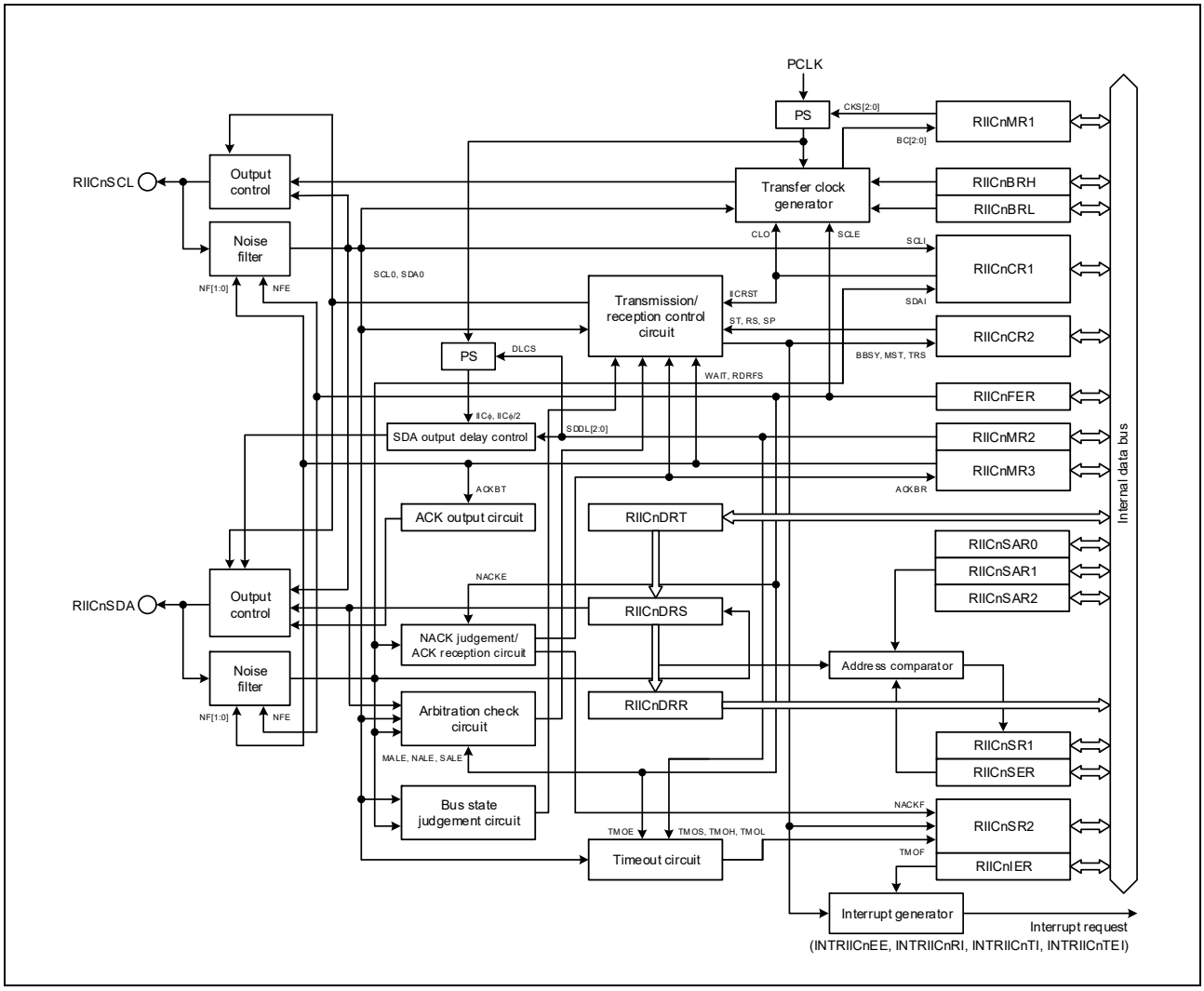


Figure 22.1 Block Diagram of RIIC

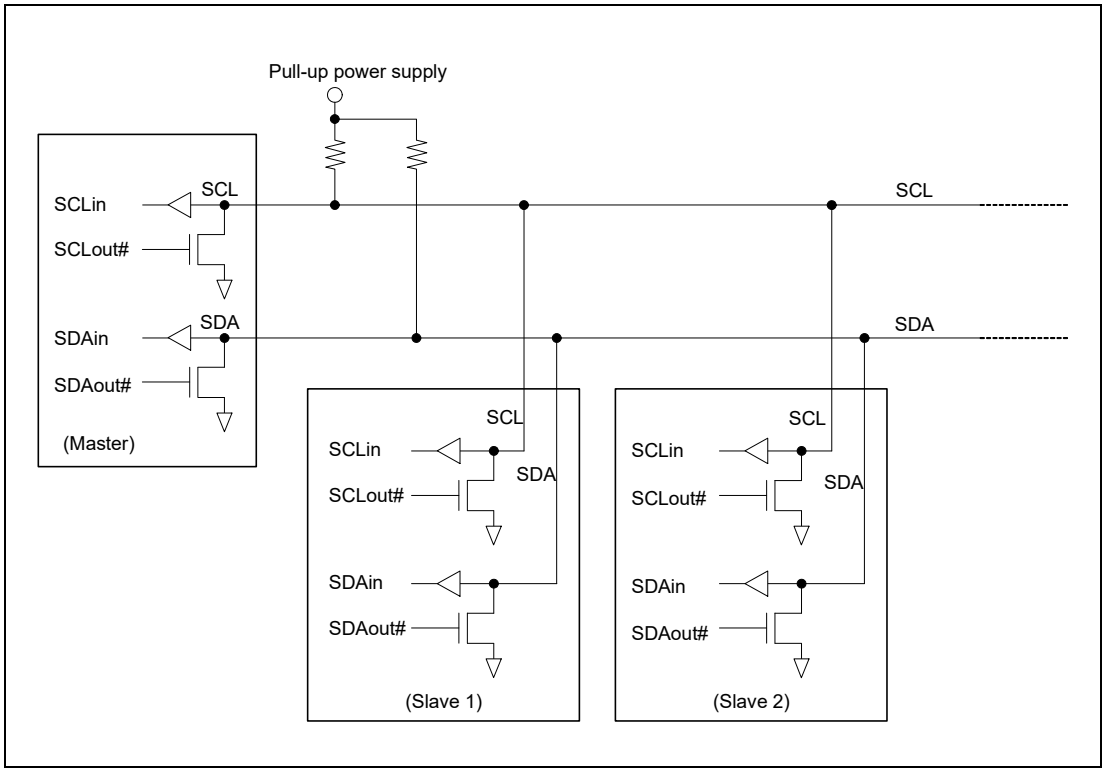


Figure 22.2 Example of Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

22.3 Registers

22.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn_base>, see **Section 22.1.2, Register Base Addresses**.

Table 22.8 List of Registers

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
RIICn	I2C Bus Control Register 1	RIICnCR1	<RIICn_base> + 0000_H	8, 16, 32	*1	—
RIICn	I2C Bus Control Register 2	RIICnCR2	<RIICn_base> + 0004_H	8, 16, 32	*1	—
RIICn	I2C Bus Mode Register 1	RIICnMR1	<RIICn_base> + 0008_H	8, 16, 32	*1	—
RIICn	I2C Bus Mode Register 2	RIICnMR2	<RIICn_base> + $000C_H$	8, 16, 32	*1	—
RIICn	I2C Bus Mode Register 3	RIICnMR3	<RIICn_base> + 0010_H	8, 16, 32	*1	—
RIICn	I2C Bus Function Enable Register	RIICnFER	<RIICn_base> + 0014_H	8, 16, 32	*1	—
RIICn	I2C Bus Status Enable Register	RIICnSER	<RIICn_base> + 0018_H	8, 16, 32	*1	—
RIICn	I2C Bus Interrupt Enable Register	RIICnIER	<RIICn_base> + $001C_H$	8, 16, 32	*1	—
RIICn	I2C Bus Status Register 1	RIICnSR1	<RIICn_base> + 0020_H	8, 16, 32	*1	—
RIICn	I2C Bus Status Register 2	RIICnSR2	<RIICn_base> + 0024_H	8, 16, 32	*1	—
RIICn	I2C Slave Address Register 0	RIICnSAR0	<RIICn_base> + 0028_H	8, 16, 32	*1	—
RIICn	I2C Slave Address Register 1	RIICnSAR1	<RIICn_base> + $002C_H$	8, 16, 32	*1	—
RIICn	I2C Slave Address Register 2	RIICnSAR2	<RIICn_base> + 0030_H	8, 16, 32	*1	—
RIICn	I2C Bus Bit Rate Low-Level Register	RIICnBRL	<RIICn_base> + 0034_H	8, 16, 32	*1	—
RIICn	I2C Bus Bit Rate High-Level Register	RIICnBRH	<RIICn_base> + 0038_H	8, 16, 32	*1	—
RIICn	I2C Bus Transmit Data Register	RIICnDRT	<RIICn_base> + $003C_H$	8, 16, 32	*1	—
RIICn	I2C Bus Receive Data Register	RIICnDRR	<RIICn_base> + 0040_H	8, 16, 32	*1	—
RIICn	I2C Bus Shift Register	RIICnDRS	—	—	—	—

Note 1. n = 0: PBG70#3
n = 1: PBG70#4

22.3.2 RIICnCR1 — I2C Bus Control Register 1

Access: RIICnCR1 register can be read or written in 32-bit units.
RIICnCR1L and RIICnCR1H registers can be read or written in 16-bit units.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH registers can be read or written in 8-bit units.

Address: RIICnCR1: <RIICn_base> + 0000_H

RIICnCR1L: <RIICn_base> + 0000_H, RIICnCR1H: <RIICn_base> + 0002_H

RIICnCR1LL: <RIICn_base> + 0000_H, RIICnCR1LH: <RIICn_base> + 0001_H,
RIICnCR1HL: <RIICn_base> + 0002_H, RIICnCR1HH: <RIICn_base> + 0003_H

Value after reset: 0000 $001F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	W	R/W	R/W	R	R

Table 22.9 RIICnCR1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ICE	I2C Bus Interface Enable 0: Disabled (the RIICnSCL and RIICnSDA pins are not driven). 1: Enabled (the RIICnSCL and RIICnSDA pins are driven). (This bit selects an RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I2C Bus Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (This bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (When read, 1 is returned.)
3	SCLO	SCL Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSCL pin low. 1: The RIICnSCL pin has been released. • Write: 0: The RIIC drives the RIICnSCL pin low. 1: Releases the RIICnSCL pin.

Table 22.9 RIICnCR1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	SDAO	SDA Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSDA pin low. 1: The RIICnSDA pin has been released. • Write: 0: The RIIC drives the RIICnSDA pin low. 1: Releases the RIICnSDA pin.
1	SCLI	SCL Line Monitor 0: RIICnSCL line is low. 1: RIICnSCL line is high.
0	SDAI	SDA Line Monitor 0: RIICnSDA line is low. 1: RIICnSDA line is high.

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the RIICnSDA and RIICnSCL signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting in the periods mentioned above is not guaranteed.

When reading these bits, the state of signals output from the RIIC is returned.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see **Section 22.4.10.2, Extra SCL Clock Cycle Output Function**.

IICRST Bit (I2C Bus Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 22.10** lists the types of RIIC reset.

The RIIC reset resets all registers (except ICE and IICRST) including the RIICnCR2.BBSY flag and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I2C bus shift register (RIICnDRS), and the I2C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 22.4.11, Reset Function of RIIC**.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error, etc.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but instead initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 22.10 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

ICE Bit (I2C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see **Table 22.10, RIIC Resets**.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 stops driving of the RIICnSCL and RIICnSDA pins.

CAUTION

Though the output from RIICnSDA or RIICnSCL is disabled while the ICE bit is 0, the input to RIICnSDA or RIICnSCL is enabled. The RIICnSCL and RIICnSDA pin functions should not be assigned to the RIIC. If assigned, it causes the slave addresses to be compared.

22.3.3 RIICnCR2 — I2C Bus Control Register 2

Access: RIICnCR2 register can be read or written in 32-bit units.
RIICnCR2L and RIICnCR2H registers can be read or written in 16-bit units.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH registers can be read or written in 8-bit units.

Address: RIICnCR2: <RIICn_base> + 0004_H

RIICnCR2L: <RIICn_base> + 0004_H, RIICnCR2H: <RIICn_base> + 0006_H

RIICnCR2LL: <RIICn_base> + 0004_H, RIICnCR2LH: <RIICn_base> + 0005_H,
RIICnCR2HL: <RIICn_base> + 0006_H, RIICnCR2HH: <RIICn_base> + 0007_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R

Table 22.11 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I2C bus is released (the bus is free). 1: The I2C bus is occupied (the bus is busy).
6	MST*1	Master/Slave Mode 0: Slave mode 1: Master mode
5	TRS*1	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. When the RIICnMR1.MTWP bit is set to 1, the MST and TRS bits can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see **Section 22.4.9, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).
Note that arbitration may be lost as the start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see **Section 22.4.9, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the RS bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see **Section 22.4.9, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode (1 or 0) by the following conditions.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)
- When 0 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode (1 or 0) by the following conditions.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

CAUTIONS

- When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
- When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

22.3.4 RIICnMR1 — I2C Bus Mode Register 1

Access: RIICnMR1 register can be read or written in 32-bit units.
RIICnMR1L and RIICnMR1H registers can be read or written in 16-bit units.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH registers can be read or written in 8-bit units.

Address: RIICnMR1: <RIICn_base> + 0008_H

RIICnMR1L: <RIICn_base> + 0008_H, RIICnMR1H: <RIICn_base> + $000A_H$

RIICnMR1LL: <RIICn_base> + 0008_H, RIICnMR1LH: <RIICn_base> + 0009_H,
RIICnMR1HL: <RIICn_base> + $000A_H$, RIICnMR1HH: <RIICn_base> + $000B_H$

Value after reset: 0000 0008_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 22.12 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MTWP	MST/TRS Write Protect 0: Disables writing to the RIICnCR2.MST and TRS bits. 1: Enables writing to the RIICnCR2.MST and TRS bits.
6 to 4	CKS[2:0]	Internal Reference Clock Selection (IICϕ) b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock
3	BCWP*1	BC Write Protect 0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.) 1: Protects the BC[2:0] bits.
2 to 0	BC[2:0]	Bit Counter b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames. When setting any value other than 000_B, set the value while the SCL line is at a low level.

[Clearing conditions]

- When 1 is written to the RIICnCR1.IICRST bit to apply a RIIC reset or an internal reset is initiated.
- Data transfer including the acknowledge bit being completed.
- A start condition including a restart condition being detected.

22.3.5 RIICnMR2 — I2C Bus Mode Register 2

Access: RIICnMR2 register can be read or written in 32-bit units.
RIICnMR2L and RIICnMR2H registers can be read or written in 16-bit units.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH registers can be read or written in 8-bit units.

Address: RIICnMR2: <RIICn_base> + $000C_H$

RIICnMR2L: <RIICn_base> + $000C_H$, RIICnMR2H: <RIICn_base> + $000E_H$

RIICnMR2LL: <RIICn_base> + $000C_H$, RIICnMR2LH: <RIICn_base> + $000D_H$,
RIICnMR2HL: <RIICn_base> + $000E_H$, RIICnMR2HH: <RIICn_base> + $000F_H$

Value after reset: $0000\ 0006_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 22.13 RIICnMR2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1
6 to 4	SDDL[2:0]	SDA Output Delay Counter • When RIICnMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When RIICnMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOH	Timeout H Count Control 0: Disables counting while the SCL line is at a high level. 1: Enables counting while the SCL line is at a high level.

Table 22.13 RIICnMR2 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TMOL	Timeout L Count Control 0: Disables counting while the SCL line is at a low level. 1: Enables counting while the SCL line is at a low level.
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see **Section 22.4.10.1, Timeout Function**.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see **Section 22.4.4, Facility for Delaying SDA Output**.

CAUTION

Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [Fm])

22.3.6 RIICnMR3 — I2C Bus Mode Register 3

Access: RIICnMR3 register can be read or written in 32-bit units.
RIICnMR3L and RIICnMR3H registers can be read or written in 16-bit units.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH registers can be read or written in 8-bit units.

Address: RIICnMR3: <RIICn_base> + 0010_H

RIICnMR3L: <RIICn_base> + 0010_H, RIICnMR3H: <RIICn_base> + 0012_H

RIICnMR3LL: <RIICn_base> + 0010_H, RIICnMR3LH: <RIICn_base> + 0011_H,
RIICnMR3HL: <RIICn_base> + 0012_H, RIICnMR3HH: <RIICn_base> + 0013_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 22.14 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	WAIT*2	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS*2	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP*1	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT*1	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Digital noise Filter Stage Selection b1 b0 0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Digital Noise Filter Stage Selection)

These bits are used to select the number of stages of the digital noise filter.

CAUTION

Set the noise range to be filtered out by the noise filter within a range less than the SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock synchronized (IICϕ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit after ACKBT reading while the ACKWP bit is set to 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

CAUTION

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

CAUTION

When the value of the WAIT bit is cleared to 0, be sure to read the RIICnDRR beforehand.

22.3.7 RIICnFER — I2C Bus Function Enable Register

Access: RIICnFER register can be read or written in 32-bit units.
RIICnFERL and RIICnFERH registers can be read or written in 16-bit units.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH registers can be read or written in 8-bit units.

Address: RIICnFER: <RIICn_base> + 0014_H

RIICnFERL: <RIICn_base> + 0014_H, RIICnFERH: <RIICn_base> + 0016_H

RIICnFERLL: <RIICn_base> + 0014_H, RIICnFERLH: <RIICn_base> + 0015_H,
RIICnFERHL: <RIICn_base> + 0016_H, RIICnFERHH: <RIICn_base> + 0017_H

Value after reset: $0000\ 0072_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.15 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) If 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode, the TRS bit is not cleared.
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see **Section 22.4.10.1, Timeout Function**.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to select enabling or disabling of the arbitration-lost detection function. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether the detection of ACK during transmission of NACK in reception (such as when slaves with the same address are present on the bus and each is transmitting different data, or when two or more masters select the same slave device simultaneously with different numbers of bytes for reception) is judged to represent a loss in arbitration.

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with a rising or falling edge on the SCL line. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

22.3.8 RIICnSER — I2C Bus Status Enable Register

Access: RIICnSER register can be read or written in 32-bit units.
RIICnSERL and RIICnSERH registers can be read or written in 16-bit units.
RIICnSERLL, RIICnSERLH, RIICnSERHL, and RIICnSERHH registers can be read or written in 8-bit units.

Address: RIICnSER: <RIICn_base> + 0018_H

RIICnSERL: <RIICn_base> + 0018_H, RIICnSERH: <RIICn_base> + $001A_H$

RIICnSERLL: <RIICn_base> + 0018_H, RIICnSERLH: <RIICn_base> + 0019_H,
RIICnSERHL: <RIICn_base> + $001A_H$, RIICnSERHH: <RIICn_base> + $001B_H$

Value after reset: $0000\ 0009_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 22.16 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

GCE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address ($0000\ 000_B$ + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID ($1111\ 100_B$) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 22.4.6.3, Device-ID Address Detection**.

22.3.9 RIICnIER — I2C Bus Interrupt Enable Register

Access: RIICnIER register can be read or written in 32-bit units.
RIICnIERL and RIICnIERH registers can be read or written in 16-bit units.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH registers can be read or written in 8-bit units.

Address: RIICnIER: <RIICn_base> + $001C_H$

RIICnIERL: <RIICn_base> + $001C_H$, RIICnIERH: <RIICn_base> + $001E_H$

RIICnIERLL: <RIICn_base> + $001C_H$, RIICnIERLH: <RIICn_base> + $001D_H$,
RIICnIERHL: <RIICn_base> + $001E_H$, RIICnIERHH: <RIICn_base> + $001F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.17 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt request (INTRIICnTI) is disabled. 1: Transmit data empty interrupt request (INTRIICnTI) is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt request (INTRIICnTEI) is disabled. 1: Transmit end interrupt request (INTRIICnTEI) is enabled.
5	RIE	Receive Complete Interrupt Enable 0: Receive complete interrupt request (INTRIICnRI) is disabled. 1: Receive complete interrupt request (INTRIICnRI) is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the RIICnSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the RIICnSR2.AL flag is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the RIICnSR2.START flag is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the RIICnSR2.STOP flag is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the RIICnSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Complete Interrupt Enable)

This bit is used to enable or disable receive complete interrupt requests (INTRIICnRI) when the RIICnSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTRIICnTEI) when the RIICnSR2.TEND flag is set to 1. An INTRIICnTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTRIICnTI) when the RIICnSR2.TDRE flag is set to 1.

22.3.10 RIICnSR1 — I2C Bus Status Register 1

Access: RIICnSR1 register can be read or written in 32-bit units.
RIICnSR1L and RIICnSR1H registers can be read or written in 16-bit units.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH registers can be read or written in 8-bit units.

Address: RIICnSR1: <RIICn_base> + 0020_H

RIICnSR1L: <RIICn_base> + 0020_H, RIICnSR1H: <RIICn_base> + 0022_H

RIICnSR1LL: <RIICn_base> + 0020_H, RIICnSR1LH: <RIICn_base> + 0021_H,
RIICnSR1HL: <RIICn_base> + 0022_H, RIICnSR1HH: <RIICn_base> + 0023_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(/W) *1	R	R(/W) *1	R(/W) *1	R(/W) *1	R(/W) *1

Note 1. Only 0 can be written to this bit.

Table 22.18 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0_B + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address ($1111\ 100_B$) plus 1[R] has matched while the setting of the RIICnSER.DIDE bit is 1 (device ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID ($1111\ 100_B$)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

22.3.11 RIICnSR2 — I2C Bus Status Register 2

Access: RIICnSR2 register can be read or written in 32-bit units.
RIICnSR2L and RIICnSR2H registers can be read or written in 16-bit units.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH registers can be read or written in 8-bit units.

Address: RIICnSR2: <RIICn_base> + 0024_H

RIICnSR2L: <RIICn_base> + 0024_H, RIICnSR2H: <RIICn_base> + 0026_H

RIICnSR2LL: <RIICn_base> + 0024_H, RIICnSR2LH: <RIICn_base> + 0025_H,
RIICnSR2HL: <RIICn_base> + 0026_H, RIICnSR2HH: <RIICn_base> + 0027_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 22.19 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.

TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (the RIICnCR2.BBSY flag is 1) in master mode (the RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (the RIICnSR1 register is not 00_H) and the bus is busy (the RIICnCR2.BBSY bit is 1) in slave mode (the RIICnCR2.MST bit is 0).
- The bus is free (the RIICnCR2.BBSY flag is 0) while generation of a start condition is requested (the RIICnCR2.ST bit is 1).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 22.20 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

RIICnFER			RIICnSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	—	—	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
—	1	—	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
—	—	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

—: Don't care

START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Complete)

[Setting conditions]

- When receive data has been transferred from RIICnDRS to RIICnDRR
  At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches the address enabled in RIICnSER after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0
- In master mode, transition to master reception while the R/W# bit appended to the slave address is set to 1

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition is detected
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When the received slave address matches the address enabled in RIICnSER after a start condition including a restart condition is detected with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is not transferred to the RIICnDRS register and the data of RIICnDRT register is retained, and the TDRE flag is not set to 1.

22.3.12 RIICnSARy — I2C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy register can be read or written in 32-bit units.
RIICnSARyL and RIICnSARyH registers can be read or written in 16-bit units.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH registers can be read or written in 8-bit units.

Address: RIICnSAR0: <RIICn_base> + 0028_H

RIICnSAR0L: <RIICn_base> + 0028_H, RIICnSAR0H: <RIICn_base> + $002A_H$

RIICnSAR0LL: <RIICn_base> + 0028_H, RIICnSAR0LH: <RIICn_base> + 0029_H,
RIICnSAR0HL: <RIICn_base> + $002A_H$, RIICnSAR0HH: <RIICn_base> + $002B_H$

RIICnSAR1: <RIICn_base> + $002C_H$

RIICnSAR1L: <RIICn_base> + $002C_H$, RIICnSAR1H: <RIICn_base> + $002E_H$

RIICnSAR1LL: <RIICn_base> + $002C_H$, RIICnSAR1LH: <RIICn_base> + $002D_H$,
RIICnSAR1HL: <RIICn_base> + $002E_H$, RIICnSAR1HH: <RIICn_base> + $002F_H$

RIICnSAR2: <RIICn_base> + 0030_H

RIICnSAR2L: <RIICn_base> + 0030_H, RIICnSAR2H: <RIICn_base> + 0032_H

RIICnSAR2LL: <RIICn_base> + 0030_H, RIICnSAR2LH: <RIICn_base> + 0031_H,
RIICnSAR2HL: <RIICn_base> + 0032_H, RIICnSAR2HH: <RIICn_base> + 0033_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—	SVA[9:1]									SVA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.21 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. • When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are valid and form a 7-bit slave address. • When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. • When the FSy bit is 0 (7-bit address format), this bit is invalid. • When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

22.3.13 RIICnBRL — I2C Bus Bit Rate Low-Level Register

Access: RIICnBRL register can be read or written in 32-bit units.
RIICnBRLL and RIICnBRLH registers can be read or written in 16-bit units.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH registers can be read or written in 8-bit units.

Address: RIICnBRL: <RIICn_base> + 0034_H

RIICnBRLL: <RIICn_base> + 0034_H, RIICnBRLH: <RIICn_base> + 0036_H

RIICnBRLLL: <RIICn_base> + 0034_H, RIICnBRLLH: <RIICn_base> + 0035_H,
RIICnBRLHL: <RIICn_base> + 0036_H, RIICnBRLHH: <RIICn_base> + 0037_H

Value after reset: 0000 $00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22.22 RIICnBRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see **Section 22.4.7, Automatic Low-Hold Function for SCL**); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.

RIICnBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: standard mode [Sm])
100 ns (up to 400 kbps: fast mode [Fm])

22.3.14 RIICnBRH — I2C Bus Bit Rate High-Level Register

Access: RIICnBRH register can be read or written in 32-bit units.
RIICnBRHL and RIICnBRHH registers can be read or written in 16-bit units.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH registers can be read or written in 8-bit units.

Address: RIICnBRH: <RIICn_base> + 0038_H

RIICnBRHL: <RIICn_base> + 0038_H, RIICnBRHH: <RIICn_base> + $003A_H$

RIICnBRHLL: <RIICn_base> + 0038_H, RIICnBRHLH: <RIICn_base> + 0039_H,
RIICnBRHHL: <RIICn_base> + $003A_H$, RIICnBRHHH:<RIICn_base> + $003B_H$

Value after reset: 0000 $00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22.23 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.

(1) When RIICnFER.SCLE = 0
Transfer rate = 1 / {[(RIICnBRH+1) + (RIICnBRL+1)] / IICφ*1 + tr + tf}
Duty cycle = {tr+[(RIICnBRH+1)/ IICφ]}/{ tr + tf + [(RIICnBRH+1) + (RIICnBRL+1)]/IICφ}

(2) When RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ = PCLK
Transfer rate = 1 / {[(RIICnBRH+3) + (RIICnBRL+3)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3)/ IICφ]}/{ tr + tf + [(RIICnBRH+3) + (RIICnBRL+3)]/IICφ}

(3) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ = PCLK
Transfer rate =1 / {[(RIICnBRH+3+nf) + (RIICnBRL+3+nf)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+3+nf) + (RIICnBRL+3+nf)]/ IICφ}

(4) RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2) + (RIICnBRL+2)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2)/ IICφ]}/{ tr + tf + [(RIICnBRH+2) + (RIICnBRL+2)]/IICφ}

(5) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2+nf) + (RIICnBRL+2+nf)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+2+nf) + (RIICnBRL+2+nf)]/ IICφ}

tf: SCL line falling time [ns]*2
tr: SCL line rising time [ns]*2
nf: Digital noise filter stage
Duty cycle: 0% < Duty < 100%

Note 1. As for IICφ, see CKS[2:0] in **Section 22.3.4, RIICnMR1 — I2C Bus Mode Register 1**.

Note 2. The SCL line rising time [tr] and SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C bus standard from NXP Semiconductors.

Table 22.24 lists examples of the RIICnBRH and RIICnBRL register settings when the SCL synchronization circuit is not used.

Table 22.24 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	8					10					12.5				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	100_B	22	$F6_H$	25	$F9_H$	101_B	13	ED_H	15	EF_H	101_B	16	$F0_H$	20	$F4_H$
50	010_B	16	$F0_H$	19	$F3_H$	010_B	21	$F5_H$	24	$F8_H$	011_B	12	EC_H	15	EF_H
100	001_B	15	EF_H	18	$F2_H$	001_B	19	$F3_H$	23	$F7_H$	001_B	24	$F8_H$	29	FD_H
400	000_B	4	$E4_H$	10	EA_H	000_B	5	$E5_H$	12	EC_H	000_B	7	$E7_H$	16	$F0_H$

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	16					20					25				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	101_B	22	$F6_H$	25	$F9_H$	110_B	13	ED_H	15	EF_H	110_B	16	$F0_H$	20	$F4_H$
50	011_B	16	$F0_H$	19	$F3_H$	011_B	21	$F5_H$	24	$F8_H$	100_B	12	EC_H	15	EF_H
100	010_B	15	EF_H	18	$F2_H$	010_B	19	$F3_H$	23	$F7_H$	010_B	24	$F8_H$	29	FD_H
400	000_B	9	$E9_H$	20	$F4_H$	000_B	11	EB_H	25	$F9_H$	001_B	7	$E7_H$	16	$F0_H$

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	30					33					40				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	110_B	20	$F4_H$	24	$F8_H$	110_B	22	$F6_H$	26	FA_H	111_B	13	ED_H	15	EF_H
50	100_B	15	EF_H	18	$F2_H$	100_B	17	$F1_H$	20	$F4_H$	100_B	21	$F5_H$	24	$F8_H$
100	011_B	14	EE_H	17	$F1_H$	011_B	16	$F0_H$	19	$F3_H$	011_B	19	$F3_H$	23	$F7_H$
400	001_B	8	$E8_H$	19	$F3_H$	001_B	9	$E9_H$	21	$F5_H$	001_B	11	EB_H	25	$F9_H$

CAUTION

CBRH/ICBRL settings in these tables are calculated using the following values:
SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I2C bus standard from NXP Semiconductors.

22.3.15 RIICnDRT — I2C Bus Transmit Data Register

Access: RIICnDRT register can be read or written in 32-bit units.
RIICnDRTL and RIICnDRTH registers can be read or written in 16-bit units.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH registers can be read or written in 8-bit units.

Address: RIICnDRT: <RIICn_base> + $003C_H$

RIICnDRTL: <RIICn_base> + $003C_H$, RIICnDRTH: <RIICn_base> + $003E_H$

RIICnDRTLL: <RIICn_base> + $003C_H$, RIICnDRTLH: <RIICn_base> + $003D_H$,
RIICnDRTHL: <RIICn_base> + $003E_H$, RIICnDRTHH: <RIICn_base> + $003F_H$

Value after reset: $0000\ 00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I2C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTRIICnTI) request is generated. When writing to bit 31 to 8, write the value after reset.

22.3.16 RIICnDRR — I2C Bus Receive Data Register

Access: RIICnDRR register is a read-only register that can be read in 32-bit units.
RIICnDRRL and RIICnDRRH registers are the read-only registers that can be read in 16-bit units.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH registers are the read-only registers that can be read in 8-bit units.

Address: RIICnDRR: <RIICn_base> + 0040_H

RIICnDRRL: <RIICn_base> + 0040_H, RIICnDRRH: <RIICn_base> + 0042_H

RIICnDRRLL: <RIICn_base> + 0040_H, RIICnDRRLH: <RIICn_base> + 0041_H,
RIICnDRRHL: <RIICn_base> + 0042_H, RIICnDRRHH: <RIICn_base> + 0043_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I2C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive complete interrupt (INTRIICnRI) request is generated.

If RIIC receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

22.3.17 RIICnDRS — I2C Bus Shift Register

Access: This register is not accessible.

Address: —

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is a shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

22.4 Operation

22.4.1 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive end, transmit data empty, and transmit end.

Table 22.25 lists details of the several interrupt requests. The receive complete and transmit data empty interrupt request are both capable of launching data transfer by the DMAC.

Table 22.25 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMACA Launching	Interrupt Condition
INTRIICnTI	Transmit Data Empty	TDRE	Possible	TDRE = 1 and TIE = 1
INTRIICnTEI	Transmit End	TEND	Not possible	TEND = 1 and TEIE = 1
INTRIICnRI	Receive End	RDRF	Possible	RDRF = 1 and RIE = 1
INTRIICnEE	Transfer Error/ Event Generation	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1

Clear or mask the each flag during interrupt handling.

CAUTIONS

1. **There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.**
2. **Since INTRIICnTI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.TDRE flag (a condition for INTRIICnTI) is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).**
3. **Since INTRIICnRI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.RDRF flag (a condition for INTRIICnRI) is automatically cleared to 0 when data are read from RIICnDRR.**
4. **When using the INTRIICnTEI interrupt, clear the RIICnSR2.TEND flag in the INTRIICnTEI interrupt processing.**
   Note that the RIICnSR2.TEND flag is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).
5. **Even when using the INTRIICnEE interrupt, clear the RIICnSR2.NACKF flag after detecting the stop condition (RIICnSR2.STOP flag = 1) as shown in Figure 22.6, Example of Master Transmission Flowchart, Figure 22.10, Example of Master**

Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes), Figure 22.11, Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More), and Figure 22.15, Example of Slave Transmission Flowchart. If the RIICnSR2.NACKF flag is cleared before detecting the STOP condition, the automatic low-hold function for SCL (Section 22.4.7.1, Function to Prevent Wrong Transmission of Transmit Data) may be enabled.

22.4.2 I2C Operation

22.4.2.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 22.3 shows the I2C bus format, and **Figure 22.4** shows the I2C bus timing.

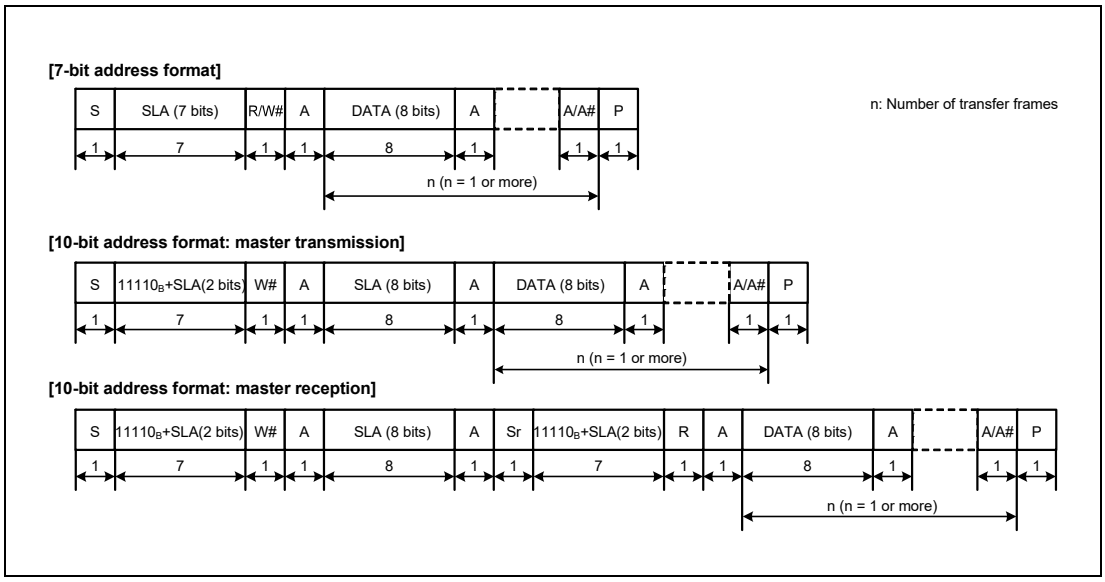


Figure 22.3 I2C Bus Format

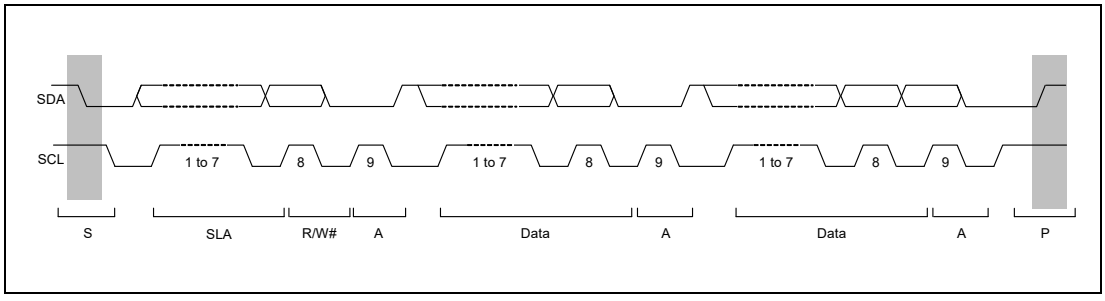


Figure 22.4 I2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

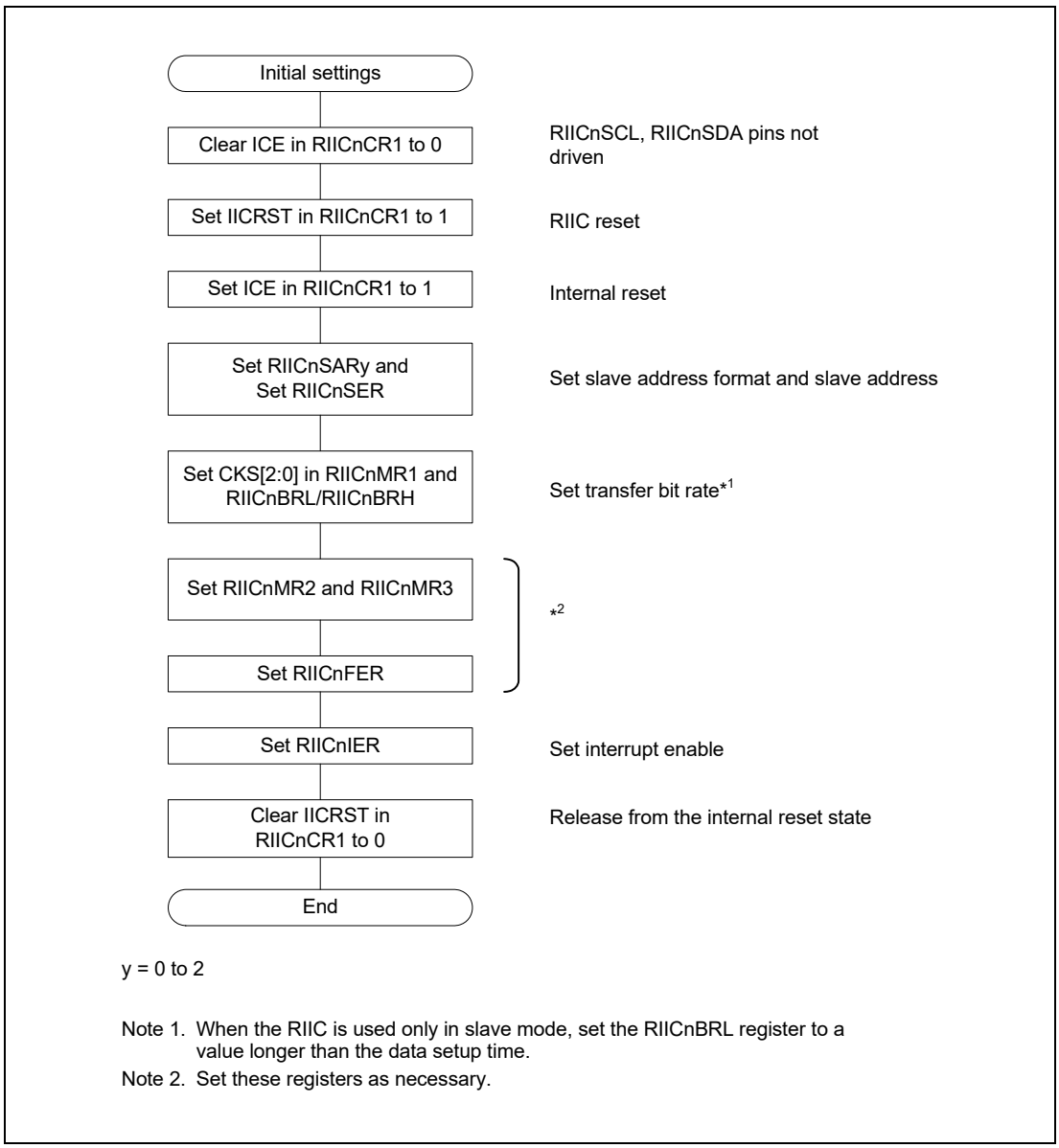
Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

22.4.2.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 22.5**. Make initial settings for the RIIC once when starting the RIIC.



Note 1. When the RIIC is used only in slave mode, set the RIICnBRL register to a value longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 22.5 Example of RIIC Initialization Flowchart

22.4.2.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 22.6** shows an example of usage of master transmission and **Figure 22.7** to **Figure 22.9** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 22.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS and MST bits to 1.

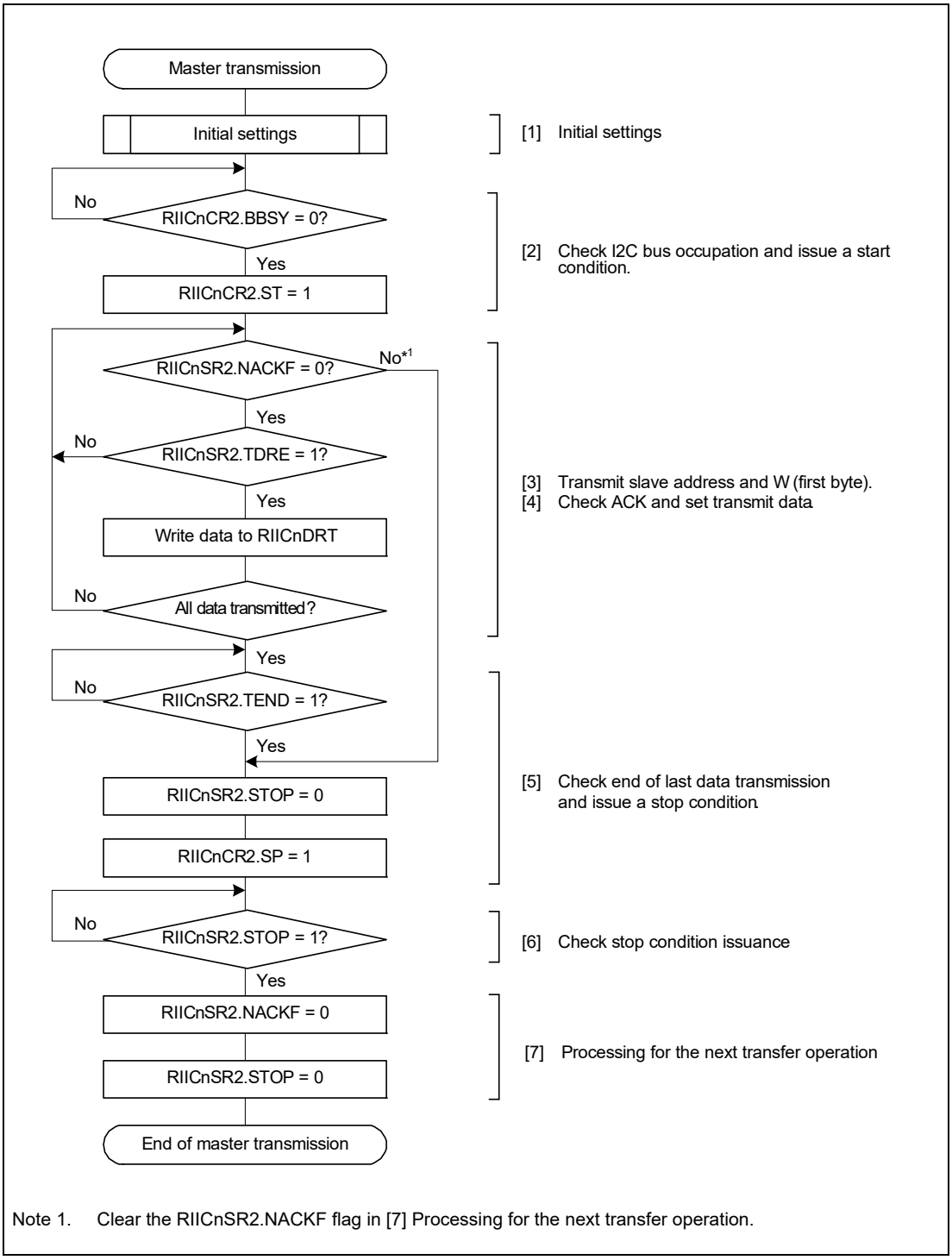
(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0_B, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After all bytes of data for transmission have been written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.



Note 1. Clear the RIICnSR2.NACKF flag in [7] Processing for the next transfer operation.

Figure 22.6 Example of Master Transmission Flowchart

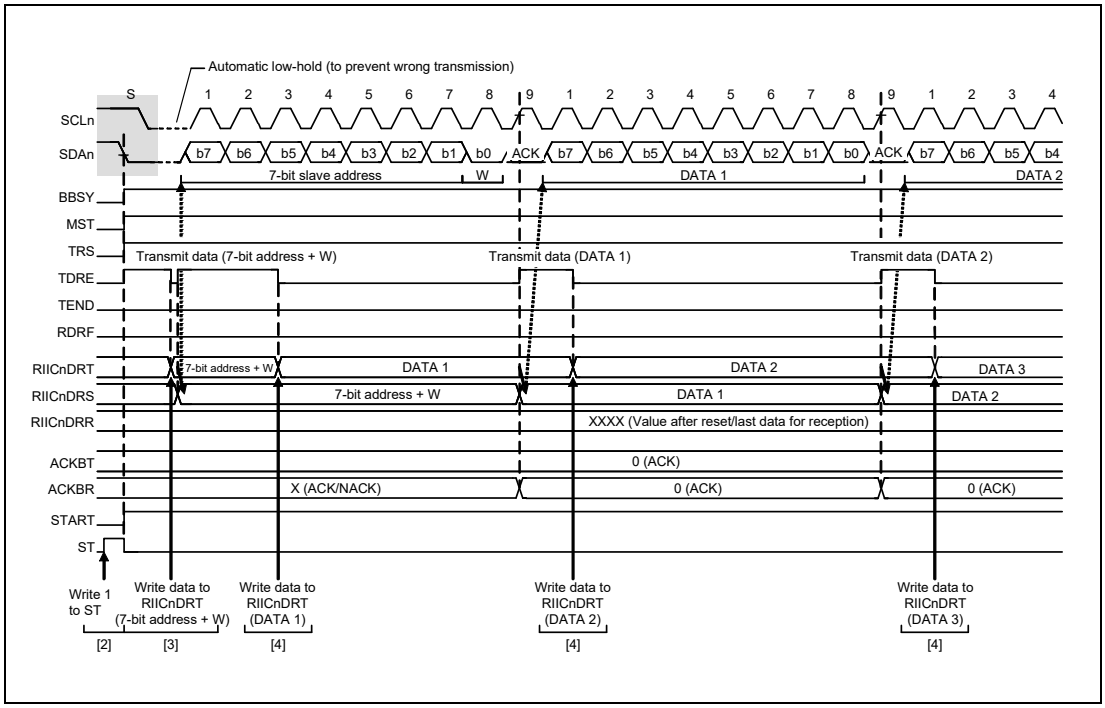


Figure 22.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

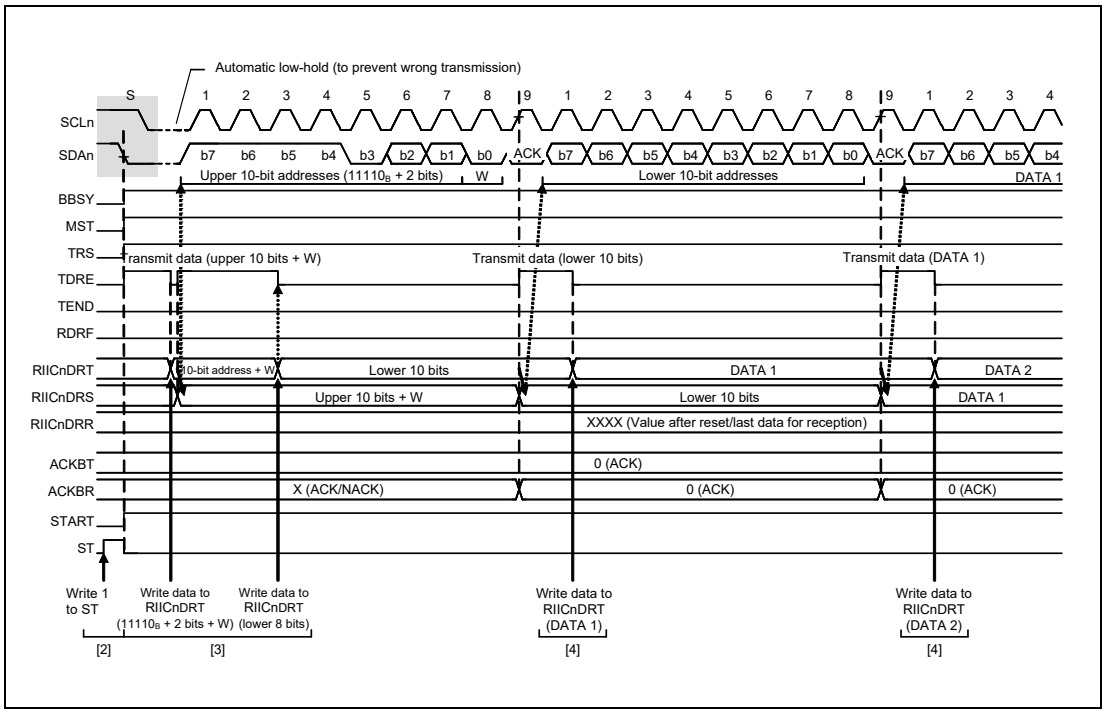


Figure 22.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

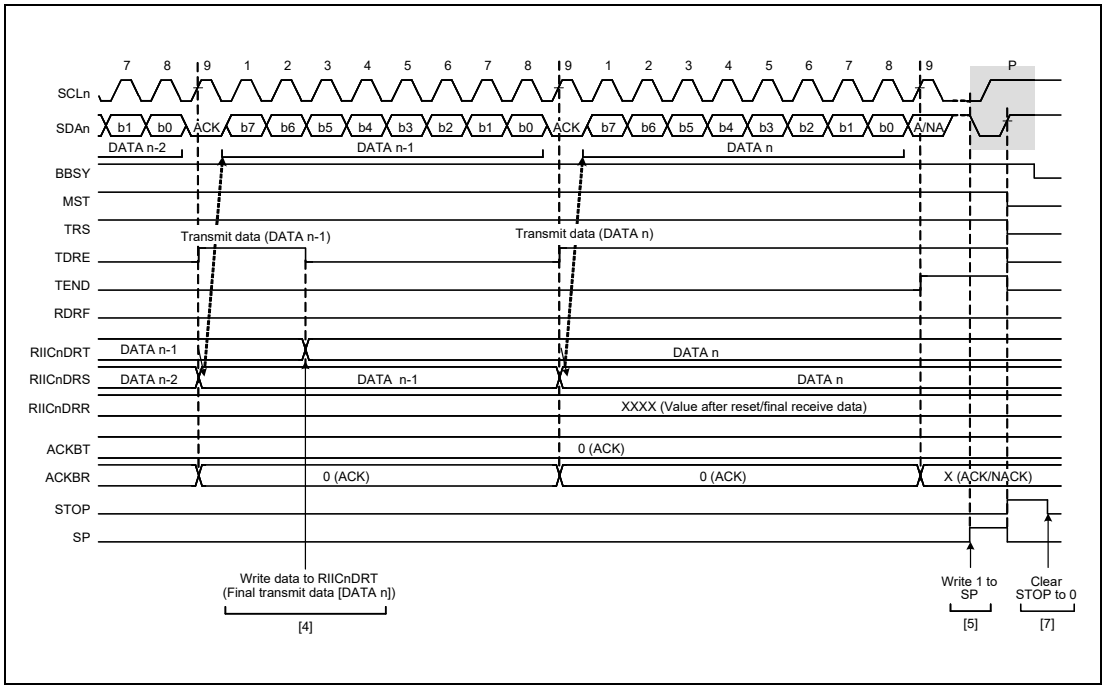


Figure 22.9 Master Transmit Operation Timing (3)

22.4.2.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 22.10 shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 22.11** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 22.12** to **Figure 22.14** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 22.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to transmit the two higher-order bits of the slave address and then the eight lower-order bits of the slave address, and issue a restart condition following generation of the transmission end interrupt (or after TEND = 1) (For details about the operation timing, see **Figure 22.13**). After that, transmitting 1111 0_B plus the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

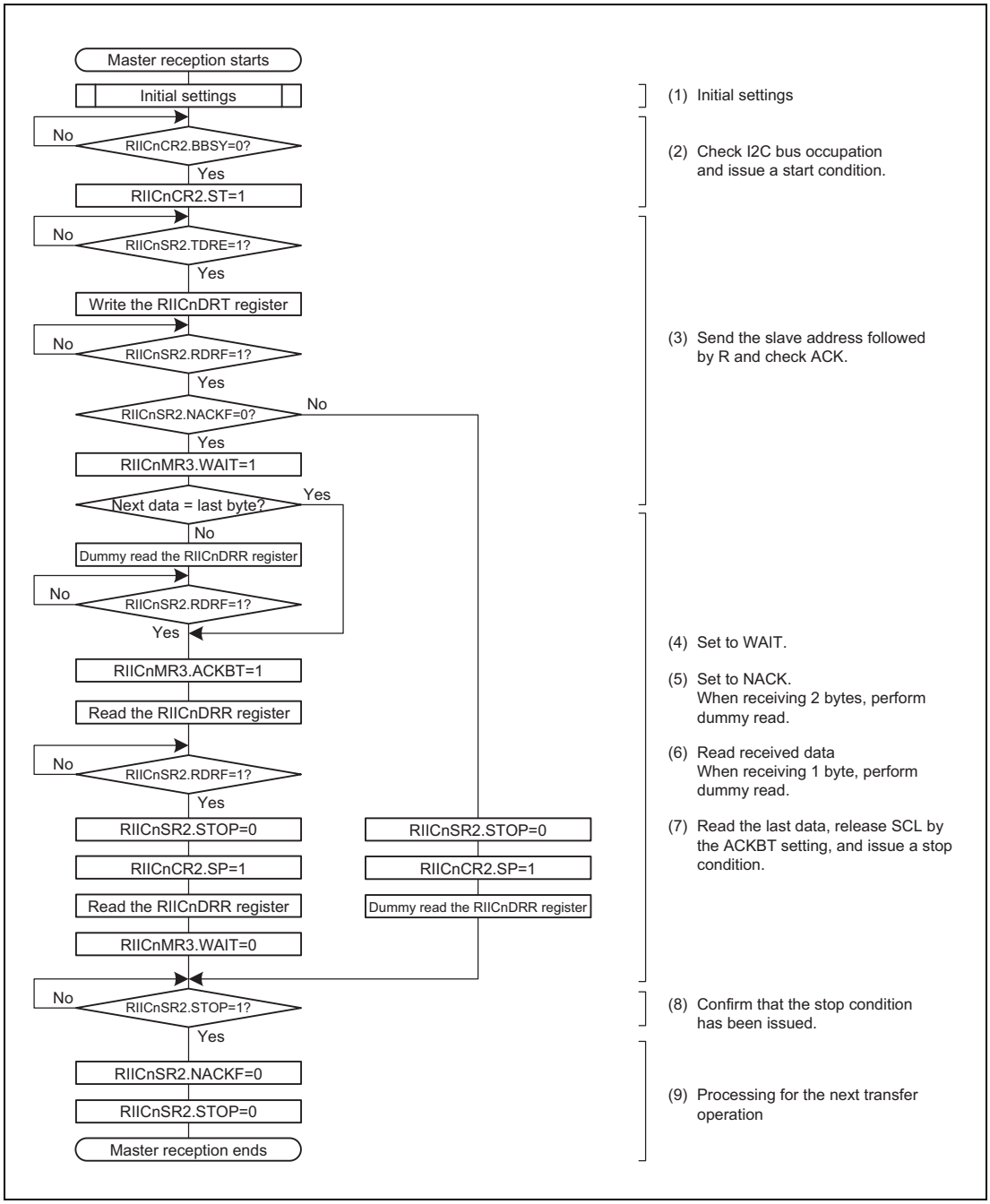


Figure 22.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)

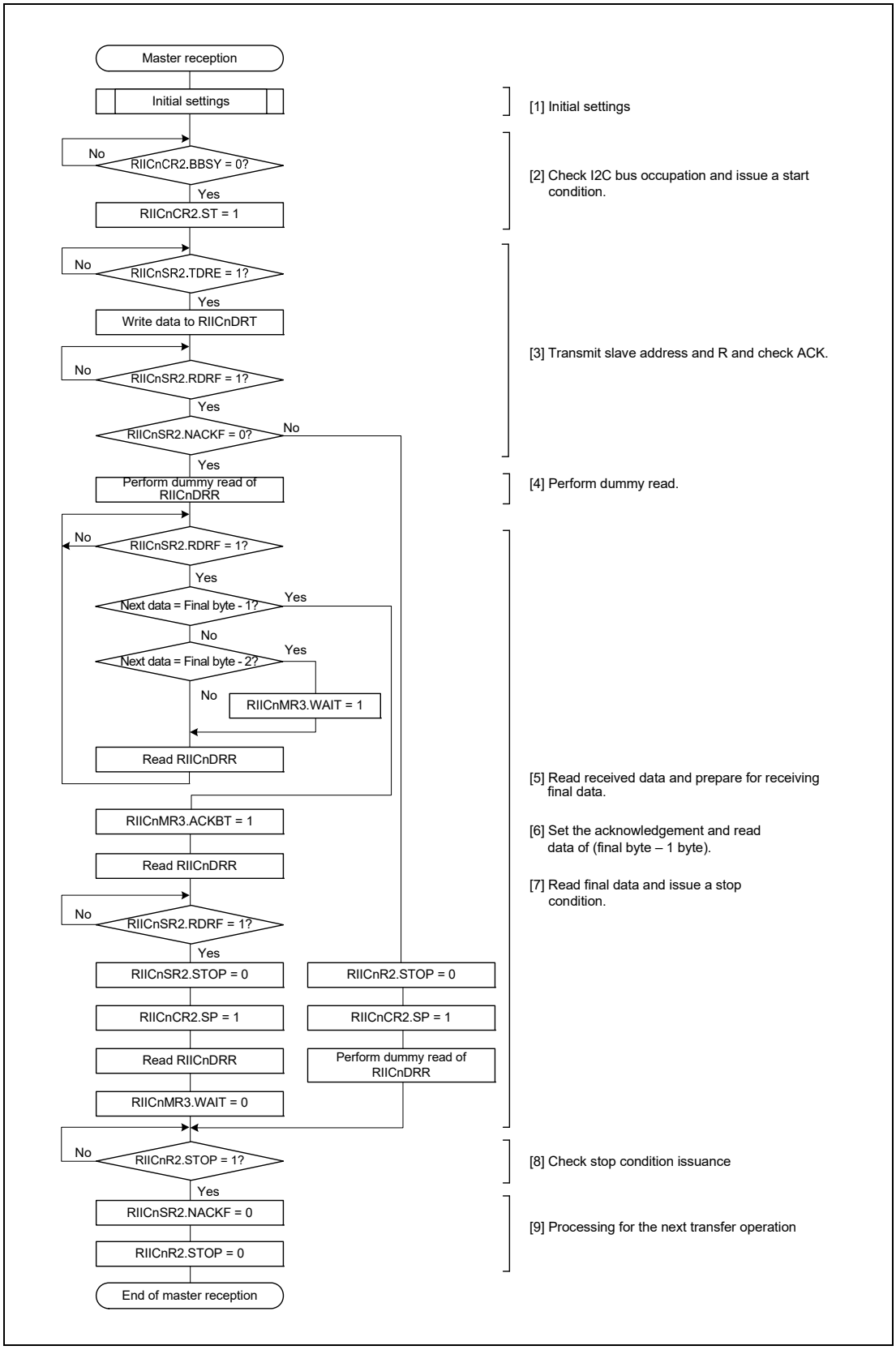


Figure 22.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

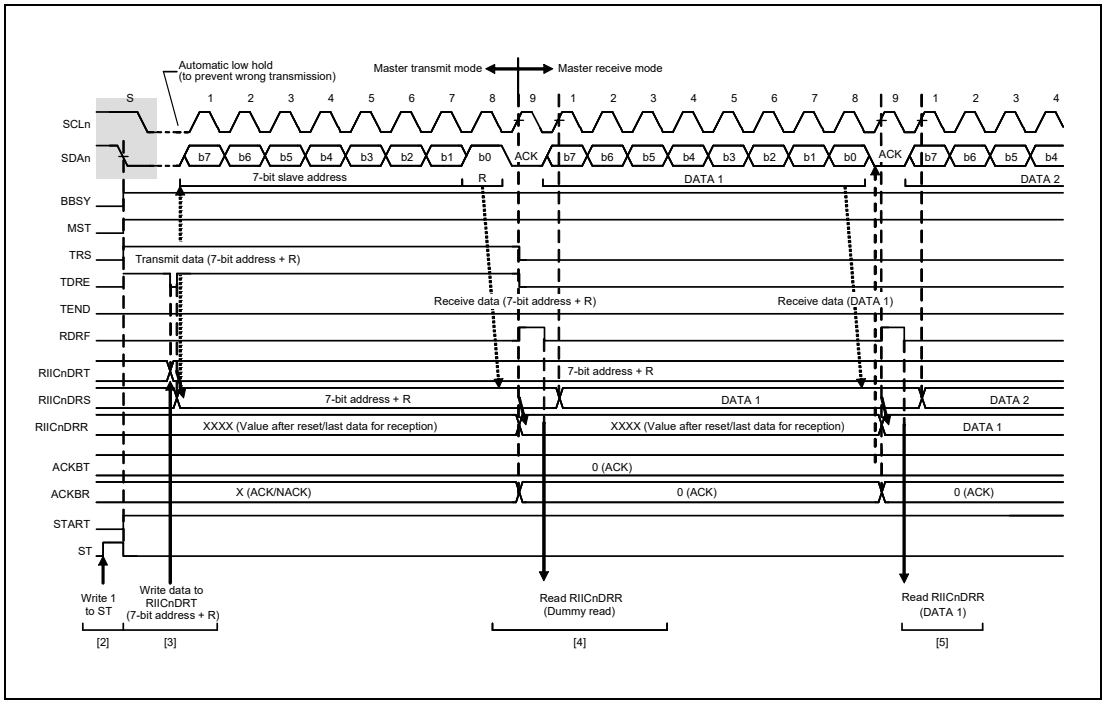


Figure 22.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

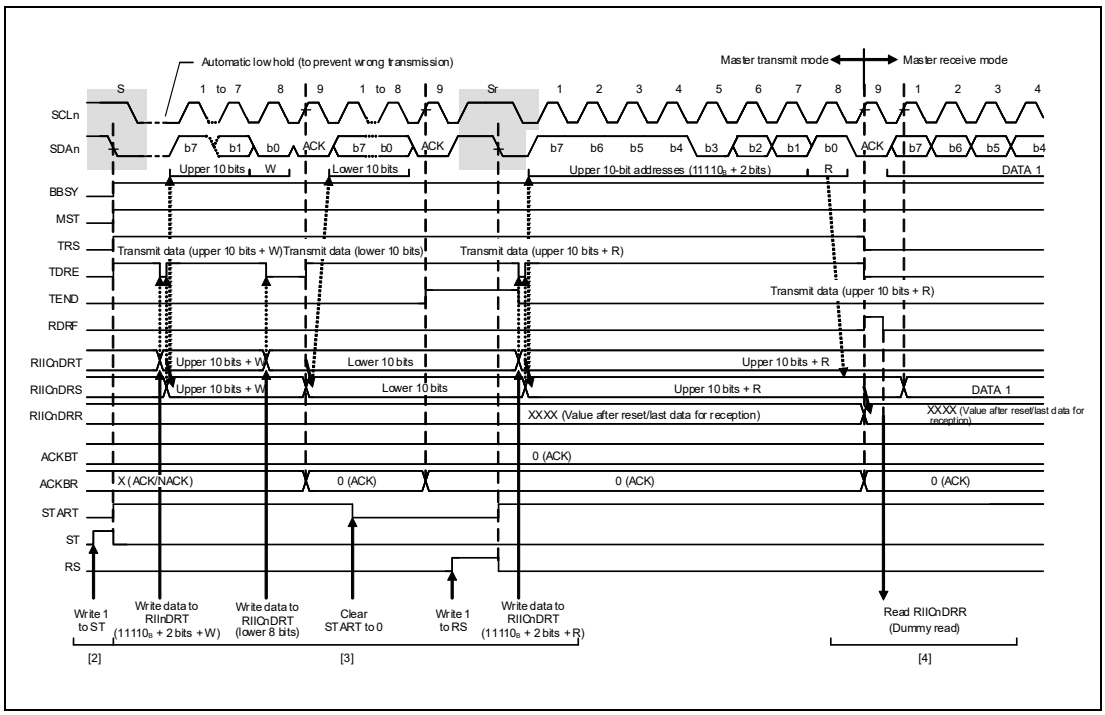


Figure 22.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

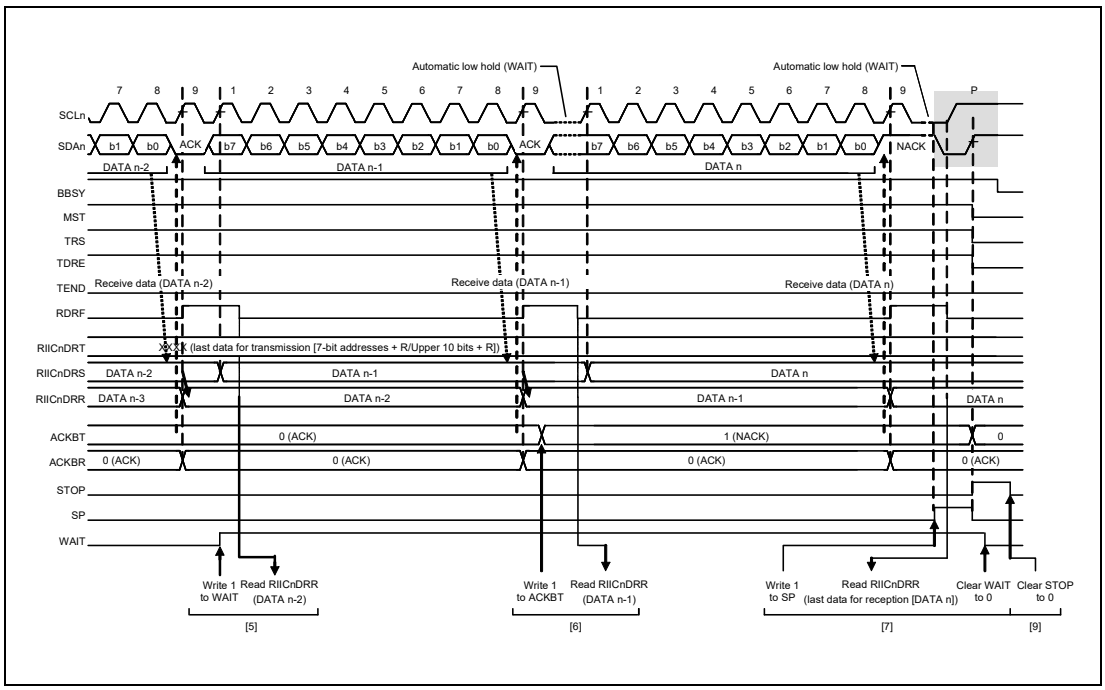


Figure 22.14 Master Receive Operation Timing (3) (when RDRFS = 0)

22.4.2.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 22.5 shows an example of usage of slave transmission and **Figure 22.16** and **Figure 22.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 22.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

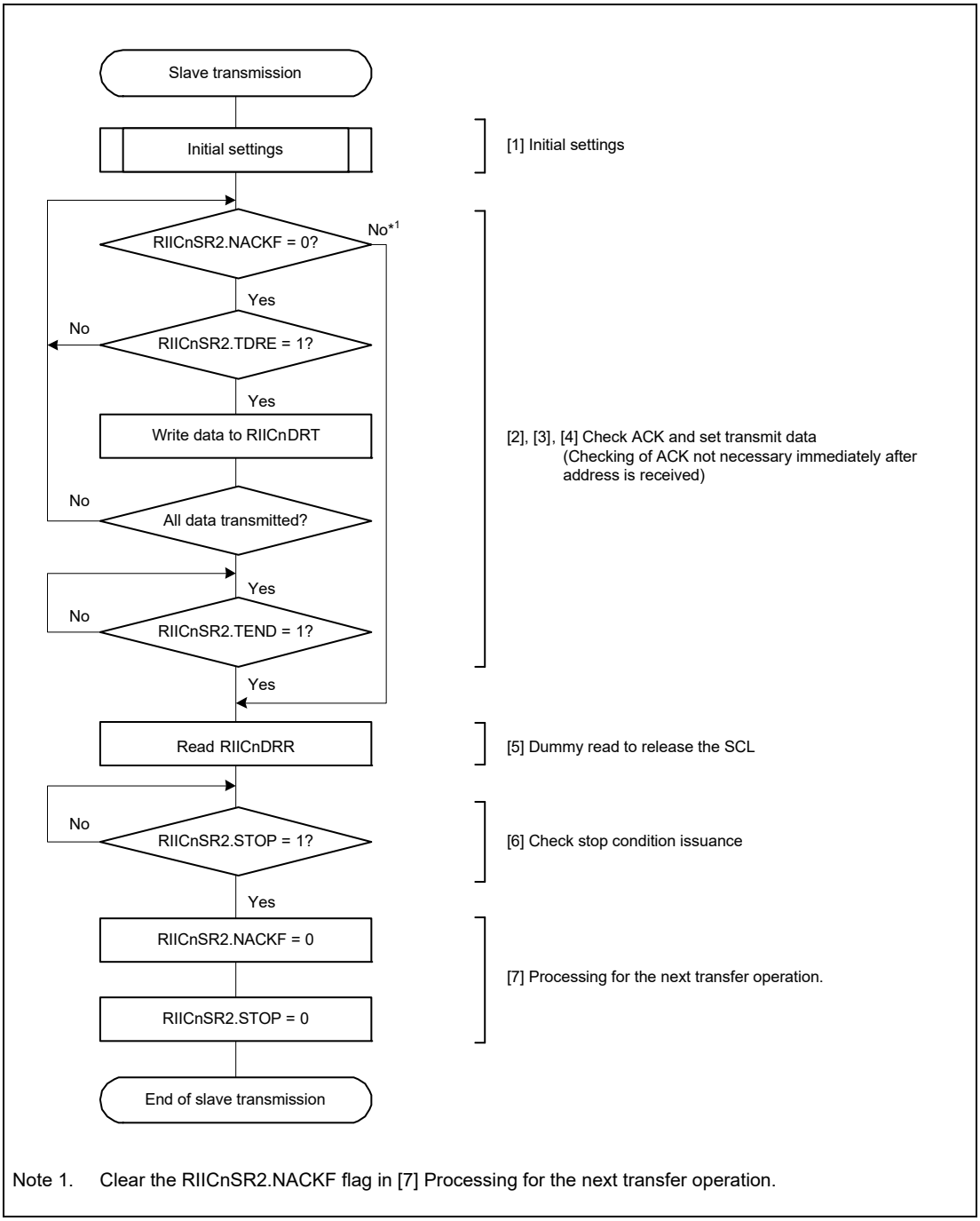
(3) After the RIICnSR2.TDRE flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.



Note 1. Clear the RIICnSR2.NACKF flag in [7] Processing for the next transfer operation.

Figure 22.15 Example of Slave Transmission Flowchart

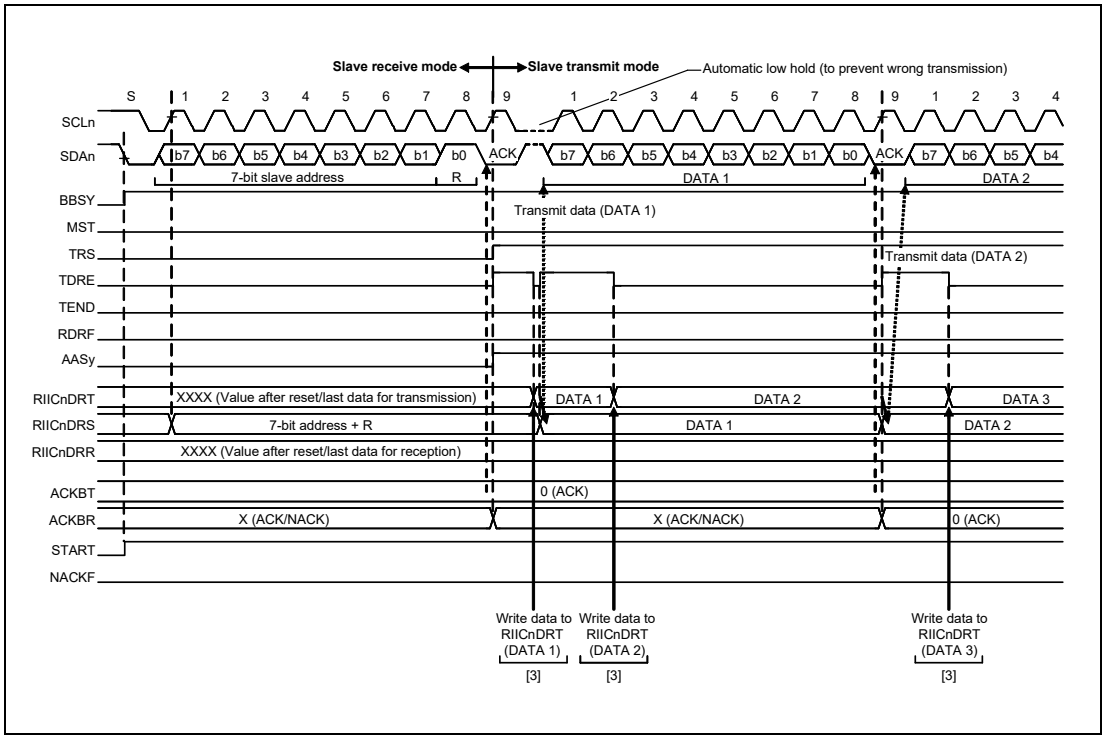


Figure 22.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

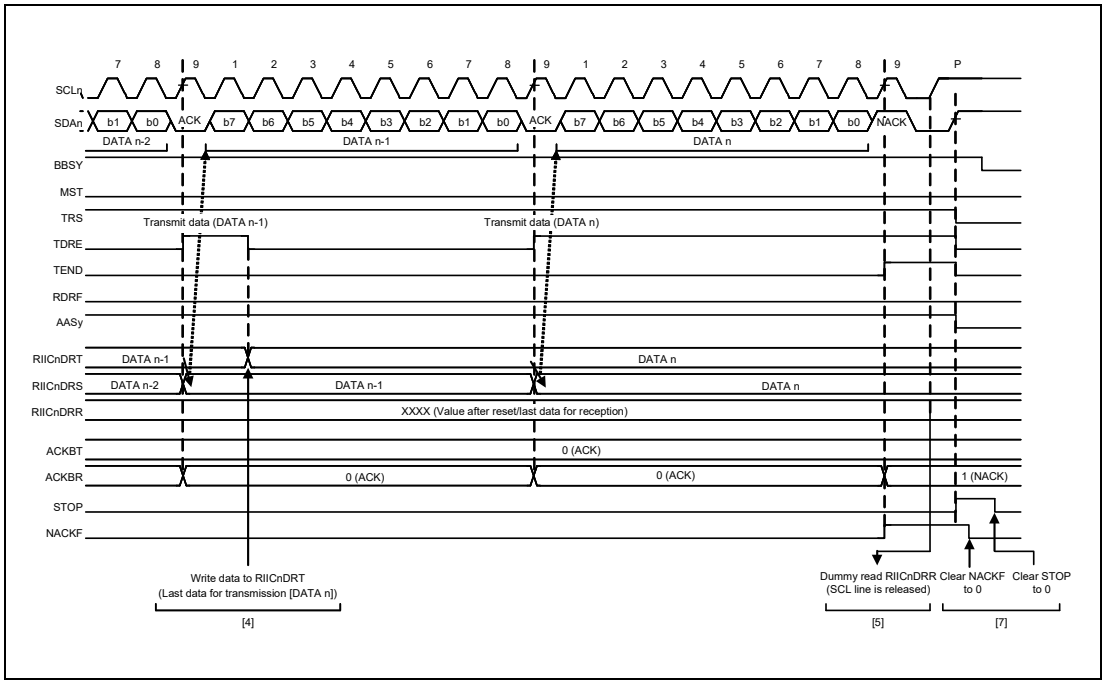


Figure 22.17 Slave Transmit Operation Timing (2)

22.4.2.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 22.18 shows an example of usage of slave reception and **Figure 22.19** and **Figure 22.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 22.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR as the first read operation (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 0. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

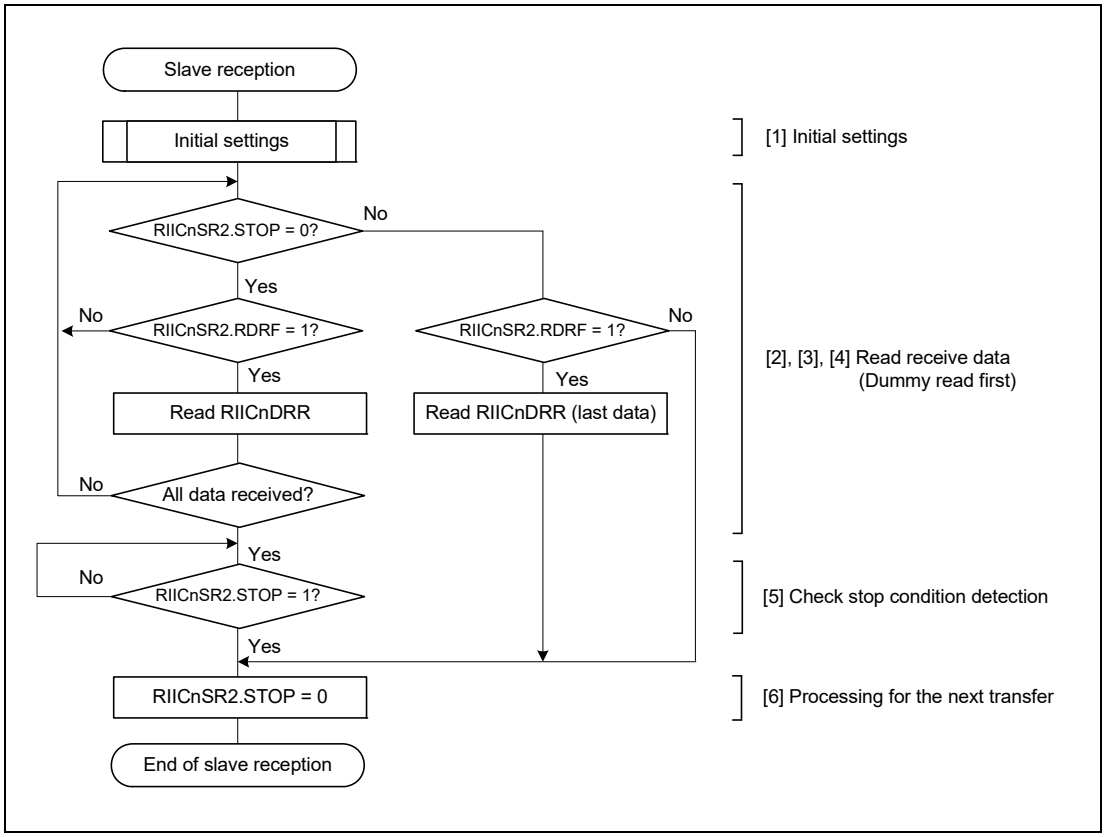


Figure 22.18 Example of Slave Reception Flowchart

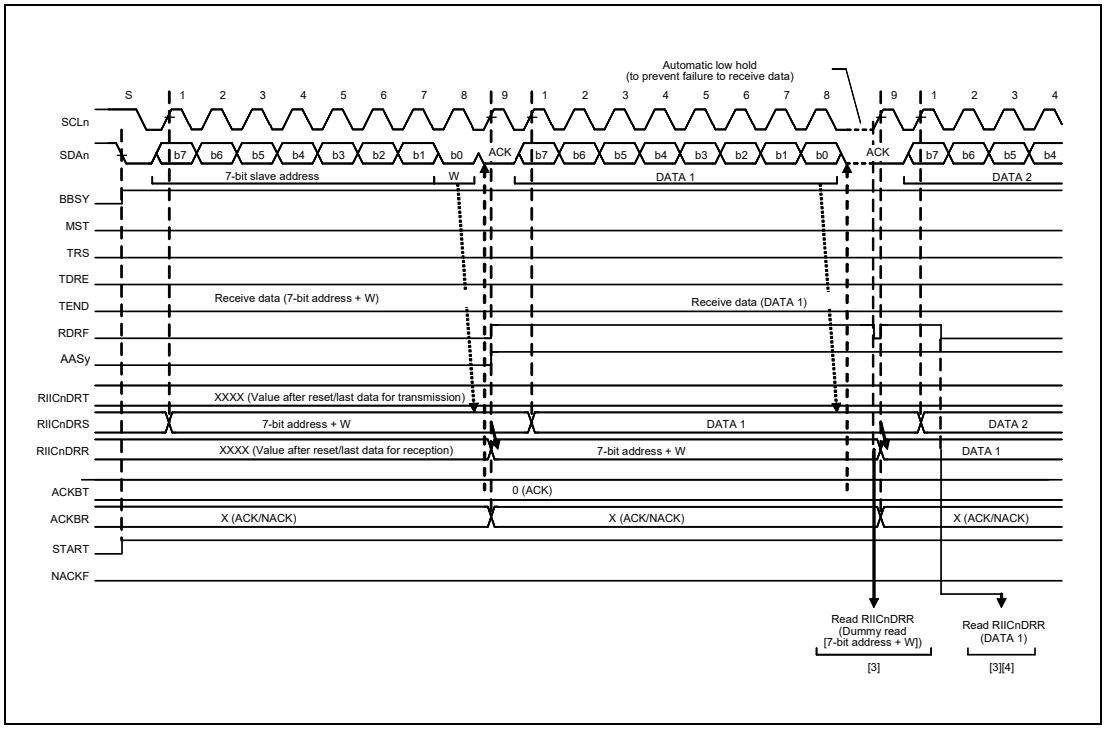


Figure 22.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

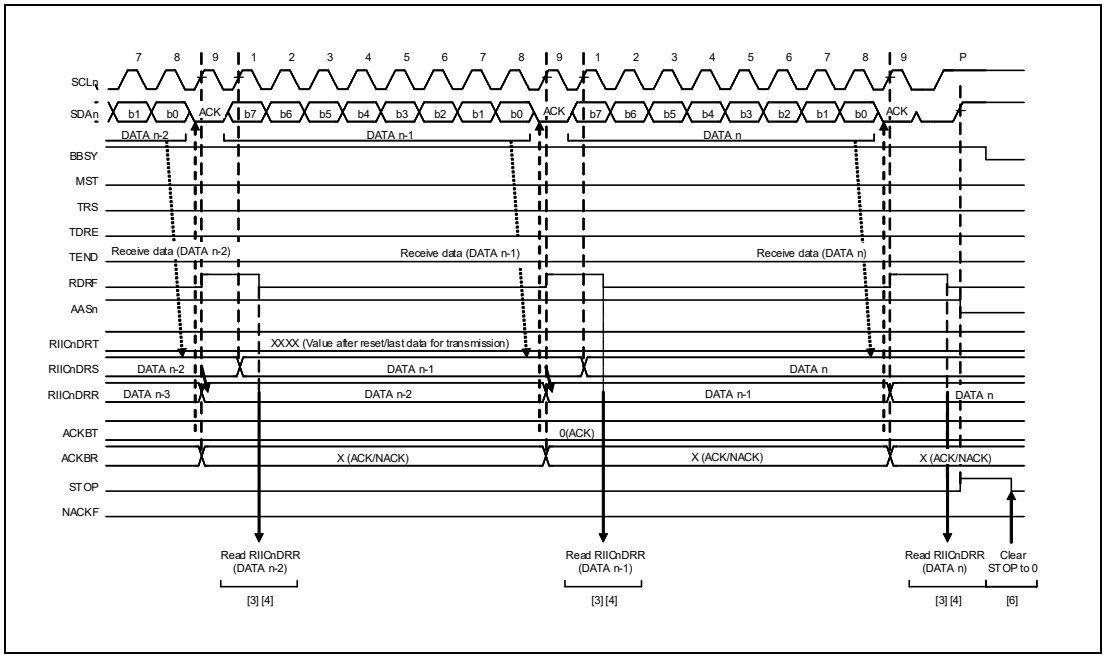


Figure 22.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

22.4.3 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.

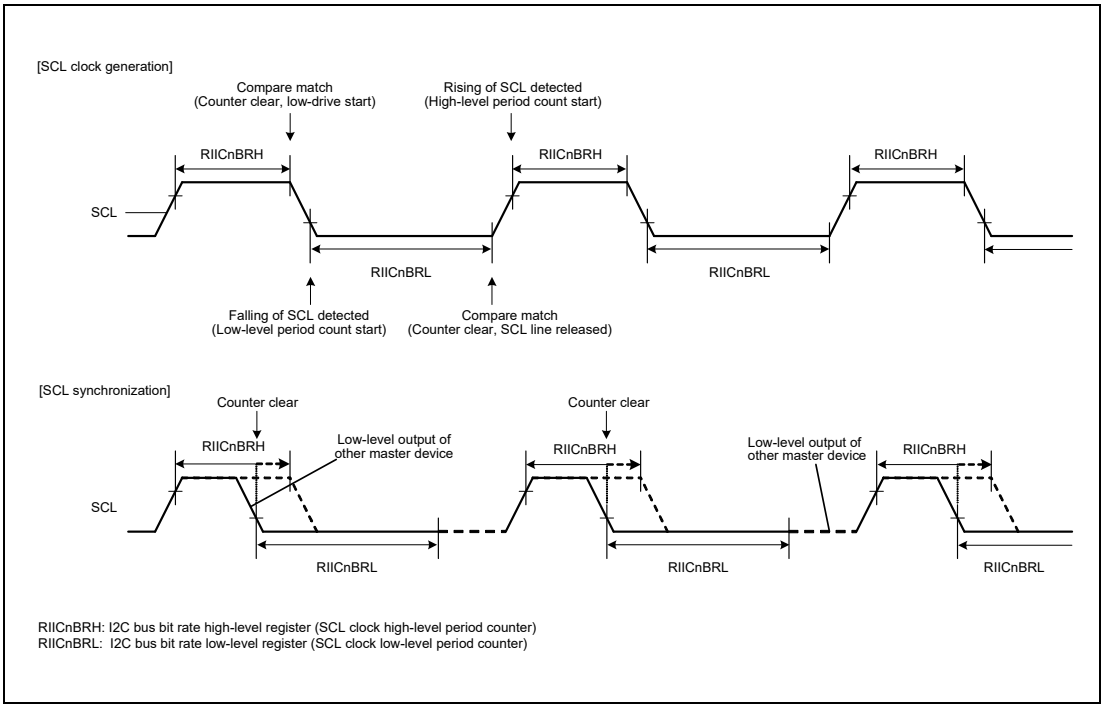


Figure 22.21 Generation and Synchronization of the SCL Signal from the RIIC

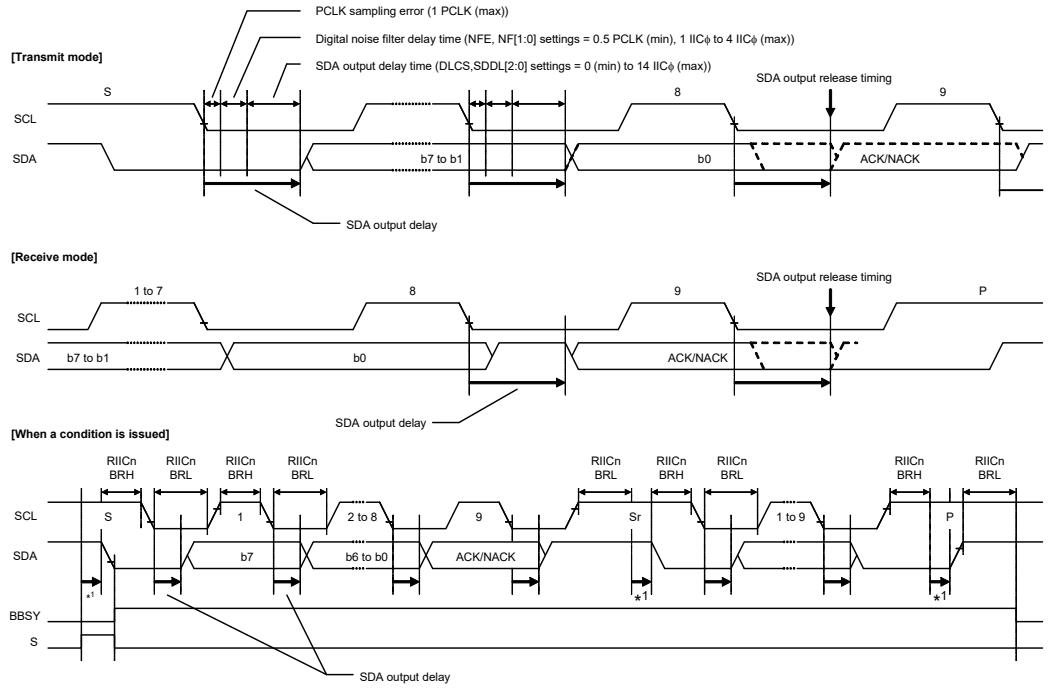
22.4.4 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000_B, and disabled by setting the same bits to 000_B.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits are set to any value other than 000_B), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay facility is set by the SDDL[2:0] bits when a start (S), restart (Sr), or stop (P) condition is issued.

Figure 22.22 SDA Output Delay Facility

22.4.5 Digital Noise-Filter Circuits

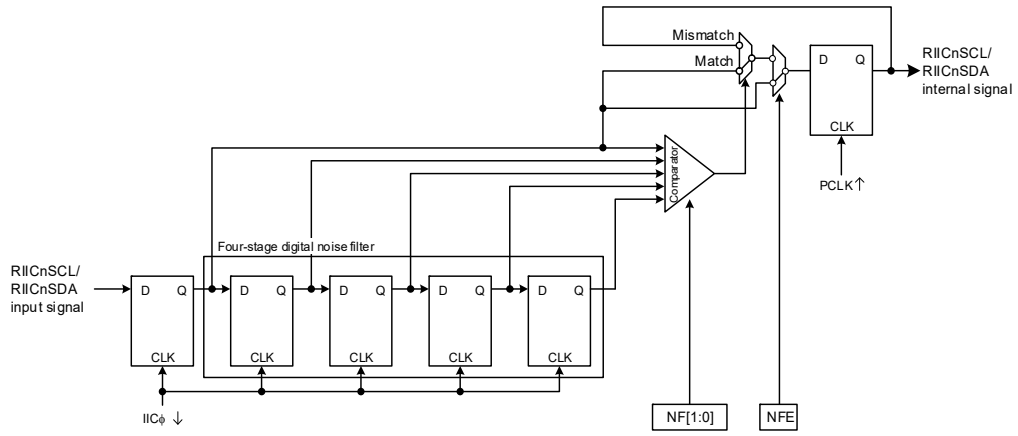
The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 22.23** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed as an internal signal. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small, the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the RIICnFER.NFE bit to 0).



Note: NFE is Noise filter enable bit. (**22.3.7, RIICnFER — I2C Bus Function Enable Register**)
NF[1:0]: Digital noise filter stage selection bit

Figure 22.23 Block Diagram of Digital Noise Filter Circuit

22.4.6 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and device ID address, and also can set 7-bit or 10-bit slave addresses.

22.4.6.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. The RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive complete interrupt (INTRIICnRI) or transmit data empty interrupt (INTRIICnTI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 22.24 to **Figure 22.26** show the AASy flag set timing in three cases.

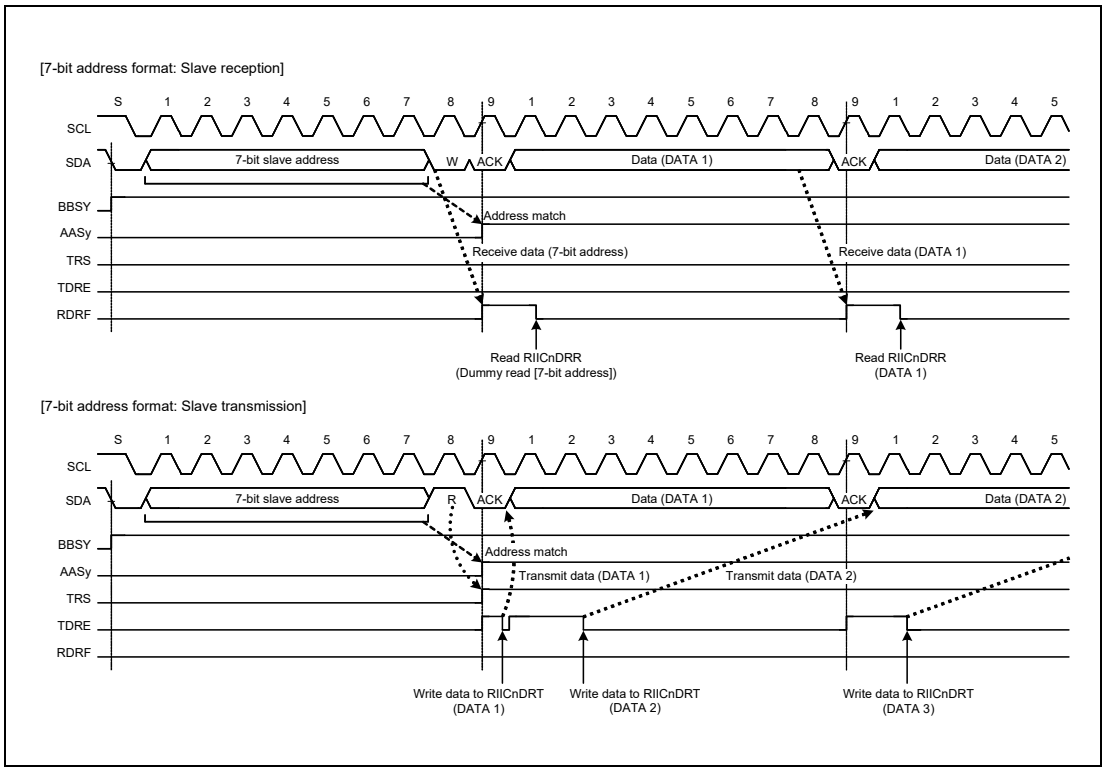


Figure 22.24 AASy Flag Set Timing with 7-Bit Address Format Selected

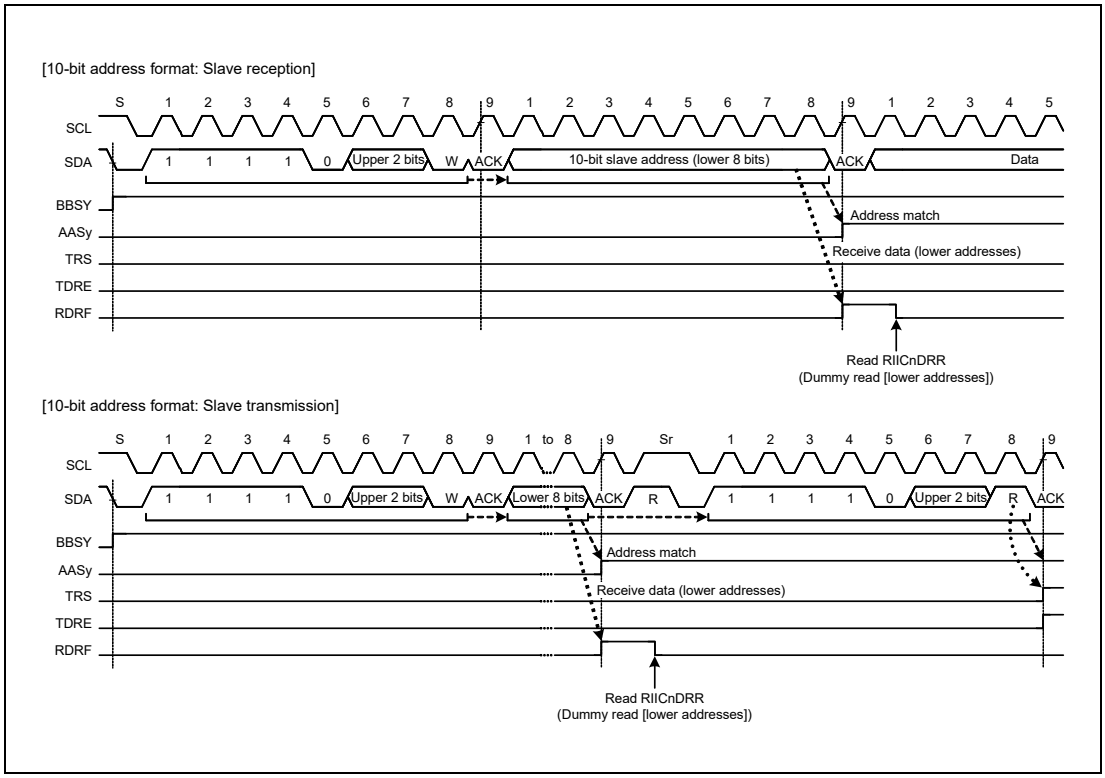


Figure 22.25 AASy Flag Set Timing with 10-Bit Address Format Selected

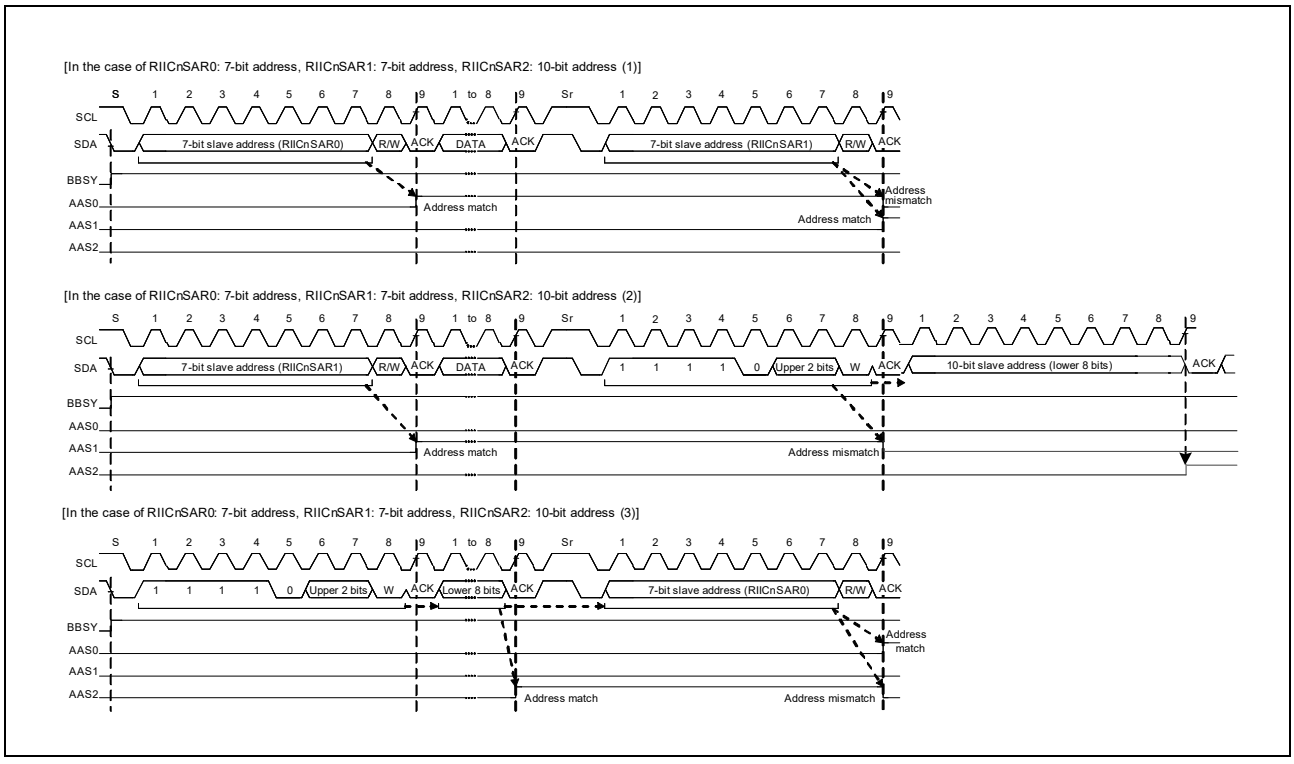


Figure 22.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

22.4.6.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000_B + 0 [W]). This is enabled by setting the RIICnSER.GCE bit to 1.

If the address received after a start or restart condition is issued is 0000 000_B + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive complete interrupt (INTRIICnRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

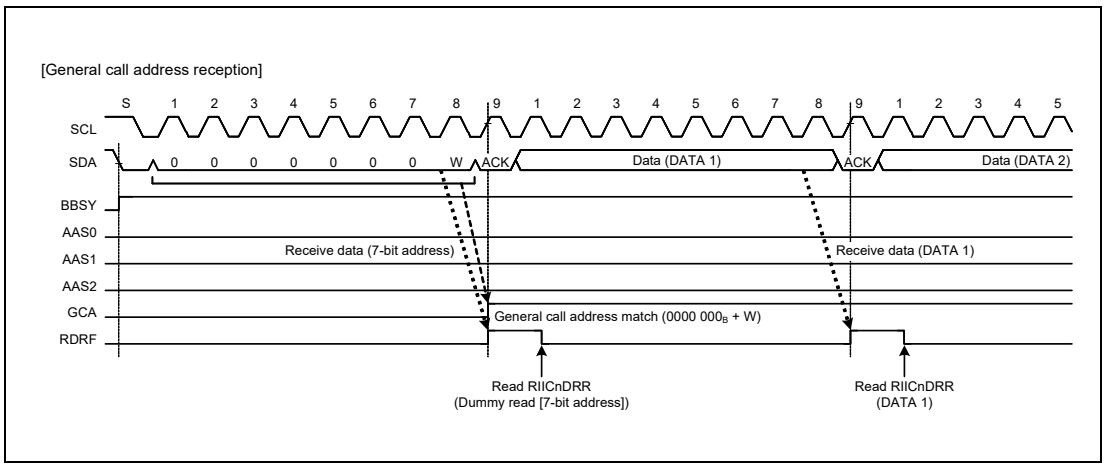


Figure 22.27 Timing of GCA Flag Setting during Reception of General Call Address

22.4.6.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conforming with the I2C bus specification (Rev. 03). When the RIIC receives 1111 100_B as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100_B) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100_B) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I2C Bus Standard from NXP Semiconductors.

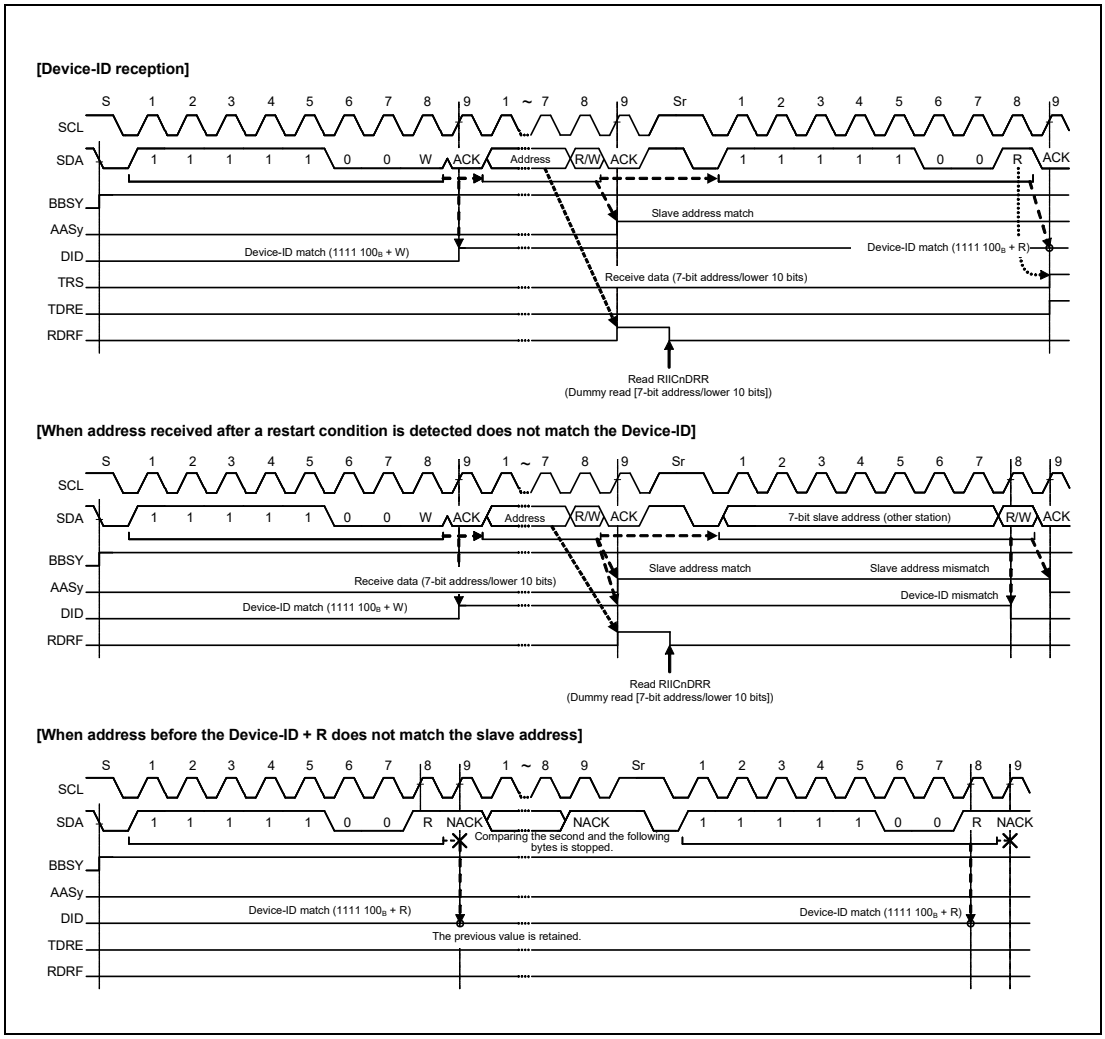


Figure 22.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

22.4.7 Automatic Low-Hold Function for SCL

22.4.7.1 Function to Prevent Wrong Transmission of Transmit Data

To prevent the unintended transmission of erroneous data, this low-hold period is extended until data for transmission have been written. In addition, the RIIC holds the SCL line low over the period until a stop condition is issued and also over the period until the RIICnDRR register is dummy read.

<Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle until a stop condition is issued

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle and the RIICnDRR register is dummy read

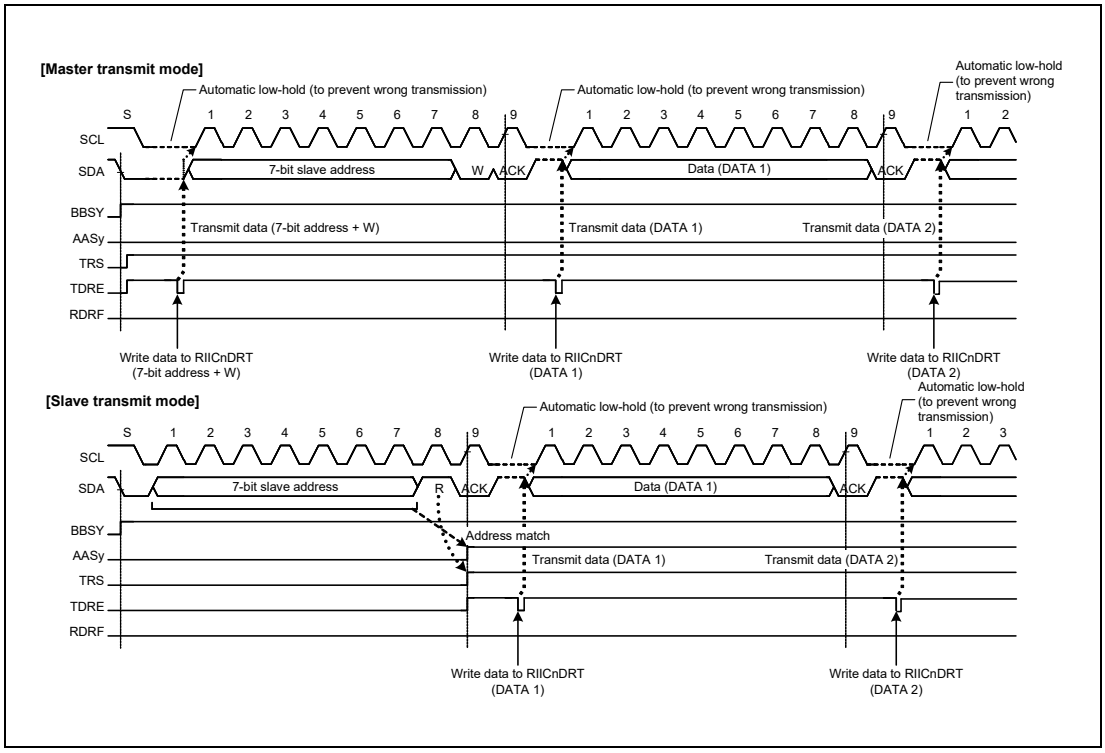


Figure 22.29 Automatic Low-Hold Operation in Transmit Mode

22.4.7.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0 after issue a restart condition or NACKF flag to 0 after issue a stop condition, and then issue a start condition again.

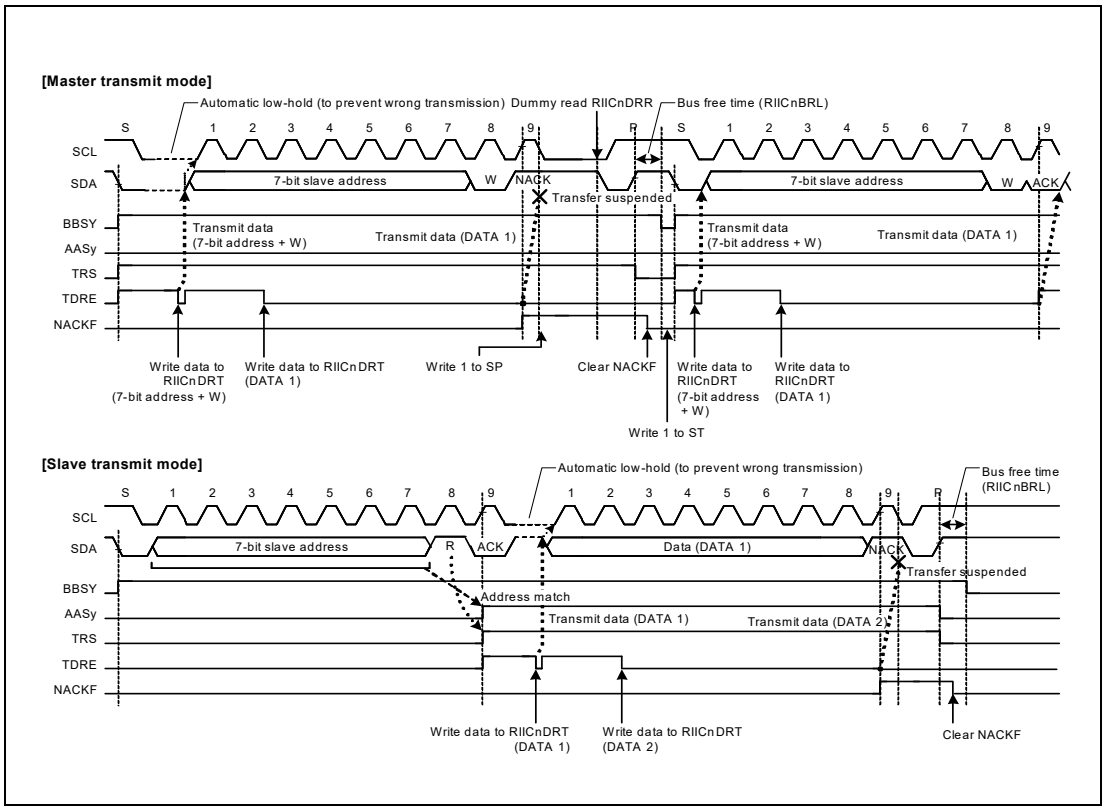


Figure 22.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

22.4.7.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive complete (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.
Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.
When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive complete) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

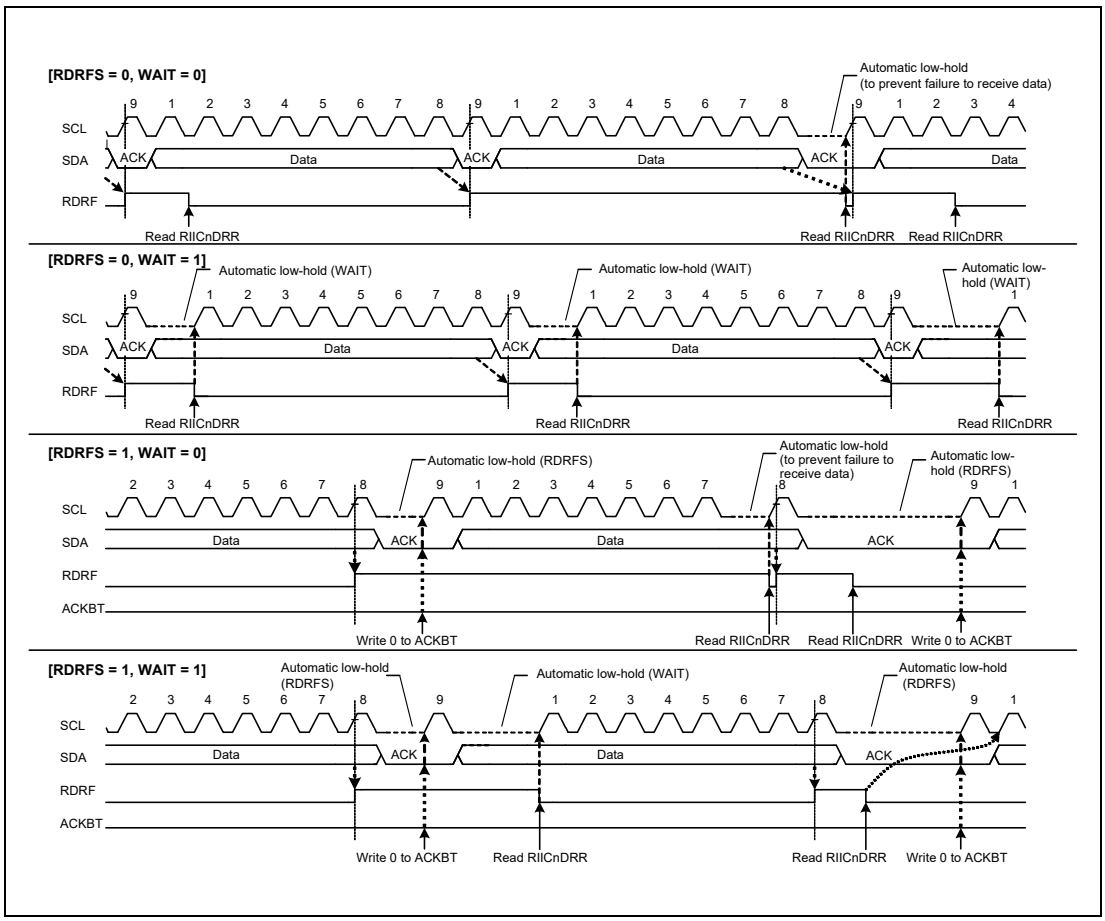


Figure 22.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

22.4.8 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

22.4.8.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if the RIICnCR2.ST bit is set to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC considers itself to have lost in arbitration, so priority is given to transfer by the other master device and no start condition is generated.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line), the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11_B)

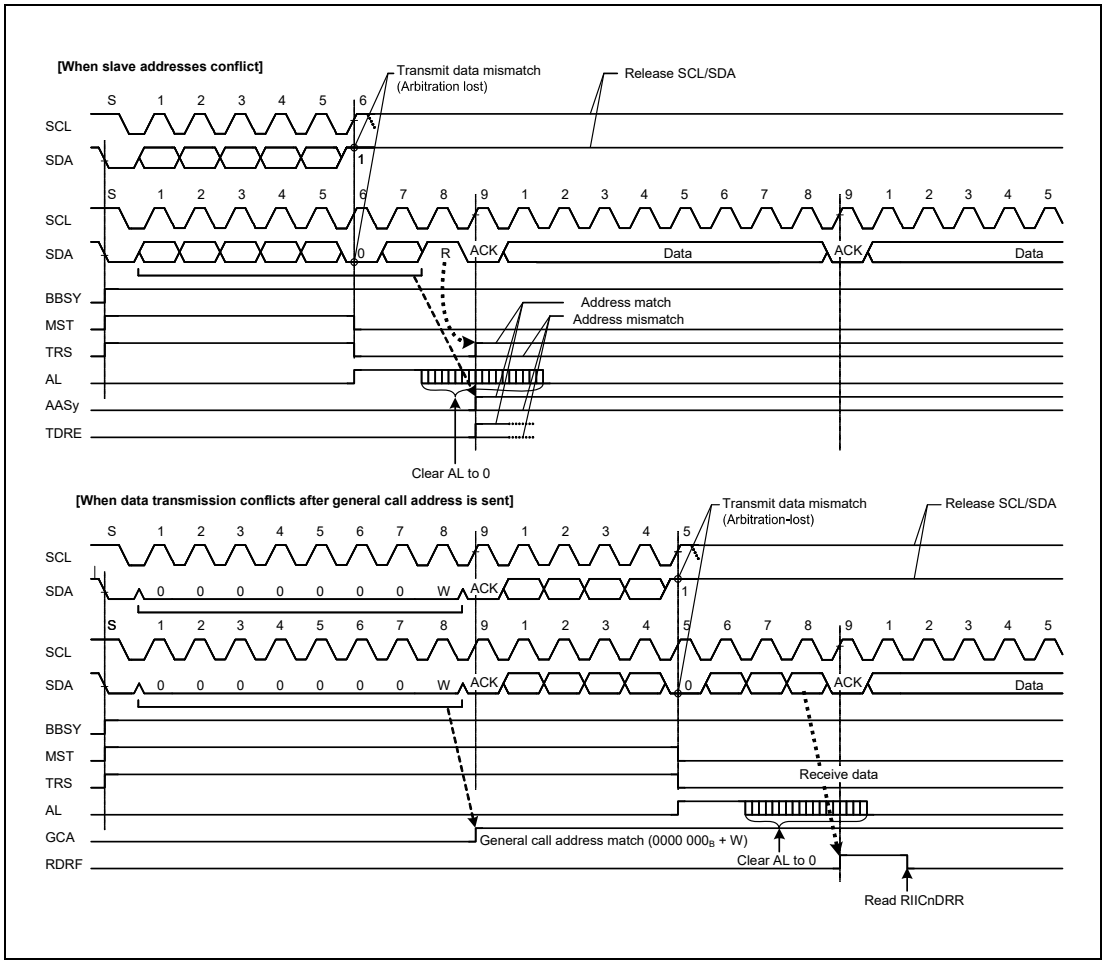


Figure 22.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

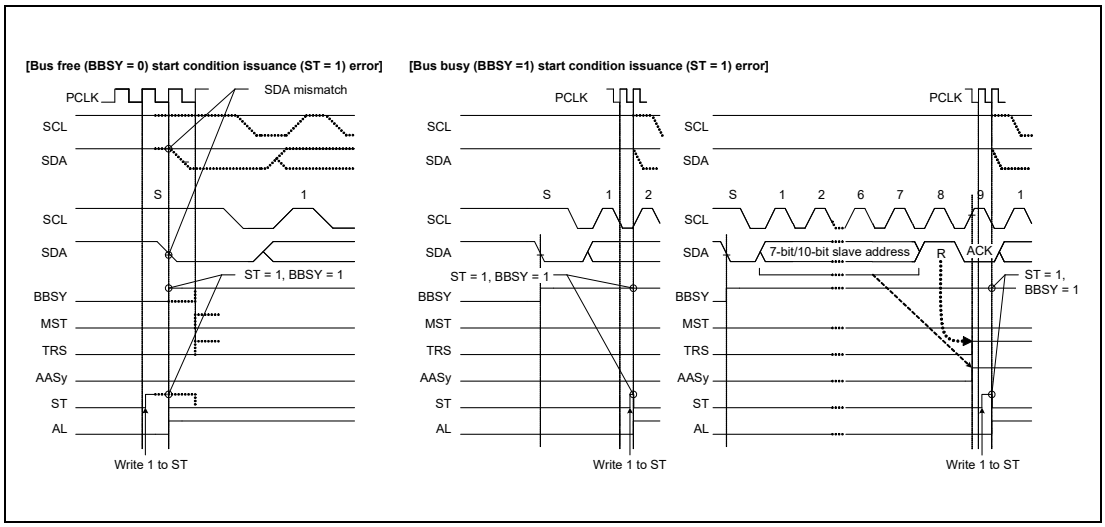


Figure 22.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

The TRS bit is not cleared if 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode.

22.4.8.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line) during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 22.34** shows an example of arbitration-lost detection during transmission of NACK.

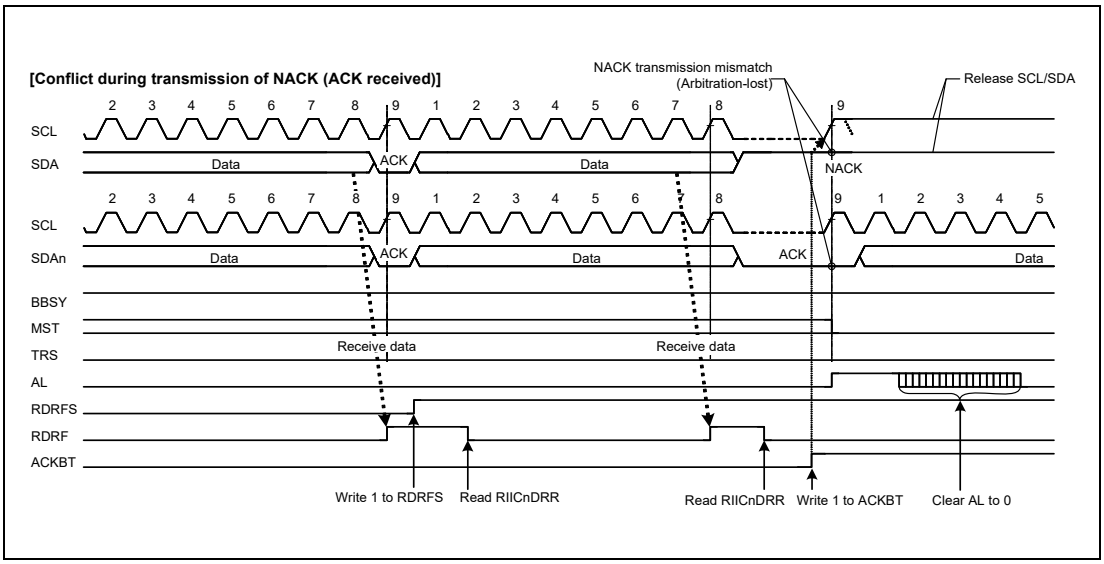


Figure 22.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

22.4.8.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high impedance state, and the low level is detected on the SDA line) in slave transmit mode.

When it loses slave arbitration, the RIIC enters slave receive mode.

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01_B)

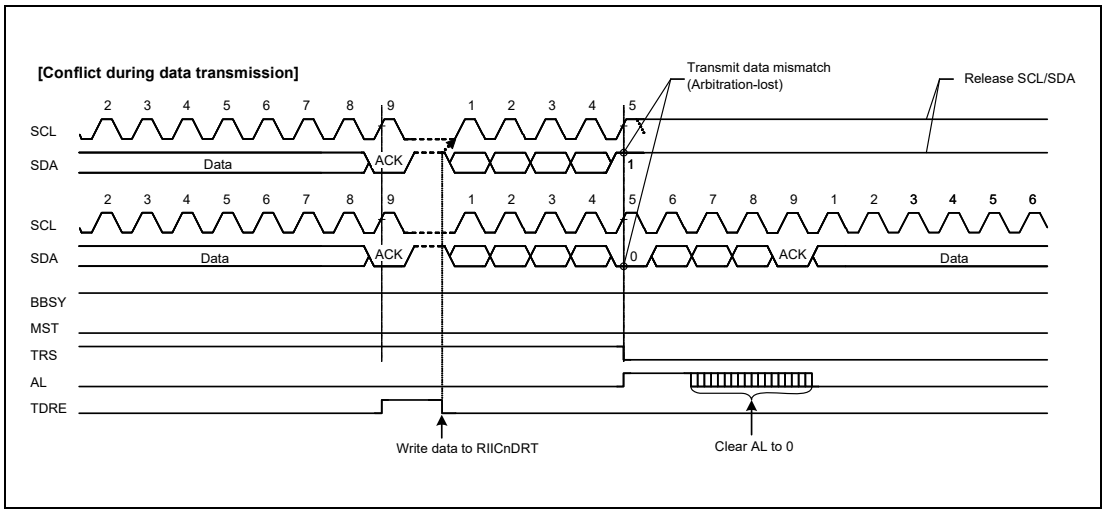


Figure 22.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

22.4.9 Start Condition/Restart Condition/Stop Condition Issuing Function

22.4.9.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

22.4.9.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made even during communication and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

(To detect the issuance of a restart condition, clear the RIICnSR2.START flag before a restart condition is issued.)

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

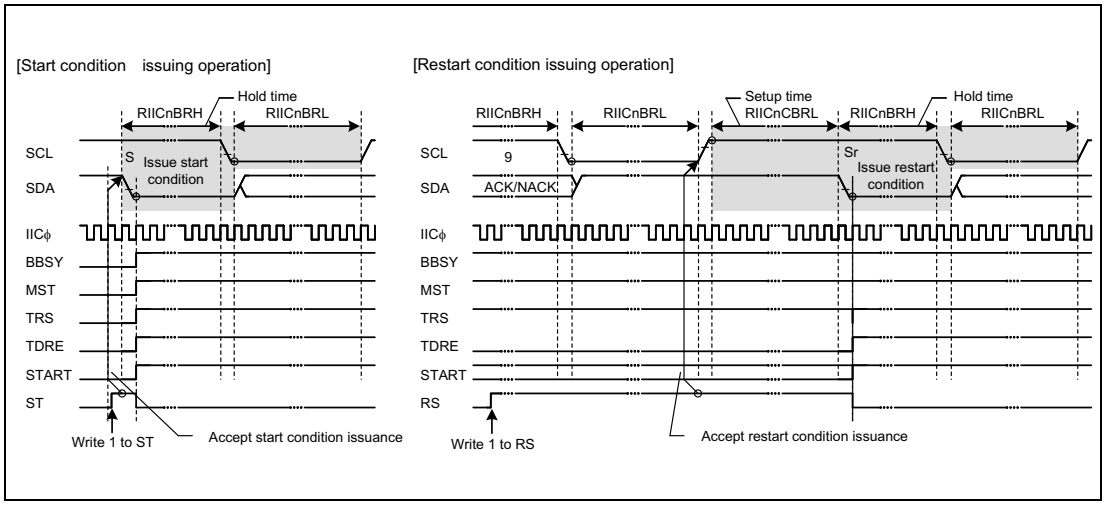


Figure 22.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

22.4.9.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

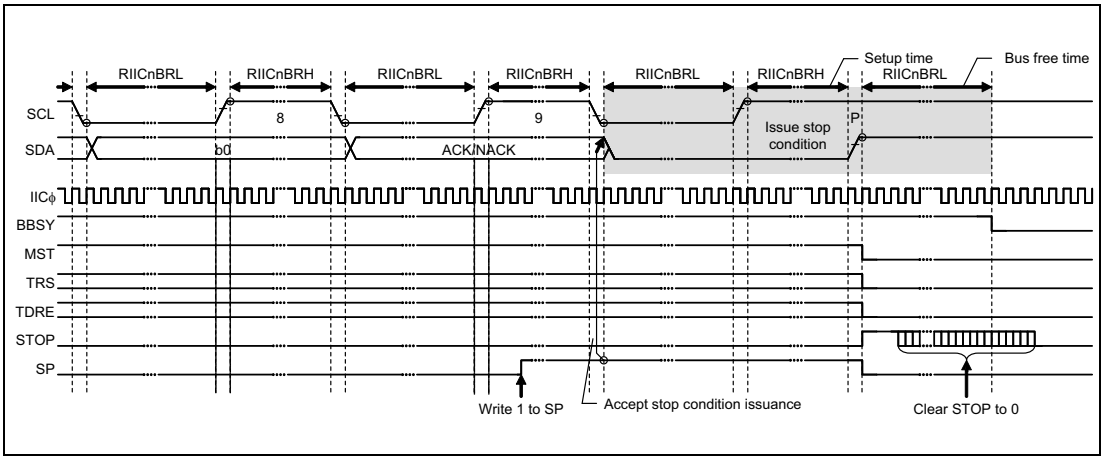


Figure 22.37 Stop Condition Issue Timing (SP Bit)

22.4.10 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

22.4.10.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

The internal counter is cleared when one of the conditions is met.

(1) When RIICnMR2.TMOH=0, and RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising

(2) When RIICnMR2.TMOH=1, and RIICnMR2.TMOL=0:
The internal counter is cleared by SCL falling

(3) When RIICnMR2.TMOH=RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising or falling

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY flag is 1) in master mode (RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (RIICnSR1 register is not 00_H) and the bus is busy (RIICnCR2.BBSY flag is 1) in slave mode (RIICnCR2.MST bit is 0).
- The bus is free (RIICnCR2.BBSY flag is 0) while generation of a START condition is requested (RIICnCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

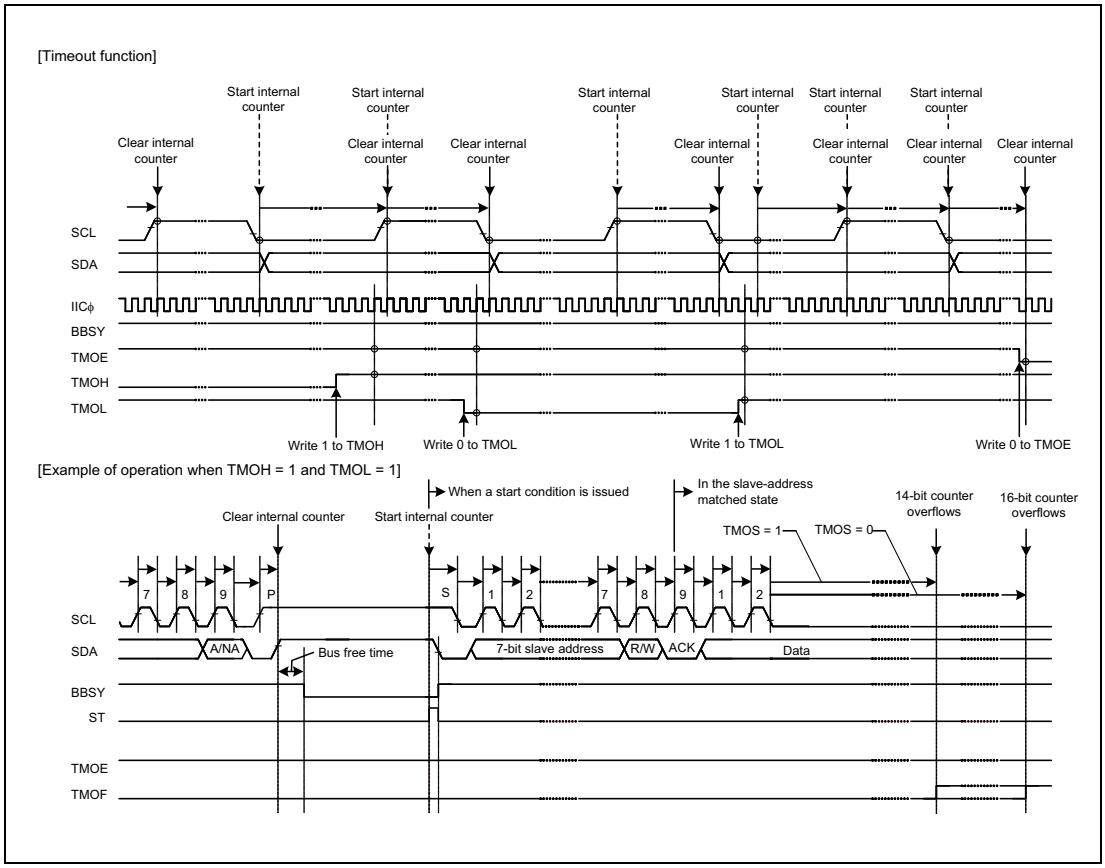


Figure 22.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

22.4.10.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the RIICnFER.MALE bit (master arbitration-lost detection disabled) cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration can be lost when the value of the RIICnCR1.SDAO bit does not match the state of the SDA line, so do not use this function.

[Additional output conditions for the SCL clock]

- In master mode and when the bus is free
- In master mode and the SCL line is not held low (the bus is busy)

Figure 22.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

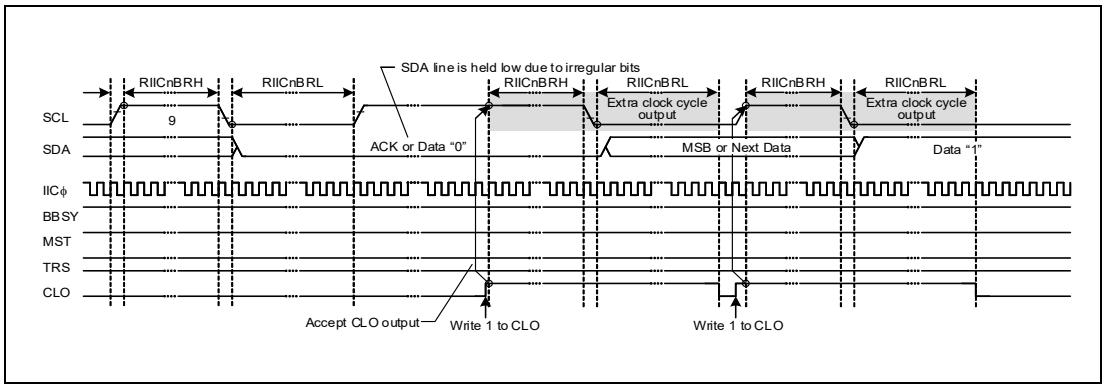


Figure 22.39 Extra SCL Clock Cycle Output Function (CLO Bit)

22.4.10.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01_B).

For a detailed description of the RIIC and internal resets, see **Section 22.4.11, Reset Function of RIIC**.

22.4.11 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES. **Table 22.26** lists the scope of each reset and reset conditions.

Table 22.26 RIIC Reset Functions (1/2)

UM		ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	Initialized	0	1	Retained	Retained
	IICRST	Initialized	1	1	Retained	Retained
	CLO	Initialized	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized	Initialized *1	Operation	Retained
	MST	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR1	MTWP	Initialized	Initialized	Retained	Retained	Retained
	CKS[2:0]	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Initialized	Retained	Initialized
RIICnSR2	TDRE	Initialized	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Initialized	Retained	Retained
RIICnSAR0, 1, 2		Initialized	Initialized	Retained	Retained	Retained

Table 22.26 RIIC Reset Functions (2/2)

UM	ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnBRH, RIICnBRL	Initialized	Initialized	Retained	Retained	Retained
RIICnDRT	Initialized	Initialized	Retained	Retained	Retained
RIICnDRR	Initialized	Initialized	Retained	Retained	Retained
RIICnDRS	Initialized	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

Section 23 CANFD Interface (RS-CANFD)

This section contains a generic description of the CANFD Interface (RS-CANFD).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CANFD.

23.1 Features RS-CANFD for RH850/U2A-EVA

23.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CANFD units.

Table 23.1 Number of Units and Channels

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)
Number of Channels	16 (w = 0 to 15)	16 (w = 0 to 15)	16 (w = 0 to 15)	16 (w = 0 to 15)	16 (w = 0 to 15)	16 (w = 0 to 15)	12 (w = 0 to 11)	11 (w = 0, 1, 3 to 11)	8 (w = 0, 1, 4, 6, 8 to 11)	7 (w = 0 to 2, 4, 6 to 8)
Number of TXMB (unit: message/ channel)	64	64	64	64	64	64	32	32	32	32
Number of AFL entry (unit: ID/ channel)	192	192	192	192	192	192	128	128	128	128
Number of RXMB (unit: message/ channel)	256	256	256	256	256	256	64	64	64	64
Name	RSCFDn									

The individual products have the CANFD Interface Channels listed below.

Table 23.2 Unit Configurations and Channels

Unit Name	Channel Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
RSCFD0	CAN0	√	√	√	√	√	√	√	√	√	√
	CAN1	√	√	√	√	√	√	√	√	√	√
	CAN2	√	√	√	√	√	√	√			√
	CAN3	√	√	√	√	√	√	√	√		
	CAN4	√	√	√	√	√	√	√	√	√	√
	CAN5	√	√	√	√	√	√	√	√		
	CAN6	√	√	√	√	√	√	√	√	√	√
	CAN7	√	√	√	√	√	√	√	√		√
RSCFD1	CAN8	√	√	√	√	√	√	√	√	√	√
	CAN9	√	√	√	√	√	√	√	√	√	
	CAN10	√	√	√	√	√	√	√	√	√	
	CAN11	√	√	√	√	√	√	√	√	√	
	CAN12	√	√	√	√	√	√				
	CAN13	√	√	√	√	√	√				
	CAN14	√	√	√	√	√	√				
	CAN15	√	√	√	√	√	√				

Table 23.3 Indices (1/2)

Index	Description
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index “n”; for example, RSCFDnCFDGCTR is the global control register of the RSCFDn unit.
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “m”; for example, RSCFDnCFDCmSTS is the channel m status register.
w	The individual channels of both RSCFD0 and RSCFD1 units are generically indicated by the index “w”; for example, CANwRX is receive data input.
j	The individual registers associated with receive rule table are generically indicated by the index “j”; for example, RSCFDnCFDGAFLIDj is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k”; for example, RSCFDnCFDCFCCk is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are generically identified by the index “x”; for example, RSCFDnCFDRFSTSx is the receive FIFO buffer status register.
d	Data field registers of receive buffers, transmit buffers,transmit/receive FIFO buffers, and receive FIFO buffers are identified by the index “d”. For example, the receive message buffer data field register is described as RSCFDnCFDRMDFd.
p	The individual transmit buffers are generically indicated by the index “p”; for example, RSCFDnCFDTMCp is the transmit buffer control register.
b	Message buffer component index of receive message buffer, receive FIFO, transmit/receive FIFO, transmit message buffer are identified by “b”. For example, the receive message buffer component register is described as RSCFDnCFDRMBCPbm.
q	Number of data byte of data field register are identified by “q”.
r	The individual RAM tests for CAN are generically indicated by the index “r”; for example, RSCFDnCFDRPGACCr is the RAM test page access register.
y	The transmit message buffer transmission are generically indicated by index “y”; for example, RSCFDnCFDTMTRSTSy is the transmit message buffer transmission request status register.
v	A global register which setting for 02 channels indicated by the index “v”; for example, RSCFDnCFDGAFLCFGv is the global AFL configuration register.
s	A global register which setting for 04 channels indicated by the index “s”; for example, RSCFDnCFDGTINTSTSs is the global transmit interrupt status register.

Table 23.3 Indices (2/2)

Index	Description
o	When a global register setting for 04 channels, the channel number indicated by the index “o”; for example, RSCFDnCFDGTINTSTSs.CFOTIFo is the Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (o + s * 4).
t	The receive message buffer new data are generically indicated by index "t"; for example, RSCFDnCFDRMNDt is the receive message buffer new data register.
u	The New Data Flag status for the corresponding RX Message Buffer is indicated by index "u"; for example RSCFDnCFDRMNDt.RMNSu is the the NewData Flag status for the corresponding RX Message Buffer.
c	The Message Buffer Component Register index is indicated by index "c"; for example, in register RSCFDnCFDRMBCPbm.Rc, "c" is RX Message Buffer Component Register index.

Note 1. The functions and descriptions of registers in this section are for the RS-CANFDs that has 8 channels (m = 0 to 7). When referring to information with indices, regard the index values as the ones corresponding to the target product.

Note 2. In some figure or table, this section use acronyms: CH, Ch, Ch instead of "channel".

The following table lists the values of indices for individual products.

Table 23.4 Indices for Individual Products

Index Correspondence of Each Product																
	RH850/ U2A-EVA (516 pins)	**RH850/ U2A16 (516 pins)**	**RH850/ U2A16 (373 pins)**	**RH850/ U2A16 (292 pins)**	**RH850/ U2A8 (373 pins)**	**RH850/ U2A8 (292 pins)**	**RH850/ U2A6 (292 pins)**		**RH850/ U2A6 (176 pins)**		**RH850/ U2A6 (156 pins)**		**RH850/ U2A6 (144 pins)**	
n	n = 0, 1						n = 0	n = 1	n = 0	n = 1	n = 0	n = 1	n = 0	n = 1
m	m = 0 to 7						m = 0 to 7	m = 0 to 3	m = 0, 1, 3 to 7	m = 0 to 3	m = 0, 1, 4, 6	m = 0 to 3	m = 0 to 2, 4, 6, 7	m = 0
w	w = 0 to 15						w = 0 to 7	w = 8 to 11	w = 0, 1, 3 to 7	w = 8 to 11	w = 0, 1, 4, 6	w = 8 to 11	w = 0 to 2, 4, 6, 7	w = 8
j	j = 0 to 15						j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15
k	k = 0 to 23						k = 0 to 23	k = 0 to 11	k = 0 to 5, 9 to 23	k = 0 to 11	k = 0 to 5, 12 to 14, 18 to 20	k = 0 to 11	k = 0 to 8, 12 to 14, 18 to 23	k = 0 to 2
x	x = 0 to 7						x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7
d	d = 0 to 15						d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15	d = 0 to 15
p	p = 0 to 511						p = 64*m to 64*m + 15, 64*m + 32 to 64*m + 47							
b	Refer to **Table 23.106, Message Buffer Component Register Start Addresses**													
q	q = 4						q = 4	q = 4	q = 4	q = 4	q = 4	q = 4	q = 4	q = 4
r	r = 0 to 63						r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63
y	y = 0 to 15						y = 0 to 15	y = 0 to 7	y = 0, 1, 2, 3, 6 to 15	y = 0 to 7	y = 0 to 3, 8, 9, 12, 13	y = 0 to 7	y = 0 to 5, 8, 9, 12 to 15	y = 0, 1
v	v = 0 to 3						v = 0 to 3	v = 0 to 1	v = 0 to 3	v = 0 to 1	v = 0, 2, 3	v = 0 to 1	v = 0 to 3	v = 0
s	s = 0 to 1						s = 0 to 1	s = 0	s = 0 to 1	s = 0	s = 0 to 1	s = 0	s = 0 to 1	s = 0
o	o = 0 to 3						o = 0 to 3	o = 0 to 3	o = 0 to 3	o = 0 to 3	o = 0 to 2	o = 0 to 3	o = 0 to 3	o = 0
t	t = 0 to 3						t = 0 to 3	t = 0 to 1	t = 0 to 3	t = 0 to 1	t = 0, 2, 3	t = 0 to 1	t = 0 to 3	t = 0
u	u = 0 to 127						u = 0 to 127	u = 0 to 63	u = 0 to 31, 48 to 127	u = 0 to 63	u = 0 to 31, 64 to 79, 96 to 111	u = 0 to 63	u = 0 to 47, 64 to 79, 96 to 127	u = 0 to 15
c	c = 0 to 18						c = 0 to 18	c = 0 to 18	c = 0 to 18	c = 0 to 18	c = 0 to 18	c = 0 to 18	c = 0 to 18	c = 0 to 18

23.1.2 Register Base Addresses

RSCFDn base addresses are listed in the following table.

RSCFDn register addresses are given as offsets from the base addresses in general.

Table 23.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RSCFD0_base>	FFF5 0000_H	P-Bus Group 8
<RSCFD1_base>	FFD0 0000_H	P-Bus Group 3

23.1.3 Clock Supply

The RSCFDn clock supply is shown in the following table.

Table 23.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RSCFDn	clk_xincan	RS-CANFD clock CLK_RCANOSC
	clkc	High-speed peripheral clock CLK_HSB
	pclk	High-speed peripheral clock CLK_HSB
	ramclk	Peripheral ultra high speed clock CLK_UHSB

The operating frequency of the RSCFDn depends on the transfer rate and the number of channels in use. **Table 23.7** shows the range of the frequency.

Table 23.7 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use

Condition		Range of Operating Frequency			
Transfer Rate	No. of Channels in Use	pclk	clk_xincan*1	clkc*1	ramclk
(Transfer Rate) ≤ 2Mbps	Max. 8 ch	80 MHz (fixed)	Max. 40 MHz	Max. 80 MHz	160 MHz (fixed)
2Mbps<(Transfer Rate) ≤ 5Mbps	Max. 8 ch	80 MHz (fixed)	(do not select)	Max. 80 MHz	160 MHz (fixed)
5Mbps<(Transfer Rate) ≤ 8Mbps	Max. 6 ch	80 MHz (fixed)	(do not select)	80 MHz	160 MHz (fixed)

Note 1. Setting the DCS bit in the RSCFDnCFDGCFG register enables to select either clk_xincan or clkc.

23.1.4 Interrupt Requests and Error Notifications

The RSCFDn interrupt requests are listed in the following table.

Table 23.8 Interrupt Requests (1/2)

Unit Interrupt Signal		Description	Interrupt Number
RSCFD0			
INTRCANGERR0		CAN global error interrupt	296
INTRCANGRECC0		CAN receive FIFO interrupt	297
CAN0	INTRCAN0ERR	CAN0 error interrupt	298
	INTRCAN0REC	CAN0 transmit/receive FIFO receive completion interrupt	299
	INTRCAN0TRX	CAN0 transmit interrupt	300
CAN1	INTRCAN1ERR	CAN1 error interrupt	301
	INTRCAN1REC	CAN1 transmit/receive FIFO receive completion interrupt	302
	INTRCAN1TRX	CAN1 transmit interrupt	303
CAN2	INTRCAN2ERR	CAN2 error interrupt	304
	INTRCAN2REC	CAN2 transmit/receive FIFO receive completion interrupt	305
	INTRCAN2TRX	CAN2 transmit interrupt	306
CAN3	INTRCAN3ERR	CAN3 error interrupt	307
	INTRCAN3REC	CAN3 transmit/receive FIFO receive completion interrupt	308
	INTRCAN3TRX	CAN3 transmit interrupt	309
CAN4	INTRCAN4ERR	CAN4 error interrupt	310
	INTRCAN4REC	CAN4 transmit/receive FIFO receive completion interrupt	311
	INTRCAN4TRX	CAN4 transmit interrupt	312
CAN5	INTRCAN5ERR	CAN5 error interrupt	313
	INTRCAN5REC	CAN5 transmit/receive FIFO receive completion interrupt	314
	INTRCAN5TRX	CAN5 transmit interrupt	315
CAN6	INTRCAN6ERR	CAN6 error interrupt	316
	INTRCAN6REC	CAN6 transmit/receive FIFO receive completion interrupt	317
	INTRCAN6TRX	CAN6 transmit interrupt	318
CAN7	INTRCAN7ERR	CAN7 error interrupt	319
	INTRCAN7REC	CAN7 transmit/receive FIFO receive completion interrupt	320
	INTRCAN7TRX	CAN7 transmit interrupt	321
RSCFD1			
INTRCANGERR1		CAN global error interrupt	322
INTRCANGRECC1		CAN receive FIFO interrupt	323
CAN8	INTRCAN8ERR	CAN8 error interrupt	324
	INTRCAN8REC	CAN8 transmit/receive FIFO receive completion interrupt	325
	INTRCAN8TRX	CAN8 transmit interrupt	326
CAN9	INTRCAN9ERR	CAN9 error interrupt	327
	INTRCAN9REC	CAN9 transmit/receive FIFO receive completion interrupt	328
	INTRCAN9TRX	CAN9 transmit interrupt	329
CAN10	INTRCAN10ERR	CAN10 error interrupt	330
	INTRCAN10REC	CAN10 transmit/receive FIFO receive completion interrupt	331
	INTRCAN10TRX	CAN10 transmit interrupt	332

Table 23.8 Interrupt Requests (2/2)

Unit Interrupt Signal		Description	Interrupt Number
CAN11	INTRCAN11ERR	CAN11 error interrupt	333
	INTRCAN11REC	CAN11 transmit/receive FIFO receive completion interrupt	334
	INTRCAN11TRX	CAN11 transmit interrupt	335
CAN12	INTRCAN12ERR	CAN12 error interrupt	336
	INTRCAN12REC	CAN12 transmit/receive FIFO receive completion interrupt	337
	INTRCAN12TRX	CAN12 transmit interrupt	338
CAN13	INTRCAN13ERR	CAN13 error interrupt	339
	INTRCAN13REC	CAN13 transmit/receive FIFO receive completion interrupt	340
	INTRCAN13TRX	CAN13 transmit interrupt	341
CAN14	INTRCAN14ERR	CAN14 error interrupt	342
	INTRCAN14REC	CAN14 transmit/receive FIFO receive completion interrupt	343
	INTRCAN14TRX	CAN14 transmit interrupt	344
CAN15	INTRCAN15ERR	CAN15 error interrupt	345
	INTRCAN15REC	CAN15 transmit/receive FIFO receive completion interrupt	346
	INTRCAN15TRX	CAN15 transmit interrupt	347

Table 23.9 sDMA/DTS Requests (1/2)

Interrupt symbol name	Unit Interrupt Signal	Description	sDMA Trigger Number	DTS Trigger Number
INTRCANRFDREQ15	can_rf_dmareq[7]	Receive FIFO DMA request 15	Group0 - 60	Group0 - 49
INTRCANRFDREQ14	can_rf_dmareq[6]	Receive FIFO DMA request 14	Group0 - 59	Group0 - 48
INTRCANRFDREQ13	can_rf_dmareq[5]	Receive FIFO DMA request 13	Group0 - 58	Group0 - 47
INTRCANRFDREQ12	can_rf_dmareq[4]	Receive FIFO DMA request 12	Group0 - 57	Group0 - 46
INTRCANRFDREQ11	can_rf_dmareq[3]	Receive FIFO DMA request 11	Group0 - 56	Group0 - 45
INTRCANRFDREQ10	can_rf_dmareq[2]	Receive FIFO DMA request 10	Group0 - 55	Group0 - 44
INTRCANRFDREQ9	can_rf_dmareq[1]	Receive FIFO DMA request 9	Group0 - 54	Group0 - 43
INTRCANRFDREQ8	can_rf_dmareq[0]	Receive FIFO DMA request 8	Group0 - 53	Group0 - 42
INTRCANRFDREQ7	can_rf_dmareq[7]	Receive FIFO DMA request 7	Group0 - 52	Group0 - 41
INTRCANRFDREQ6	can_rf_dmareq[6]	Receive FIFO DMA request 6	Group0 - 51	Group0 - 40
INTRCANRFDREQ5	can_rf_dmareq[5]	Receive FIFO DMA request 5	Group0 - 50	Group0 - 39
INTRCANRFDREQ4	can_rf_dmareq[4]	Receive FIFO DMA request 4	Group0 - 49	Group0 - 38
INTRCANRFDREQ3	can_rf_dmareq[3]	Receive FIFO DMA request 3	Group0 - 48	Group0 - 37
INTRCANRFDREQ2	can_rf_dmareq[2]	Receive FIFO DMA request 2	Group0 - 47	Group0 - 36
INTRCANRFDREQ1	can_rf_dmareq[1]	Receive FIFO DMA request 1	Group0 - 46	Group0 - 35
INTRCANRFDREQ0	can_rf_dmareq[0]	Receive FIFO DMA request 0	Group0 - 45	Group0 - 34
INTRCANCFDREQ15	can_cf_dmareq[7]	Common FIFO DMA request 15	Group0 - 44	Group0 - 33
INTRCANCFDREQ14	can_cf_dmareq[6]	Common FIFO DMA request 14	Group0 - 43	Group0 - 32
INTRCANCFDREQ13	can_cf_dmareq[5]	Common FIFO DMA request 13	Group0 - 42	Group0 - 31
INTRCANCFDREQ12	can_cf_dmareq[4]	Common FIFO DMA request 12	Group0 - 41	Group0 - 30
INTRCANCFDREQ11	can_cf_dmareq[3]	Common FIFO DMA request 11	Group0 - 40	Group0 - 29
INTRCANCFDREQ10	can_cf_dmareq[2]	Common FIFO DMA request 10	Group0 - 39	Group0 - 28
INTRCANCFDREQ9	can_cf_dmareq[1]	Common FIFO DMA request 9	Group0 - 38	Group0 - 27
INTRCANCFDREQ8	can_cf_dmareq[0]	Common FIFO DMA request 8	Group0 - 37	Group0 - 26

Table 23.9 sDMA/DTS Requests (2/2)

Interrupt symbol name	Unit Interrupt Signal	Description	sDMA Trigger Number	DTS Trigger Number
INTRCANCFDREQ7	can_cf_dmareq[7]	Common FIFO DMA request 7	Group0 - 36	Group0 - 25
INTRCANCFDREQ6	can_cf_dmareq[6]	Common FIFO DMA request 6	Group0 - 35	Group0 - 24
INTRCANCFDREQ5	can_cf_dmareq[5]	Common FIFO DMA request 5	Group0 - 34	Group0 - 23
INTRCANCFDREQ4	can_cf_dmareq[4]	Common FIFO DMA request 4	Group0 - 33	Group0 - 22
INTRCANCFDREQ3	can_cf_dmareq[3]	Common FIFO DMA request 3	Group0 - 32	Group0 - 21
INTRCANCFDREQ2	can_cf_dmareq[2]	Common FIFO DMA request 2	Group0 - 31	Group0 - 20
INTRCANCFDREQ1	can_cf_dmareq[1]	Common FIFO DMA request 1	Group0 - 30	Group0 - 19
INTRCANCFDREQ0	can_cf_dmareq[0]	Common FIFO DMA request 0	Group0 - 29	Group0 - 18

NOTE

For selecting the DMA requests, see **Section 7.1.5, sDMAC Transfer Requests**.

For selecting the DTS requests, see **Section 8.1.5, DTS Transfer Requests**.

This module has no error notifications.

23.1.5 Reset Sources

Table 23.10 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RSCFDn	All registers	√	√	√	√	√	√	—

23.1.6 External Input/Output Signals

External input/output signals of RSCFDn are listed below.

Table 23.11 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RSCFD0		
CANwRX (w = 0 to 7)	CANw receive data input	CANwRX (w = 0 to 7)
CANwTX (w = 0 to 7)	CANw transmit data output	CANwTX (w = 0 to 7)
RSCFD1		
CANwRX (w = 8 to 15)	CANw receive data input	CANwRX (w = 8 to 15)
CANwTX (w = 8 to 15)	CANw transmit data output	CANwTX (w = 8 to 15)

23.2 Overview

23.2.1 Functional Overview

Table 23.12 shows the RS-CANFD module specifications. **Figure 23.1** shows the RS-CANFD module block diagram.

Table 23.12 RS-CANFD Module Specifications (1/3)

Item	Specification
Protocol	ISO 11898-1 (2015) compliant Using frames is selectable by switching interface modes.
Communication speed	Classical CAN mode: • Maximum 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ CANm bit time = CANmTq × Tq count per bit $\text{CANmTq} = \frac{\text{(NBRP[9:0] bits in the RSCFDnCFDCmNCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCFDnCFDGCFG register) CANFD mode: • Nominal bit rate: max.1 Mbps, data bit rate: max. 8 Mbps $\text{Transmission rate (CANm nominal bit time clock)} = \frac{1}{\text{CANm nominal bit time}}$ $\text{Transmission rate (CANm data bit time clock)} = \frac{1}{\text{CANm data bit time}}$ CANm nominal bit time = CANmTq(N) × Tq count per nominal bit CANm data bit time = CANmTq(D) × Tq count per data bit $\text{CANmTq(N)} = \frac{\text{(NBRP[9:0] bits in the RSCFDnCFDCmNCFG register + 1)}}{\text{fCAN}}$ $\text{CANmTq(D)} = \frac{\text{(DBRP[7:0] bits in the RSCFDnCFDCmDCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCFDnCFDGCFG register) n = 0, 1 m = 0 to 7 Tq: Time quantum
Buffer	(Number of TXMB + Number of RXMB) x (Number of Channel) in total when 64 byte data payload. • Individual buffers: (Number of TXMB) x (Number of Channel) buffers. Transmit buffer: (Number of TXMB) buffers per channel. Transmit queue: Four queue per channel (shared with the transmit buffer; up to (Number of TXMB) buffers allocatable) • Shared buffers: (Number of RXMB) x (Number of Channel), shared among all the CAN channels in a unit. Receive buffer: 0 to 16 buffers for one channel Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/Receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) • ECC included
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (reception of message transmitted from message the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)

Table 23.12 RS-CANFD Module Specifications (2/3)

Item	Specification
Reception filter function	• Selects receive messages according to: (Number of AFL entry) x (Number of Channel) receive rules. • Sets the number of receive rules (U2A-EVA, U2A16, U2A8: maximum is 384; U2A6: maximum is 255) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Enables DLC filter check for each acceptance rule.
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, and/or transmit queue. • Label addition function (Pointer / Pointer field) Stores label information together with a message in a receive buffer, FIFO buffer, and transmit queue.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer, transmit/receive FIFO buffer, and transmit queue. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages. Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information.
Gateway function	Transmits a received message automatically.
Bus off recovery mode selection	Selects the method for returning from bus off state. • ISO 11898-1 (2015) compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from bus-off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	[Number of units] x 2 + [Number of channels] x 3 sources in total • Global Interrupts (4 sources) – Receive FIFO interrupt for unit 0/1 – Global error interrupt for unit 0/1 • Channel interrupts (3 sources/channel) – CANm transmit interrupt – CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) – CANm error interrupt

Table 23.12 RS-CANFD Module Specifications (3/3)

Item	Specification
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CANFD module.
CAN clock source	Selects the clkc or the clk_xincan. As for the range of operating frequency, see **Table 23.7**.
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Internal CAN Bus Communication Test Mode [CRC error test enabled]

23.2.2 Interface Modes

The RS-CANFD has two interface modes.

- Classical CAN mode: Handles only classical CAN frames.
- CANFD mode: Handles classical CAN frames and CANFD frames.

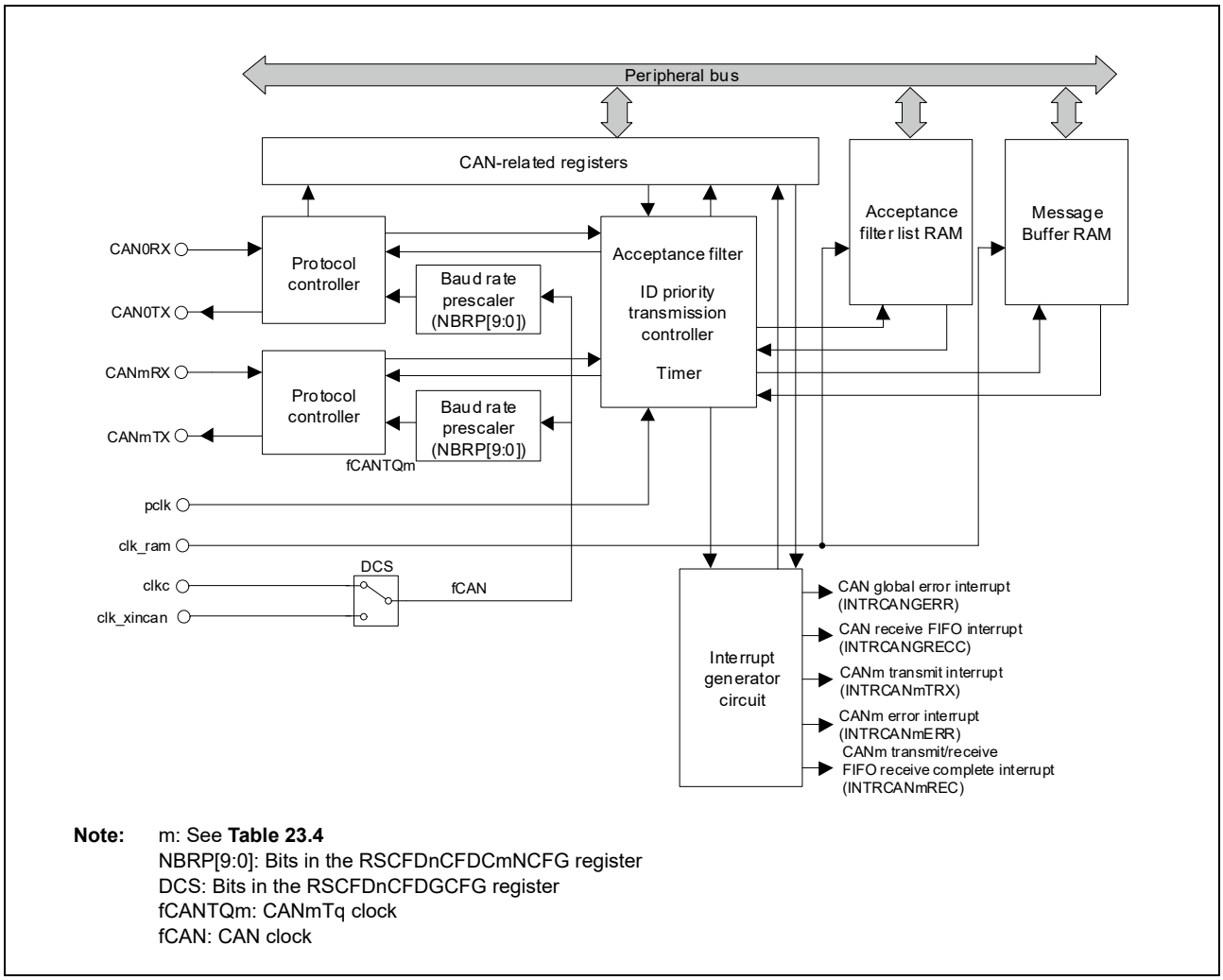
Interface modes are switched by the FDOE bit and CLOE bit in the RSCFDnCFDCmFDCFG register.

23.2.3 CANFD Protocol

This product supports CANFD according to the ISO 11898-1 (2015) protocol that specifies the new CRC field including stuff counters.

23.2.4 Block Diagram

In CANFD mode, different clock signals are input to the baud rate prescaler and the protocol controller respectively. See **Section 23.5.1.3, Baud Rate**.



Note: m: See **Table 23.4**
NBRP[9:0]: Bits in the RSCFDnCFDCmNCFG register
DCS: Bits in the RSCFDnCFDGCFG register
fCANTQm: CANmTq clock
fCAN: CAN clock

Figure 23.1 RS-CANFD Module Block Diagram

Tx/Rx:

Input/Output pins of the CAN module

Protocol controller:

Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling, etc.

Acceptance filter list RAM:

This RAM is used to store the message acceptance filtering entries for all channels. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer. The AFLRAM is divided in two parts to accelerate the AFL access process.

Message Buffer RAM:

This RAM is used to store messages after reception or for transmission using a normal Message Buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.

Acceptance filter:

Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.

Timer:

Two timers

- Reception Timestamp function
- Transmission separation time for FIFO Buffers

Interrupt generator:

Generates several types of global and channel interrupts

CAN SFRs:

Register associated with CAN. Refer to **Section 23.3**.

23.3 Registers

This section describes all registers of RS-CANFD.

23.3.1 List of Registers

The 'Value after Reset' shown in the table for the RAM area, consisting of RSCFDnCFDGAFLIDj, RSCFDnCFDGAFLMj, RSCFDnCFDGAFLP0j, RSCFDnCFDGAFLP1j, RSCFDnCFDRMBCPb, RSCFDnCFDRFMBCPb, RSCFDnCFDCFMBCPb, RSCFDnCFDTMBCPb, RSCFDnCFDTHLACC0m, RSCFDnCFDTHLACC1m and RSCFDnCFDRPGACCr is True, after, initialization after HW Reset. Refer to **Section 23.5.2, CAN Module Configuration after H/W Reset** for details of the initialization process.

Also for the RAM area, if a write access with a size of 8 or 16 bits is done then the RS-CANFD module does a read modify write access to the RAM location, because the RAM requires a 32-bit access via the ECC module.

In case of single bit error the correct data will be written back.

In case of multiple bit errors unknown data will be written back.

Note: The RAM area is initialized after HW reset, refer **Section 23.5.2, CAN Module Configuration after H/W Reset**.

Users should not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

RSCFD registers are listed in the table below.
For details about <RSCFDn_base>, see **Section 23.1.2, Register Base Addresses**.

Table 23.13 Registers (1/4)

Register Name	Symbol	Address	Access Size	Access Protection	
				PBG	Other
Channel m Nominal Bit Rate Configuration Register	RSCFDnCFDCmNCFG	<RSCFDn_base> + 0000_H + m×0010_H	8, 16, 32	*1	—
Channel m Control Register	RSCFDnCFDCmCTR	<RSCFDn_base> + 0004_H + m×0010_H	8, 16, 32	*1	—
Channel m Status Register	RSCFDnCFDCmSTS	<RSCFDn_base> + 0008_H + m×0010_H	8, 16, 32	*1	—
Channel m Error Flag Register	RSCFDnCFDCmERFL	<RSCFDn_base> + $000C_H$ + m×0010_H	8, 16, 32	*1	—
Global Configuration Register	RSCFDnCFDGCFG	<RSCFDn_base> + 0084_H	8, 16, 32	*2	—
Global Control Register	RSCFDnCFDGCTR	<RSCFDn_base> + 0088_H	8, 16, 32	*2	—
Global Status Register	RSCFDnCFDGSTS	<RSCFDn_base> + $008C_H$	8, 16, 32	*2	—
Global Error Flag Register	RSCFDnCFDGERFL	<RSCFDn_base> + 0090_H	8, 16, 32	*2	—
Global Timestamp Counter Register	RSCFDnCFDGTSC	<RSCFDn_base> + 0094_H	16, 32	*2	—
Global Acceptance Filter List Entry Control Register	RSCFDnCFDGAFLECTR	<RSCFDn_base> + 0098_H	8, 16, 32	*2	—
Global Acceptance Filter List Configuration Register v	RSCFDnCFDGAFLCFGv	<RSCFDn_base> + $009C_H$ + v×0004_H	8, 16, 32	*2	—
Receive Message Buffer Number Register	RSCFDnCFDRMNB	<RSCFDn_base> + $00AC_H$	8, 16, 32	*2	—
Receive Message Buffer New Data Register t	RSCFDnCFDRMNDt	<RSCFDn_base> + $00B0_H$ + t×0004_H	8, 16, 32	*2	—
Receive FIFO Configuration / Control Register x	RSCFDnCFDRFCCx	<RSCFDn_base> + $00C0_H$ + x×0004_H	8, 16, 32	*2	—
Receive FIFO Status Register x	RSCFDnCFDRFSTSx	<RSCFDn_base> + $00E0_H$ + x×0004_H	8, 16, 32	*2	—
Receive FIFO Pointer Control Register x	RSCFDnCFDRFPCTRx	<RSCFDn_base> + 0100_H + x×0004_H	8, 16, 32	*2	—
Transmit/Receive FIFO Configuration / Control Register k	RSCFDnCFDCFCCk	<RSCFDn_base> + 0120_H + k×0004_H	8, 16, 32	*1	—
Transmit/Receive FIFO Configuration / Control Enhancement Register k	RSCFDnCFDCFCCEk	<RSCFDn_base> + 0180_H + k×0004_H	8, 16, 32	*1	—
Transmit/Receive FIFO Status Register k	RSCFDnCFDCFSTSk	<RSCFDn_base> + $01E0_H$ + k×0004_H	8, 16, 32	*1	—

Table 23.13 Registers (2/4)

Register Name	Symbol	Address	Access Size	Access Protection	
				PBG	Other
Transmit/Receive FIFO Pointer Control Register k	RSCFDnCFDCFPCTRk	<RSCFDn_base> + 0240_H + k×0004_H	8, 16, 32	*1	—
FIFO Empty Status Register	RSCFDnCFDFESTS	<RSCFDn_base> + $02A0_H$	8, 16, 32	*2	—
FIFO Full Status Register	RSCFDnCFDFFSTS	<RSCFDn_base> + $02A4_H$	8, 16, 32	*2	—
FIFO Message Lost Status Register	RSCFDnCFDFMSTS	<RSCFDn_base> + $02A8_H$	8, 16, 32	*2	—
Receive FIFO Interrupt Flag Status Register	RSCFDnCFDRFISTS	<RSCFDn_base> + $02AC_H$	8, 16, 32	*2	—
Transmit/Receive FIFO Receive Interrupt Flag Status Register	RSCFDnCFDCFRISTS	<RSCFDn_base> + $02B0_H$	8, 16, 32	*2	—
Transmit/Receive FIFO Transmit Interrupt Flag Status Register	RSCFDnCFDCFTISTS	<RSCFDn_base> + $02B4_H$	8, 16, 32	*2	—
Transmit/Receive FIFO One Frame Receive Interrupt Flag Status Register	RSCFDnCFDCFOFRISTS	<RSCFDn_base> + $02B8_H$	8, 16, 32	*2	—
Transmit/Receive FIFO One Frame Transmit Interrupt Flag Status Register	RSCFDnCFDCFOFTISTS	<RSCFDn_base> + $02BC_H$	8, 16, 32	*2	—
Transmit/Receive FIFO Message Over Write Status Register	RSCFDnCFDCFMOWSTS	<RSCFDn_base> + $02C0_H$	8, 16, 32	*2	—
FIFO FDC Level Full Status Register	RSCFDnCFDFFFSTS	<RSCFDn_base> + $02C4_H$	8, 16, 32	*2	—
Transmit Message Buffer Control Register p	RSCFDnCFDTMCp	<RSCFDn_base> + $02D0_H$ + p×0001_H	8	*1	—
Transmit Message Buffer Status Register p	RSCFDnCFDTMSTSp	<RSCFDn_base> + $07D0_H$ + p×0001_H	8	*1	—
Transmit Message Buffer Transmission Request Status Register y	RSCFDnCFDTMTRSTSy	<RSCFDn_base> + $0CD0_H$ + y×0004_H	8, 16, 32	*1	—
Transmit Message Buffer Transmission Abort Request Status Register y	RSCFDnCFDTMTARSTSy	<RSCFDn_base> + $0D70_H$ + y×0004_H	8, 16, 32	*1	—
Transmit Message Buffer Transmission Completion Status Register y	RSCFDnCFDTMTCSTSy	<RSCFDn_base> + $0E10_H$ + y×0004_H	8, 16, 32	*1	—
Transmit Message Buffer Transmission Abort Status Register y	RSCFDnCFDTMTASTSy	<RSCFDn_base> + $0EB0_H$ + y×0004_H	8, 16, 32	*1	—
Transmit Message Buffer Interrupt Enable Configuration Register y	RSCFDnCFDTMIECy	<RSCFDn_base> + $0F50_H$ + y×0004_H	8, 16, 32	*1	—
Transmit Queue Configuration / Control Register 0 m	RSCFDnCFDTXQCC0m	<RSCFDn_base> + 1000_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Status Register 0 m	RSCFDnCFDTXQSTS0m	<RSCFDn_base> + 1020_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Pointer Control Register 0 m	RSCFDnCFDTXQPCTR0m	<RSCFDn_base> + 1040_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Configuration / Control Register 1 m	RSCFDnCFDTXQCC1m	<RSCFDn_base> + 1060_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Status Register 1 m	RSCFDnCFDTXQSTS1m	<RSCFDn_base> + 1080_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Pointer Control Register 1 m	RSCFDnCFDTXQPCTR1m	<RSCFDn_base> + $10A0_H$ + m×0004_H	8, 16, 32	*1	—
Transmit Queue Configuration / Control Register 2 m	RSCFDnCFDTXQCC2m	<RSCFDn_base> + $10C0_H$ + m×0004_H	8, 16, 32	*1	—
Transmit Queue Status Register 2 m	RSCFDnCFDTXQSTS2m	<RSCFDn_base> + $10E0_H$ + m×0004_H	8, 16, 32	*1	—
Transmit Queue Pointer Control Register 2 m	RSCFDnCFDTXQPCTR2m	<RSCFDn_base> + 1100_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Configuration / Control Register 3 m	RSCFDnCFDTXQCC3m	<RSCFDn_base> + 1120_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Status Register 3 m	RSCFDnCFDTXQSTS3m	<RSCFDn_base> + 1140_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Pointer Control Register 3 m	RSCFDnCFDTXQPCTR3m	<RSCFDn_base> + 1160_H + m×0004_H	8, 16, 32	*1	—
Transmit Queue Empty Status Register	RSCFDnCFDTXQESTS	<RSCFDn_base> + 1180_H	8, 16, 32	*2	—
Transmit Queue Full Interrupt Status Register	RSCFDnCFDTXQFISTS	<RSCFDn_base> + 1184_H	8, 16, 32	*2	—
Transmit Queue Message Lost Status Register	RSCFDnCFDTXQMSTS	<RSCFDn_base> + 1188_H	8, 16, 32	*2	—
Transmit Queue Message Overwrite Status Register	RSCFDnCFDTXQOWSTS	<RSCFDn_base> + $118C_H$	8, 16, 32	*2	—
Transmit Queue Interrupt Status Register	RSCFDnCFDTXQISTS	<RSCFDn_base> + 1190_H	8, 16, 32	*2	—
Transmit Queue One Frame Transmit Interrupt Status Register	RSCFDnCFDTXQOFTISTS	<RSCFDn_base> + 1194_H	8, 16, 32	*2	—

Table 23.13 Registers (3/4)

Register Name	Symbol	Address	Access Size	Access Protection	
				PBG	Other
Transmit Queue One Frame Receive Interrupt Status Register	RSCFDnCFDTXQOFRISTS	<RSCFDn_base> + 1198_H	8, 16, 32	*2	—
Transmit Queue Full Status Register	RSCFDnCFDTXQFSTS	<RSCFDn_base> + $119C_H$	8, 16, 32	*2	—
Transmit History List Configuration / Control Register m	RSCFDnCFDTHLCCm	<RSCFDn_base> + 1200_H + m×0004_H	8, 16, 32	*1	—
Transmit History List Status Register m	RSCFDnCFDTHLSTSm	<RSCFDn_base> + 1220_H + m×0004_H	8, 16, 32	*1	—
Transmit History List Pointer Control Register m	RSCFDnCFDTHLPCTRm	<RSCFDn_base> + 1240_H + m×0004_H	8, 16, 32	*1	—
Global Transmit Interrupt Status Register s	RSCFDnCFDGTINTSTSs	<RSCFDn_base> + 1300_H + s×0004_H	8, 16, 32	*2	—
Global Test Configuration Register	RSCFDnCFDGTSTCFG	<RSCFDn_base> + 1308_H	8, 16, 32	*2	—
Global Test Control Register	RSCFDnCFDGTSTCTR	<RSCFDn_base> + $130C_H$	8, 16, 32	*2	—
Global FD Configuration register	RSCFDnCFDGFDCFG	<RSCFDn_base> + 1314_H	8, 16, 32	*2	—
Global Lock Key Register	RSCFDnCFDGLOCKK	<RSCFDn_base> + $131C_H$	16, 32	*2	—
DMA Transfer Control Register	RSCFDnCFDCDTCT	<RSCFDn_base> + 1330_H	8, 16, 32	*2	—
DMA Transfer Status Register	RSCFDnCFDCDTSTS	<RSCFDn_base> + 1334_H	8, 16, 32	*2	—
DMA Transmit Transfer Control Register	RSCFDnCFDCDTTCT	<RSCFDn_base> + 1340_H	8, 16, 32	*2	—
DMA Transmit Transfer Status Register	RSCFDnCFDCDTTSTS	<RSCFDn_base> + 1344_H	8, 16, 32	*2	—
Global Receive Interrupt Status Register m	RSCFDnCFDGRINTSTSm	<RSCFDn_base> + 1350_H + m×0004_H	8, 16, 32	*1	—
Global Reset Control Register	RSCFDnCFDGRSTC	<RSCFDn_base> + 1380_H	16, 32	*2	—
Global Flexible CAN mode Configuration Register	RSCFDnCFDGFCMC	<RSCFDn_base> + 1384_H	8, 16, 32	*2	—
Global Flexible transmission buffer assignment Configuration Register	RSCFDnCFDGFTBAC	<RSCFDn_base> + $138C_H$	8, 16, 32	*2	—
Channel m Data Bit Rate Configuration Register	RSCFDnCFDCmDCFG	<RSCFDn_base> + 1400_H + m×0020_H	8, 16, 32	*1	—
Channel m CANFD Configuration Register	RSCFDnCFDCmFDCFG	<RSCFDn_base> + 1404_H + m×0020_H	8, 16, 32	*1	—
Channel m CANFD Control Register	RSCFDnCFDCmFDCTR	<RSCFDn_base> + 1408_H + m×0020_H	8, 16, 32	*1	—
Channel m CANFD Status Register	RSCFDnCFDCmFDSTS	<RSCFDn_base> + $140C_H$ + m×0020_H	8, 16, 32	*1	—
Channel m CANFD CRC Register	RSCFDnCFDCmFDCRC	<RSCFDn_base> + 1410_H + m×0020_H	8, 16, 32	*1	—
Channel m Bus load Control Register	RSCFDnCFDCmBLCT	<RSCFDn_base> + 1418_H + m×0020_H	8, 16, 32	*1	—
Channel m Bus load Status Register	RSCFDnCFDCmBLSTS	<RSCFDn_base> + $141C_H$ + m×0020_H	8, 16, 32	*1	—
Global Acceptance Filter List ID Register j	RSCFDnCFDGAFLIDj	<RSCFDn_base> + 1800_H + j×0010_H	8, 16, 32	*2	—
Global Acceptance Filter List Mask Register j	RSCFDnCFDGAFLMj	<RSCFDn_base> + 1804_H + j×0010_H	8, 16, 32	*2	—
Global Acceptance Filter List Pointer 0 Register j	RSCFDnCFDGAFLP0j	<RSCFDn_base> + 1808_H + j×0010_H	8, 16, 32	*2	—
Global Acceptance Filter List Pointer 1 Register j	RSCFDnCFDGAFLP1j	<RSCFDn_base> + $180C_H$ + j×0010_H	8, 16, 32	*2	—
Channel m Transmit History List Access Register 0	RSCFDnCFDTHLACC0m	<RSCFDn_base> + 8000_H + m×0008_H	8, 16, 32	*1	—
Channel m Transmit History List Access Register 1	RSCFDnCFDTHLACC1m	<RSCFDn_base> + 8004_H + m×0008_H	8, 16, 32	*1	—
RAM Test Page Access Register r	RSCFDnCFDRPGACCr	<RSCFDn_base> + 8400_H + r×0004_H	8, 16, 32	*2	—
Receive Message Buffer Component bm	RSCFDnCFDRMBCPbm	Refer to **Table 23.106**	8, 16, 32	*1	—
Receive Message Buffer ID Register	RSCFDnCFDRMID	Refer to **Table 23.106**	8, 16, 32	*1	—
Receive Message Buffer Pointer Register	RSCFDnCFDRMPTR	Refer to **Table 23.106**	8, 16, 32	*1	—
Receive Message Buffer CANFD Status Register	RSCFDnCFDRMFDSTS	Refer to **Table 23.106**	8, 16, 32	*1	—
Receive Message Buffer Data Field d Register	RSCFDnCFDRMDFd	Refer to **Table 23.106**	8, 16, 32	*1	—
Receive FIFO Access Message Buffer Component b	RSCFDnCFDRFMBCPb	Refer to **Table 23.106**	8, 16, 32	*2	—
Receive FIFO Access ID Register	RSCFDnCFDRFID	Refer to **Table 23.106**	8, 16, 32	*2	—
Receive FIFO Access Pointer Register	RSCFDnCFDRFPTR	Refer to **Table 23.106**	8, 16, 32	*2	—
Receive FIFO Access CANFD Status Register	RSCFDnCFDRFFDSTS	Refer to **Table 23.106**	8, 16, 32	*2	—
Receive FIFO Access Data Field d Register	RSCFDnCFDRFDFd	Refer to **Table 23.106**	8, 16, 32	*2	—
Transmit/Receive FIFO Access Message Buffer Component bm	RSCFDnCFDCFMBCPbm	Refer to **Table 23.106**	8, 16, 32	*1	—

Table 23.13 Registers (4/4)

Register Name	Symbol	Address	Access Size	Access Protection	
				PBG	Other
Transmit/Receive FIFO Access ID Register	RSCFDnCFDCFID	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit/Receive FIFO Access Pointer Register	RSCFDnCFDCFPTR	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit/Receive FIFO Access CANFD Control/Status Register	RSCFDnCFDCFFDCSTS	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit/Receive FIFO Access Data Field d Register	RSCFDnCFDCFDFd	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit Message Buffer Component bm	RSCFDnCFDTMBCPbm	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit Message Buffer ID Register	RSCFDnCFDTMID	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit Message Buffer Pointer Register	RSCFDnCFDTMPTR	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit Message Buffer CANFD Control Register	RSCFDnCFDTMFDCTR	Refer to **Table 23.106**	8, 16, 32	*1	—
Transmit Message Buffer Data Field d Register	RSCFDnCFDTMDFd	Refer to **Table 23.106**	8, 16, 32	*1	—

Note 1.
- n = 0, m = 0: PBG80#2
- n = 0, m = 1: PBG80#3
- n = 0, m = 2: PBG80#4
- n = 0, m = 3: PBG80#5
- n = 0, m = 4: PBG80#6
- n = 0, m = 5: PBG80#7
- n = 0, m = 6: PBG80#8
- n = 0, m = 7: PBG80#9
- n = 1, m = 0: PBG32#13
- n = 1, m = 1: PBG32#14
- n = 1, m = 2: PBG32#15
- n = 1, m = 3: PBG33#0
- n = 1, m = 4: PBG33#1
- n = 1, m = 5: PBG33#2
- n = 1, m = 6: PBG33#3
- n = 1, m = 7: PBG33#4

Note 2. n = 0, RSCFD0 Global: PBG80#10 n = 1, RSCFD1 Global: PBG33#5

23.3.2 Details of Channel-Related Registers

23.3.2.1 RSCFDnCFDCmNCFG — Channel m Nominal Bit Rate Configuration Register

Access: RSCFDnCFDCmNCFG register can be read or written in 32-bit units
RSCFDnCFDCmNCFGL, RSCFDnCFDCmNCFGH register can be read or written in 16-bit units
RSCFDnCFDCmNCFGLL, RSCFDnCFDCmNCFGLH, RSCFDnCFDCmNCFGHL,
RSCFDnCFDCmNCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmNCFG: <RSCFDn_base> + 0000_H + (10_H × m)

RSCFDnCFDCmNCFGL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCFDnCFDCmNCFGH: <RSCFDn_base> + 0002_H + (10_H × m)

RSCFDnCFDCmNCFGLL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCFDnCFDCmNCFGLH: <RSCFDn_base> + 0001_H + (10_H × m),
RSCFDnCFDCmNCFGHL: <RSCFDn_base> + 0002_H + (10_H × m),
RSCFDnCFDCmNCFGHH: <RSCFDn_base> + 0003_H + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NTSEG2[6:0]							NTSEG1[7:0]								NSJW[6]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NSJW[5:0]						NBRP[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.14 RSCFDnCFDCmNCFG Register Contents

Bit Position	Bit Name	Function
31 to 25	NTSEG2[6:0]	Time Segment 2 0000000: Setting prohibited 0000001: 2 Tq : : 1111110: 127 Tq 1111111: 128 Tq
24 to 17	NTSEG1[7:0]	Time Segment 1 00000000: Setting prohibited 00000001: 2 Tq : : 11111110: 255 Tq 11111111: 256 Tq
16 to 10	NSJW[6:0]	Nominal Bit Rate Resynchronization Jump Width Control 0000000: 1 Tq 0000001: 2 Tq : : 1111110: 127 Tq 1111111: 128 Tq
9 to 0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler Nominal Baud Rate Prescaler division ratio

This register is used to configure the transmission / reception nominal Baud Rate parameters of the channels.

RSCFDnCFDCmNCFG.NTSEG2[6:0]

Timing Segment 2.

These bits are used to set the segment TSEG2 to compensate for edges on the CAN Bus with a negative phase error.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should configure a Tq value, only between 2 and 128 inclusive.

RSCFDnCFDCmNCFG.NTSEG1[7:0]

Timing Segment 1.

These bits are used to set the segment TSEG1 to compensate for edges on the CAN Bus with a positive phase error.

It also contains the propagation segment.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should configure a Tq value, only between 2 and 256 inclusive. See **Section 23.5.1.2** for more details.

RSCFDnCFDCmNCFG.NSJW[6:0]

Resynchronization Jump Width.

These bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDCmNCFG.NBRP[9:0]

Channel Nominal Baud Rate Prescaler.

These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.2.2 RSCFDnCFDCmCTR — Channel m Control Register

Access: RSCFDnCFDCmCTR register can be read or written in 32-bit units
RSCFDnCFDCmCTRL, RSCFDnCFDCmCTRH register can be read or written in 16-bit units
RSCFDnCFDCmCTRLL, RSCFDnCFDCmCTRLH, RSCFDnCFDCmCTRHL, RSCFDnCFDCmCTRHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmCTR: <RSCFDn_base> + 0004_H + (10_H × m)

RSCFDnCFDCmCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCFDnCFDCmCTRH: <RSCFDn_base> + 0006_H + (10_H × m)

RSCFDnCFDCmCTRLL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCFDnCFDCmCTRLH: <RSCFDn_base> + 0005_H + (10_H × m),
RSCFDnCFDCmCTRHL: <RSCFDn_base> + 0006_H + (10_H × m),
RSCFDnCFDCmCTRHH: <RSCFDn_base> + 0007_H + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ROM	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 23.15 RSCFDnCFDCmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	ROM	Restricted Operation Mode 0: Restricted Operation Mode is disabled 1: Restricted Operation Mode is enabled
30	CRCT	CRC Error Test 0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted
29 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26, 25	CTMS[1:0]	Channel Test Mode Select 00: Basic test mode 01: Listen-Only mode 10: Self-test mode 0 (External Loop back mode) 11: Self-test mode 1 (Internal Loop back mode)
24	CTME	Channel Test Mode Enable 0: Channel Test Mode disabled 1: Channel Test Mode enabled
23	ERRD	Channel Error Display 0: Only the 1st set of error codes displayed 1: Accumulated error codes displayed
22, 21	BOM[1:0]	Channel Bus-Off Mode 00: Normal mode (comply with ISO 11898-1 (2015)) 01: Entry to Halt Mode automatically at Bus-Off start 10: Entry to Halt Mode automatically at Bus-Off end 11: Entry to Halt Mode (during Bus-Off Recovery Period) by S/W
20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	TDCVFIE	Transceiver Delay Compensation Violation Interrupt enable 0: Transceiver Delay Compensation Violation Interrupt disabled 1: Transceiver Delay Compensation Violation Interrupt enabled

Table 23.15 RSCFDnCFDCmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled
17	EOCOIE	Error occurrence counter overflow Interrupt enable 0: Error occurrence counter overflow Interrupt disabled 1: Error occurrence counter overflow Interrupt is enabled
16	TAIE	Transmit Abort Interrupt Enable 0: Tx abort Interrupt disabled 1: Tx abort Interrupt enabled
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration Lost Interrupt disabled 1: Arbitration Lost Interrupt enabled
14	BLIE	Bus Lock Interrupt Enable 0: Bus Lock Interrupt disabled 1: Bus Lock Interrupt enabled
13	OLIE	Overload Interrupt Enable 0: Overload Interrupt disabled 1: Overload Interrupt enabled
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-Off Recovery Interrupt disabled 1: Bus-Off Recovery Interrupt enabled
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-Off Entry Interrupt disabled 1: Bus-Off Entry Interrupt enabled
10	EPIE	Error Passive Interrupt Enable 0: Error Passive Interrupt disabled 1: Error Passive Interrupt enabled
9	EWIE	Error Warning Interrupt Enable 0: Error Warning Interrupt disabled 1: Error Warning Interrupt enabled
8	BEIE	Bus Error Interrupt Enable 0: Bus Error Interrupt disabled 1: Bus Error Interrupt enabled
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RTBO	Return from Bus-Off 0: Channel is not forced to return from Bus-Off 1: Channel is forced to return from Bus-Off
2	CSLPR	Channel Stop Request 0: Channel Stop Request disabled 1: Channel Stop Request enabled
1, 0	CHMDC[1:0]	Channel Mode Control 00: Channel Communication Mode request 01: Channel Reset request 10: Channel Halt request 11: Keep current value

Each Channel control register is used to control the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

RSCFDnCFDCmCTR.ROM

Restricted Operation Mode.

The Restricted Operation Mode is enabled if RSCFDnCFDCmCTR.ROM and RSCFDnCFDCmCTR.CTME are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_HALT mode.

This mode should only be used in Basic Test mode RSCFDnCFDCmCTR.CTMS(1:0) = 00_B.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmCTR.CRCT

CRC Error Test.

This bit is used to check the internal CRC generator logic of the protocol controller.
It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:
RSCFDnCFDCmERFL.CRCREG (classical CAN frames)
RSCFDnCFDCmFDCRC.CRCREG (CANFD frames)

There are some limitations when using this bit:

- It is not possible to use this feature with CAN nodes connected to the MCU externally, only with nodes connected to the Internal CAN Bus communication can be used.
- One CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream
- Note, the transmitter and receiver mode are sharing the same CRC generator, therefore, it is not necessary to consider the modes separately when testing this limitation.
- The CRC Error Test Mode is enabled if RSCFDnCFDCmCTR.CRCT (new control signal which is inverting the first bit of the bit stream) and RSCFDnCFDCmCTR.CTME are both 1_B.

This bit cannot be set in CH_STOP mode.

Users can write to this bit only when the related RS-CANFD channel is in CH_HALT mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.CTMS[1:0]

Channel Test Mode Select

The RSCFDnCFDCmCTR.CTMS bits are used to select the required test mode.

These bits cannot be set in CH_STOP or CH_RESET mode.

Users should write to these bits only when the related RS-CANFD channel is in CH_HALT mode.

The field is cleared automatically when the related RS-CANFD channel moves to CH_RESET mode.

RSCFDnCFDCmCTR.CTME

Channel Test Mode Enable.

This bit is set to enable the channel test modes.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT mode

This bit cannot be set in CH_STOP mode.

The bit is cleared automatically when the related RS-CANFD channel transits to CH_RESET mode.

RSCFDnCFDCmCTR.ERRD

Channel Error Display.

This bit controls the display mode of the error flag bits (bits [14:8]) in the Channel Error Flag Register

(RSCFDnCFDCmERFL).
If the RSCFDnCFDCmCTR.ERRD bit is 0_B and more than one error occurs at the same time, then the error flag bits will set for all the errors that occurred at the same time. No further errors are flagged until RSCFDnCFDCmERFL[14:8] is cleared.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDCmCTR.BOM[1:0]

Channel Bus-Off Mode.

These bits control the timing of the recovery from Bus-Off mode of the RS-CANFD Channel.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.TDCVFIE

Transceiver Delay Compensation Violation Interrupt enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.TDCVFIE bit is 1_B and the RSCFDnCFDCmFDSTS.TDCVF

bit belonging to the corresponding CAN channel is 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmCTR.SOCOIE

Successful Occurrence Counter Overflow Interrupt enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.SOCOIE bit is 1_B and the RSCFDnCFDCmFDSTS.SOCO bit belonging to the corresponding CAN channel is 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.EOCOIE

Error occurrence counter overflow Interrupt enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.EOCOIE bit is 1_B and the RSCFDnCFDCmFDSTS.EOCO bit

belonging to the corresponding CAN channel is 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.TAIE

Transmission abort Interrupt Enable.

An interrupt is generated, when the RSCFDnCFDCmCTR.TAIE bit is 1_B and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.ALIE

Arbitration Lost Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.ALIE bit and the RSCFDnCFDCmERFL.ALF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.BLIE

Bus Lock Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.BLIE bit and the RSCFDnCFDCmERFL.BLF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.OLIE

Overload Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.OLIE bit and the RSCFDnCFDCmERFL.OVLF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.BORIE

Bus-Off Recovery Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.BORIE bit and the RSCFDnCFDCmERFL.BORF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.BOEIE

Bus-Off Entry Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.BOEIE bit and the RSCFDnCFDCmERFL.BOEF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.EPIE

Error Passive Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.EPIE bit and the RSCFDnCFDCmERFL.EPF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.EWIE

Error Warning Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.EWIE bit and the RSCFDnCFDCmERFL.EWF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.BEIE

Bus Error Interrupt Enable.

An error interrupt is generated, when the RSCFDnCFDCmCTR.BEIE bit and the RSCFDnCFDCmERFL.BEF are both 1_B.

This bit cannot be set in CH_STOP mode.

Users should write to this bit only when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmCTR.RTBO

Return from Bus-Off.

When the protocol controller of the CAN channel enters Bus-Off state, users can force it to recover from Bus-Off state by setting the RSCFDnCFDCmCTR.RTBO bit in the Channel Control Register to 1_B.
The error state changes from Bus-Off state to integrating with a maximum delay of 1 CAN Bit time.

If the RSCFDnCFDCmCTR.RTBO bit is set to 1_B, the REC and TEC registers are initialized and the Bus-Off status bit (Channel Bus-off Status, RSCFDnCFDCmSTS .BOSTS) is set to 0_B.

The other registers are not initialized by this command. Even if RSCFDnCFDCmCTR.BORIE is set, a Bus-Off recovery interrupt is not generated by this recovery from the Bus-Off state.

This bit cannot be set in CH_STOP mode.
Setting this bit in any state (other than Bus-Off) will have no effect and the bit will be cleared immediately.

Read value is always 0.

Return from Bus-Off command should be used only when RSCFDnCFDCmCTR.BOM is set to 00_B.
Users should only write to this bit when the related RS-CANFD channel is in CH_COMMUNICATION mode.

This bit is automatically cleared once set by the System SW.

RSCFDnCFDCmCTR.CSLPR

Channel Stop Request.

A Stop Mode request is generated for the corresponding CAN channel, when the RSCFDnCFDCmCTR.CSLPR bit is 1_B.
A request to exit Stop Mode is generated for the corresponding CAN channel, when the RSCFDnCFDCmCTR.CSLPR bit is 0_B.

Users can only write to this bit when the related RS-CANFD channel is in CH_RESET or CH_STOP mode.

RSCFDnCFDCmCTR.CHMDC[1:0]

Channel Mode Control.

RSCFDnCFDCmCTR.CHMDC bits can be used to configure the modes of the CAN channel.
CAN mode transitions are described in more details in **Section 23.4.2**

Setting RSCFDnCFDCmCTR.CHMDC to 11_B by the CPU has no effect.
When the RS-CANFD module is in GL_TEST mode, these bits can only be set to 10_B or 01_B.
This bit cannot be set in CH_STOP mode.

RSCFDnCFDCmCTR.CHMDC can change automatically in case of transition to Halt mode due to RSCFDnCFDCmCTR.BOM settings.
If CPU write access to RSCFDnCFDCmCTR.CHMDC happens at the same time when the CAN channel is about to enter Halt Mode (at the start of Bus-Off when RSCFDnCFDCmCTR.BOM=01_B, or at the end of Bus-Off when RSCFDnCFDCmCTR.BOM =10_B), then the CPU write access will have the highest priority.

The CAN channel changes the value of RSCFDnCFDCmCTR.CHMDC within the Channel Control Register for the above cases only if the RSCFDnCFDCmCTR.CHMDC value is 00_B (Communication Mode).

23.3.2.3 RSCFDnCFDCmSTS — Channel m Status Register

Access: RSCFDnCFDCmSTS register can be read or written in 32-bit units
RSCFDnCFDCmSTS L, RSCFDnCFDCmSTS H register can be read or written in 16-bit units
RSCFDnCFDCmSTS LL, RSCFDnCFDCmSTS LH, RSCFDnCFDCmSTS HL, RSCFDnCFDCmSTS HH register can be read or written in 8-bit units

Address: RSCFDnCFDCmSTS : <RSCFDn_base> + 0008_H + (10_H × m)

RSCFDnCFDCmSTS L: <RSCFDn_base> + 0008_H + (10_H × m),
RSCFDnCFDCmSTS H: <RSCFDn_base> + $000A_H$ + (10_H × m)

RSCFDnCFDCmSTS LL: <RSCFDn_base> + 0008_H + (10_H × m),
RSCFDnCFDCmSTS LH: <RSCFDn_base> + 0009_H + (10_H × m),
RSCFDnCFDCmSTS HL: <RSCFDn_base> + $000A_H$ + (10_H × m),
RSCFDnCFDCmSTS HH: <RSCFDn_base> + $000B_H$ + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 23.16 RSCFDnCFDCmSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	Transmission Error Count This register increments or decrements the counter value according to error status of the CAN channel during Transmission.
23 to 16	REC[7:0]	Reception Error Count This register increments or decrements the counter value according to error status of the CAN channel during Reception.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	ESIF	Error State Indication Flag 0: No CANFD message has been received with the ESI flag was set 1: At least 1 CANFD message was received where the ESI flag was set
7	COMSTS	Channel Communication Status 0: Channel is not ready for communication 1: Channel is ready for communication
6	RECSTS	Channel Receive Status 0: Channel is not receiving 1: Channel is receiving
5	TRMSTS	Channel Transmit Status 0: Channel is not transmitting 1: Channel is transmitting
4	BOSTS	Channel Bus-Off Status 0: Channel not in Bus-Off state 1: Channel in Bus-Off state
3	EPSTS	Channel Error Passive Status 0: Channel not in Error Passive state. 1: Channel in Error Passive state.
2	CSLPSTS	Channel STOP Status 0: Channel not in Stop Mode 1: Channel in Stop Mode

Table 23.16 RSCFDnCFDCmSTS Register Contents (2/2)

Bit Position	Bit Name	Function
1	CHLTSTS	Channel HALT Status 0: Channel not in Halt Mode 1: Channel in Halt Mode
0	CRSTSTS	Channel RESET Status 0: Channel not in Reset Mode 1: Channel in Reset Mode

Each Channel Status Register shows the mode, Error and TX / RX status of the related channel together with its Reception and Transmission Error Count values.

RSCFDnCFDCmSTS .TEC[7:0]

Transmission Error Count

The value of TEC error counter is displayed.

Users should only write to these bits when in Test Mode and RS-CANFD channel is in CH_HALT mode.

These bits are cleared automatically when RS-CANFD module is in GL_RESET or CAN channel is in CH_RESET mode.

RSCFDnCFDCmSTS .REC[7:0]

Reception Error Count

The value of the REC error counter is displayed.

The value in Bus-Off state is indeterminate.

These bits are cleared automatically when RS-CANFD module enters GL_RESET or CAN channel is in CH_RESET mode.

RSCFDnCFDCmSTS .ESIF

Error State Indication Flag

This flag is set when the ESI bit was sampled recessive for a reception CAN message without any error.
Note that in case of Loopback or Mirror Mode the self transmitted messages are considered as reception messages.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.
This bit is cleared automatically when the related CAN channel is in Reset.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

RSCFDnCFDCmSTS .COMSTS

Channel Communication Status

Indicates if the related CAN channel is ready for communication.

This bit is set automatically when the related CAN channel is ready to perform communication following the detection of eleven consecutive recessive bits after leaving the Reset or Halt mode.

This bit is cleared automatically when the related CAN channel is in Reset or Halt Mode.

NOTE

This bit is 1_B during Bus-Off status.

RSCFDnCFDCmSTS .RECSTS

Channel Receive Status

Indicates if the related CAN channel is receiving a message.

This bit is set automatically when the related CAN channel is operating as a receiver node.

This bit is cleared automatically when the related CAN channel is in the bus-idle state or starts operating as a transmitter node.

RSCFDnCFDCmSTS .TRMSTS

Channel Transmit Status

Indicates if the related CAN channel is transmitting a message.

This bit is set automatically when the related CAN channel is operating as a transmitter node or is in the Bus-Off state.

This bit is cleared automatically when the related CAN channel is in the bus-idle state or starts operating as a receiver node.

RSCFDnCFDCmSTS .BOSTS

Channel Bus-Off Status

Indicates if the related CAN channel has entered the error Bus-Off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 255 and the related CAN channel is in the Bus-Off state (CAN Transmission Error Count Register > 255).

This bit is cleared automatically when the related CAN channel exits the Bus-Off state.

RSCFDnCFDCmSTS .EPSTS

Channel Error Passive Status.

Indicates if the related CAN channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 127.

This bit is cleared automatically when the related CAN channel exits the Error Passive state or enters Reset Mode.

RSCFDnCFDCmSTS .CSLPSTS

Channel STOP Status.

Indicates if the related CAN channel is in Stop Mode.

This bit is set automatically when the related CAN channel enters Stop Mode.

This bit is cleared automatically when the related CAN channel exits Stop Mode.

RSCFDnCFDCmSTS .CHLTSTS

Channel HALT Status.

Indicates if the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN channel enters Halt Mode.

This bit is cleared automatically when the related CAN channel exits Halt Mode.

RSCFDnCFDCmSTS .CRSTSTS

Channel RESET Status.

Indicates if the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset Mode. If the mode is changed from Reset Mode to Stop Mode, RSCFDnCFDCmSTS.CRSTSTS remains set to 1_B.

This bit is cleared automatically when the related CAN channel exits the Channel Reset Mode, except when changing to Stop Mode.

23.3.2.4 RSCFDnCFDCmERFL — Channel m Error Flag Register

Access: RSCFDnCFDCmERFL register can be read or written in 32-bit units
RSCFDnCFDCmERFLL, RSCFDnCFDCmERFLH register can be read or written in 16-bit units
RSCFDnCFDCmERFLLL, RSCFDnCFDCmERFLLH, RSCFDnCFDCmERFLHL, RSCFDnCFDCmERFLHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmERFL: <RSCFDn_base> + $000C_H$ + (10_H × m)

RSCFDnCFDCmERFLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCFDnCFDCmERFLH: <RSCFDn_base> + $000E_H$ + (10_H × m)

RSCFDnCFDCmERFLLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCFDnCFDCmERFLLH: <RSCFDn_base> + $000D_H$ + (10_H × m),
RSCFDnCFDCmERFLHL: <RSCFDn_base> + $000E_H$ + (10_H × m),
RSCFDnCFDCmERFLHH: <RSCFDn_base> + $000F_H$ + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.17 RSCFDnCFDCmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Register value These bits show the CRC value calculated for the CAN2.0 CAN Frame.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	ADERR	Acknowledge Delimiter Error 0: Channel Ack Del Error not detected 1: Channel Ack Del Error detected
13	B0ERR	Bit 0 Error 0: Channel Bit 0 Error not detected 1: Channel Bit 0 Error detected
12	B1ERR	Bit 1 Error 0: Channel Bit 1 Error not detected 1: Channel Bit 1 Error detected
11	CERR	CRC Error 0: Channel CRC Error not detected 1: Channel CRC Error detected
10	AERR	Acknowledge Error 0: Channel Ack Error not detected 1: Channel Ack Error detected
9	FERR	Form Error 0: Channel Form Error not detected 1: Channel Form Error detected
8	SERR	Stuff Error 0: Channel stuff Error not detected 1: Channel stuff Error detected

Table 23.17 RSCFDnCFDCmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
7	ALF	Arbitration Lost Flag 0: Channel Arbitration Lost not detected 1: Channel Arbitration Lost detected
6	BLF	Bus Lock Flag 0: Channel Bus Lock not detected 1: Channel Bus Lock detected
5	OVLF	Overload Flag 0: Channel Overload not detected 1: Channel Overload detected
4	BORF	Bus-Off Recovery Flag 0: Channel Bus-Off Recovery not detected 1: Channel Bus-Off Recovery detected
3	BOEF	Bus-Off Entry Flag 0: Channel Bus-Off Entry not detected 1: Channel Bus-Off Entry detected
2	EPF	Error Passive Flag 0: Channel Error Passive not detected 1: Channel Error Passive detected
1	EWF	Error Warning Flag 0: Channel Error Warning not detected 1: Channel Error Warning detected
0	BEF	Bus Error Flag 0: Channel Bus Error not detected 1: Channel Bus Error detected

Each Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1 (2015)) to check when each error condition can occur.

For this register, when only a single bit is to be cleared by the program, do not use bit clear instruction – use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

An example in assembler language for clearing the RSCFDnCFDCmERFL.BEF bit:

```
mov.b #0Feh, RSCFDnCFDCmERFL ;
```

RSCFDnCFDCmERFL.CRCREG[14:0]

CRC Register value.

The calculated CRC value can be read from these bits, only when RSCFDnCFDCmCTR.CTME is 1_B for the channel.

If RSCFDnCFDCmCTR.CTME bit is 0_B, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the RS-CANFD channel logic if the CTME bit is enabled.

The RSCFDnCFDCmERFL.CRCREG value is updated in the 1st bit of CRC field of the CAN Frame (reception as well as transmission).

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDCmERFL.ADERR

Acknowledge Delimiter Error.

Indicates a detection of a Acknowledge Delimiter bit error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:
1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a form error is detected during the Acknowledge Delimiter state of frame transmission.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.B0ERR

Bit 0 Error.

Indicates a detection of a Dominant bit error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:

1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a Dominant bit error (expected dominant bit, sampled as recessive bit) is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.B1ERR

Bit 1 Error.

Indicates a detection of a Recessive bit error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:
1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a Recessive bit error (expected recessive bit, sampled as dominant bit) is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.CERR

CRC Error.

Indicates a detection of a CAN CRC Error.

This bit can only be cleared by writing 0 to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:

1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a CRC error is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.AERR

Acknowledge Error

Indicates a detection of a CAN Acknowledge Error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:

1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when an Acknowledge error is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.FERR

Form Error

Indicates a detection of a CAN Form Error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:
1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a Form error is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.SERR

Stuff Error

Indicates a detection of a CAN Stuff Error.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

For clearing this bit, the following sequence should be used:
1. Clear the corresponding flag bit
2. Read if the flag bit is cleared
3. If yes, continue, else, go back to step 1.

This bit is set automatically when a Stuff error is detected.
If RSCFDnCFDCmCTR.ERRD bit is 1_B, then this bit will be set if the set and clear conditions occur at the same time for this bit.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.
If RSCFDnCFDCmCTR.ERRD bit is 0_B and the set and clear conditions occur at the same time for this bit, it will be cleared if a bit at RSCFDnCFDCmERFL[14:8] is already set and it will be set if RSCFDnCFDCmERFL[14:8] is 00_H.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.ALF

Arbitration Lost Flag

Indicates a detection of a CAN channel Bus Arbitration Lost condition.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Communication Mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.BLF

Bus Lock Flag

Indicates a detection of a CAN channel Bus Lock condition.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Communication Mode.
If the set condition occurs simultaneously with the clear condition, the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.OVLF

Overload Flag

Indicates a detection of a CAN channel Overload State.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when an overload condition is detected.
If the set condition occurs simultaneously with the clear condition, the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.BORF

Bus-Off Recovery Flag

Indicates a detection of a CAN channel Bus-Off Recovery State

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically if CAN channel recovers from Bus-Off state in the following conditions:

- When RSCFDnCFDCmCTR.BOM is 00_B and normal recovery (11 consecutive recessive bits X 128 times detected) occurs.
- When RSCFDnCFDCmCTR.BOM is 10_B and normal recovery (11 consecutive recessive bits X 128 times detected) occurs.
- When RSCFDnCFDCmCTR.BOM is 11_B and normal recovery (11 consecutive recessive bits X 128 times detected) occurs.

The bit is not set if CAN channel recovers from Bus-Off state in the following conditions:

- When CAN Reset Mode is requested
- When RSCFDnCFDCmCTR.RTBO is instructed (the CAN channel returns to error active)
- When RSCFDnCFDCmCTR.BOM is 01_B
- When RSCFDnCFDCmCTR.BOM is 11_B and a Halt Request is asserted before the CAN channel reaches the end of the Bus-Off State.

The bit is cleared by writing 0 to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

If the set condition occurs simultaneously with the clear condition, the bit is set.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.BOEF

Bus-Off Entry Flag

Indicates a detection of a CAN channel Bus-Off Entry State.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when the CAN error state enters the Bus-Off State.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

If the set condition occurs simultaneously with the clear condition, the bit is set.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.EPF

Error Passive Flag

Indicates a detection of a CAN channel Error Passive State.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when the CAN error state becomes Error Passive State.
The setting of this bit only occurs when the TEC or REC initially exceed 127. Hence if the TEC or REC remain > 127 and the RSCFDnCFDCmERFL.EPF bit is cleared by the system SW, it will not be set again until both the TEC and REC go below 128 and either TEC or REC cross over again from a value ≤ 127 to a value > 127.
If the set condition occurs simultaneously with the clear condition, the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.EWF

Error Warning Flag

This bit indicates if an Error Warning condition has been detected for the CAN channel.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when either TEC or REC exceeds 95.
The setting of this bit only occurs when the TEC or REC initially exceed 95. Hence if the TEC or REC remain > 95 and the EWF bit is cleared by the system SW, it will not be set again until both the TEC and REC go below 96 and either TEC or REC cross over again from a value ≤ 95 to a value > 95.
If the set condition occurs simultaneously with the clear condition, the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDCmERFL.BEF

Bus Error Flag

Indicates a detection of a CAN channel Bus error state, flagged by the bits [14:8] in this register.

This bit can only be cleared by writing 0_B to it when it is 1_B.
This bit can only be set by RS-CANFD module logic.
Writing 1_B has no influence on the bit value.

The bit is set automatically when a Bus Error is detected.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.
It is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

23.3.2.5 RSCFDnCFDCmDCFG — Channel m Data Bit Rate Configuration Register

Access: RSCFDnCFDCmDCFG register can be read or written in 32-bit units
RSCFDnCFDCmDCFGL, RSCFDnCFDCmDCFGH register can be read or written in 16-bit units
RSCFDnCFDCmDCFGLL, RSCFDnCFDCmDCFGLH, RSCFDnCFDCmDCFGHL,
RSCFDnCFDCmDCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmDCFG: <RSCFDn_base> + 1400_H + (20_H × m)

RSCFDnCFDCmDCFGL: <RSCFDn_base> + 1400_H + (20_H × m),
RSCFDnCFDCmDCFGH: <RSCFDn_base> + 1402_H + (20_H × m)

RSCFDnCFDCmDCFGLL: <RSCFDn_base> + 1400_H + (20_H × m),
RSCFDnCFDCmDCFGLH: <RSCFDn_base> + 1401_H + (20_H × m),
RSCFDnCFDCmDCFGHL: <RSCFDn_base> + 1402_H + (20_H × m),
RSCFDnCFDCmDCFGHH: <RSCFDn_base> + 1403_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DSJW[3:0]				—	—	—	—	DTSEG2[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DTSEG1[4:0]					DBRP[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.18 RSCFDnCFDCmDCFG Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	DSJW[3:0]	Resynchronization Jump Width 0000: 1 Tq 0001: 2 Tq : 1111: 16 Tq
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	DTSEG2[3:0]	Time Segment 2 0000: Setting prohibited 0001: 2 Tq : 1110: 15 Tq 1111: 16 Tq
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	DTSEG1[4:0]	Time Segment 1 00000: Setting prohibited 00001: 2 Tq 00010: 3 Tq 00011: 4 Tq : 11110: 31 Tq 11111: 32 Tq
7 to 0	DBRP[7:0]	Channel Data Rate Prescaler Data Rate Prescaler division ratio

This register is used to configure the transmission/reception Data Baud Rate parameters of the channels.

The channel of Classical CAN only mode does not have to perform the configuration of this register.

RSCFDnCFDCmDCFG.DSJW[3:0]

Resynchronization Jump Width.

These bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDCmDCFG.DTSEG2[3:0]

Timing Segment 2.

These bits are used to set the segment TSEG2 to compensate for edges on the CAN Bus with a negative phase error.

Any value from 2 to 16 time quanta can be written to these bits.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.
Users should not write any other value to these bits.

RSCFDnCFDCmDCFG.DTSEG1[4:0]

Timing Segment 1.

These bits are used to set the segment TSEG1 to compensate for edges on the CAN Bus with a positive phase error.
It also contains the propagation segment.

Any value from 2 to 32 time quanta can be configured.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.
Users should not write any other value to these bits. See **Section 23.5.1.2** for more details.

RSCFDnCFDCmDCFG.DBRP[7:0]

Channel Data Baud Rate Prescaler.

These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.2.6 RSCFDnCFDCmFDCFG — Channel m CANFD Configuration Register

Access: RSCFDnCFDCmFDCFG register can be read or written in 32-bit units
RSCFDnCFDCmFDCFGL, RSCFDnCFDCmFDCFGH register can be read or written in 16-bit units
RSCFDnCFDCmFDCFGLL, RSCFDnCFDCmFDCFGLH, RSCFDnCFDCmFDCFGHL,
RSCFDnCFDCmFDCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmFDCFG: <RSCFDn_base> + 1404_H + (20_H × m)

RSCFDnCFDCmFDCFGL: <RSCFDn_base> + 1404_H + (20_H × m),
RSCFDnCFDCmFDCFGH: <RSCFDn_base> + 1406_H + (20_H × m)

RSCFDnCFDCmFDCFGLL: <RSCFDn_base> + 1404_H + (20_H × m),
RSCFDnCFDCmFDCFGLH: <RSCFDn_base> + 1405_H + (20_H × m),
RSCFDnCFDCmFDCFGHL: <RSCFDn_base> + 1406_H + (20_H × m),
RSCFDnCFDCmFDCFGHH: <RSCFDn_base> + 1407_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CLOE	REFE	FDOE	—	GWBRS	GWFDF	GWEN	TDCO[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.19 RSCFDnCFDCmFDCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	CLOE	Classical CAN only enable 0: Classical only mode disabled 1: Classical only mode enabled
29	REFE	Rx edge filter enable 0: Rx edge filter is disabled 1: Rx edge filter is enabled
28	FDOE	CANFD only enable 0: CANFD only mode disabled 1: CANFD only mode enabled
27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	GWBRS	Gateway BRS configuration bit 0: GW frame is transmitted with BRS = 0 1: GW frame is transmitted with BRS = 1
25	GWFDF	Gateway FDF configuration bit 0: GW frame is transmitted as Classical CAN frame 1: GW frame is transmitted as CANFD frame
24	GWEN	CAN2.0, CANFD <> CAN2.0, CANFD Multi Gateway Enable 0: Multi Gateway Disabled 1: Multi Gateway Enable
23 to 16	TDCO[7:0]	Transceiver Delay Compensation Offset Transmitter delay compensation offset value
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 23.19 RSCFDnCFDCmFDCFG Register Contents (2/2)

Bit Position	Bit Name	Function
10	ESIC	Error State Indication Configuration 0: The ESI bit in the frame will be representing the Error state of the node itself 1: The ESI bit in the frame will be representing the Error state of message buffer if the node itself is not in error passive. If the node is in Error Passive then the ESI bit will be driven by the node itself
9	TDCE	Transceiver Delay Compensation Enable 0: Transceiver Delay Compensation Disabled 1: Transceiver Delay Compensation Enabled
8	TDCOC	Transceiver Delay Compensation Offset Configuration 0: Measured + offset 1: offset only
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	EOCCFG[2:0]	Error Occurrence Counter Configuration 000: All Transmitter or Receiver CAN Frames 001: All Transmitter CAN Frames 010: All Receiver CAN Frames 011: Setting prohibited 100: Only Transmitter or Receiver CANFD Data-Phase (fast bits) 101: Only Transmitter CANFD Data-Phase (fast bits) 110: Only Receiver CANFD Data-Phase (fast bits) 111: Setting prohibited

The Channel m CANFD Configuration Register is used to configure which communication direction (transmitter and/or receiver) errors will be counted.

RSCFDnCFDCmFDCFG.CLOE

Classical CAN only enable.

This bits enables the Classical CAN only mode. If this bit is 1_B, then the protocol controller can only send Classical Frames and will react with a Form or CRC error on FD frames.

User should not set RSCFDnCFDCmFDCFG.CLOE and RSCFDnCFDCmFDCFG.FDOE simultaneously.

RSCFDnCFDCmFDCFG.CLOE	RSCFDnCFDCmFDCFG.FDOE	channel mode
0	0	CANFD mode
0	1	CANFD only mode
1	0	Classical CAN only mode
1	1	Reserved

This bit cannot be written in CH_COMMUNICATION, CH_HALT and CH_STOP mode.
Users should only write to these bits when the RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDCFG.REFE

RX edge filter enable.

This bits enables the RX edge filter during the IDLE detection (bus integration). When it is enabled then 2 consecutive dominant TQ required to detect a synchronization edge.

This bit cannot be written in CH_COMMUNICATION, CH_HALT and CH_STOP mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.FDOE

CANFD only enable.

This bits enables the reception and transmission of CANFD only frames. If enabled then communication in Classical CAN frame format is disabled. Transmission of classical CAN frames is not possible, the FDF bit of the message buffer is don't care (RSCFDnCFDCFFDCSTS.CFFDF / RSCFDnCFDTMFDCTR.TMFDF). If messages with classical CAN frame format is received then the protocol controller will treated them as invalid frames and response with error frames.

In case a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, so a CANFD frame is sent. If the DLC is configured bigger than 8 the remaining data bytes are padded with CCh.

User should not set RSCFDnCFDCmFDCFG.FDOE and RSCFDnCFDCmFDCFG.CLOE simultaneously.

This bit cannot be written in CH_COMMUNICATION, CH_HALT and CH_STOP mode.

RSCFDnCFDCmFDCFG.GWBRS

Gateway BRS configuration bit.

When the bit RSCFDnCFDCmFDCFG.GWEN is set to 1_B then the BRS bit of the transmitting GW frame will be replaced by the value of RSCFDnCFDCmFDCFG.GWBRS.

In classical CAN frames the BRS bit has no meaning

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.GWFDF

Gateway FDF configuration bit.

When the bit RSCFDnCFDCmFDCFG.GWEN is set to 1_B then the FDF bit of the transmitting GW frame will be replaced by the value of RSCFDnCFDCmFDCFG.GWFDF.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.GWEN

CAN2.0, CANFD <> CAN2.0, CANFD Multi Gateway Enable

When this bit is enabled then a multi gateway is enabled. Message received on one node can be routed to another node by using the Transmit/Receive FIFO when they are configured as GW FIFO (RSCFDnCFDCFCCk.CFM = 10_B).

Moreover, when TX Queue is set as GW mode, the message received on one node can be stored in TX Queue, and it can be sent to another node.

The FDF and BRS bit of the routed message can be changed by the configuration value of the RSCFDnCFDCmFDCFG.GWFDF and RSCFDnCFDCmFDCFG.GWBRS.

By this the transmitted value of these bits can be replaced.

Example :

RSCFDnCFDCmFDCFG.GWEN = 1 on channel y

RSCFDnCFDCmFDCFG.GWFDF = 1

If a Classical CAN frame is received on channel x and routed to a GW FIFO or TX Queue of channel y. Then this CAN frame is sent on Channel y as a CANFD frame, due to the RSCFDnCFDCmFDCFG.GWFDF bit.

The table below shows how the message information is changed depending on received and configured data.

Routed CAN frame	Routed Received DLC	CAN BRS	Configured RSCFDnCFDCmFDCFG.GWFDF bit	GW message DLC	GW message BRS Bit	GW message frame type
CAN2.0	≤ 8	N/A	1	≤ 8	Based on configuration RSCFDnCFDCmFDCFG.GWBRS	FD
CAN2.0	> 8	N/A	1	= 8	Based on configuration RSCFDnCFDCmFDCFG.GWBRS	FD
FD	≤ 8	?	1	≤ 8	Based on configuration RSCFDnCFDCmFDCFG.GWBRS	FD
FD	> 8	?	1	> 8	Based on configuration RSCFDnCFDCmFDCFG.GWBRS	FD
CAN2.0	≤ 8	N/A	0	≤ 8	N/A	CAN2.0
CAN2.0	> 8	N/A	0	> 8	N/A	CAN2.0
FD	≤ 8	?	0	≤ 8	N/A	CAN2.0
FD	> 8	?	0	= 8	N/A	CAN2.0

Note this Gateway is limited to 8 byte data payload for different frame type. If routing and target frame type is same then the DLC value is untouched.

In case the source frame was a CANFD with more than 8 data byte, then on classical destination node the Data payload is cut down to 8 byte. Only 8 bytes of top data performs GW transmission, and the remaining byte cancels it.

In case the source frame was a CLASSICAL with a DLC value > 8, then on CANFD destination node the DLC value itself is changed to 8.

NOTE

Other transmission buffer than the GW-FIFO are not affected by this feature.

Remote frames should not be routed via the Gateway, when RSCFDnCFDCmFDCFG.GWEN is set.

When a destination node is RSCFDnCFDCmFDCFG.FDOE=1, set RSCFDnCFDCmFDCFG.GWEN=1 and RSCFDnCFDCmFDCFG.GWFDF=1.

When a destination node is RSCFDnCFDCmFDCFG.CLOE=1, set RSCFDnCFDCmFDCFG.GWEN=1 and RSCFDnCFDCmFDCFG.GWFDF=0.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDCFG.TDCO[7:0]

Transceiver Delay Compensation Offset

These bits set the secondary sample point offset. How this value is used, depends on the RSCFDnCFDCmFDCFG.TDCOC setting.

If RSCFDnCFDCmFDCFG.TDCOC = 0_B, the Transceiver Delay Compensation result is equal to the Trv_Delay (measured delay) + the value in RSCFDnCFDCmFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the Transceiver Delay Compensation result is equal to the value in RSCFDnCFDCmFDCFG.TDCO. Refer to **Section 23.5.1.5** for details on how RSCFDnCFDCmFDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. E.g if 4 is set in TDCO, the offset is 5 clock cycle.

Clock cycle is 1 cycle of CAN channel DLL clock.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should write to these bits, only when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.ESIC

Error State Indication Configuration

Bus controllers to be used as CAN to CAN gateways shall support that in every forwarded CANFD message the ESI flag is not changed to reflect the status of the gateway/bridge/router but instead the ESI flag is sent as it was in the original message.

This Error State Indication Configuration, controls the sending of either the own ESI flag information or the message ESI flag information (RSCFDnCFDCFFDCSTS.CFESI or RSCFDnCFDTMFDCTR.TMESI)

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.TDCE

Transceiver Delay Compensation Enable

This bit enables the Transceiver Delay Compensation for the RS-CANFD channel.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.TDCOC

Transceiver Delay Compensation Offset Configuration

This bit selects which offset is used when defining the position of the secondary sample point (SSP) for the RS-CANFD channel. If the bit is set to 0_B then the position of the SSP is the measured Transceiver delay plus the fixed offset. If the bit is 1_B then the position of the SSP is defined only by the offset.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

Users should not set to this bit when Classical only mode.

RSCFDnCFDCmFDCFG.EOCCFG[2:0]

Error Occurrence Counter Configuration

These bits select which type of CAN frame configuration and direction, protocol errors are counted in.

This bit cannot be written in CH_COMMUNICATION or CH_STOP mode.

Users should only write to these bits when the RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.2.7 RSCFDnCFDCmFDCTR — Channel m CANFD Control Register

Access: RSCFDnCFDCmFDCTR register can be read or written in 32-bit units
RSCFDnCFDCmFDCTRL, RSCFDnCFDCmFDCTRH register can be read or written in 16-bit units
RSCFDnCFDCmFDCTRLL, RSCFDnCFDCmFDCTRLH, RSCFDnCFDCmFDCTRHL,
RSCFDnCFDCmFDCTRHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmFDCTR: <RSCFDn_base> + 1408_H + (20_H × m)

RSCFDnCFDCmFDCTRL: <RSCFDn_base> + 1408_H + (20_H × m),
RSCFDnCFDCmFDCTRH: <RSCFDn_base> + $140A_H$ + (20_H × m)

RSCFDnCFDCmFDCTRLL: <RSCFDn_base> + 1408_H + (20_H × m),
RSCFDnCFDCmFDCTRLH: <RSCFDn_base> + 1409_H + (20_H × m),
RSCFDnCFDCmFDCTRHL: <RSCFDn_base> + $140A_H$ + (20_H × m),
RSCFDnCFDCmFDCTRHH: <RSCFDn_base> + $140B_H$ + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 23.20 RSCFDnCFDCmFDCTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SOCCLR	Successful Occurrence Counter Clear 0: No Successful Occurrence Counter clear 1: Clear Successful Occurrence Counter
0	EOCCLR	Error Occurrence Counter Clear 0: No Error Occurrence Counter clear 1: Clear Error Occurrence Counter

The Channel m CANFD Control Register is used to control the error and successful occurrence counters.

RSCFDnCFDCmFDCTR.SOCCLR

Successful Occurrence Counter Clear.

This bit is used to clear the Successful Occurrence Counter.

This bit cannot be set in CH_STOP mode or CH_RESET.

Read value is always 0.

This bit is cleared automatically by the RS-CANFD logic.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDCTR.EOCCLR

Error Occurrence Counter Clear.

This bit is used to clear the error occurrence counter.

This bit cannot be set in CH_STOP mode or CH_RESET.

Read value is always 0.

This bit is cleared automatically by the RS-CANFD logic.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.2.8 RSCFDnCFDCmFDSTS — Channel m CANFD Status Register

Access: RSCFDnCFDCmFDSTS register can be read or written in 32-bit units
RSCFDnCFDCmFDSTSL, RSCFDnCFDCmFDSTSH register can be read or written in 16-bit units
RSCFDnCFDCmFDSTSLL, RSCFDnCFDCmFDSTSLH, RSCFDnCFDCmFDSTSHL,
RSCFDnCFDCmFDSTSHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmFDSTS: <RSCFDn_base> + $140C_H$ + (20_H × m)

RSCFDnCFDCmFDSTSL: <RSCFDn_base> + $140C_H$ + (20_H × m),
RSCFDnCFDCmFDSTSH: <RSCFDn_base> + $140E_H$ + (20_H × m)

RSCFDnCFDCmFDSTSLL: <RSCFDn_base> + $140C_H$ + (20_H × m),
RSCFDnCFDCmFDSTSLH: <RSCFDn_base> + $140D_H$ + (20_H × m),
RSCFDnCFDCmFDSTSHL: <RSCFDn_base> + $140E_H$ + (20_H × m),
RSCFDnCFDCmFDSTSHH: <RSCFDn_base> + $140F_H$ + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOC[7:0]								EOC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDCVF	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 23.21 RSCFDnCFDCmFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	SOC[7:0]	Successful occurrence counter register These bits show the successful occurrence counter value.
23 to 16	EOC[7:0]	Error occurrence counter register These bits show the error occurrence counter value.
15	TDCVF	Transmitter Delay Compensation Violation Flag 0: Transmitter Delay Compensation Violation has not occurred 1: Transmitter Delay Compensation Violation has occurred
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	SOCO	Successful occurrence counter overflow 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed
8	EOCO	Error occurrence counter overflow 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed
7 to 0	TDCR[7:0]	Transmitter Delay Compensation Result Transmitter delay compensation result

The Channel m CANFD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

RSCFDnCFDCmFDSTS.SOC[7:0]

Successful occurrence counter register

These bits are set only by RS-CANFD module logic.

Writing any value has no influence on these bits.

These bits are updated (incremented) when the occurrence of any error-free messages on the bus is detected. Any means (received or transmitted). When the counter reaches the value of FF_H then updating is stopped.

Note in case of Loopback mode the counter would be incremented twice.

These bits are cleared by writing 1_B to RSCFDnCFDCmFDCTR.SOCCLR
These bits are cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDSTS.EOC[7:0]

Error occurrence counter register

This counter is used together with the RSCFDnCFDCmFDSTS.SOC counter to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing reduced payload bit length experience significantly higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the RSCFDnCFDCmFDCFG.EOCCFG

These bits are set only by RS-CANFD module logic.

Writing any value has no influence on these bits.

These bits are updated (incremented) when an error occurs, according to the configuration of the RSCFDnCFDCmFDCFG.EOCCFG bits. When the counter reaches the value of FF_H then updating is stopped.

These bits are cleared by writing 1_B to RSCFDnCFDCmFDCTR.EOCCLR
These bits are cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDSTS.TDCVF

Transceiver Delay Compensation Violation Flag

The RS-CANFD module is capturing internally the transmitted data bit by a Bit.
This data is compared against the received CAN-Bus level which is delayed by the Transceiver loop delay.
The Transceiver delay has some variation depending on physical parameters like temperature. The result flag RSCFDnCFDCmFDSTS.TDCR will be updated by every message. Hence temporary max. delay violation could be missed. This bit is capturing this violation.

Writing 1_B has no influence on the bit value.

This bit is set automatically when the Transceiver Delay Compensation is greater than the max. delay compensation (6 data bit times – 2clk_dlc) and the internal Bit is overrun.

This bit is cleared by writing a 0_B to it.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

RSCFDnCFDCmFDSTS.SOCO

Successful occurrence counter overflow

Indicates if the related CAN channel successful occurrence counter has overflowed.

Writing 1_B has no influence on the bit value.

This bit is set automatically when RSCFDnCFDCmFDSTS.SOC is FF_H and, successful message reception or successful message transmission occurs.

This bit is cleared by writing a 0_B to it.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

RSCFDnCFDCmFDSTS.EOCO

Error occurrence counter overflow

Indicates if the related CAN channel error occurrence counter has overflowed.

Writing 1_B has no influence on the bit value.

This bit is set automatically when RSCFDnCFDCmFDSTS.EOC is FF_H and a CAN bus error is detected based on the configuration defined in RSCFDnCFDCmFDCFG.EOCCFG.

This bit is cleared by writing a 0_B to it.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

RSCFDnCFDCmFDSTS.TDCR[7:0]

Transceiver Delay Compensation Result

The measured delay is a multiple of the CAN channel DLL clock.

This result depends on the RSCFDnCFDCmFDCFG.TDCOC configuration and the offset value in RSCFDnCFDCmFDCFG.TDCO. Refer to **Section 23.5.1.5** for details on how this value is derived.

These bits are set when the transceiver delay has been measured.

These bits are updated at the falling edge between FDF and res bit if RSCFDnCFDCmFDCFG.TDCOC = 0_B and the transceiver delay compensation is enabled (RSCFDnCFDCmFDCFG.TDCE = 1_B).

These bits are cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.2.9 RSCFDnCFDCmFDCRC — Channel m CANFD CRC Register

Access: RSCFDnCFDCmFDCRC register is a read-only register that can be read in 32-bit units
RSCFDnCFDCmFDCRCL, RSCFDnCFDCmFDCRCH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCmFDCRCLL, RSCFDnCFDCmFDCRCLH, RSCFDnCFDCmFDCRCHL, RSCFDnCFDCmFDCRCHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCmFDCRC: <RSCFDn_base> + 1410_H + (20_H × m)

RSCFDnCFDCmFDCRCL: <RSCFDn_base> + 1410_H + (20_H × m),
RSCFDnCFDCmFDCRCH: <RSCFDn_base> + 1412_H + (20_H × m)

RSCFDnCFDCmFDCRCLL: <RSCFDn_base> + 1410_H + (20_H × m),
RSCFDnCFDCmFDCRCLH: <RSCFDn_base> + 1411_H + (20_H × m),
RSCFDnCFDCmFDCRCHL: <RSCFDn_base> + 1412_H + (20_H × m),
RSCFDnCFDCmFDCRCHH: <RSCFDn_base> + 1413_H + (20_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRCREG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.22 RSCFDnCFDCmFDCRC Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned.
27 to 24	SCNT[3:0]	Stuff bit count These bits show the stuff bit count (mod 8) for the CANFD frame.
23 to 21	Reserved	When read, the value after reset is returned.
20 to 0	CRCREG[20:0]	CRC Register value These bits show the CRC value calculated for the CANFD Frame.

The CRC Register holds the CRC value calculated for the CANFD Frame

RSCFDnCFDCmFDCRC.SCNT[3:0]

Stuff bit count

These bits show the Stuff Count value of the CANFD frame. It shows the number of inserted stuff bits (modulo 8, Gray-coded) for a CANFD frame if the RSCFDnCFDCmCTR.CTME bit is enabled on RSCFDnCFDCmFDCRC.SCNT[3:1]. And the corresponding Parity bit to this counter value on RSCFDnCFDCmFDCRC.SCNT[0].

If RSCFDnCFDCmCTR.CTME bit is 0_B, then these bits are always read as 0.

The RSCFDnCFDCmFDCRC.SCNT value is updated in the 1st bit of CRC field of the CANFD Frame (reception as well as transmission).

These bits are cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

RSCFDnCFDCmFDCRC.CRCREG[20:0]

CRC Register value

The calculated CRC value can be read from these bits, only when RSCFDnCFDCmCTR.CTME is 1_B for the

channel.

If RSCFDnCFDCmCTR.CTME bit is 0_B, then these bits are always read as 0.

If CRC_17 (17 bit CRC) is used, then bits [20:17] are always read as 0.

These bits show the CRC value calculated by the RS-CANFD channel logic if the RSCFDnCFDCmCTR.CTME bit is enabled.

The RSCFDnCFDCmFDCRC.CRCREG value is updated in the 1st bit of CRC field of the CANFD Frame (reception as well as transmission).

These bits are cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.3 Details of Global-Related Registers

23.3.3.1 RSCFDnCFDGCFG — Global Configuration Register

Access: RSCFDnCFDGCFG register can be read or written in 32-bit units
RSCFDnCFDGCFGL, RSCFDnCFDGCFGH register can be read or written in 16-bit units
RSCFDnCFDGCFGLL, RSCFDnCFDGCFGLH, RSCFDnCFDGCFGHL, RSCFDnCFDGCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDGCFG: <RSCFDn_base> + 0084_H

RSCFDnCFDGCFGL: <RSCFDn_base> + 0084_H, RSCFDnCFDGCFGH: <RSCFDn_base> + 0086_H

RSCFDnCFDGCFGLL: <RSCFDn_base> + 0084_H, RSCFDnCFDGCFGLH: <RSCFDn_base> + 0085_H,
RSCFDnCFDGCFGHL: <RSCFDn_base> + 0086_H, RSCFDnCFDGCFGHH: <RSCFDn_base> + 0087_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.23 RSCFDnCFDGCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO Interval timer prescaler value
15 to 13	TSBTCS[2:0]	Timestamp Bit Time Channel Select 000: select clock from Channel 0 001: select clock from Channel 1 010: select clock from Channel 2 011: select clock from Channel 3 100: select clock from Channel 4 101: select clock from Channel 5 110: select clock from Channel 6 111: select clock from Channel 7
12	TSSS	Timestamp Source Select 0: Source clock for Timestamp counter is peripheral clock 1: Source clock for Timestamp counter is bit time clock
11 to 8	TSP[3:0]	Timestamp Prescaler 0000: Timestamp Prescaler = 1 0001: Timestamp Prescaler = 2 0010: Timestamp Prescaler = 4 0011: Timestamp Prescaler = 8 : 1101: Timestamp Prescaler = 8192 1110: Timestamp Prescaler = 16384 1111: Timestamp Prescaler = 32768
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CMPOC	CANFD message Payload overflow configuration 0: Message is rejected 1: Messages payload is cut to fit to configured message size
4	DCS	Data Link Controller Clock Select 0: Internal clean clock 1: External Clock source connected to clk_xincan pin

Table 23.23 RSCFDnCFDGCFG Register Contents (2/2)

Bit Position	Bit Name	Function
3	MME	Mirror Mode Enable 0: Mirror Mode disabled 1: Mirror Mode enabled
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled
0	TPRI	Transmission Priority 0: ID Priority 1: Message Buffer Number Priority

The Global Configuration Register is used to select the transmission priority to be used for all the TX Message Buffers and the clock source for the CAN protocol engine of all CAN channels. It is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

RSCFDnCFDGCFG.ITRCP[15:0]

Interval Timer Reference Clock Prescaler.

These bits allow the definition of a reference clock for the FIFO interval timer source clock.

When the RSCFDnCFDGCFG.ITRCP[15:0] bits are 0000_H, then the timer is disabled.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.TSBTCS[2:0]

Timestamp Bit Time Channel Select.

These bits allow the selection of the Bit Time clock of a particular channel for the Timestamp counter.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.TSSS

Timestamp Source Select.

This bit allows the selection of the clock source for the Timestamp counter.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

Users should not write 1_B when CANFD communication will be used.

Note bit time clock could be variable depending on the nominal and data rate bit configuration.

RSCFDnCFDGCFG.TSP[3:0]

Timestamp Prescaler.

The value configured in these bits defines the period of the clock source used for the Timestamp counter.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.CMPOC

CANFD message Payload overflow configuration.

The received message payload is always compared with the available message payload size in the Message Buffer. This bit controls the message payload acceptance mechanism in the case when the received payload is higher than the Message Buffer payload size RSCFDnCFDRMNB.RMPLS, RSCFDnCFDRFCCx.RFPLS, RSCFDnCFDCFCCk.CFPLS.

Users cannot write to this bit in GL_STOP & GL_OPERATING mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, DLC value is stored in a RXMB or FIFO without changing.

RSCFDnCFDGCFG.DCS

Data Link Controller Clock Select.

This bit selects the clock source for the CAN communications. Internal clean clock has a smaller clock jitter than the Peripheral clock (pclk).

Users cannot write to this bit in GL_STOP & GL_OPERATING mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.MME

Mirror Mode Enable.

This bit enables the Mirror Mode for all CAN channels.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.DRE

DLC Replacement Enable.

If this bit is 1_B and RSCFDnCFDGCFG.DCE is 1_B, then RS-CANFD will store the configured value (RSCFDnCFDGAFLP0j.GAFLDLC) of DLC in the destination RX Message Buffer or FIFO buffer if the DLC check passes. Otherwise the DLC value in the destination RX Message Buffer or FIFO buffer is unchanged.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.DCE

DLC Check Enable.

This bit enables the DLC check for all CAN channels.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGCFG.TPRI

Transmission Priority.

This bit selects the transmission priority for all CAN channels.

Users cannot write to this bit in GL_STOP mode.

Message Buffer Number Priority should not be used together with TX Queue transmission.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

23.3.3.2 RSCFDnCFDGCTR — Global Control Register

Access: RSCFDnCFDGCTR register can be read or written in 32-bit units
RSCFDnCFDGCTRL, RSCFDnCFDGCTRH register can be read or written in 16-bit units
RSCFDnCFDGCTRLL, RSCFDnCFDGCTRLH, RSCFDnCFDGCTRHL, RSCFDnCFDGCTRHH register can be read or written in 8-bit units

Address: RSCFDnCFDGCTR: <RSCFDn_base> + 0088_H

RSCFDnCFDGCTRL: <RSCFDn_base> + 0088_H, RSCFDnCFDGCTRH: <RSCFDn_base> + $008A_H$

RSCFDnCFDGCTRLL: <RSCFDn_base> + 0088_H, RSCFDnCFDGCTRLH: <RSCFDn_base> + 0089_H,
RSCFDnCFDGCTRHL: <RSCFDn_base> + $008A_H$, RSCFDnCFDGCTRHH: <RSCFDn_base> + $008B_H$

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOWEIE	QMEIE	—	QOWEIE	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.24 RSCFDnCFDGCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset
15	MOWEIE	GW FIFO Message overwrite Error Interrupt Enable 0: GW FIFO Message overwrite Error Interrupt Disabled 1: GW FIFO Message overwrite Error Interrupt Enabled
14	QMEIE	TXQ Message lost Error Interrupt Enable 0: TXQ Message Lost Error Interrupt Disabled 1: TXQ Message Lost Error Interrupt Enabled
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	QOWEIE	TXQ Message overwrite Error Interrupt Enable 0: TXQ Message overwrite Error Interrupt Disabled 1: TXQ Message overwrite Error Interrupt Enabled
11	CMPOFIE	CANFD message payload overflow Flag Interrupt enable 0: CANFD message payload overflow Flag Interrupt Disabled 1: CANFD message payload overflow Flag Interrupt Enabled
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX History List Entry Lost Interrupt Disabled 1: TX History List Entry Lost Interrupt Enabled
9	MEIE	Message lost Error Interrupt Enable 0: Message Lost Error Interrupt Disabled 1: Message Lost Error Interrupt Enabled
8	DEIE	DLC check Interrupt Enable 0: DLC check Interrupt Disabled 1: DLC check Interrupt Enabled
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 23.24 RSCFDnCFDGCTR Register Contents (2/2)

Bit Position	Bit Name	Function
2	GSLPR	Global Stop Request 0: Global Stop Request Disabled 1: Global Stop Request Enabled
1, 0	GMDC[1:0]	Global Mode Control 00: Global Operating Mode Request 01: Global Reset Mode Request 10: Global Test Mode Request 11: Keep Current Value

The Global Control Register is used to control the global mode of the RS-CANFD module and to control the timestamp function. It is also used to enable and disable the global error interrupts

RSCFDnCFDGCTR.TSRST

Timestamp Reset.

When this bit is 1_B, the Global Timestamp Register is reset to 0000_H.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the RS-CANFD logic.

RSCFDnCFDGCTR.MOWEIE

GW FIFO Message overwrite Error Interrupt Enable.

If this bit is 1_B, then an Interrupt will be generated when GW mode and GW FIFO Message over write condition occurs.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.QMEIE

TXQ Message lost Error Interrupt Enable.

If this bit is 1_B, then an Interrupt will be generated when a TXQ Message Lost condition occurs

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.QOWEIE

TXQ Message overwrite Error Interrupt Enable.

If this bit is 1_B, then an Interrupt will be generated when TXQ Message overwrite condition occurs.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.CMPOFIE

CANFD message payload overflow Flag Interrupt enable.

If this bit is 1_B, then an Interrupt will be generated when a CANFD message payload overflow condition occurs.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.THLEIE

TX History List Entry Lost Interrupt Enable.

If this bit is 1_B, then an Interrupt will be generated when a TX History List Entry Lost condition occurs.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.MEIE

Message lost Error Interrupt Enable.

If this bit is 1_B, then an Interrupt will be generated when a Message Lost condition occurs.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.DEIE

DLC check Interrupt Enable

If this bit is 1_B, then an interrupt will be generated when a DLC error is detected in received frames.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDGCTR.GSLPR

Global Stop Request

This bit globally selects the Stop request for RS-CANFD module including all CAN channels (Channel Stop request is set automatically for all channels).

Users can only write to this bit when RS-CANFD module is in GL_RESET or GL_STOP mode.

RSCFDnCFDGCTR.GMDC[1:0]

Global Mode Control

RSCFDnCFDGCTR.GMDC bits can be used to configure the modes for the RS-CANFD module.

Additionally, if RSCFDnCFDGCTR.GSLPR is 1_B when the RS-CANFD module is in Reset Mode, then the RS-CANFD module transits to Global Stop Mode.

Setting RSCFDnCFDGCTR.GMDC to 11_B has no effect. Mode transition is described in detail later in **Section 23.4.1**.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

23.3.3.3 RSCFDnCFDGSTS — Global Status Register

Access: RSCFDnCFDGSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDGSTSL, RSCFDnCFDGSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDGSTSLL, RSCFDnCFDGSTSLH, RSCFDnCFDGSTSHL, RSCFDnCFDGSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDGSTS: <RSCFDn_base> + $008C_H$

RSCFDnCFDGSTSL: <RSCFDn_base> + $008C_H$, RSCFDnCFDGSTSH: <RSCFDn_base> + $008E_H$

RSCFDnCFDGSTSLL: <RSCFDn_base> + $008C_H$, RSCFDnCFDGSTSLH: <RSCFDn_base> + $008D_H$,
RSCFDnCFDGSTSHL: <RSCFDn_base> + $008E_H$, RSCFDnCFDGSTSHH: <RSCFDn_base> + $008F_H$

Value after reset: $0000\ 000D_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.25 RSCFDnCFDGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is finished 1: RAM initialization ongoing
2	GSLPSTS	Global Stop Status 0: Not in Stop Mode 1: In Stop Mode
1	GHLTSTS	Global Test Status 0: Not in Test Mode 1: In Test Mode
0	GRSTSTS	Global Reset Status 0: Not in Reset Mode 1: In Reset Mode

The Global Status Register indicates the global status of the RS-CANFD module.

RSCFDnCFDGSTS.GRAMINIT

Global RAM Initialization.

Indicates the Global RS-CANFD Module RAM initialization state.

This bit is set automatically when the RS-CANFD module enters Global Stop Mode after HW Reset.

This bit is cleared automatically when the RS-CANFD module has finished the RAM initialization.

This bit is cleared when the test_mode input port is set to 1_B.

RSCFDnCFDGSTS.GSLPSTS

Global Stop Status.

Indicates the Global RS-CANFD Module Stop mode.

This bit is set automatically when the RS-CANFD module enters Global Stop Mode.

This bit is cleared automatically when the RS-CANFD module exits the Stop Mode.

RSCFDnCFDGSTS.GHLTSTS

Global Test Status.

Indicates the Global RS-CANFD Module Test mode.

This bit is set automatically when the RS-CANFD module enters Global Test Mode.

This bit is cleared automatically when the RS-CANFD module exits the Test Mode.

RSCFDnCFDGSTS.GRSTSTS

Global Reset Status.

Indicates the Global RS-CANFD Module Reset mode.

This bit is set automatically when the RS-CANFD module enters Global Reset Mode. When the mode is changed from Global Reset Mode to Global Stop Mode, RSCFDnCFDGSTS.GRSTSTS remains set.

This bit is cleared automatically when the RS-CANFD module exits the Global Reset Mode.

23.3.3.4 RSCFDnCFDGERFL — Global Error Flag Register

Access: RSCFDnCFDGERFL register can be read or written in 32-bit units
RSCFDnCFDGERFLL, RSCFDnCFDGERFLH register can be read or written in 16-bit units
RSCFDnCFDGERFLLL, RSCFDnCFDGERFLLH, RSCFDnCFDGERFLHL, RSCFDnCFDGERFLHH register can be read or written in 8-bit units

Address: RSCFDnCFDGERFL: <RSCFDn_base> + 0090_H

RSCFDnCFDGERFLL: <RSCFDn_base> + 0090_H, RSCFDnCFDGERFLH: <RSCFDn_base> + 0092_H

RSCFDnCFDGERFLLL: <RSCFDn_base> + 0090_H, RSCFDnCFDGERFLLH: <RSCFDn_base> + 0091_H, RSCFDnCFDGERFLHL: <RSCFDn_base> + 0092_H, RSCFDnCFDGERFLHH: <RSCFDn_base> + 0093_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EEF7	EEF6	EEF5	EEF4	EEF3	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MOWES	QMES	—	QOWES	CMPOF	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 23.26 RSCFDnCFDGERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	EEF7	ECC Error Flag for Channel 7 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
22	EEF6	ECC Error Flag for Channel 6 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
21	EEF5	ECC Error Flag for Channel 5 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
20	EEF4	ECC Error Flag for Channel 4 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
19	EEF3	ECC Error Flag for Channel 3 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
18	EEF2	ECC Error Flag for Channel 2 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
17	EEF1	ECC Error Flag for Channel 1 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
16	EEF0	ECC Error Flag for Channel 0 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MOWES	Message overwrite Error Status 0: Message overwrite Error not detected 1: Message overwrite Error detected

Table 23.26 RSCFDnCFDGERFL Register Contents (2/2)

Bit Position	Bit Name	Function
6	QMES	TXQ Message Lost Error Status 0: TXQ Message lost Error not detected 1: TXQ Message lost Error detected
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	QOWES	TXQ Message overwrite Error Status 0: TXQ Message overwrite Error not detected 1: TXQ Message overwrite Error detected
3	CMPOF	CANFD Message payload overflow Flag 0: CANFD message payload overflow not detected 1: CANFD message payload overflow detected
2	THLES	TX History List Entry Lost Error Status 0: TX History List Entry Lost Error not detected 1: TX History List Entry Lost Error detected
1	MES	Message Lost Error Status 0: Message lost Error not detected 1: Message lost Error detected
0	DEF	DLC Error Flag 0: DLC Error not detected 1: DLC Error detected

The Global Error Flag register indicates the detection of global errors.

RSCFDnCFDGERFL.EEFm

ECC Error Flag for Channel m.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP or GL_RESET mode.

Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

The bit is cleared by writing 0_B to it.
This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.MOWES

Message overwrite Error Status.

This bit is set automatically when GW mode and Transmit/Receive FIFO Message overwrite Error is detected.

This bit is cleared automatically when all Transmit/Receive FIFO Message overwrite flags are cleared.

This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.QMES

TXQ Message lost Error Status.

This bit is set automatically when TXQ Message lost Error is detected.

This bit is cleared automatically when all TXQ Message lost flags are cleared.

This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.QOWES

TXQ Message overwrite Error Status.

This bit is set automatically when TXQ Message overwrite Error is detected.

This bit is cleared automatically when all TXQ Message overwrite flags are cleared.

This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.CMPOF

CANFD message payload overflow Flag.

If this bit is set then a payload overflow has happened on at least one channel.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP or GL_RESET mode. Writing 1_B has no influence on the bit values.

This bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

The bit is cleared by writing 0_B to it.
Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.THLES

TX History List Entry Lost Error Status.

This bit is set automatically when a TX History List Entry Lost Error is detected.

This bit is cleared automatically when all TX History List Entry Lost flags are cleared.
This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.MES

Message Lost Error Status.

This bit is set automatically when a FIFO Message Lost Error is detected.

This bit is cleared automatically when all FIFO Message Lost flags are cleared.
This bit will be cleared automatically in GL_RESET mode.

RSCFDnCFDGERFL.DEF

DLC Error Flag.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP or GL_RESET mode. Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically when a DLC Error is detected in a received frame.

The bit is cleared by writing 0_B to it.
This bit will be cleared automatically in GL_RESET mode.

23.3.3.5 RSCFDnCFDGTSC — Global Timestamp Counter Register

Access: RSCFDnCFDGTSC register is a read-only register that can be read in 32-bit units.
RSCFDnCFDGTSCL, RSCFDnCFDGTSCH register is a read-only register that can be read in 16-bit units.

Address: RSCFDnCFDGTSC: <RSCFDn_base> + 0094_H

RSCFDnCFDGTSCL: <RSCFDn_base> + 0094_H, RSCFDnCFDGTSCH: <RSCFDn_base> + 0096_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.27 RSCFDnCFDGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	TS[15:0]	Timestamp Value

The Timestamp counter register stores the Timestamp based on the selected configuration.

RSCFDnCFDGTSC.TS[15:0]

Timestamp Value.

The proper incrementing of the Time Stamp counter cannot be guaranteed when moving to Halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

Users cannot write to these bits when the RS-CANFD module is in GL_RESET or GL_STOP mode.

These bits will be cleared automatically in GL_RESET mode.

23.3.3.6 RSCFDnCFDGTINTSTSs — Global Transmit Interrupt Status Register s

Access: RSCFDnCFDGTINTSTSs register is a read-only register that can be read in 32-bit units
RSCFDnCFDGTINTSTSsL, RSCFDnCFDGTINTSTSsH register is a read-only register that can be read in 16-bit units
RSCFDnCFDGTINTSTSsLL, RSCFDnCFDGTINTSTSsLH, RSCFDnCFDGTINTSTSsHL, RSCFDnCFDGTINTSTSsHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDGTINTSTSs: <RSCFDn_base> + 1300_H + (4_H × s)

RSCFDnCFDGTINTSTSsL: <RSCFDn_base> + 1300_H + (4_H × s),
RSCFDnCFDGTINTSTSsH: <RSCFDn_base> + 1300_H + (4_H × s)

RSCFDnCFDGTINTSTSsLL: <RSCFDn_base> + 1300_H + (4_H × s),
RSCFDnCFDGTINTSTSsLH: <RSCFDn_base> + 1300_H + (4_H × s),
RSCFDnCFDGTINTSTSsHL: <RSCFDn_base> + 1300_H + (4_H × s),
RSCFDnCFDGTINTSTSsHH: <RSCFDn_base> + 1300_H + (4_H × s)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CFOTIF3	TQOFIF3	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	CFOTIF2	TQOFIF2	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CFOTIF1	TQOFIF1	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	CFOTIF0	TQOFIF0	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.28 RSCFDnCFDGTINTSTSs Register Contents (1/3)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30	CFOTIF3	Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (3 + (s * 4)) 0: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag not set 1: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag set
29	TQOFIF3	TX Queue One Frame Transmission Interrupt Flag Channel (3 + (s * 4)) 0: Channel m TX Queue One Frame Transmission Interrupt flag not set 1: Channel m TX Queue One Frame Transmission Interrupt flag set
28	THIF3	TX History List Interrupt Channel (3 + (s * 4)) 0: Channel m TX History List Interrupt flag not set 1: Channel m TX History List Interrupt flag set
27	CFTIF3	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel (3 + (s * 4)) 0: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag not set 1: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag set
26	TQIF3	TX Queue Interrupt Flag Channel (3 + (s * 4)) 0: Channel m TX Queue Interrupt flag not set 1: Channel m TX Queue Interrupt flag set
25	TAIF3	TX Abort Interrupt Flag Channel (3 + (s * 4)) 0: Channel m TX abort Interrupt flag not set 1: Channel m TX abort Interrupt flag set
24	TSIF3	TX Successful Interrupt Flag Channel (3 + (s * 4)) 0: Channel m TX Successful completion Interrupt flag not set 1: Channel m TX Successful completion Interrupt flag set
23	Reserved	When read, the value after reset is returned.

Table 23.28 RSCFDnCFDGTINTSTSs Register Contents (2/3)

Bit Position	Bit Name	Function
22	CFOTIF2	Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (2 + (s * 4)) 0: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag not set 1: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag set
21	TQOFIF2	TX Queue One Frame Transmission Interrupt Flag Channel (2 + (s * 4)) 0: Channel m TX Queue One Frame Transmission Interrupt flag not set 1: Channel m TX Queue One Frame Transmission Interrupt flag set
20	THIF2	TX History List Interrupt Channel (2 + (s * 4)) 0: Channel m TX History List Interrupt flag not set 1: Channel m TX History List Interrupt flag set
19	CFTIF2	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel (2 + (s * 4)) 0: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag not set 1: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag set
18	TQIF2	TX Queue Interrupt Flag Channel (2 + (s * 4)) 0: Channel m TX Queue Interrupt flag not set 1: Channel m TX Queue Interrupt flag set
17	TAIF2	TX Abort Interrupt Flag Channel (2 + (s * 4)) 0: Channel m TX abort Interrupt flag not set 1: Channel m TX abort Interrupt flag set
16	TSIF2	TX Successful Interrupt Flag Channel (2 + (s * 4)) 0: Channel m TX Successful completion Interrupt flag not set 1: Channel m TX Successful completion Interrupt flag set
15	Reserved	When read, the value after reset is returned.
14	CFOTIF1	Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (1 + (s * 4)) 0: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag not set 1: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag set
13	TQOFIF1	TX Queue One Frame Transmission Interrupt Flag Channel (1 + (s * 4)) 0: Channel m TX Queue One Frame Transmission Interrupt flag not set 1: Channel m TX Queue One Frame Transmission Interrupt flag set
12	THIF1	TX History List Interrupt Channel (1 + (s * 4)) 0: Channel m TX History List Interrupt flag not set 1: Channel m TX History List Interrupt flag set
11	CFTIF1	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel (1 + (s * 4)) 0: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag not set 1: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag set
10	TQIF1	TX Queue Interrupt Flag Channel (1 + (s * 4)) 0: Channel m TX Queue Interrupt flag not set 1: Channel m TX Queue Interrupt flag set
9	TAIF1	TX Abort Interrupt Flag Channel (1 + (s * 4)) 0: Channel m TX abort Interrupt flag not set 1: Channel m TX abort Interrupt flag set
8	TSIF1	TX Successful Interrupt Flag Channel (1 + (s * 4)) 0: Channel m TX Successful completion Interrupt flag not set 1: Channel m TX Successful completion Interrupt flag set
7	Reserved	When read, the value after reset is returned.
6	CFOTIF0	Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (0 + (s * 4)) 0: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag not set 1: Channel m Transmit/Receive FIFO One Frame Transmission Interrupt flag set

Table 23.28 RSCFDnCFDGTINTSTSs Register Contents (3/3)

Bit Position	Bit Name	Function
5	TQOFIF0	TX Queue One Frame Transmission Interrupt Flag Channel (0 + (s * 4)) 0: Channel m TX Queue One Frame Transmission Interrupt flag not set 1: Channel m TX Queue One Frame Transmission Interrupt flag set
4	THIF0	TX History List Interrupt Channel (0 + (s * 4)) 0: Channel m TX History List Interrupt flag not set 1: Channel m TX History List Interrupt flag set
3	CFTIF0	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel (0 + (s * 4)) 0: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag not set 1: Channel m Transmit/Receive FIFO TX/GW mode Interrupt flag set
2	TQIF0	TX Queue Interrupt Flag Channel (0 + (s * 4)) 0: Channel m TX Queue Interrupt flag not set 1: Channel m TX Queue Interrupt flag set
1	TAIF0	TX Abort Interrupt Flag Channel (0 + (s * 4)) 0: Channel m TX abort Interrupt flag not set 1: Channel m TX abort Interrupt flag set
0	TSIF0	TX Successful Interrupt Flag Channel (0 + (s * 4)) 0: Channel m TX Successful completion Interrupt flag not set 1: Channel m TX Successful completion Interrupt flag set

(s) can be calculated from the desired Interrupt Channel (m) status flags using the formula s = floor(m / 4)

Byte position can be calculated using the formula (m – (s * 4))

e.g. for Interrupt Channel 7 Status Flags we have:

m = 7, s = floor(7 / 4) = 1 and Byte position = (7 – (1 * 4)) = 3

Therefore, Users should read the Byte 3 (4^{th} Byte) of RSCFDnCFDGTINTSTS1.

RSCFDnCFDGTINTSTSs.CFOTIFo

Transmit/Receive FIFO One Frame Transmission Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.CFOTIFo Bit (Transmit/Receive FIFO One Frame Transmission Interrupt flag Channel m)

This bit is set automatically when the related Transmit/Receive FIFO One Frame Transmission Interrupt Flag (RSCFDnCFDCFSTSk.CFOFTXIF) is set when the Interrupt is enabled.

This bit is cleared automatically when related Transmit/Receive FIFO One Frame Transmission Interrupt Flag (RSCFDnCFDCFSTSk.CFOFTXIF) is cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.TQOFIFo

TX Queue One Frame Transmission Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.TQOFIFo Bit (TX Queue One Frame Transmission Interrupt Flag Channel m)

This bit is set automatically when the related TX Queue One Frame TX Interrupt flag of the related channel is set when the Interrupt is enabled.

This bit is cleared automatically when related TX Queue One Frame TX Interrupt flag is cleared or the

Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.THIFo

TX History List Interrupt Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.THIFo Bit (TX History List Interrupt flag Channel m)

This bit is set automatically when the related TX History List Interrupt Flag (RSCFDnCFDTHLSTSm.THLIF) is set when the Interrupt is enabled.

This bit is cleared automatically when related TX History List Interrupt Flag (RSCFDnCFDTHLSTSm.THLIF) is cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.CFTIFo

Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.CFTIFo Bit (Transmit/Receive FIFO in TX/GW mode Interrupt flag Channel m)

This bit is set automatically when the related Common TX/GW FIFO Interrupt Flag (RSCFDnCFDCFSTSk.CFTXIF) is set when the Interrupt is enabled.

This bit is cleared automatically when related Common TX/GW FIFO Interrupt Flag is cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.TQIFo

TX Queue Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.TQIFo Bit (TX Queue Interrupt Flag Channel m)

This bit is set automatically when the related TX Queue Interrupt flag of the related channel is set when the Interrupt is enabled.

This bit is cleared automatically when related TX Queue Interrupt flag is cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.TAIFo

TX Abort Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.TAIFo Bit (TX Abort Interrupt Flag Channel m)

This bit is set automatically when abort Successful flag of the related channel is set when the Interrupt is enabled.

This bit is cleared automatically when the related TX MB Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGTINTSTSs.TSIFo

TX Successful Interrupt Flag Channel (o + (s * 4))

RSCFDnCFDGTINTSTSs.TSIFo Bit (TX Successful Completion Interrupt Flag Channel m)

This bit is set automatically when the Transmission Successful flag of the related channel is set when the Interrupt is enabled.

This bit is cleared automatically when related TX MB Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

23.3.3.7 RSCFDnCFDGRINTSTSm — Global Receive Interrupt Status Register m

Access: RSCFDnCFDGRINTSTSm register is a read-only register that can be read in 32-bit units
RSCFDnCFDGRINTSTSmL, RSCFDnCFDGRINTSTSmH register is a read-only register that can be read in 16-bit units
RSCFDnCFDGRINTSTSmLL, RSCFDnCFDGRINTSTSmLH, RSCFDnCFDGRINTSTSmHL, RSCFDnCFDGRINTSTSmHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDGRINTSTSm: <RSCFDn_base> + 1350_H + (04_H × m)

RSCFDnCFDGRINTSTSmL: <RSCFDn_base> + 1350_H + (04_H × m),
RSCFDnCFDGRINTSTSmH: <RSCFDn_base> + 1352_H + (04_H × m)

RSCFDnCFDGRINTSTSmLL: <RSCFDn_base> + 1350_H + (04_H × m),
RSCFDnCFDGRINTSTSmLH: <RSCFDn_base> + 1351_H + (04_H × m),
RSCFDnCFDGRINTSTSmHL: <RSCFDn_base> + 1352_H + (04_H × m),
RSCFDnCFDGRINTSTSmHH: <RSCFDn_base> + 1353_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CFOFRIF[2:0]			—	CFRFIF[2:0]			—	—	—	—	—	CFRIF[2:0]		
Value after reset	R/W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	BQOFRIF[1:0]		—	QOFRIF[2:0]			—	—	BQFIF[1:0]		—	QFIF[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.29 RSCFDnCFDGRINTSTSm Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 28	CFOFRIF[2:0]	Transmit/Receive FIFO One Frame RX Interrupt Flag Channel m 0: Corresponding Transmit/Receive FIFO One Frame RX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO One Frame RX interrupt flag is set
27	Reserved	When read, the value after reset is returned.
26 to 24	CFRFIF[2:0]	Transmit/Receive FIFO FDC level Full Interrupt Flag Channel m 0: Corresponding Transmit/Receive FIFO Full interrupt flag is not set 1: Corresponding Transmit/Receive FIFO Full interrupt flag is set
23 to 19	Reserved	When read, the value after reset is returned.
18 to 16	CFRIF[2:0]	Transmit/Receive FIFO RX Interrupt Flag Channel m 0: Corresponding Transmit/Receive FIFO RX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO RX interrupt flag is set
15, 14	Reserved	When read, the value after reset is returned.
13, 12	BQOFRIF[1:0]	Borrowed TXQ One Frame RX Interrupt Flag Channel m 0: Corresponding TXQ One Frame RX interrupt flag is not set 1: Corresponding TXQ One Frame RX interrupt flag is set
11	Reserved	When read, the value after reset is returned.
10 to 8	QOFRIF[2:0]	TXQ One Frame RX Interrupt Flag Channel m 0: Corresponding TXQ One Frame RX interrupt flag is not set 1: Corresponding TXQ One Frame RX interrupt flag is set
7, 6	Reserved	When read, the value after reset is returned.
5, 4	BQFIF[1:0]	Borrowed TXQ Full Interrupt Flag Channel m 0: Corresponding TXQ Full interrupt flag is not set 1: Corresponding TXQ Full interrupt flag is set
3	Reserved	When read, the value after reset is returned.

Table 23.29 RSCFDnCFDGRINTSTSm Register Contents (2/2)

Bit Position	Bit Name	Function
2 to 0	QFIF[2:0]	TXQ Full Interrupt Flag Channel m 0: Corresponding TXQ Full interrupt flag is not set 1: Corresponding TXQ Full interrupt flag is set

RSCFDnCFDGRINTSTSm.CFOFRIF[2:0]

Transmit/Receive FIFO One Frame RX Interrupt Flag Channel m.

This bit is set automatically when the Transmit/Receive FIFO One Frame RX Interrupt flag of the related channel is set when the Interrupt is enabled.
This bit is cleared automatically when related Transmit/Receive FIFO RX Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGRINTSTSm.CFRFIF[2:0]

Transmit/Receive FIFO FDC level full Interrupt Flag Channel m.

This bit is set automatically when the Transmit/Receive FIFO Full Interrupt flag of the related channel is set when the Interrupt is enabled.
This bit is cleared automatically when related Transmit/Receive FIFO RX Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGRINTSTSm.CFRIF[2:0]

Transmit/Receive FIFO RX Interrupt Flag Channel m.

This bit is set automatically when the Transmit/Receive FIFO RX Interrupt flag of the related channel is set when the Interrupt is enabled.
This bit is cleared automatically when related Transmit/Receive FIFO RX Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGRINTSTSm.BQOFRIF[1:0]

Borrowed TXQ One Frame RX Interrupt Flag Channel m.

This bit is set, when a Flexible transmission buffer assignment function is used and borrowed TXQ receives one frame. Operation is the same as RSCFDnCFDGRINTSTSm.QOFRIF.

This bit of the channel which lends TXMB is a reserve bit.

RSCFDnCFDGRINTSTSm.QOFRIF[2:0]

TXQ One Frame RX Interrupt Flag Channel m.

This bit is set automatically when the TXQ One Frame RX Interrupt flag of the related channel is set when the Interrupt is enabled.
This bit is cleared automatically when related TXQ Result status bits are cleared or the Interrupt enable is disabled.
This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

RSCFDnCFDGRINTSTSm.BQFIF[1:0]

Borrowed TXQ Full Interrupt Flag Channel m.

This bit is set, when a Flexible transmission buffer assignment function is used and borrowed TXQ is in full status. Operation is the same as RSCFDnCFDGRINTSTSm.QFIF.

This bit of the channel which lends TXMB is a reserve bit.

RSCFDnCFDGRINTSTSm.QFIF[2:0]

TXQ Full Interrupt Flag Channel m.

This bit is set automatically when the TXQ Full Interrupt flag of the related channel is set when the Interrupt is enabled.

This bit is cleared automatically when related TXQ Result status bits are cleared or the Interrupt enable is disabled.

This bit will be cleared automatically in GL_RESET mode or CH_RESET mode.

23.3.3.8 RSCFDnCFDGFDCFG — Global FD Configuration Register

Access: RSCFDnCFDGFDCFG register can be read or written in 32-bit units
RSCFDnCFDGFDCFGL, RSCFDnCFDGFDCFGH register can be read or written in 16-bit units
RSCFDnCFDGFDCFGLL, RSCFDnCFDGFDCFGLH, RSCFDnCFDGFDCFGHL, RSCFDnCFDGFDCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDGFDCFG: <RSCFDn_base> + 1314_H

RSCFDnCFDGFDCFGL: <RSCFDn_base> + 1314_H,
RSCFDnCFDGFDCFGH: <RSCFDn_base> + 1316_H

RSCFDnCFDGFDCFGLL: <RSCFDn_base> + 1314_H,
RSCFDnCFDGFDCFGLH: <RSCFDn_base> + 1315_H,
RSCFDnCFDGFDCFGHL: <RSCFDn_base> + 1316_H,
RSCFDnCFDGFDCFGHH: <RSCFDn_base> + 1317_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSCCFG		—	—	—	—	—	—	—	RPED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.30 RSCFDnCFDGFDCFG Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TSCCFG	Timestamp capture configuration 00: Timestamp capture at the sample point of SOF (start of frame). 01: Timestamp capture at frame valid indication. 10: Timestamp capture at the sample point of RES bit 11: Setting prohibited
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RPED	RES bit Protocol exception disable 0: Protocol exception event detection enabled. 1: Protocol exception event detection disabled.

RSCFDnCFDGFDCFG.TSCCFG

Timestamp capture configuration.

These bits configure the different capture points of the timestamp; for transmission and reception. When RSCFDnCFDGFDCFG.TSCCFG[1:0] = 10_B then the timestamp capture is done for CANFD frames at RES bit and for Classical frames at the start of frame.

Users can set these bits only when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGFDCFG.RPED

RES bit Protocol exception state disable.

This bit configures the protocol exception event handling according to ISO 11898-1 (2015). When this bit is enabled then the protocol exception event detection is disabled, and the protocol controller will transmit an error frame when the protocol exception event is detected (RES-bit is sampled recessive).

Users can set this bit only when RS-CANFD module is in GL_RESET mode.

23.3.3.9 RSCFDnCFDGRSTC — Global Reset Control Register

Access: RSCFDnCFDGRSTC register can be read or written in 32-bit units
RSCFDnCFDGRSTCL, RSCFDnCFDGRSTCH register can be read or written in 16-bit units

Address: RSCFDnCFDGRSTC: <RSCFDn_base> + 1380_H

RSCFDnCFDGRSTCL: <RSCFDn_base> + 1380_H,
RSCFDnCFDGRSTCH: <RSCFDn_base> + 1382_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY[7:0]								—	—	—	—	—	—	—	SRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	R	R	R	R	R	R/W

Table 23.31 RSCFDnCFDGRSTC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	KEY[7:0]	Key code These bits control the right or wrong of rewriting of a SRST bit.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRST	Software reset 0: Normal state. 1: Software reset state.

RSCFDnCFDGRSTC.KEY[7:0]

KEY code.

When $C4_H$ is written in these bits, the write of a RSCFDnCFDGRSTC.SRST bit becomes available.

Read value from these bits is always 00_H.

Users should write a CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit simultaneously.

RSCFDnCFDGRSTC.SRST

Software Reset

When this bit is set, RS-CANFD module will be in the same state as hardware reset.

When reset of IP is required, users should write 1 to this bit. Then user should write 0 to this bit.

When this bit is cleared, a RS-CANFD module is in GL_STOP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

RAM is not initialized when software reset is performed during the initialization of RAM.

Software needs to perform the initialization of RAM.

23.3.3.10 RSCFDnCFDGFCMC — Global Flexible CAN mode Configuration Register

Access: RSCFDnCFDGFCMC register can be read or written in 32-bit units
RSCFDnCFDGFCMCL, RSCFDnCFDGFCMCH register can be read or written in 16-bit units
RSCFDnCFDGFCMCLL, RSCFDnCFDGFCMCLH, RSCFDnCFDGFCMCHL, RSCFDnCFDGFCMCHH register can be read or written in 8-bit units

Address: RSCFDnCFDGFCMC: <RSCFDn_base> + 1384_H

RSCFDnCFDGFCMCL: <RSCFDn_base> + 1384_H,
RSCFDnCFDGFCMCH: <RSCFDn_base> + 1386_H

RSCFDnCFDGFCMCLL: <RSCFDn_base> + 1384_H,
RSCFDnCFDGFCMCLH: <RSCFDn_base> + 1385_H,
RSCFDnCFDGFCMCHL: <RSCFDn_base> + 1386_H,
RSCFDnCFDGFCMCHH: <RSCFDn_base> + 1387_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FLXC3	FLXC2	FLXC1	FLXC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 23.32 RSCFDnCFDGFCMC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FLXC3	Flexible CAN mode between Channel 7 and Channel 6 0: Normal mode. 1: Flexible CAN mode.
2	FLXC2	Flexible CAN mode between Channel 5 and Channel 4 0: Normal mode. 1: Flexible CAN mode.
1	FLXC1	Flexible CAN mode between Channel 3 and Channel 2 0: Normal mode. 1: Flexible CAN mode.
0	FLXC0	Flexible CAN mode between Channel 0 and Channel 1 0: Normal mode. 1: Flexible CAN mode.

Flexible transmission buffer assignment configured in RSCFDnCFDGFTBAC register and Flexible CAN mode configured in RSCFDnCFDGFCMC register should not be used simultaneously.

RSCFDnCFDGFCMC.FLXC3

Flexible CAN mode between Channel 6 and Channel 7.

When this bit is set, Channel 6 and Channel 7 of a RS-CANFD module are Flexible CAN mode.

Channel 7 uses TX/RX terminal of Channel 6. The TX/RX terminal of Channel 7 can not use.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGFCMC.FLXC2

Flexible CAN mode between Channel 4 and Channel 5.

When this bit is set, Channel 4 and Channel 5 of a RS-CANFD module are Flexible CAN mode.

Channel 5 uses TX/RX terminal of Channel 4. The TX/RX terminal of Channel 5 can not use.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGFCMC.FLXC1

Flexible CAN mode between Channel 2 and Channel 3.

When this bit is set, Channel 2 and Channel 3 of a RS-CANFD module are Flexible CAN mode.

Channel 3 uses TX/RX terminal of Channel 2. The TX/RX terminal of Channel 3 can not use.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGFCMC.FLXC0

Flexible CAN mode between Channel 0 and Channel 1.

When this bit is set, Channel 0 and Channel 1 of a RS-CANFD module are Flexible CAN mode.

Channel 1 uses TX/RX terminal of Channel 0. The TX/RX terminal of Channel 1 can not use.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

23.3.3.11 RSCFDnCFDGFTBAC — Global Flexible transmission buffer assignment Configuration Register

Access: RSCFDnCFDGFTBAC register can be read or written in 32-bit units
RSCFDnCFDGFTBACL, RSCFDnCFDGFTBACH register can be read or written in 16-bit units
RSCFDnCFDGFTBACLL, RSCFDnCFDGFTBACLH, RSCFDnCFDGFTBACHL, RSCFDnCFDGFTBACHH register can be read or written in 8-bit units

Address: RSCFDnCFDGFTBAC: <RSCFDn_base> + $138C_H$

RSCFDnCFDGFTBACL: <RSCFDn_base> + $138C_H$,
RSCFDnCFDGFTBACH: <RSCFDn_base> + $138F_H$

RSCFDnCFDGFTBACLL: <RSCFDn_base> + $138C_H$,
RSCFDnCFDGFTBACLH: <RSCFDn_base> + $138D_H$,
RSCFDnCFDGFTBACHL: <RSCFDn_base> + $138E_H$,
RSCFDnCFDGFTBACHH: <RSCFDn_base> + $138F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	FLXMB3[3:0]				—	—	—	—	FLXMB2[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FLXMB1[3:0]				—	—	—	—	FLXMB0[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 23.33 RSCFDnCFDGFTBAC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	FLXMB3[3:0]	Flexible transmission buffer assignment between Channel 6 and Channel 7 0000: 0 0001: 4 0010: 8 0011: 12 0100: 16 0101: 20 0110: 24 0111: 28 1000: 32 Other than the above: Setting prohibited By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	FLXMB2[3:0]	Flexible transmission buffer assignment between Channel 4 and Channel 5 0000: 0 0001: 4 0010: 8 0011: 12 0100: 16 0101: 20 0110: 24 0111: 28 1000: 32 Other than the above: Setting prohibited By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel.

Table 23.33 RSCFDnCFDGFTBAC Register Contents (2/2)

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	FLXMB1[3:0]	Flexible transmission buffer assignment between Channel 2 and Channel 3 0000: 0 0001: 4 0010: 8 0011: 12 0100: 16 0101: 20 0110: 24 0111: 28 1000: 32 Other than the above: Setting prohibited By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	FLXMB0[3:0]	Flexible transmission buffer assignment between Channel 0 and Channel 1 0000: 0 0001: 4 0010: 8 0011: 12 0100: 16 0101: 20 0110: 24 0111: 28 1000: 32 Other than the above: Setting prohibited By setting these bits the even channel can use the configured number of TX mailboxes of the odd channel.

RSCFDnCFDGFTBAC.FLXMBv[3:0]

Flexible transmission buffer assignment between Channel m-1 and Channel m.

Channel m-1 can use the number TXMB of Channel m from 0 to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] by the configuration of these bits.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

Flexible transmission buffer assignment configured in RSCFDnCFDGFTBAC register and Flexible CAN mode configured in RSCFDnCFDGFCMC register should not be used simultaneously. (m =1, 3, 5, 7)

23.3.4 Details of Acceptance Filter List-Related Registers

23.3.4.1 RSCFDnCFDGAFLECTR — Global Acceptance Filter List Entry Control Register

Access: RSCFDnCFDGAFLECTR register can be read or written in 32-bit units
RSCFDnCFDGAFLECTRL, RSCFDnCFDGAFLECTRH register can be read or written in 16-bit units
RSCFDnCFDGAFLECTRLL, RSCFDnCFDGAFLECTRLH, RSCFDnCFDGAFLECTRHL,
RSCFDnCFDGAFLECTRHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLECTR: <RSCFDn_base> + 0098_H

RSCFDnCFDGAFLECTRL: <RSCFDn_base> + 0098_H, RSCFDnCFDGAFLECTRH: <RSCFDn_base> + $009A_H$

RSCFDnCFDGAFLECTRLL: <RSCFDn_base> + 0098_H, RSCFDnCFDGAFLECTRLH: <RSCFDn_base> + 0099_H,
RSCFDnCFDGAFLECTRHL: <RSCFDn_base> + $009A_H$, RSCFDnCFDGAFLECTRHH: <RSCFDn_base> + $009B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	AFLPN[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.34 RSCFDnCFDGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List Data access disabled 1: Acceptance Filter List Data access enabled
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	AFLPN[6:0]	Acceptance Filter List Page Number Selected an Acceptance Filter List Page

This register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

RSCFDnCFDGAFLECTR.AFLDAE

Acceptance Filter List Data Access Enable.

This bit prevents Acceptance Filter List write access if cleared after configuration of the Acceptance Filter List.
Users can read data from Acceptance Filter List independent of the status of this bit.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

This bit should be set to enable write access to Acceptance Filter List.

RSCFDnCFDGAFLECTR.AFLPN[6:0]

Acceptance Filter List Page Number

These bits select the Page Number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/Write accesses to the Acceptance Filter List can only be performed via a fixed window.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

Users should enter, only the values between 0 and 95 ($5F_H$) inclusive.

23.3.4.2 RSCFDnCFDGAFLCFGv — Global Acceptance Filter List Configuration Register v

Access: RSCFDnCFDGAFLCFGv register can be read or written in 32-bit units
RSCFDnCFDGAFLCFGvL, RSCFDnCFDGAFLCFGvH register can be read or written in 16-bit units
RSCFDnCFDGAFLCFGvLL, RSCFDnCFDGAFLCFGvLH, RSCFDnCFDGAFLCFGvHL,
RSCFDnCFDGAFLCFGvHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLCFGv: <RSCFDn_base> + $009C_H$ + (04_H × v),

RSCFDnCFDGAFLCFGvL: <RSCFDn_base> + $009C_H$ + (04_H × v), RSCFDnCFDGAFLCFGvH: <RSCFDn_base> + $009E_H$ + (04_H × v),

RSCFDnCFDGAFLCFGvLL: <RSCFDn_base> + $009C_H$ + (04_H × v), RSCFDnCFDGAFLCFGvLH: <RSCFDn_base> + $009D_H$ + (04_H × v), RSCFDnCFDGAFLCFGvHL: <RSCFDn_base> + $009E_H$ + (04_H × v), RSCFDnCFDGAFLCFGvHH: <RSCFDn_base> + $009F_H$ + (04_H × v)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNC(0 + v*2) [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RNC(1 + v*2) [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.35 RSCFDnCFDGAFLCFGv Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 16	RNC(0 + v*2) [8:0]	Rule Number for Channel (0 +v*2)
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	RNC(1 + v*2) [8:0]	Rule Number for Channel (1 +v*2)

Below is an example to calculate index for U2A16:

(no_of_channels = no_of_RNs = 8)

(no_of_RNs_per_RSCFDnCFDGAFLCFG = 2)

(no_of_RSCFDnCFDGAFLCFGs = ceil(no_of_RNs / no_of_RNs_per_RSCFDnCFDGAFLCFG) = ceil(8 / 2) = 4)

v = [0...no_of_RSCFDnCFDGAFLCFGs-1]

(v) can be calculated from the desired Channel (m) Rule Number using the formula v = floor (m / 2)

Word position can be calculated using the formula: (1 – m) + (v * 2)

e.g. for Channel 7 Rule Number we have:
m = 7, v = floor(7/2) = 3 and Word position = ((1 – 7) + (3 * 2)) = 0
Therefore, Users should read the Word 0 (First Word of RSCFDnCFDGAFLCFG3.

This register is used to define the number of Acceptance Filter List Entries (called “Rules”) applicable for channels 0 to 7 in the Acceptance Filter List.
The total number of available entries in the Acceptance Filter List is 192 * (m+1) (e.g. 1536 for 8 CAN channels). However, the filters can be allocated flexibly to the different channels depending on

requirements as long as both of the following conditions are satisfied:
the maximum number of Acceptance Filter per channel is 384 and the total number of rules defined for all channels is not exceeding the number of available entries in the Acceptance Filter List.

RSCFDnCFDGAFLCFGv.RNC(0 + v*2) [8:0]

Rule Number for Channel (0 + v*2)

These bits define the number of rules in the Acceptance Filter List for channel (0 + v*2).

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDGAFLCFGv.RNC(1 + v*2) [8:0]

Rule Number for Channel (1 + v*2)

These bits define the number of rules in the Acceptance Filter List for channel (1 + v*2).

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

23.3.4.3 RSCFDnCFDGAFLIDj — Global Acceptance Filter List ID Register j

Access: RSCFDnCFDGAFLIDj register can be read or written in 32-bit units
RSCFDnCFDGAFLIDjL, RSCFDnCFDGAFLIDjH register can be read or written in 16-bit units
RSCFDnCFDGAFLIDjLL, RSCFDnCFDGAFLIDjLH, RSCFDnCFDGAFLIDjHL, RSCFDnCFDGAFLIDjHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLIDj: <RSCFDn_base> + 1800_H + (10_H × j)

RSCFDnCFDGAFLIDjL: <RSCFDn_base> + 1800_H + (10_H × j),
RSCFDnCFDGAFLIDjH: <RSCFDn_base> + 1802_H + (10_H × j)

RSCFDnCFDGAFLIDjLL: <RSCFDn_base> + 1800_H + (10_H × j),
RSCFDnCFDGAFLIDjLH: <RSCFDn_base> + 1801_H + (10_H × j),
RSCFDnCFDGAFLIDjHL: <RSCFDn_base> + 1802_H + (10_H × j),
RSCFDnCFDGAFLIDjHH: <RSCFDn_base> + 1803_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.36 RSCFDnCFDGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard Identifier of Rule entry ID is valid for acceptance filtering 1: Extended Identifier of Rule entry ID is valid for acceptance filtering
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data Frame 1: Remote Frame
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering has attribute 'RX' 1: Global Acceptance Filter List entry ID for acceptance filtering has attribute 'TX'
28 to 0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry

These registers are used to configure the ID field of the Rule Entries in the Global Acceptance Filter List.

RSCFDnCFDGAFLIDj.GAFLIDE

Global Acceptance Filter List Entry IDE Field

This bit allows the configuration of the ID format (Standard ID or Extended ID) for each of the Global Acceptance Filter List entry. For each Rule entry of the related CAN channel the Acceptance filter process compares this bit against the IDE bit of the received CAN message.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLIDj.GAFLRTR

Global Acceptance Filter List Entry RTR Field

This bit allows the configuration of the specified frame format (Data Frame or Remote Frame) for each Global Acceptance Filter List entry. For each Rule entry in a CAN channel the Acceptance filter process compares this bit against the RTR bit of the received CAN message.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLIDj.GAFLLB

Global Acceptance Filter List Entry Loopback Configuration

This bit selects if the Global Acceptance Filter List entry gets the attribute ‘RX’ or ‘TX’. This attribute decides about the validity of the entry in the mirror mode case, loopback test mode case and during standard (nonloopback) reception. See **Table 23.145** for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, type of loopback mode and RX/TX attribute.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLIDj.GAFLID[28:0]

Global Acceptance Filter List Entry ID Field

These bits represent the CAN Identifier (ID) field of each of the Global Acceptance Filter List entry. Acceptance filter process compares this field against the ID of a received CAN message.

For alignment of these bits in standard and extended frame format, see **Section 23.4.3**.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.4.4 RSCFDnCFDGAFLMj — Global Acceptance Filter List Mask Register j

Access: RSCFDnCFDGAFLMj register can be read or written in 32-bit units
RSCFDnCFDGAFLMjL, RSCFDnCFDGAFLMjH register can be read or written in 16-bit units
RSCFDnCFDGAFLMjLL, RSCFDnCFDGAFLMjLH, RSCFDnCFDGAFLMjHL, RSCFDnCFDGAFLMjHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLMj: <RSCFDn_base> + 1804_H + (10_H × j)

RSCFDnCFDGAFLMjL: <RSCFDn_base> + 1804_H + (10_H × j),
RSCFDnCFDGAFLMjH: <RSCFDn_base> + 1806_H + (10_H × j)

RSCFDnCFDGAFLMjLL: <RSCFDn_base> + 1804_H + (10_H × j),
RSCFDnCFDGAFLMjLH: <RSCFDn_base> + 1805_H + (10_H × j),
RSCFDnCFDGAFLMjHL: <RSCFDn_base> + 1806_H + (10_H × j),
RSCFDnCFDGAFLMjHH: <RSCFDn_base> + 1807_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.37 RSCFDnCFDGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not considered for ID matching 1: IDE bit is considered for ID matching
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not considered for ID matching 1: RTR bit is considered for ID matching
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List Information Label bit 1
28 to 0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field bits

The Global Rule Mask entry registers are used to configure the Mask field of each Rule Entries in the Global Acceptance Filter List.

RSCFDnCFDGAFLMj.GAFLIDEM

Global Acceptance Filter List IDE Mask

This bit allows the configuration of the IDE mask bit for each Global Acceptance Filter List entry.
When IDE mask bit is 0_B, then the ID comparison depends upon the received IDE bit.
If received IDE bit is 0_B, then STD-ID comparison takes place.
If received IDE bit is 1_B, then EXT-ID comparison takes place.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLMj.GAFLRTRM

Global Acceptance Filter List Entry RTR Mask

This bit allows the configuration of the RTR mask bit for each Global Acceptance Filter List entry.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLMj.GAFLIFL1

Global Acceptance Filter List Information Label 1

These bits allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a MSB bit of an information label.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field [1] (RSCFDnCFDRMFDSTS.RMIFL [1], RSCFDnCFDRFFDSTS.RFIFL [1], RSCFDnCFDCFFDCSTS.CFIFL [1]) of the storage location of an incoming message.

This bit is stored in RSCFDnCFDTHLACC1m.TIFL [1] when RSCFDnCFDTHLCCm.THLDGE=1 is set up using GW function.

RSCFDnCFDGAFLMj.GAFLIDM[28:0]

Global Acceptance Filter List ID Mask Field

These bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0_B	Corresponding STD-ID / EXT-ID bit is not considered for ID matching
1_B	Corresponding STD-ID / EXT-ID bit is considered for ID matching

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.4.5 RSCFDnCFDGAFLP0j — Global Acceptance Filter List Pointer 0 Register j

Access: RSCFDnCFDGAFLP0j register can be read or written in 32-bit units
RSCFDnCFDGAFLP0jL, RSCFDnCFDGAFLP0jH register can be read or written in 16-bit units
RSCFDnCFDGAFLP0jLL, RSCFDnCFDGAFLP0jLH, RSCFDnCFDGAFLP0jHL, RSCFDnCFDGAFLP0jHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLP0j: <RSCFDn_base> + 1808_H + (10_H × j)

RSCFDnCFDGAFLP0jL: <RSCFDn_base> + 1808_H + (10_H × j),
RSCFDnCFDGAFLP0jH: <RSCFDn_base> + $180A_H$ + (10_H × j)

RSCFDnCFDGAFLP0jLL: <RSCFDn_base> + 1808_H + (10_H × j),
RSCFDnCFDGAFLP0jLH: <RSCFDn_base> + 1809_H + (10_H × j),
RSCFDnCFDGAFLP0jHL: <RSCFDn_base> + $180A_H$ + (10_H × j),
RSCFDnCFDGAFLP0jHH: <RSCFDn_base> + $180B_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							GAFLIFL0	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLDLC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.38 RSCFDnCFDGAFLP0j Register Contents

Bit Position	Bit Name	Function
31 to 16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer Field Global Acceptance Filter List Pointer
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Global Acceptance Filter List Single Message Buffer Direction Pointer is invalid 1: Global Acceptance Filter List Single Message Buffer Direction Pointer is valid
14 to 8	GAFLRMDP[6:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX Message Buffer number for storage of received messages
7	GAFLIFL0	Global Acceptance Filter List Information Label 0 Global Acceptance Filter List Information Label bit 0
6	GAFLSRD2	Global Acceptance Filter List Select Routing destination 2 0: Routing target is CFIFO2 1: Routing target is TX Queue 2 instead of CFIFO2
5	GAFLSRD1	Global Acceptance Filter List Select Routing destination 1 0: Routing target is CFIFO1 1: Routing target is TX Queue 1 instead of CFIFO1
4	GAFLSRD0	Global Acceptance Filter List Select Routing destination 0 0: Routing target is CFIFO0 1: Routing target is TX Queue 0 instead of CFIFO0
3 to 0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum no. of Data Bytes in a Data Frame required for its acceptance

The Global Acceptance Filter List Pointer 0 registers are used to configure the DLC, SW Pointer, Single Message Buffer select and Message Buffer direction pointer for each Rule Entry in the Global Acceptance Filter List.

RSCFDnCFDGAFLP0j.GAFLPTR[15:0]

Global Acceptance Filter List Pointer Field

These bits allow the configuration of a 16-bit pointer that will be attached to a received message accepted by the related Global Acceptance Filter List entry. The Pointer will be added during message storage in the Message Buffer area and can be used by the application as support function. The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLP0j.GAFLRMV

Global Acceptance Filter List RX Message Buffer Valid

This bit allows the enabling/disabling of a single reception Message Buffer as the target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLP0j.GAFLRMDP[6:0]

Global Acceptance Filter List RX Message Buffer Direction Pointer

These bits allow the configuration of a single reception Message Buffer as the destination target for a received message that is passing the acceptance check of the related Global Acceptance Filter List entry.
The value entered is the single destination Message Buffer number.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.
RSCFDnCFDRMNB.NRXMB[7:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX Message Buffers. The value to be entered in RSCFDnCFDGAFLP0j.GAFLRMDP[6:0] bits should only be between 0_H and (RSCFDnCFDRMNB.NRXMB[7:0] - 1_H).
If RSCFDnCFDRMNB.NRXMB[7:0] = 0_H, then the GAFLRMV bit should be configured as 0_B.

RSCFDnCFDGAFLP0j.GAFLIFL0

Global Acceptance Filter List Information Label 0

These bits allow the configuration of a 2-bit Information label that will be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to the bit when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (RSCFDnCFDRMFDSTS.RMIFL[0], RSCFDnCFDRFFDSTS.RFIFL[0], RSCFDnCFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

This bit is stored in RSCFDnCFDTHLACC1m.TIFL[0] when RSCFDnCFDTHLCCm.THLDGE=1 is set up using GW function.

RSCFDnCFDGAFLP0j.GAFLSRD2

Global Acceptance Filter List Select Routing destination 2

This bit changes a copy destination to CFIFO2 or TXQ2 by routing.

If this bit is set as 1, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose TX Queue.

If this bit is cleared to 0, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose Transmit/ Receive FIFO.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to the bit when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLP0j.GAFLSRD1

Global Acceptance Filter List Select Routing destination 1

This bit changes a copy destination to CFIFO1 or TXQ1 by routing.

If this bit is set as 1, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose TX Queue.

If this bit is cleared to 0, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose Transmit/ Receive FIFO.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to the bit when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLP0j.GAFLSRD0

Global Acceptance Filter List Select Routing destination 0

This bit changes a copy destination to CFIFO0 or TXQ0 by routing.

If this bit is set as 1, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose TX Queue.

If this bit is cleared to 0, the preset value of RSCFDnCFDGAFLP1j.GAFLFDP will choose Transmit/ Receive FIFO.

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to the bit when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

RSCFDnCFDGAFLP0j.GAFLDLC[3:0]

Global Acceptance Filter List DLC Field.

These bits allow the configuration of the minimum DLC (Data Length Code) value for a message to be accepted by the related Global Acceptance Filter List entry (automatic DLC filter function). DLC filter process is only passed if the DLC value of the message accepted by a Global Acceptance Filter List entry is equal or higher than the DLC value configured for this related Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding Rule Entry when this field is set to 0_H.

Following binary values can be configured:

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	0	0	0	DLC of received message = 0 or more (DLC Filter check is disabled)
CAN and CANFD	0	0	0	1	DLC of received message = 1 or more
CAN and CANFD	0	0	1	0	DLC of received message = 2 or more
CAN and CANFD	0	0	1	1	DLC of received message = 3 or more
CAN and CANFD	0	1	0	0	DLC of received message = 4 or more
CAN and CANFD	0	1	0	1	DLC of received message = 5 or more
CAN and CANFD	0	1	1	0	DLC of received message = 6 or more
CAN and CANFD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CANFD	1	0	0	0	DLC of received message = 8 or more
CANFD	1	0	0	1	DLC of received message = 12 or more
CANFD	1	0	1	0	DLC of received message = 16 or more
CANFD	1	0	1	1	DLC of received message = 20 or more
CANFD	1	1	0	0	DLC of received message = 24 or more
CANFD	1	1	0	1	DLC of received message = 32 or more
CANFD	1	1	1	0	DLC of received message = 48 or more
CANFD	1	1	1	1	DLC of received message = 64

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.

23.3.4.6 RSCFDnCFDGAFLP1j — Global Acceptance Filter List Pointer 1 Register j

Access: RSCFDnCFDGAFLP1j register can be read or written in 32-bit units
RSCFDnCFDGAFLP1jL, RSCFDnCFDGAFLP1jH register can be read or written in 16-bit units
RSCFDnCFDGAFLP1jLL, RSCFDnCFDGAFLP1jLH, RSCFDnCFDGAFLP1jHL, RSCFDnCFDGAFLP1jHH register can be read or written in 8-bit units

Address: RSCFDnCFDGAFLP1j: <RSCFDn_base> + $180C_H$ + (10_H × j)

RSCFDnCFDGAFLP1jL: <RSCFDn_base> +$180C_H$ + (10_H × j),
RSCFDnCFDGAFLP1jH: <RSCFDn_base> + $180E_H$ + (10_H × j)

RSCFDnCFDGAFLP1jLL: <RSCFDn_base> + $180C_H$ + (10_H × j),
RSCFDnCFDGAFLP1jLH: <RSCFDn_base> + $180D_H$ + (10_H × j),
RSCFDnCFDGAFLP1jHL: <RSCFDn_base> + $180E_H$ + (10_H × j),
RSCFDnCFDGAFLP1jHH: <RSCFDn_base> + $180F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLFDP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.39 RSCFDnCFDGAFLP1j Register Contents

Bit Position	Bit Name	Function
31 to 0	GAFLFDP[31:0]	Global Acceptance Filter List FIFO Direction Pointer (GAFLFDP[((m+1)*3+7):0]) FIFO direction pointer bits for received message storage

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

RSCFDnCFDGAFLP1j.GAFLFDP[31:0]

Global Acceptance Filter List FIFO Direction Pointer (GAFLFDP[((m+1)*3+7):0])

These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the RSCFDnCFDGAFLP1j.GAFLFDP[31:0] is configuring a dedicated FIFO:

Bit	Value (Binary)	Function
31	0	Disable Transmit/Receive FIFO 23 and Channel 7 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 23 as target for reception GAFLSRD2=1:Enable Channel 7 TX Queue 2 as target for reception
30	0	Disable Transmit/Receive FIFO 22 and Channel 7 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 22 as target for reception GAFLSRD1=1:Enable Channel 7 TX Queue 1 as target for reception
29	0	Disable Transmit/Receive FIFO 21 and Channel 7 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 21 as target for reception GAFLSRD0=1:Enable Channel 7 TX Queue 0 as target for reception

Bit	Value (Binary)	Function
28	0	Disable Transmit/Receive FIFO 20 and Channel 6 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 20 as target for reception GAFLSRD2=1:Enable Channel 6 TX Queue 2 as target for reception
27	0	Disable Transmit/Receive FIFO 19 and Channel 6 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 19 as target for reception GAFLSRD1=1:Enable Channel 6 TX Queue 1 as target for reception
26	0	Disable Transmit/Receive FIFO 18 and Channel 6 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 18 as target for reception GAFLSRD0=1:Enable Channel 6 TX Queue 0 as target for reception
25	0	Disable Transmit/Receive FIFO 17 and Channel 5 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 17 as target for reception GAFLSRD2=1:Enable Channel 5 TX Queue 2 as target for reception
24	0	Disable Transmit/Receive FIFO 16 and Channel 5 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 16 as target for reception GAFLSRD1=1:Enable Channel 5 TX Queue 1 as target for reception
23	0	Disable Transmit/Receive FIFO 15 and Channel 5 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 15 as target for reception GAFLSRD0=1:Enable Channel 5 TX Queue 0 as target for reception
22	0	Disable Transmit/Receive FIFO 14 and Channel 4 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 14 as target for reception GAFLSRD2=1:Enable Channel 4 TX Queue 2 as target for reception
21	0	Disable Transmit/Receive FIFO 13 and Channel 4 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 13 as target for reception GAFLSRD1=1:Enable Channel 4 TX Queue 1 as target for reception
20	0	Disable Transmit/Receive FIFO 12 and Channel 4 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 12 as target for reception GAFLSRD0=1:Enable Channel 4 TX Queue 0 as target for reception
19	0	Disable Transmit/Receive FIFO 11 and Channel 3 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 11 as target for reception GAFLSRD2=1:Enable Channel 3 TX Queue 2 as target for reception
18	0	Disable Transmit/Receive FIFO 10 and Channel 3 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 10 as target for reception GAFLSRD1=1:Enable Channel 3 TX Queue 1 as target for reception
17	0	Disable Transmit/Receive FIFO 9 and Channel 3 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 9 as target for reception GAFLSRD0=1:Enable Channel 3 TX Queue 0 as target for reception
16	0	Disable Transmit/Receive FIFO 8 and Channel 2 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 8 as target for reception GAFLSRD2=1:Enable Channel 2 TX Queue 2 as target for reception

Bit	Value (Binary)	Function
15	0	Disable Transmit/Receive FIFO 7 and Channel 2 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 7 as target for reception GAFLSRD1=1:Enable Channel 2 TX Queue 1 as target for reception
14	0	Disable Transmit/Receive FIFO 6 and Channel 2 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 6 as target for reception GAFLSRD0=1:Enable Channel 2 TX Queue 0 as target for reception
13	0	Disable Transmit/Receive FIFO 5 and Channel 1 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 5 as target for reception GAFLSRD2=1:Enable Channel 1 TX Queue 2 as target for reception
12	0	Disable Transmit/Receive FIFO 4 and Channel 1 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 4 as target for reception GAFLSRD1=1:Enable Channel 1 TX Queue 1 as target for reception
11	0	Disable Transmit/Receive FIFO 3 and Channel 1 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 3 as target for reception GAFLSRD0=1:Enable Channel 1 TX Queue 0 as target for reception
10	0	Disable Transmit/Receive FIFO 2 and Channel 0 TX Queue 2 as target for reception
	1	GAFLSRD2=0:Enable Transmit/Receive FIFO 2 as target for reception GAFLSRD2=1:Enable Channel 0 TX Queue 2 as target for reception
9	0	Disable Transmit/Receive FIFO 1 and Channel 0 TX Queue 1 as target for reception
	1	GAFLSRD1=0:Enable Transmit/Receive FIFO 1 as target for reception GAFLSRD1=1:Enable Channel 0 TX Queue 1 as target for reception
8	0	Disable Transmit/Receive FIFO 0 and Channel 0 TX Queue 0 as target for reception
	1	GAFLSRD0=0:Enable Transmit/Receive FIFO 0 as target for reception GAFLSRD0=1:Enable Channel 0 TX Queue 0 as target for reception
7	0	Disable RX FIFO 7 as target for reception
	1	Enable RX FIFO 7 as target for reception
6	0	Disable RX FIFO 6 as target for reception
	1	Enable RX FIFO 6 as target for reception
5	0	Disable RX FIFO 5 as target for reception
	1	Enable RX FIFO 5 as target for reception
4	0	Disable RX FIFO 4 as target for reception
	1	Enable RX FIFO 4 as target for reception
3	0	Disable RX FIFO 3 as target for reception
	1	Enable RX FIFO 3 as target for reception
2	0	Disable RX FIFO 2 as target for reception
	1	Enable RX FIFO 2 as target for reception
1	0	Disable RX FIFO 1 as target for reception
	1	Enable RX FIFO 1 as target for reception
0	0	Disable RX FIFO 0 as target for reception
	1	Enable RX FIFO 0 as target for reception

Users cannot write to these bits when RSCFDnCFDGAFLECTR.AFLDAE bit is 0_B.
For storage in Transmit/Receive FIFO, target for reception can only be those Transmit/Receive FIFO Buffers that are configured as RX FIFO or GW FIFO.

For storage in TX Queue, when these TX Queue buffers of a target are in GW mode, it can do.

Users should only write to these bits when the related RS-CANFD channel is in CH_RESET or CH_HALT mode.
Users should only configure up to 8 destination FIFO Buffers or 7 destination FIFO Buffers plus one RX Message Buffer or 8 destination TX Queue Buffers or 7 destination TX Queue Buffers plus one RX Message Buffer.

Or a setup of a maximum of 8 destinations in all is possible at FIFO buffer and TX Queue buffer.

23.3.5 Details of Receive Buffer-Related Registers

23.3.5.1 RSCFDnCFDRMNB — Receive Message Buffer Number Register

Access: RSCFDnCFDRMNB register can be read or written in 32-bit units
RSCFDnCFDRMNBL, RSCFDnCFDRMNBH register can be read or written in 16-bit units
RSCFDnCFDRMNBLL, RSCFDnCFDRMNBLH, RSCFDnCFDRMNBHL, RSCFDnCFDRMNBHH register can be read or written in 8-bit units

Address: RSCFDnCFDRMNB: <RSCFDn_base> + $00AC_H$

RSCFDnCFDRMNBL: <RSCFDn_base> + $00AC_H$, RSCFDnCFDRMNBH: <RSCFDn_base> + $00AE_H$

RSCFDnCFDRMNBLL: <RSCFDn_base> + $00AC_H$, RSCFDnCFDRMNBLH: <RSCFDn_base> + $00AD_H$,
RSCFDnCFDRMNBHL: <RSCFDn_base> + $00AE_H$, RSCFDnCFDRMNBHH: <RSCFDn_base> + $00AF_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RMPLS[2:0]			NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.40 RSCFDnCFDRMNB Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	RMPLS[2:0]	Reception Message Buffer Payload Data Size 000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes
7 to 0	NRXMB[7:0]	Number of RX Message Buffers Used to define the number of RX Message Buffer

The RX Message Buffer Number register is used to configure the total number of RX Message Buffers allocated to all channels.

RSCFDnCFDRMNB.RMPLS[2:0]

Reception Message Buffer Payload Data Size.

These bits are used to configure the message buffer payload data size.

Users can only write to these bits when RS-CANFD module is in GL_RESET modes.

RSCFDnCFDRMNB.NRXMB[7:0]

Number of RX Message Buffers.

These bits are used to configure the number of RX Message Buffers.

Users can only write to these bits when RS-CANFD module is in GL_RESET modes.

Users should enter, only the values between 0 and ((m+1)*16) inclusive, with 0_H indicating that, no RX Message Buffer is allocated.

23.3.5.2 RSCFDnCFDRMNDt — Receive Message Buffer New Data Register t

Access: RSCFDnCFDRMNDt register can be read or written in 32-bit units
RSCFDnCFDRMNDtL, RSCFDnCFDRMNDtH register can be read or written in 16-bit units
RSCFDnCFDRMNDtLL, RSCFDnCFDRMNDtLH, RSCFDnCFDRMNDtHL, RSCFDnCFDRMNDtHH register can be read or written in 8-bit units

Address: RSCFDnCFDRMNDt: <RSCFDn_base> + $00B0_H$ + (04_H × t)

RSCFDnCFDRMNDtL: <RSCFDn_base> + $00B0_H$ + (04_H × t),
RSCFDnCFDRMNDtH: <RSCFDn_base> + $00B2_H$ + (04_H × t)

RSCFDnCFDRMNDtLL: <RSCFDn_base> + $00B0_H$ + (04_H × t),
RSCFDnCFDRMNDtLH: <RSCFDn_base> + $00B1_H$ + (04_H × t),
RSCFDnCFDRMNDtHL: <RSCFDn_base> + $00B2_H$ + (04_H × t),
RSCFDnCFDRMNDtHH: <RSCFDn_base> + $00B3_H$ + (04_H × t)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSu[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSu[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.41 RSCFDnCFDRMNDt Register Contents

Bit Position	Bit Name	Function
31 to 0	RMNSu[31:0]	RX Message Buffer New Data Status 0: New Data not stored in corresponding RX Message Buffer 1: New Data stored in corresponding RX Message Buffer

The RX Message Buffer New Data status register bits show the New Data storage status of the RX Message Buffers.

RSCFDnCFDRMNDt.RMNSu[31:0]

RX Message Buffer New Data Status

These bits show the NewData Flag status for the corresponding RX Message Buffer. RMNS bit 0 corresponds to RX Message Buffer 0 and so on.

Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(no_of_RSCFDnCFDRMBCPs_per_channel = No. of RX Message Buffer Components per Channel = 16)
(no_of_RSCFDnCFDRMBCPs = No. of RX Message Buffer Components = no_of_channels*no_of_RSCFDnCFDRMBCPs_per_channel = 8 * 16 = 128)
(no_of_bits_per_register = 32)
(no_of_RSCFDnCFDRMNDs = No. of RSCFDnCFDRMND Register = no_of_RSCFDnCFDRMBCPs/no_of_bits_per_register = 128 / 32 = 4)
(t = [0...no_of_RSCFDnCFDRMNDs-1])
(u = [t*32...(no_of_RSCFDnCFDRMBCPs – ((no_of_RSCFDnCFDRMNDs – 1 – t) * 32) – 1)])
(t) can be calculated from the desired NewData status flag (u) using the formula t = floor(u / 32)
Bit position can be calculated using the formula (u – (t * 32))
e.g. for NewData Status Flag 47 we have:

u = 47, t = floor(47 / 32) = 1 and bit position = (47 – (1 * 32)) = 15
Therefore, Users should read the bit 15 of RSCFDnCFDRMND1.
Users cannot write to these bits when RS-CANFD module is in GL_RESET or GL_STOP mode.
Writing 1_B has no influence on the bit values.

This bit cannot be cleared when message storage in the corresponding RX Message Buffer is in progress.

Do not use bit clear instruction for clearing these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

The field is set automatically when storage of a new message starts in the corresponding RX Message Buffer.

The duration of message storage time is 6 peripheral clock (pclk) cycles for RSCFDnCFDRMNB.RMPLS = 000_B (max 8 byte payload).

For RSCFDnCFDRMNB.RMPLS > 000_B it is 6 + 1 for each four byte (max 20 peripheral clock (pclk) cycles for 64 byte).

This bit is cleared by writing a 0_B to it.

This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.

23.3.6 Details of Receive FIFO Buffer-Related Registers

23.3.6.1 RSCFDnCFDRFCCx — Receive FIFO Configuration / Control Register x

Access: RSCFDnCFDRFCCx register can be read or written in 32-bit units
RSCFDnCFDRFCCxL, RSCFDnCFDRFCCxH register can be read or written in 16-bit units
RSCFDnCFDRFCCxLL, RSCFDnCFDRFCCxLH, RSCFDnCFDRFCCxHL, RSCFDnCFDRFCCxHH register can be read or written in 8-bit units

Address: RSCFDnCFDRFCCx: <RSCFDn_base> + $00C0_H$ + (04_H × x)

RSCFDnCFDRFCCxL: <RSCFDn_base> + $00C0_H$ + (04_H × x),
RSCFDnCFDRFCCxH: <RSCFDn_base> + $00C2_H$ + (04_H × x)

RSCFDnCFDRFCCxLL: <RSCFDn_base> + $00C0_H$ + (04_H × x),
RSCFDnCFDRFCCxLH: <RSCFDn_base> + $00C1_H$ + (04_H × x),
RSCFDnCFDRFCCxHL: <RSCFDn_base> + $00C2_H$ + (04_H × x),
RSCFDnCFDRFCCxHH: <RSCFDn_base> + $00C3_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Table 23.42 RSCFDnCFDRFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RFFIE	RX FIFO Full interrupt Enable 0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled
15 to 13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 000: Interrupt generated when FIFO is 1/8th full. 001: Interrupt generated when FIFO is 1/4th full. 010: Interrupt generated when FIFO is 3/8th full. 011: Interrupt generated when FIFO is 1/2 full. 100: Interrupt generated when FIFO is 5/8th full. 101: Interrupt generated when FIFO is 3/4th full. 110: Interrupt generated when FIFO is 7/8th full. 111: Interrupt generated when FIFO is full.
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	RFDC[2:0]	RX FIFO Depth Configuration 000: FIFO Depth = 0 messages 001: FIFO Depth = 4 messages 010: FIFO Depth = 8 messages 011: FIFO Depth = 16 messages 100: FIFO Depth = 32 messages 101: FIFO Depth = 48 messages 110: FIFO Depth = 64 messages 111: FIFO Depth = 128 messages

Table 23.42 RSCFDnCFDRFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	RFPLS[2:0]	RX FIFO Payload Data Size configuration 000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RFIE	RX FIFO Interrupt Enable 0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled

The RX FIFO Configuration / Control registers are used to configure and control the 8 RX FIFOs.

(x = RX FIFO index = [0…no_of_RFIFOs-1])

(no_of_RFIFOs = No. of RX FIFOs = 8)

RSCFDnCFDRFCCx.RFFIE

RX FIFO Full Interrupt Enable

This bit enables generation of the RXFIFO Full Interrupt when it is set.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

The following contents are imagined as how to use interruption.

1. Interruption output in number of arbitrary stages (RSCFDnCFDRFCCx.RFIGCV)

2. Interruption output in FIFO full state

Management of the receiving data of FIFO can be performed by these notices of interruption.

RSCFDnCFDRFCCx.RFIGCV[2:0]

RX FIFO Interrupt Generation Counter Value

These bits select the counter value of the FIFO for generation of FIFO Interrupt. These values represent fractions of the FIFO depth for which Interrupt is generated.

Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.

The setting of these bits should be synchronized with the RSCFDnCFDRFCCx.RFDC bits. Refer to **Section 23.7.2.1(5), FIFO Interrupt Configuration** for detailed information.
Users should only write to these bits when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDRFCCx.RFIM

RX FIFO Interrupt Mode

This bit selects the Interrupt generation condition for the FIFO.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDRFCCx.RFDC[2:0]

RX FIFO Depth Configuration

These bits select the depth of the FIFO in terms of number of Messages. If the FIFO depth is configured to 0 Messages then the FIFO cannot be used.

Users can only write to these bits when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDRFCCx.RFPLS[2:0]

Rx FIFO Payload Data Size configuration

These bits define the message data payload allocation in the RAM.
This is the max. number of Bytes which can be received by this FIFO.
Refer to **Section 23.7.2.1(4), FIFO Payload Size Configuration** for details.

Users can only write to these bits when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDRFCCx.RFIE

RX FIFO Interrupt Enable

This bit enables generation of the FIFO Interrupt when it is set.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

RSCFDnCFDRFCCx.RFE

RX FIFO Enable

This bit enables the FIFO when it is set. If this bit is cleared, the RX FIFO will be cleared and is empty.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING modes.

This bit can only be set if the configured FIFO depth is greater than 0_H (RSCFDnCFDRFCCx.RFDC > 000_B).

The RSCFDnCFDRFCCd.RFE bit should be set by a separate write access to the RSCFDnCFDRFCCd register, after all the other bits of the RSCFDnCFDRFCCd register have been set.

This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.

23.3.6.2 RSCFDnCFDRFSTSx — Receive FIFO Status Register x

Access: RSCFDnCFDRFSTSx register can be read or written in 32-bit units
RSCFDnCFDRFSTSxL, RSCFDnCFDRFSTSxH register can be read or written in 16-bit units
RSCFDnCFDRFSTSxLL, RSCFDnCFDRFSTSxLH, RSCFDnCFDRFSTSxHL, RSCFDnCFDRFSTSxHH register can be read or written in 8-bit units

Address: RSCFDnCFDRFSTSx: <RSCFDn_base> + $00E0_H$ + (04_H × x)

RSCFDnCFDRFSTSxL: <RSCFDn_base> + $00E0_H$ + (04_H × x),
RSCFDnCFDRFSTSxH: <RSCFDn_base> + $00E2_H$ + (04_H × x)

RSCFDnCFDRFSTSxLL: <RSCFDn_base> + $00E0_H$ + (04_H × x),
RSCFDnCFDRFSTSxLH: <RSCFDn_base> + $00E1_H$ + (04_H × x),
RSCFDnCFDRFSTSxHL: <RSCFDn_base> + $00E2_H$ + (04_H × x),
RSCFDnCFDRFSTSxHH: <RSCFDn_base> + $00E3_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R

Table 23.43 RSCFDnCFDRFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RFFIF	RX FIFO Full Interrupt Flag 0: FIFO Full interrupt condition not satisfied 1: FIFO Full interrupt condition satisfied
15 to 8	RFMC[7:0]	RX FIFO Message Counter Number of Messages stored in FIFO
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RFIF	RX FIFO Interrupt Flag 0: FIFO Interrupt condition not satisfied 1: FIFO Interrupt condition satisfied
2	RFMLT	RX FIFO Message Lost 0: NO Message Lost in FIFO 1: FIFO Message Lost
1	RFFLL	RX FIFO Full 0: FIFO Not Full 1: FIFO Full
0	RFEMP	RX FIFO Empty 0: FIFO Not Empty 1: FIFO Empty

The FIFO status registers show the status of the messages stored in corresponding FIFO Buffers.

(x = RX FIFO index = [0…no_of_RFIFOs-1])

(no_of_RFIFOs = No. of RX FIFOs = 8)

RSCFDnCFDRFSTSx.RFFIF

RX FIFO Full Interrupt Flag.

This bit will not be cleared automatically if the RX FIFO Buffer is disabled.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING modes.

Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically when the Interrupt condition is satisfied.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.

The bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDRFSTSx.RFMC[7:0]

RX FIFO Message Count.

These bits indicate the number of CAN messages stored in the RX FIFO that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled.

These bits are cleared automatically when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDRFSTSx.RFIF

RX FIFO Interrupt Flag.

This bit will not be cleared automatically if the RX FIFO Buffer is disabled.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING modes.
Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically when the configured Interrupt condition is satisfied.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDRFSTSx.RFMLT

RX FIFO Message Lost.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING modes.

Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO is already full.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDRFSTSx.RFFLL

RX FIFO Full.

This bit is set automatically when number of CAN messages stored in the FIFO matches the configured FIFO depth.

This bit is cleared automatically when the number of CAN messages stored in the FIFO is less than the configured FIFO depth.
This bit is cleared automatically when RX FIFO is disabled by setting the RSCFDnCFDRFCCx.RFE bit to 0_B.
This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDRFSTSx.RFEMP

RX FIFO Empty.

This bit is set automatically when RSCFDnCFDRFSTSx.RFMC is 0_H.
This bit is set automatically when RX FIFO is disabled by setting the RSCFDnCFDRFCCx.RFE bit to 0_B.
This bit is set automatically when RS-CANFD module enters GL_RESET mode.

This bit is cleared automatically when the first message is stored in the RX FIFO Buffer.

23.3.6.3 RSCFDnCFDRFPCTRx — Receive FIFO Pointer Control Register x

Access: RSCFDnCFDRFPCTRx register is a write-only register that can be written in 32-bit units
RSCFDnCFDRFPCTRxL, RSCFDnCFDRFPCTRxH register is a write-only register that can be written in 16-bit units
RSCFDnCFDRFPCTRxLL, RSCFDnCFDRFPCTRxLH, RSCFDnCFDRFPCTRxHL,
RSCFDnCFDRFPCTRxHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDRFPCTRx: <RSCFDn_base> + 0100_H + (04_H × x)

RSCFDnCFDRFPCTRxL: <RSCFDn_base> + 0100_H + (04_H × x),
RSCFDnCFDRFPCTRxH: <RSCFDn_base> + 0101_H + (04_H × x)

RSCFDnCFDRFPCTRxLL: <RSCFDn_base> + 100_H + (04_H × x),
RSCFDnCFDRFPCTRxLH: <RSCFDn_base> + 0101_H + (04_H × x),
RSCFDnCFDRFPCTRxHL: <RSCFDn_base> + 0102_H + (04_H × x),
RSCFDnCFDRFPCTRxHH: <RSCFDn_base> + 0103_H + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.44 RSCFDnCFDRFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO Buffers

These registers can be used to increment the Read Pointer of the corresponding RX FIFO Buffers.

(x = RX FIFO index = [0…no_of_RFIFOs-1])

(no_of_RFIFOs = No. of RX FIFOs = 8)

RSCFDnCFDRFPCTRx.RFPC[7:0]

RX FIFO Pointer Control

When the value FF_H is written to these bits, then the Pointer of the corresponding RX FIFO Buffer is moved to the next FIFO entry.

Read value from these bits is always 0_H.
Users can only write to these bits when RS-CANFD module is in GL_TEST or GL_OPERATING modes.

Users should only write FF_H to these registers when the corresponding RX FIFO is enabled and not empty.

Users should not write to the FIFO control registers when DMA is enabled.

23.3.7 Details of Transmit/Receive FIFO Buffer Related Registers

23.3.7.1 RSCFDnCFDCFCCk — Transmit/Receive FIFO Configuration/Control Register k

Access: RSCFDnCFDCFCCk register can be read or written in 32-bit units
RSCFDnCFDCFCCkL, RSCFDnCFDCFCCkH register can be read or written in 16-bit units
RSCFDnCFDCFCCkLL, RSCFDnCFDCFCCkLH, RSCFDnCFDCFCCkHL, RSCFDnCFDCFCCkHH register can be read or written in 8-bit units

Address: RSCFDnCFDCFCCk: <RSCFDn_base> + 0120_H + (04_H × k)

RSCFDnCFDCFCCkL: <RSCFDn_base> + 0120_H + (04_H × k),
RSCFDnCFDCFCCkH: <RSCFDn_base> + 0122_H + (04_H × k)

RSCFDnCFDCFCCkLL: <RSCFDn_base> + 0120_H + (04_H × k),
RSCFDnCFDCFCCkLH: <RSCFDn_base> + 0121_H + (04_H × k),
RSCFDnCFDCFCCkHL: <RSCFDn_base> + 0122_H + (04_H × k),
RSCFDnCFDCFCCkHH: <RSCFDn_base> + 0123_H + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFDC[2:0]			CFTML[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	CFITR	CFITSS	CFM[1:0]		—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 23.45 RSCFDnCFDCFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Transmit/Receive FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX or GW mode, delay is a multiple of basic Interval Timer Clock Source unit
23 to 21	CFDC[2:0]	Transmit/Receive Depth Configuration 000: FIFO Depth = 0 messages 001: FIFO Depth = 4 messages 010: FIFO Depth = 8 messages 011: FIFO Depth = 16 messages 100: FIFO Depth = 32 messages 101: FIFO Depth = 48 messages 110: FIFO Depth = 64 messages 111: FIFO Depth = 128 messages
20 to 16	CFTML[4:0]	Transmit/Receive FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Interrupt Generation Counter Value 000: Interrupt generated = when FIFO is 1/8th full. 001: Interrupt generated = when FIFO is 1/4th full. 010: Interrupt generated = when FIFO is 3/8th full. 011: Interrupt generated = when FIFO is 1/2 full. 100: Interrupt generated = when FIFO is 5/8th full. 101: Interrupt generated = when FIFO is 3/4th full. 110: Interrupt generated = when FIFO is 7/8th full. 111: Interrupt generated = when FIFO is full.

Table 23.45 RSCFDnCFDCFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Mode 0: RX FIFO Mode: RX Interrupt generated when Transmit/Receive FIFO counter reaches CFIGCV value from a lower value; TX FIFO Mode: TX Interrupt generated when Transmit/Receive FIFO transmits the last message successfully; GW FIFO Mode: For RX interrupt flag: Interrupt generated when FIFO counter increments and reaches the value configured in CFIGCV; For TX interrupt flag: Interrupt generated when FIFO transmits the last message successfully; 1: RX FIFO Mode: RX Interrupt generated at the end of every received message storage; TX FIFO Mode: Interrupt generated for every successfully transmitted message; GW FIFO Mode: For RX interrupt flag: Interrupt generated when a message is stored in the FIFO; For TX interrupt flag: Interrupt generated when a message is successfully transmitted from the FIFO;
11	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Reference Clock Period x1 1: Reference Clock Period x10
10	CFITSS	Transmit/Receive FIFO Interval Timer Source Select 0: Reference Clock (x1 / x10 period) 1: Bit Time Clock of related channel (FIFO is linked to fixed channel)
9, 8	CFM[1:0]	Transmit/Receive FIFO Mode 00: RX FIFO Mode 01: TX FIFO Mode 10: CAN – CAN GW FIFO Mode 11: Setting prohibited
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	CFPLS[2:0]	Transmit/Receive FIFO Payload Data size configuration 000: 8 bytes 001: 12 bytes 010: 16 bytes 011: 20 bytes 100: 24 bytes 101: 32 bytes 110: 48 bytes 111: 64 bytes
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CFTXIE	Transmit/Receive FIFO TX Interrupt Enable 0: FIFO Interrupt generation disabled for Frame TX 1: FIFO Interrupt generation enabled for Frame TX
1	CFRXIE	Transmit/Receive FIFO RX Interrupt Enable 0: FIFO Interrupt generation disabled for Frame RX 1: FIFO Interrupt generation enabled for Frame RX
0	CFE	Transmit/Receive FIFO Enable 0: FIFO disabled 1: FIFO enabled

The Transmit/Receive FIFO Configuration / Control registers are used to configure the Transmit/ Receive FIFOs.
Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(no_of_CFIFOs_per_channel = No. of Transmit/Receive FIFOs per Channel = 3)
Where the total number of CFIFOs = no_of_CFIFOs = no_of_CFIFOs_per_channel * no_of_channels = 3 * 8
= 24 as shown in **Figure 23.27**
(k = Transmit/Receive FIFO index = [0 .. no_of_CFIFOs -1])

RSCFDnCFDCFCCk.CFITT[7:0]

Transmit/Receive FIFO Interval Transmission Time.

These bits select the delay in the start of transmission for all messages transmitted from this FIFO when configured in TX or GW mode. The delay is a multiple of the basic Interval Timer Clock Source period (Reference Clock x1, Reference Clock x10 or Bit Time Clock of the related CAN channel).

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RSCFDnCFDCFCCk.CFE bit is set to 1_B.
For RSCFDnCFDGCFG.ITRCP[15:0] = 0_H these bits should only be set to 0_H.

RSCFDnCFDCFCCk.CFDC[2:0]

Transmit/Receive FIFO Depth Configuration.

These bits select the depth of the Transmit/Receive FIFO in terms of number of Messages. If the FIFO depth is configured to 0 Messages then the FIFO cannot be used.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFTML[4:0]

Transmit/Receive FIFO TX Message Buffer Link.

These bits select the normal transmit Message Buffer position where the TX or GW FIFO is linked to, for transmission scanning.

Users cannot write to these bits in GL_OPERATING & GL_STOP modes.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFIGCV[2:0]

Transmit/Receive FIFO Interrupt Generation Counter Value.

These bits select the message counter value for the generation of the FIFO Interrupt. These values represent fractions of the FIFO depth at which Interrupt is to be generated.

Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.

The setting of these bits should be synchronized with the RSCFDnCFDCFCCk.CFDC bits. Refer to **Section 23.7.2.1(5), FIFO Interrupt Configuration** for detailed information.
Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFIM

Transmit/Receive FIFO Interrupt Mode.

This bit selects the Interrupt generation condition for the FIFO.

Users cannot write to this bit in GL_STOP mode.

Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFITR

Transmit/Receive FIFO Interval Timer Resolution.

This bit selects the resolution of the Reference Clock for the Interval Transmission Timer (Peripheral Clock is the source for the Reference Clock).

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RSCFDnCFDCFCCk.CFE bit is set to 1_B.

RSCFDnCFDCFCCk.CFITSS

Transmit/Receive FIFO Interval Timer Source Select.

This bit selects the basic clock source for the Interval Transmission Timer.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RSCFDnCFDCFCCk.CFE bit is set to 1_B.

Users should not write 1_B when CANFD communication will be used.
Note bit time clock could be variable depending on the nominal and data rate bit configuration.

RSCFDnCFDCFCCk.CFM[1:0]

Transmit/Receive FIFO Mode.

These bits select the Mode of the FIFO. When HW Reset is active, all the Transmit/Receive FIFO Buffers will be configured in RX FIFO mode.

Users cannot write to these bits in GL_OPERATING & GL_STOP modes.

Users should not configure these bits to 11_B.
Users should only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFPLS[2:0]

Transmit/Receive FIFO Payload Data size configuration.

These bits define the message data payload allocation in the RAM.
This is the max. number of Bytes which can be received or transmitted by this FIFO.
Refer to **Section 23.7** for details.

Users can only write to this bit when RS-CANFD module is in GL_RESET mode.

RSCFDnCFDCFCCk.CFTXIE

Transmit/Receive FIFO TX Interrupt Enable.

This bit enables the generation of the Transmit/Receive FIFO Interrupt when the Interrupt flag is set after transmission of a frame from the corresponding FIFO.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDCFCCk.CFRXIE

Transmit/Receive FIFO RX Interrupt Enable.

This bit enables generation of the FIFO Interrupt when the Interrupt flag is set after reception of a frame in the corresponding FIFO.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDCFCCk.CFE

Transmit/Receive FIFO Enable.

This bit enables the FIFO when it is set.
FIFO is disabled when this bit is cleared.
This bit can also be used, by clearing it, to abort transmission from Transmit/Receive FIFO when configured in TX Mode or GW Mode.
This bit can also be used, by clearing it, to stop reception into the Transmit/Receive FIFO in RX mode.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING modes and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
This bit can only be set if the configured FIFO depth is greater than 0_H (RSCFDnCFDCFCCk.CFDC > 000_B).

The RSCFDnCFDCFCCk.CFE bit should be set by a separate write access to the RSCFDnCFDCFCCk register, after all the other bits of the RSCFDnCFDCFCCk register have been set.

This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.
This bit is cleared automatically when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

23.3.7.2 RSCFDnCFDCFCCEk — Transmit/Receive FIFO Configuration / Control Enhancement Register

Access: RSCFDnCFDCFCCEk register can be read or written in 32-bit units
RSCFDnCFDCFCCEkL, RSCFDnCFDCFCCEkH register can be read or written in 16-bit units
RSCFDnCFDCFCCEkLL, RSCFDnCFDCFCCEkLH, RSCFDnCFDCFCCEkHL, RSCFDnCFDCFCCEkHH register can be read or written in 8-bit units

Address: RSCFDnCFDCFCCEk: <RSCFDn_base> + 0180_H + (04_H × k)

RSCFDnCFDCFCCEkL: <RSCFDn_base> + 0180_H + (04_H × k),
RSCFDnCFDCFCCEkH: <RSCFDn_base> + 0182_H + (04_H × k)

RSCFDnCFDCFCCEkLL: <RSCFDn_base> + 0180_H + (04_H × k),
RSCFDnCFDCFCCEkLH: <RSCFDn_base> + 0181_H + (04_H × k),
RSCFDnCFDCFCCEkHL: <RSCFDn_base> + 0182_H + (04_H × k),
RSCFDnCFDCFCCEkHH: <RSCFDn_base> + 0183_H + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFBME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CFMOWM	—	—	—	—	—	CFOFTXIE	CFOFRXIE	CFFIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.46 RSCFDnCFDCFCCEk Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	CFBME	Transmit/Receive FIFO Buffering Mode Enable 0: Transmission from Transmit/Receive FIFO 1: Transmission halt from Transmit/Receive FIFO
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CFMOWM	Transmit/Receive FIFO message overwrite mode 0: Message discarded mode 1: Message overwrite mode
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CFOFTXIE	Transmit/Receive FIFO One Frame Transmission Interrupt Enable 0: One Frame TX Interrupt generation disabled 1: One Frame TX Interrupt generation enabled
1	CFOFRXIE	Transmit/Receive FIFO One Frame Reception Interrupt Enable 0: One Frame RX Interrupt generation disabled 1: One Frame RX Interrupt generation enabled
0	CFFIE	Transmit/Receive FIFO Full interrupt Enable 0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled

The Transmit/Receive FIFO Configuration / Control Enhancement registers are used to configure the transmit/receive FIFOs.
Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(no_of_CFIFOs_per_channel = No. of Transmit/Receive FIFOs per Channel = 3)

Where the total number of CFIFOs = no_of_CFIFOs = no_of_CFIFOs_per_channel * no_of_channels
= 3 * 8
= 24 as shown in **Figure 23.27**
(k = Transmit/Receive FIFO index = [0 .. no_of_CFIFOs -1])

RSCFDnCFDCFCCEk.CFBME

Transmit/Receive FIFO Buffering Mode Enable

When this bit is 0, messages are transmitted from FIFO.
When this bit is 1, messages are not transmitted from FIFO.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
Users should not write 1 from 0 for this bit when the RSCFDnCFDCFCCk.CFE bit is 1_B.

RSCFDnCFDCFCCEk.CFMOWM

Transmit/Receive FIFO message overwrite mode

A receiving message is discarded, when this bit is 0 and FIFO is full.
A receiving message is overwritten, when this bit is 1 and FIFO is full.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
Users should only write 1 to this bit when the Transmit/Receive FIFO is in GW mode.
Users should not write change for this bit when the RSCFDnCFDCFCCk.CFE bit is 1_B.

RSCFDnCFDCFCCEk.CFOFTXIE

Transmit/Receive FIFO One Frame Transmission Interrupt Enable

This bit enables generation of the One Frame Transmission Interrupt when the Interrupt flag is set after transmission of a frame in the corresponding FIFO.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDCFCCEk.CFOFRXIE

Transmit/Receive FIFO One Frame Reception Interrupt Enable

This bit enables generation of the One Frame Reception Interrupt when the Interrupt flag is set after reception of a frame in the corresponding FIFO.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

RSCFDnCFDCFCCEk.CFFIE

Transmit/Receive FIFO Full interrupt Enable

This bit enables generation of the FIFO full Interrupt when the Interrupt flag is set after reception of a frame in the corresponding FIFO.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

The following contents are imagined as how to use interruption.

1. Interruption output in number of arbitrary stages (RSCFDnCFDCFCCk.CFIGCV)

2. Interruption output in FIFO full state

Management of the receiving data of FIFO can be performed by these notices of interruption.

23.3.7.3 RSCFDnCFDCFSTSk — Transmit/Receive FIFO Status Register k

Access: RSCFDnCFDCFSTSk register can be read or written in 32-bit units
RSCFDnCFDCFSTSkL, RSCFDnCFDCFSTSkH register can be read or written in 16-bit units
RSCFDnCFDCFSTSkLL, RSCFDnCFDCFSTSkLH, RSCFDnCFDCFSTSkHL, RSCFDnCFDCFSTSkHH register can be read or written in 8-bit units

Address: RSCFDnCFDCFSTSk: <RSCFDn_base> + $01E0_H$ + (04_H × k)

RSCFDnCFDCFSTSkL: <RSCFDn_base> + $01E0_H$ + (04_H × k),
RSCFDnCFDCFSTSkH: <RSCFDn_base> + $01E2_H$ + (04_H × k)

RSCFDnCFDCFSTSkLL: <RSCFDn_base> + $01E0_H$ + (04_H × k),
RSCFDnCFDCFSTSkLH: <RSCFDn_base> + $01E1_H$ + (04_H × k),
RSCFDnCFDCFSTSkHL: <RSCFDn_base> + $01E2_H$ + (04_H × k),
RSCFDnCFDCFSTSkHH: <RSCFDn_base> + $01E3_H$ + (04_H × k)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CFMOW	—	—	—	—	—	CFOFTXIF	CFOFRXIF	CFFIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 23.47 RSCFDnCFDCFSTSk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CFMOW	Transmit/Receive FIFO message overwrite 0: No Message overwrite occurred in FIFO 1: Message overwrite occurred in FIFO
23 to 19	Reserved	These bits are read as the value after reset. When writing, write the value after reset.
18	CFOFTXIF	Transmit/Receive FIFO One Frame Transmission Every FIFO transmits a frame, a corresponding interrupt is set.
17	CFOFRXIF	Transmit/Receive FIFO One Frame Reception Every FIFO receives a frame, a corresponding interrupt is set.
16	CFFIF	Transmit/Receive FIFO Full Interrupt Flag 0: Interrupt condition not satisfied for FIFO Full interrupt 1: Interrupt condition satisfied for FIFO Full interrupt
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Count No. of Messages stored in FIFO
7 to 5	Reserved	These bits are read as the value after reset. When writing, write the value after reset.
4	CFTXIF	Transmit/Receive TX FIFO Interrupt Flag 0: FIFO Interrupt condition not satisfied after Frame Transmission 1: FIFO Interrupt condition satisfied after Frame Transmission
3	CFRXIF	Transmit/Receive RX FIFO Interrupt Flag 0: FIFO Interrupt condition not satisfied after Frame Reception 1: FIFO Interrupt condition satisfied after Frame Reception
2	CFMLT	Transmit/Receive FIFO Message Lost 0: No Message lost in FIFO 1: FIFO Message Lost

Table 23.47 RSCFDnCFDCFSTSk Register Contents (2/2)

Bit Position	Bit Name	Function
1	CFFLL	Transmit/Receive FIFO Full 0: FIFO Not Full 1: FIFO Full
0	CFEMP	Transmit/Receive FIFO Empty 0: FIFO Not Empty 1: FIFO Empty

The FIFO status registers show the status of the messages stored in corresponding FIFO Buffers.
Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(no_of_CFIFOs_per_channel = No. of Transmit/Receive FIFOs per Channel = 3)
Where the total number of CFIFOs = no_of_CFIFOs = no_of_CFIFOs_per_channel * no_of_channels
= 3 * 8
= 24 as shown in **Figure 23.27**
(k = Transmit/Receive FIFO index = [0 .. no_of_CFIFOs -1])

RSCFDnCFDCFSTSk.CFMOW

Transmit/Receive FIFO Message overwrite.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.
Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.
This bit is set automatically whenever a message is overwrite storage of a new message when RSCFDnCFDCFCCEk.CFMOWM=1 and FIFO is already full in GW mode.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

RSCFDnCFDCFSTSk.CFOFTXIF

Transmit/Receive FIFO One Frame Transmission Interrupt Flag.

This bit will not be cleared automatically if the Transmit/Receive FIFO Buffer is disabled.
Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.
Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.
This bit is set automatically when the One Frame Transmission Interrupt condition is satisfied for transmit/receive FIFO Buffers configured in GW mode or TX mode.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.
This bit is not influenced by the value of RSCFDnCFDCFCCk.CFIM.

RSCFDnCFDCFSTSk.CFOFRXIF

Transmit/Receive FIFO One Frame Reception Interrupt Flag.

This bit will not be cleared automatically if the Transmit/Receive FIFO Buffer is disabled.
Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.
Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.
This bit is set automatically when the One Frame Reception Interrupt condition is satisfied for Transmit/Receive FIFO Buffers when configured in GW mode or RX mode.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.
This bit is not influenced by the value of RSCFDnCFDCFCCk.CFIM.

RSCFDnCFDCFSTSk.CFFIF

Transmit/Receive FIFO Full Interrupt Flag.

This bit will not be cleared automatically if the Transmit/Receive FIFO Buffer is disabled.
Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.
Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.
This bit is set automatically when the FIFO full Interrupt condition is satisfied for Transmit/Receive FIFO Buffers when configured in GW mode or RX mode.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.
The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

RSCFDnCFDCFSTSk.CFMC[7:0]

Transmit/Receive FIFO Message Count.

These bits indicate the following:
Number of CAN messages stored by the CPU in the FIFO configured in TX Mode pending for transmission.
Number of CAN messages stored in the FIFO Buffer configured in RX Mode by RS-CANFD to be read by the CPU.
Number of CAN messages stored by the RS-CANFD in the GW FIFO pending for transmission.
These bits are cleared automatically when the FIFO is disabled.
These bits are cleared automatically when RS-CANFD module enters GL_RESET mode.
These bits are cleared automatically when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

RSCFDnCFDCFSTSk.CFTXIF

Transmit/Receive TX FIFO Interrupt Flag.

This bit will not be cleared automatically if the Transmit/Receive FIFO Buffer is disabled.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically when the configured Interrupt condition is satisfied for Transmit/Receive FIFO Buffers configured in GW mode or TX mode.

The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

RSCFDnCFDCFSTSk.CFRXIF

Transmit/Receive RX FIFO Interrupt Flag.

This bit will not be cleared automatically if the Transmit/Receive FIFO Buffer is disabled.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically when the configured Interrupt condition is satisfied for Transmit/Receive FIFO Buffers when configured in GW mode or RX mode.

The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in GW mode.

RSCFDnCFDCFSTSk.CFMLT

Transmit/Receive FIFO Message Lost.

Users can only write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode and the related RS-CANFD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.
Writing 1_B has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

This bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX or GW mode.
If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The bit is cleared by writing 0_B to it.
The bit is cleared when RS-CANFD module enters GL_RESET mode.
The bit is cleared when the related channel enters CH_RESET mode if the FIFO is configured in TX or GW mode.

RSCFDnCFDCFSTSk.CFFLL

Transmit/Receive FIFO Full.

This bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

This bit is cleared automatically when the number of CAN messages stored in the FIFO is less than the configured FIFO depth.
This bit is cleared automatically when the FIFO is disabled by setting the CFE bit to 0_B.
This bit is cleared automatically when RS-CANFD module enters GL_RESET.
This bit is cleared automatically when RS-CANFD module enters CH_RESET when FIFO is configured in TX or GW Mode.

RSCFDnCFDCFSTSk.CFEMP

Transmit/Receive FIFO Empty.

This bit is set automatically when the CPU has read all messages from the FIFO configured in RX mode.
This bit is set automatically when all messages have been transmitted from the FIFO configured in TX or GW Mode.
This bit is set automatically when FIFO is disabled by setting the CFE bit to 0_B.
This bit is set automatically when RS-CANFD module enters GL_RESET mode.
This bit is set automatically when RS-CANFD module enters CH_RESET when FIFO configured in TX Mode or GW Mode.
This bit is cleared automatically when the first reception message is stored in the FIFO when configured in RX Mode.
This bit is cleared automatically when the first message to be transmitted is stored in the FIFO when configured in TX or GW Mode.

23.3.7.4 RSCFDnCFDCFPCTRk — Transmit/Receive FIFO Pointer Control Register k

Access: RSCFDnCFDCFPCTRk register is a write-only register that can be written in 32-bit units
RSCFDnCFDCFPCTRkL, RSCFDnCFDCFPCTRkH register is a write-only register that can be written in 16-bit units
RSCFDnCFDCFPCTRkLL, RSCFDnCFDCFPCTRkLH, RSCFDnCFDCFPCTRkHL, RSCFDnCFDCFPCTRkHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDCFPCTRk: <RSCFDn_base> + 0240_H + (04_H × k)

RSCFDnCFDCFPCTRkL: <RSCFDn_base> + 0240_H + (04_H × k),
RSCFDnCFDCFPCTRkH: <RSCFDn_base> + 0242_H + (04_H × k)

RSCFDnCFDCFPCTRkLL: <RSCFDn_base> + 0240_H + (04_H × k),
RSCFDnCFDCFPCTRkLH: <RSCFDn_base> + 0241_H + (04_H × k),
RSCFDnCFDCFPCTRkHL: <RSCFDn_base> + 0242_H + (04_H × k),
RSCFDnCFDCFPCTRkHH: <RSCFDn_base> + 0243_H + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.48 RSCFDnCFDCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control Increments read or write pointer of the corresponding Transmit/Receive FIFO Buffers depending upon the mode configuration

These registers can be used to increment the Read or Write Pointer of the corresponding Transmit/ Receive FIFO.
Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(no_of_CFIFOs_per_channel = No. of Transmit/Receive FIFOs per Channel = 3)
Where the total number of CFIFOs = no_of_CFIFOs = no_of_CFIFOs_per_channel * no_of_channels
= 3 * 8
= 24 as shown in **Figure 23.27**
(k = Transmit/Receive FIFO index = [0 .. no_of_CFIFOs -1])

RSCFDnCFDCFPCTRk.CFPC[7:0]

Transmit/Receive FIFO Pointer Control

When the value FF_H is written into these bits, then the Read Pointer of the corresponding Transmit/Receive FIFO Buffer, when configured in RX mode, or the Write Pointer of the corresponding Transmit/Receive FIFO Buffer, when configured in TX mode, moves to the next FIFO entry.

Read value from these bits is always 0_H.
Users can only write to these bits when RS-CANFD module is in GL_TEST or GL_OPERATING modes.

Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not empty if configured in RX mode.
Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not full if configured in TX mode.
Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not configured in GW mode.

Users should not write to the FIFO control registers when DMA is enabled.

23.3.8 Details of FIFO Status-Related Registers

23.3.8.1 RSCFDnCFDFESTS — FIFO Empty Status Register

Access: RSCFDnCFDFESTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDFESTSL, RSCFDnCFDFESTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDFESTSLL, RSCFDnCFDFESTSLH, RSCFDnCFDFESTSHL, RSCFDnCFDFESTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDFESTS: <RSCFDn_base> + $02A0_H$

RSCFDnCFDFESTSL: <RSCFDn_base> + $02A0_H$, RSCFDnCFDFESTSH: <RSCFDn_base> + $02A2_H$

RSCFDnCFDFESTSLL: <RSCFDn_base> + $02A0_H$, RSCFDnCFDFESTSLH: <RSCFDn_base> + $02A1_H$,
RSCFDnCFDFESTSHL: <RSCFDn_base> + $02A2_H$, RSCFDnCFDFESTSHH: <RSCFDn_base> + $02A3_H$

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFkEMP[23:8]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkEMP[7:0]								RFxEMP[7:0]							
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.49 RSCFDnCFDFESTS Register Contents

Bit Position	Bit Name	Function
31 to 8	CFkEMP[23:0]	Transmit/Receive FIFO Empty Status 0: Corresponding FIFO not Empty 1: Corresponding FIFO Empty
7 to 0	RFxEMP[7:0]	RX FIFO Empty Status 0: Corresponding FIFO not Empty 1: Corresponding FIFO Empty

The FIFO Empty status register bits show the status of the Empty bits of the FIFO Buffers.

RSCFDnCFDFESTS.CFkEMP[23:0]

Transmit/Receive FIFO Empty Status

This bit is set when the RS-CANFD module enters GL_RESET mode.
Each bit is set automatically when the corresponding bit is set in the Transmit/Receive FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the Transmit/Receive FIFO Status Register.

RSCFDnCFDFESTS.RFxEMP[7:0]

RX FIFO Empty Status

This bit is set when the RS-CANFD module enters GL_RESET mode.
Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Register.

23.3.8.2 RSCFDnCFDFFSTS — FIFO Full Status Register

Access: RSCFDnCFDFFSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDFFSTSL, RSCFDnCFDFFSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDFFSTSLL, RSCFDnCFDFFSTSLH, RSCFDnCFDFFSTSHL, RSCFDnCFDFFSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDFFSTS: <RSCFDn_base> + $02A4_H$

RSCFDnCFDFFSTSL: <RSCFDn_base> + $02A4_H$, RSCFDnCFDFFSTSH: <RSCFDn_base> + $02A6_H$

RSCFDnCFDFFSTSLL: <RSCFDn_base> + $02A4_H$, RSCFDnCFDFFSTSLH: <RSCFDn_base> + $02A5_H$,
RSCFDnCFDFFSTSHL: <RSCFDn_base> + $02A6_H$, RSCFDnCFDFFSTSHH: <RSCFDn_base> + $02A7_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFkFLL[23:8]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkFLL[7:0]								RFxFLL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.50 RSCFDnCFDFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 8	CFkFLL[23:0]	Transmit/Receive FIFO Full Status 0: Corresponding FIFO not Full 1: Corresponding FIFO Full
7 to 0	RFxFLL[7:0]	RX FIFO Full Status 0: Corresponding FIFO not Full 1: Corresponding FIFO Full

The FIFO Full status register bits show the status of the Full bits of the FIFO Buffers.

RSCFDnCFDFFSTS.CFkFLL[23:0]

Transmit/Receive FIFO Full Status

Each bit is set automatically when the corresponding bit is set in the Transmit/Receive FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the Transmit/Receive FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDFFSTS.RFxFLL[7:0]

RX FIFO Full Status

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.8.3 RSCFDnCFDFMSTS — FIFO Message Lost Status Register

Access: RSCFDnCFDFMSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDFMSTSL, RSCFDnCFDFMSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDFMSTSLL, RSCFDnCFDFMSTSLH, RSCFDnCFDFMSTSHL, RSCFDnCFDFMSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDFMSTS: <RSCFDn_base> + $02A8_H$

RSCFDnCFDFMSTSL: <RSCFDn_base> + $02A8_H$, RSCFDnCFDFMSTSH: <RSCFDn_base> + $02AA_H$

RSCFDnCFDFMSTSLL: <RSCFDn_base> + $02A8_H$, RSCFDnCFDFMSTSLH: <RSCFDn_base> + $02A9_H$,
RSCFDnCFDFMSTSHL: <RSCFDn_base> + $02AA_H$, RSCFDnCFDFMSTSHH: <RSCFDn_base> + $02AB_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFkMLT[23:8]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkMLT[7:0]								RFxMLT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.51 RSCFDnCFDFMSTS Register Contents

Bit Position	Bit Name	Function
31 to 8	CFkMLT[23:0]	Transmit/Receive FIFO Msg Lost Status 0: Transmit/Receive FIFO Msg Lost flag not set 1: Transmit/Receive FIFO Msg Lost flag set
7 to 0	RFxMLT[7:0]	RX FIFO Msg Lost Status 0: Transmit/Receive FIFO Msg Lost flag not set 1: Transmit/Receive FIFO Msg Lost flag set

The FIFO Msg Lost status register bits show the status of the Msg Lost bits of the FIFO Buffers.

RSCFDnCFDFMSTS.CFkMLT[23:0]

Transmit/Receive FIFO Msg Lost Status

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is set automatically when the corresponding bit is set in the Transmit/Receive FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the Transmit/Receive FIFO Status Register.

RSCFDnCFDFMSTS.RFxMLT[7:0]

RX FIFO Msg Lost Status

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Register.

23.3.8.4 RSCFDnCFDRFISTS — Receive FIFO Interrupt Flag Status Register

Access: RSCFDnCFDRFISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDRFISTSL, RSCFDnCFDRFISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDRFISTSLL, RSCFDnCFDRFISTSLH, RSCFDnCFDRFISTSHL, RSCFDnCFDRFISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDRFISTS: <RSCFDn_base> + $02AC_H$

RSCFDnCFDRFISTSL: <RSCFDn_base> + $02AC_H$, RSCFDnCFDRFISTSH: <RSCFDn_base> + $02AE_H$

RSCFDnCFDRFISTSLL: <RSCFDn_base> + $02AC_H$, RSCFDnCFDRFISTSLH: <RSCFDn_base> + $02AD_H$, RSCFDnCFDRFISTSHL: <RSCFDn_base> + $02AE_H$, RSCFDnCFDRFISTSHH: <RSCFDn_base> + $02AF_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RFxFFLL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFxIF[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.52 RSCFDnCFDRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 16	RFxFFLL[7:0]	RX FIFO[x] Interrupt Full Flag Status 0: Corresponding RX FIFO interrupt Full flag not set 1: Corresponding RX FIFO interrupt Full flag set
15 to 8	Reserved	When read, the value after reset is returned.
7 to 0	RFxIF[7:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO interrupt flag not set 1: Corresponding RX FIFO interrupt flag set

The FIFO Interrupt Flag status register bits show the status of the Interrupt Flag bits of the RX FIFO Buffers.

RSCFDnCFDRFISTS.RFxFFLL[7:0]

RX FIF0[x] Interrupt Full Flag Status

Each bit is set automatically when the corresponding interrupt Full flag bit is set in the RX FIFO Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding interrupt Full flag bit is cleared in the RX FIFO Status Register.

RSCFDnCFDRFISTS.RFxIF[7:0]

RX FIFO[x] Interrupt Flag Status

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Register.

23.3.8.5 RSCFDnCFDCFRISTS — Transmit/Receive FIFO Receive Interrupt Flag Status Register

Access: RSCFDnCFDCFRISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCFRISTSL, RSCFDnCFDCFRISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCFRISTSLL, RSCFDnCFDCFRISTSLH, RSCFDnCFDCFRISTSHL, RSCFDnCFDCFRISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFRISTS: <RSCFDn_base> + $02B0_H$

RSCFDnCFDCFRISTSL: <RSCFDn_base> + $02B0_H$, RSCFDnCFDCFRISTSH: <RSCFDn_base> + $02B2_H$

RSCFDnCFDCFRISTSLL: <RSCFDn_base> + $02B0_H$, RSCFDnCFDCFRISTSLH: <RSCFDn_base> + $02B1_H$,
RSCFDnCFDCFRISTSHL: <RSCFDn_base> + $02B2_H$, RSCFDnCFDCFRISTSHH: <RSCFDn_base> + $02B3_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFkRXIF[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkRXIF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.53 RSCFDnCFDCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CFkRXIF[23:0]	Transmit/Receive FIFO [k] RX Interrupt Flag Status 0: Corresponding Transmit/Receive FIFO RX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO RX interrupt flag is set

The FIFO Interrupt Flag status register bits show the status of the Interrupt Flag bits of the Transmit/ Receive FIFO Buffers.

RSCFDnCFDCFRISTS.CFkRXIF[23:0]

Transmit/Receive FIFO [k] RX Interrupt Flag Status

Each bit is set automatically when the corresponding RX interrupt flag bit is set in the Transmit/ Receive FIFO Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding RX interrupt flag bit is cleared in the Transmit/Receive FIFO Status Register.

23.3.8.6 RSCFDnCFDCFTISTS — Transmit/Receive FIFO Transmit Interrupt Flag Status Register

Access: RSCFDnCFDCFTISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCFTISTSL, RSCFDnCFDCFTISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCFTISTSLL, RSCFDnCFDCFTISTSLH, RSCFDnCFDCFTISTSHL, RSCFDnCFDCFTISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFTISTS: <RSCFDn_base> + $02B4_H$

RSCFDnCFDCFTISTSL: <RSCFDn_base> + $02B4_H$, RSCFDnCFDCFTISTSH: <RSCFDn_base> + $02B6_H$

RSCFDnCFDCFTISTSLL: <RSCFDn_base> + $02B4_H$, RSCFDnCFDCFTISTSLH: <RSCFDn_base> + $02B5_H$,
RSCFDnCFDCFTISTSHL: <RSCFDn_base> + $02B6_H$, RSCFDnCFDCFTISTSHH: <RSCFDn_base> + $02B7_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFkTXIF[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkTXIF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.54 RSCFDnCFDCFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CFkTXIF[23:0]	Transmit/Receive FIFO [k] TX Interrupt Flag Status 0: Corresponding Transmit/Receive FIFO TX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO TX interrupt flag is set

The FIFO Interrupt Flag status register bits show the status of the Interrupt Flag bits of the Transmit/ Receive FIFO Buffers.

RSCFDnCFDCFTISTS.CFkTXIF[23:0]

Transmit/Receive FIFO [k] TX Interrupt Flag Status

Each bit is set automatically when the corresponding TX interrupt flag bit is set in the Transmit/ Receive FIFO Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding TX interrupt flag bit is cleared in the Transmit/Receive FIFO Status Register.

23.3.8.7 RSCFDnCFDFFFSTS — FIFO FDC level Full Status Register

Access: RSCFDnCFDFFFSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDFFFSTSL, RSCFDnCFDFFFSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDFFFSTSLL, RSCFDnCFDFFFSTSLH, RSCFDnCFDFFFSTSHL, RSCFDnCFDFFFSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDFFFSTS: <RSCFDn_base> + $02C4_H$

RSCFDnCFDFFFSTSL: <RSCFDn_base> + $02C4_H$, RSCFDnCFDFFFSTSH: <RSCFDn_base> + $02C6_H$

RSCFDnCFDFFFSTSLL: <RSCFDn_base> + $02C4_H$, RSCFDnCFDFFFSTSLH: <RSCFDn_base> + $02C5_H$,
RSCFDnCFDFFFSTSHL: <RSCFDn_base> + $02C6_H$, RSCFDnCFDFFFSTSHH: <RSCFDn_base> + $02C7_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFkFFLL[23:8]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkFFLL[7:0]								RFxFFLL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.55 RSCFDnCFDFFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 8	CFkFFLL[23:0]	Transmit/Receive FIFO FDC level full Status 0: Corresponding FIFO Full interrupt not set 1: Corresponding FIFO Full interrupt is set
7 to 0	RFxFFLL[7:0]	RX FIFO FDC level full Status 0: Corresponding FIFO Full interrupt not set 1: Corresponding FIFO Full interrupt is set

The FIFO Full status register bits show the status of the Full Interrupt Flag bits of the FIFO Buffers.

RSCFDnCFDFFFSTS.CFkFFLL[23:0]

Transmit/Receive FIFO FDC level Full Status

Each bit is set automatically when the corresponding bit is set in the Transmit/Receive FIFO Status Register.
Each bit is cleared automatically when the corresponding bit is cleared in the Transmit/Receive FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDFFFSTS.RFxFFLL[7:0]

RX FIFO FDC level Full Status.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Register.
Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.8.8 RSCFDnCFDCFMOWSTS — Transmit/Receive FIFO Message OverWrite Status Register

Access: RSCFDnCFDCFMOWSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCFMOWSTSL, RSCFDnCFDCFMOWSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCFMOWSTSLL, RSCFDnCFDCFMOWSTSLH, RSCFDnCFDCFMOWSTSHL, RSCFDnCFDCFMOWSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFMOWSTS: <RSCFDn_base> + $02C0_H$

RSCFDnCFDCFMOWSTSL: <RSCFDn_base> + $02C0_H$, RSCFDnCFDCFMOWSTSH: <RSCFDn_base> + $02C2_H$

RSCFDnCFDCFMOWSTSLL: <RSCFDn_base> + $02C0_H$, RSCFDnCFDCFMOWSTSLH: <RSCFDn_base> + $02C1_H$,
RSCFDnCFDCFMOWSTSHL: <RSCFDn_base> + $02C2_H$, RSCFDnCFDCFMOWSTSHH: <RSCFDn_base> + $02C3_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFkMOW[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkMOW[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.56 RSCFDnCFDCFMOWSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CFkMOW[23:0]	Transmit/Receive FIFO [k] Message overwrite Status 0: Corresponding FIFO overwrite flag is not set 1: Corresponding FIFO overwrite flag is set

RSCFDnCFDCFMOWSTS.CFkMOW[23:0]

Transmit/Receive FIFO [k] Message OverWrite Status.

This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is set automatically when the corresponding bit is set in the Transmit/Receive FIFO Status Register.
Each bit is cleared automatically when the corresponding bit is cleared in the Transmit/Receive FIFO Status Register.
This register is effective only in GW mode.

23.3.8.9 RSCFDnCFDCFOFRISTS — Transmit/Receive FIFO One Frame Receive Interrupt Flag Status

Access: RSCFDnCFDCFOFRISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCFOFRISTSL, RSCFDnCFDCFOFRISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCFOFRISTSLL, RSCFDnCFDCFOFRISTSLH, RSCFDnCFDCFOFRISTSHL, RSCFDnCFDCFOFRISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFOFRISTS: <RSCFDn_base> + $02B8_H$

RSCFDnCFDCFOFRISTSL: <RSCFDn_base> + $02B8_H$, RSCFDnCFDCFOFRISTSH: <RSCFDn_base> + $02BA_H$

RSCFDnCFDCFOFRISTSLL: <RSCFDn_base> + $02B8_H$, RSCFDnCFDCFOFRISTSLH: <RSCFDn_base> + $02B9_H$,
RSCFDnCFDCFOFRISTSHL: <RSCFDn_base> + $02BA_H$, RSCFDnCFDCFOFRISTSHH: <RSCFDn_base> + $02BB_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFkOFRXIF[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkOFRXIF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.57 RSCFDnCFDCFOFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CFkOFRXIF [23:0]	Transmit/Receive FIFO [k] One Frame RX Interrupt Flag Status 0: Corresponding Transmit/Receive FIFO One Frame RX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO One Frame RX interrupt flag is set

The FIFO One Frame RX Interrupt Flag status register bits show the status of the Interrupt Flag bits of the Transmit/Receive FIFO Buffers.

RSCFDnCFDCFOFRISTS.CFkOFRXIF[23:0]

Transmit/Receive FIFO [k] One Frame RX Interrupt Flag Status.

Each bit is set automatically when the corresponding One Frame RX interrupt flag bit is set in the Transmit/Receive FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding One Frame RX interrupt flag bit is cleared in the Transmit/Receive FIFO Status Register.

23.3.8.10 RSCFDnCFDCFOFTISTS — Transmit/Receive FIFO One Frame Transmit Interrupt Flag Status

Access: RSCFDnCFDCFOFTISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCFOFTISTSL, RSCFDnCFDCFOFTISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCFOFTISTSLL, RSCFDnCFDCFOFTISTSLH, RSCFDnCFDCFOFTISTSHL, RSCFDnCFDCFOFTISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFOFTISTS: <RSCFDn_base> + $02BC_H$

RSCFDnCFDCFOFTISTSL: <RSCFDn_base> + $02BC_H$, RSCFDnCFDCFOFTISTSH: <RSCFDn_base> + $02BE_H$

RSCFDnCFDCFOFTISTSLL: <RSCFDn_base> + $02BC_H$, RSCFDnCFDCFOFTISTSLH: <RSCFDn_base> + $02BD_H$,
RSCFDnCFDCFOFTISTSHL: <RSCFDn_base> + $02BE_H$, RSCFDnCFDCFOFTISTSHH: <RSCFDn_base> + $02BF_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFkOFTXIF[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFkOFTXIF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.58 RSCFDnCFDCFOFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CFkOFTXIF [23:0]	Transmit/Receive FIFO [k] One Frame TX Interrupt Flag Status 0: Corresponding Transmit/Receive FIFO One Frame TX interrupt flag is not set 1: Corresponding Transmit/Receive FIFO One Frame TX interrupt flag is set

The FIFO One Frame TX Interrupt Flag status register bits show the status of the Interrupt Flag bits of the Transmit/Receive FIFO Buffers.

RSCFDnCFDCFOFTISTS.CFkOFTXIF[23:0]

Transmit/Receive FIFO [k] One Frame TX Interrupt Flag Status.

Each bit is set automatically when the corresponding One Frame TX interrupt flag bit is set in the Transmit/Receive FIFO Status Register.
This bit is cleared when RS-CANFD module enters GL_RESET mode.
Each bit is cleared automatically when the corresponding One Frame TX interrupt flag bit is cleared in the Transmit/Receive FIFO Status Register.

23.3.9 Details of FIFO DMA-Related Registers

23.3.9.1 RSCFDnCFDCDTCT — DMA Transfer Control Register

Access: RSCFDnCFDCDTCT register can be read or written in 32-bit units
RSCFDnCFDCDTCTL, RSCFDnCFDCDTCTH register can be read or written in 16-bit units
RSCFDnCFDCDTCTLL, RSCFDnCFDCDTCTLH, RSCFDnCFDCDTCTHL, RSCFDnCFDCDTCTHH register can be read or written in 8-bit units

Address: RSCFDnCFDCDTCT: <RSCFDn_base> + 1330_H

RSCFDnCFDCDTCTL: <RSCFDn_base> + 1330_H
RSCFDnCFDCDTCTH: <RSCFDn_base> + 1332_H

RSCFDnCFDCDTCTLL: <RSCFDn_base> + 1330_H,
RSCFDnCFDCDTCTLH: <RSCFDn_base> + 1331_H,
RSCFDnCFDCDTCTHL: <RSCFDn_base> + 1332_H,
RSCFDnCFDCDTCTHH: <RSCFDn_base> + 1333_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDMAE7	CFDMAE6	CFDMAE5	CFDMAE4	CFDMAE3	CFDMAE2	CFDMAE1	CFDMAE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.59 RSCFDnCFDCDTCT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	CFDMAE7	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 7 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
14	CFDMAE6	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 6 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
13	CFDMAE5	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 5 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
12	CFDMAE4	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 4 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
11	CFDMAE3	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 3 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
10	CFDMAE2	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 2 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
9	CFDMAE1	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 1 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m
8	CFDMAE0	DMA Transfer Enable for Transmit/Receive FIFO 0 of channel 0 0: DMA Transfer Request disabled for channel m 1: DMA Transfer Request enabled for channel m

Table 23.59 RSCFDnCFDCDTCT Register Contents (2/2)

Bit Position	Bit Name	Function
7	RFDMAE7	DMA Transfer Enable for RXFIFO 7 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
6	RFDMAE6	DMA Transfer Enable for RXFIFO 6 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
5	RFDMAE5	DMA Transfer Enable for RXFIFO 5 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
4	RFDMAE4	DMA Transfer Enable for RXFIFO 4 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
3	RFDMAE3	DMA Transfer Enable for RXFIFO 3 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
2	RFDMAE2	DMA Transfer Enable for RXFIFO 2 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
1	RFDMAE1	DMA Transfer Enable for RXFIFO 1 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled
0	RFDMAE0	DMA Transfer Enable for RXFIFO 0 0: DMA Transfer Request disabled 1: DMA Transfer Request enabled

The DMA Transfer Control Register bits control the start and stop of the DMA transfer operation.

RSCFDnCFDCDTCT.CFDMAEm

DMA Transfer Enable for Transmit/Receive FIFO 0 of channel m.

By this bit only Transmit/Receive FIFO0 can be linked to a DMA channel, to link Transmit/Receive FIFO2 see RSCFDnCFDCDTTCT.CFDMAEm.

Transmit/Receive FIFO1 cannot be linked to a DMA channel.

This bit cannot be set in GL_STOP or GL_RESET mode.

Users should not enable a DMA transfer for a Transmit/Receive FIFO that is configured as TX or GW FIFO.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTCT.RFDMAEe

DMA Transfer Enable for RXFIFO e.

(e = DMA Transfer Enable FIFO index = [0... No. of RX FIFOs-1]) (No. of RX FIFOs = 8)

This bit cannot be set in GL_STOP or GL_RESET mode.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.9.2 RSCFDnCFDCDTSTS — DMA Transfer Status Register

Access: RSCFDnCFDCDTSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCDTSTSL, RSCFDnCFDCDTSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCDTSTSLL, RSCFDnCFDCDTSTSLH, RSCFDnCFDCDTSTSHL, RSCFDnCFDCDTSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCDTSTS: <RSCFDn_base> + 1334_H

RSCFDnCFDCDTSTSL: <RSCFDn_base> + 1334_H
RSCFDnCFDCDTSTSH: <RSCFDn_base> + 1336_H

RSCFDnCFDCDTSTSLL: <RSCFDn_base> + 1334_H,
RSCFDnCFDCDTSTSLH: <RSCFDn_base> + 1335_H,
RSCFDnCFDCDTSTSHL: <RSCFDn_base> + 1336_H,
RSCFDnCFDCDTSTSHH: <RSCFDn_base> + 1337_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDMASTS7	CFDMASTS6	CFDMASTS5	CFDMASTS4	CFDMASTS3	CFDMASTS2	CFDMASTS1	CFDMASTS0	RFDMASTS7	RFDMASTS6	RFDMASTS5	RFDMASTS4	RFDMASTS3	RFDMASTS2	RFDMASTS1	RFDMASTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.60 RSCFDnCFDCDTSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	CFDMASTS7	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 7 0: DMA transfer stopped 1: DMA transfer ongoing
14	CFDMASTS6	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 6 0: DMA transfer stopped 1: DMA transfer ongoing
13	CFDMASTS5	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 5 0: DMA transfer stopped 1: DMA transfer ongoing
12	CFDMASTS4	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 4 0: DMA transfer stopped 1: DMA transfer ongoing
11	CFDMASTS3	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 3 0: DMA transfer stopped 1: DMA transfer ongoing
10	CFDMASTS2	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 2 0: DMA transfer stopped 1: DMA transfer ongoing
9	CFDMASTS1	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 1 0: DMA transfer stopped 1: DMA transfer ongoing
8	CFDMASTS0	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel 0 0: DMA transfer stopped 1: DMA transfer ongoing
7	RFDMASTS7	DMA Transfer Status for RX FIFO 7 0: DMA transfer stopped R 1: DMA transfer ongoing

Table 23.60 RSCFDnCFDCDTSTS Register Contents (2/2)

Bit Position	Bit Name	Function
6	RFDMASTS6	DMA Transfer Status for RX FIFO 6 0: DMA transfer stopped R 1: DMA transfer ongoing
5	RFDMASTS5	DMA Transfer Status for RX FIFO 5 0: DMA transfer stopped R 1: DMA transfer ongoing
4	RFDMASTS4	DMA Transfer Status for RX FIFO 4 0: DMA transfer stopped R 1: DMA transfer ongoing
3	RFDMASTS3	DMA Transfer Status for RX FIFO 3 0: DMA transfer stopped R 1: DMA transfer ongoing
2	RFDMASTS2	DMA Transfer Status for RX FIFO 2 0: DMA transfer stopped R 1: DMA transfer ongoing
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped R 1: DMA transfer ongoing
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped R 1: DMA transfer ongoing

The DMA Transfer Status Register bits show the status of the DMA transfer.

RSCFDnCFDCDTSTS.CFDMASTSm

DMA Transfer Status only for Transmit/Receive FIFO 0 of channel m

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.
When RSCFDnCFDCDTCT.CFDMAEm is set to 0_B while DMA transfer for the corresponding FIFO is on going, RSCFDnCFDCDTSTS.CFDMASTSm becomes 0_B when the DMA transfer is completed. This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTSTS.RFDMASTSe

DMA Transfer Status for RX FIFO e

(e = DMA Transfer Enable FIFO index = [0... No. of RX FIFOs-1]) (No. of RX FIFOs = 8)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.
When RSCFDnCFDCDTCT.RFDMAEe is set to 0_B while DMA transfer for the corresponding FIFO is on going, RSCFDnCFDCDTSTS.RFDMASTSe becomes 0_B when the DMA transfer is completed. This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.9.3 RSCFDnCFDCDTTCT — DMA Transmit Transfer Control Register

Access: RSCFDnCFDCDTTCT register can be read or written in 32-bit units
RSCFDnCFDCDTTCTL, RSCFDnCFDCDTTCTH register can be read or written in 16-bit units
RSCFDnCFDCDTTCTLL, RSCFDnCFDCDTTCTLH, RSCFDnCFDCDTTCTHL,
RSCFDnCFDCDTTCTHHregister can be read or written in 8-bit units

Address: RSCFDnCFDCDTTCT: <RSCFDn_base> + 1340_H

RSCFDnCFDCDTTCTL: <RSCFDn_base> + 1340_H
RSCFDnCFDCDTTCTH: <RSCFDn_base> + 1342_H

RSCFDnCFDCDTTCTLL: <RSCFDn_base> + 1340_H,
RSCFDnCFDCDTTCTLH: <RSCFDn_base> + 1341_H,
RSCFDnCFDCDTTCTHL: <RSCFDn_base> + 1342_H,
RSCFDnCFDCDTTCTHH: <RSCFDn_base> + 1343_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFDMAE7	CFDMAE6	CFDMAE5	CFDMAE4	CFDMAE3	CFDMAE2	CFDMAE1	CFDMAE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TQ3DMAE7	TQ3DMAE6	TQ3DMAE5	TQ3DMAE4	TQ3DMAE3	TQ3DMAE2	TQ3DMAE1	TQ3DMAE0	TQ0DMAE7	TQ0DMAE6	TQ0DMAE5	TQ0DMAE4	TQ0DMAE3	TQ0DMAE2	TQ0DMAE1	TQ0DMAE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.61 RSCFDnCFDCDTTCT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	CFDMAE7	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 7 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
22	CFDMAE6	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 6 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
21	CFDMAE5	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 5 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
20	CFDMAE4	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 4 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
19	CFDMAE3	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 3 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
18	CFDMAE2	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 2 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
17	CFDMAE1	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 1 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
16	CFDMAE0	DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel 0 0: DMA TX Transfer Request disabled for channel m 1: DMA TX Transfer Request enabled for channel m
15	TQ3DMAE7	DMA TX Transfer Enable for TXQ 3 of channel 7 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled

Table 23.61 RSCFDnCFDCDTTCT Register Contents (2/2)

Bit Position	Bit Name	Function
14	TQ3DMAE6	DMA TX Transfer Enable for TXQ 3 of channel 6 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
13	TQ3DMAE5	DMA TX Transfer Enable for TXQ 3 of channel 5 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
12	TQ3DMAE4	DMA TX Transfer Enable for TXQ 3 of channel 4 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
11	TQ3DMAE3	DMA TX Transfer Enable for TXQ 3 of channel 3 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
10	TQ3DMAE2	DMA TX Transfer Enable for TXQ 3 of channel 2 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
9	TQ3DMAE1	DMA TX Transfer Enable for TXQ 3 of channel 1 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
8	TQ3DMAE0	DMA TX Transfer Enable for TXQ 3 of channel 0 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
7	TQ0DMAE7	DMA TX Transfer Enable for TXQ 0 of channel 7 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
6	TQ0DMAE6	DMA TX Transfer Enable for TXQ 0 of channel 6 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
5	TQ0DMAE5	DMA TX Transfer Enable for TXQ 0 of channel 5 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
4	TQ0DMAE4	DMA TX Transfer Enable for TXQ 0 of channel 4 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
3	TQ0DMAE3	DMA TX Transfer Enable for TXQ 0 of channel 3 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
2	TQ0DMAE2	DMA TX Transfer Enable for TXQ 0 of channel 2 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
1	TQ0DMAE1	DMA TX Transfer Enable for TXQ 0 of channel 1 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled
0	TQ0DMAE0	DMA TX Transfer Enable for TXQ 0 of channel 0 0: DMA TX Transfer Request disabled 1: DMA TX Transfer Request enabled

The DMA TX Transfer Control Register bits control the start and stop of the DMA transfer operation.

RSCFDnCFDCDTTCT.CFDMAEm

DMA TX Transfer Enable for Transmit/Receive FIFO 2 of channel m.

By this bit only Transmit/Receive FIFO2 can be linked to a DMA channel, to link Transmit/Receive FIFO0 see RSCFDnCFDCDTCT.CFDMAEm.

Transmit/Receive FIFO1 cannot be linked to a DMA channel.

This bit cannot be set in GL_STOP or GL_RESET mode.

Users should not enable a DMA transfer for a Transmit/Receive FIFO that is configured as RX or GW FIFO.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTTCT.TQ3DMAEm

DMA TX Transfer Enable for TXQ 3 of channel m.

This bit cannot be set in GL_STOP or GL_RESET mode.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTTCT.TQ0DMAEm

DMA TX Transfer Enable for TXQ 0 of channel m.

This bit cannot be set in GL_STOP or GL_RESET mode.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.9.4 RSCFDnCFDCDTTSTS — DMA Transmit Transfer Status Register

Access: RSCFDnCFDCDTTSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCDTTSTSL, RSCFDnCFDCDTTSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCDTTSTSLL, RSCFDnCFDCDTTSTSLH, RSCFDnCFDCDTTSTSHL, RSCFDnCFDCDTTSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCDTTSTS: <RSCFDn_base> + 1344_H

RSCFDnCFDCDTTSTSL: <RSCFDn_base> + 1344_H
RSCFDnCFDCDTTSTSH: <RSCFDn_base> + 1346_H

RSCFDnCFDCDTTSTSLL: <RSCFDn_base> + 1344_H,
RSCFDnCFDCDTTSTSLH: <RSCFDn_base> + 1345_H,
RSCFDnCFDCDTTSTSHL: <RSCFDn_base> + 1346_H,
RSCFDnCFDCDTTSTSHH: <RSCFDn_base> + 1347_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CFDMASTS7	CFDMASTS6	CFDMASTS5	CFDMASTS4	CFDMASTS3	CFDMASTS2	CFDMASTS1	CFDMASTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TQ3DMASTS7	TQ3DMASTS6	TQ3DMASTS5	TQ3DMASTS4	TQ3DMASTS3	TQ3DMASTS2	TQ3DMASTS1	TQ3DMASTS0	TQ0DMASTS7	TQ0DMASTS6	TQ0DMASTS5	TQ0DMASTS4	TQ0DMASTS3	TQ0DMASTS2	TQ0DMASTS1	TQ0DMASTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.62 RSCFDnCFDCDTTSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	CFDMASTS7	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 7 0: DMA transfer stopped 1: DMA transfer enable
22	CFDMASTS6	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 6 0: DMA transfer stopped 1: DMA transfer enable
21	CFDMASTS5	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 5 0: DMA transfer stopped 1: DMA transfer enable
20	CFDMASTS4	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 4 0: DMA transfer stopped 1: DMA transfer enable
19	CFDMASTS3	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 3 0: DMA transfer stopped 1: DMA transfer enable
18	CFDMASTS2	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 2 0: DMA transfer stopped 1: DMA transfer enable
17	CFDMASTS1	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 1 0: DMA transfer stopped 1: DMA transfer enable
16	CFDMASTS0	DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel 0 0: DMA transfer stopped 1: DMA transfer enable
15	TQ3DMASTS7	DMA TX Transfer Status for TXQ 3 of channel 7 0: DMA transfer stopped 1: DMA transfer enable

Table 23.62 RSCFDnCFDCDTTSTS Register Contents (2/2)

Bit Position	Bit Name	Function
14	TQ3DMASTS6	DMA TX Transfer Status for TXQ 3 of channel 6 0: DMA transfer stopped 1: DMA transfer enable
13	TQ3DMASTS5	DMA TX Transfer Status for TXQ 3 of channel 5 0: DMA transfer stopped 1: DMA transfer enable
12	TQ3DMASTS4	DMA TX Transfer Status for TXQ 3 of channel 4 0: DMA transfer stopped 1: DMA transfer enable
11	TQ3DMASTS3	DMA TX Transfer Status for TXQ 3 of channel 3 0: DMA transfer stopped 1: DMA transfer enable
10	TQ3DMASTS2	DMA TX Transfer Status for TXQ 3 of channel 2 0: DMA transfer stopped 1: DMA transfer enable
9	TQ3DMASTS1	DMA TX Transfer Status for TXQ 3 of channel 1 0: DMA transfer stopped 1: DMA transfer enable
8	TQ3DMASTS0	DMA TX Transfer Status for TXQ 3 of channel 0 0: DMA transfer stopped 1: DMA transfer enable
7	TQ0DMASTS7	DMA TX Transfer Status for TXQ 0 of channel 7 0: DMA transfer stopped 1: DMA transfer enable
6	TQ0DMASTS6	DMA TX Transfer Status for TXQ 0 of channel 6 0: DMA transfer stopped 1: DMA transfer enable
5	TQ0DMASTS5	DMA TX Transfer Status for TXQ 0 of channel 5 0: DMA transfer stopped 1: DMA transfer enable
4	TQ0DMASTS4	DMA TX Transfer Status for TXQ 0 of channel 4 0: DMA transfer stopped 1: DMA transfer enable
3	TQ0DMASTS3	DMA TX Transfer Status for TXQ 0 of channel 3 0: DMA transfer stopped 1: DMA transfer enable
2	TQ0DMASTS2	DMA TX Transfer Status for TXQ 0 of channel 2 0: DMA transfer stopped 1: DMA transfer enable
1	TQ0DMASTS1	DMA TX Transfer Status for TXQ 0 of channel 1 0: DMA transfer stopped 1: DMA transfer enable
0	TQ0DMASTS0	DMA TX Transfer Status for TXQ 0 of channel 0 0: DMA transfer stopped 1: DMA transfer enable

The DMA Transfer Status Register bits show the status of the DMA transfer.

RSCFDnCFDCDTTSTS.CFDMASTSm

DMA TX Transfer Status only for Transmit/Receive FIFO 2 of channel m.

This bit is set when the RSCFDnCFDCDTTCT.CFDMAEm bit in the corresponding RSCFDnCFDCDTTCT register is set.

This bit is cleared when the RSCFDnCFDCDTTCT.CFDMAEm bit in the corresponding RSCFDnCFDCDTTCT register is cleared.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTTSTS.TQ3DMASTSm

DMA TX Transfer Status for TXQ3 of channel m.

This bit is set when the RSCFDnCFDCDTTCT.TQ3DMAEm bit in the corresponding RSCFDnCFDCDTTCT register is set.

This bit is cleared when the RSCFDnCFDCDTTCT.TQ3DMAEm bit in the corresponding RSCFDnCFDCDTTCT register is cleared.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDCDTTSTS.TQ0DMASTSm

DMA TX Transfer Status for TXQ0 of channel m.

This bit is set when the RSCFDnCFDCDTTCT.TQ0DMAEm bit in the corresponding RSCFDnCFDCDTTCT register is set.

This bit is cleared when the RSCFDnCFDCDTTCT.TQ0DMAEm bit in the corresponding RSCFDnCFDCDTTCT register is cleared.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

23.3.10 Details of Transmit Buffer-Related Registers

23.3.10.1 RSCFDnCFDTMCp — Transmit Message Buffer Control Register p

Access: RSCFDnCFDTMCp register can be read or written in 8-bit units

Address: RSCFDnCFDTMCp: <RSCFDn_base> + $02D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.63 RSCFDnCFDTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOM	TX Message Buffer One-shot Mode 0: TX Message Buffer not configured in one-shot mode 1: TX Message Buffer configured in one-shot mode
1	TMTAR	TX Message Buffer Transmission abort Request 0: TX Message Buffer transmission request abort not requested 1: TX Message Buffer transmission request abort requested
0	TMTR	TX Message Buffer Transmission Request 0: TX Message Buffer Transmission not requested 1: TX Message Buffer Transmission requested

The TX Message Buffer Control register configures the TX Message Buffer functions.
Below is an example to calculate index for U2A16:
(no_of_channels = 8)

(no_of_TMBCTRs_per_channel = No. of TX Message Buffer Control Register per Channel = 64)
Where the total number of TMBCTRs = no_of_TMBCTRs = no_of_TMBCTRs_per_channel * no_of_channels = 64 * 8 = 512
(p = TX Message Buffer Control Register index = [0…no_of_TMBCTRs-1])

RSCFDnCFDTMCp.TMOM

TX Message Buffer One-shot Mode.

If this bit is set, then the RS-CANFD module logic will attempt transmission of the message only once. If the transmission is successful, the RSCFDnCFDTMSTSp.TMTRF bits are set to 10_B or 11_B. If it is not successful due to bus error or bus arbitration lost, the transmission is automatically aborted and RSCFDnCFDTMSTSp.TMTRF bits are set to 01_B.
The RSCFDnCFDTMCp.TMOM bit will remain set if the transmission is completed successfully or aborted due to error or loss of arbitration.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should set this bit at the same time as RSCFDnCFDTMCp.TMTR bit.
Users should clear this bit by a write access.

If a message has been already requested for transmission, then users should not write to this bit until the message has been successfully transmitted or transmission has been aborted.

This bit will be automatically cleared by the RS-CANFD module logic when the RS-CANFD module enters GL_RESET mode or the related channel enters CH_RESET mode.

RSCFDnCFDTMCp.TMTAR

TX Message Buffer Transmission abort Request.

If this bit is set, then the RS-CANFD module logic will try to abort the transmission of the frame stored in the corresponding Message Buffer.
In most cases, transmission cannot be aborted if the internal scan for transmission is completed and the Message Buffer has already been selected for Transmission. In this case, frame may be transmitted successfully from the Message Buffer. The Message Buffer selection will be released by entering CH_HALT mode.
However, MB selected for transmission can be aborted by Abort request when the CAN node detects a new message on the bus (RX pin) before it can start transmission from the selected MB.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.
This bit can only be set when the related transmit request (RSCFDnCFDTMCp.TMTR) bit is set.
This bit cannot be cleared by a CPU write access.

Clearing of this bit by RS-CANFD has priority over setting by CPU write access.

This bit is automatically cleared by the RS-CANFD module logic at the end of successful transmission
This bit is automatically cleared by the RS-CANFD module logic at the end of transmission abort
This bit is automatically cleared by the RS-CANFD module logic when there is the detection of CAN bus error or arbitration loss
This bit is automatically cleared by the RS-CANFD module logic when the RS-CANFD module enters GL_RESET mode or the related channel enters CH_RESET mode.

RSCFDnCFDTMCp.TMTR

TX Message Buffer Transmission Request.

If this bit is set, then the RS-CANFD module logic will try to transmit the message stored in the corresponding Message Buffer.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.
Users cannot set this bit if the corresponding TX Message Buffer is linked to a Transmit/Receive FIFO in TX or GW mode or is a part of TX Queue.
This bit cannot be directly cleared by a CPU write access.
This bit can only be set when Transmission Result Flag bits (RSCFDnCFDTMSTSp.TMTRF) in the RSCFDnCFDTMSTSp register corresponding to the MB are cleared to 00_B.

This bit is automatically cleared by the RS-CANFD module logic at the end of successful transmission.
This bit is automatically cleared by the RS-CANFD module logic at the end of transmission abort, requested by the corresponding RSCFDnCFDTMCp.TMTAR bit.
This bit is automatically cleared by the RS-CANFD module logic when there is the detection of CAN bus error or arbitration loss if RSCFDnCFDTMCp.TMOM bit is set for the Message Buffer.

This bit is automatically cleared by the RS-CANFD module logic when the RS-CANFD module enters GL_RESET mode or the related channel enters CH_RESET mode.

23.3.10.2 RSCFDnCFDTMSTSp — Transmit Message Buffer Status Register p

Access: RSCFDnCFDTMSTSp register can be read or written in 8-bit units

Address: RSCFDnCFDTMSTSp: <RSCFDn_base> + $07D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 23.64 RSCFDnCFDTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TMTARM	TX Message Buffer Transmission abort Request Mirrored 0: TX Message Buffer transmission request abort not requested 1: TX Message Buffer transmission request abort requested
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX Message Buffer Transmission not requested 1: TX Message Buffer Transmission requested
2, 1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 00: No Result 01: Transmission aborted from the TX MB 10: Transmission successful from the TX MB & Transmission abort was not requested 11: Transmission successful from the TX MB & Transmission abort was requested
0	TMTSTS	TX Message Buffer Transmission Status 0: No transmission ongoing 1: Transmission ongoing

The TX Message Buffer Status Register shows the Transmission and Transmission abort status for the corresponding Message Buffers.
No_of_TMBSTRs_per_channel = No. of TX Message Buffer Status Registers per Channel = 64)
Where the total number of TMBSTRs = no_of_TMBSTRs = no_of_TMBSTRs_per_channel * no_of_channels = 64 * 8 = 512
(p = TX Message Buffer Status Register index = [0…no_of_TMBSTRs-1])

RSCFDnCFDTMSTSp.TMTARM

TX Message Buffer Transmission abort Request Mirrored.

This bit is set when the RSCFDnCFDTMCp.TMTAR bit in the corresponding RSCFDnCFDTMCp register is set.

This bit is cleared when the RSCFDnCFDTMCp.TMTAR bit in the corresponding RSCFDnCFDTMCp register is cleared.

RSCFDnCFDTMSTSp.TMTRM

TX Message Buffer Transmission Request Mirrored.

This bit is set when the RSCFDnCFDTMCp.TMTR bit in the corresponding RSCFDnCFDTMCp register is set.

This bit is cleared when the RSCFDnCFDTMCp.TMTR bit in the corresponding RSCFDnCFDTMCp register is cleared.

RSCFDnCFDTMSTSp.TMTRF[1:0]

TX Message Buffer Transmission Result Flag.

These bits show the result for the corresponding TX MB. The status is as follows:
00: Transmission in progress or has not been requested.
01: Transmission has been aborted from the corresponding TX MB.
10: Transmission was successful from the corresponding TX MB and TMTAR bit was not set for this TX MB.
11: Transmission was successful from the corresponding TX MB. But, TMTAR bit was set for this TX MB.

Users can only write to this bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

These bits will be cleared automatically when the RS-CANFD module enters GL_RESET or the related channel enters CH_RESET mode.

RSCFDnCFDTMSTSp.TMTSTS

TX Message Buffer Transmission Status.

This bit is set automatically at the start of the transmission from the corresponding TX Message Buffer.

This bit is cleared automatically when the transmission stops.
This bit is cleared automatically when the RS-CANFD module enters GL_RESET mode.
This bit is cleared automatically when the related channel enters CH_RESET mode.

23.3.10.3 RSCFDnCFDTMIECy — Transmit Message Buffer Interrupt Enable Configuration Register y

Access: RSCFDnCFDTMIECy register can be read or written in 32-bit units
RSCFDnCFDTMIECyL, RSCFDnCFDTMIECyH register can be read or written in 16-bit units
RSCFDnCFDTMIECyLL, RSCFDnCFDTMIECyLH, RSCFDnCFDTMIECyHL, RSCFDnCFDTMIECyHH register can be read or written in 8-bit units

Address: RSCFDnCFDTMIECy: <RSCFDn_base> + $0F50_H$ + (04_H × y)

RSCFDnCFDTMIECyL: <RSCFDn_base> + $0F50_H$ + (04_H × y),
RSCFDnCFDTMIECyH: <RSCFDn_base> + $0F52_H$ + (04_H × y)

RSCFDnCFDTMIECyLL: <RSCFDn_base> + $0F50_H$ + (04_H × y),
RSCFDnCFDTMIECyLH: <RSCFDn_base> + $0F51_H$ + (04_H × y),
RSCFDnCFDTMIECyHL: <RSCFDn_base> + $0F52_H$ + (04_H × Y),
RSCFDnCFDTMIECyHH: <RSCFDn_base> + $0F53_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.65 RSCFDnCFDTMIECy Register Contents

Bit Position	Bit Name	Function
31 to 0	TMIEp[31:0]	TX Message Buffer Interrupt Enable 0: TX Message Buffer Interrupt disabled for corresponding TX message buffer 1: TX Message Buffer Interrupt enabled for corresponding TX message buffer

Below is an example to calculate ymin and ymax for U2A16:
(no_of_channels = 8)
(no_of_TXMBs_per_channel = 64)
(no_of_bits_per_register = 32)
(no_of_TXMBs = No. of TX Message Buffers = no_of_channels*no_of_TXMBs_per_channel = 512)
(no_of_RSCFDnCFDTMIEC = No. of RSCFDnCFDTMIEC = no_of_TXMBs/
no_of_bits_per_register = 512 / 32 = 16)
(y = RSCFDnCFDTMIEC index = [0...no_of_RSCFDnCFDTMIEC-1])
y = [0 to 15]
When m=0, y=[0, 1] ymin=0, ymax=1
When m=1, y=[2, 3] ymin=2, ymax=3
When m=2, y=[4, 5] ymin=4, ymax=5
When m=3, y=[6, 7] ymin=6, ymax=7
When m=4, y=[8, 9] ymin=8, ymax=9
When m=5, y=[10, 11] ymin=10, ymax=11
When m=6, y=[12, 13] ymin=12, ymax=13
When m=7, y=[14, 15] ymin=14, ymax=15

RSCFDnCFDTMIECy.TMIEp[31:0]

TX Message Buffer Interrupt Enable

If this bit is set, then an interrupt will be generated at the end of a successful transmission from the corresponding Message Buffer.

Refer to **Section 23.8** for TX Message Buffer Interrupt specification.

An example in U2A16 for alignment of the bits is as shown in **Table 23.66**.

Table 23.66 Alignment of TMIE bits

Bit position	p = TX Message Buffer Number
m*64-ymin*32	m*64+0
m*64+1-ymin*32	m*64+1
.	.
.	.
m*64+31-ymin*32	m*64+31
m*64+32-ymax*32	m*64+32
m*64+33-ymax*32	m*64+33
.	.
.	.
m*64+62-ymax*32	m*64+62
m*64+63-ymax*32	m*64+63

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to these bits when the related CAN_channel is in CH_STOP mode.

Users should not write to these bits if the corresponding TX Message Buffer is part of a TX Queue.

Users should not write to these bits if the corresponding TX Message Buffer is linked to a Transmit/ Receive FIFO (via RSCFDnCFDCFCCk.CFTML bits)

23.3.11 Details of Transmit Buffer Status-Related Registers

23.3.11.1 RSCFDnCFDTMTRSTSy — Transmit Message Buffer Transmission Request Status Register y

Access: RSCFDnCFDTMTRSTSy register is a read-only register that can be read in 32-bit units
RSCFDnCFDTMTRSTSyL, RSCFDnCFDTMTRSTSyH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTMTRSTSyLL, RSCFDnCFDTMTRSTSyLH, RSCFDnCFDTMTRSTSyHL, RSCFDnCFDTMTRSTSyHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTMTRSTSy: <RSCFDn_base> + $0CD0_H$ + (04_H × y)

RSCFDnCFDTMTRSTSyL: <RSCFDn_base> + $0CD0_H$ + (04_H × y),
RSCFDnCFDTMTRSTSyH: <RSCFDn_base> + $0CD2_H$ + (04_H × y)

RSCFDnCFDTMTRSTSyLL: <RSCFDn_base> + $0CD0_H$ + (04_H × y),
RSCFDnCFDTMTRSTSyLH: <RSCFDn_base> + $0CD1_H$ + (04_H × y),
RSCFDnCFDTMTRSTSyHL: <RSCFDn_base> + $0CD2_H$ + (04_H × y),
RSCFDnCFDTMTRSTSyHH: <RSCFDn_base> + $0CD3_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDTMTRSTSp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDTMTRSTSp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.67 RSCFDnCFDTMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	CFDTMTRSTSp [31:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX Message Buffer 1: Transmission requested for corresponding TX Message Buffer

Below is an example to calculate ymin and ymax for U2A16:
(no_of_channels = 8)
(no_of_TXMBs_per_channel = 64)
(no_of_bits_per_register = 32)
(no_of_TXMBs = No. of TX Message Buffers = no_of_channels*no_of_TXMBs_per_channel = 512)
(no_of_CFDTMTRSTS = No. of CFDTMTRSTS = no_of_TXMBs/no_of_bits_per_register = 512 / 32 = 16)
(y = CFDTMTRSTS index = [0... . no_of_CFDTMTRSTS -1])
y = [0 to 15]
When m=0, y=[0, 1] ymin=0, ymax=1
When m=1, y=[2, 3] ymin=2, ymax=3
When m=2, y=[4, 5] ymin=4, ymax=5
When m=3, y=[6, 7] ymin=6, ymax=7
When m=4, y=[8, 9] ymin=8, ymax=9
When m=5, y=[10, 11] ymin=10, ymax=11
When m=6, y=[12, 13] ymin=12, ymax=13
When m=7, y=[14, 15] ymin=14, ymax=15

RSCFDnCFDTMTRSTSy.CFDTMTRSTSp[31:0]

TX Message Buffer Transmission Request Status

These bits show the status of the RSCFDnCFDTMCp.TMTR bits of the TX Message Buffer Control Register.
An example in U2A16 for alignment of the bits is as shown in **Table 23.68**.

Table 23.68 Alignment of CFDTMTRSTS mirror bits

Bit position	p = TX Message Buffer Number
m*64-ymin*32	m*64+0
m*64+1-ymin*32	m*64+1
.	.
.	.
m*64+31-ymin*32	m*64+31
m*64+32-ymax*32	m*64+32
m*64+33-ymax*32	m*64+33
.	.
.	.
m*64+62-ymax*32	m*64+62
m*64+63-ymax*32	m*64+63

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Register, only when the Message Buffer is not belonging to a TX Queue.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Message Buffer Control Register.
Each bit is cleared automatically when the RS-CANFD module enters GL_RESET or CH_RESET mode.

23.3.11.2 RSCFDnCFDTMTARSTSy — Transmit Message Buffer Transmission Abort Request Status Register y

Access: RSCFDnCFDTMTARSTSy register is a read-only register that can be read in 32-bit units
RSCFDnCFDTMTARSTSyL, RSCFDnCFDTMTARSTSyH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTMTARSTSyLL, RSCFDnCFDTMTARSTSyLH, RSCFDnCFDTMTARSTSyHL, RSCFDnCFDTMTARSTSyHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTMTARSTSy: <RSCFDn_base> + $0D70_H$ + (04_H × y)

RSCFDnCFDTMTARSTSyL: <RSCFDn_base> + $0D70_H$ + (04_H × y),
RSCFDnCFDTMTARSTSyH: <RSCFDn_base> + $0D72_H$ + (04_H × y)

RSCFDnCFDTMTARSTSyLL: <RSCFDn_base> + $0D70_H$ + (04_H × y),
RSCFDnCFDTMTARSTSyLH: <RSCFDn_base> + $0D71_H$ + (04_H × y),
RSCFDnCFDTMTARSTSyHL: <RSCFDn_base> + $0D72_H$ + (04_H × y),
RSCFDnCFDTMTARSTSyHH: <RSCFDn_base> + $0D23_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDTMTARSTSp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDTMTARSTSp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.69 RSCFDnCFDTMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	CFDTMTARSTSp [31:0]	TX Message Buffer Transmission abort Request Status 0: Transmission abort not requested for corresponding TX Message Buffer 1: Transmission abort requested for corresponding TX Message Buffer

Below is an example to calculate ymin and ymax for U2A16:
(no_of_channels = 8)
(no_of_TXMBs_per_channel = 64)
(no_of_bits_per_register = 32)
(no_of_TXMBs = No. of TX Message Buffers = no_of_channels*no_of_TXMBs_per_channel = 512)
(no_of_TMTARSTS = No. of TMTARSTS = no_of_TXMBs/no_of_bits_per_register = 512 / 32 = 16)
(y = CFDTMTARSTS index = [0...no_of_TMTARSTS-1])
y = [0 to 15]
When m=0, y=[0, 1] ymin=0, ymax=1
When m=1, y=[2, 3] ymin=2, ymax=3
When m=2, y=[4, 5] ymin=4, ymax=5
When m=3, y=[6, 7] ymin=6, ymax=7
When m=4, y=[8, 9] ymin=8, ymax=9
When m=5, y=[10, 11] ymin=10, ymax=11
When m=6, y=[12, 13] ymin=12, ymax=13
When m=7, y=[14, 15] ymin=14, ymax=15

RSCFDnCFDTMTARSTSy.CFDTMTARSTSp[31:0]

TX Message Buffer Transmission abort Request Status

These bits show the status of the RSCFDnCFDTMCp.TMTAR bits of the TX Message Buffer Control Register.

An example in U2A16 for alignment of the bits is as shown in **Table 23.70**.

Table 23.70 Alignment of RSCFDnCFDTMTARSTS mirror bits

Bit position	p = TX Message Buffer Number
m*64-ymin*32	m*64+0
m*64+1-ymin*32	m*64+1
.	.
.	.
m*64+31-ymin*32	m*64+31
m*64+32-ymax*32	m*64+32
m*64+33-ymax*32	m*64+33
.	.
.	.
m*64+62-ymax*32	m*64+62
m*64+63-ymax*32	m*64+63

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, also when the Message Buffer is belonging to a TX Queue.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Message Buffer Control Register.
Each bit is cleared automatically when the RS-CANFD module enters GL_RESET or CH_RESET mode.
If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

23.3.11.3 RSCFDnCFDTMTCSTSy — Transmit Message Buffer Transmission Complete Status Register y

Access: RSCFDnCFDTMTCSTSy register is a read-only register that can be read in 32-bit units
RSCFDnCFDTMTCSTSyL, RSCFDnCFDTMTCSTSyH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTMTCSTSyLL, RSCFDnCFDTMTCSTSyLH, RSCFDnCFDTMTCSTSyHL, RSCFDnCFDTMTCSTSyHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTMTCSTSy: <RSCFDn_base> + $0E10_H$ + (04_H × y)

RSCFDnCFDTMTCSTSyL: <RSCFDn_base> + $0E10_H$ + (04_H × y),
RSCFDnCFDTMTCSTSyH: <RSCFDn_base> + $0E12_H$ + (04_H × y)

RSCFDnCFDTMTCSTSyLL: <RSCFDn_base> + $0E10_H$ + (04_H × y),
RSCFDnCFDTMTCSTSyLH: <RSCFDn_base> + $0E11_H$ + (04_H × y),
RSCFDnCFDTMTCSTSyHL: <RSCFDn_base> + $0E12_H$ + (04_H × y),
RSCFDnCFDTMTCSTSyHH: <RSCFDn_base> + $0E13_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDTMTCSTSp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDTMTCSTSp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.71 RSCFDnCFDTMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	CFDTMTCSTSp [31:0]	TX Message Buffer Transmission Completion Status 0: Transmission not complete for corresponding TX Message Buffer 1: Transmission completed for corresponding TX Message Buffer

Below is an example to calculate ymin and ymax for U2A16:
(no_of_channels = 8)
(no_of_TXMBs_per_channel = 64)
(no_of_bits_per_register = 32)
(no_of_TXMBs = No. of TX Message Buffers = no_of_channels*no_of_TXMBs_per_channel = 512)
(no_of_CFDTMTCSTS = No. of CFDTMTCSTS = no_of_TXMBs/no_of_bits_per_register = 512 / 32 = 16)
(y = CFDTMTCSTS index = [0...no_of_CFDTMTCSTS-1])
y = [0...15]
When m=0, y=[0, 1] ymin=0, ymax=1
When m=1, y=[2, 3] ymin=2, ymax=3
When m=2, y=[4, 5] ymin=4, ymax=5
When m=3, y=[6, 7] ymin=6, ymax=7
When m=4, y=[8, 9] ymin=8, ymax=9
When m=5, y=[10, 11] ymin=10, ymax=11
When m=6, y=[12, 13] ymin=12, ymax=13
When m=7, y=[14, 15] ymin=14, ymax=15

RSCFDnCFDTMTCSTSy.CFDTMTCSTSp[31:0]

TX Message Buffer Transmission Completion Status

These bits show the status of successful completion of the TX Message Buffer Status Register.

An example in U2A16 for alignment of the bits is as shown in **Table 23.72**.

Table 23.72 Alignment of CFDTMTCSTS mirror bits

Bit position	p = TX Message Buffer Number
m*64-ymin*32	m*64+0
m*64+1-ymin*32	m*64+1
.	.
.	.
m*64+31-ymin*32	m*64+31
m*64+32-ymax*32	m*64+32
m*64+33-ymax*32	m*64+33
.	.
.	.
m*64+62-ymax*32	m*64+62
m*64+63-ymax*32	m*64+63

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Message Buffer Status Register.
Each bit is cleared automatically when the RS-CANFD module enters GL_RESET or CH_RESET mode.
If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

23.3.11.4 RSCFDnCFDTMTASTSy — Transmit Message Buffer Transmission Abort Status Register y

Access: RSCFDnCFDTMTASTSy register is a read-only register that can be read in 32-bit units
RSCFDnCFDTMTASTSyL, RSCFDnCFDTMTASTSyH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTMTASTSyLL, RSCFDnCFDTMTASTSyLH, RSCFDnCFDTMTASTSyHL, RSCFDnCFDTMTASTSyHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTMTASTSy: <RSCFDn_base> + $0EB0_H$ + (04_H × y)

RSCFDnCFDTMTASTSyL: <RSCFDn_base> + $0EB0_H$ + (04_H × y),
RSCFDnCFDTMTASTSyH: <RSCFDn_base> + $0EB2_H$ + (04_H × y)

RSCFDnCFDTMTASTSyLL: <RSCFDn_base> + $0EB0_H$ + (04_H × y),
RSCFDnCFDTMTASTSyLH: <RSCFDn_base> + $0EB1_H$ + (04_H × y),
RSCFDnCFDTMTASTSyHL: <RSCFDn_base> + $0EB2_H$ + (04_H × y),
RSCFDnCFDTMTASTSyHH: <RSCFDn_base> + $0EB3_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDTMTASTSp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDTMTASTSp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.73 RSCFDnCFDTMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 0	CFDTMTASTSp [31:0]	TX Message Buffer Transmission abort Status 0: Transmission not aborted for corresponding TX Message Buffer 1: Transmission aborted for corresponding TX Message Buffer

Below is an example to calculate ymin and ymax for U2A16:
(no_of_channels = 8)
(no_of_TXMBs_per_channel = 64)
(no_of_bits_per_register = 32)
(no_of_TXMBs = No. of TX Message Buffers = no_of_channels*no_of_TXMBs_per_channel = 512)
(no_of_CFDTMTASTS = No. of CFDTMTASTS = no_of_TXMBs/no_of_bits_per_register = 512 / 32 = 16)
(y = CFDTMTASTS index = [0...no_of_CFDTMTASTS-1])
y = [0 to 15]
When m=0, y=[0, 1] ymin=0, ymax=1
When m=1, y=[2, 3] ymin=2, ymax=3
When m=2, y=[4, 5] ymin=4, ymax=5
When m=3, y=[6, 7] ymin=6, ymax=7
When m=4, y=[8, 9] ymin=8, ymax=9
When m=5, y=[10, 11] ymin=10, ymax=11
When m=6, y=[12, 13] ymin=12, ymax=13
When m=7, y=[14, 15] ymin=14, ymax=15

RSCFDnCFDTMTASTSy.CFDTMTASTSp[31:0]

TX Message Buffer Transmission abort Status

These bits show the status of the successful transmission abort of the corresponding TX Message Buffer.

An example in U2A16 for alignment of the bits is as shown in **Table 23.74**.

Table 23.74 Alignment of CFDTMTASTS mirror bits

Bit position	p = TX Message Buffer Number
m*64-ymin*32	m*64+0
m*64+1-ymin*32	m*64+1
.	.
.	.
m*64+31-ymin*32	m*64+31
m*64+32-ymax*32	m*64+32
m*64+33-ymax*32	m*64+33
.	.
.	.
m*64+62-ymax*32	m*64+62
m*64+63-ymax*32	m*64+63

Each bit is set automatically when the RSCFDnCFDTMSTSp.TMTRF bits are set to 01_B in the corresponding TX Message Buffer Status Register.

Each bit is cleared automatically when the RSCFDnCFDTMSTSp.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register.
Each bit is cleared automatically when the RS-CANFD module enters GL_RESET or CH_RESET mode.
If a CAN channel enters CH_RESET mode, then the bits related to that channel will be cleared.

23.3.12 Details of Transmit Queue-Related Registers

23.3.12.1 RSCFDnCFDTXQCC0m — Transmit Queue Configuration / Control Register 0 m

Access: RSCFDnCFDTXQCC0m register can be read or written in 32-bit units
RSCFDnCFDTXQCC0mL, RSCFDnCFDTXQCC0mH register can be read or written in 16-bit units
RSCFDnCFDTXQCC0mLL, RSCFDnCFDTXQCC0mLH, RSCFDnCFDTXQCC0mHL,
RSCFDnCFDTXQCC0mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQCC0m: <RSCFDn_base> + 1000_H + (04_H × m)

RSCFDnCFDTXQCC0mL: <RSCFDn_base> + 1000_H + (04_H × m),
RSCFDnCFDTXQCC0mH: <RSCFDn_base> + 1002_H + (04_H × m)

RSCFDnCFDTXQCC0mLL: <RSCFDn_base> + 1000_H + (04_H × m),
RSCFDnCFDTXQCC0mLH: <RSCFDn_base> + 1001_H + (04_H × m),
RSCFDnCFDTXQCC0mHL: <RSCFDn_base> + 1002_H + (04_H × m),
RSCFDnCFDTXQCC0mHH: <RSCFDn_base> + 1003_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W

Table 23.75 RSCFDnCFDTXQCC0m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX Interrupt generation disabled 1: One Frame TX Interrupt generation enabled
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX Interrupt generation disabled 1: One Frame RX Interrupt generation enabled
16	TXQFIE	TXQ Full interrupt Enable 0: TX Queue Full Interrupt generation disabled 1: TX Queue Full Interrupt generation enabled
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	TXQDC[4:0]	TX Queue Depth Configuration 00000: 0 messages 00001: Setting prohibited 00010: 3 messages 00011: 4 messages : 11110: 31 messages 11111: 32 messages
7	TXQIM	TX Queue Interrupt Mode 0: when the last message is successfully transmitted 1: At every successful transmission
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 23.75 RSCFDnCFDTXQCC0m Register Contents (2/2)

Bit Position	Bit Name	Function
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX Interrupt disabled 1: TX Queue TX Interrupt enabled
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled

The TX Queue Configuration/Control Register is used to configure the TX Queue transmission.

TXQ0 is composed of TXMB0 to TXMB31 (at the maximum) when TXQE is enabled.

RSCFDnCFDTXQCC0m.TXQOFTXIE

TXQ One Frame Transmission Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS0m.TXQOFTXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC0m.TXQOFRXIE

TXQ One Frame Reception Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS0m.TXQOFRXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC0m.TXQGWE=1).

RSCFDnCFDTXQCC0m.TXQFIE

TXQ Full Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS0m.TXQFIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC0m.TXQGWE=1).

RSCFDnCFDTXQCC0m.TXQDC[4:0]

TX Queue Depth Configuration.

These bits select the depth of the transmission queue. The Message Buffer selection starts from MB[0] up to MB[31] depending upon the configured depth.

When users use TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.

Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.
Users cannot write to these bits when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to these bits when the related RS-CANFD channel is in CH_STOP.

RSCFDnCFDTXQCC0m.TXQIM

TX Queue Interrupt Mode.

This bit selects the Interrupt generation condition for the TX Queue.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDTXQCC0m.TXQTXIE

TX Queue TX Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the TXQIM bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC0m.TXQOWE

TX Queue Overwrite Mode Enable.

When this bit is set, the TX queue is in TX queue overwrite mode.

An overwrite function is valid when the same ID as ID of the data written in from the gateway or CPU is in TX Queue, e.g. when a frame is received and is stored into the TX queue, if a message with the same ID has been stored in the TX queue, the old message will be overwritten by the new message.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When users use the function in GW mode, the depth of TXQ (RSCFDnCFDTXQCC0m.TXQDC) should be configured to the value which is the various number of ID which is used in the TX queue plus 3.

Then the function is valid for the standard ID frame and is invalid for the extended ID frame.

Users should not write change for this bit when the RSCFDnCFDTXQCC0m.TXQE bit is 1_B.

RSCFDnCFDTXQCC0m.TXQGWE

TX Queue Gateway Mode Enable.

When this bit is set, the TX queue is in TX queue GW mode.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When this bit is set, CPU must not access the TX queue.

RSCFDnCFDTXQCC0m.TXQE

TX Queue Enable.

Users cannot write to this bit when the related RS-CANFD channel is in CH_RESET mode.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
This bit cannot be set if the configured TX Queue depth is 0_H (RSCFDnCFDTXQCC0m.TXQDC = 0_H).
Users cannot write to this bit when the related channel is in CH_STOP mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.12.2 RSCFDnCFDTXQCC1m — Transmit Queue Configuration / Control Register 1 m

Access: RSCFDnCFDTXQCC1m register can be read or written in 32-bit units
RSCFDnCFDTXQCC1mL, RSCFDnCFDTXQCC1mH register can be read or written in 16-bit units
RSCFDnCFDTXQCC1mLL, RSCFDnCFDTXQCC1mLH, RSCFDnCFDTXQCC1mHL,
RSCFDnCFDTXQCC1mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQCC1m: <RSCFDn_base> + 1060_H + (04_H × m)

RSCFDnCFDTXQCC1mL: <RSCFDn_base> + 1060_H + (04_H × m),
RSCFDnCFDTXQCC1mH: <RSCFDn_base> + 1062_H + (04_H × m)

RSCFDnCFDTXQCC1mLL: <RSCFDn_base> + 1060_H + (04_H × m),
RSCFDnCFDTXQCC1mLH: <RSCFDn_base> + 1061_H + (04_H × m),
RSCFDnCFDTXQCC1mHL: <RSCFDn_base> + 1062_H + (04_H × m),
RSCFDnCFDTXQCC1mHH: <RSCFDn_base> + 1063_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W

Table 23.76 RSCFDnCFDTXQCC1m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX Interrupt generation disabled 1: One Frame TX Interrupt generation enabled
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX Interrupt generation disabled 1: One Frame RX Interrupt generation enabled
16	TXQFIE	TXQ Full interrupt Enable 0: TX Queue Full Interrupt generation disabled 1: TX Queue Full Interrupt generation enabled
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	TXQDC[4:0]	TX Queue Depth Configuration 00000: 0 messages 00001: Setting prohibited 00010: 3 messages 00011: 4 messages : 11110: 31 messages 11111: 32 messages
7	TXQIM	TX Queue Interrupt Mode 0: when the last message is successfully transmitted 1: At every successful transmission
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX Interrupt disabled 1: TX Queue TX Interrupt enabled

Table 23.76 RSCFDnCFDTXQCC1m Register Contents (2/2)

Bit Position	Bit Name	Function
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled

The TX Queue Configuration / Control Register is used to configure the TX Queue transmission.

TXQ1 is composed of TXMB31 to TXMB0 (at the maximum) when TXQE is enabled.

RSCFDnCFDTXQCC1m.TXQOFTXIE

TXQ One Frame Transmission Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS1m.TXQOFTXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC1m.TXQOFRXIE

TXQ One Frame Reception Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS1m.TXQOFRXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC1m.TXQGWE=1).

RSCFDnCFDTXQCC1m.TXQFIE

TXQ Full Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS1m.TXQFIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC1m.TXQGWE=1).

RSCFDnCFDTXQCC1m.TXQDC[4:0]

TX Queue Depth Configuration.

These bits select the depth of the transmission queue. The Message Buffer selection starts from MB[31] down to MB[0] depending upon the configured depth.

When users use TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.
Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.
Users cannot write to these bits when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to these bits when the related RS-CANFD channel is in CH_STOP.

RSCFDnCFDTXQCC1m.TXQIM

TX Queue Interrupt Mode.

This bit selects the Interrupt generation condition for the TX Queue.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDTXQCC1m.TXQTXIE

TX Queue TX Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the TXQIM bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC1m.TXQOWE

TX Queue Overwrite Mode Enable.

When this bit is set, the TX queue is in TX queue overwrite mode.

An overwrite function is valid when the same ID as ID of the data written in from the gateway or CPU is in TX Queue, e.g. when a frame is received and is stored into the TX queue, if a message with the same ID has been stored in the TX queue, the old message will be overwritten by the new message.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When users use the function in GW mode, the depth of TXQ (RSCFDnCFDTXQCC1m.TXQDC) should be configured to the value which is the various number of ID which is used in the TX queue plus 3.

Then the function is valid for the standard ID frame and is invalid for the extended ID frame.

Users should not write change for this bit when the RSCFDnCFDTXQCC1m.TXQE bit is 1_B.

RSCFDnCFDTXQCC1m.TXQGWE

TX Queue Gateway Mode Enable.

When this bit is set, the TX queue is in TX queue GW mode.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When this bit is set, CPU must not access the TX queue.

RSCFDnCFDTXQCC1m.TXQE

TX Queue Enable.

Users cannot write to this bit when the related RS-CANFD channel is in CH_RESET mode.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
This bit cannot be set if the configured TX Queue depth is 0_H (RSCFDnCFDTXQCC1m.TXQDC = 0_H).
Users cannot write to this bit when the related channel is in CH_STOP mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.12.3 RSCFDnCFDTXQCC2m — Transmit Queue Configuration / Control Register 2 m

Access: RSCFDnCFDTXQCC2m register can be read or written in 32-bit units
RSCFDnCFDTXQCC2mL, RSCFDnCFDTXQCC2mH register can be read or written in 16-bit units
RSCFDnCFDTXQCC2mLL, RSCFDnCFDTXQCC2mLH, RSCFDnCFDTXQCC2mHL,
RSCFDnCFDTXQCC2mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQCC2m: <RSCFDn_base> + $10C0_H$ + (04_H × m)

RSCFDnCFDTXQCC2mL: <RSCFDn_base> + $10C0_H$ + (04_H × m),
RSCFDnCFDTXQCC2mH: <RSCFDn_base> + $10C2_H$ + (04_H × m)

RSCFDnCFDTXQCC2mLL: <RSCFDn_base> + $10C0_H$ + (04_H × m),
RSCFDnCFDTXQCC2mLH: <RSCFDn_base> + $10C1_H$ + (04_H × m),
RSCFDnCFDTXQCC2mHL: <RSCFDn_base> + $10C2_H$ + (04_H × m),
RSCFDnCFDTXQCC2mHH: <RSCFDn_base> + $10C3_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W

Table 23.77 RSCFDnCFDTXQCC2m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX Interrupt generation disabled 1: One Frame TX Interrupt generation enabled
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX Interrupt generation disabled 1: One Frame RX Interrupt generation enabled
16	TXQFIE	TXQ Full interrupt Enable 0: TX Queue Full Interrupt generation disabled 1: TX Queue Full Interrupt generation enabled
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	TXQDC[4:0]	TX Queue Depth Configuration 00000: 0 messages 00001: Setting prohibited 00010: 3 messages 00011: 4 messages : 11110: 31 messages 11111: 32 messages
7	TXQIM	TX Queue Interrupt Mode 0: when the last message is successfully transmitted 1: At every successful transmission
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX Interrupt disabled 1: TX Queue TX Interrupt enabled

Table 23.77 RSCFDnCFDTXQCC2m Register Contents (2/2)

Bit Position	Bit Name	Function
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled

The TX Queue Configuration / Control Register is used to configure the TX Queue transmission.

TXQ2 is composed of TXMB32 to TXMB63 (at the maximum) when TXQE is enabled.

RSCFDnCFDTXQCC2m.TXQOFTXIE

TXQ One Frame Transmission Interrupt Enable

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS2m.TXQOFTXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC2m.TXQOFRXIE

TXQ One Frame Reception Interrupt Enable

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS2m.TXQOFRXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC2m.TXQGWE=1)

RSCFDnCFDTXQCC2m.TXQFIE

TXQ Full Interrupt Enable

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS2m.TXQFIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should write 1 to this bit only when the Gateway mode (RSCFDnCFDTXQCC2m.TXQGWE=1)

RSCFDnCFDTXQCC2m.TXQDC[4:0]

TX Queue Depth Configuration

These bits select the depth of the transmission queue. The Message Buffer selection starts from MB[32] up to MB[63] depending upon the configured depth.

When users use TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.
Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.
Users cannot write to these bits when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to these bits when the related RS-CANFD channel is in CH_STOP.

RSCFDnCFDTXQCC2m.TXQIM

TX Queue Interrupt Mode

This bit selects the Interrupt generation condition for the TX Queue.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDTXQCC2m.TXQTXIE

TX Queue TX Interrupt Enable

If this bit is set, then an interrupt will be generated based on the setting of the TXQIM bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC2m.TXQOWE

TX Queue Overwrite Mode Enable

When this bit is set, the TX queue is in TX queue overwrite mode.

An overwrite function is valid when the same ID as ID of the data written in from the gateway or CPU is in TX Queue, e.g. when a frame is received and is stored into the TX queue, if a message with the same ID has been stored in the TX queue, the old message will be overwritten by the new message.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When users use the function in GW mode, the depth of TXQ (RSCFDnCFDTXQCC2m.TXQDC) should be configured to the value which is the various number of ID which is used in the TX queue plus 3.

Then the function is valid for the standard ID frame and is invalid for the extended ID frame.

Users should not write change for this bit when the RSCFDnCFDTXQCC2m.TXQE bit is 1_B.

RSCFDnCFDTXQCC2m.TXQGWE

TX Queue Gateway Mode Enable

When this bit is set, the TX queue is in TX queue GW mode.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When this bit is set, CPU must not access the TX queue.

RSCFDnCFDTXQCC2m.TXQE

TX Queue Enable

Users cannot write to this bit when the related RS-CANFD channel is in CH_RESET mode.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
This bit cannot be set if the configured TX Queue depth is 0_H (RSCFDnCFDTXQCC2m.TXQDC = 0_H).
Users cannot write to this bit when the related channel is in CH_STOP mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.12.4 RSCFDnCFDTXQCC3m — Transmit Queue Configuration / Control Register 3 m

Access: RSCFDnCFDTXQCC3m register can be read or written in 32-bit units
RSCFDnCFDTXQCC3mL, RSCFDnCFDTXQCCnH register can be read or written in 16-bit units
RSCFDnCFDTXQCC3mLL, RSCFDnCFDTXQCCnLH, RSCFDnCFDTXQCC3mHL,
RSCFDnCFDTXQCC3mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQCC3m: <RSCFDn_base> + 1120_H + (04_H × m)

RSCFDnCFDTXQCC3mL: <RSCFDn_base> + 1120_H + (04_H × m),
RSCFDnCFDTXQCC3mH: <RSCFDn_base> + 1122_H + (04_H × m)

RSCFDnCFDTXQCC3mLL: <RSCFDn_base> + 1120_H + (04_H × m),
RSCFDnCFDTXQCC3mLH: <RSCFDn_base> + 1121_H + (04_H × m),
RSCFDnCFDTXQCC3mHL: <RSCFDn_base> + 1122_H + (04_H × m),
RSCFDnCFDTXQCC3mHH: <RSCFDn_base> + 1123_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R	R/W

Table 23.78 RSCFDnCFDTXQCC3m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX Interrupt generation disabled 1: One Frame TX Interrupt generation enabled
17 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	TXQDC[4:0]	TX Queue Depth Configuration 00000: 0 messages 00001: Setting prohibited 00010: 3 messages 00011: 4 messages : 11110: 31 messages 11111: 32 messages
7	TXQIM	TX Queue Interrupt Mode 0: when the last message is successfully transmitted 1: At every successful transmission
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX Interrupt disabled 1: TX Queue TX Interrupt enabled
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled

Table 23.78 RSCFDnCFDTXQCC3m Register Contents (2/2)

Bit Position	Bit Name	Function
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled

The TX Queue Configuration / Control Register is used to configure the TX Queue transmission.

TXQ3 is composed of TXMB63 to TXMB32 (at the maximum) when TXQE is enabled.

RSCFDnCFDTXQCC3m.TXQOFTXIE

TXQ One Frame Transmission Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the RSCFDnCFDTXQSTS3m.TXQOFTXIF bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC3m.TXQDC[4:0]

TX Queue Depth Configuration.

These bits select the depth of the transmission queue. The Message Buffer selection starts from MB[63] down to MB[32] depending upon the configured depth.

When users use TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.
Users cannot write to these bits when the RS-CANFD module is in GL_STOP mode.
Users cannot write to these bits when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to these bits when the related RS-CANFD channel is in CH_STOP.

RSCFDnCFDTXQCC3m.TXQIM

TX Queue Interrupt Mode.

This bit selects the Interrupt generation condition for the TX Queue.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_HALT or CH_COMMUNICATION mode.

RSCFDnCFDTXQCC3m.TXQTXIE

TX Queue TX Interrupt Enable.

If this bit is set, then an interrupt will be generated based on the setting of the TXQIM bit.

Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when the RS-CANFD module is in CH_STOP mode.

RSCFDnCFDTXQCC3m.TXQOWE

TX Queue Overwrite Mode Enable.

When this bit is set, the TX queue is in TX queue overwrite mode.

An overwrite function is valid when the same ID as ID of the data written in from CPU is in TX Queue, e.g. when a frame is received and is stored into the TX queue, if a message with the same ID has been stored in the TX queue, the old message will be overwritten by the new message.

Users cannot write to the bit when the RS-CANFD module is in GL_STOP mode.

Users cannot write to the bit when the related RS-CANFD channel is in CH_HALT or CH_COMMUNICATION mode.

Users should not write to the bit when the related RS-CANFD channel is in CH_STOP.

When users use the function, the depth of TXQ (RSCFDnCFDTXQCC3m.TXQDC) should be configured to the value which is the various number of ID which is used in the TX queue plus 3.

Then the function is valid for the standard ID frame and is invalid for the extended ID frame.

Users should not write change for this bit when the RSCFDnCFDTXQCC3m.TXQE bit is set to 1_B.

RSCFDnCFDTXQCC3m.TXQE

TX Queue Enable.

Users cannot write to this bit when the related RS-CANFD channel is in CH_RESET mode.
Users cannot write to this bit when the RS-CANFD module is in GL_STOP mode.
This bit cannot be set if the configured TX Queue depth is 0_H (RSCFDnCFDTXQCC3m.TXQDC = 0_H).
Users cannot write to this bit when the related channel is in CH_STOP mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.12.5 RSCFDnCFDTXQSTS0m — Transmit Queue Status Register 0 m

Access: RSCFDnCFDTXQSTS0m register can be read or written in 32-bit units
RSCFDnCFDTXQSTS0mL, RSCFDnCFDTXQSTSnH register can be read or written in 16-bit units
RSCFDnCFDTXQSTS0mLL, RSCFDnCFDTXQSTS0mLH, RSCFDnCFDTXQSTS0mHL, RSCFDnCFDTXQSTS0mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQSTS0m: <RSCFDn_base> + 1020_H + (04_H × m)

RSCFDnCFDTXQSTS0mL: <RSCFDn_base> + 1020_H + (04_H × m),
RSCFDnCFDTXQSTS0mH: <RSCFDn_base> + 1022_H + (04_H × m)

RSCFDnCFDTXQSTS0mLL: <RSCFDn_base> + 1020_H + (04_H × m),
RSCFDnCFDTXQSTS0mLH: <RSCFDn_base> + 1021_H + (04_H × m),
RSCFDnCFDTXQSTS0mHL: <RSCFDn_base> + 1022_H + (04_H × m),
RSCFDnCFDTXQSTS0mHH: <RSCFDn_base> + 1023_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 23.79 RSCFDnCFDTXQSTS0m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	TXQMOW	TXQ message overwrite 0: No Message overwrite in TXQ 1: Message overwrite in TXQ
19	TXQMLT	TXQ Message Lost 0: No Message Lost in TXQ 1: TXQ Message Lost
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag If one frame transmits from TXQ, an interrupt will set.
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag If TXQ receives one frame, an interrupt will set.
16	TXQFIF	TXQ Full Interrupt Flag If TXQ will be in full status, an interrupt will set.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	TXQMC[5:0]	TX Queue Message Count No. of Messages in the TX Queue
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue TX interrupt condition not satisfied after Frame TX 1: TX Queue TX interrupt condition satisfied after Frame TX
1	TXQFLL	TX Queue Full 0: TX Queue Not Full 1: TX Queue Full

Table 23.79 RSCFDnCFDTXQSTS0m Register Contents (2/2)

Bit Position	Bit Name	Function
0	TXQEMP	TX Queue Empty 0: TX Queue Not Empty 1: TX Queue Empty

The TX Queue status registers show the status of the TX Queue of corresponding CAN Channel.

RSCFDnCFDTXQSTS0m.TXQMOW

TXQ message overwrite.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When RSCFDnCFDTXQCC0m.TXQOWE=1 and Message overwrite occurs in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQMLT

TXQ Message Lost.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When Message lost occurs in the Gateway mode of TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQOFTXIF

TXQ One Frame Transmission Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When transmission is successful in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQOFRXIF

TXQ One Frame Reception Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When receiving data is stored in TX queue in Gateway mode, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

This function can use only the Gateway mode of TX queue.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQFIF

TXQ Full Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only when the Gateway mode (RSCFDnCFDTXQCC0m.TXQGWE=1), this bit is set automatically when TX Queue transits to a buffer full status.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQMC[5:0]

TX Queue Message Count

These bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDTXQSTS0m.TXQTXIF

TX Queue TX Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS0m.TXQFLL

TX Queue Full

This bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when the number of CAN messages stored in the TX Queue is less than the configured TX Queue depth.
This bit is cleared automatically when the related channel enters CH_RESET mode.

RSCFDnCFDTXQSTS0m.TXQEMP

TX Queue Empty

This bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.
This bit is set automatically when the last message is transmitted from the TX Queue.
This bit is set automatically when the related RS-CANFD channel enters CH_RESET mode.
The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

23.3.12.6 RSCFDnCFDTXQSTS1m — Transmit Queue Status Register 1 m

Access: RSCFDnCFDTXQSTS1m register can be read or written in 32-bit units
RSCFDnCFDTXQSTS1mL, RSCFDnCFDTXQSTS1mH register can be read or written in 16-bit units
RSCFDnCFDTXQSTS1mLL, RSCFDnCFDTXQSTS1mLH, RSCFDnCFDTXQSTS1mHL,
RSCFDnCFDTXQSTS1mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQSTS1m: <RSCFDn_base> + 1080_H + (04_H × m)

RSCFDnCFDTXQSTS1mL: <RSCFDn_base> + 1080_H + (04_H × m),
RSCFDnCFDTXQSTS1mH: <RSCFDn_base> + 1082_H + (04_H × m)

RSCFDnCFDTXQSTS1mLL: <RSCFDn_base> + 1080_H + (04_H × m),
RSCFDnCFDTXQSTS1mLH: <RSCFDn_base> + 1081_H + (04_H × m),
RSCFDnCFDTXQSTS1mHL: <RSCFDn_base> + 1082_H + (04_H × m),
RSCFDnCFDTXQSTS1mHH: <RSCFDn_base> + 1083_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 23.80 RSCFDnCFDTXQSTS1m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	TXQMOW	TXQ message overwrite 0: No Message overwrite in TXQ 1: Message overwrite in TXQ
19	TXQMLT	TXQ Message Lost 0: No Message Lost in TXQ 1: TXQ Message Lost
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag If one frame transmits from TXQ, an interrupt will set.
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag If TXQ receives one frame, an interrupt will set.
16	TXQFIF	TXQ Full Interrupt Flag If TXQ will be in full status, an interrupt will set.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	TXQMC[5:0]	TX Queue Message Count No. of Messages in the TX Queue
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue TX interrupt condition not satisfied after Frame TX 1: TX Queue TX interrupt condition satisfied after Frame TX
1	TXQFLL	TX Queue Full 0: TX Queue Not Full 1: TX Queue Full

Table 23.80 RSCFDnCFDTXQSTS1m Register Contents (2/2)

Bit Position	Bit Name	Function
0	TXQEMP	TX Queue Empty 0: TX Queue Not Empty 1: TX Queue Empty

The TX Queue status registers show the status of the TX Queue of corresponding CAN Channel.

RSCFDnCFDTXQSTS1m.TXQMOW

TXQ message overwrite.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When RSCFDnCFDTXQCC1m.TXQOWE=1 and Message overwrite occurs in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQMLT

TXQ Message Lost.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When Message lost occurs in the Gateway mode of TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQOFTXIF

TXQ One Frame Transmission Interrupt Flag.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When transmission is successful in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQOFRXIF

TXQ One Frame Reception Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When receiving data is stored in TX queue in Gateway mode, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

This function can use only the Gateway mode of TX queue.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQFIF

TXQ Full Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only when the Gateway mode (RSCFDnCFDTXQCC0m.TXQGWE=1), this bit is set automatically when TX Queue transits to a buffer full status.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQMC[5:0]

TX Queue Message Count.

These bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDTXQSTS1m.TXQTXIF

TX Queue TX Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS1m.TXQFLL

TX Queue Full.

This bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when the number of CAN messages stored in the TX Queue is less than the configured TX Queue depth.
This bit is cleared automatically when the related channel enters CH_RESET mode.

RSCFDnCFDTXQSTS1m.TXQEMP

TX Queue Empty.

This bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.
This bit is set automatically when the last message is transmitted from the TX Queue.
This bit is set automatically when the related RS-CANFD channel enters CH_RESET mode.
The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

23.3.12.7 RSCFDnCFDTXQSTS2m — Transmit Queue Status Register 2 m

Access: RSCFDnCFDTXQSTS2m register can be read or written in 32-bit units
RSCFDnCFDTXQSTS2mL, RSCFDnCFDTXQSTS2mH register can be read or written in 16-bit units
RSCFDnCFDTXQSTS2mLL, RSCFDnCFDTXQSTS2mLH, RSCFDnCFDTXQSTS2mHL, RSCFDnCFDTXQSTS2mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQSTS2m: <RSCFDn_base> + $10E0_H$ + (04_H × m)

RSCFDnCFDTXQSTS2mL: <RSCFDn_base> + $10E0_H$ + (04_H × m),
RSCFDnCFDTXQSTS2mH: <RSCFDn_base> + $10E2_H$ + (04_H × m)

RSCFDnCFDTXQSTS2mLL: <RSCFDn_base> + $10E0_H$ + (04_H × m),
RSCFDnCFDTXQSTS2mLH: <RSCFDn_base> + $10E1_H$ + (04_H × m),
RSCFDnCFDTXQSTS2mHL: <RSCFDn_base> + $10E2_H$ + (04_H × m),
RSCFDnCFDTXQSTS2mHH: <RSCFDn_base> + $10E3_H$ + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.81 RSCFDnCFDTXQSTS2m Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	TXQMOW	TXQ message overwrite 0: No Message overwrite in TXQ 1: Message overwrite in TXQ
19	TXQMLT	TXQ Message Lost 0: No Message Lost in TXQ 1: TXQ Message Lost
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag If one frame transmits from TXQ, an interrupt will set.
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag If TXQ receives one frame, an interrupt will set.
16	TXQFIF	TXQ Full Interrupt Flag If TXQ will be in full status, an interrupt will set.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	TXQMC	TX Queue Message Count No. of Messages in the TX Queue
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue TX interrupt condition not satisfied after Frame TX 1: TX Queue TX interrupt condition satisfied after Frame TX
1	TXQFLL	TX Queue Full 0: TX Queue Not Full 1: TX Queue Full

Table 23.81 RSCFDnCFDTXQSTS2m Register Contents (2/2)

Bit Position	Bit Name	Function
0	TXQEMP	TX Queue Empty 0: TX Queue Not Empty 1: TX Queue Empty

The TX Queue status registers show the status of the TX Queue of corresponding CAN Channel.

RSCFDnCFDTXQSTS2m.TXQMOW

TXQ message overwrite.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When RSCFDnCFDTXQCC2m.TXQOWE=1 and Message overwrite occurs in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQMLT

TXQ Message Lost.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When Message lost occurs in the Gateway mode of TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQOFTXIF

TXQ One Frame Transmission Interrupt Flag.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When transmission is successful in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQOFRXIF

TXQ One Frame Reception Interrupt Flag.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When receiving data is stored in TX queue in Gateway mode, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

This function can use only the Gateway mode of TX queue.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQFIF

TXQ Full Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only when the Gateway mode (RSCFDnCFDTXQCC0m.TXQGWE=1), this bit is set automatically when TX Queue transits to a buffer full status.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQMC[5:0]

TX Queue Message Count.

These bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDTXQSTS2m.TXQTXIF

TX Queue TX Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS2m.TXQFLL

TX Queue Full.

This bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when the number of CAN messages stored in the TX Queue is less than the configured TX Queue depth.

This bit is cleared automatically when the related channel enters CH_RESET mode.

RSCFDnCFDTXQSTS2m.TXQEMP

TX Queue Empty.

This bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when the last message is transmitted from the TX Queue.

This bit is set automatically when the related RS-CANFD channel enters CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

23.3.12.8 RSCFDnCFDTXQSTS3m — Transmit Queue Status Register 3 m

Access: RSCFDnCFDTXQSTS3m register can be read or written in 32-bit units
RSCFDnCFDTXQSTS3mL, RSCFDnCFDTXQSTS3mH register can be read or written in 16-bit units
RSCFDnCFDTXQSTS3mLL, RSCFDnCFDTXQSTS3mLH, RSCFDnCFDTXQSTS3mHL,
RSCFDnCFDTXQSTS3mHH register can be read or written in 8-bit units

Address: RSCFDnCFDTXQSTS3m: <RSCFDn_base> + 1140_H + (04_H × m)

RSCFDnCFDTXQSTS3mL: <RSCFDn_base> + 1140_H + (04_H × m),
RSCFDnCFDTXQSTS3mH: <RSCFDn_base> + 1142_H + (04_H × m)

RSCFDnCFDTXQSTS3mLL: <RSCFDn_base> + 1140_H + (04_H × m),
RSCFDnCFDTXQSTS3mLH: <RSCFDn_base> +1141_H + (04_H × m),
RSCFDnCFDTXQSTS3mHL: <RSCFDn_base> + 1142_H + (04_H × m),
RSCFDnCFDTXQSTS3mHH: <RSCFDn_base> + 1143_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	—	TXQOFTXIF	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.82 RSCFDnCFDTXQSTS3m Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	TXQMOW	TXQ message overwrite 0: No Message overwrite in TXQ 1: Message overwrite in TXQ
19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag If one frame transmits from TXQ, an interrupt will set.
17 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	TXQMC[5:0]	TX Queue Message Count No. of Messages in the TX Queue
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue TX Interrupt condition not satisfied after Frame TX 1: TX Queue TX Interrupt condition satisfied after Frame TX
1	TXQFLL	TX Queue Full 0: TX Queue Not Full 1: TX Queue Full
0	TXQEMP	TX Queue Empty 0: TX Queue Not Empty 1: TX Queue Empty

The TX Queue status registers show the status of the TX Queue of corresponding CAN Channel.

RSCFDnCFDTXQSTS3m.TXQMOW

TXQ message overwrite.

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When RSCFDnCFDTXQCC3m.TXQOWE=1 and Message overwrite occurs in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS3m.TXQOFTXIF

TXQ One Frame Transmission Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When transmission is successful in TX queue, this bit is set automatically.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_Bhas no influence on the bit values.

RSCFDnCFDTXQSTS3m.TXQMC[5:0]

TX Queue Message Count.

These bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDTXQSTS3m.TXQTXIF

TX Queue TX Interrupt Flag

This bit will not be cleared automatically if the TX Queue is disabled.

Users cannot write to this bit when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

The bit is cleared by writing 0_B to it.

The bit is cleared if the related RS-CANFD channel enters CH_RESET mode.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1_B.

Writing 1_B has no influence on the bit values.

RSCFDnCFDTXQSTS3m.TXQFLL

TX Queue Full.

This bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when the number of CAN messages stored in the TX Queue is less than the configured TX Queue depth.

This bit is cleared automatically when the related channel enters CH_RESET mode.

RSCFDnCFDTXQSTS3m.TXQEMP

TX Queue Empty.

This bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when the last message is transmitted from the TX Queue.

This bit is set automatically when the related RS-CANFD channel enters CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

23.3.12.9 RSCFDnCFDTXQPCTR0m — Transmit Queue Pointer Control Register 0 m

Access: RSCFDnCFDTXQPCTR0m register is a write-only register that can be written in 32-bit units
RSCFDnCFDTXQPCTR0mL, RSCFDnCFDTXQPCTR0mH register is a write-only register that can be written in 16-bit units
RSCFDnCFDTXQPCTR0mLL, RSCFDnCFDTXQPCTR0mLH, RSCFDnCFDTXQPCTR0mHL, RSCFDnCFDTXQPCTR0mHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDTXQPCTR0m: <RSCFDn_base> + 1040_H + (04_H × m)

RSCFDnCFDTXQPCTR0mL: <RSCFDn_base> + 1040_H + (04_H × m),
RSCFDnCFDTXQPCTR0mH: <RSCFDn_base> + 1042_H + (04_H × m)

RSCFDnCFDTXQPCTR0mLL: <RSCFDn_base> + 1040_H + (04_H × m),
RSCFDnCFDTXQPCTR0mLH: <RSCFDn_base> + 1041_H + (04_H × m),
RSCFDnCFDTXQPCTR0mHL: <RSCFDn_base> + 1042_H + (04_H × m),
RSCFDnCFDTXQPCTR0mHH: <RSCFDn_base> + 1043_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.83 RSCFDnCFDTXQPCTR0m Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

These registers can be used to confirm storage of a full message in the corresponding TX Queue Buffers.

RSCFDnCFDTXQPCTR0m.TXQPC [7:0]

TX Queue Pointer Control.

When the value FF_H is written to these bits, then the Write Pointer of the corresponding TX Queue Buffer is updated and a transmit request is initiated for this message.

Read value from these bits is always 0_H.

Users cannot write to these bits when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

Users should only write FF_H to this register when the corresponding TX Queue is enabled and not full.

Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not configured in GW mode.

Users should not write to the FIFO control registers when DMA is enabled.

23.3.12.10 RSCFDnCFDTXQPCTR1m — Transmit Queue Pointer Control Register 1 m

Access: RSCFDnCFDTXQPCTR1m register is a write-only register that can be written in 32-bit units
RSCFDnCFDTXQPCTR1mL, RSCFDnCFDTXQPCTR1mH register is a write-only register that can be written in 16-bit units
RSCFDnCFDTXQPCTR1mLL, RSCFDnCFDTXQPCTR1mLH, RSCFDnCFDTXQPCTR1mHL, RSCFDnCFDTXQPCTR1mHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDTXQPCTR1m: <RSCFDn_base> + $10A0_H$ + (04_H × m)

RSCFDnCFDTXQPCTR1mL: <RSCFDn_base> + $10A0_H$ + (04_H × m),
RSCFDnCFDTXQPCTR1mH: <RSCFDn_base> + $10A2_H$ + (04_H × m)

RSCFDnCFDTXQPCTR1mLL: <RSCFDn_base> + $10A0_H$ + (04_H × m),
RSCFDnCFDTXQPCTR1mLH: <RSCFDn_base> + $10A1_H$ + (04_H × m),
RSCFDnCFDTXQPCTR1mHL: <RSCFDn_base> + $10A2_H$ + (04_H × m),
RSCFDnCFDTXQPCTR1mHH: <RSCFDn_base> + $10A3_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.84 RSCFDnCFDTXQPCTR1m Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

These registers can be used to confirm storage of a full message in the corresponding TX Queue Buffers.

RSCFDnCFDTXQPCTR1m.TXQPC [7:0]

TX Queue Pointer Control.

When the value FF_H is written to these bits, then the Write Pointer of the corresponding TX Queue Buffer is updated and a transmit request is initiated for this message.

Read value from these bits is always 0_H.

Users cannot write to these bits when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

Users should only write FF_H to this register when the corresponding TX Queue is enabled and not full.

Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not configured in GW mode.

23.3.12.11 RSCFDnCFDTXQPCTR2m — Transmit Queue Pointer Control Register 2 m

Access: RSCFDnCFDTXQPCTR2m register is a write-only register that can be written in 32-bit units
RSCFDnCFDTXQPCTR2mL, RSCFDnCFDTXQPCTR2mH register is a write-only register that can be written in 16-bit units
RSCFDnCFDTXQPCTR2mLL, RSCFDnCFDTXQPCTR2mLH, RSCFDnCFDTXQPCTR2mHL, RSCFDnCFDTXQPCTR2mHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDTXQPCTR2m: <RSCFDn_base> + 1100_H + (04_H × m)

RSCFDnCFDTXQPCTR2mL: <RSCFDn_base> + 1100_H + (04_H × m),
RSCFDnCFDTXQPCTR2mH: <RSCFDn_base> + 1102_H + (04_H × m)

RSCFDnCFDTXQPCTR2mLL: <RSCFDn_base> + 1100_H + (04_H × m),
RSCFDnCFDTXQPCTR2mLH: <RSCFDn_base> + 1101_H + (04_H × m),
RSCFDnCFDTXQPCTR2mHL: <RSCFDn_base> + 1102_H + (04_H × m),
RSCFDnCFDTXQPCTR2mHH: <RSCFDn_base> + 1103_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.85 RSCFDnCFDTXQPCTR2m Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

These registers can be used to confirm storage of a full message in the corresponding TX Queue Buffers.

RSCFDnCFDTXQPCTR2mTXQPC [7:0]

TX Queue Pointer Control.

When the value FF_H is written to these bits, then the Write Pointer of the corresponding TX Queue Buffer is updated and a transmit request is initiated for this message.

Read value from these bits is always 0_H.

Users cannot write to these bits when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

Users should only write FF_H to this register when the corresponding TX Queue is enabled and not full.

Users should only write FF_H to this register when the Transmit/Receive FIFO is enabled and is not configured in GW mode.

23.3.12.12 RSCFDnCFDTXQPCTR3m — Transmit Queue Pointer Control Register 3 m

Access: RSCFDnCFDTXQPCTR3m register is a write-only register that can be written in 32-bit units
RSCFDnCFDTXQPCTR3mL, RSCFDnCFDTXQPCTR3mH register is a write-only register that can be written in 16-bit units
RSCFDnCFDTXQPCTR3mLL, RSCFDnCFDTXQPCTR3mLH, RSCFDnCFDTXQPCTR3mHL, RSCFDnCFDTXQPCTR3mHH register is a write-only register that can be written in 8-bit units

Address: RSCFDnCFDTXQPCTR3m: <RSCFDn_base> + 1160_H + (04_H × m)

RSCFDnCFDTXQPCTR3mL: <RSCFDn_base> + 1160_H + (04_H × m),
RSCFDnCFDTXQPCTR3mH: <RSCFDn_base> + 1162_H + (04_H × m)

RSCFDnCFDTXQPCTR3mLL: <RSCFDn_base> + 1160_H + (04_H × m),
RSCFDnCFDTXQPCTR3mLH: <RSCFDn_base> + 1161_H + (04_H × m),
RSCFDnCFDTXQPCTR3mHL: <RSCFDn_base> + 1162_H + (04_H × m),
RSCFDnCFDTXQPCTR3mHH: <RSCFDn_base> + 1163_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.86 RSCFDnCFDTXQPCTR3m Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel

These registers can be used to confirm storage of a full message in the corresponding TX Queue Buffers.

RSCFDnCFDTXQPCTR3m.TXQPC [7:0]

TX Queue Pointer Control.

When the value FF_H is written to these bits, then the Write Pointer of the corresponding TX Queue Buffer is updated and a transmit request is initiated for this message.

Read value from these bits is always 0_H.

Users cannot write to these bits when the related RS-CANFD channel is in CH_STOP or CH_RESET mode.

Users should only write FF_H to this register when the corresponding TX Queue is enabled and not full.

Users should not write to the FIFO control registers when DMA is enabled.

23.3.12.13 RSCFDnCFDTXQESTS — Transmit Queue Empty Status Register

Access: RSCFDnCFDTXQESTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQESTSL, RSCFDnCFDTXQESTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQESTSLL, RSCFDnCFDTXQESTSLH, RSCFDnCFDTXQESTSHL, RSCFDnCFDTXQESTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQESTS: <RSCFDn_base> + 1180_H

RSCFDnCFDTXQESTSL: <RSCFDn_base> + 1180_H,
RSCFDnCFDTXQESTSH: <RSCFDn_base> + 1182_H

RSCFDnCFDTXQESTSLL: <RSCFDn_base> + 1180_H,
RSCFDnCFDTXQESTSLH: <RSCFDn_base> + 1181_H,
RSCFDnCFDTXQESTSHL: <RSCFDn_base> + 1182_H,
RSCFDnCFDTXQESTSHH: <RSCFDn_base> + 1183_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXQ31EMP	TXQ30EMP	TXQ29EMP	TXQ28EMP	TXQ27EMP	TXQ26EMP	TXQ25EMP	TXQ24EMP	TXQ23EMP	TXQ22EMP	TXQ21EMP	TXQ20EMP	TXQ19EMP	TXQ18EMP	TXQ17EMP	TXQ16EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ15EMP	TXQ14EMP	TXQ13EMP	TXQ12EMP	TXQ11EMP	TXQ10EMP	TXQ9EMP	TXQ8EMP	TXQ7EMP	TXQ6EMP	TXQ5EMP	TXQ4EMP	TXQ3EMP	TXQ2EMP	TXQ1EMP	TXQ0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.87 RSCFDnCFDTXQESTS Register Contents

Bit Position	Bit Name	Function
31 to 0	TXQ31EMP to TXQ0EMP	TXQ empty Status 0: TXQ not empty 1: TXQ empty

The TXQ status register bits show the status of the Empty bits of the TXQ Buffers.

RSCFDnCFDTXQESTS.TXQ31EMP to TXQ0EMP

TXQ empty Status

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register.

This bit is set when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	channel 0 TX Queue0
1	channel 0 TX Queue1
2	channel 0 TX Queue2
3	channel 0 TX Queue3
4	channel 1 TX Queue0
5	channel 1 TX Queue1
:	:
26	channel 6 TX Queue2
27	channel 6 TX Queue3
28	channel 7 TX Queue0
29	channel 7 TX Queue1
30	channel 7 TX Queue2
31	channel 7 TX Queue3

23.3.12.14 RSCFDnCFDTXQFISTS — Transmit Queue Full Interrupt Status Register

Access: RSCFDnCFDTXQFISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQFISTSL, RSCFDnCFDTXQFISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQFISTSLL, RSCFDnCFDTXQFISTSLH, RSCFDnCFDTXQFISTSHL, RSCFDnCFDTXQFISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQFISTS: <RSCFDn_base> + 1184_H

RSCFDnCFDTXQFISTSL: <RSCFDn_base> + 1184_H,
RSCFDnCFDTXQFISTSH: <RSCFDn_base> + 1186_H

RSCFDnCFDTXQFISTSLL: <RSCFDn_base> + 1184_H,
RSCFDnCFDTXQFISTSLH: <RSCFDn_base> + 1185_H,
RSCFDnCFDTXQFISTSHL: <RSCFDn_base> + 1186_H,
RSCFDnCFDTXQFISTSHH: <RSCFDn_base> + 1187_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TXQ30FULL	TXQ29FULL	TXQ28FULL	—	TXQ26FULL	TXQ25FULL	TXQ24FULL	—	TXQ22FULL	TXQ21FULL	TXQ20FULL	—	TXQ18FULL	TXQ17FULL	TXQ16FULL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TXQ14FULL	TXQ13FULL	TXQ12FULL	—	TXQ10FULL	TXQ9FULL	TXQ8FULL	—	TXQ6FULL	TXQ5FULL	TXQ4FULL	—	TXQ2FULL	TXQ1FULL	TXQ0FULL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.88 RSCFDnCFDTXQFISTS Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 28	TXQ30FULL to TXQ28FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
27	Reserved	When read, the value after reset is returned.
26 to 24	TXQ26FULL to TXQ24FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
23	Reserved	When read, the value after reset is returned.
22 to 20	TXQ22FULL to TXQ20FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
19	Reserved	When read, the value after reset is returned.
18 to 16	TXQ18FULL to TXQ16FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
15	Reserved	When read, the value after reset is returned.
14 to 12	TXQ14FULL to TXQ12FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
11	Reserved	When read, the value after reset is returned.
10 to 8	TXQ10FULL to TXQ8FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
7	Reserved	When read, the value after reset is returned.

Table 23.88 RSCFDnCFDTXQFISTS Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	TXQ6FULL to TXQ4FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set
3	Reserved	When read, the value after reset is returned.
2 to 0	TXQ2FULL to TXQ0FULL	TXQ Full Interrupt Status 0: TXQ Full Interrupt is not set 1: TXQ Full Interrupt is set

The TXQ status register bits show the status of the Full Interrupt bits of the TXQ Buffers.

RSCFDnCFDTXQFISTS.TXQ30FULL to TXQ28FULL, TXQ26FULL to TXQ24FULL, TXQ22FULL to TXQ20FULL, TXQ18FULL to TXQ16FULL, TXQ14FULL to TXQ12FULL, TXQ10FULL to TXQ8FULL, TXQ6FULL to TXQ4FULL, TXQ2FULL to TXQ0FULL

TXQ Full Status.

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	channel 0 TX Queue0
1	channel 0 TX Queue1
2	channel 0 TX Queue2
3	reserve
4	channel 1 TX Queue0
5	channel 1 TX Queue1
:	:
26	channel 6 TX Queue2
27	reserve
28	channel 7 TX Queue0
29	channel 7 TX Queue1
30	channel 7 TX Queue2
31	reserve

23.3.12.15 RSCFDnCFDTXQMSTS — Transmit Queue Message lost Status Register

Access: RSCFDnCFDTXQMSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQMSTSL, RSCFDnCFDTXQMSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQMSTSLL, RSCFDnCFDTXQMSTSLH, RSCFDnCFDTXQMSTSHL, RSCFDnCFDTXQMSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQMSTS: <RSCFDn_base> + 1188_H

RSCFDnCFDTXQMSTSL: <RSCFDn_base> + 1188_H,
RSCFDnCFDTXQMSTSH: <RSCFDn_base> + $118A_H$

RSCFDnCFDTXQMSTSLL: <RSCFDn_base> + 1188_H,
RSCFDnCFDTXQMSTSLH: <RSCFDn_base> + 1189_H,
RSCFDnCFDTXQMSTSHL: <RSCFDn_base> + $118A_H$,
RSCFDnCFDTXQMSTSHH: <RSCFDn_base> + $118B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TXQ30ML	TXQ29ML	TXQ28ML	—	TXQ26ML	TXQ25ML	TXQ24ML	—	TXQ22ML	TXQ21ML	TXQ20ML	—	TXQ18ML	TXQ17ML	TXQ16ML
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TXQ14ML	TXQ13ML	TXQ12ML	—	TXQ10ML	TXQ9ML	TXQ8ML	—	TXQ6ML	TXQ5ML	TXQ4ML	—	TXQ2ML	TXQ1ML	TXQ0ML
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.89 RSCFDnCFDTXQMSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 28	TXQ30ML to TXQ28ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
27	Reserved	When read, the value after reset is returned.
26 to 24	TXQ26ML to TXQ24ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
23	Reserved	When read, the value after reset is returned.
22 to 20	TXQ22ML to TXQ20ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
19	Reserved	When read, the value after reset is returned.
18 to 16	TXQ18ML to TXQ16ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
15	Reserved	When read, the value after reset is returned.
14 to 12	TXQ14ML to TXQ12ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
11	Reserved	When read, the value after reset is returned.
10 to 8	TXQ10ML to TXQ8ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
7	Reserved	When read, the value after reset is returned.

Table 23.89 RSCFDnCFDTXQMSTS Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	TXQ6ML to TXQ4ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set
3	Reserved	When read, the value after reset is returned.
2 to 0	TXQ2ML to TXQ0ML	TXQ message lost Status 0: TXQ message lost flag is not set 1: TXQ message lost flag is set

The TXQ status register bits show the status of the Message lost bits of the TXQ Buffers.

RSCFDnCFDTXQMSTS.TXQ30ML to TXQ28ML, TXQ26ML to TXQ24ML, TXQ22ML to TXQ20ML, TXQ18ML to TXQ16ML, TXQ14ML to TXQ12ML, TXQ10ML to TXQ8ML, TXQ6ML to TXQ4ML, TXQ2ML to TXQ0ML

TXQ message lost Status

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	channel 0 TX Queue0
1	channel 0 TX Queue1
2	channel 0 TX Queue2
3	reserve
4	channel 1 TX Queue0
5	channel 1 TX Queue1
:	:
26	channel 6 TX Queue2
27	reserve
28	channel 7 TX Queue0
29	channel 7 TX Queue1
30	channel 7 TX Queue2
31	reserve

23.3.12.16 RSCFDnCFDTXQOWSTS — Transmit Queue Message Overwrite Status Register

Access: RSCFDnCFDTXQOWSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQOWSTSL, RSCFDnCFDTXQOWSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQOWSTSLL, RSCFDnCFDTXQOWSTSLH, RSCFDnCFDTXQOWSTSHL, RSCFDnCFDTXQOWSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQOWSTS: <RSCFDn_base> + $118C_H$

RSCFDnCFDTXQOWSTSL: <RSCFDn_base> + $118C_H$,
RSCFDnCFDTXQOWSTSH: <RSCFDn_base> + $118E_H$

RSCFDnCFDTXQOWSTSLL: <RSCFDn_base> + $118C_H$,
RSCFDnCFDTXQOWSTSLH: <RSCFDn_base> + $118D_H$,
RSCFDnCFDTXQOWSTSHL: <RSCFDn_base> + $118E_H$,
RSCFDnCFDTXQOWSTSHH: <RSCFDn_base> + $118F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXQ31OW	TXQ30OW	TXQ29OW	TXQ28OW	TXQ27OW	TXQ26OW	TXQ25OW	TXQ24OW	TXQ23OW	TXQ22OW	TXQ21OW	TXQ20OW	TXQ19OW	TXQ18OW	TXQ17OW	TXQ16OW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ15OW	TXQ14OW	TXQ13OW	TXQ12OW	TXQ11O1W	TXQ10OW	TXQ9OW	TXQ8OW	TXQ7OW	TXQ6OW	TXQ5OW	TXQ4OW	TXQ3OW	TXQ2OW	TXQ1OW	TXQ0OW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.90 RSCFDnCFDTXQOWSTS Register Contents

Bit Position	Bit Name	Function
31 to 0	TXQ31OW to TXQ0OW	TXQ message overwrite Status 0: TXQ message overwrite flag is not set 1: TXQ message overwrite flag is set

The TXQ status register bits show the status of the Message overwrite bits of the TXQ Buffers.

RSCFDnCFDTXQOWSTS.TXQ31OW to TXQ0OW

TXQ message overwrite Status

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register.

This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	Channel 0 TX Queue0
1	Channel 0 TX Queue1
2	Channel 0 TX Queue2
3	Channel 0 TX Queue3
4	Channel 1 TX Queue0
5	Channel 1 TX Queue1
:	:
26	Channel 6 TX Queue2
27	Channel 6 TX Queue3
28	Channel 7 TX Queue0
29	Channel 7 TX Queue1
30	Channel 7 TX Queue2
31	Channel 7 TX Queue3

23.3.12.17 RSCFDnCFDTXQISTS — Transmit Queue Interrupt Status Register

Access: RSCFDnCFDTXQISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQISTSL, RSCFDnCFDTXQISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQISTSLL, RSCFDnCFDTXQISTSLH, RSCFDnCFDTXQISTSHL, RSCFDnCFDTXQISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQISTS: <RSCFDn_base> + 1190_H

RSCFDnCFDTXQISTSL: <RSCFDn_base> + 1190_H,
RSCFDnCFDTXQISTSH: <RSCFDn_base> + 1192_H

RSCFDnCFDTXQISTSLL: <RSCFDn_base> + 1190_H,
RSCFDnCFDTXQISTSLH: <RSCFDn_base> + 1191_H,
RSCFDnCFDTXQISTSHL: <RSCFDn_base> + 1192_H,
RSCFDnCFDTXQISTSHH: <RSCFDn_base> + 1193_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXQ31ISF	TXQ30ISF	TXQ29ISF	TXQ28ISF	TXQ27ISF	TXQ26ISF	TXQ25ISF	TXQ24ISF	TXQ23ISF	TXQ22ISF	TXQ21ISF	TXQ20ISF	TXQ19ISF	TXQ18ISF	TXQ17ISF	TXQ16ISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ15ISF	TXQ14ISF	TXQ13ISF	TXQ12ISF	TXQ11ISF	TXQ10ISF	TXQ9ISF	TXQ8ISF	TXQ7ISF	TXQ6ISF	TXQ5ISF	TXQ4ISF	TXQ3ISF	TXQ2ISF	TXQ1ISF	TXQ0ISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.91 RSCFDnCFDTXQISTS Register Contents

Bit Position	Bit Name	Function
31 to 0	TXQ31ISF to TXQ0ISF	TXQ Interrupt Status Flag 0: TXQ Interrupt flag is not set. 1: TXQ Interrupt flag is set.

The TXQ status register bits show the status of the Interrupt flag bits of the TXQ Buffers.

RSCFDnCFDTXQISTS.TXQ31ISF to TXQ0ISF

TXQ Interrupt Status Flag.

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register. This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	Channel 0 TX Queue0
1	Channel 0 TX Queue1
2	Channel 0 TX Queue2
3	Channel 0 TX Queue3
4	Channel 1 TX Queue0
5	Channel 1 TX Queue1
:	:
26	Channel 6 TX Queue2
27	Channel 6 TX Queue3
28	Channel 7 TX Queue0
29	Channel 7 TX Queue1
30	Channel 7 TX Queue2
31	Channel 7 TX Queue3

23.3.12.18 RSCFDnCFDTXQOFTISTS — Transmit Queue One Frame Transmit Interrupt Status Register

Access: RSCFDnCFDTXQOFTISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQOFTISTSL, RSCFDnCFDTXQOFTISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQOFTISTSLL, RSCFDnCFDTXQOFTISTSLH, RSCFDnCFDTXQOFTISTSHL, RSCFDnCFDTXQOFTISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQOFTISTS: <RSCFDn_base> + 1194_H

RSCFDnCFDTXQOFTISTSL: <RSCFDn_base> + 1194_H,
RSCFDnCFDTXQOFTISTSH: <RSCFDn_base> + 1196_H

RSCFDnCFDTXQOFTISTSLL: <RSCFDn_base> + 1194_H,
RSCFDnCFDTXQOFTISTSLH: <RSCFDn_base> + 1195_H,
RSCFDnCFDTXQOFTISTSHL: <RSCFDn_base> + 1196_H,
RSCFDnCFDTXQOFTISTSHH: <RSCFDn_base> + 1197_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXQ31 OFTISF	TXQ30 OFTISF	TXQ29 OFTISF	TXQ28 OFTISF	TXQ27 OFTISF	TXQ26 OFTISF	TXQ25 OFTISF	TXQ24 OFTISF	TXQ23 OFTISF	TXQ22 OFTISF	TXQ21 OFTISF	TXQ20 OFTISF	TXQ19 OFTISF	TXQ18 OFTISF	TXQ17 OFTISF	TXQ16 OFTISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ15 OFTISF	TXQ14 OFTISF	TXQ13 OFTISF	TXQ12 OFTISF	TXQ11 OFTISF	TXQ10 OFTISF	TXQ9O FTISF	TXQ8O FTISF	TXQ7O FTISF	TXQ6O FTISF	TXQ5O FTISF	TXQ4O FTISF	TXQ3O FTISF	TXQ2O FTISF	TXQ1O FTISF	TXQ0O FTISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.92 RSCFDnCFDTXQOFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 0	TXQ31OFTISF to TXQ0OFTISF	TXQ One Frame TX Interrupt Status Flag 0: TXQ One Frame TX Interrupt flag is not set. 1: TXQ One Frame TX Interrupt flag is set.

The TXQ status register bits show the status of the One Frame TX Interrupt flag bits of the TXQ Buffers.

RSCFDnCFDTXQOFTISTS.TXQ31OFTISF to TXQ0OFTISF

TXQ One Frame TX Interrupt Status Flag.

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register. This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	Channel 0 TX Queue0
1	Channel 0 TX Queue1
2	Channel 0 TX Queue2
3	Channel 0 TX Queue3
4	Channel 1 TX Queue0
5	Channel 1 TX Queue1
:	:
26	Channel 6 TX Queue2
27	Channel 6 TX Queue3
28	Channel 7 TX Queue0
29	Channel 7 TX Queue1
30	Channel 7 TX Queue2
31	Channel 7 TX Queue3

23.3.12.19 RSCFDnCFDTXQOFRISTS — Transmit Queue One Frame Receive Interrupt Status Register

Access: RSCFDnCFDTXQOFRISTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQOFRISTSL, RSCFDnCFDTXQOFRISTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQOFRISTSLL, RSCFDnCFDTXQOFRISTSLH, RSCFDnCFDTXQOFRISTSHL, RSCFDnCFDTXQOFRISTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQOFRISTS: <RSCFDn_base> + 1198_H

RSCFDnCFDTXQOFRISTSL: <RSCFDn_base> + 1198_H,
RSCFDnCFDTXQOFRISTSH: <RSCFDn_base> + $119A_H$

RSCFDnCFDTXQOFRISTSLL: <RSCFDn_base> + 1198_H,
RSCFDnCFDTXQOFRISTSLH: <RSCFDn_base> + 1199_H,
RSCFDnCFDTXQOFRISTSHL: <RSCFDn_base> + $119A_H$,
RSCFDnCFDTXQOFRISTSHH: <RSCFDn_base> + $119B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TXQ30 OFRISF	TXQ29 OFRISF	TXQ28 OFRISF	—	TXQ26 OFRISF	TXQ25 OFRISF	TXQ24 OFRISF	—	TXQ22 OFRISF	TXQ21 OFRISF	TXQ20 OFRISF	—	TXQ18 OFRISF	TXQ17 OFRISF	TXQ16 OFRISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TXQ14 OFRISF	TXQ13 OFRISF	TXQ12 OFRISF	—	TXQ10 OFRISF	TXQ9O FRISF	TXQ8O FRISF	—	TXQ6O FRISF	TXQ5O FRISF	TXQ4O FRISF	—	TXQ2O FRISF	TXQ1O FRISF	TXQ0O FRISF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.93 RSCFDnCFDTXQOFRISTS Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 28	TXQ30OFRISF to TXQ28OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
27	Reserved	When read, the value after reset is returned.
26 to 24	TXQ26OFRISF to TXQ24OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
23	Reserved	When read, the value after reset is returned.
22 to 20	TXQ22OFRISF to TXQ20OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
19	Reserved	When read, the value after reset is returned.
18 to 16	TXQ18OFRISF to TXQ16OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
15	Reserved	When read, the value after reset is returned.
14 to 12	TXQ14OFRISF to TXQ12OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
11	Reserved	When read, the value after reset is returned.
10 to 8	TXQ10OFRISF to TXQ8OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
7	Reserved	When read, the value after reset is returned.

Table 23.93 RSCFDnCFDTXQOFRISTS Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	TXQ6OFRISF to TXQ4OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.
3	Reserved	When read, the value after reset is returned.
2 to 0	TXQ2OFRISF to TXQ0OFRISF	TXQ One Frame RX Interrupt Status Flag 0: TXQ One Frame RX Interrupt flag is not set. 1: TXQ One Frame RX Interrupt flag is set.

The TXQ status register bits show the status of the One Frame RX Interrupt flag bits of the TXQ Buffers.

RSCFDnCFDTXQOFRISTS.TXQ30OFRISF to TXQ28OFRISF, TXQ26OFRISF to TXQ24OFRISF, TXQ22OFRISF to TXQ20OFRISF, TXQ18OFRISF to TXQ16OFRISF, TXQ14OFRISF to TXQ12OFRISF, TXQ10OFRISF to TXQ8OFRISF, TXQ6OFRISF to TXQ4OFRISF, TXQ2OFRISF to TXQ0OFRISF

TXQ One Frame RX Interrupt Status Flag.

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register. This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	Channel 0 TX Queue0
1	Channel 0 TX Queue1
2	Channel 0 TX Queue2
3	Reserved
4	Channel 1 TX Queue0
5	Channel 1 TX Queue1
:	:
26	Channel 6 TX Queue2
27	Reserved
28	Channel 7 TX Queue0
29	Channel 7 TX Queue1
30	Channel 7 TX Queue2
31	Reserved

23.3.12.20 RSCFDnCFDTXQFSTS — Transmit Queue Full Status Register

Access: RSCFDnCFDTXQFSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDTXQFSTSL, RSCFDnCFDTXQFSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTXQFSTSLL, RSCFDnCFDTXQFSTSLH, RSCFDnCFDTXQFSTSHL, RSCFDnCFDTXQFSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTXQFSTS: <RSCFDn_base> + $119C_H$

RSCFDnCFDTXQFSTSL: <RSCFDn_base> + $119C_H$,
RSCFDnCFDTXQFSTSH: <RSCFDn_base> + $119E_H$

RSCFDnCFDTXQFSTSLL: <RSCFDn_base> + $119C_H$,
RSCFDnCFDTXQFSTSLH: <RSCFDn_base> + $119D_H$,
RSCFDnCFDTXQFSTSHL: <RSCFDn_base> + $119E_H$,
RSCFDnCFDTXQFSTSHH: <RSCFDn_base> + $119F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXQ31FSF	TXQ30FSF	TXQ29FS	TXQ28FS	TXQ27FSF	TXQ26FSF	TXQ25FSF	TXQ24FSF	TXQ23FSF	TXQ22FSF	TXQ21FSF	TXQ20FSF	TXQ19FSF	TXQ18FSF	TXQ17FSF	TXQ16FSF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXQ15FSF	TXQ14FSF	TXQ13FSF	TXQ12FSF	TXQ11FSF	TXQ10FSF	TXQ9FSF	TXQ8FSF	TXQ7FSF	TXQ6FSF	TXQ5FSF	TXQ4FSF	TXQ3FSF	TXQ2FSF	TXQ1FSF	TXQ0FSF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.94 RSCFDnCFDTXQFSTS Register Contents

Bit Position	Bit Name	Function
31 to 0	TXQ31FSF to TXQ0FSF	TXQ Full Status Flag 0: TXQ Full flag is not set. 1: TXQ Full flag is set.

The TXQ status register bits show the status of the Full Status flag bits of the TXQ Buffers.

RSCFDnCFDTXQFSTS.TXQ31FSF to TXQ0FSF

TXQ Full Status Flag

Each bit is set automatically when the corresponding bit is set in the TXQ Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TXQ Status Register. This bit is cleared when RS-CANFD module enters GL_RESET mode.

Bit position	Corresponding TXQueue
0	Channel 0 TX Queue0
1	Channel 0 TX Queue1
2	Channel 0 TX Queue2
3	Channel 0 TX Queue3
4	Channel 1 TX Queue0
5	Channel 1 TX Queue1
:	:
26	Channel 6 TX Queue2
27	Channel 6 TX Queue3
28	Channel 7 TX Queue0
29	Channel 7 TX Queue1
30	Channel 7 TX Queue2
31	Channel 7 TX Queue3

23.3.13 Details of Transmit History-Related Registers

23.3.13.1 RSCFDnCFDTHLCCm — Transmit History List Configuration/Control Register m

Access: RSCFDnCFDTHLCCm register can be read or written in 32-bit units
RSCFDnCFDTHLCCmL, RSCFDnCFDTHLCCmH register can be read or written in 16-bit units
RSCFDnCFDTHLCCmLL, RSCFDnCFDTHLCCmLH, RSCFDnCFDTHLCCmHL, RSCFDnCFDTHLCCmHH register can be read or written in 8-bit units

Address: RSCFDnCFDTHLCCm: <RSCFDn_base> + 1200_H + (04_H × m)

RSCFDnCFDTHLCCmL: <RSCFDn_base> + 1200_H + (04_H × m),
RSCFDnCFDTHLCCmH: <RSCFDn_base> + 1202_H + (04_H × m)

RSCFDnCFDTHLCCmLL: <RSCFDn_base> + 1200_H + (04_H × m),
RSCFDnCFDTHLCCmLH: <RSCFDn_base> + 1201_H + (04_H × m),
RSCFDnCFDTHLCCmHL: <RSCFDn_base> + 1202_H + (04_H × m),
RSCFDnCFDTHLCCmHH: <RSCFDn_base> + 1203_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	THLDGE	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.95 RSCFDnCFDTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	THLDGE	TX History List Dedicated GW Enable 0: Not dedicate Gateway FIFO + Gateway TX Queue. 1: Dedicate Gateway FIFO + Gateway TX Queue.
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue. 1: Flat TX MB + TX FIFO + TX Queue.
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches three-quarters of the TX History List depth. 1: Interrupt generated for every successfully stored entry.
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled. 1: TX History List Interrupt enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	THLE	TX History List Enable 0: TX History List disabled. 1: TX History List enabled.

The TX History List Configuration/Control register configures the TX History List functions.

RSCFDnCFDTHLCCm.THLDGE

TX History List Dedicated GW Enable.

This bit selects the conditions for storing an entry in the TX History list after successful transmission.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode.

RSCFDnCFDTHLCCm.THLDTE

TX History List Dedicated TX Enable.

This bit selects the conditions for storing an entry in the TX History list after successful transmission.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode.

RSCFDnCFDTHLCCm.THLIM

TX History List Interrupt Mode.

This bit selects the Interrupt generation condition for the FIFO.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

Users should not write to this bit when RS-CANFD module is in GL_TEST or GL_OPERATING mode.

RSCFDnCFDTHLCCm.THLIE

TX History List Interrupt Enable.

This bit enables the generation of the TX History List Interrupt when it is set.

Users cannot write to this bit when RS-CANFD module is in GL_STOP mode.

RSCFDnCFDTHLCCm.THLE

TX History List Enable.

This bit enables the TX History List Buffer when it is set.

Users cannot write to this bit when the related RS-CANFD channel is in CH_RESET or CH_STOP mode.

This bit is cleared automatically when the related RS-CANFD channel is in CH_RESET mode.

23.3.13.2 RSCFDnCFDTHLSTSm — Transmit History List Status Register m

Access: RSCFDnCFDTHLSTSm register can be read or written in 32-bit units
RSCFDnCFDTHLSTSmL, RSCFDnCFDTHLSTSmH register can be read or written in 16-bit units
RSCFDnCFDTHLSTSmLL, RSCFDnCFDTHLSTSmLH, RSCFDnCFDTHLSTSmHL, RSCFDnCFDTHLSTSmHH register can be read or written in 8-bit units

Address: RSCFDnCFDTHLSTSm: <RSCFDn_base> + 1220_H + (04_H × m)

RSCFDnCFDTHLSTSmL: <RSCFDn_base> + 1220_H + (04_H × m),
RSCFDnCFDTHLSTSmH: <RSCFDn_base> + 1222_H + (04_H × m)

RSCFDnCFDTHLSTSmLL: <RSCFDn_base> + 1220_H + (04_H × m),
RSCFDnCFDTHLSTSmLH: <RSCFDn_base> + 1221_H + (04_H × m),
RSCFDnCFDTHLSTSmHL: <RSCFDn_base> + 1222_H + (04_H × m),
RSCFDnCFDTHLSTSmHH: <RSCFDn_base> + 1223_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	THLMC[5:0]						—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R

Table 23.96 RSCFDnCFDTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	THLMC[5:0]	TX History List Message Count Number of messages stored in TX history list
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	THLIF	TX History List Interrupt Flag 0: TX History List Interrupt condition not satisfied. 1: TX History List Interrupt condition satisfied.
2	THLELT	TX History List Entry Lost 0: No Entry Lost in TX History List. 1: TX History List Entry Lost.
1	THLFLL	TX History List Full 0: TX History List Not Full. 1: TX History List Full.
0	THLEMP	TX History List Empty 0: TX History List Not Empty. 1: TX History List Empty.

The TX History List Status register shows the status of the data stored in the TX History List Buffer.

RSCFDnCFDTHLSTSm.THLMC[5:0]

TX History List Message Count.

These bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related channel is in CH_RESET mode.

RSCFDnCFDTHLSTSm.THLIF

TX History List Interrupt Flag.

Users can write to this bit only when RS-CANFD module is in CH_HALT or CH_COMMUNICATION modes.

Writing 1 has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

This bit is set when the configured Interrupt condition is satisfied.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

RSCFDnCFDTHLSTSm.THLELT

TX History List Entry Lost.

Users can write to this bit only when RS-CANFD module is in CH_HALT or CH_COMMUNICATION modes.

Writing 1 has no influence on the bit values.

Do not use bit clear instruction for clearing this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Then the other bits remain set to 1.

This bit is set when a new Entry cannot be stored as the related TX History List Buffer is already full.

This bit is cleared by writing 0 to it.

This bit is automatically cleared in CH_RESET mode.

RSCFDnCFDTHLSTSm.THLFLL

TX History List Full.

Each TX History List can store up to 32 entries (each channel has a dedicated TX History List).

This bit is set automatically when the number of entries in the TX History List Buffer matches the TX History List depth.

This bit is cleared automatically when the number of entries in the TX History List Buffer is less than the TX History List depth.

This bit is cleared automatically when TX History List is disabled.

This bit is cleared automatically when the corresponding CAN channel enters CH_RESET mode.

RSCFDnCFDTHLSTSm.THLEMP

TX History List Empty.

This bit is set automatically when CPU has read all the entries from the TX History List Buffer.

This bit is set automatically when TX History List is disabled.

This bit is set automatically when the corresponding CAN channel enters CH_RESET mode.

This bit is cleared automatically when the first entry is stored to the TX History List.

23.3.13.3 RSCFDnCFDTHLPCTRm — Transmit History List Pointer Control Register

Access: RSCFDnCFDTHLPCTRm register is a write-only register that can be write in 32-bit units
RSCFDnCFDTHLPCTRmL, RSCFDnCFDTHLPCTRmH register is a write-only register that can be write in 16-bit units
RSCFDnCFDTHLPCTRmLL, RSCFDnCFDTHLPCTRmLH, RSCFDnCFDTHLPCTRmHL, RSCFDnCFDTHLPCTRmHH register is a write-only register that can be write in 8-bit units

Address: RSCFDnCFDTHLPCTRm: <RSCFDn_base> + 1240_H + (04_H × m)

RSCFDnCFDTHLPCTRmL: <RSCFDn_base> + 1240_H + (04_H × m),
RSCFDnCFDTHLPCTRmH: <RSCFDn_base> + 1242_H + (04_H × m)

RSCFDnCFDTHLPCTRmLL: <RSCFDn_base> + 1240_H + (04_H × m),
RSCFDnCFDTHLPCTRmLH: <RSCFDn_base> + 1241_H + (04_H × m),
RSCFDnCFDTHLPCTRmHL: <RSCFDn_base> + 1242_H + (04_H × m),
RSCFDnCFDTHLPCTRmHH: <RSCFDn_base> + 1243_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.97 RSCFDnCFDTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel.

This register can be used to increment the Read Pointer of the TX History List.

RSCFDnCFDTHLPCTRm.THLPC[7:0]

TX History List Pointer Control.

When FF_H is written to these bits, the Read Pointer of the TX History List is moved to the next TX History List entry address.

Read value from these bits is always 00_H.

Users can write to this bit only when RS-CANFD module is in CH_HALT or CH_COMMUNICATION modes.

Users should only write FF_H to these registers when the corresponding TX History List is enabled and is not empty.

23.3.13.4 RSCFDnCFDTHLACC0m — Channel m Transmit History List Access Register 0

Access: RSCFDnCFDTHLACC0m register is a read-only register that can be read in 32-bit units
RSCFDnCFDTHLACC0mL, RSCFDnCFDTHLACC0mH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTHLACC0mLL, RSCFDnCFDTHLACC0mLH, RSCFDnCFDTHLACC0mHL, RSCFDnCFDTHLACC0mHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTHLACC0m: <RSCFDn_base> + 8000_H + (08_H × m)

RSCFDnCFDTHLACC0mL: <RSCFDn_base> + 8000_H + (08_H × m),
RSCFDnCFDTHLACC0mH: <RSCFDn_base> + 8002_H + (08_H × m)

RSCFDnCFDTHLACC0mLL: <RSCFDn_base> + 8000_H + (08_H × m),
RSCFDnCFDTHLACC0mLH: <RSCFDn_base> + 8001_H + (08_H × m),
RSCFDnCFDTHLACC0mHL: <RSCFDn_base> + 8002_H + (08_H × m),
RSCFDnCFDTHLACC0mHH: <RSCFDn_base> + 8003_H + (08_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TGW	—	—	—	—	—	BN[6:0]							BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.98 RSCFDnCFDTHLACC0m Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTS[15:0]	Transmit Timestamp Transmit Timestamp value for SW drivers.
15	TGW	Transmit Gateway Buffer indication 0: Not transmission from Gateway. 1: Transmission from Gateway.
14 to 10	Reserved	When read, the value after reset is returned.
9 to 3	BN[6:0]	Buffer Number Number of the Message Buffer
2 to 0	BT[2:0]	Buffer Type 001: Flat TX Message Buffer 010: TX FIFO MB Number and GW FIFO MB Number. 100: TX Queue MB Number. Other than the above: Setting prohibited

The TX History List Access register provides access to the entry in the TX History List based on the Read Timestamp value.

RSCFDnCFDTHLACC0m.TMTS[15:0]

Transmit Timestamp.

These bits indicate the Timestamp for use by the SW drivers.

RSCFDnCFDTHLACC0m.TGW

Transmit Gateway Buffer indication.

This bit is automatically set to 1, when transmission is completed in GW mode.

RSCFDnCFDTHLACC0m.BN[6:0]

Buffer Number.

These bits show the Message Buffer from which transmission was successfully completed. If a message from a Transmit/Receive FIFO is transmitted, then these bits show the Message Buffer that is linked to the Transmit/Receive FIFO for transmission.

RSCFDnCFDTHLACC0m.BT[2:0]

Buffer Type.

These bits indicate if the data has been stored following a transmission from a FIFO, a TX Queue or a TX MB.

23.3.13.5 RSCFDnCFDTHLACC1m — Channel m Transmit History List Access Register 1

Access: RSCFDnCFDTHLACC1m register is a read-only register that can be read in 32-bit units
RSCFDnCFDTHLACC1mL, RSCFDnCFDTHLACC1mH register is a read-only register that can be read in 16-bit units
RSCFDnCFDTHLACC1mLL, RSCFDnCFDTHLACC1mLH, RSCFDnCFDTHLACC1mHL, RSCFDnCFDTHLACC1mHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTHLACC1m: <RSCFDn_base> + 8004_H + (08_H × m)

RSCFDnCFDTHLACC1mL: <RSCFDn_base> + 8004_H + (08_H × m),
RSCFDnCFDTHLACC1mH: <RSCFDn_base> + 8006_H + (08_H × m)

RSCFDnCFDTHLACC1mLL: <RSCFDn_base> + 8004_H + (08_H × m),
RSCFDnCFDTHLACC1mLH: <RSCFDn_base> + 8005_H + (08_H × m),
RSCFDnCFDTHLACC1mHL: <RSCFDn_base> + 8006_H + (08_H × m),
RSCFDnCFDTHLACC1mHH: <RSCFDn_base> + 8007_H + (08_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIFL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.99 RSCFDnCFDTHLACC1m Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17, 16	TIFL[1:0]	Transmit Information Label MB information label or TX FIFO information label or AFL information label is stored for SW drivers.
15 to 0	TID[15:0]	Transmit ID MB reference ID or TX FIFO references ID or AFL pointer field is stored for SW drivers.

The TX History List Access register1 provides access to the entry in the TX History List based on the Read Pointer value.

RSCFDnCFDTHLACC1m.TIFL[1:0]

Transmit Information Label.

These bits indicate the MB information label (RSCFDnCFDTMFDCTR.TMIFL) or the TX FIFO information label (RSCFDnCFDCFFDCSTS.CFIFL) for use by the SW drivers.

In a case of transmission in GW mode, these bits indicate the AFL pointer field (RSCFDnCFDGAFLMj.GAFLIFL1, RSCFDnCFDGAFLP0j.GAFLIFL0) instead of the MB information label (RSCFDnCFDTMFDCTR.TMIFL).

RSCFDnCFDTHLACC1m.TID[15:0]

Transmit ID.

These bits indicate the MB reference ID (RSCFDnCFDTMFDCTR.TMPTR) or the TX FIFO reference ID (RSCFDnCFDCFFDCSTS.CFPTR) for use by the SW drivers.

In a case of transmission in GW mode, these bits indicate the AFL pointer field (RSCFDnCFDGAFLP0j.GAFLPTR) instead of the MB reference ID (RSCFDnCFDTMFDCTR.TMPTR).

23.3.14 Details of Test-Related Registers

23.3.14.1 RSCFDnCFDGTSTCFG — Global Test Configuration Register

Access: RSCFDnCFDGTSTCFG register can be read or written in 32-bit units
RSCFDnCFDGTSTCFGL, RSCFDnCFDGTSTCFGH register can be read or written in 16-bit units
RSCFDnCFDGTSTCFGLL, RSCFDnCFDGTSTCFGLH, RSCFDnCFDGTSTCFGHL,
RSCFDnCFDGTSTCFGHH register can be read or written in 8-bit units

Address: RSCFDnCFDGTSTCFG: <RSCFDn_base> + 1308_H

RSCFDnCFDGTSTCFGL: <RSCFDn_base> + 1308_H,
RSCFDnCFDGTSTCFGH: <RSCFDn_base> + $130A_H$

RSCFDnCFDGTSTCFGLL: <RSCFDn_base> + 1308_H,
RSCFDnCFDGTSTCFGLH: <RSCFDn_base> + 1309_H,
RSCFDnCFDGTSTCFGHL: <RSCFDn_base> + $130A_H$,
RSCFDnCFDGTSTCFGHH: <RSCFDn_base> + $130B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	RTMPS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CmICBCE[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.100 RSCFDnCFDGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	RTMPS[9:0]	RAM Test Mode Page Select Select a RAM Test Mode page
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CmICBCE[7:0]	Channel m Internal CAN Bus Communication Test Mode Enable 0: Channel m internal CAN bus communication disabled. 1: Channel m internal CAN bus communication enabled.

The Global Test Configuration Register is used to configure the CAN channels joining the internal CAN Bus Communication Test Mode and the RAM Test Mode Page.

RSCFDnCFDGTSTCFG.RTMPS[9:0]

RAM Test Mode Page Select.

These bits select the RAM Page mode for CPU read/write access when RS-CANFD module is configured in RAM Test mode.

Refer to **Section 23.11.2.1** for RAM Test Mode Specification.

Users cannot write to these bits when the RS-CANFD module is in GL_RESET or GL_STOP mode.

In U2A-EVA, U2A16, U2A8 users should only enter values from 0 to 95 ($5F_H$) for the AFL RAM and 96 to 868 (364_H) for the MB RAM.

In U2A6 users should only enter values from 0 to 63 ($3F_H$) for the AFL RAM and 64 to 304 (130_H) for the MB RAM for RSCFD0; from 0 to 31 ($1F_H$) for the AFL RAM and 32 to 152 (98_H) for the MB RAM for RSCFD1.

Users should write to this bit only when RS-CANFD module is in GL_TEST mode.

These bits are cleared automatically when the related RS-CANFD channel is in GL_RESET mode.

RSCFDnCFDGTSTCFG.CmICBCE[7:0]

Channel m Internal CAN Bus Communication Test Mode Enable.

If this bit is set and RS-CANFD module is configured in Internal CAN Bus Communication Test Mode, then CAN channel m will participate to the Internal CAN Bus Communication Test Mode operation.

Users cannot write to this bit when the RS-CANFD module is in GL_RESET or GL_STOP mode.

Users should write to this bit only when RS-CANFD module is in GL_TEST mode.

These bits are cleared automatically when the RS-CANFD module is in GL_RESET mode.

23.3.14.2 RSCFDnCFDGTSTCTR — Global Test Control Register

Access: RSCFDnCFDGTSTCTR register can be read or written in 32-bit units
RSCFDnCFDGTSTCTRL, RSCFDnCFDGTSTCTRH register can be read or written in 16-bit units
RSCFDnCFDGTSTCTRLL, RSCFDnCFDGTSTCTRLH, RSCFDnCFDGTSTCTRHL,
RSCFDnCFDGTSTCTRHH register can be read or written in 8-bit units

Address: RSCFDnCFDGTSTCTR: <RSCFDn_base> + $130C_H$

RSCFDnCFDGTSTCTRL: <RSCFDn_base> + $130C_H$,
RSCFDnCFDGTSTCTRH: <RSCFDn_base> + $130E_H$

RSCFDnCFDGTSTCTRLL: <RSCFDn_base> + $130C_H$,
RSCFDnCFDGTSTCTRLH: <RSCFDn_base> + $130D_H$,
RSCFDnCFDGTSTCTRHL: <RSCFDn_base> + $130E_H$,
RSCFDnCFDGTSTCTRHH: <RSCFDn_base> + $130F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 23.101 RSCFDnCFDGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	RTME	RAM Test Mode Enable 0: RAM Test Mode disabled. 1: RAM Test Mode enabled.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICBCTME	Internal CAN Bus Communication Test Mode Enable 0: Internal CAN Bus communication test mode disabled. 1: Internal CAN Bus communication test mode enabled.

The Global Test Control register is used to control the global test modes of the RS-CANFD module.

RSCFDnCFDGTSTCTR.RTME

RAM Test Mode Enable.

If this bit is set, RS-CANFD module is configured in RAM Test Mode. Refer to **Section 23.11.2.1** for RAM Test Mode Specification.

Users can set this bit only when RS-CANFD module is in GL_TEST mode.

Users should clear this bit when RS-CANFD module is in GL_TEST mode.

This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.

RSCFDnCFDGTSTCTR.ICBCTME

Internal CAN Bus Communication Test Mode Enable.

If this bit is set, internal CAN Bus Communication is enabled for the CAN channels that are configured for Internal CAN Bus Communication participation. Refer to **Section 23.11.2.2, Internal CAN Bus Communication Test Mode**.

Users can set this bit only when RS-CANFD module is in GL_TEST mode.

Users should clear this bit when RS-CANFD module is in GL_TEST mode.

This bit is cleared automatically when RS-CANFD module enters GL_RESET mode.

23.3.14.3 RSCFDnCFDGLOCKK — Global Lock Key Register

Access: RSCFDnCFDGLOCKK is a write-only register that can be written in 32-bit units
RSCFDnCFDGLOCKKL, RSCFDnCFDGLOCKKH is a write-only register that can be written in 16-bit units

Address: RSCFDnCFDGLOCKK: <RSCFDn_base> + $131C_H$

RSCFDnCFDGLOCKKL: <RSCFDn_base> + $131C_H$,
RSCFDnCFDGLOCKKH: <RSCFDn_base> + $131E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 23.102 RSCFDnCFDGLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes.

The Global Lock Key register is a write only register that is used to unlock the protection for special test bits.

Refer to **Section 23.11.2** for Lock Key specification.

RSCFDnCFDGLOCKK.LOCK[15:0]

Lock Key.

The Unlock Key sequence must be written in these bits to configure RS-CANFD module in FIFO OTB disable and RAM Test.

Read value from these bits is always 0000_H.

Users can not write to these bits when RS-CANFD module is in GL_STOP or GL_RESET mode.

Users should not write to these bits when RS-CANFD module is in GL_OPERATING mode.

23.3.14.4 RSCFDnCFDRPGACCr — RAM Test Page Access Register r

Access: RSCFDnCFDRPGACCr register can be read or written in 32-bit units
RSCFDnCFDRPGACCrL, RSCFDnCFDRPGACCrH register can be read or written in 16-bit units
RSCFDnCFDRPGACCrLL, RSCFDnCFDRPGACCrLH, RSCFDnCFDRPGACCrHL,
RSCFDnCFDRPGACCrHH register can be read or written in 8-bit units

Address: RSCFDnCFDRPGACCr: <RSCFDn_base> + 8400_H + (04_H × r)

RSCFDnCFDRPGACCrL: <RSCFDn_base> + 8400_H + (04_H × r),
RSCFDnCFDRPGACCrH: <RSCFDn_base> + 8402_H + (04_H × r)

RSCFDnCFDRPGACCrLL: <RSCFDn_base> + 8400_H + (04_H × r),
RSCFDnCFDRPGACCrLH: <RSCFDn_base> + 8401_H + (04_H × r),
RSCFDnCFDRPGACCrHL: <RSCFDn_base> + 8402_H + (04_H × r),
RSCFDnCFDRPGACCrHH: <RSCFDn_base> + 8403_H + (04_H × r)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.103 RSCFDnCFDRPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA[31:0]	RAM Data Test Access RAM Data Bytes

(Average number of entries for 1 channel = no_of_entries = 64)

(r = [0...no_of_entries-1])

RSCFDnCFDRPGACCr.RDTA[31:0]

RAM Data Test Access.

Data can be read from or written into these register bits when RS-CANFD module is configured in RAM Test Mode.

Users can only write to this bit when RS-CANFD module is in GL_TEST mode, and RAM test mode is enabled.

Software data should be read/written in the RAM Test Page Access registers during RAM Test Mode.

23.3.15 Details of BUS Load Counter Register

23.3.15.1 RSCFDnCFDCmBLCT — Channel m Bus load Control Register

Access: RSCFDnCFDCmBLCT register can be read or written in 32-bit units
RSCFDnCFDCmBLCTL, RSCFDnCFDCmBLCTH register can be read or written in 16-bit units
RSCFDnCFDCmBLCTLL, RSCFDnCFDCmBLCTLH, RSCFDnCFDCmBLCTHL, RSCFDnCFDCmBLCTHH register can be read or written in 8-bit units

Address: RSCFDnCFDCmBLCT: <RSCFDn_base> + 1418_H + (20_H × m)

RSCFDnCFDCmBLCTL: <RSCFDn_base> + 1418_H + (20_H × m),
RSCFDnCFDCmBLCTH: <RSCFDn_base> + $141A_H$ + (20_H × m)

RSCFDnCFDCmBLCTLL: <RSCFDn_base> + 1418_H + (20_H × m),
RSCFDnCFDCmBLCTLH: <RSCFDn_base> + 1419_H + (20_H × m),
RSCFDnCFDCmBLCTHL: <RSCFDn_base> + $141A_H$ + (20_H × m),
RSCFDnCFDCmBLCTHH: <RSCFDn_base> + $141B_H$ + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	BLCLD	—	—	—	—	—	—	—	BLCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R/W

Table 23.104 RSCFDnCFDCmBLCT Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	BLCLD	BUS Load counter load When RSCFDnCFDCmBLCT.BLCLD is set, it is reset after a BUS Load counter value is loaded.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BLCE	BUS Load counter Enable 0: BUS Load counter disable. 1: BUS Load counter enable.

RSCFDnCFDCmBLCT.BLCLD

BUS Load counter load.

When RSCFDnCFDCmBLCT.BLCLD is set, the BUS Load counter value is loaded to RSCFDnCFDCmBLSTS.BLC and then the BUS Load counter is reset.

Read value is always 0.

RSCFDnCFDCmBLCT.BLCE

BUS Load counter enable.

The enabling signal of a BUS Load counter.

When this bit is 0, BUS Load counter stops. But counter is not clear.

The bit is cleared when the related channel enters CH_RESET mode.

This bit cannot be written in CH_STOP mode.

Write in this bit, after setting up a RSCFDnCFDCmNCFG register.

23.3.15.2 RSCFDnCFDCmBLSTS — Channel m Bus load Status Register

Access: RSCFDnCFDCmBLSTS register is a read-only register that can be read in 32-bit units
RSCFDnCFDCmBLSTSL, RSCFDnCFDCmBLSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCmBLSTSLL, RSCFDnCFDCmBLSTSLH, RSCFDnCFDCmBLSTSHL,
RSCFDnCFDCmBLSTSHH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDCmBLSTS: <RSCFDn_base> + $141C_H$ + (20_H × m)

RSCFDnCFDCmBLSTSL: <RSCFDn_base> + $141C_H$ + (20_H × m),
RSCFDnCFDCmBLSTSH: <RSCFDn_base> + $141E_H$ + (20_H × m)

RSCFDnCFDCmBLSTSLL: <RSCFDn_base> + $141C_H$ + (20_H × m),
RSCFDnCFDCmBLSTSLH: <RSCFDn_base> + $141D_H$ + (20_H × m),
RSCFDnCFDCmBLSTSHL: <RSCFDn_base> + $141E_H$ + (20_H × m),
RSCFDnCFDCmBLSTSHH: <RSCFDn_base> + $141F_H$ + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLC [31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLC [15:3]													—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.105 RSCFDnCFDCmBLSTS Register Contents

Bit Position	Bit Name	Function
31 to 3	BLC[31:3]	BUS Load counter Status
2 to 0	Reserved	When read, the value after reset is returned.

RSCFDnCFDCmBLSTS.BLC[31:3]

BUS Load counter status.

The bus load counter increases by clkc period while CAN bus is in an idle state.

When RSCFDnCFDCmBLCT.BLCLD bit is set, the BUS Load counter value is loaded to BLC bit and then the BUS Load counter is reset.

Bit 2 to Bit 0 is fixed 000_B.

These bits are set only by RS-CANFD module.

Writing any value has no influence on these bits.

23.3.16 Identifier Bits Alignment

Standard Identifier (11 bit) format: ID28 to ID18 is aligned to bit 10 to bit 0.

Extended Identifier (29 bit) format: ID28 to ID0 is aligned to bit 28 to bit 0.

For Standard Identifier format bit 11 to bit 28 should be 0 0000_H.

(1) Bit alignment for Standard Identifier (11 bit) format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDE=0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

(2) Bit alignment for Extended Identifier (29 bit) format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDE=0	RTR	—	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

23.3.17 Message Buffer Component Structure

23.3.17.1 Start Addresses

The start address for each of the Message Buffer Components is calculated using the number of related Message Buffer Components and the number of channels.

The start addresses for each register in the Message Buffer Component are depicted in **Table 23.106**.

Table 23.106 Message Buffer Component Register Start Addresses

b = Message Buffer Component Index	MBCP	Register	Regular Start Address*1
[0 .. 15]	RMBCPbm	RMID	2000_H + b*0080_H + m*800_H
		RMPTR	2004_H + b*0080_H + m*800_H
		RMFDSTS	2008_H + b*0080_H + m*800_H
		RMDFd	$200C_H$ + d*0004_H + b*0080_H + m*800_H
[0 .. 7]	RFMBCPb	RFID	6000_H + b*0080_H
		RFPTR	6004_H + b*0080_H
		RFFDSTS	6008_H + b*0080_H
		RFDFd	$600C_H$ + d*0004_H + b*0080_H
[0 .. 2]	CFMBCPbm	CFID	6400_H + b*0080_H + m*180_H
		CFPTR	6404_H + b*0080_H + m*180_H
		CFFDCSTS	6408_H + b*0080_H + m*180_H
		CFDFd	$640C_H$ + d*0004_H + b*0080_H + m*180_H
U2A-EVA, U2A16, U2A8: [0 .. 63] U2A6: [0 .. 15, 31, 32 .. 47, 63]	TMBCPbm	TMID	10000_H + b*0080_H + m*2000_H
		TMPTR	10004_H + b*0080_H + m*2000_H
		TMFDCTR	10008_H + b*0080_H + m*2000_H
		TMDFd	$1000C_H$ + d*0004_H + b*0080_H + m*2000_H

Note 1. Regular Start Address columns are offset address to RS-CANFD base addresses: <RSCFDn_base>.

The message buffer configuration consists of 4 types of message buffer components, namely RX Message Buffer Component (RSCFDnCFDRMBCPbm), RX FIFO Access Message Buffer Component (RSCFDnCFDRFMBCPb), Transmit/Receive FIFO Access Message Buffer Component (RSCFDnCFDCFMBCPbm) and TX Message Buffer Component (RSCFDnCFDTMBCPbm). Where b = the Message Buffer Component index that has a range that varies based on the type of message buffer component. For a summary of this configuration, refer to **Figure 23.27**. For a detailed description of the number of and the different types of message buffers, refer to **Section 23.7**.

As mentioned in **Section 23.4** each message buffer component consists of the following registers: Identifier (ID), Pointer (PTR) and Data Field (DFp) where p = the Data Field register index that has a range that varies based on the type of message buffer component.

Rc is the message buffer component register where c = Message Buffer Component Register index that has a range that varies based on the type of message buffer component.

A description of the registers, their associated bits and their accessibility is shown below the summary and detailed figures of each component.

In each of the figures below, a cell that contains '-' means reserved. Same behavior as the reserved bits for registers in **Section 23.4.2**.

23.3.17.2 RSCFDnCFDRMBCPbm — Receive Message Buffer Component bm

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Rc[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Rc[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.107 RSCFDnCFDRMBCPbm Register Contents

Bit Position	Bit Name	Function
31 to 0	Rc[31:0]	RX Message Buffer Component c Refer to **Table 23.108**, **Table 23.109** and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.

Below is an example to calculate index for U2A16:

(no_of_channels = 8)

(b = RX Message Buffer Component index = [0…no_of_RSCFDnCFDRMBCPs_per_channel-1])
(no_of_RSCFDnCFDRMBCPs_per_channel = No. of RX Message Buffer Components per Channel = 16)

Where the total number of RSCFDnCFDRMBCPs = no_of_RSCFDnCFDRMBCPs_per_channel * no_of_channels = 16 * 8 = 128

as shown in **Section 23.7, FIFO Buffers & Normal MB Configuration**.

(c = RX Message Buffer Component Register index = [0…no_of_REGs_per_RSCFDnCFDRMBCP-1]).

(no_of_REGs_per_RSCFDnCFDRMBCP = No. of Registers per RX Message Buffer Component =19).

RSCFDnCFDRMBCPbm.Rc[31:0]

RX Message Buffer Component c

The RX Message Buffer Component is made up of the following registers: RSCFDnCFDRMID, RSCFDnCFDRMPTR, RSCFDnCFDRMFDSTS, and RSCFDnCFDRMDFd. Refer to **Table 23.109** for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 23.108 RX Message Buffer Component Summary

	RX Message Buffer Component (RMBCP)
Rc	RX Message Buffer Component (RMBCP)
R0	RX Message Buffer (b) ID Register CHm
R1	RX Message Buffer (b) Pointer Register CHm
R2	RX Message Buffer (b) CANFD Status Register CHm
R3	RX Message Buffer (b) Data Field 0 Register CHm
R4	RX Message Buffer (b) Data Field 1 Register CHm
R5	RX Message Buffer (b) Data Field 2 Register CHm
R6	RX Message Buffer (b) Data Field 3 Register CHm
R7	RX Message Buffer (b) Data Field 4 Register CHm
R8	RX Message Buffer (b) Data Field 5 Register CHm
R9	RX Message Buffer (b) Data Field 6 Register CHm
R10	RX Message Buffer (b) Data Field 7 Register CHm
R11	RX Message Buffer (b) Data Field 8 Register CHm
R12	RX Message Buffer (b) Data Field 9 Register CHm
R13	RX Message Buffer (b) Data Field 10 Register CHm
R14	RX Message Buffer (b) Data Field 11 Register CHm
R15	RX Message Buffer (b) Data Field 12 Register CHm
R16	RX Message Buffer (b) Data Field 13 Register CHm
R17	RX Message Buffer (b) Data Field 14 Register CHm
R18	RX Message Buffer (b) Data Field 15 Register CHm

Table 23.109 RX Message Buffer Component Detailed

RX Message Buffer Component (RMBCP)																																	
Rc	**d**	**Symbol**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0	x	RSCFDnCFDRMID	RMIDE	RMRTR	—	RMID																												
R1	x	RSCFDnCFDRMPTR	RMDLC				—	—	—	—	—	—	—	—	—	—	—	—	RMTS															
R2	x	RSCFDnCFDRMFDSTS	RMPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	RSCFDnCFDRMDFd	RMDB((d*q)+(q-1))								RMDB((d*q)+(q-2))								RMDB((d*q)+(q-3))								RMDB((d*q)+(q-4))							
R[4...18]	[1…15]	RSCFDnCFDRMDFd	RMDB((d*q)+(q-1))								RMDB((d*q)+(q-2))								RMDB((d*q)+(q-3))								RMDB((d*q)+(q-4))							

23.3.17.3 RSCFDnCFDRMID — Receive Message Buffer ID Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.110 RSCFDnCFDRMID Register Contents

Bit Position	Bit Name	Function
31	RMIDE	RX Message Buffer IDE Bit 0_B: STD-ID is stored 1_B: EXT-ID is stored
30	RMRTR	RX Message Buffer RTR 0: Data Frame 1: Remote Frame
29	Reserved	When read, the value after reset is read.
28 to 0	RMID[28:0]	RX Message Buffer ID Field STD-ID / EXT-ID fields

The RX Message Buffer ID register stores the ID field, IDE bit and RTR bit of the received message.

RSCFDnCFDRMID.RMIDE

RX Message Buffer IDE Bit.

This bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX Message Buffer.

RSCFDnCFDRMID.RMRTR

RX Message Buffer RTR Bit.

This bit shows whether a Data Frame or a Remote Frame was stored in the RX Message Buffer.

NOTE

There are no remote frames in CANFD format. In case a CANFD frame was received the register reflects the state of the received value (RRS bit in FD frame format).

RSCFDnCFDRMID.RMID[28:0]

RX Message Buffer ID Field.

These are the bits of the STD-ID / EXT-ID fields of the message stored in the RX Message Buffer. For alignment of these bits in standard and extended frame format, see **Section 23.4.3, Global Mode – Channel Mode transition interactions**.

Refer to **Section 23.3.17.1, Start Addresses** for details of how to interpret the structure of this buffer component.

23.3.17.4 RSCFDnCFDRMPTR — Receive Message Buffer Pointer Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.111 RSCFDnCFDRMPTR Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	RX Message Buffer DLC Field No. of Data Bytes received in a CAN Frame
27 to 16	Reserved	When read, the value after reset is read.
15 to 0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message stored in the RX Message Buffer

The RX Message Buffer Pointer register stores the DLC and Timestamp fields for the received message.

RSCFDnCFDRMPTR.RMDLC[3:0]

RX Message Buffer DLC Field.

The number of Data Bytes that were received in the RX Message Buffer is stored in these bits.

Refer to Table 5 in ISO 11898-1 (2015) Specification for details defining the number of Data Bytes that were received.

NOTE

The max. capacity of the buffer belongs to the RSCFDnCFDRMNB.RMPLS.

RSCFDnCFDRMPTR.RMTS[15:0]

RX Message Buffer Timestamp Field

The Timestamp value taken at the capture point as configured by RSCFDnCFDGFDCFG.TSCCFG of the received message is stored in these bits.

23.3.17.5 RSCFDnCFDRMFDSTS — Receive Message Buffer CANFD Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMIFL[1:0]		—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.112 RSCFDnCFDRMFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 16	RMPTR[15:0]	RX Message Buffer Pointer Field RX Message Buffer Pointer
15 to 10	Reserved	When read, the value after reset is read.
9, 8	RMIFL[1:0]	RX Message Buffer Information label Field RX Message Buffer Information Label
7 to 3	Reserved	When read, the value after reset is read.
2	RMFDF	CANFD Format bit 0: Non CANFD frame received 1: CANFD frame received
1	RMBRS	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch
0	RMESI	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node

The RX Message Buffer CANFD Status register shows the status of the FDF, BRS and ESI bits, Pointer of the received CANFD frame.

RSCFDnCFDRMFDSTS.RMPTR[15:0]

RX Message Buffer Pointer Field.

The Pointer value from the related Global Acceptance Filter List entry is stored in these bits.

RSCFDnCFDRMFDSTS.RMIFL[1:0]

RX Message Buffer Information label Field.

The Information Label value from the related Global Acceptance Filter List entry is stored in these bits.

RSCFDnCFDRMFDSTS.RMFDF

CANFD Format bit.

This bit is the same value as the FDF bit of the received CANFD frame.

RSCFDnCFDRMFDSTS.RMBRS

Bit Rate Switch bit.

This bit is the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0_B, means a CAN2.0 frame is received, 0_B is always stored to this bit.

RSCFDnCFDRMFDSTS.RMESI

Error State Indicator bit.

This bit is the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0_B, means a CAN2.0 frame is received, 0_B is always stored to this bit.

23.3.17.6 RSCFDnCFDRMDFd — Receive Message Buffer Data Field d Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB(d*q) + (q-1)[7:0]								RMDB(d*q) + (q-2)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB(d*q) + (q-3)[7:0]								RMDB(d*q) + (q-4)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.113 RSCFDnCFDRMDFd Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB(d*q) + (q-1) [7:0]	RX Message Buffer Data Byte (d*q) + (q-1)
23 to 16	RMDB(d*q) + (q-2) [7:0]	RX Message Buffer Data Byte (d*q) + (q-2)
15 to 8	RMDB(d*q) + (q-3) [7:0]	RX Message Buffer Data Byte (d*q) + (q-3)
7 to 0	RMDB(d*q) + (q-4) [7:0]	RX Message Buffer Data Byte (d*q) + (q-4)

RSCFDnCFDRMDFd.RMDB(d*q) + (q-1)[7:0]

RX Message Buffer Data Byte (d*q) + (q-1)

Data Byte (d*q) + (q-1) of the Message stored in the RX Message Buffer. Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRMDFd.RMDB(d*q) + (q-2)[7:0]

RX Message Buffer Data Byte (d*q) + (q-2)

Data Byte (d*q) + (q-2) of the Message stored in the RX Message Buffer. Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRMDFd.RMDB(d*q) + (q-3)[7:0]

RX Message Buffer Data Byte (d*q) + (q-3)

Data Byte (d*q) + (q-3) of the Message stored in the RX Message Buffer. Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRMDFd.RMDB(d*q) + (q-4)[7:0]

RX Message Buffer Data Byte (d*q) + (q-4)

Data Byte (d*q) + (q-4) of the Message stored in the RX Message Buffer. Unused Data Bytes will be filled with 0_H.

23.3.17.7 RSCFDnCFDRFMBCPb — Receive FIFO Access Message Buffer Component b

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Rc[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Rc[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.114 RSCFDnCFDRFMBCPb Register Contents

Bit Position	Bit Name	Function
31 to 0	Rc[31:0]	RX FIFO Access Message Buffer Component c Refer to **Table 23.115**, **Table 23.116** and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.

Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(b = RX FIFO Access Message Buffer Component index = [0…no_of_RSCFDnCFDRFMBCPs-1])
(no_of_RSCFDnCFDRFMBCPs = No. of RX FIFO Access Message Buffer Components = 8) as shown in **Section 23.7, FIFO Buffers & Normal MB Configuration** (c = RX FIFO Access Message Buffer Component Register index = [0…no_of_REGs_per_RSCFDnCFDRFMBCP-1])
no_of_REGs_per_RSCFDnCFDRFMBCP = No. of Registers per RX FIFO Access Message Buffer Component = 19

RSCFDnCFDRFMBCPb.Rc[31:0]

RX FIFO Access Message Buffer Component c.

The RX FIFO Access Message Buffer Component is made up of the following registers: RSCFDnCFDRFID, RSCFDnCFDRFPTR, RSCFDnCFDRFFDSTS, and RSCFDnCFDRFDFd. Refer to **Table 23.116, RX FIFO Access Message Buffer Component Detailed** for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 23.115 RX FIFO Access Message Buffer Component Summary

RX FIFO Access Message Buffer Component (RFMBCP)	
Rc	
R0	RX FIFO Access ID Register
R1	RX FIFO Access Pointer Register
R2	RX FIFO Access CANFD Status Register
R3	RX FIFO Access Data Field 0 Register
R4	RX FIFO Access Data Field 1 Register
R5	RX FIFO Access Data Field 2 Register
R6	RX FIFO Access Data Field 3 Register
R7	RX FIFO Access Data Field 4 Register
R8	RX FIFO Access Data Field 5 Register
R9	RX FIFO Access Data Field 6 Register
R10	RX FIFO Access Data Field 7 Register
R11	RX FIFO Access Data Field 8 Register
R12	RX FIFO Access Data Field 9 Register
R13	RX FIFO Access Data Field 10 Register
R14	RX FIFO Access Data Field 11 Register
R15	RX FIFO Access Data Field 12 Register
R16	RX FIFO Access Data Field 13 Register
R17	RX FIFO Access Data Field 14 Register
R18	RX FIFO Access Data Field 15 Register
R[19 to 31]	—

Table 23.116 RX FIFO Access Message Buffer Component Detailed

RX FIFO Access Message Buffer Component (RFMBCP)																																	
Rc	**d**	**Symbol**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0	x	RSCFDnCFDRFID	RFIDE	RFRTR	—	RFID																												
R1	x	RSCFDnCFDRFPTR	RFDLC				—	—	—	—	—	—	—	—	—	—	—	—	RFTS															
R2	x	RSCFDnCFDRFFDSTS	RFPTR																—	—	—	—	—	—	RMIFL		—	—	—	—	—	RMFDF	RMBRS	RMESI
R3	0	RSCFDnCFDRFDFd	RFDB((d*q)+(q-1))								RFDB((d*q)+(q-2))								RFDB((d*q)+(q-3))								RFDB((d*q)+(q-4))							
R[4...18]	[1…15]	RSCFDnCFDRFDFd	RFDB((d*q)+(q-1))								RFDB((d*q)+(q-2))								RFDB((d*q)+(q-3))								RFDB((d*q)+(q-4))							
R[19...31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

23.3.17.8 RSCFDnCFDRFID — Receive FIFO Access ID Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.117 RSCFDnCFDRFID Register Contents

Bit Position	Bit Name	Function
31	RFIDE	RX FIFO Buffer IDE Bit 0_B: STD-ID has been received 1_B: EXT-ID has been received
30	RFRTR	RX FIFO Buffer RTR Bit 0_B: Data Frame 1_B: Remote Frame
29	Reserved	When read, the value after reset is read.
28 to 0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID / EXT-ID fields

The RX FIFO Access ID registers stores the ID field, IDE bit and RTR bit of the message.

RSCFDnCFDRFID.RFID[28:0]

RX FIFO Buffer ID Field.

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer. For alignment of these bits in standard and extended frame format, see **Section 23.4.3, Global Mode – Channel Mode transition interactions**.

RSCFDnCFDRFID.RFRTR

RX FIFO Buffer RTR Bit.

This bit shows whether a Data Frame or a Remote Frame was stored in the FIFO Buffer.

NOTE

There are no remote frames in CANFD format. In case a CANFD frame was received the register reflects the state of the received value (RRS bit in FD frame format).

RSCFDnCFDRFID.RFIDE

RX FIFO Buffer IDE Bit.

This bit shows whether message with Standard Identifier or Extended Identifier was received in the FIFO Buffer.

23.3.17.9 RSCFDnCFDRFPTR — Receive FIFO Access Pointer Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.118 RSCFDnCFDRFPTR Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC	RX FIFO Buffer DLC Field No. of Data Bytes received in a CAN Frame
27 to 16	Reserved	When read, the value after reset is read.
15 to 0	RFTS	RX FIFO Timestamp Value Timestamp value of the received CAN Frame

The FIFO Access Pointer registers store the DLC & Timestamp fields for the received message.

RSCFDnCFDRFPTR.RFDLC

RX FIFO Buffer DLC Field.

The number of Data Bytes that were received in the RX FIFO Buffer is stored in these bits.

Refer to Table 5 in ISO 11898-1 (2015) Specification for details defining the number of Data Bytes that were received.

RSCFDnCFDRFPTR.RFTS

RX FIFO Timestamp Value.

The Timestamp value taken at the capture point as configured by RSCFDnCFDGFDCFG.TSCCFG of the received message is stored in these bits.

23.3.17.10 RSCFDnCFDRFFDSTS — Receive FIFO Access CANFD Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDRFPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RFIFL[1:0]		—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.119 RSCFDnCFDRFFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 16	CFDRFPTR [15:0]	RX FIFO Buffer Pointer Field FIFO Buffer Pointer
15 to 10	Reserved	When read, the value after reset is read.
9, 8	RFIFL[1:0]	RX FIFO Buffer Information label Field RX FIFO Buffer Information Label
7 to 3	Reserved	When read, the value after reset is read.
2	RFFDF	CANFD Format bit 0_B: Non CANFD frame received 1_B: CANFD frame received
1	RFBRS	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch
0	RFESI	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node

The RX FIFO Access CANFD Status register shows the status of the FDF, BRS and ESI bits, Pointer of the received CANFD frame.

RSCFDnCFDRFFDSTS.CFDRFPTR[15:0]

RX FIFO Buffer Pointer Field

The Pointer value from the related Global Acceptance Filter List entry is stored in these bits.

RSCFDnCFDRFFDSTS.RFIFL[1:0]

RX FIFO Buffer Information Label Field.

The Information label value from the related Global Acceptance Filter List entry is stored in these bits.

RSCFDnCFDRFFDSTS.RFFDF

CANFD Format bit.

This bit is the same value as the FDF bit of the received CANFD frame.

RSCFDnCFDRFFDSTS.RFBRS

Bit Rate Switch bit.

This bit is the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0_B, means a CAN2.0 frame is received, 0_B is always stored to this bit.

RSCFDnCFDRFFDSTS.RFESI

Error State Indicator bit.

This bit is the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0_B, means a CAN2.0 frame is received, 0_B is always stored to this bit.

23.3.17.11 RSCFDnCFDRFDFd — Receive FIFO Access Data Field d Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB(d*q) + (q-1)[7:0]								RFDB(d*q) + (q-2)7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB(d*q) + (q-3)[7:0]								RFDB(d*q) + (q-4)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.120 RSCFDnCFDRFDFd Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB(d*q) + (q-1) [7:0]	FIFO Buffer Data Byte (d*q) + (q-1)
23 to 16	RFDB(d*q) + (q-2) [7:0]	FIFO Buffer Data Byte (d*q) + (q-2)
15 to 8	RFDB(d*q) + (q-3) [7:0]	FIFO Buffer Data Byte (d*q) + (q-3)
7 to 0	RFDB(d*q) + (q-4) [7:0]	FIFO Buffer Data Byte (d*q) + (q-4)

RSCFDnCFDRFDFd.RFDB(d*q) + (q-1)[7:0]

RX FIFO Buffer Data Byte (d*q) + (q-1).

Data Byte (d*q) + (q-1) of the Message present in the FIFO Buffer.

Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRFDFd.RFDB(d*q) + (q-2)[7:0]

RX FIFO Buffer Data Byte (d*q) + (q-2).

Data Byte (d*q) + (q-2) of the Message present in the FIFO Buffer. Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRFDFd.RFDB(d*q) + (q-3)[7:0]

RX FIFO Buffer Data Byte (d*q) + (q-3).

Data Byte (d*q) + (q-3) of the Message present in the FIFO Buffer. Unused Data Bytes will be filled with 0_H.

RSCFDnCFDRFDFd.RFDB(d*q) + (q-4)[7:0]

RX FIFO Buffer Data Byte (d*q) + (q-4).

Data Byte (d*q) + (q-4) of the Message present in the FIFO Buffer.

Unused Data Bytes will be filled with 0_H, according to there configured data payload size RSCFDnCFDRFCCx.RFPLS.

23.3.17.12 RSCFDnCFDCFMBCPbm — Transmit/Receive FIFO Access Message Buffer Component bm

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Rc[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Rc[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.121 RSCFDnCFDCFMBCPbm Register Contents

Bit Position	Bit Name	Function
31 to 0	Rc[31:0]	Transmit/Receive FIFO Access Message Buffer Component c Refer to **Table 23.122**, **Table 23.123** and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.

Below is an example to calculate index for U2A16:
(no_of_channels = 8)
(b = Transmit/Receive FIFO Message Buffer Component index = [0…no_of_RSCFDnCFDCFMBCPs_per_channel-1]) (no_of_RSCFDnCFDCFMBCPs_per_channel = No. of Transmit/Receive FIFO Message Buffer Components per Channel = 3) Where the total number of RSCFDnCFDCFMBCPs = no_of_RSCFDnCFDCFMBCPs_per_channel * no_of_channels = 3 * 8 = 24

as shown in **Figure 23.27**

(c = Transmit/Receive FIFO Message Buffer Component Register index = [0…no_of_REGs_per_RSCFDnCFDCFMBCP-1]) no_of_REGs_per_RSCFDnCFDCFMBCP = No. of Register per Transmit/Receive FIFO Message Buffer Component = 19

RSCFDnCFDCFMBCPbm.Rc[31:0]

Transmit/Receive FIFO Access Message Buffer Component c.

The Transmit/Receive FIFO Access Message Buffer Component is made up of the following registers: RSCFDnCFDCFID, RSCFDnCFDCFPTR, CFFDSTS, and RSCFDnCFDCFDFd. Refer to **Table 23.123** for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 23.122 Transmit/Receive FIFO Access Message Buffer Component Summary

Transmit/Receive FIFO Access Message Buffer Component (CFMBCP)	
Rc	
R0	Transmit/Receive FIFO Access ID Register
R1	Transmit/Receive FIFO Access Pointer Register
R2	Transmit/Receive FIFO Access CANFD Status Register
R3	Transmit/Receive FIFO Access Data Field 0 Register
R4	Transmit/Receive FIFO Access Data Field 1 Register
R5	Transmit/Receive FIFO Access Data Field 2 Register
R6	Transmit/Receive FIFO Access Data Field 3 Register
R7	Transmit/Receive FIFO Access Data Field 4 Register
R8	Transmit/Receive FIFO Access Data Field 5 Register
R9	Transmit/Receive FIFO Access Data Field 6 Register
R10	Transmit/Receive FIFO Access Data Field 7 Register
R11	Transmit/Receive FIFO Access Data Field 8 Register
R12	Transmit/Receive FIFO Access Data Field 9 Register
R13	Transmit/Receive FIFO Access Data Field 10 Register
R14	Transmit/Receive FIFO Access Data Field 11 Register
R15	Transmit/Receive FIFO Access Data Field 12 Register
R16	Transmit/Receive FIFO Access Data Field 13 Register
R17	Transmit/Receive FIFO Access Data Field 14 Register
R18	Transmit/Receive FIFO Access Data Field 15 Register
R[19 to 31]	-

Table 23.123 Transmit/Receive FIFO Access Message Buffer Component Detailed

Transmit/Receive FIFO Access Message Buffer Component (CFMBCP)																																
Rc	**d**	**Symbol**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0	x	RSCFDnCFDCFID	CFIDE	CFRTR	THLEN	CFID																												
R1	x	RSCFDnCFDCFPTR	CFDLC			—	—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	RSCFDnCFDCFFDSTS	CFPTR																—	—	—	—	—	—	CFIFL	—	—	—	—	—	CFFDF	CFBRS	CFESI
R3	0	RSCFDnCFDCFDFd	CFDB((d*q)+(q-1))								CFDB((d*q)+(q-2))								CFDB((d*q)+(q-3))								CFDB((d*q)+(q-4))							
R[4...18]	[1…15]	RSCFDnCFDCFDFd	CFDB((d*q)+(q-1))								CFDB((d*q)+(q-2))								CFDB((d*q)+(q-3))								CFDB((d*q)+(q-4))							
R[19...31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

23.3.17.13 RSCFDnCFDCFID — Transmit/Receive FIFO Access ID Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.124 RSCFDnCFDCFID Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE Bit 0_B: STD-ID will be transmitted or has been received 1_B: EXT-ID will be transmitted or has been received
30	CFRTR	Transmit/Receive FIFO Buffer RTR Bit 0_B: Data Frame 1_B: Remote Frame
29	THLEN	THL Entry enable TX FIFO Mode: 0_B: Entry will not be stored in THL after successful TX. 1_B: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Field STD-ID / EXT-ID fields

The Transmit/Receive FIFO Access ID registers store the ID field, IDE bit and RTR bit of the message.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

RSCFDnCFDCFID.CFIDE

Transmit/Receive FIFO Buffer IDE Bit.

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

RSCFDnCFDCFID.CFRTR

Transmit/Receive FIFO Buffer RTR Bit.

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

NOTE

There are no remote frames in CANFD format. In case a CANFD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CANFD transmission (TX or GW mode RSCFDnCFDCFID.CFFDF = 1_B) the bit is always transmitted dominant (Data Frame).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

RSCFDnCFDCFID.THLEN

THL Entry enable.

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

In GW mode, users cannot write data to the FIFO buffers.

RSCFDnCFDCFID.CFID[28:0]

Transmit/Receive FIFO Buffer ID Field.

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer. For alignment of these bits in standard and extended frame format, see **Section 23.4.3, Global Mode – Channel Mode transition interactions**.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

In GW mode, users cannot write data to the FIFO buffers.

23.3.17.14 RSCFDnCFDCFPTR — Transmit/Receive FIFO Access Pointer Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.125 RSCFDnCFDCFPTR Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Field No. of Data Bytes received in a CAN Frame, or to be transmitted in a CAN Frame
27 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Timestamp Value Timestamp value of the received CAN Frame (FIFO in RX Mode)

The Transmit/Receive FIFO Access Pointer registers store the DLC and Timestamp fields.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

RSCFDnCFDCFPTR.CFDLC[3:0]

Transmit/Receive FIFO Buffer DLC Field.

The number of Data Bytes that were received in the FIFO Buffer or are to be transmitted, is stored in these bits.

Refer to Table 5 in ISO 11898-1 (2015) Specification for details defining the number of Data Bytes In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

Users cannot read data for the other entries in the FIFO when configured in TX mode.

RSCFDnCFDCFPTR.CFTS

Transmit/Receive FIFO Timestamp Value.

The Timestamp value taken at the capture point as configured by RSCFDnCFDGFDCFG.TSCCFG of the received message is stored in these bits (if FIFO is configured in RX mode).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

23.3.17.15 RSCFDnCFDCFFDCSTS — Transmit/Receive FIFO Access CANFD Control/ Status Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CFIFL[1:0]		—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.126 RSCFDnCFDCFFDCSTS Register Contents

Bit Position	Bit Name	Function
31 to 16	CFPTR[15:0]	Transmit/Receive FIFO Buffer Pointer Field FIFO Message Buffer Pointer
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	CFIFL[1:0]	Transmit/Receive FIFO Buffer Information label Field Transmit/Receive FIFO Buffer Information Label
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CFFDF	CANFD Format bit 0_B: Non CANFD frame received or to transmit 1_B: CANFD frame received or to transmit
1	CFBRS	Bit Rate Switch bit 0_B: CANFD frame received or to transmit with no bit rate switch 1_B: CANFD frame received or to transmit with bit rate switch
0	CFESI	Error State Indicator bit 0_B: CANFD frame received from or to transmit by error active node 1_B: CANFD frame received from or to transmit by error passive node

The Transmit/Receive FIFO Access CANFD Status register shows the status of the FDF, BRS and ESI bits, Pointer of the received CANFD frame or the CANFD frame to transmit.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

RSCFDnCFDCFFDCSTS.CFPTR[15:0]

Transmit/Receive FIFO Buffer Pointer Field

If the Transmit/Receive FIFO is configured in TX Mode, the value programmed in RSCFDnCFDCFFDCSTS.CFPTR[15:0] will be stored together with further message information, to the TX History List after successful transmission of the message.

The Pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, users can write and read from FIFO buffers. In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

RSCFDnCFDCFFDCSTS.CFIFL[1:0]

Transmit/Receive FIFO Buffer Information Label Field

If the Transmit/Receive FIFO is configured in TX Mode, the value programmed in RSCFDnCFDCFFDCSTS.CFIFL[1:0] will be stored together with further message information, to the TX History List after successful transmission of the message.

The Information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, users can write and read from FIFO buffers. In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

RSCFDnCFDCFFDCSTS.CFFDF

CANFD Format bit

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, this bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0_B) or a CANFD frame (1_B).

In TX mode, the RS-CANFD module will either transmit a 0_B to indicate a CAN 2.0 frame is to be transmitted or a 1_B to indicate a CANFD frame is to be transmitted.

RSCFDnCFDCFFDCSTS.CFBRS

Bit Rate Switch bit

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, this bit is updated with the BRS bit value of the CANFD frame when it has been received, indicating whether there is a bit rate switch (1_B) or not (0_B) on the CANFD frame.

In RX or GW mode 0_B is stored to this bit when the received FDF bit is 0_B, means a CAN 2.0 frame is received.

In TX mode, the RS-CANFD module will either transmit a 0_B to indicate no bit rate switch in the frame to be transmitted or a 1_B to indicate a bit rate switch in the frame to be transmitted.

RSCFDnCFDCFFDCSTS.CFESI

Error State Indicator bit.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, this bit is updated with the ESI bit value of the CANFD frame when it has been received, indicating the error state of the transmitting node.

In RX or GW mode 0_B is stored to this bit when the received FDF bit is 0_B, means a CAN 2.0 frame is received.

In TX mode, if the RS-CANFD module is not in error passive, then this bit equals the write value, else it is don’t care and the bit is transmitted as 1_B on the CAN bus; indicating that this is an error passive node.

23.3.17.16 RSCFDnCFDCFDFd — Transmit/Receive FIFO Access Data Field d Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB(d*q) + (q-1)[7:0]								CFDB(d*q) + (q-2)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB(d*q) + (q-3)[7:0]								CFDB(d*q) + (q-4)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.127 RSCFDnCFDCFDFd Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB(d*q) + (q-1) [7:0]	Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-1) FIFO Buffer Data Byte (d*q) + (q-1)
23 to 16	CFDB(d*q) + (q-2) [7:0]	Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-2) FIFO Buffer Data Byte (d*q) + (q-2)
15 to 8	CFDB(d*q) + (q-3) [7:0]	Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-3) FIFO Buffer Data Byte (d*q) + (q-3)
7 to 0	CFDB(d*q) + (q-4) [7:0]	Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-4) FIFO Buffer Data Byte (d*q) + (q-4)

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

RSCFDnCFDCFDFd.CFDB(d*q) + (q-1)[7:0]

Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-1).

Data Byte (d*q) + (q-1)of the Message present in the FIFO Buffer. In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, unused Data Bytes will be filled with 0_H, according to their configured data payload size RSCFDnCFDCFCCk.CFPLS.

RSCFDnCFDCFDFd.CFDB(d*q) + (q-2)[7:0]

Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-2).

Data Byte (d*q) + (q-2) of the Message present in the FIFO Buffer. In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, unused Data Bytes will be filled with 0_H, according to their configured data payload size RSCFDnCFDCFCCk.CFPLS.

RSCFDnCFDCFDFd.CFDB(d*q) + (q-3)[7:0]

Transmit/Receive FIFO Buffer Data Byte(d*q) + (q-3).

Data Byte (d*q) + (q-3) of the Message present in the FIFO Buffer. In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, unused Data Bytes will be filled with 0_H, according to their configured data payload size RSCFDnCFDCFCCk.CFPLS.

RSCFDnCFDCFDFd.CFDB(d*q) + (q-4)[7:0]

Transmit/Receive FIFO Buffer Data Byte (d*q) + (q-4).

Data Byte (d*q) + (q-4) of the Message present in the FIFO Buffer. In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers. In GW mode, users cannot write data to the FIFO buffers.

In RX or GW mode, unused Data Bytes will be filled with 0_H, according to their configured data payload size RSCFDnCFDCFCCk.CFPLS.

23.3.17.17 RSCFDnCFDTMBCPbm — Transmit Message Buffer Component bm

Access: This register is a read-only register that can be read in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Rc[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Rc[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.128 RSCFDnCFDTMBCPbm Register Contents

Bit Position	Bit Name	Function
31 to 0	Rc[31:0]	TX Message Buffer Component c Refer to **Table 23.129**, **Table 23.130** and the descriptions that follow for a detailed. description of each register and its related bits, contained within this message buffer component.

Below is an example to calculate index for U2A16:

(no_of_channels = 8)

(b = TX Message Buffer Component index = [0…no_of_RSCFDnCFDTMBCPs_per_channel-1]) (no_of_RSCFDnCFDTMBCPs_per_channel = No. of TX Message Buffer Components per Channel = 64)

Where the total number of RSCFDnCFDTMBCPs= no_of_RSCFDnCFDTMBCPs_per_channel * no_of_channels = 64 * 8 = 512

as shown in **Figure 23.28**.

(c = TX Message Buffer Component Register index = [0…no_of_REGs_per_RSCFDnCFDTMBCP-1]) no_of_REGs_per_RSCFDnCFDTMBCP = No. of Register per TX Message Buffer Component = 19

RSCFDnCFDTMBCPbm.Rc[31:0]

TX Message Buffer Component c.

The TX Message Buffer Component is made up of the following registers: RSCFDnCFDTMID, RSCFDnCFDTMPTR, RSCFDnCFDTMFDCTR, and RSCFDnCFDTMDFd. Refer to **Table 23.129** for details of how to interpret the structure of this buffer component and how to access the respective registers.

Table 23.129 TX Message Buffer Component Summary

TX Message Buffer Component (TMBCP)	
Rc	
R0	TX Message Buffer (b) ID Register CHm
R1	TX Message Buffer (b) Pointer Register CHm
R2	TX Message Buffer (b) CANFD Status Register CHm
R3	TX Message Buffer (b) Data Field 0 Register CHm
R4	TX Message Buffer (b) Data Field 1 Register CHm
R5	TX Message Buffer (b) Data Field 2 Register CHm
R6	TX Message Buffer (b) Data Field 3 Register CHm
R7	TX Message Buffer (b) Data Field 4 Register CHm
R8	TX Message Buffer (b) Data Field 5 Register CHm
R9	TX Message Buffer (b) Data Field 6 Register CHm
R10	TX Message Buffer (b) Data Field 7 Register CHm
R11	TX Message Buffer (b) Data Field 8 Register CHm
R12	TX Message Buffer (b) Data Field 9 Register CHm
R13	TX Message Buffer (b) Data Field 10 Register CHm
R14	TX Message Buffer (b) Data Field 11 Register CHm
R15	TX Message Buffer (b) Data Field 12 Register CHm
R16	TX Message Buffer (b) Data Field 13 Register CHm
R17	TX Message Buffer (b) Data Field 14 Register CHm
R18	TX Message Buffer (b) Data Field 15 Register CHm

Table 23.130 TX Message Buffer Component Detailed

TX Message Buffer Component (TMBCP)																																
Rc	**d**	**Symbol**	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R0	x	RSCFDnCFDTMID	TMIDE	TMRTR	THLEN	TMID																												
R1	x	RSCFDnCFDTMPTR	CFDLC			—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R2	x	RSCFDnCFDTMFDCTR	TMPTR																—	—	—	—	—	—	TMIFL		—	—	—	—	—	TMFDF	TMBRS	TMESI
R3	0	RSCFDnCFDTMDFd	TMDB((d*q)+(q-1))								TMDB((d*q)+(q-2))								TMDB((d*q)+(q-3))								TMDB((d*q)+(q-4))							
R[4...18]	[1…15]	RSCFDnCFDTMDFd	TMDB((d*q)+(q-1))								TMDB((d*q)+(q-2))								TMDB((d*q)+(q-3))								TMDB((d*q)+(q-4))							

23.3.17.18 RSCFDnCFDTMID — Transmit Message Buffer ID Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.131 RSCFDnCFDTMID Register Contents

Bit Position	Bit Name	Function
31	TMIDE	TX Message Buffer IDE Bit 0_B: STD-ID will be transmitted 1_B: EXT-ID will be transmitted
30	TMRTR	TMRTR TX Message Buffer RTR Bit 0_B: Data Frame 1_B: Remote Frame
29	THLEN	Tx History List Entry 0_B: Entry not stored in THL after successful TX 1_B: Entry stored in THL after successful TX
28 to 0	TMID[28:0]	TX Message Buffer ID Field STD-ID / EXT-ID fields

Each TX Message Buffer ID register is used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

RSCFDnCFDTMID.TMIDE

TX Message Buffer IDE Bit.

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from this TX Message Buffer. Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMID.TMRTR

TX Message Buffer RTR Bit.

This bit selects whether a Data Frame or a Remote Frame will be transmitted from this TX Message Buffer.

NOTE

There are no remote frames in CANFD format. In case of CANFD transmission(RSCFDnCFDTMFDCTR.CFFDF = 1_B) the bit is always transmitted dominant (Data Frame).

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMID.THLEN

Tx History List Entry.

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMID.TMID[28:0]

TX Message Buffer ID Field.

These are the bits of the STD-ID / EXT-ID fields of the message stored in this TX MB. For alignment of these bits in standard and extended frame format, see **Section 23.4.3, Global Mode – Channel Mode transition interactions**.

Users should not write to these bits when corresponding channel is in CH_STOP.

23.3.17.19 RSCFDnCFDTMPTR — Transmit Message Buffer Pointer Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.132 RSCFDnCFDTMPTR Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	TX Message Buffer DLC Field No. of Data Bytes to be transmitted in a CAN Frame
27 to 0	Reserved	When read, the value after reset is read. When writing, write the value after reset.

Each TX Message Buffer Pointer register is used to store the DLC fields of the message to transmit from the associated buffer.

RSCFDnCFDTMPTR.TMDLC[3:0]

TX Message Buffer DLC Field

These bits select the number of Data Bytes that will be transmitted from this TX Message Buffer if corresponding TMRTR bit is configured as 0_B.

Refer to Table 5 in ISO 11898-1 (2015) Specification for details defining the number of Data Bytes that will be transmitted.

Users should not write to these bits when corresponding channel is in CH_STOP.

23.3.17.20 RSCFDnCFDTMFDCTR — Transmit Message Buffer CANFD Control Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMPTR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TMIFL[1:0]		—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.133 RSCFDnCFDTMFDCTR Register Contents

Bit Position	Bit Name	Function
31 to 16	TMPTR[15:0]	TX Message Buffer Pointer Field TX Message Buffer Pointer
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TMIFL[1:0]	TMIFL TX Message Buffer Information Label Field TX Message Buffer Information Label
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMFDF	CANFD Format bit 0_B: Non CANFD frame to transmit 1_B: CANFD frame to transmit
1	TMBRS	Bit Rate Switch bit 0_B: CANFD frame to transmit with no bit rate switch 1_B: CANFD frame to transmit with bit rate switch
0	TMESI	Error State Indicator bit 0_B: CANFD frame to transmit by error active node 1_B: CANFD frame to transmit by error passive node

The TX Message Buffer CANFD Control register shows the status of the FDF, BRS and ESI bits and Pointer fields of the CANFD frame to be transmitted.

RSCFDnCFDTMFDCTR.TMPTR[15:0]

TX Message Buffer Pointer Field.

The Pointer value is stored in these bits. It will be copied, together with further message information, in the TX History List after successful transmission of the message.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMFDCTR.TMIFL[1:0]

TX Message Buffer Information label Field.

The Information label value is stored in these bits. It will be copied, together with further message information, in the TX History List after successful transmission of the message.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMFDCTR.TMFDF

CANFD Format bit.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMFDCTR.TMBRS

Bit Rate Switch bit.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMFDCTR.TMESI

Error State Indicator bit.

If the channel is not in error passive, then this bit equals the write value, else it is don't care and the bit is transmitted as 1_B on the CAN bus; indicating that this is an error passive node.

Users should not write to these bits when corresponding channel is in CH_STOP.

23.3.17.21 RSCFDnCFDTMDFd — Transmit Message Buffer Data Field d Register

Access: This register can be read or written in 32-bit units.

Address: Refer to **Table 23.106**

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB(d*q) + (q-1)[7:0]								TMDB(d*q) + (q-2)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB(d*q) + (q-3)[7:0]								TMDB(d*q) + (q-4)[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.134 RSCFDnCFDTMDFd Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB(d*q) + (q-1) [7:0]	TX Message Buffer Data Byte (d*q) + (q-1)
23 to 16	TMDB(d*q) + (q-2) [7:0]	TX Message Buffer Data Byte (d*q) + (q-2)
15 to 8	TMDB(d*q) + (q-3) [7:0]	TX Message Buffer Data Byte (d*q) + (q-3)
7 to 0	TMDB(d*q) + (q-4) [7:0]	TX Message Buffer Data Byte (d*q) + (q-4)

RSCFDnCFDTMDFd.TMDB(d*q) + (q-1)

TX Message Buffer Data Byte (d*q) + (q-1).

Data Byte (d*q) + (q-1) of the Message stored in the TX Message Buffer.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMDFd.TMDB(d*q) + (q-2)

TX Message Buffer Data Byte (d*q) + (q-2).

Data Byte (d*q) + (q-2) of the Message stored in the TX Message Buffer.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMDFd.TMDB(d*q) + (q-3)

TX Message Buffer Data Byte (d*q) + (q-3).

Data Byte (d*q) + (q-3) of the Message stored in the TX Message Buffer.

Users should not write to these bits when corresponding channel is in CH_STOP.

RSCFDnCFDTMDFd.TMDB(d*q) + (q-4)

TX Message Buffer Data Byte (d*q) + (q-4).

Data Byte (d*q) + (q-4) of the Message stored in the TX Message Buffer.

Users should not write to these bits when corresponding channel is in CH_STOP.

23.4 CAN Modes

The RS-CANFD module has four global modes to control entire RS-CANFD module status and four channel modes to control individual channel status. Details of global modes are described in **Section 23.4.1, Global Modes**, and details of channel modes are described in **Section 23.4.2, Channel Modes**.

- Global stop mode: Stops clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

23.4.1 Global Modes

Figure 23.2 shows the transitions of global modes. Refer to **Section 23.4.3.1, Global Mode change timing**.

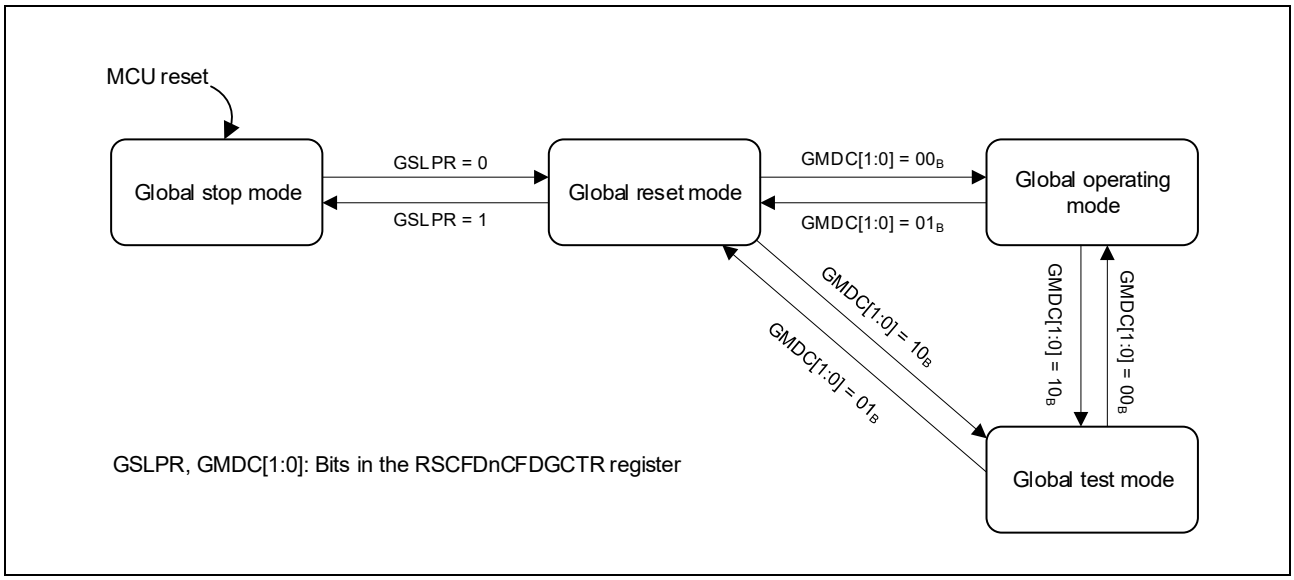


Figure 23.2 Transitions of Global Modes

Table 23.135 Possible CAN channel Modes versus Global Module Modes

Current Global mode \ Target Global mode	Stop	Reset	Test	Operating
Stop		Ch-Stop: keep Ch-Reset: N/A Ch-Halt: N/A Ch-Comm: N/A		
Reset	Ch-Stop: keep Ch-Reset:→ Ch-Stop Ch-Halt: N/A Ch-Comm: N/A		Ch-Stop: keep Ch-Reset: keep Ch-Halt: N/A Ch-Comm: N/A	Ch-Stop: keep Ch-Reset: keep Ch-Halt: N/A Ch-Comm: N/A
Test		Ch-Stop: keep Ch-Reset: keep Ch-Halt:→ Ch-Reset Ch-Comm: N/A		Ch-Stop: keep Ch-Reset: keep Ch-Halt: keep Ch-Comm: N/A
Operating		Ch-Stop: keep Ch-Reset: keep Ch-Halt:→ Ch-Reset Ch-Comm:→ Ch-Reset	Ch-Stop: keep Ch-Reset: keep Ch-Halt: keep Ch-Comm:→ Ch-Halt	

23.4.1.1 Global Stop Mode

After the release of the hardware reset or after setting and clearing a RSCFDnCFDGRSTC.SRST bit, the RS-CANFD module automatically enters Global Stop Mode.

The RS-CANFD module will also enter this mode, when the Global Stop Request bit is set while it is in Global Reset Mode.

This control bit cannot be set in Global Test Mode or Global Operating Mode.

Setting the Global Stop Request bit will set all Channel Stop Request bits and force all channels into the Channel Stop Mode.

Stop Mode is used for power saving purpose. When RS-CANFD module is in Global Stop Mode, only the clock for CPU write access to the Global Stop Mode Request bit is active. All other clocks are stopped and all other functions of the RS-CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting Global Stop Request bit, it is necessary to confirm that the Global Stop status has been updated indicating successful transition to Global Stop Mode before the Global Stop Request bit can be cleared again.

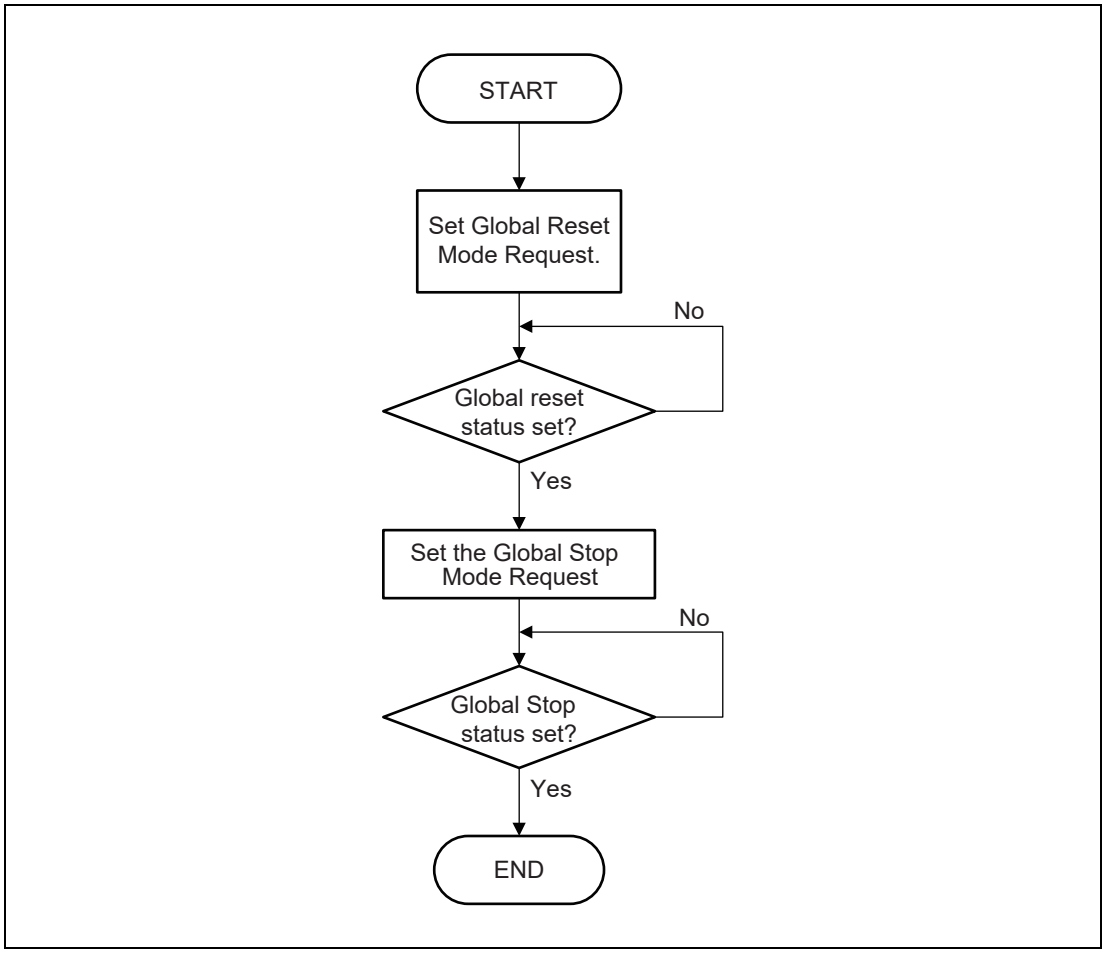


Figure 23.3 Procedure for entering global Stop Mode

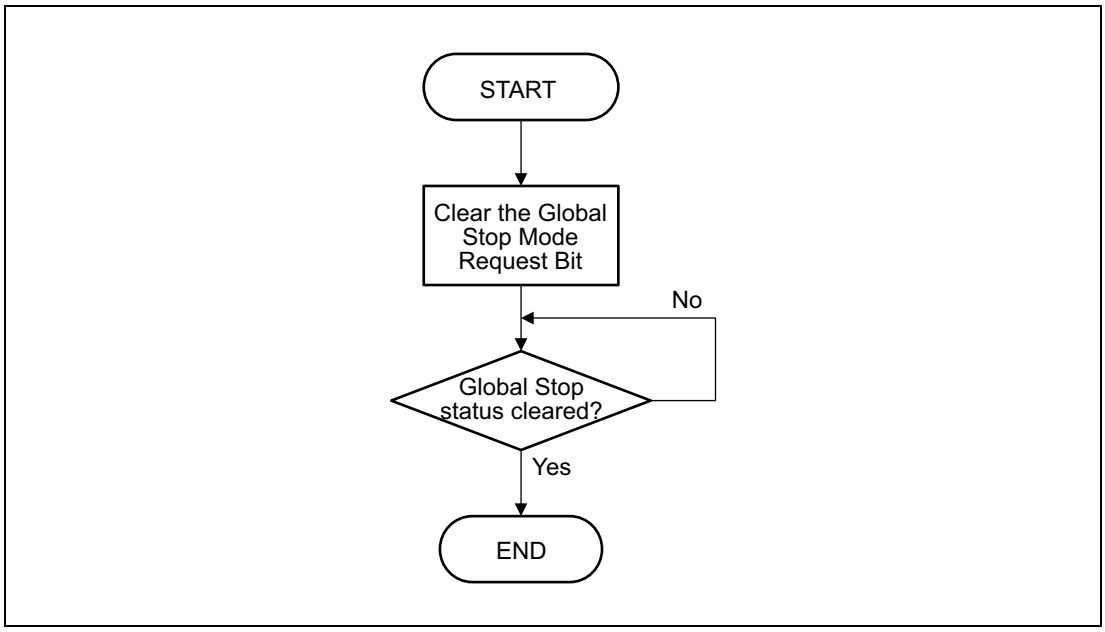


Figure 23.4 Procedure for exiting global Stop Mode

23.4.1.2 Global Reset Mode

The RS-CANFD module enters this mode in the following ways:

Global Mode Control RSCFDnCFDGCTR.GMDC in the Global Control Register is configured for Global Reset Mode while the RS-CANFD module is in Global Test Mode or Global Operating Mode

Global Stop Mode request bit is cleared while RS-CANFD module is in Global Stop Mode

In Global Reset Mode, all RS-CANFD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and all channel TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU Reset values and the RS-CANFD module can be configured.

Refer to **Section 23.4.3, Global Mode – Channel Mode transition interactions** for detailed description of the behavior of all registers when transition to Global Reset Mode is performed.

Setting the global mode to Reset by setting the Global Mode Control bits RSCFDnCFDGCTR.GMDC in the Global Control Register to 01_B will set all Channel Mode Control bits RSCFDnCFDCmCTR.CHMDC in the Channel Control Register to 01_B and force all channels into the Channel Reset Mode.

For channels that are already in Channel Reset Mode or Channel Stop Mode this automatic transition is not performed (RSCFDnCFDCmCTR.CHMDC of related channel already set to 01_B).

After setting Global Mode Control RSCFDnCFDGCTR.GMDC to Reset Mode it is necessary to confirm that the Reset Mode Status RSCFDnCFDGSTS.GRSTSTS in the Global Status Register has been updated indicating successful transition to Global Reset Mode before RSCFDnCFDGCTR.GMDC can be changed again.

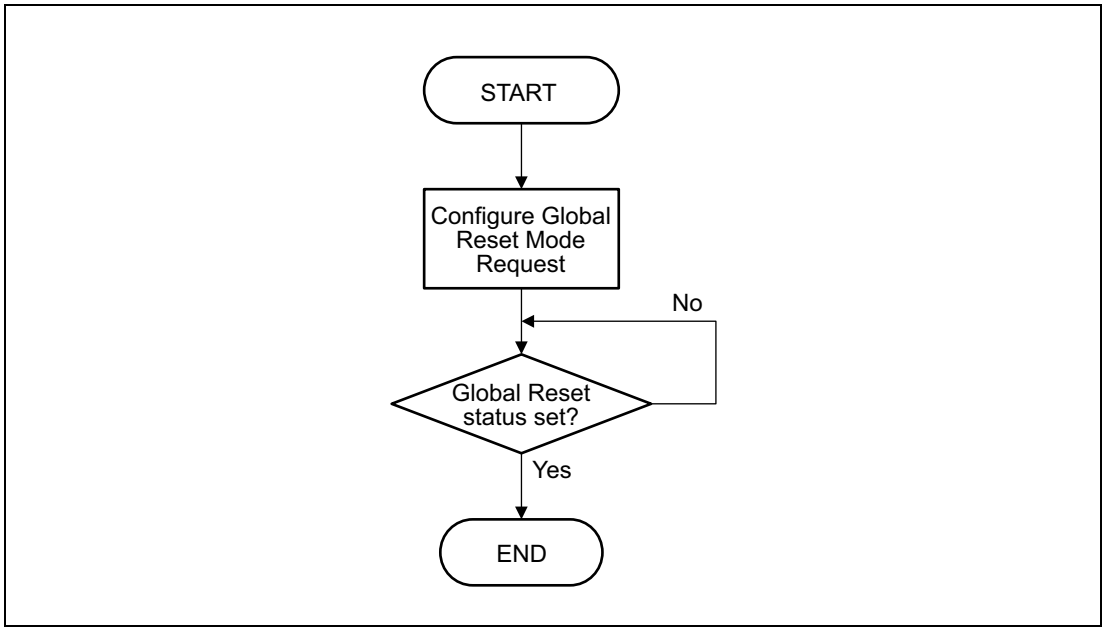


Figure 23.5 Procedure for entering global Reset Mode

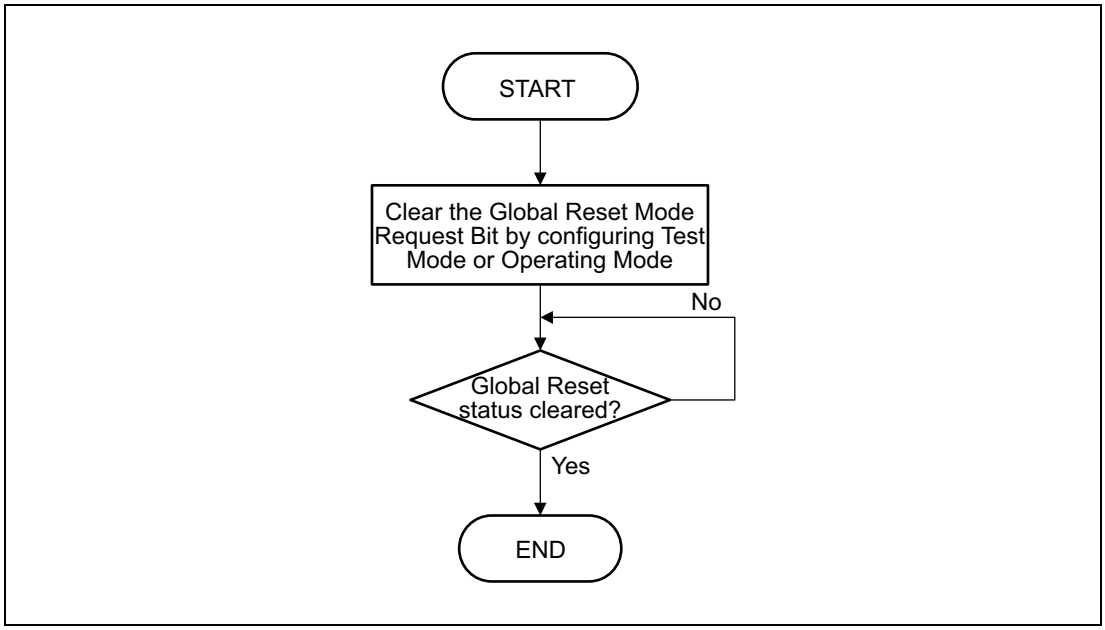


Figure 23.6 Procedure for exiting global Reset Mode

23.4.1.3 Global Test Mode

The RS-CANFD module enters this mode in the following ways:

Global Mode Control RSCFDnCFDGCTR.GMDC in the Global Control Register is configured for Global Test Mode while the RS-CANFD module is in Global Reset Mode

- The channels will be in either Channel Reset Mode or Channel Stop Mode and will remain in this mode

Global Mode Control RSCFDnCFDGCTR.GMDC in the Global Control Register is configured for Global Test Mode while the RS-CANFD module is in Global Operating Mode

- All channels in Channel Reset Mode, Channel Halt Mode or Channel Stop Mode will remain in this mode
- All channels in Channel Communication Mode will transit to Channel Halt Mode
- Global Test Mode Status bit is set when all channels have left Channel Communication Mode

If a transmission or reception is ongoing for a channel the transition to Channel Halt Mode is delayed until the completion of the communication.

Similarly, if a channel is in Bus-Off, the full Bus-Off recovery sequence may be delayed depending on the channel configuration.

In the Global Test Mode, all communications are suspended and RS-CANFD logic will not cause any change to status and flag registers (only when a channel is in the Bus-Off its REC and TEC values are cleared).

Also the test mode configuration and control registers are not initialized in this mode.

The Global Test Mode should be used to configure global module test modes.

Refer to **Section 23.4.3, Global Mode – Channel Mode transition interactions** for a detailed description of the behavior of all registers when transition to Global Test Mode is performed.

Setting the global mode to Test by setting the Global Mode Control bits RSCFDnCFDGCTR.GMDC in the Global Control Register to 10_B will set all Channel Mode Control bits RSCFDnCFDCmCTR.CHMDC in the Channel Control Register to 10_B for the channels that are in Channel Communication Mode and force these channels into the Channel Halt Mode.

For channels that are already in Channel Reset Mode, Channel Halt Mode or Channel Stop Mode this automatic transition is not performed.

Therefore, the Global Test Mode request can be used to shut down all CAN channel communications without loss of messages and disruption on the related CAN Bus (no interruption of reception/ transmission processes on the channels).

After setting Global Mode Control RSCFDnCFDGCTR.GMDC to Test Mode it is necessary to confirm that the Test Mode status RSCFDnCFDGSTS.GHLTSTS in the Global Status Register has been updated indicating successful transition to Global Test Mode. User should not do any other SFR setting until confirming RSCFDnCFDGSTS.GHLTSTS is set.

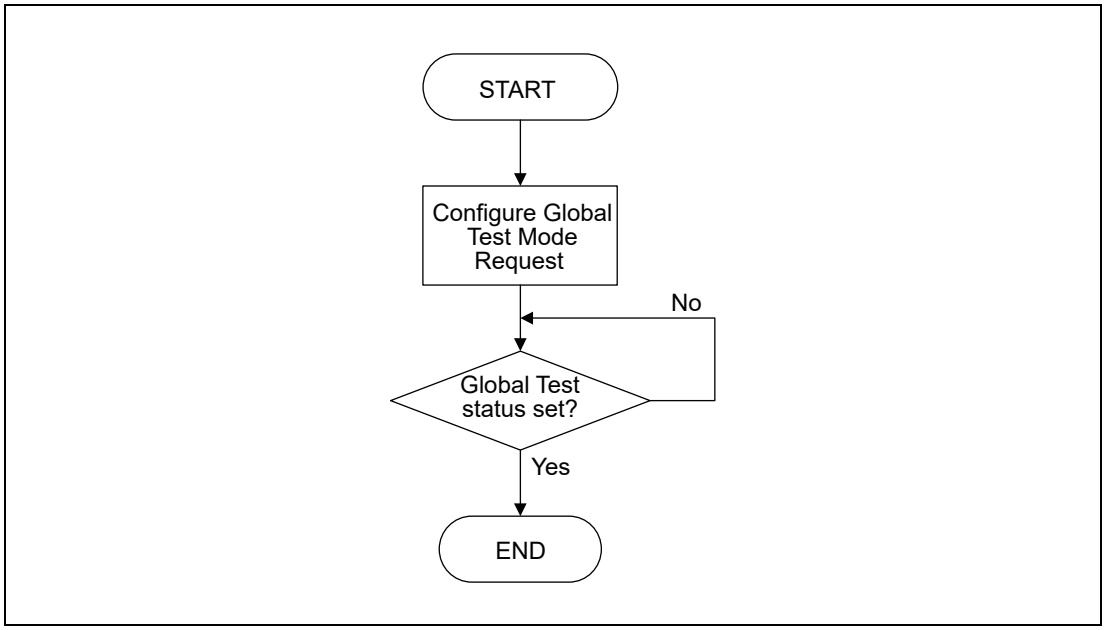


Figure 23.7 Procedure for entering global Test Mode

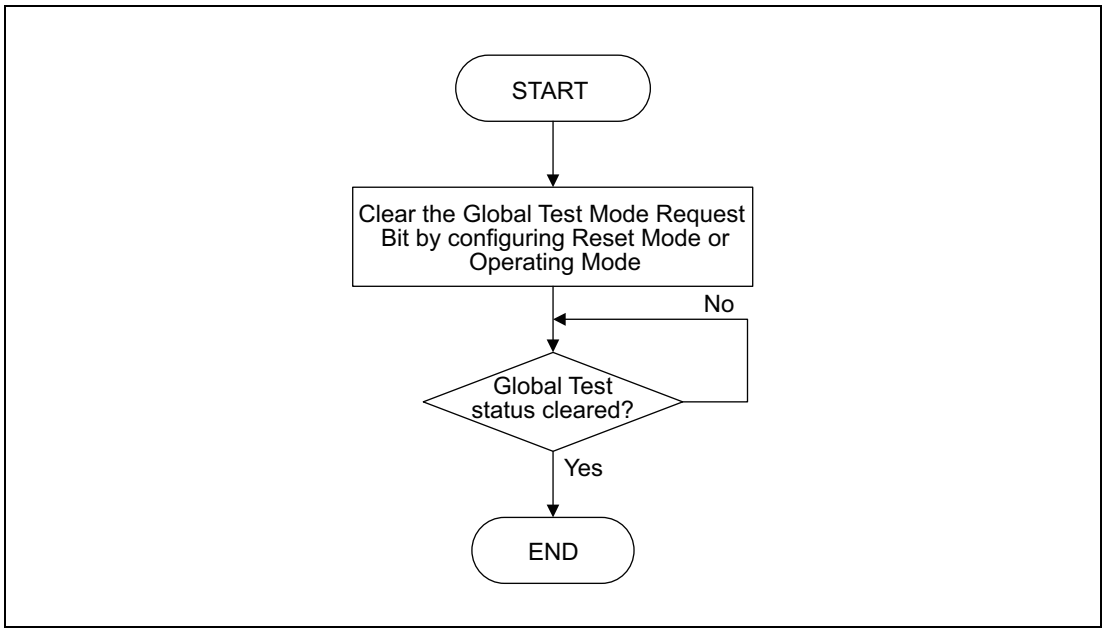


Figure 23.8 Procedure for exiting global Test Mode

23.4.1.4 Global Operating Mode

The RS-CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operating Mode.

CAN channels can only be set to Channel Operation Mode and start CAN communication when RS-CANFD is in Global Operating Mode.

After setting Global Mode Control RSCFDnCFDGCTR.GMDC to Global Operating Mode it is necessary to confirm that the Global Reset Mode status RSCFDnCFDGSTS.GRSTSTS and the Global Test Mode status RSCFDnCFDGSTS.GHLTSTS in the Global Status Register have been cleared indicating successful transition to Global Operating Mode before RSCFDnCFDGCTR.GMDC can be changed again.

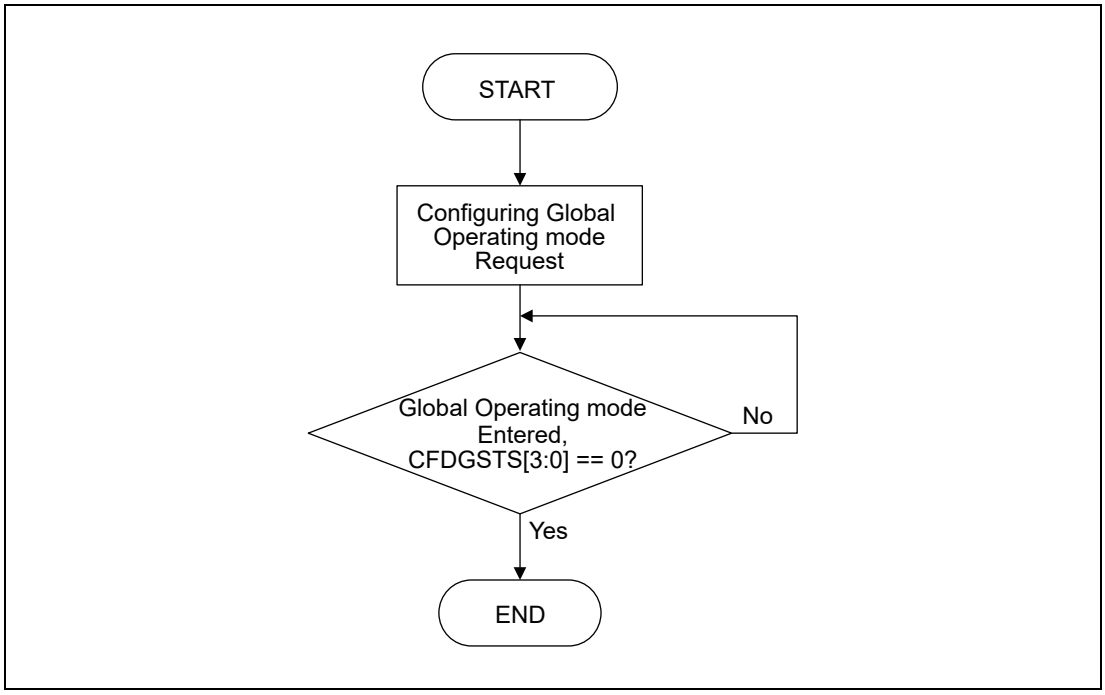


Figure 23.9 Procedure for Entering Global Operating Mode

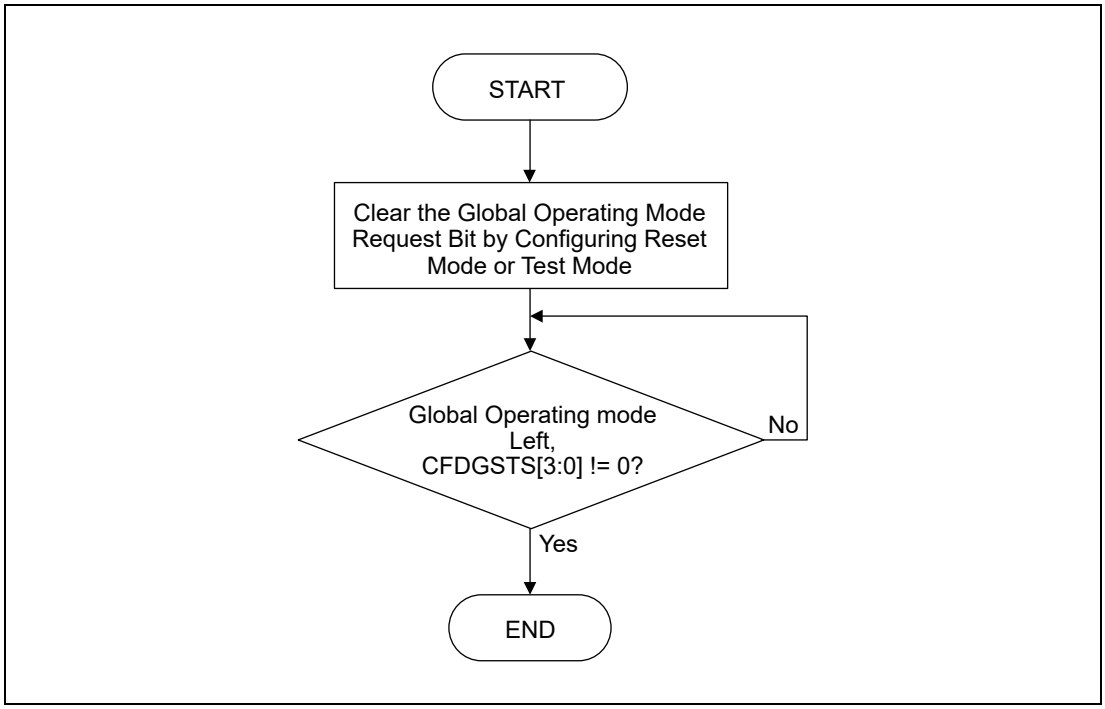
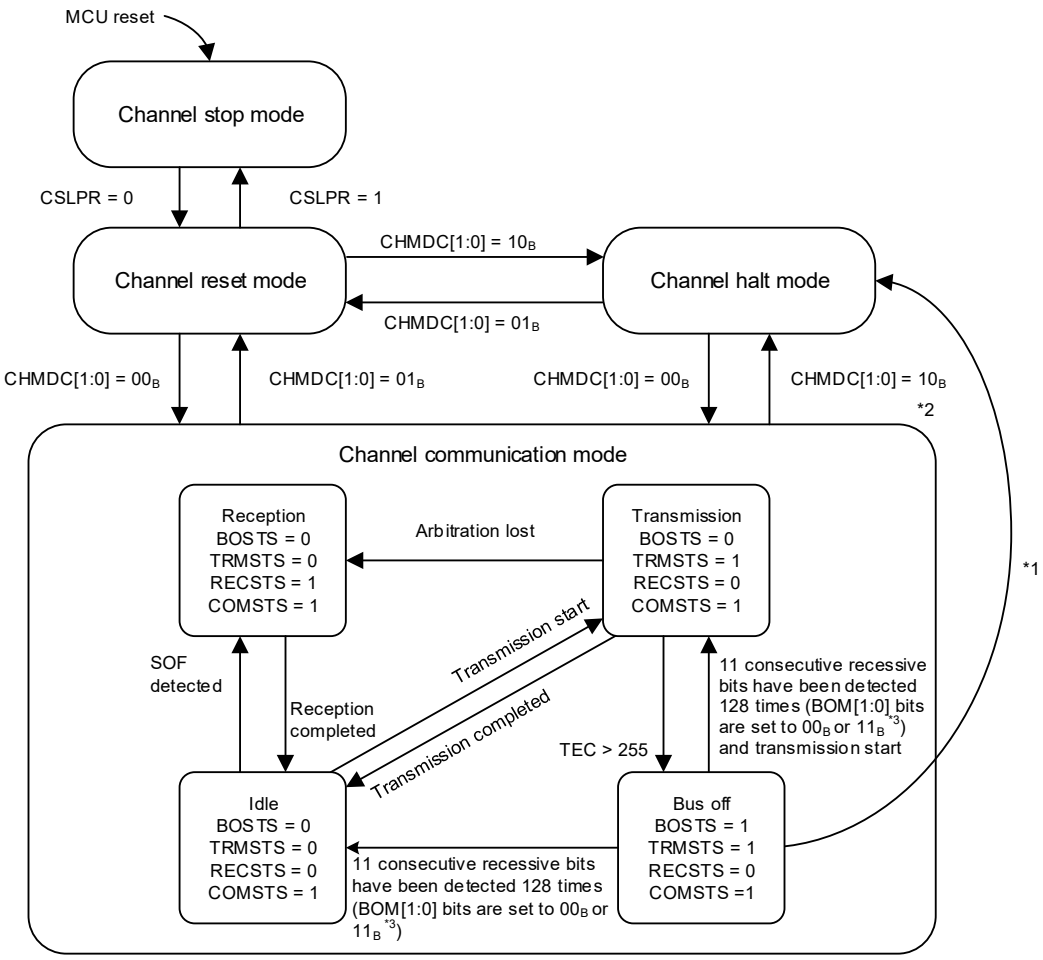


Figure 23.10 Procedure for Exiting Global Operating Mode

23.4.2 Channel Modes

Figure 23.11 shows a channel mode state transition chart. Refer to **Section 23.4.3.2, Channel Mode change timing**.



Note: CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RSCFDnCFDCmCTR register (m = 0 to 7)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RSCFDnCFDCmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked to the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode

Note 3. When 11 consecutive recessive bits are detected 128 times before the CHMDC[1:0] bits are set to 10_B

Figure 23.11 Channel Mode State Transition Chart

23.4.2.1 CAN Channel Stop Mode

After the release of the hardware reset or after setting and clearing a RSCFDnCFDGRSTC.SRST bit, each CAN channel of the RS-CANFD module automatically enters Channel Stop Mode.

Each CAN channel will also enter this mode, when the related Channel Stop Mode Request bit is set while the CAN channel is in Channel Reset Mode.

This control bit should not be set in Channel Halt Mode or Channel Communication Mode.

Entering the CAN channel Stop Mode instantly stops the clock supplied to the CAN channel unit and thereby reduces power consumption.

After setting Channel Stop Mode request bit it is necessary to confirm that the Channel Stop Mode Status has been updated indicating successful transition to Channel Stop Mode before the Channel Stop Mode request bit can be cleared again.

During Channel Stop Mode, users cannot write to channel related registers. Read operation is still possible.

23.4.2.2 CAN channel Reset Mode

An RS-CANFD CAN channel enters this mode in the following ways:

Channel Mode Control RSCFDnCFDCmCTR.CHMDC in the Channel Control Register is configured for Channel Reset Mode while the related CAN channel is in Channel Halt Mode or Channel Communication Mode

Channel Stop Mode request bit is cleared while the related CAN channel is in Channel Stop Mode

Global Mode Control RSCFDnCFDGCTR.GMDC is set to Global Reset Mode and CAN channel is not in Channel Stop Mode or Channel Reset Mode

In Channel Reset Mode, all CAN channel status and flag registers are initialized.

Additionally all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the RS-CANFD module CAN channel can be configured for communication.

Refer to **Section 23.4.3, Global Mode – Channel Mode transition interactions** for detailed description of the behavior of all registers when transition to Channel Reset Mode is performed.

After setting Channel Mode Control RSCFDnCFDCmCTR.CHMDC to Channel Reset Mode, it is necessary to confirm that the Reset Mode Status RSCFDnCFDCmSTS .CRSTSTS in the related Channel Status Register has been updated indicating successful transition to Channel Reset Mode before the related RSCFDnCFDCmCTR.CHMDC can be changed again.

Refer to **Table 23.136** below regarding the influence of transition to Channel Reset Mode while CAN communication is ongoing.

23.4.2.3 CAN channel Halt Mode

An RS-CANFD CAN channel enters this mode in the following ways:

Channel Mode Control RSCFDnCFDCmCTR.CHMDC in the Channel Control Register is configured for Channel Halt Mode while the related CAN channel is in Channel Reset Mode or Channel Communication Mode

Global Mode Control RSCFDnCFDGCTR.GMDC is set to Global Test Mode and CAN channel is in Channel Communication Mode

In Channel Halt Mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt Mode entry (except for the Bus-Off case where REC and TEC values are cleared for this channel).

In addition, the channel test mode configuration and control registers are not initialized in this mode.

The Channel Halt Mode should be used to configure channel test modes.

Refer to **Section 23.4.3, Global Mode – Channel Mode transition interactions** for detailed description of the behavior of all registers when transition to Channel Halt Mode is performed.

After setting Channel Mode Control RSCFDnCFDCmCTR.CHMDC to Channel Halt Mode it is necessary to confirm that the Halt Mode status RSCFDnCFDCmSTS .CHLTSTS in the related Channel Status Register has been updated indicating successful transition to Channel Halt Mode before the related RSCFDnCFDCmCTR.CHMDC can be changed again.

Refer to **Table 23.136** regarding the influence of transition to Channel Halt Mode while CAN communication is ongoing.

Table 23.136 Behavior in CAN Reset / Halt Mode

State / Mode	Receiver	Transmitter	Bus-Off
CAN channel Reset Mode (RSCFDnCFDCmCTR.CHMDC= 01_B)	The CAN channel transits to Channel Reset Mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to Channel Reset Mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to Channel Reset Mode without waiting for the completion of the Bus-Off Recovery.
CAN channel Halt Mode (RSCFDnCFDCmCTR.CHMDC= 10_B)	CAN channel transits to Channel Halt Mode at the end of the ongoing reception or error.*2	CAN channel transits to Channel Halt Mode after completion of the ongoing transmission.	When RSCFDnCFDCmCTR.BOM is set to 00_B, a Channel Halt Mode request will be accepted only after the completion of the full Bus-Off Recovery sequence. When RSCFDnCFDCmCTR.BOM is set to 10_B, then, the CAN channel transits automatically to Channel Halt Mode after waiting for the completion of the Bus-Off Recovery. When RSCFDnCFDCmCTR.BOM is set to 01_B, then, the CAN channel transits automatically to Channel Halt Mode without waiting for the completion of the Bus-Off Recovery. When RSCFDnCFDCmCTR.BOM is set to 11_B, the CAN channel transits to Channel Halt Mode as soon as the Channel Halt Mode is requested (without waiting for the completion of the Bus-Off Recovery).

Note 1. If the entry to Channel Reset Mode is required only at the end of an ongoing communication, then Channel Halt Mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset Mode. After the CAN channel enters Channel Halt Mode the Channel Reset Mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, Application SW can detect this situation by monitoring the channel related BusLock Flag and resolve lock condition by setting the CAN channel to Channel Reset Mode.

23.4.2.4 CAN Channel Communication Mode

The Channel Communication Mode is activated by setting the RSCFDnCFDCmCTR.CHMDC bits to 00_B. If 11 consecutive recessive bits are detected after entering the CAN Communication Mode, then the RSCFDnCFDCmSTS.COMSTS bit is set and the CAN channel:

Enables channel's communication functions allowing the channel to become an active node on the CAN network

releases the internal fault confinement logic including receive and transmit error counters

At this point, it can start transmission and reception of CAN messages.

Within the CAN channel Communication Mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see **Figure 23.12**):

Channel idle: The CAN channel is neither receiving nor transmitting.

Channel receives: The channel is receiving a CAN message sent by another CAN node.

Channel transmits: The channel is transmitting a CAN message.

NOTE

The channel may receive its own message simultaneously when Self Test Mode is enabled.

Channel is in Bus-Off state: The CAN channel is cut-off from CAN Bus communication

After setting Channel Mode Control RSCFDnCFDCmCTR.CHMDC to Communication Mode, it is necessary to confirm that the Channel Reset Mode Status RSCFDnCFDCmSTS.CRSTSTS and the Channel Halt Mode Status RSCFDnCFDCmSTS.CHLTSTS in the Channel Status Register have been updated indicating successful transition to Channel Communication Mode before the related RSCFDnCFDCmCTR.CHMDC can be changed again.

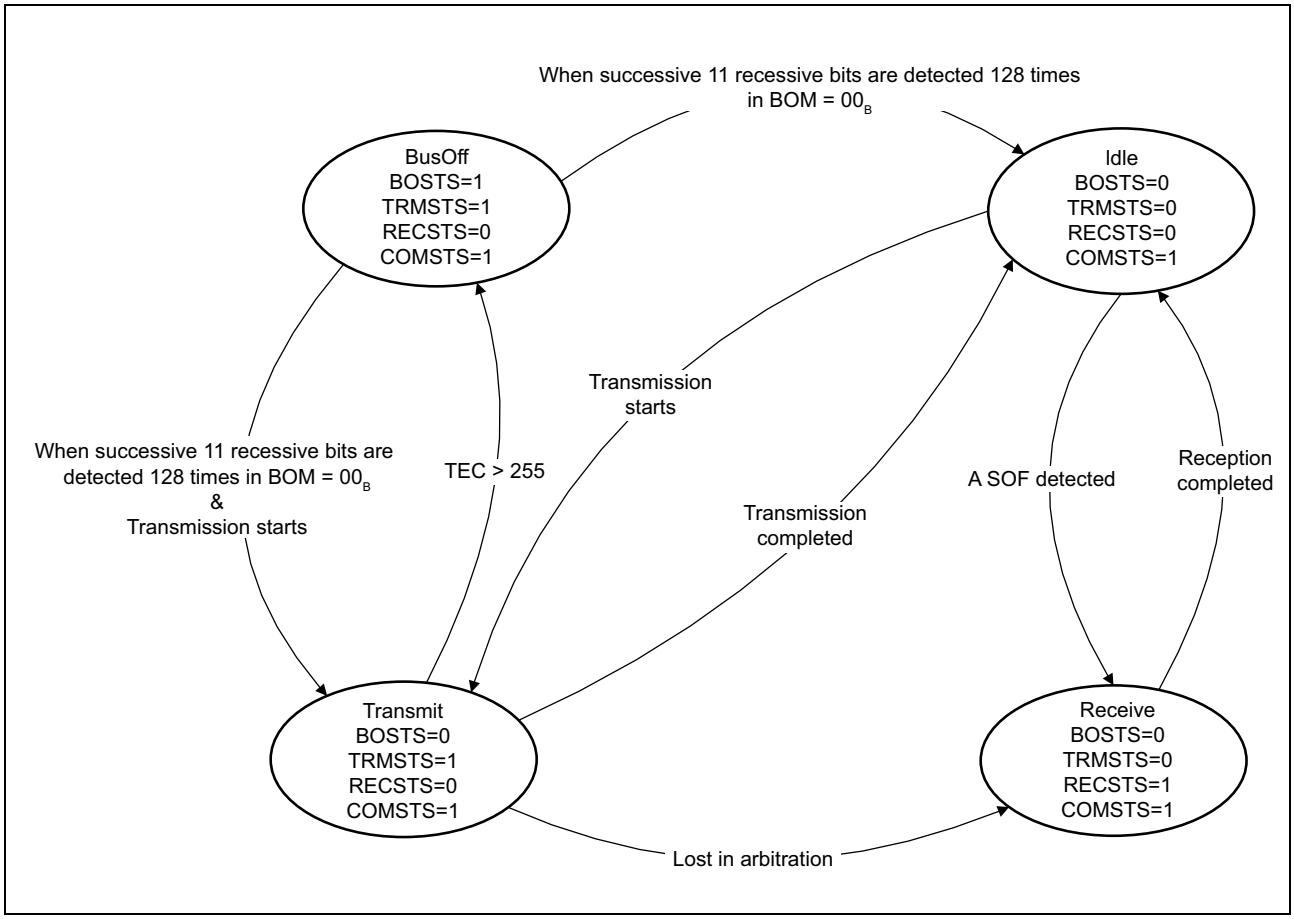


Figure 23.12 Sub-Modes of CAN Channel Communication Mode (only BOM = 00_B)

23.4.2.5 CAN channel Bus-Off State

The CAN channel Bus-Off state is entered according to the fault confinement rules of the CAN specification. Following modes for returning to the CAN channel Communication Mode from the Bus-Off state can be configured:

RSCFDnCFDCmCTR.BOM = 00_B:

Bus-Off recovery is compliant to ISO 11898-1 (2015), namely the CAN channel re-enters CAN communication (Error Active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF is set in this case.

RSCFDnCFDCmCTR.BOM = 01_B:

The CAN channel changes the value of the RSCFDnCFDCmCTR.CHMDC bits within the CAN channel Control Register to 10B and switches immediately to Channel Halt Mode automatically after entering the Bus-Off state. TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF is not set in this case.

RSCFDnCFDCmCTR.BOM = 10_B:

The CAN channel changes the value of the RSCFDnCFDCmCTR.CHMDC bits within the CAN channel Control Register to 10_B as soon as it reaches the Bus-Off state and enters Channel Halt Mode automatically after the CAN channel has completed the Bus-Off recovery sequence (i.e. after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to zero. The Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF is set in this case.

RSCFDnCFDCmCTR.BOM = 11_B:

Bus-Off recovery is initiated but CAN channel can enter immediately the Channel Halt Mode when still in Bus-Off state if a request is made to enter Channel Halt Mode.

TEC and REC counters are initialized to zero. In this case, the Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF is not set and TEC and REC counters are initialized to zero.

Without setting RSCFDnCFDCmCTR.CHMDC [1:0] =10_B and when 11recessive bits is detected 128 times continuously, transition conditions become the same as RSCFDnCFDCmCTR.BOM = 00_B.

Note, however, that if the recovery from Bus-Off occurs normally in this mode (i.e. after waiting for 128 sequences of 11 consecutive recessive bits), and no Halt request has been generated during this period, then the Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF is set.

In the case where Application SW writes into RSCFDnCFDCmCTR.CHMDC at the same time as the CAN channel is due to enter Halt Mode (at the start of Bus-Off when RSCFDnCFDCmCTR.BOM = 01_B, or at the end of Bus-Off when RSCFDnCFDCmCTR.BOM = 10_B) then the System SW request will have the highest priority.

Note that, in the above cases, the automatic setting of the RSCFDnCFDCmCTR.CHMDC to Channel Halt mode request is performed when the RSCFDnCFDCmCTR.CHMDC value is previously 00_B (Channel Communication Mode).

Additional it is possible to force the CAN channel to recover from the Bus-Off state by setting RSCFDnCFDCmCTR.RTBO to 1_B. The error state changes from Bus-Off state to integrating state

with a maximum delay of 1 CAN Bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case.

TEC and REC counters are initialized to zero.

Before setting RSCFDnCFDCmCTR.RTBO to 1_B, all pending transmissions from TX Message Buffers, TX Queues and/or Transmit/Receive FIFO in TX or GW Mode should be disabled.

The disable of the pending transmission Message Buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For TX Message Buffer these are the Transmission Result Flags (RSCFDnCFDTMSTSp.TMTRF) for the TX Queue it is the TX Queue empty flag (RSCFDnCFDTXQSTSm.TXQEMP) and for the FIFO it is the FIFO empty flag (RSCFDnCFDCFSTSk.CFEMP).

The RSCFDnCFDCmCTR.RTBO bit should be used for Bus-Off recovery only when RSCFDnCFDCmCTR.BOM is set to 00_B.

Setting this bit in any state (other than Bus-Off) will have no effect and the bit will be cleared immediately.

Table 23.137 summarises the setting of Bus-Off Entry Flag RSCFDnCFDCmERFL.BOEF and Bus-Off Recovery Flag RSCFDnCFDCmERFL.BORF for different configurations of RSCFDnCFDCmCTR.BOM.

Table 23.137 Bus-Off entry / recovery flag behavior

BOM	BOEF bit set	BORF bit set
00_B	Always (on entry to Bus-Off)	Always (on exit from Bus-Off)
00_B RSCFDnCFDCmCTR. RTBO set to '1'	Always (on entry to Bus-Off)	Only if normal Bus-Off recovery occurs before System SW sets RSCFDnCFDCmCTR.RTBO to '1'
01_B	Always (on entry to Bus-Off)	Never
10_B	Always (on entry to Bus-Off)	Always (on exit from Bus-Off)
11_B	Always (on entry to Bus-Off)	Only if normal Bus-Off recovery occurs before System SW issues Halt request

To make an efficient SW procedure it is not mandatory to wait for BusOff recovery sequence end.

It is possible to do the transmission re-initialization during the BusOff recovery. To do this following SW-flow is recommended to use, refer to **Figure 23.13**.

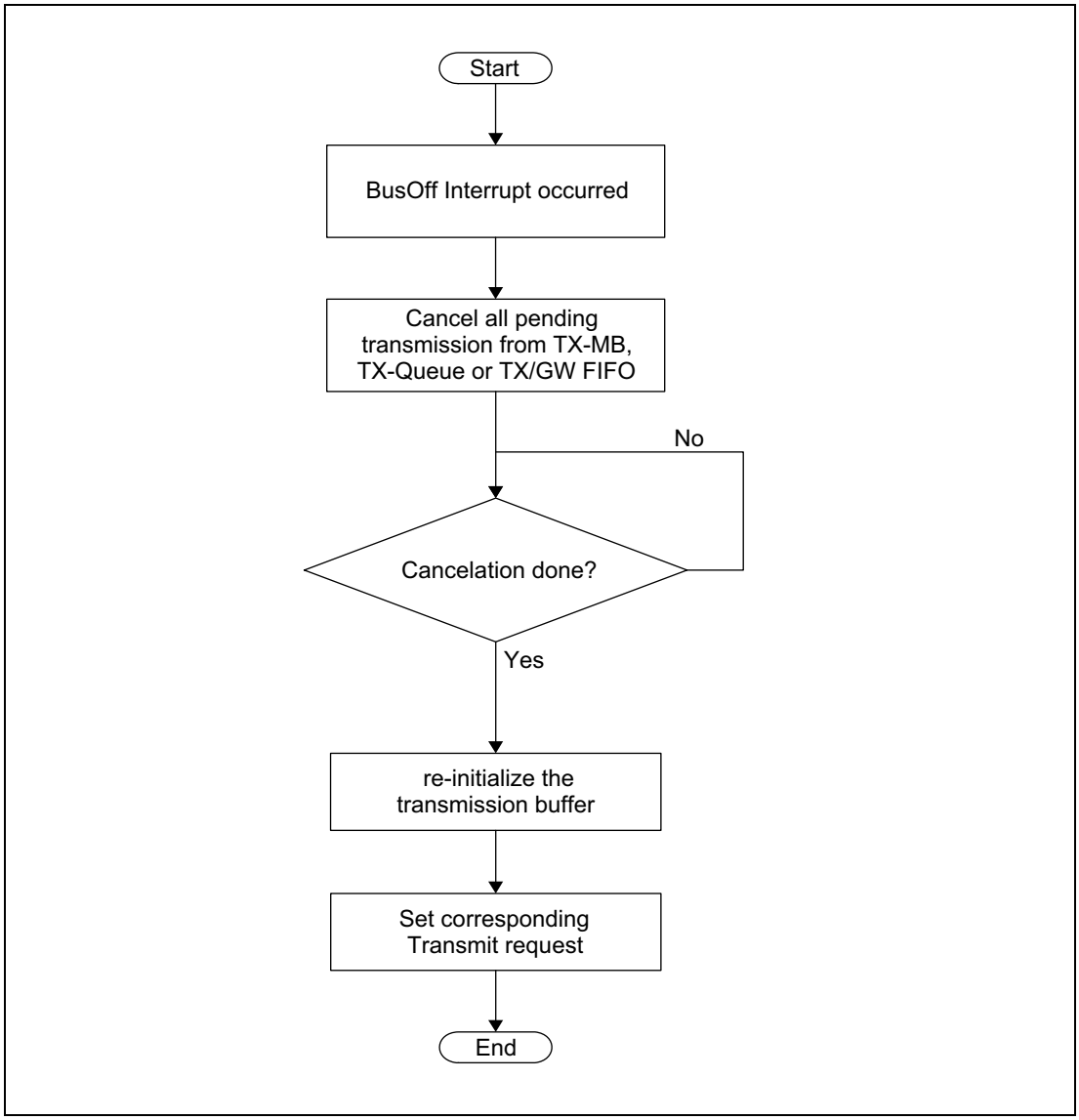


Figure 23.13 Transmission re-initialization during Bus-Off

23.4.3 Global Mode – Channel Mode transition interactions

In the following, the interaction between global mode setting and channel mode setting is summarized:

Changing of Channel Mode Control RSCFDnCFDCmCTR.CHMDC in the Channel Control Register does not have any influence on the Global Mode Control RSCFDnCFDGCTR.GMDC.

Changing of Global Mode Control RSCFDnCFDGCTR.GMDC is influencing the channel mode control in the way described in **Table 23.138**.

Table 23.138 Global – Channel mode transition interaction

Global Mode Change	Channel Mode	Channel Mode Transition Action
Stop → Reset	Stop	channel remains in Stop Mode
Stop → Halt	- (global mode change not possible)	
Stop → Communication	- (global mode change not possible)	
Reset → Stop	Stop	channels remains in Stop Mode
	Reset	channel Stop request bit is set automatically, channel transits to Stop Mode
Reset → Halt	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
Reset → Communication	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
Halt → Stop	- (global mode change not possible)	
Halt → Reset	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
	Halt	channel mode control is set to Reset Mode, channel transits to Reset Mode
Halt → Communication	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
	Halt	channel remains in Halt Mode
Communication → Stop	- (global mode change not possible)	
Communication → Reset	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
	Halt	channel mode control is set to Reset Mode, channel transits to Reset Mode
	Communication	channel mode control is set to Reset Mode, channel transits to Reset Mode
Communication → Halt	Stop	channel remains in Stop Mode
	Reset	channel remains in Reset Mode
	Halt	channel remains in Halt Mode
	Communication	channel mode control is set to Halt Mode, channel transits to Halt Mode after communication finished

23.4.3.1 Global Mode change timing

The transition time for the Global mode changes are shown below.

Table 23.139 Global Mode change timing

From	To	max. transition time
GL_STOP	GL_RESET	3 peripheral clock cycle*2
GL_RESET	GL_STOP	3 peripheral clock cycle
GL_RESET	GL_TEST	10 peripheral clock cycle
GL_RESET	GL_OPERATING	10 peripheral clock cycle
GL_TEST	GL_RESET	2 CAN bit times
GL_TEST	GL_OPERATING	3 peripheral clock cycle
GL_OPERATING	GL_RESET	2 CAN bit times
GL_OPERATING	GL_TEST	3 CAN frames*1, *3

Note 1. The given transition time is the time without any errors on the bus. In case of Error condition the transition time could lengthen to an uncalculated result. As well it could also come to stuck condition in case of locked RX lines or continues error conditions.

Note 2. Users should leave GL_STOP mode, only when RSCFDnCFDGSTS.GRAMINIT is cleared.

Note 3. TQ, CAN frame and CAN bits are related to the individual channels, for the max transition time the channel with the lowest Baud Rate has to be used.

23.4.3.2 Channel Mode change timing

The transition time for the Channel mode changes are shown below.

Table 23.140 Channel Mode change timing

From	To	Max. transition time
CH_STOP	CH_RESET	3 peripheral clock cycle
CH_RESET	CH_STOP	3 peripheral clock cycle
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_COMMUNICATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_COMMUNICATION	4 CAN bit times*3
CH_COMMUNICATION	CH_RESET	2 CAN bit times
CH_COMMUNICATION	CH_HALT	2 CAN frames*1, *2

Note 1. The time specified for this transition does not include the case where channel enters Bus-Off state. In case of Bus-Off the timing depends upon the configuration of the RSCFDnCFDCmCTR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of Error condition the transition time could lengthen to an uncalculated result. As well it could also come to a stuck condition in case of locked RX lines or continues error conditions.

Note 3. In general if the Baudrate prescaler value RSCFDnCFDCmNCFG.NBRP is changed in CH_HALT then the transition time could be deviate from above. As the internal prescaler is a free running down counter to create the TQ clock. And new BRP value will earliest be captured when the counter reached the value zero.

23.4.3.3 Register behavior in global/channel Modes

The following table shows the bit behavior when RS-CANFD module changes state.

Table 23.141 Register behavior in global/channel Modes (1/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
1. Channel Register													
RSCFDnCFDC mNCFG	m_chsfr	Channel m Configuration Register	[9:0]	NBRP	Nominal Baudrate Prescaler	00_H	00_H	unch	unch	unch	unch	unch	unch
			[16:10]	NSJW	Nominal Synchronization Jump Width	0000000_B	0000000_B	unch	unch	unch	unch	unch	unch
			[24:17]	NTSEG1	Nominal Time Segment 1	00000000_B	00000000_B	unch	unch	unch	unch	unch	unch
			[31:25]	NTSEG2	Nominal Time Segment 2	0000000_B	0000000_B	unch	unch	unch	unch	unch	unch
RSCFDnCFDC mDCFG	m_chsfr	Channel m Configuration Register	[7:0]	DBRP	Data Baudrate Prescaler	00_H	00_H	unch	unch	unch	unch	unch	unch
			[12:8]	DTSEG1	Data Time Segment 1	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[15:13]	–	reserved	–	–	–	–	–	–	–	–
			[19:16]	DTSEG2	Data Time Segment 2	0000_B	0000_B	unch	unch	unch	unch	unch	unch
			[23:20]	–	reserved	–	–	–	–	–	–	–	–
			[27:24]	DSJW	Data Synchronization Jump Width	0000_B	0000_B	unch	unch	unch	unch	unch	unch
			[31:28]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDC mCTR	m_chsfr	Channel m Control Register	[1:0]	CHMDC	Channel Mode Control	01_B	01_B	unch	01_B	unch if channel in Stop, Reset, Halt, otherwise bit will be set to 10_B	unch	01_B	10_B
			2	CSLPR	Channel Stop Request	1	1	1	unch	unch	unch	unch	unch
			3	RTBO	Return from Bus-Off	0	0	unch	unch	unch	unch	unch	unch
			[7:4]	–	reserved	–	–	–	–	–	–	–	–
			8	BEIE	Bus Error Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			9	EWIE	Error Warning Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			10	EPIE	Error Passive Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			11	BOEIE	Bus-Off Entry Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			12	BORIE	Bus-Off Recovery Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			13	OLIE	Overload Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			14	BLIE	Bus Lock Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			15	ALIE	Arbitration Lost Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			16	TAIE	Transmission Abortion Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			19	TDCVFIE	Transceiver Delay Compensation Bit FIFO Msg Lost Interrupt enable	0	0	unch	unch	unch	unch	unch	unch
			20	–	reserved	–	–	–	–	–	–	–	–
			[22:21]	BOM	Bus-Off Mode	00_B	00_B	unch	unch	unch	unch	unch	unch
			23	ERRD	Error Display	0	0	unch	unch	unch	unch	unch	unch
			24	CTME	Channel Test Mode Enable	0	0	unch	0	unch	unch	0	unch
			[26:25]	CTMS	Channel Test Mode Select	00_B	00_B	unch	00_B	unch	unch	00_B	unch
			[29:27]	–	reserved	–	–	–	–	–	–	–	–
			30	CRCT	CRC Test mode	0	0	unch	0	unch	unch	0	unch
			31	ROM	Restricted Operation Mode	0	0	unch	0	unch	unch	0	unch

Table 23.141 Register behavior in global/channel Modes (2/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDC mSTS	m_chsfr	Channel m Status Register	0	CRSTSTS	Channel RESET State	1	1	unch	1	unch	unch	1	0
			1	CHLTSTS	Channel TEST State	0	0	unch	0	unch if channel in Stop, Reset, Halt, otherwise bit will be set to 1	unch	0	1
			2	CSLPSTS	Channel Stop State	1	1	1	unch	unch	1	unch	unch
			3	EPSTS	Error Passive Status	0	0	unch	0	unch	unch	0	unch
			4	BOSTS	Bus-Off Status	0	0	unch	0	0	unch	0	0
			5	TRMSTS	Transmit Status	0	0	unch	0	0	unch	0	0
			6	RECSTS	Receive Status	0	0	unch	0	0	unch	0	0
			7	COMSTS	Communication Status	0	0	unch	0	0	unch	0	0
			8	ESIF	Error State Indication Flag	0	0	unch	0	unch	unch	0	unch
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			[23:16]	REC	Reception Error Count	00_H	00_H	unch	00_H	unch	unch	00_H	unch
			[31:24]	TEC	Transmission Error Count	00_H	00_H	unch	00_H	unch	unch	00_H	unch
RSCFDnCFDC mERFL	m_chsfr	Channel m Error Flag Register	0	BEF	Bus Error Flag	0	0	unch	0	unch	unch	0	unch
			1	EWF	Error Warning Flag	0	0	unch	0	unch	unch	0	unch
			2	EPF	Error Passive Flag	0	0	unch	0	unch	unch	0	unch
			3	BOEF	Bus-Off Entry Flag	0	0	unch	0	unch	unch	0	unch
			4	BORF	Bus-Off Recovery Flag	0	0	unch	0	unch	unch	0	unch
			5	OVLF	Overload Flag	0	0	unch	0	unch	unch	0	unch
			6	BLF	Bus Lock Flag	0	0	unch	0	unch	unch	0	unch
			7	ALF	Arbitration Lost Flag	0	0	unch	0	unch	unch	0	unch
			8	SERR	Stuff Error	0	0	unch	0	unch	unch	0	unch
			9	FERR	Form Error	0	0	unch	0	unch	unch	0	unch
			10	AERR	Ack Error	0	0	unch	0	unch	unch	0	unch
			11	CERR	CRC Error	0	0	unch	0	unch	unch	0	unch
			12	B1ERR	Bit 1 Error	0	0	unch	0	unch	unch	0	unch
			13	B0ERR	Bit 0 Error	0	0	unch	0	unch	unch	0	unch
			14	ADERR	Ack Del Error	0	0	unch	0	unch	unch	0	unch
			15	–	reserved	–	–	–	–	–	–	–	–
			[30:16]	CRCREG	CRC register value	0	0	unch	0	unch	unch	0	unch
			31	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDC mFDCFG	m_chsfr	Channel m CANFD Configuration Register	[2:0]	EOCCFG	Error Occurrence Counter Configuration	000_B	000_B	unch	unch	unch	unch	unch	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			8	TDCOC	Transceiver Delay Compensation Offset Configuration	0	0	unch	unch	unch	unch	unch	unch
			9	TDCE	Transceiver Delay Compensation Enable	0	0	unch	unch	unch	unch	unch	unch
			10	ESIC	Error State Indication Configuration	0	0	unch	unch	unch	unch	unch	unch
			[15:11]	–	reserved	–	–	–	–	–	–	–	–
			[23:16]	TDCO	Transceiver Delay Compensation Offset	0	0	unch	unch	unch	unch	unch	unch
			24	GWEN	CAN2.0 <> CANFD GW Enable	0	0	unch	unch	unch	unch	unch	unch
			25	GWFDF	GW FDF Configuration	0	0	unch	unch	unch	unch	unch	unch
			26	GWBRS	GW BRS Configuration	0	0	unch	unch	unch	unch	unch	unch
			27	–	reserved	–	–	–	–	–	–	–	–
			28	FDOE	CANFD only enable	0	0	unch	unch	unch	unch	unch	unch
			29	REFE	RX edge filter enable	0	0	unch	unch	unch	unch	unch	unch
			30	CLOE	Classical CAN only enable	0	0	unch	unch	unch	unch	unch	unch
			31	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDC mFDCTR	m_chsfr	Channel m CANFD Control Register	0	EOCCLR	Error Occurrence Counter Clear	0	0	unch	0	unch	unch	0	unch
			1	SOCCLR	Successful Occurrence Counter Clear	0	0	unch	0	unch	unch	0	unch
			[31:2]	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (3/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDCmFDSTS	m_chsfr	Channel m CANFD Status Register	[7:0]	TDCR	Transceiver Delay Compensation Result	0	0	unch	0	unch	unch	0	unch
			8	EOCO	Error occurrence counter overflow	0	0	unch	0	unch	unch	0	unch
			9	SOCO	Successful occurrence counter overflow	0	0	unch	0	unch	unch	0	unch
			[14:10]	–	reserved	–	–	–	–	–	–	–	–
			15	TDCVF	Transceiver Delay Compensation violation flag	0	0	unch	0	unch	unch	0	unch
			[23:16]	EOC	Error occurrence counter register	0	0	unch	0	unch	unch	0	unch
			[31:24]	SOC	Successful occurrence counter register	0	0	unch	0	unch	unch	0	unch
RSCFDnCFDCnFDCR C	m_chsfr	Channel m CANFD CRC Register	[20:0]	CRCREG	CRC Register value	0	0	unch	0	unch	unch	0	unch
			[23:21]	–	reserved	–	–	–	–	–	–	–	–
			[27:24]	SCNT	Stuff bit count	0	0	unch	0	unch	unch	0	unch
			[31:28]	–	reserved	–	–	–	–	–	–	–	–
2. Global Configuration / Control Register													
RSCFDnCFDGCFG	m_comsfr	Global Configuration Register	0	TPRI	Transmission Priority	0	0	unch	unch	unch	–	–	–
			1	DCE	DLC Check Enable	0	0	unch	unch	unch	–	–	–
			2	DRE	DLC Replacement Enable	0	0	unch	unch	unch	–	–	–
			3	MME	Mirror Mode Enable	0	0	unch	unch	unch	–	–	–
			4	DCS	PLL By-Pass	0	0	unch	unch	unch	–	–	–
			5	CMPOC	CANFD message Payload overflow configuration	0	0	unch	unch	unch	–	–	–
			[7:6]	–	reserved	–	–	–	–	–	–	–	–
			[11:8]	TSP	Timestamp Prescaler	0000_B	0000_B	unch	unch	unch	–	–	–
			12	TSSS	Timestamp Source Select	0	0	unch	unch	unch	–	–	–
			[15:13]	TSBTCS	Timestamp Bit Time Channel Select	000_B	000_B	unch	unch	unch	–	–	–
			[31:16]	ITRCP	Interval Timer Reference Clock Prescaler	0000_H	0000_H	unch	unch	unch	–	–	–
RSCFDnCFDGCTR	m_comsfr	Global Control Register	[1:0]	GMDC	Global Mode Control	01_B	01_B	unch	unch	unch	–	–	–
			2	GSLPR	Global Stop Request	1	1	unch	unch	unch	unch	unch	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			8	DEIE	DLC Error Interrupt Enable	0	0	unch	unch	unch	–	–	–
			9	MEIE	Message Lost Error Interrupt Enable	0	0	unch	unch	unch	–	–	–
			10	THLEIE	TX History List Entry Lost Interrupt Enable	0	0	unch	unch	unch	–	–	–
			11	CMPOFIE	CANFD message payload overflow Flag Interrupt enable	0	0	unch	unch	unch	–	–	–
			12	QOWEIE	TXQ Message overwrite Error Interrupt Enable	0	0	unch	unch	unch	–	–	–
			13	–	reserved	–	–	–	–	–	–	–	–
			14	QMEIE	TXQ Message lost Error Interrupt Enable	0	0	unch	unch	unch	–	–	–
			15	MOWEIE	Message overwrite Error Interrupt Enable	0	0	unch	unch	unch	–	–	–
			16	TSRST	TS Reset	0	0	unch	unch	unch	–	–	–
			[31:17]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGFDCFG	m_comsfr	Global FD configuration register	0	RPED	Protocol exception state disable	0	0	unch	unch	unch	–	–	–
			[7:1]	–	reserved	–	–	–	–	–	–	–	–
			[9:8]	TSCCFG	Time stamp capture configuration	0	0	unch	unch	unch	–	–	–
			[31:10]	–	reserved	–	–	–	–	–	–	–	–
3. Global Status Register													
RSCFDnCFDGSTS	m_comsfr	Global Status Register	0	GRSTSTS	Global RESET Status	1	1	unch	1	0	–	–	–
			1	GHLTSTS	Global TEST Status	0	0	unch	0	1	–	–	–
			2	GSLPSTS	Global Stop Status	1	1	1	unch	unch	–	–	–
			3	GRAMINIT	Global RAM Initialization Status	1	1	1→0	0	0	–	–	–
			[31:4]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGERFL	m_comsfr	Global Error Flag Register	0	DEF	DLC Error Flag	0	0	unch	0	unch	–	–	–
			1	MES	Message Lost Error Status	0	0	unch	0	unch	–	–	–
			2	THLES	TX History List Entry Lost Error Status	0	0	unch	0	unch	–	–	–
			3	CMPOF	CANFD message payload overflow Flag	0	0	unch	0	unch	–	–	–
			4	QOWES	TXQ Message overwrite Error Status	0	0	unch	0	unch	–	–	–
			5	–	reserved	–	–	–	–	–	–	–	–
			6	QMES	TXQ Message Lost Error Status	0	0	unch	0	unch	–	–	–
			7	MOWES	Message overwrite Error Status	0	0	unch	0	unch	–	–	–

Table 23.141 Register behavior in global/channel Modes (4/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
			[15:8]	–	reserved	–	–	–	–	–	–	–	–
			[23:16]	EEFm	ECC Error Flag for Channel m	0	0	unch	0	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGTSC	m_timestamp	Global Timestamp Counter Register	[15:0]	TS	Timestamp Value	0000_H	0000_H	unch	0000_H	unch	–	–	–
			[31:16]	–	reserved	–	–	–	–	–	–	–	–
4. Global Acceptance Filter List Configuration Register													
RSCFDnCFDGAFLECTR	m_acsfr	Global Acceptance Filter List Entry Control Register	[6:0]	AFLPN	Acceptance Filter List Page Number	0000000_B	0000000_B	unch	unch	unch	–	–	–
			7	–	reserved	–	–	–	–	–	–	–	–
			8	AFLDAE	Acceptance Filter List Access Enable Data	0	0	unch	unch	unch	–	–	–
			[31:9]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGAFLCFG 0	m_acsfr	Global Acceptance Filter List Configuration 0 Register	[8:0]	RNC1	Rule Number for Channel 1	000_H	000_H	unch	unch	unch	–	–	–
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			[24:16]	RNC0	Rule Number for Channel 0	000_H	000_H	unch	unch	unch	–	–	–
			[31:25]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGAFLCFG 1	m_acsfr	Global Acceptance Filter List Configuration 1 Register	[8:0]	RNC3	Rule Number for Channel 3	000_H	000_H	unch	unch	unch	–	–	–
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			[24:16]	RNC2	Rule Number for Channel 2	000_H	000_H	unch	unch	unch	–	–	–
			[31:25]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGAFLCFG 2	m_acsfr	Global Acceptance Filter List Configuration 2 Register	[8:0]	RNC5	Rule Number for Channel 5	000_H	000_H	unch	unch	unch	–	–	–
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			[24:16]	RNC4	Rule Number for Channel 4	000_H	000_H	unch	unch	unch	–	–	–
			[31:25]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGAFLCFG 3	m_acsfr	Global Acceptance Filter List Configuration 3 Register	[8:0]	RNC7	Rule Number for Channel 7	000_H	000_H	unch	unch	unch	–	–	–
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			[24:16]	RNC6	Rule Number for Channel 6	000_H	000_H	unch	unch	unch	–	–	–
			[31:25]	–	reserved	–	–	–	–	–	–	–	–
5. RX Mailbox Register													
RSCFDnCFDRMNB	m_acsfr	RX Mailbox Number Register	[7:0]	NRXMB	Number of RX MB	00_H	00_H	unch	unch	unch	–	–	–
			[10:8]	RMPLS	Reception Message Buffer Payload Data Size	000_B	000_B	unch	unch	unch	–	–	–
			[31:11]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDRMNDt	m_acsfr	RX-Mailbox NewData Register	[31:0]	RMNS	RX Mailboxes Newdata Flag	0000000_H	0000000_H	unch	00000000_H	unch	–	–	–
6. RX FIFO Register													
RSCFDnCFDRFCC	m_rxfifounit	RX FIFO Configuration / Control Register [7:0]	0	RFE	RX FIFO Enable	0	0	unch	0	unch	–	–	–
			1	RFIE	RX FIFO Interrupt Enable	0	0	unch	unch	unch	–	–	–
			[3:2]	–	reserved	–	–	–	–	–	–	–	–
			[6:4]	RFPLS	Rx FIFO Payload Data Size configuration	000_B	000_B	unch	unch	unch	–	–	–
			7	–	reserved	–	–	–	–	–	–	–	–
			[10:8]	RFDC	RX FIFO Depth Configuration	000_B	000_B	unch	unch	unch	–	–	–
			11	–	reserved	–	–	–	–	–	–	–	–
			12	RFIM	RX FIFO Interrupt Mode	0	0	unch	unch	unch	–	–	–
			[15:13]	RFIGCV	RX FIFO Interrupt Generation Counter Value	000_B	000_B	unch	unch	unch	–	–	–
			16	RFFIE	RX FIFO Full interrupt Enable	0	0	unch	unch	unch	–	–	–
			[31:17]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDRFSTS	m_rxfifounit	RX FIFO Status Register [7:0]	0	RFEMP	RX FIFO Empty	1	1	unch	1	unch	–	–	–
			1	RFFLL	RX FIFO Full	0	0	unch	0	unch	–	–	–
			2	RFMLT	RX FIFO Msg Lost Flag	0	0	unch	0	unch	–	–	–
			3	RFIF	RX FIFO Interrupt Flag	0	0	unch	0	unch	–	–	–
			[7:4]	–	reserved	–	–	–	–	–	–	–	–
			[15:8]	RFMC	RX FIFO Message Count	00_H	00_H	unch	00_H	unch	–	–	–
			16	RFFIF	RX FIFO Full Interrupt Flag	0	0	unch	0	unch	–	–	–
			[31:17]	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (5/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDRFPCTR	m_rxfifounit	RX FIFO Pointer Control Register[7:0]	[7:0]	RFPC	RX FIFO Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
7. Transmit/Receive FIFO Register													
RSCFDnCFDFCCk	m_comfifounit	Transmit/ Receive FIFO Configuration / Control Register [23:0]	0	CFE	Transmit/Receive FIFO Enable	0	0	unch	0	unch	unch	0: TX / GW FIFO unch: RX FIFO	unch
			1	CFRXIE	Transmit/Receive FIFO Interrupt Enable for RX Mode	0	0	unch	unch	unch	unch	unch	unch
			2	CFTXIE	Transmit/Receive FIFO Interrupt Enable for TX Mode	0	0	unch	unch	unch	unch	unch	unch
			3	–	reserved	–	–	–	–	–	–	–	–
			[6:4]	CFPLS	Transmit/Receive FIFO Payload Data Size configuration	000_B	000_B	unch	unch	unch	–	–	–
			7	–	reserved	–	–	–	–	–	–	–	–
			[9:8]	CFM	Transmit/Receive FIFO Mode	00_B	00_B	unch	unch	unch	unch	unch	unch
			10	CFITSS	Transmit/Receive FIFO Interval Timer Source Select	0	0	unch	unch	unch	unch	unch	unch
			11	CFITR	Transmit/Receive FIFO Interval Timer Reference Clock Resolution	0	0	unch	unch	unch	unch	unch	unch
			12	CFIM	Transmit/Receive FIFO Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			[15:13]	CFIGCV	Transmit/Receive FIFO Interrupt Generation Counter Value	000_B	000_B	unch	unch	unch	unch	unch	unch
			[20:16]	CFTML	Transmit/Receive FIFO TX-Mailbox Link	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[23:21]	CFDC	Transmit/Receive FIFO Depth Configuration	000_B	000_B	unch	unch	unch	unch	unch	unch
			[31:24]	CFITT	Transmit/Receive FIFO Interval Transmission Time	0	0	unch	unch	unch	unch	unch	unch
RSCFDnCFDFCCEk	m_comfifounit	Transmit/ Receive FIFO Configuration / Control Register2 [23:0]	0	CFFIE	Transmit/Receive FIFO Full interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			1	CFOFRXIE	Transmit/Receive FIFO One Frame Reception Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			2	CFOFTXIE	Transmit/Receive FIFO One Frame Transmission Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			8	CFMOWM	Transmit/Receive FIFO message overwrite mode	0	0	unch	unch	unch	unch	unch	unch
			[15:9]	–	reserved	–	–	–	–	–	–	–	–
			16	CFBME	Transmit/Receive FIFO Buffering Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:17]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDFSTSk	m_comfifounit	Transmit/ Receive FIFO status registers [23:0]	0	CFEMP	Transmit/Receive FIFO Empty	1	1	unch	1	unch	–	1: TX / GW FIFO unch: RX FIFO	–
			1	CFFLL	Transmit/Receive FIFO Full	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			2	CFMLT	Transmit/Receive FIFO Msg Lost	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			3	CFRXIF	Transmit/Receive RX FIFO Interrupt Flag	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			4	CFTXIF	Transmit/Receive TX FIFO Interrupt Flag	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			[7:5]	–	reserved	–	–	–	–	–	–	–	–
			[15:8]	CFMC	Transmit/Receive FIFO Message Count	00_H	00_H	unch	00_H	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			16	CFFIF	Transmit/Receive FIFO Full Interrupt Flag	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–

Table 23.141 Register behavior in global/channel Modes (6/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDCFSTSk	m_comfifoun it	Transmit/ Receive FIFO status registers [23:0]	17	CFOFRXIF	Transmit/Receive FIFO One Frame Reception	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			18	CFOFTXIF	Transmit/Receive FIFO One Frame Transmission	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			[23:19]	–	reserved	–	–	–	–	–	–	–	–
			24	CFMOW	Transmit/Receive FIFO message overwrite	0	0	unch	0	unch	–	0: TX / GW FIFO unch: RX FIFO	–
			[31:25]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCFPCTRk	m_comfifoun it	Transmit/ Receive FIFO Pointer Control Register [23:0]	[7:0]	CFPC	Transmit/Receive FIFO Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
8. FIFO Status Support Register													
RSCFDnCFDFESTS	m_acsfr	FIFO Empty Status Register	[7:0]	RFxEMP	RX FIFO Empty Status	ff_H	ff_H	unch	ff_H	unch	–	–	–
			[31:8]	CFkEMP	Transmit/Receive FIFO Empty Status	111 * n_B	111 * n_B	unch	111 * n_B	unch	–	mirror of Transmit/ Receive FIFO Status	–
RSCFDnCFDFSTS	m_acsfr	FIFO Full Status Register	[7:0]	RFxFLL	RX FIFO full Status	00000000_H	00000000_H	unch	00_H	unch	–	–	–
			[31:8]	CFkFLL	Transmit/Receive FIFO Full Status			unch	000000_H	unch	–	mirror of Transmit/ Receive FIFO Status	–
RSCFDnCFDFFSTS	m_acsfr	FIFO FDC level Full Status Register	[7:0]	RFxFFLL	RX FIFO FDC level full Status	00000000_H	00000000_H	unch	00_H	unch	–	–	–
			[31:8]	CFxFFLL	Transmit/Receive FIFO Full Status			unch	000000_H	unch	–	mirror of Transmit/ Receive FIFO Status	–
RSCFDnCFDFMSTS	m_acsfr	FIFO Msg Lost Status Register	[7:0]	RFxMLT	RX FIFO Msg Lost Status	00000000_H	00000000_H	unch	00_H	unch	–	–	–
			[31:8]	CFkMLT	Transmit/Receive FIFO Msg Lost Status			unch	000000_H	unch	–	–	–
RSCFDnCFDCFMOWSTS	m_acsfr	Transmit/ Receive FIFO Message OverWrite Status	[23:0]	CFkMOW	Transmit/Receive FIFO [x] Massage overwrite status	000000_H	000000_H	unch	000000_H	unch	–	–	–
			[31:24]	–	reserved			–	–	–	–	–	–
RSCFDnCFDRFISTS	m_acsfr	Receive FIFO Interrupt Flag Status Register	[7:0]	RFxIF	RX FIFO Interrupt Flag Status	00_H	00_H	unch	00_H	unch	–	–	–
			[15:8]	–	reserved	–	–	–	–	–	–	–	–
			[23:16]	RFxFFLL	RX FIFO Full Interrupt Flag Status	8'h00	8'h00	unch	8'h00	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCFRISTS	m_acsfr	Transmit/ Receive FIFO Receive Interrupt Flag Status Register	[23:0]	CFkRXIF	Transmit/Receive FIFO RX Interrupt Flag Status	000000_H	000000_H	unch	000000_H	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCFTISTS	m_acsfr	Transmit/ Receive FIFO Transmit Interrupt Flag Status Register	[23:0]	CFkTXIF	Transmit/Receive FIFO TX Interrupt Flag Status	000000_H	000000_H	unch	000000_H	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCFOFRIS TS	m_acsfr	Transmit/ Receive FIFO One Frame RX Interrupt Flag Status	[23:0]	CFxOFRXIF	Transmit/Receive FIFO [x] One Frame RX Interrupt Flag Status	000000_H	000000_H	unch	000000_H	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCFOFTIS TS	m_acsfr	Transmit/ Receive FIFO One Frame TX Interrupt Flag Status	[23:0]	CFxOFTXIF	Transmit/Receive FIFO [x] One Frame TX Interrupt Flag Status	000000_H	000000_H	unch	000000_H	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
9. Special CANFD register													
RSCFDnCFDDTCT	m_dmaif	DMA Transfer Control Register	[7:0]	RFDMAEx	DMA Transfer Enable for RXFIFO x	0000_H	0000_H	unch	00_H	unch	unch	unch	unch
			[15:8]	CFDMAEx	DMA Transfer Enable for Transmit/Receive FIFO x			unch	00_H	unch	unch	unch	unch
			[31:16]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDDTSTS	m_dmaif	DMA Transfer Status Register	[7:0]	RFDMASTSx	DMA Transfer Status for RX FIFO x	0000_H	0000_H	unch	00_H	unch	–	–	–
			[15:8]	CFDMASTSx	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel x			unch	00_H	unch	–	–	–
			[31:16]	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (7/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDCDTTCT	m_dmaif	DMA TX Transfer Control Register	[7:0]	TQ0DMAEx	DMA TX Transfer Enable for TXQ0 x	00_H	00_H	unch	00_H	unch	unch	unch	unch
			[15:8]	TQ3DMAEx	DMA TX Transfer Enable for TXQ3 x	00_H	00_H	unch	00_H	unch	unch	unch	unch
			[23:16]	CFDMAEx	DMA Transfer Enable for Transmit/Receive FIFO x	00_H	00_H	unch	00_H	unch	unch	unch	unch
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDCDTTSTS	m_dmaif	DMA TX Transfer Status Register	[7:0]	TQ0DMASTSx	DMA Transfer Status for TXQ0 x	00_H	00_H	unch	00_H	unch	–	–	–
			[15:8]	TQ3DMASTSx	DMA Transfer Status for TXQ3 x	00_H	00_H	unch	00_H	unch	–	–	–
			[23:16]	CFDMASTSx	DMA Transfer Status only for Transmit/Receive FIFO 0 of channel x	00_H	00_H	unch	00_H	unch	–	–	–
			[31:24]	–	reserved	–	–	–	–	–	–	–	–
10. TX Mailbox Register													
RSCFDnCFDTMCp	m_mbctrl	TX Mailbox Control Register 64*n	0	TMTR	TX Mailbox Transmission Request	0	0	unch	0	unch	unch	0	unch
			1	TMTAR	TX Mailbox Transmission Abortion Request	0	0	unch	0	unch	unch	0	unch
			2	TMOM	TX Mailbox One-shot mode	0	0	unch	0	unch	unch	0	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTMSTS	m_mbctrl	TX Mailbox Status Register 64*n	0	TMTSTS	TX Mailbox Transmission Status	0	0	unch	0	0	unch	0	0
			[2:1]	TMTRF	TX Mailbox Transmission Result Flag	00_B	00_B	unch	00_B	unch	unch	00_B	unch
			3	TMTRM	TX Mailbox Transmission Request	0	0	unch	0	unch	unch	0	unch
			4	TMTARM	TX Mailbox Transmission Abortion Request	0	0	unch	0	unch	unch	0	unch
			[7:5]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTMTRSTSy	m_mbctrl	TX Mailboxes Transmission Request Status Register	[31:0]	CFDTMTRSTS	TX Mailbox Transmission Request Status	00000000_H	00000000_H	unch	00000000_H	unch	unch	will be cleared partly by channel reset	unch
RSCFDnCFDTMTARSTSy	m_mbctrl	TX Mailboxes Transmission Abortion Request Status Register	[31:0]	CFDTMTARSTS	TX Mailbox Transmission Abortion Request Status	00000000_H	00000000_H	unch	00000000_H	unch	unch	will be cleared partly by channel reset	unch
RSCFDnCFDTMTCSTSy	m_mbctrl	TX Mailboxes Transmission Completion Status Register	[31:0]	CFDTMTCSTS	TX Mailbox Transmission Completion Status	00000000_H	00000000_H	unch	00000000_H	unch	unch	will be cleared partly by channel reset	unch
RSCFDnCFDTMTASTSy	m_mbctrl	TX Mailboxes Transmission Abortion Status Register	[31:0]	CFDTMTASTS	TX Mailbox Transmission Abortion Status	00000000_H	00000000_H	unch	00000000_H	unch	unch	will be cleared partly by channel reset	unch
RSCFDnCFDTMIECy	m_mbctrl	TX Mailboxes Interrupt Enable Configuration Register	[31:0]	TMIE	TX Mailbox Interrupt Enable	00000000_H	00000000_H	unch	unch	unch	unch	unch	unch
11. TX Queue Register													
RSCFDnCFDTXQCC0m	m_mbctrl	TX Queue0 Configuration / Control Register n	0	TXQE	TX Queue Enable	0	0	unch	0	unch	unch	0	unch
			1	TXQGWE	TX Queue Gateway Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			2	TXQOWE	TX Queue Overwrite Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			[4:3]	–	reserved	–	–	–	–	–	–	–	–
			5	TXQTXIE	TX Queue TX Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			6	–	reserved	–	–	–	–	–	–	–	–
			7	TXQIM	TX Queue Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			[12:8]	TXQDC	TX Queue Depth Configuration	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[15:13]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIE	TXQ Full interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:19]	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (8/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDTXQCC1m	m_mbctrl	TX Queue1 Configuration / Control Register n	0	TXQE	TX Queue Enable	0	0	unch	0	unch	unch	0	unch
			1	TXQGWE	TX Queue Gateway Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			2	TXQOWE	TX Queue Overwrite Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			[4:3]	–	reserved	–	–	–	–	–	–	–	–
			5	TXQTXIE	TX Queue TX Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			6	–	reserved	–	–	–	–	–	–	–	–
			7	TXQIM	TX Queue Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			[12:8]	TXQDC	TX Queue Depth Configuration	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[15:13]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIE	TXQ Full interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:19]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQCC2m	m_mbctrl	TX Queue2 Configuration / Control Register n	0	TXQE	TX Queue Enable	0	0	unch	0	unch	unch	0	unch
			1	TXQGWE	TX Queue Gateway Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			2	TXQOWE	TX Queue Overwrite Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			[4:3]	–	reserved	–	–	–	–	–	–	–	–
			5	TXQTXIE	TX Queue TX Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			6	–	reserved	–	–	–	–	–	–	–	–
			7	TXQIM	TX Queue Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			[12:8]	TXQDC	TX Queue Depth Configuration	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[15:13]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIE	TXQ Full interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:19]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQCC3m	m_mbctrl	TX Queue3 Configuration / Control Register n	0	TXQE	TX Queue Enable	0	0	unch	0	unch	unch	0	unch
			1	–	reserved	–	–	–	–	–	–	–	–
			2	TXQOWE	TX Queue Overwrite Mode Enable	0	0	unch	unch	unch	unch	unch	unch
			[4:3]	–	reserved	–	–	–	–	–	–	–	–
			5	TXQTXIE	TX Queue TX Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			6	–	reserved	–	–	–	–	–	–	–	–
			7	TXQIM	TX Queue Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			[12:8]	TXQDC	TX Queue Depth Configuration	00000_B	00000_B	unch	unch	unch	unch	unch	unch
			[17:13]	–	reserved	–	–	–	–	–	–	–	–
			18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:19]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQSTS0m	m_mbctrl	TX Queue0 Status Register n	0	TXQEMP	TX Queue Empty	1	1	unch	1	unch	unch	1	unch
			1	TXQFLL	TX Queue Full	0	0	unch	0	unch	unch	0	unch
			2	TXQTXIF	TX Queue TX Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			[13:8]	TXQMC	CHn TX Queue Message counter	00_H	00_H	unch	00_H	unch	unch	00_H	unch
			[15:14]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIF	TXQ Full Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			19	TXQMLT	TXQ Message Lost	0	0	unch	0	unch	unch	0	unch
			20	TXQMOW	TXQ message overwrite	0	0	unch	0	unch	unch	0	unch
			[31:21]	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (9/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDTXQSTS1m	m_mbctrl	TX Queue1 Status Register n	0	TXQEMP	TX Queue Empty	1	1	unch	1	unch	unch	1	unch
			1	TXQFLL	TX Queue Full	0	0	unch	0	unch	unch	0	unch
			2	TXQTXIF	TX Queue TX Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			[13:8]	TXQMC	CHn TX Queue Message counter	00_H	00_H	unch	00_H	unch	unch	00_H	unch
			[15:14]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIF	TXQ Full Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			19	TXQMLT	TXQ Message Lost	0	0	unch	0	unch	unch	0	unch
			20	TXQMOW	TXQ message overwrite	0	0	unch	0	unch	unch	0	unch
			[31:21]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQSTS2m	m_mbctrl	TX Queue2 Status Register n	0	TXQEMP	TX Queue Empty	1	1	unch	1	unch	unch	1	unch
			1	TXQFLL	TX Queue Full	0	0	unch	0	unch	unch	0	unch
			2	TXQTXIF	TX Queue TX Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			[13:8]	TXQMC	CHn TX Queue Message counter	00_H	00_H	unch	00_H	unch	unch	00_H	unch
			[15:14]	–	reserved	–	–	–	–	–	–	–	–
			16	TXQFIF	TXQ Full Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			19	TXQMLT	TXQ Message Lost	0	0	unch	0	unch	unch	0	unch
			20	TXQMOW	TXQ message overwrite	0	0	unch	0	unch	unch	0	unch
			[31:21]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQSTS3m	m_mbctrl	TX Queue3 Status Register n	0	TXQEMP	TX Queue Empty	1	1	unch	1	unch	unch	1	unch
			1	TXQFLL	TX Queue Full	0	0	unch	0	unch	unch	0	unch
			2	TXQTXIF	TX Queue TX Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			[7:3]	–	reserved	–	–	–	–	–	–	–	–
			[13:8]	TXQMC	CHn TX Queue Message counter	00_H	00_H	unch	00_H	unch	unch	00_H	unch
			[17:14]	–	reserved	–	–	–	–	–	–	–	–
			18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			19	–	reserved	–	–	–	–	–	–	–	–
			20	TXQMOW	TXQ message overwrite	0	0	unch	0	unch	unch	0	unch
			[31:21]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQPCTR0m	m_mbctrl	TX Queue0 Pointer Control Register n	[7:0]	TXQPC	Channel m TX Queue Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQPCTR1m	m_mbctrl	TX Queue1 Pointer Control Register n	[7:0]	TXQPC	Channel m TX Queue Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQPCTR2m	m_mbctrl	TX Queue2 Pointer Control Register n	[7:0]	TXQPC	Channel m TX Queue Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQPCTR3m	m_mbctrl	TX Queue3 Pointer Control Register n	[7:0]	TXQPC	Channel m TX Queue Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQESTS	m_mbctrl	TX Queue Empty Status Register	[31:0]	TXQEMP	TXQ empty Status	$FFFFFFFF_H$	$FFFFFFFF_H$	–	–	–	unch	$FFFFFFFF_H$	unch
RSCFDnCFDTXQFISTS	m_mbctrl	TX Queue Full Interrupt Status Register	[2:0]	TXQFULL [2:0]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			3	–	reserved	–	–	–	–	–	–	–	–
			[6:4]	TXQFULL [6:4]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			7	–	reserved	–	–	–	–	–	–	–	–
			[10:8]	TXQFULL [10:8]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			11	–	reserved	–	–	–	–	–	–	–	–
			[14:12]	TXQFULL [14:12]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch

Table 23.141 Register behavior in global/channel Modes (10/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDTXQFISTS	m_mbctrl	TX Queue Full Interrupt Status Register	15	–	reserved	–	–	–	–	–	–	–	–
			[18:16]	TXQFULL [18:16]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			19	–	reserved	–	–	–	–	–	–	–	–
			[22:20]	TXQFULL [22:20]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			23	–	reserved	–	–	–	–	–	–	–	–
			[26:24]	TXQFULL [26:24]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			27	–	reserved	–	–	–	–	–	–	–	–
			[30:28]	TXQFULL [30:28]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			31	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQMSTS	m_mbctrl	TX Queue Message lost Status Register	[2:0]	TXQML[2:0]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			3	–	reserved	–	–	–	–	–	–	–	–
			[6:4]	TXQML[6:4]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			7	–	reserved	–	–	–	–	–	–	–	–
			[10:8]	TXQML [10:8]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			11	–	reserved	–	–	–	–	–	–	–	–
			[14:12]	TXQML [14:12]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			15	–	reserved	–	–	–	–	–	–	–	–
			[18:16]	TXQML [18:16]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			19	–	reserved	–	–	–	–	–	–	–	–
			[22:20]	TXQML [22:20]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			23	–	reserved	–	–	–	–	–	–	–	–
			[26:24]	TXQML [26:24]	TXQ message lost Status	000_B	000_B	–	–	–	unch	000_B	unch
			27	–	reserved	–	–	–	–	–	–	–	–
			[30:28]	TXQML [30:28]	TXQ Full Interrupt Status	000_B	000_B	–	–	–	unch	000_B	unch
			31	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQOWS TS	m_mbctrl	TX Queue Message Overwrite Status Register	[31:0]	TXQOW	TXQ message overwrite Status	0000000_{0H}	0000000_{0H}	–	–	–	unch	0000000_{0H}	unch
RSCFDnCFDTXQISTS	m_mbctrl	TX Queue Interrupt Status Register	[31:0]	TXQISF	TXQ Interrupt Status Flag	0000000_{0H}	0000000_{0H}	–	–	–	unch	0000000_{0H}	unch
RSCFDnCFDTXQOFTISTS	m_mbctrl	TX Queue One Frame TX Interrupt Status Register	[31:0]	TXQOFTISF	TXQ One Frame Tx Interrupt Status Flag	0000000_{0H}	0000000_{0H}	–	–	–	unch	0000000_{0H}	unch
RSCFDnCFDTXQOFRISTS	m_mbctrl	TX Queue One Frame RX Interrupt Status Register	[2:0]	TXQOFRISF [2:0]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			3	–	reserved	–	–	–	–	–	–	–	–
			[6:4]	TXQOFRISF [6:4]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			7	–	reserved	–	–	–	–	–	–	–	–
			[10:8]	TXQOFRISF [10:8]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			11	–	reserved	–	–	–	–	–	–	–	–
			[14:12]	TXQOFRISF [14:12]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			15	–	reserved	–	–	–	–	–	–	–	–
			[18:16]	TXQOFRISF [18:16]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			19	–	reserved	–	–	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (11/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDTXQOFRISTS	m_mbctrl	TX Queue One Frame RX Interrupt Status Register	[22:20]	TXQOFRISF [22:20]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			23	–	reserved	–	–	–	–	–	–	–	–
			[26:24]	TXQOFRISF [26:24]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			27	–	reserved	–	–	–	–	–	–	–	–
			[30:28]	TXQOFRISF [30:28]	TXQ One Frame RX Interrupt Status Flag	000_B	000_B	–	–	–	unch	000_B	unch
			31	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTXQFSTS	m_mbctrl	TX Queue Full Status Register	[31:0]	TXQFSF	TXQ Full Status	0000000_H	0000000_H	–	–	–	unch	0000000_H	unch
12. TX HistoryList Register													
RSCFDnCFDTHLCCm	m_thlfifounit	Channel m TX History List Configuration / Control Register	0	THLE	TX History List Enable	0	0	unch	0	unch	unch	0	unch
			[7:1]	–	reserved	–	–	–	–	–	–	–	–
			8	THLIE	TX History List Interrupt Enable	0	0	unch	unch	unch	unch	unch	unch
			9	THLIM	TX History List Interrupt Mode	0	0	unch	unch	unch	unch	unch	unch
			10	THLDTE	TX History List Dedicated TX Enable	0	0	unch	unch	unch	unch	unch	unch
			11	THLDGE	TX History List Dedicated GW Enable	0	0	unch	unch	unch	unch	unch	unch
			[31:12]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTHLSTSm	m_thlfifounit	Channel m TX HistoryList Status Register	0	THLEMP	TX History List Empty	1	1	unch	1	unch	unch	1	unch
			1	THLFLL	TX History List Full	0	0	unch	0	unch	unch	0	unch
			2	THLELT	TX History List Entry Lost Flag	0	0	unch	0	unch	unch	0	unch
			3	THLIF	TX History List Interrupt Flag	0	0	unch	0	unch	unch	0	unch
			[7:4]	–	reserved	–	–	–	–	–	–	–	–
			[13:8]	THLMC	TX History List Message Count	00_H	00_H	unch	00_H	unch	–	00_H	–
			[31:14]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDTHLPCTR	m_thlfifounit	CHn TX History List pointer control register	[7:0]	THLPC	TX History List Pointer Control	–	–	–	–	–	–	–	–
			[31:8]	–	reserved	–	–	–	–	–	–	–	–
13. TX Interrupt Status Register													

Table 23.141 Register behavior in global/channel Modes (12/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDGTINTSTSs	m_interrupt	Global Interrupt Status Register s	0	TSIF0	TX Successful Transmission Interrupt Flag Channel 0	0	0	unch	0	unch	unch	0	unch
			1	TAIF0	TX Abortion Interrupt Flag Channel 0	0	0	unch	0	unch	unch	0	unch
			2	TQIF0	TX Queue Interrupt Flag Channel 0	0	0	unch	0	unch	unch	0	unch
			3	CFTIF0	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel 0	0	0	unch	0	unch	unch	0	unch
			4	THIF0	TX History List Interrupt Channel 0	0	0	unch	0	unch	unch	0	unch
			5	TQOFIF0	TX Queue One Frame Transmission Interrupt Flag Channel 0	0	0	unch	0	unch	unch	0	unch
			6	CFOTIF0	Transmit/Receive FIFO One Frame Transmission Interrupt Channel 0	0	0	unch	0	unch	unch	0	unch
			7	–	reserved	–	–	–	–	–	–	–	–
			8	TSIF1	TX Successful Transmission Interrupt Flag Channel 1	0	0	unch	0	unch	unch	0	unch
			9	TAIF1	TX Abortion Interrupt Flag Channel 1	0	0	unch	0	unch	unch	0	unch
			10	TQIF1	TX Queue Interrupt Flag Channel 1	0	0	unch	0	unch	unch	0	unch
			11	CFTIF1	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel 1	0	0	unch	0	unch	unch	0	unch
			12	THIF1	TX History List Interrupt Channel 1	0	0	unch	0	unch	unch	0	unch
			13	TQOFIF1	TX Queue One Frame Transmission Interrupt Flag Channel 1	0	0	unch	0	unch	unch	0	unch
			14	CFOTIF1	Transmit/Receive FIFO One Frame Transmission Interrupt Channel 1	0	0	unch	0	unch	unch	0	unch
			15	–	reserved	–	–	–	–	–	–	–	–
			16	TSIF2	TX Successful Transmission Interrupt Flag Channel 2	0	0	unch	0	unch	unch	0	unch
			17	TAIF2	TX Abortion Interrupt Flag Channel 2	0	0	unch	0	unch	unch	0	unch
			18	TQIF2	TX Queue Interrupt Flag Channel 2	0	0	unch	0	unch	unch	0	unch
			19	CFTIF2	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel 2	0	0	unch	0	unch	unch	0	unch
			20	THIF2	TX History List Interrupt Channel 2	0	0	unch	0	unch	unch	0	unch
			21	TQOFIF2	TX Queue One Frame Transmission Interrupt Flag Channel 2	0	0	unch	0	unch	unch	0	unch
			22	CFOTIF2	Transmit/Receive FIFO One Frame Transmission Interrupt Channel 2	0	0	unch	0	unch	unch	0	unch
			23	–	reserved	–	–	–	–	–	–	–	–
			24	TSIF3	TX Successful Transmission Interrupt Flag Channel 3	0	0	unch	0	unch	unch	0	unch
			25	TAIF3	TX Abortion Interrupt Flag Channel 3	0	0	unch	0	unch	unch	0	unch
			26	TQIF3	TX Queue Interrupt Flag Channel 3	0	0	unch	0	unch	unch	0	unch
			27	CFTIF3	Transmit/Receive FIFO TX/GW Mode Interrupt Flag Channel 3	0	0	unch	0	unch	unch	0	unch
			28	THIF3	TX History List Interrupt Channel 3	0	0	unch	0	unch	unch	0	unch
			29	TQOFIF3	TX Queue One Frame Transmission Interrupt Flag Channel 3	0	0	unch	0	unch	unch	0	unch
			30	CFOTIF3	Transmit/Receive FIFO One Frame Transmission Interrupt Channel 3	0	0	unch	0	unch	unch	0	unch
			31	–	reserved	–	–	–	–	–	–	–	–
14. Global Test/Evaluation/Diag Control Register													
RSCFDnCFDGTSTCFG	m_test	Global Test Configuration Register	[7:0]	CmICBCE	Channel m Internal CAN Bus Communication Test Mode Enable	0000000_{0_B}	0000000_{0_B}	unch	00000000_B	unch	–	–	–
			[15:8]	–	reserved	–	–	–	–	–	–	–	–
			[25:16]	RTMPS	RAM Test Mode Page Select	0000000_{00_B}	0000000_{00_B}	unch	0000000000_B	unch	–	–	–
			[31:26]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGTSTCTR	m_test	Global Test Control Register	0	ICBCTME	Internal CAN Bus Communication Test Mode Enable	0	0	unch	0	unch	–	–	–
			1	–	reserved	–	–	–	–	–	–	–	–
			2	RTME	RAM Test Mode Enable	0	0	unch	0	unch	–	–	–
			[31:3]	–	reserved	–	–	–	–	–	–	–	–
RSCFDnCFDGLOCKK	m_test	Global Lock Key Register	[15:0]	LOCK	Lock Key	–	–	–	–	–	–	–	–
			[31:16]	–	reserved	–	–	–	–	–	–	–	–
15. Bus load counter Register													
RSCFDnCFDCmBLSTS	m_busload	BUS Load counter Register	[2:0]	–	reserved	–	–	–	–	–	–	–	–
			[31:3]	BLC	BUS load counter status	0000000_{0_H}	0000000_{0_H}	–	–	–	–	–	–

Table 23.141 Register behavior in global/channel Modes (13/13)

RS-CANFD SFR feature List						Initial Value (after transition, in the state -> initialized value is kept in the state)							
	Module	Register name	Bit	Symbol	Description	MCU_ RESET	Software - RESET	G_CAN_ STOP	G_CAN_ RESET	G_CAN_ TEST	CH_CAN _STOP	CH_CAN _RESET	CH_CAN _HALT
RSCFDnCFDCmBLCT	m_busload	BUS Load counter control Register	0	BLCE	BUS load counter enable	0	0	unch	0	–	unch	0	–
			[7 :1]	–	reserved	–	–	–	–	–	–	–	–
			8	BLCLD	BUS load counter load	–	–	–	–	–	–	–	–
			[31:9]	–	reserved	–	–	–	–	–	–	–	–
16. Flexible CAN mode Register													
RSCFDnCFDGFCMC	m_comsfr	Global Flexible CAN mode Configuration Register	0	FLXC0	Flexible CAN mode between Channel 0 and Channel 1	0	0	unch	unch	unch	–	–	–
			1	FLXC1	Flexible CAN mode between Channel 2 and Channel 3	0	0	unch	unch	unch	–	–	–
			2	FLXC2	Flexible CAN mode between Channel 4 and Channel 5	0	0	unch	unch	unch	–	–	–
			3	FLXC3	Flexible CAN mode between Channel 6 and Channel 7	0	0	unch	unch	unch	–	–	–
			[31:4]	–	reserved	–	–	–	–	–	–	–	–
17. Flexible transmission buffer assignment													
RSCFDnCFDGFTBAC	m_bus_if	Global Flexible transmission buffer assignment Configuration Register	[3:0]	FLXMB0	Flexible transmission buffer assignment bitween Channel 0 and Channel 1	0	0	unch	unch	unch	–	–	–
			[7:4]	–	reserved	–	–	–	–	–	–	–	–
			[11:8]	FLXMB1	Flexible transmission buffer assignment bitween Channel 2 and Channel 3	0	0	unch	unch	unch	–	–	–
			[15:12]	–	reserved	–	–	–	–	–	–	–	–
			[19:16]	FLXMB2	Flexible transmission buffer assignment bitween Channel 4 and Channel 5	0	0	unch	unch	unch	–	–	–
			[23:20]	–	reserved	–	–	–	–	–	–	–	–
			[27:24]	FLXMB3	Flexible transmission buffer assignment bitween Channel 6 and Channel 7	0	0	unch	unch	unch	–	–	–
			[31:28]	–	reserved	–	–	–	–	–	–	–	–
18. Reset Control Register													
RSCFDnCFDGRSTC	m_bus_if	Global SW reset Register	0	SRST	SW reset	0	unch	unch	unch	unch	–	–	–
			[7:1]	–	reserved	–	–	–	–	–	–	–	–
			[15:8]	KEY	Key code	–	–	–	–	–	–	–	–
			[31:16]	–	reserved	–	–	–	–	–	–	–	–
19. RX Interrupt Status Register (Transmit/Receive FIFO RX-GW mode & TXQ GW mode)													
RSCFDnCFDGRINTSTSm	m_interrupt	Global RX Interrupt Status Register m	[2:0]	QFIF	TXQ Full Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[3]	–	reserved	–	–	–	–	–	–	–	–
			[5:4]	BQFIF	Borrowed TXQ Full Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[7:6]	–	reserved	–	–	–	–	–	–	–	–
			[10:8]	QOFRIF	TXQ One Frame RX Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[11]	–	reserved	–	–	–	–	–	–	–	–
			[13:12]	BQOFRIF	Borrowed TXQ One Frame RX Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[15:14]	–	reserved	–	–	–	–	–	–	–	–
			[18:16]	CFRIF	Transmit/Receive FIFO RX Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[23:19]	–	reserved	–	–	–	–	–	–	–	–
			[26:24]	CFRFIF	Transmit/Receive FIFO FDC level Full Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[27]	–	reserved	–	–	–	–	–	–	–	–
			[30:28]	CFOFRIF	Transmit/Receive FIFO One Frame RX Interrupt Flag Channel m	0	0	unch	0	unch	unch	0	unch
			[31]	–	reserved	–	–	–	–	–	–	–	–

Note: After HW Reset, the RSCFDnCFDGSTS.GRAMINIT is set to indicate that the RS-CANFD module is initializing the RAM.
This bit is cleared automatically when the RAM initialization is completed and will not set again until HW reset is activated.

23.5 Initialization

Before joining CAN communications the following shall be configured:

Clock setting

Bit timing setting (nominal and data rate)

Baud Rate setting (nominal and data rate)

CANFD setting

Acceptance Filter setting (configuration of Global Acceptance Filter List)

Reception-, Transmission- and GW-FIFO setting

CAN Communication Mode setting

23.5.1 Initialization of CAN Clock, Bit Timing and Baud Rate

23.5.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

(1) Each segment setting

SS = Fixed to 1 TQ

TSEG1 = Refer to (RSCFDnCFDCmNCFG) and (RSCFDnCFDCmDCFG)

TSEG2 = Refer to (RSCFDnCFDCmNCFG) and (RSCFDnCFDCmDCFG)

SJW = Refer to (RSCFDnCFDCmNCFG) and (RSCFDnCFDCmDCFG)

SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

(2) Limitations on TSEG1, TSEG2 and SJW

TSEG1(N) > TSEG2(N) ≥ SJW(N)

TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)

When only classical frames will be used, the user should configure the bit fields TSEG1 and TSEG2 of RSCFDnCFDCmDCFG to valid values.

Table 23.142 shows an example of how to set the bit timing to achieve required Sample Point settings.

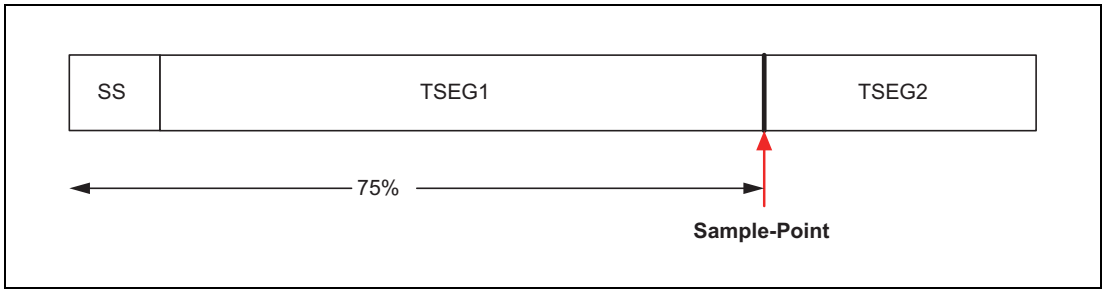
Table 23.142 Bit timing examples (1/2)

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33

Table 23.142 Bit timing examples (2/2)

1 bit	Set value (TQ)				Sample point*1(%)
	SS	TSEG1	TSEG2	SJW	
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)



23.5.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments, which can be configured individually for each channel via the related RSCFDnCFDCmNCFG and RSCFDnCFDCmDCFG registers.

Figure 23.14 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (hereafter TSEG1) and Time Segment 2 (hereafter TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN Bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (hereafter TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the Baud Rate Prescaler (nominal and data rate).

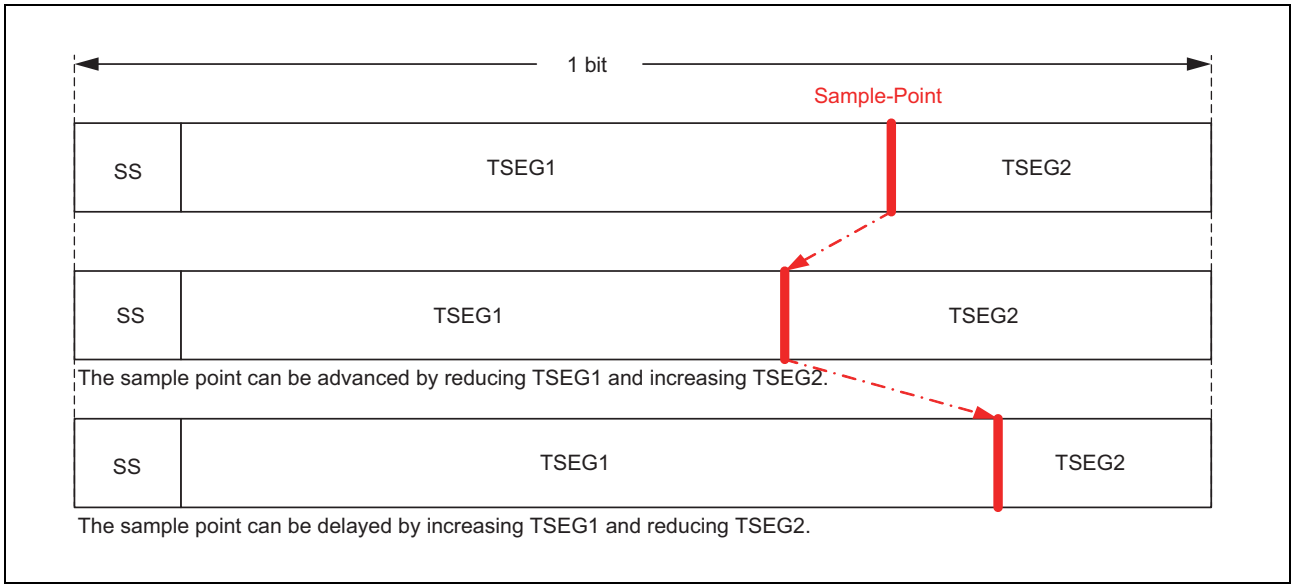


Figure 23.14 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the Interframe Space (comprised of Intermission, Suspend Transmission, and Bus Idle; during Bus Idle, all nodes can start transmission)
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW (While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error).
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.
   The above figure reports only one symbolic sample point.

23.5.1.3 Baud Rate

Either the CAN channel system clock (Clean clock) or the External oscillator clock can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL Clock, the divide-by-N value of the Baud Rate Prescaler, and the number of TQs in one bit.

$$baudrate = \frac{DLL\_Clock}{number\_of\_time\_quanta\_per\_bit) \times (BRP + 1)}$$

Figure 23.15 shows a block diagram of the circuit that generates the CAN channel system clock and **Table 23.144** shows a Baud Rate examples.

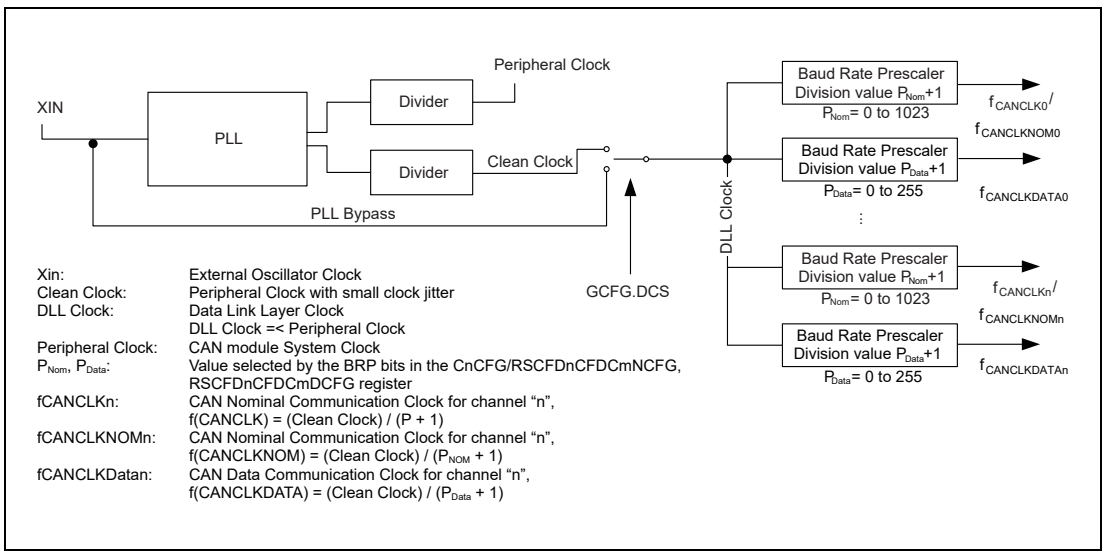


Figure 23.15 Block diagram of the circuit that generates the CAN channel communication clock

Table 23.143 Nominal Baud Rate calculation formula and example CAN communication configurations

Baud rate	DLL Clock								
	80MHz	40MHz	32MHz	30MHz	24MHz	20MHz	16MHz	10MHz	8MHz*2
1Mbps	8TQ (10) 20TQ (4)	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500 kbps	8TQ (20) 20TQ (8)	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250 kbps	8TQ (40) 20TQ (16)	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125 kbps	8TQ (80) 20TQ (32)	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3 kbps	8TQ (120) 12TQ (80) 16TQ (60) 24TQ (40)	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3 kbps	8TQ (300) 12TQ (200) 16TQ (150) 20TQ (120) 24TQ (100)	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 20TQ (30) 24TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

Note 1. Baud Rate Prescaler divide-by-N value = P + 1 (P = 0-1023)
P: value selected by the BRP bits in the Channel Configuration Register.

Note 2. Minimum Frequency to achieve max. Nominal Baud Rate of 1Mbps

Note: Shown in ( ) are the Baud Rate Prescaler divide-by-N values

Table 23.144 Baud Rate calculation example for nominal and data bit rate CAN communication configurations

Baud rate	DLL Clock		
	80MHz	40MHz	20MHz
Nominal 1Mbps	80TQ (1)	40TQ (1)	20TQ (1)
Data 8Mbps	10TQ (1)	5TQ (1)	Not possible
Nominal 1Mbps	80TQ (1)	40TQ (1)	20TQ (1)
Data 5Mbps	16TQ (1)	8TQ (1)	Not possible
Nominal 500Kbps	160TQ (1)	80TQ (1)	40TQ (1)
Data 2Mbps	40TQ (1)	20TQ (1)	10TQ (1)

Note 1. Baud Rate Prescaler divide-by-N value = P + 1 (P = 0-1023)
P: value selected by the BRP bits in the Channel Configuration Register.

Note: Shown in ( ) are the Baud Rate Prescaler divide-by-N values

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in the nominal bit time and in the data bit time; means RSCFDnCFDCmNCFG.NBRP = RSCFDnCFDCmDCFG.DBRP.

Further if Transceiver Delay Compensation is used then the RSCFDnCFDCmDCFG.DBRP shall not be programmed greater than 1; 1 means divide by 2.

23.5.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 23.16 shows the procedure for setting the CAN clock and the Baud Rate for each channel.

These settings should be performed during Channel Reset Mode (Configuration Mode) for the CAN channels.

Before going to channel communication state the Baud Rate must be configured, otherwise the mode will not switch correctly.

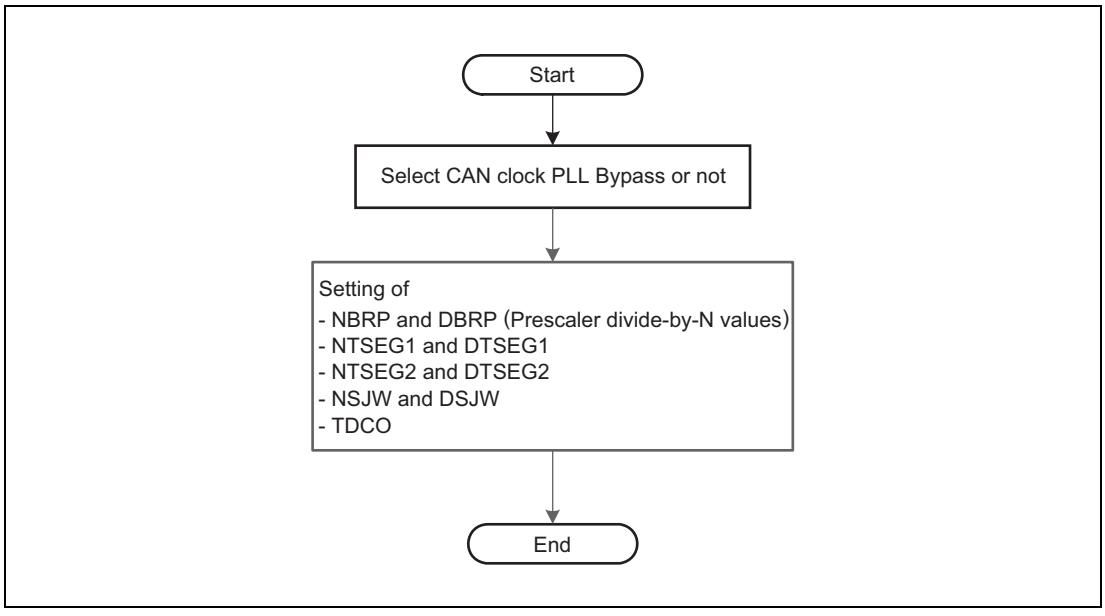


Figure 23.16 Procedure for setting the CAN bit timing and Baud Rate

23.5.1.5 Transmitter delay compensation

In case a high Baud Rate is used such as 5 to 8 Mbps for the Data phase, the Transmitter delay could become greater than TSEG1. In this case the transmitter would always detect a bit-error in the data-phase of the CANFD frame. The TDC compensates the transmitter's inability to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the Data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result (RSCFDnCFDCmFDSTS.TDCR) as shown in **Figure 23.17**.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

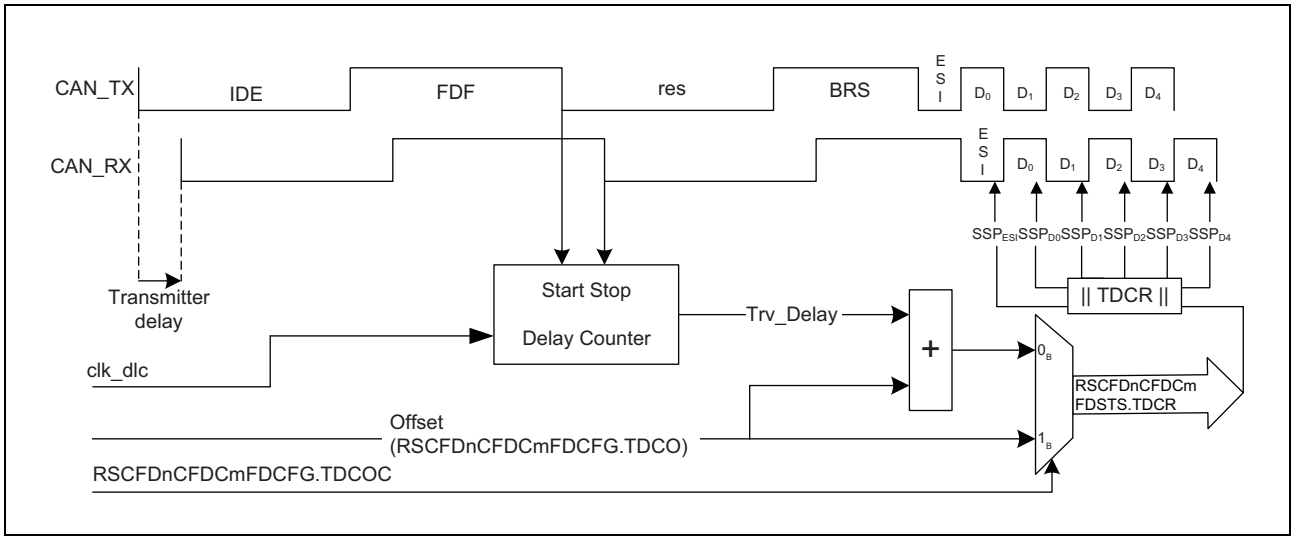


Figure 23.17 Transmitter Delay Compensation

The measured Trv_Delay is based on integer number of clk_dlc clock cycles.The delay is counted up by one for each started clock until the dominant value is seen on CAN_RX. The figure below illustrates the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

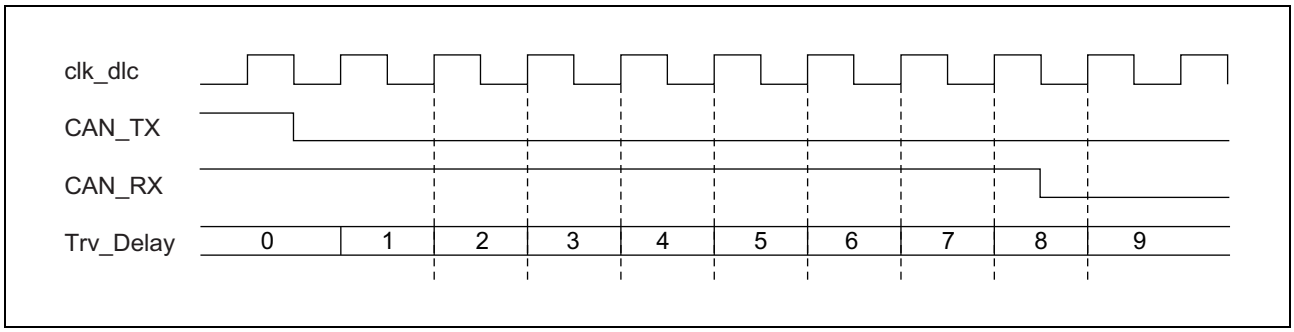


Figure 23.18 Trv_Delay measurement example

The SSP is calculated by taking the result from RSCFDnCFDCmFDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

The **Figure 23.19** illustrates the positioning of the secondary sample point. In case the RSCFDnCFDCmFDCFG.TDCOC is equal to 0_B the SSP is equal to the Trv_Delay (measured delay) + RSCFDnCFDCmFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually

the TDCO value should have the size of ($SyncSegment_{data}$ + $TSEG1_{data}$) to position the SSP to the theoretical location of a sample point.

If the RSCFDnCFDCmFDCFG.TDCOC is equal to 1_B then the SSP is just defined by the RSCFDnCFDCmFDCFG.TDCO. If the RSCFDnCFDCmDCFG.DBRP > zero then the value will also be rounded down to the nearest integer number of time quanta.

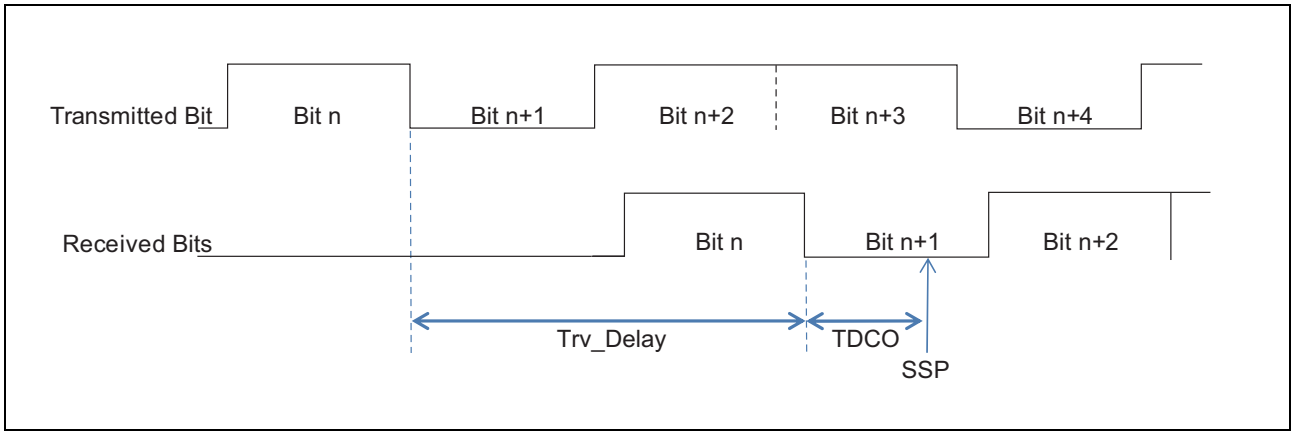


Figure 23.19 Position of the Secondary Sample Point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the RS-CANFD module is (6 data bits – 2clk_dlc).

The ISO 11898-1 (2015) allows the users to set different values for BRP_data and BRP_nom.

If different values are used for RSCFDnCFDCmNCFG.NBRP and RSCFDnCFDCmDCFG.DBRP, then 2 CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after Sample point of the BRS bit. This condition is shown in **Figure 23.20** below.

The length of the time quantum should be the same in the nominal bit time and in the data bit time; means RSCFDnCFDCmNCFG.NBRP = RSCFDnCFDCmDCFG.DBRP.

Different Bit rates can be achieved by selecting different configuration values for the Time Segments. Note the Nominal Bit rate can be configured from 8 to 385TQs and the Data Bit rate from 5 to 49TQs.

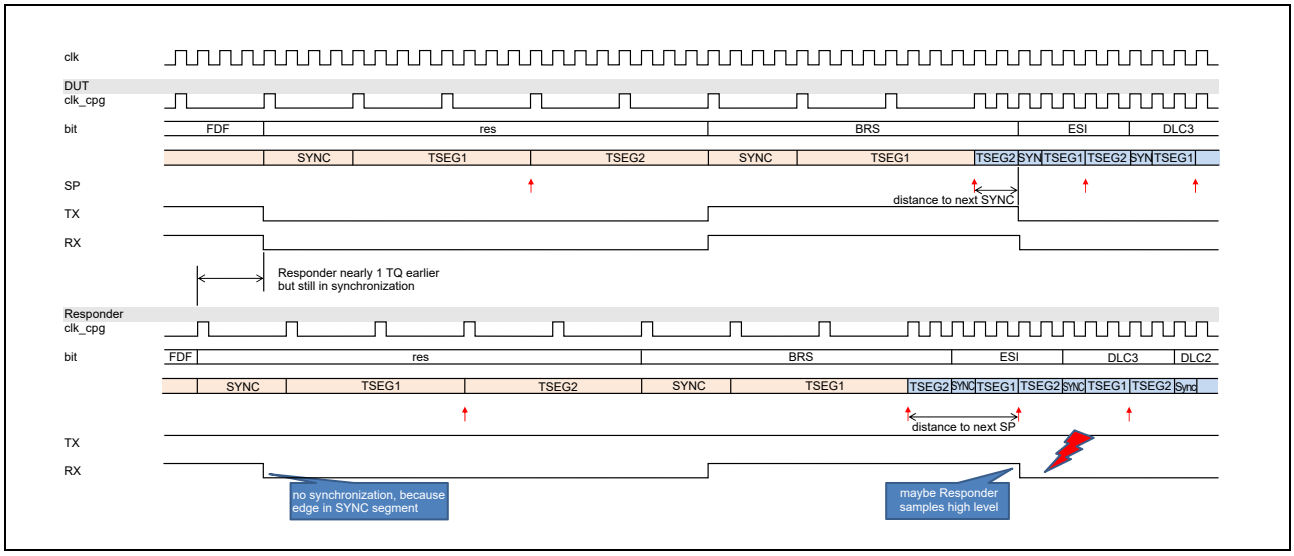


Figure 23.20 Loss of synchronization between 2 CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to res bit when configured accordingly (RSCFDnCFDCmFDCFG.TDCE = 1_B, RSCFDnCFDCmFDCFG.TDCOC = 0_B)

Figure 23.21 shows the read flow to get the measured transmitter delay compensation result.

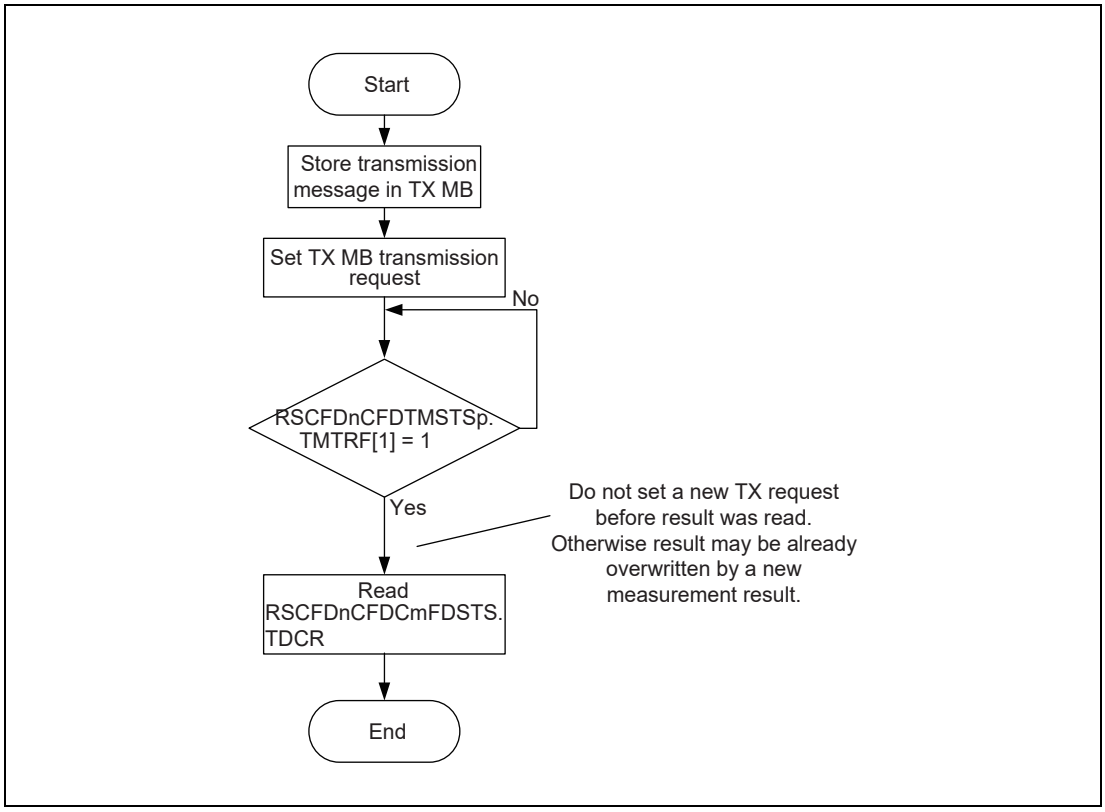


Figure 23.21 TDC result read flow

23.5.2 CAN Module Configuration after H/W Reset

After MCU HW is reset or after setting and clearing a RSCFDnCFDGRSTC.SRST bit, the RS-CANFD module enters Global Stop Mode automatically.

To enable configuration of the RS-CANFD module, the Stop Mode has to be left by clearing the Global Stop Request bit RSCFDnCFDGCTR.GSLPR to 0_B.

After MCU HW reset the module starts the RAM initialization, the RSCFDnCFDGSTS.GRAMINIT bit in the Global Status Register is set automatically indicating that the RS-CANFD logic is initializing the RAM.

After the RAM initialization is completed, this bit is cleared automatically.

The RAM initialization is necessary to avoid setting of false ECC error flag after HW reset due to random data present in the RAM.

The registers of RS-CANFD should not be accessed (in either read or write) until the RAM initialization is complete and the RSCFDnCFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode the Global Acceptance Filter List and message FIFO buffers must be configured. Furthermore each required CAN channel has to be configured (e.g. CAN bit timing)

For this all required CAN channels have to be released from channel Stop Mode and have to be configured for communication in channel Reset Mode (Configuration Mode).

Figure 23.22, Configuration procedure after H/W Reset shows the configuration procedure. For details about each step, refer to **Section 23.6, Acceptance Filtering Function using Global Acceptance Filter List (AFL)**, **Section 23.7, FIFO Buffers & Normal MB Configuration**, **Section 23.8, Interrupts and DMA** and **Section 23.5.1.3, Baud Rate**.

The RS-CANFD module does not perform the RAM initialization sequence after executing SW Reset by setting RSCFDnCFDGRSTC.SRST.

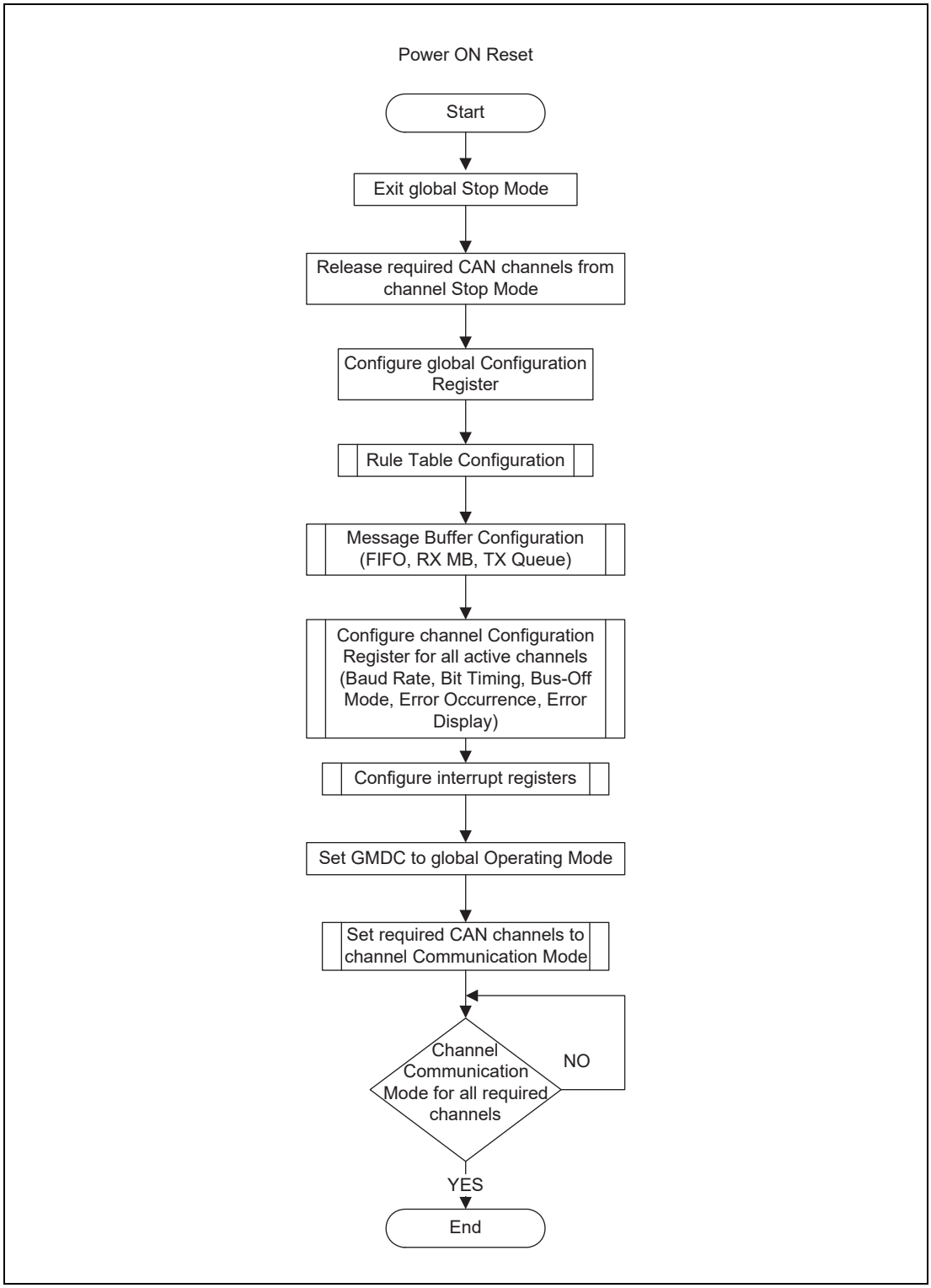


Figure 23.22 Configuration procedure after H/W Reset

23.6 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

23.6.1 Overview

The RS-CANFD module can handle message acceptance filtering for all channels with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

Following actions will be done based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the RSCFDnCFDGCFG.CMPOC bit
- Storage of accepted messages in the Message Buffer objects defined in the related AFL entry
- Attaching a 16 bit pointer to the stored messages defined in the related AFL entry e.g. to support AUTOSAR applications
- Attaching a 2 bit information label to the stored messages defined in the related AFL entry

The RS-CANFD module allows a maximum of (Number of AFL) x (Number of Channel) entries across all channels with a maximum of 384 AFL entries for U2A-EVA, U2A16, U2A8 and 255 AFL entries for U2A6 per single channel in a unit.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. Check is starting from the lowest AFL entry number for this channel.

AFL search is stopped when a match of the received Identifier with a configured Identifier/Mask combination occurs or when the received Identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC Check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, then the DLC check is passed.

If DLC replacement (RSCFDnCFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0000_H and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional Data Bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional Data Bytes will be stored as 0_H in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0_H, then the received value of DLC will be stored in the destination RX MB or FIFO Buffer.

If DLC replacement (RSCFDnCFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN Bus is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional Data Bytes received from the CAN Bus are also stored in the destination RXMB or FIFO Buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and it is not stored in any RXMB or FIFO Buffer.

Additionally, DLC check failure will be flagged by the DLC Error Flag in the Global Error Flag Register.

If configured, an error interrupt will also be generated.

The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception Message Buffer and/or in FIFO buffers configured for reception or gateway function or Tx queue configured for gateway function.

This message storage target information is also defined in the same AFL entry. Users should not set a target at the AFL entry which is not configured.

Each accepted received message could be stored into a maximum of 8 different target destinations (single reception Message Buffer and/or FIFO buffers).

(Programming of more than 8 target destinations is not allowed. In the case more destinations are programmed then internal timing race condition can occur and received message may not be stored to the Message RAM. Correct configuration of the numbers of Target destination is the responsibility of the application)

Further protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (RSCFDnCFDRMNB.RMPLS, RSCFDnCFDRFCCx.RFPLS or RSCFDnCFDCFCCk.CFPLS).

If the RSCFDnCFDGCFG.CMPOC = 0_B then the message is completely rejected and will not be stored in the target destination. In the case of RSCFDnCFDGCFG.CMPOC = 0_B and RX or Transmit/Receive FIFO full and the received message contains more data payload Bytes than possible to store in the target destination (RSCFDnCFDRMNB.RMPLS, RSCFDnCFDRFCCx.RFPLS or RSCFDnCFDCFCCk.CFPLS), the corresponding RSCFDnCFDFMSTS.RFxMLT or RSCFDnCFDFMSTS.CFkMLT will not be set to 1_B respectively.

If the RSCFDnCFDGCFG.CMPOC = 1_B then the received Data Bytes greater than RSCFDnCFDRMNB.RMPLS will be rejected. In the case of RSCFDnCFDGCFG.CMPOC = 1_B and RX or Transmit/Receive FIFO full and the received message contains more data payload Bytes than possible to store in the target destination (RSCFDnCFDRMNB.RMPLS, RSCFDnCFDRFCCx.RFPLS or RSCFDnCFDCFCCk.CFPLS), the corresponding RSCFDnCFDFMSTS.RFxMLT or RSCFDnCFDFMSTS.CFkMLT will be set to 1_B respectively.

Depending on the RSCFDnCFDGCFG.DRE the original received DLC or the DLC value configured at the AFL entry will be stored.

Regardless of RSCFDnCFDGCFG.CMPOC configuration, RSCFDnCFDGERFL.CMPOF will be set to 1_B if payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. Hence for one reception frame only one flag could be set at the same time RSCFDnCFDGERFL.DEF or RSCFDnCFDGERFL.CMPOF.

23.6.2 Allocation of AFL entries to each CAN channel

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Register (see **Figure 23.23**).

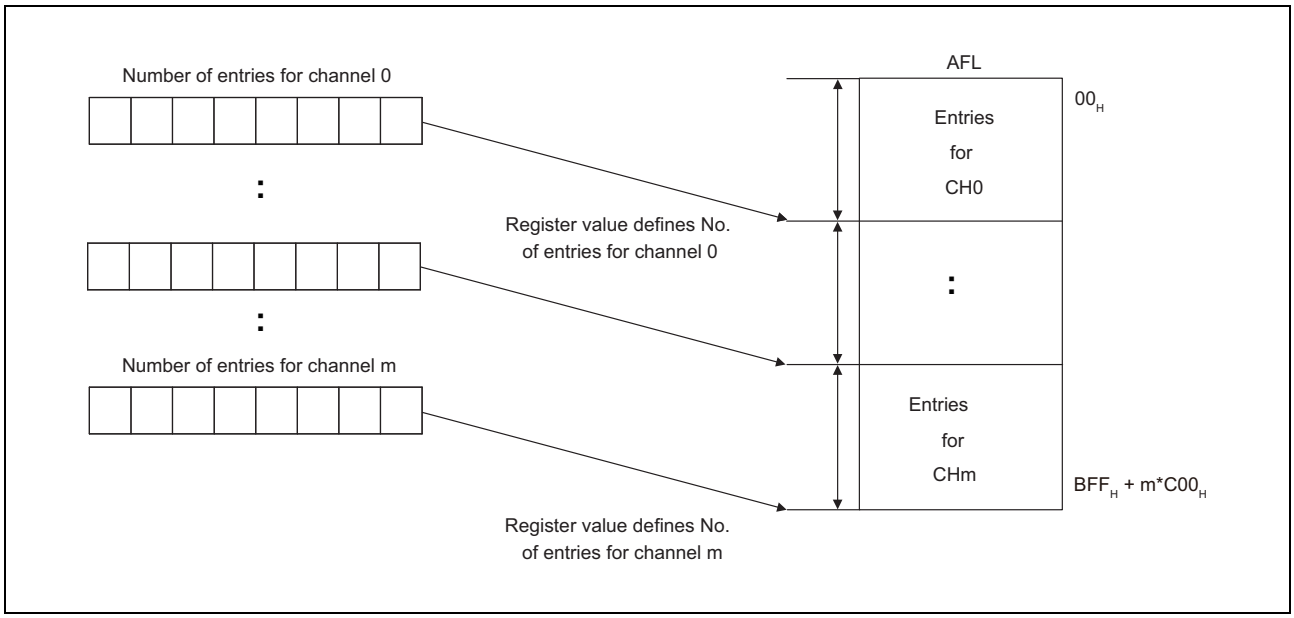


Figure 23.23 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel), the maximum number of entries for one channel is 384 for U2A-EVA, U2A16, U2A8 and 255 for U2A6. The total number of entries for all channels in a unit should not exceed the maximum limit of (m+1) x (Number of AFL).

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The RS-CANFD module will not flag errors related to the configuration of the AFL.

23.6.3 AFL entry description

Each AFL entry consists of 16 Bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

Identifier (11 bits for Standard Frame Format, 29 bits for Extended Frame Format):

Acceptance filter unit will check Identifier field of received message against Identifier field of each AFL entry (full 29 bit masking of Identifier bits possible, see below).

IDE bit:

Acceptance filter unit will check IDE bit of received message against this bit and selects the relevant part of the Identifier field for acceptance filtering (masking of IDE bit possible, see below).

RTR bit:

Acceptance filter unit will only accept Data Frames (RTR = 0) or Remote Frames (RTR = 1) according to the setting of this bit (masking of RTR bit possible, see below).

Loopback Configuration bit:

This bit can enable/disable the AFL entry depending on the Loopback Configuration or Mirror Mode condition.

Mask for Identifier bits (29 bits):

Each bit in the Identifier Mask can mask the corresponding Identifier bit in the AFL entry during acceptance filtering (**Figure 23.24**).

Mask for IDE bit:

If this Mask bit masks the IDE bit of the AFL entry both Standard Identifier and Extended Identifier format messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.

Mask for RTR bit:

If this Mask bit masks the RTR bit of the AFL entry both frame formats ‘Data Frame’ and ‘Remote Frame’ will be accepted by this AFL entry.

Pointer information (16 bits):

This 16 bit pointer will be attached to a received message accepted by the related AFL entry. The Pointer will be added during message storage in the Message Buffer area and can be used by application as support function. The pointer information could be used for example to support PDU Identifier allocation for the received message in AUTOSAR systems.

Information Label (2 bits):

This 2bit label will be attached to a received message accepted by the related AFL entry. The label will be added during message storage in the Message Buffer area and can be used by application as support function.

DLC value for automatic DLC filtering:

If the DLC value of the received message is equal or higher than the configured DLC value the DLC check is passed.

If the DLC value in this AFL entry is configured to 1’b0 DLC filtering is effectively disabled for this entry (all accepted messages will pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages.

Message Buffer number of one single reception Message Buffer as target for received message storage.

Single reception Message Buffer enable bit to configure the single reception Message Buffer number as valid/invalid as target for received message storage

FIFO direction pointer: each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message.

NOTE

A message received on channel “A” can be routed to Transmit/Receive FIFO Buffer of another channel and if this Transmit/Receive FIFO Buffer is configured in GW mode, then the message stored in this Transmit/Receive FIFO Buffer will be transmitted on that channel since Transmit/Receive FIFO Buffer is associated with channel.

There is no hardware protection against such storage of message. Hence the FIFO direction pointer should be configured carefully.

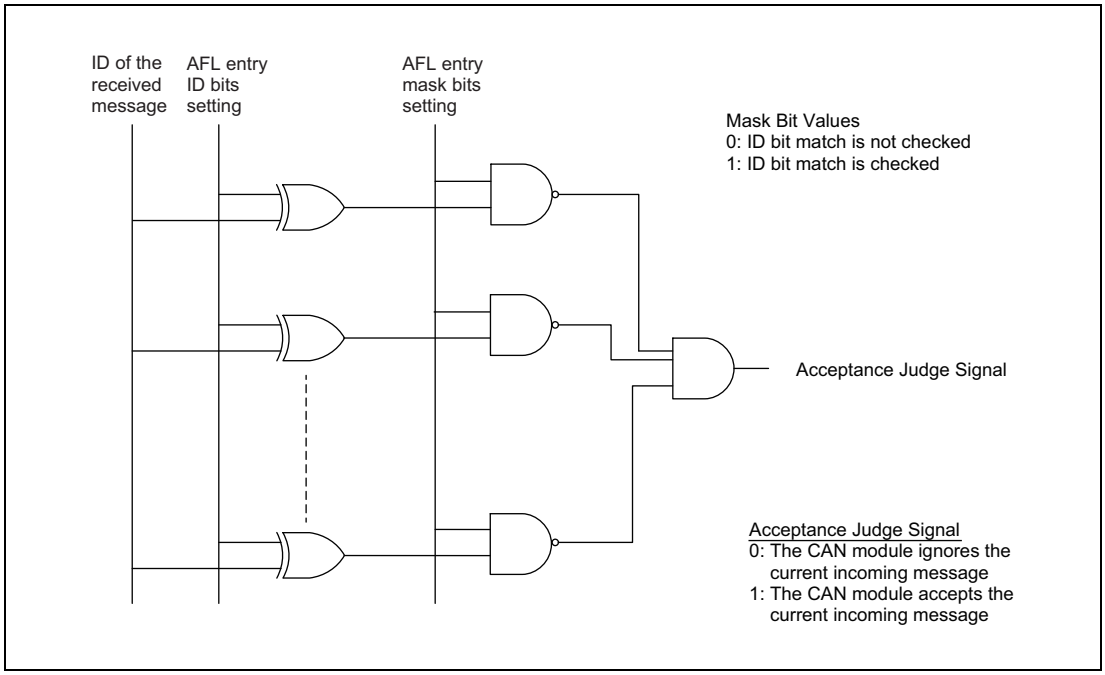


Figure 23.24 Acceptance Function

23.6.4 Entering entries in the AFL

Application SW can enter one full entry into the AFL via following registers:

- Global AFL ID Register: Part 1 of the AFL entry
- Global AFL Mask Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Register: Part 4 of the AFL entry

16 sets of these registers form a group of AFL entries. Each group can be accessed via a page mechanism. For the RS-CANFD module, 8-channel version, 96 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT. Pages are linked to the AFL entries in the following way:

NOTE

In U2A6, the maximum number of page of RSCFD0 is 64, the maximum number of page of RSCFD1 is 32.

Page 0	Entry 0 to 15
Page 1	Entry 16 to 31
Page 2	Entry 32 to 47
Page 3	Entry 48 to 63
Page 4	Entry 64 to 79
Page 5	Entry 80 to 95
Page 6	Entry 96 to 111
Page 7	Entry 112 to 127

Page 8	Entry 128 to 143
Page 9	Entry 144 to 159
Page 10	Entry 160 to 175
Page 11	Entry 176 to 191
Page 12	Entry 192 to 207
Page 13	Entry 208 to 223
Page 14	Entry 224 to 239
Page 15	Entry 240 to 255
Page 16	Entry 256 to 271
Page 17	Entry 272 to 287
Page 18	Entry 288 to 303
Page 19	Entry 304 to 319
Page 20	Entry 320 to 335
Page 21	Entry 336 to 351
Page 22	Entry 352 to 367
Page 23	Entry 368 to 383
Page 24	Entry 384 to 399
Page 25	Entry 400 to 415
Page 26	Entry 416 to 431
Page 27	Entry 432 to 447
Page 28	Entry 448 to 463
Page 29	Entry 464 to 479
Page 30	Entry 480 to 495
Page 31	Entry 496 to 511
Page 32	Entry 512 to 527
Page 33	Entry 528 to 543
Page 34	Entry 544 to 559
Page 35	Entry 560 to 575
Page 36	Entry 576 to 591
Page 37	Entry 592 to 607
Page 38	Entry 608 to 623
Page 39	Entry 624 to 639
Page 40	Entry 640 to 655
Page 41	Entry 656 to 671
Page 42	Entry 672 to 687
Page 43	Entry 688 to 703
Page 44	Entry 704 to 719
Page 45	Entry 720 to 735
Page 46	Entry 736 to 751
Page 47	Entry 752 to 767
Page 48	Entry 768 to 783
Page 49	Entry 784 to 799
Page 50	Entry 800 to 815
Page 51	Entry 816 to 831
Page 52	Entry 832 to 847
Page 53	Entry 848 to 863

Page 54	Entry 864 to 879
Page 55	Entry 880 to 895
Page 56	Entry 896 to 911
Page 57	Entry 912 to 927
Page 58	Entry 928 to 943
Page 59	Entry 944 to 959
Page 60	Entry 960 to 975
Page 61	Entry 976 to 991
Page 62	Entry 992 to 1007
Page 63	Entry 1008 to 1023
Page 64	Entry 1024 to 1039
Page 65	Entry 1040 to 1055
Page 66	Entry 1056 to 1071
Page 67	Entry 1072 to 1087
Page 68	Entry 1088 to 1103
Page 69	Entry 1104 to 1119
Page 70	Entry 1120 to 1135
Page 71	Entry 1136 to 1151
Page 72	Entry 1152 to 1167
Page 73	Entry 1168 to 1183
Page 74	Entry 1184 to 1199
Page 75	Entry 1200 to 1215
Page 76	Entry 1216 to 1231
Page 77	Entry 1232 to 1247
Page 78	Entry 1248 to 1263
Page 79	Entry 1264 to 1279
Page 80	Entry 1280 to 1295
Page 81	Entry 1296 to 1311
Page 82	Entry 1312 to 1327
Page 83	Entry 1328 to 1343
Page 84	Entry 1344 to 1359
Page 85	Entry 1360 to 1375
Page 86	Entry 1376 to 1391
Page 87	Entry 1392 to 1407
Page 88	Entry 1408 to 1423
Page 89	Entry 1424 to 1439
Page 90	Entry 1440 to 1455
Page 91	Entry 1456 to 1471
Page 92	Entry 1472 to 1487
Page 93	Entry 1488 to 1503
Page 94	Entry 1504 to 1519
Page 95	Entry 1520 to 1535

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (RSCFDnCFDGAFLECTR) (**Figure 23.25**). This register has following fields:

- 7 bit to select the AFL Page number
- 1 bit to enable/disable the AFL Data access to prevent unwanted write access to the AFL

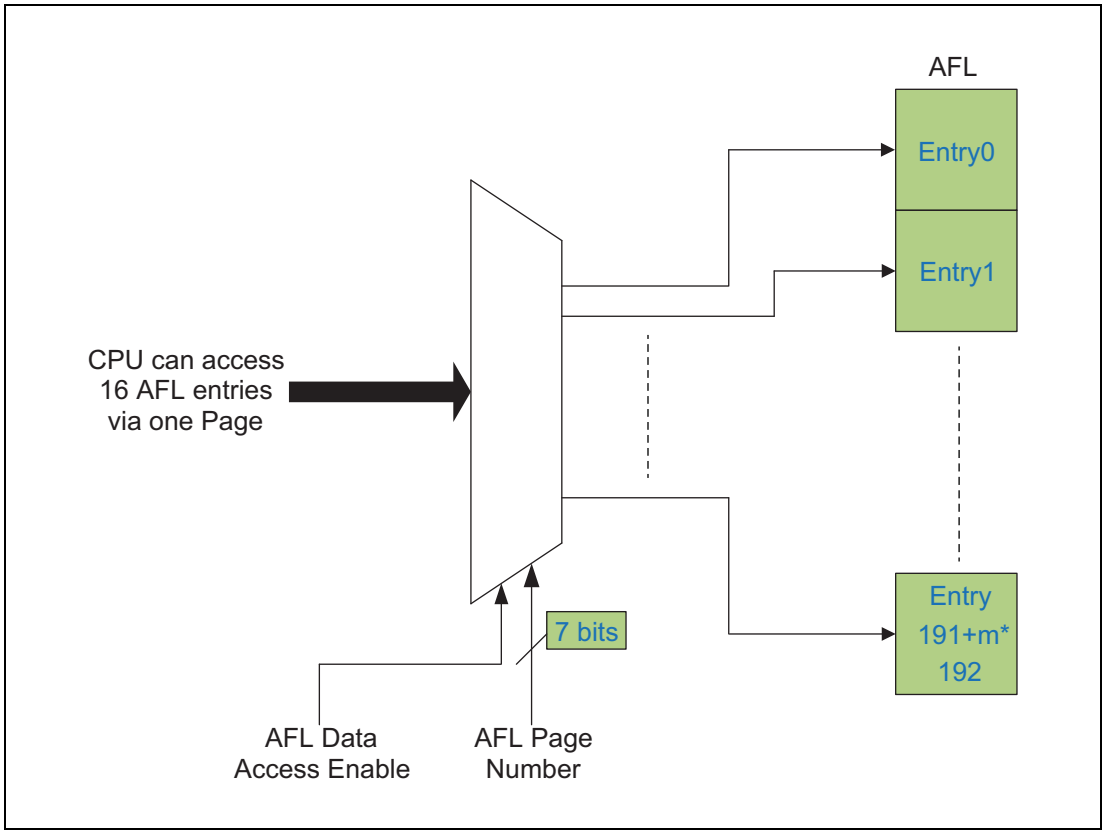


Figure 23.25 AFL page access

Application SW should not write numbers higher than $5F_H$ for the AFL Page number.

The configuration flow shown in **Figure 23.26** should be followed to program the AFL.

After entering all entries in Configuration Mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all global modes if the lock bit is set (GL_RESET, GL_TEST, and GL_OPERATING).

Read access to AFL is still possible during all global modes even if AFL data access is disabled (consistency check of AFL contents possible during runtime).

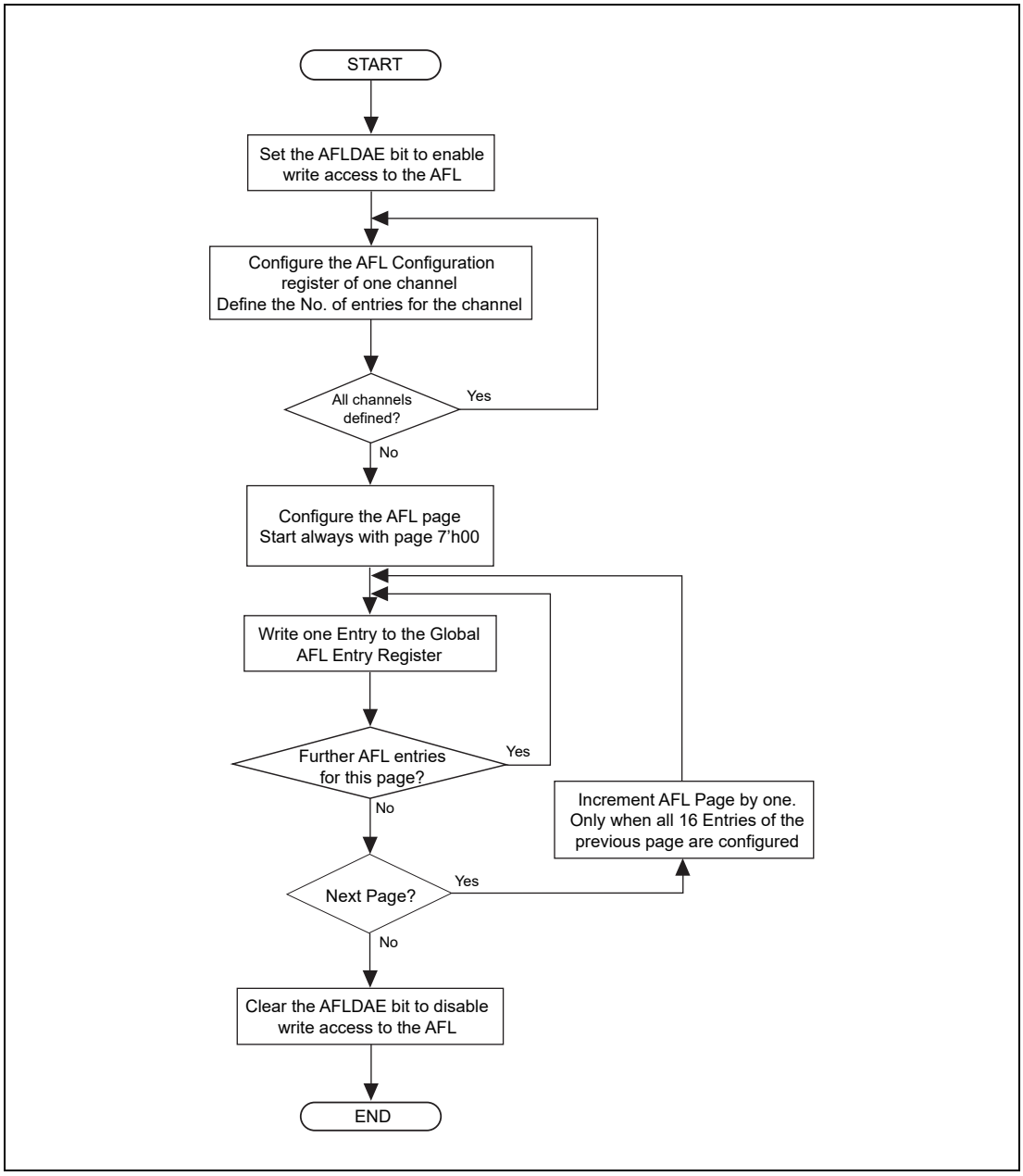


Figure 23.26 AFL configuration flow

23.6.5 Loopback modes

If the Loopback Configuration bit is set, the AFL entry is only valid in loopback test mode (Self test mode 0 or Self test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus (the expression ‘valid or invalid’ for the related entry means that this AFL entry ‘will or will not’ be compared against the received message ID respectively).

If the Loopback Configuration bit is 0_B the AFL entry is only valid for

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror mode
- Received messages transmitted by other CAN nodes or the CAN channel itself in loopback test mode

The mirror mode can be enabled via the RSCFDnCFDGCFG.MME bit in the Global Configuration Register. If RSCFDnCFDGCFG.MME bit is set, then a successfully Transmitted message can be stored back in an RX MB or FIFO Buffer if a matching Entry is configured in the AFL for that channel.

The Loopback configuration bit in the matching AFL Entry must be set for storing this Frame.

If mirror mode and loopback test mode are configured at the same time the loopback test mode behavior applies.

Table 23.145 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 23.145 Acceptance filter behavior based on Loopback Configuration setting in AFL entry

Mirror mode enable (MME configuration bit)	Loopback in test mode (Selftest Mode 0 or Selftest Mode 1)	Channel Mode	Loopback configuration bit in AFL Entry	AFL entry
0	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	invalid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid
1	0	Receiver	0	valid
			1	invalid
		Transmitter	0	invalid
			1	valid
	1	Receiver	0	valid
			1	invalid
		Transmitter	0	valid
			1	valid

Note: The expression ‘valid’ or ‘invalid’ for the related entry means that this AFL entry ‘will or will not’ be compared against the received message ID respectively.

23.6.6 IDE Masking

When the GAFLIDEM bit is 0_B in an AFL entry, then the IDE bit configured in the AFL Entry is not considered for ID matching. In this case the decision of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following Example:

The ID & Mask fields of an AFL Entry “x” is configured as follows:

RSCFDnCFDGAFLID [x] = $C0553A20_H$ → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 220_H / ID[28:0] = $0553A20_H$

RSCFDnCFDGAFLMj = $0000FFFF_H$ → IDEM = 0. RTRM = 0, IDM[10:0] = $7FF_H$ / IDM[28:0] = $0000FFFF_H$

The result of comparison of 4 different received Ids with AFL Entry X is described below:

If a frame with IDE = 0_B & ID = 220_H is received, then this is considered as a match.

If a frame with IDE = 0_B & ID = 320_H is received, then this is not a match.

If a frame with IDE = 1_B & ID = $1FFF3A20_H$ is received, then this is considered as a match.

If a frame with IDE = 1_B & ID = 08803220_H is received, then this is not a match.

23.7 FIFO Buffers & Normal MB Configuration

This Section describes the process for configuration of the number of RX Message Buffers, the FIFO Buffers and the Flat TX Message Buffers in the RS-CANFD module. The Message Buffers are mapped as shown in **Figure 23.27** below.

The RX Message Buffers can be accessed via RX Message Buffer Register.

The RX FIFO buffers and the Transmit/Receive FIFO buffers configured in RX mode or TX mode or GW mode can only be accessed via the FIFO Access Register.

If the Transmit/Receive FIFO is configured in TX mode, then users can only write data into the FIFO via FIFO Access registers.

If the Transmit/Receive FIFO is configured in GW mode or RX mode then users can only read data from the FIFO Access Register.

The TX Message Buffers can be accessed via the TX Message Buffer Register.

If users read unused Message Buffer locations, the MB locations are read as unknown values.

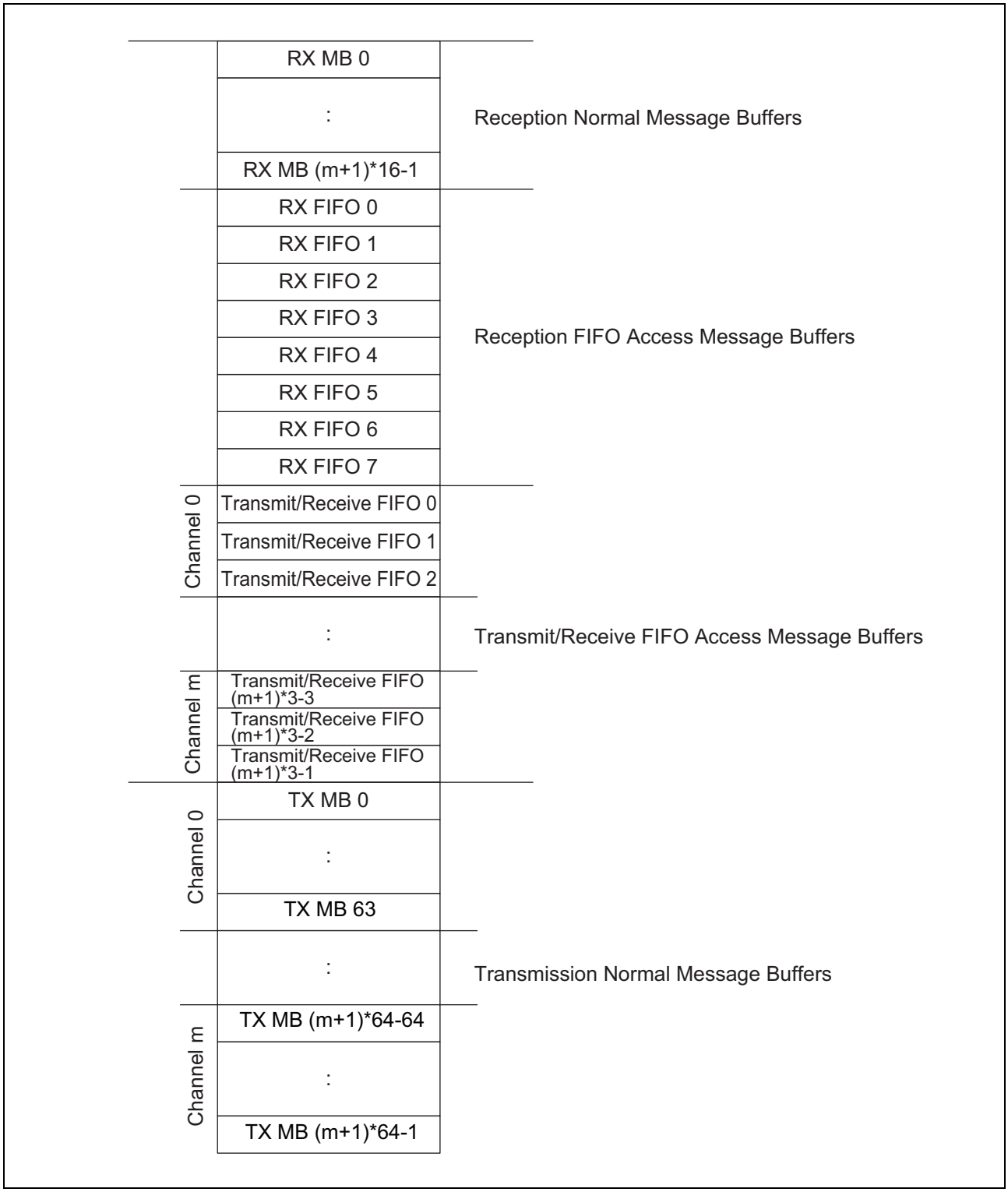


Figure 23.27 Message Buffer Configuration

23.7.1 Normal RX Message Buffers

In RS-CANFD module, the frames received by various channels can be stored in Normal RX Message Buffers based on the configuration of the AFL entries.

Additionally, the number of Normal RX Message Buffers required in the system can be chosen up to a fixed maximum limit.

23.7.1.1 Normal RX Message Buffer configuration

In RS-CANFD module, the number of normal RX Message Buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of Message Buffers are:

Minimum Value = 0_H (no Normal RX MB)

Maximum Value = (16 * No. of CAN channels)

= 80_H (128 Flat RX MBs for 8 channels)

Users should not use values outside these limits.

The AFL entries for routing the received messages to normal RX Message Buffers should be configured to match the requirements of the system.

The AFL entries should also be configured properly. An AFL entry for normal RX Message Buffers should not exceed the number of Message Buffers configured in the RX Message Buffer Number Register.

NOTE

There is no internal check procedure provided in RS-CANFD module against wrong configuration of the AFL.

The data field size of the RX Message Buffer can be configured via the RSCFDnCFDRMNB.RMPLS. The default size is 8 Bytes the max. data payload size is 64 Bytes.

In case the receiving frame exceeds the data field size then the acceptance depends on the configuration of the RSCFDnCFDGCFG.CMPOC (message rejecting or data payload cut).

23.7.2 FIFO Buffers

The RS-CANFD module provides a fixed number of FIFO Buffers to support storage of frames for reception, transmission and gateway functions for various CAN channels.

Number of reception-only FIFO Buffers is fixed to 8.

However, 3 Transmit/Receive FIFO Buffers per channel can be configured for storing messages for transmission or reception or gateway function.

These FIFO Buffers can be enabled or disabled and the size, Interrupt structure and Message Lost mechanism and Message overwrite mechanism of the FIFO Buffers, as well as the location of the TX FIFO or GW FIFO can be configured to match the system requirements.

In case the receiving frame exceeds the data field size then the acceptance depends on the configuration of the RSCFDnCFDGCFG.CMPOC (message rejecting or data payload cut).

23.7.2.1 FIFO Buffers configuration

In RS-CANFD module, the FIFO Buffers can be configured to match the system requirements.

The total number of FIFO Buffers = 8 RX FIFO Buffers + 24 Transmit/Receive FIFO Buffers =32 FIFO Buffers for 8 channels and Message overwrite mechanism

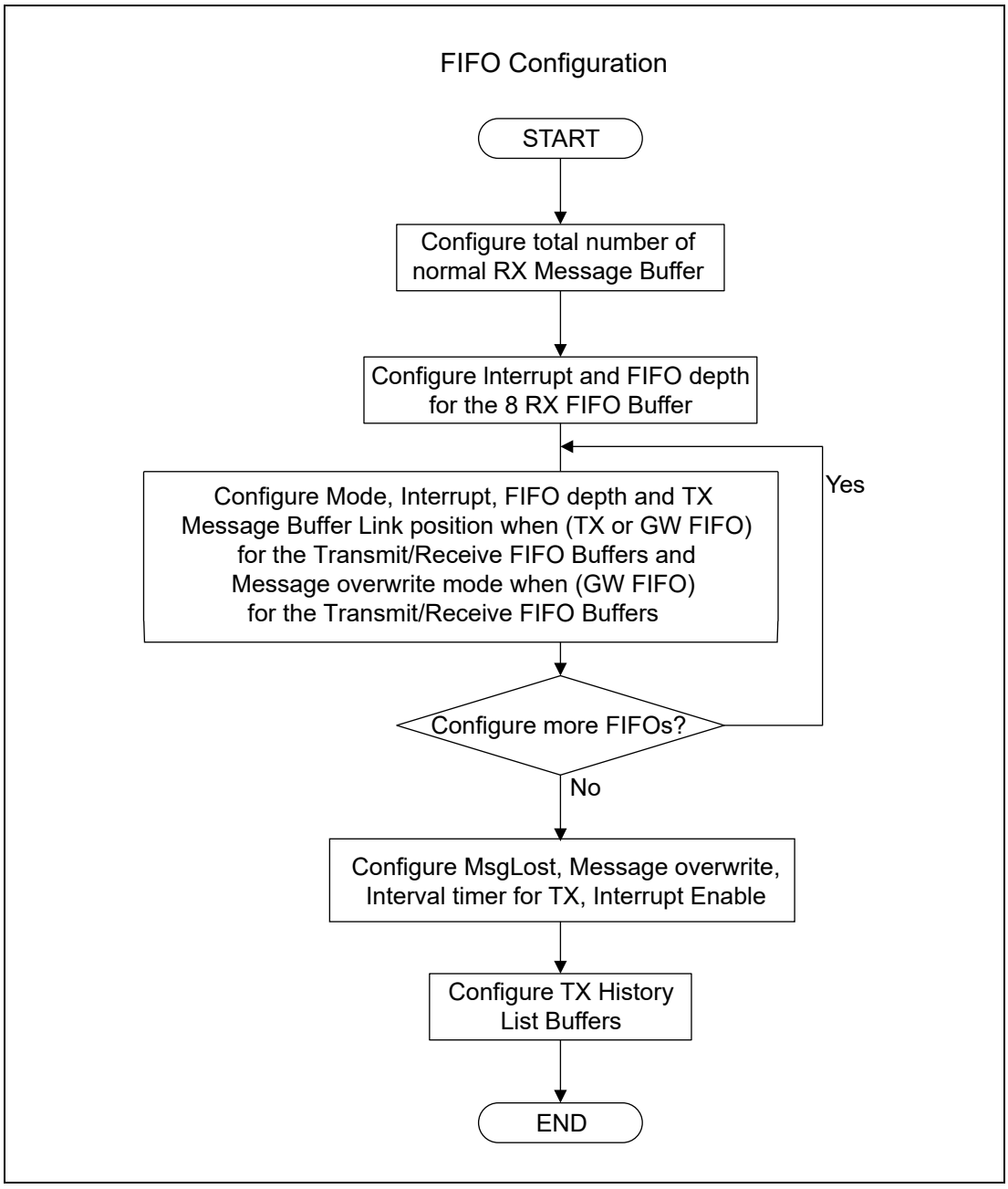


Figure 23.28 FIFO Buffer configuration flow in RS-CANFD module

As shown in **Figure 23.28**, the various FIFO Buffers can be configured by writing to the RX FIFO Configuration/Control Register and the Transmit/Receive FIFO Configuration/Control Register.

For the 8 RX FIFO Buffers, the Interrupts, the FIFO depth and the FIFO payload data size can be configured.

For the Transmit/Receive FIFO Buffers, the mode, Interrupts FIFO depth, the FIFO payload data size and the FIFO TX link position can be configured.

(1) FIFO Mode Configuration of Transmit/Receive FIFO Buffers

The mode of the Transmit/Receive FIFO Buffers can be configured by writing to the RSCFDnCFDCFCCk.CFM[1:0] bits in the Transmit/Receive FIFO Configuration/Control Register. The possible modes of configuration for Transmit/Receive FIFO Buffers are:

00_B: RX mode (default mode after HW Reset)

01_B: TX mode

10_B: GW mode.

11_B: Reserved (Users should not write this value to the Register bits)

Messages can only be read from the RX FIFO Buffers and the Transmit/Receive FIFO Buffers configured in RX Mode.

Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Transmit/Receive FIFO Buffers configured in TX mode. These messages will be transmitted on the appropriate CAN channel.

Messages can only be read from the Transmit/Receive FIFO Buffers configured in GW mode. However, the CPU read access has no impact on the read or write pointers.

The pointers can only be incremented when a new message is stored in the FIFO Buffer and decremented when a message is transmitted on the corresponding CAN channel by the RS-CANFD module.

After HW reset, all the Transmit/Receive FIFO Buffers are configured in RX mode by default. Users should only enable the FIFO Buffers after configuring the Transmit/Receive FIFO Buffers in the required modes.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer will be overwritten with the message received or the message will be discarded. The behavior is determined by setting RSCFDnCFDCFCCEk.CFMOWM bit.

When RSCFDnCFDCFCCEk.CFMOWM=0:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message will be discarded. And RSCFDnCFDCFSTSk.CFMLT bit is set to 1.

When RSCFDnCFDCFCCEk.CFMOWM=1:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer will be overwritten with the received message.

The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.

Then, RSCFDnCFDCFSTSk.CFMOW bit is set to 1, which notifies that the oldest message has been overwritten with the received message.

In addition, in a case a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and re-transmission for the message is not performed. Then the read point moves to the next message automatically.

Users should not write change for this bit when the RSCFDnCFDCFCCk.CFE bit is 1_B.

(2) FIFO TX-Message Buffer Link configuration

When the Transmit/Receive FIFO is configured as TX or GW FIFO, then the FIFO Buffer must be linked to a normal TX Message Buffer to participate in the transmission scan of a CAN channel.

The link to a normal TX Message Buffer should be unique i.e. the same TX Message Buffer cannot be shared between 2 or more Transmit/Receive FIFO Buffers.

Users should not write data into a TX Message Buffer that is linked to a Transmit/Receive FIFO buffer.

Also, the TX Message Buffer linked to a Transmit/Receive FIFO buffer should not be a part of the TX Queue.

The TX Message Buffer link of each Transmit/Receive FIFO Buffer can be configured by writing to the RSCFDnCFDCFCCk.CFTML[4:0] bits in the Transmit/Receive FIFO Configuration / Control Register. Available options for TX Message Buffer link configuration are:

00000_B: TX Message Buffer 32

00001_B: TX Message Buffer 33

: :

: :

11110_B: TX Message Buffer 62

11111_B: TX Message Buffer 63

(3) FIFO Depth Configuration

The depth of each FIFO Buffer can be configured by writing to the RSCFDnCFDRFCCx.RFDC[2:0] bits and RSCFDnCFDCFCCk.CFDC[2:0] bits in the RX FIFO Configuration / Control Register and the Transmit/Receive FIFO Configuration / Control Register. The 8 available options for depth configuration are:

000_B: 0 Messages (FIFO Buffer cannot be enabled)

001_B: 4 Messages

010_B: 8 Messages

011_B: 16 Messages

100_B: 32 Messages

101_B: 48 Messages

110_B: 64 Messages

111_B: 128 Messages

The RAM allocation for RX Message Buffers along with FIFO Buffers is limited to (m+1)*256 messages. Configuration of the RX Message Buffers, along with FIFO Buffers, that exceeds this maximum limit should not be done.

RS-CANFD module logic will not check the validity of the configuration.

NOTE

If the FIFO depth of a Transmit/Receive FIFO is 4 messages or more (RSCFDnCFDCFCCk.CFDC[2:0] > 000_B), then the Transmit/Receive FIFO TX Message Buffer link is valid when the FIFO is disabled as well as enabled.

If FIFO depth is 0 messages, then the Transmit/Receive FIFO TX Message Buffer link is not valid when the FIFO is disabled as well as enabled.

(4) FIFO Payload Size Configuration

The data size of each FIFO Buffer can be configured by writing to the RSCFDnCFDRFCCx.RFPLS[2:0] bits and RSCFDnCFDCFCCk.CFPLS[2:0] bits in the RX FIFO Configuration / Control Register and the Transmit/Receive FIFO Configuration / Control Register. The 8 available options for depth configuration are:

000_B: 8 Bytes

001_B: 12 Bytes

010_B: 16 Bytes

011_B: 20 Bytes

100_B: 24 Bytes

101_B: 32 Bytes

110_B: 48 Bytes

111_B: 64 Bytes

The RAM allocation for RX Message Buffers along with FIFO Buffers is limited to (m+1)*256 messages with 64 Data Bytes. Configuration of the RX Message Buffers, along with FIFO Buffers, that exceeds this maximum limit should not be done.

RS-CANFD module logic will not check the validity of the configuration.

(5) FIFO Interrupt Configuration

The Interrupt generation conditions for the FIFO Buffers can be configured by writing to the RSCFDnCFDRFCCx.RFIM and RSCFDnCFDCFCCk.CFIM bit in the RX FIFO Configuration / Control Register and the Transmit/Receive FIFO Configuration / Control Register. The 2 available options are:

0:

RX FIFO Mode: Interrupt generated when Transmit/Receive FIFO counter reaches RSCFDnCFDRFCCx.RFIGCV / RSCFDnCFDCFCCk.CFIGCV value from below values;

TX FIFO Mode: Interrupt generated when Transmit/Receive FIFO transmits last message successfully;

GW FIFO Mode:

Frame RX: Interrupt generated when message counter increments and reaches the Interrupt Threshold value;

Frame TX: Interrupt generated when last message is transmitted successfully from FIFO;

1:

RX FIFO Mode: Interrupt generated at the end of storage of every received message;

TX FIFO Mode: Interrupt generated for every successfully transmitted message;

GW FIFO Mode:

Frame RX: Interrupt generated when message is stored in the FIFO;

Frame TX: Interrupt generated when message is successfully transmitted from the FIFO;

If the interrupt mode bit is 0_B for a RX FIFO, then interrupt is generated based on the configuration of the RSCFDnCFDRFCCx.RFIGCV[2:0] bits.

Similarly, if the interrupt mode bit is 0_B for a Transmit/Receive FIFO configured in RX mode, then interrupt is generated based on the configuration of RSCFDnCFDCFCCk.CFIGCV[2:0] bits.

The 8 available options for configuring the FIFO counter value for generation of an interrupt are:

000: Interrupt generated when FIFO is 1/8th Full

001: Interrupt generated when FIFO is 1/4th Full

010: Interrupt generated when FIFO is 3/8th Full

011: Interrupt generated when FIFO is 1/2 Full

100: Interrupt generated when FIFO is 5/8th Full

101: Interrupt generated when FIFO is 3/4th Full

110: Interrupt generated when FIFO is 7/8th Full

111: Interrupt generated when FIFO is Full

In this case, an interrupt is generated when the Message Count matches the configured value.

However, there are some limitations on the configuration of the RSCFDnCFDRFCCx.RFIGCV[2:0] and RSCFDnCFDCFCCk.CFIGCV[2:0] bits depending upon the FDC[2:0] bits (FIFO Depth Configuration) see **Table 23.146**.

Table 23.146 FIFO Interrupt generation counter vs. FIFO depth configuration

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111	110	101	100	011	010	001	000
000	Don't care (FIFO can not be enabled)							
001	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010	Allowed							
011	Allowed							
100	Allowed							
101	Allowed							
110	Allowed							
111	Allowed							

Transmit/Receive FIFO can set an interrupt output at the time of the completion of transmitting of one frame, or the completion of reception. Moreover, Transmit/Receive FIFO and RX FIFO can set an interrupt output, when stored to the set-up number (CFDC/RFDC) of FIFO stages

23.7.2.2 FIFO Buffers control

The FIFO Interrupt should be enabled by setting the RSCFDnCFDRFCCx.RFIE bit or RSCFDnCFDRFCCx.RFFIE bits in the RX FIFO Configuration/Control Register and RSCFDnCFDCFCCk.CFRXIE or RSCFDnCFDCFCCk.CFTXIE bits or RSCFDnCFDCFCCEk.CFFIE bits or RSCFDnCFDCFCCEk.CFOFRXIE or RSCFDnCFDCFCCEk.CFOFTXIE in the Transmit/Receive FIFO Configuration/Control Register.

After configuration is complete, each FIFO can be enabled by setting the RSCFDnCFDRFCCx.RFE and RSCFDnCFDCFCCk.CFE bit in the RX FIFO Configuration / Control Register and the Transmit/ Receive FIFO Configuration/Control Register to allow transmission and reception of messages.

In the case of RSCFDnCFDCFCCEk.CFBME = 1, it becomes FIFO buffering mode, send data is stored in Transmit/Receive FIFO, and transmission is stopped. Transmission will be started if it is set as RSCFDnCFDCFCCEk.CFBME = 0.

Users should not write 1 from 0 for this bit when the RSCFDnCFDCFCCk.CFE bit is 1_B.

23.8 Interrupts and DMA

23.8.1 Interrupts

The RS-CANFD module generates several Interrupts.

The interrupt output, which is connected to the Interrupt Controller Unit, can be controlled by the corresponding interrupt enable bit.

The status flag will be set independent from this enable bit.

The channel Transmission Interrupt has an additional Status flag register; these Status bits will only be set when the corresponding interrupt enables are set.

This register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The Interrupts in the RS-CANFD module can be classified into 2 groups, Global Interrupts and Channel Interrupts:

Global Interrupts:

The RS-CANFD module can generate 2 Global Interrupts:

1. One Global Interrupt for successful reception into the 8 RX FIFO buffers
2. One Global Error Interrupt

Channel Interrupts:

Each channel of the RS-CANFD module can generate 3 Channel Interrupts:

1. Channel Transmission
2. Channel Error Interrupt
3. Successful Reception in a transmit/receive FIFO in RX or GW mode for a channel or Successful Reception in a TXQ

The interrupts are cleared when the corresponding flag bits are cleared or Interrupt enable bits are cleared.

If the set from the RS-CANFD module occurs simultaneously with the clear by the write access, then each flag bit is set. The set condition of the flag bits is prioritized.

Table 23.147 below gives an overview of interrupt sources for the different interrupt outputs.

The Interrupt outputs are Active High.

Table 23.147 Interrupt source overview (1/3)

	Interrupt terminal	Interrupt	Interrupt source	Interrupt clearing
Global Interrupts	can_rxf_int	Successful reception into at least one RX FIFO	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
		FIFO full into at least one RX FIFO	FIFO full interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the FIFO full interrupt flags of corresponding RX FIFO buffer for which interrupt is enabled
	can_glerr_int	Global Error	Any of the following: • DLC Error Flag • Message Lost Status bit • Message Over Write Status bit • TXQ Message Lost Status bit • TXQ Message Over Write Status bit • TX History Entry Lost Status bit • CANFD Message Payload overflow flag	Clear all of: • DLC Error Flag • Message Lost Flags in all of the FIFO Status Register • Message Overwrite Flags in all of the Transmit/Receive FIFO Status Register • Message Lost Flags in all of the TXQ Status Register • Message Overwrite Flags in all of the TXQ Status Register • TX History List Entry Lost Flag • CANFD Message Payload overflow flag

Table 23.147 Interrupt source overview (2/3)

	Interrupt terminal	Interrupt	Interrupt source	Interrupt clearing
Channel Transmission Interrupts	can_tx_int [m:0]	Channel m successful transmission	Any channel related TX MB Successful flag when Interrupt is enabled NOTE These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a Transmit/Receive FIFO. Separate interrupts are provided for Transmit/ Receive FIFO buffers & TX Queue	Clear all channel related TX MB Result status bits for which the Interrupt is enabled
		Channel m Abort	Any channel related TX MB Abort flag when Interrupt is enabled NOTE These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a Transmit/Receive FIFO. Separate interrupts are provided for Transmit/ Receive FIFO buffers & TX Queue	Clear all channel related TX MB Result Status bits for which the Interrupt is enabled globally
		Channel m transmission from TX Queue	Related channel TX Queue Interrupt Flag	Clear related channel TX Queue Interrupt Flag
		Channel m THL Interrupt	Channel m THL Interrupt status flag	Clear the relevant THL Interrupt status flag
		Channel m Transmit/ Receive FIFO TX Interrupt	Interrupt Flag for Transmit/Receive FIFOs in TX or GW mode belonging to the related channel	Clear the interrupt flags of Transmit/Receive FIFOs in TX or GW mode belonging to the related channel
		Channel m Transmit/ Receive FIFO One Frame TX Interrupt	One Frame Transmission Interrupt Flag for Transmit/Receive FIFOs in TX or GW mode belonging to the related channel	Clear the One Frame Transmission interrupt flags of Transmit/Receive FIFOs in TX or GW mode belonging to the related channel
		Channel m TXQ One Frame TX Interrupt	One Frame Transmission Interrupt Flag for TXQs belonging to the related channel	Clear the One Frame Transmission interrupt flags of TXQs belonging to the related channel

Table 23.147 Interrupt source overview (3/3)

	Interrupt terminal	Interrupt	Interrupt source	Interrupt clearing
Channel Transmit/ Receive FIFO RX Interrupt	can_com_frx_int[m:0]	Channel m Transmit/ Receive FIFO RX Interrupt	Interrupt Flag for Transmit/Receive FIFOs in RX or GW mode belonging to the related channel	Clear the interrupt flags of Transmit/Receive FIFOs in RX or GW mode belonging to the related channel
		Channel m Transmit/ Receive FIFO One Frame RX Interrupt	One Frame Reception Interrupt Flag for Transmit/ Receive FIFOs in RX or GW mode belonging to the related channel	Clear the One Frame Reception interrupt flags of Transmit/Receive FIFOs in RX or GW mode belonging to the related channel
		Channel m Transmit/ Receive FIFO Full Interrupt	FIFO Full Interrupt Flag for Transmit/Receive FIFOs in RX or GW mode belonging to the related channel	Clear the FIFO full interrupt flags of Transmit/Receive FIFOs in RX or GW mode belonging to the related channel
		Channel m TXQ One Frame RX Interrupt	One Frame Reception Interrupt Flag for TXQs in GW mode belonging to the related channel	Clear the One Frame Reception interrupt flags of TXQs in GW mode belonging to the related channel
		Channel m TXQ Full Interrupt	TXQ Full Interrupt Flag for TXQs in GW mode belonging to the related channel	Clear the TXQ full interrupt flags of TXQs in GW mode belonging to the related channel
Channel Error Interrupt	can_cherr_int[m:0]	Channel m Error	Any channel related error flag in the Channel Error Flag Register for which Interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which Interrupt is enabled in the Channel Error Interrupt Enable Register

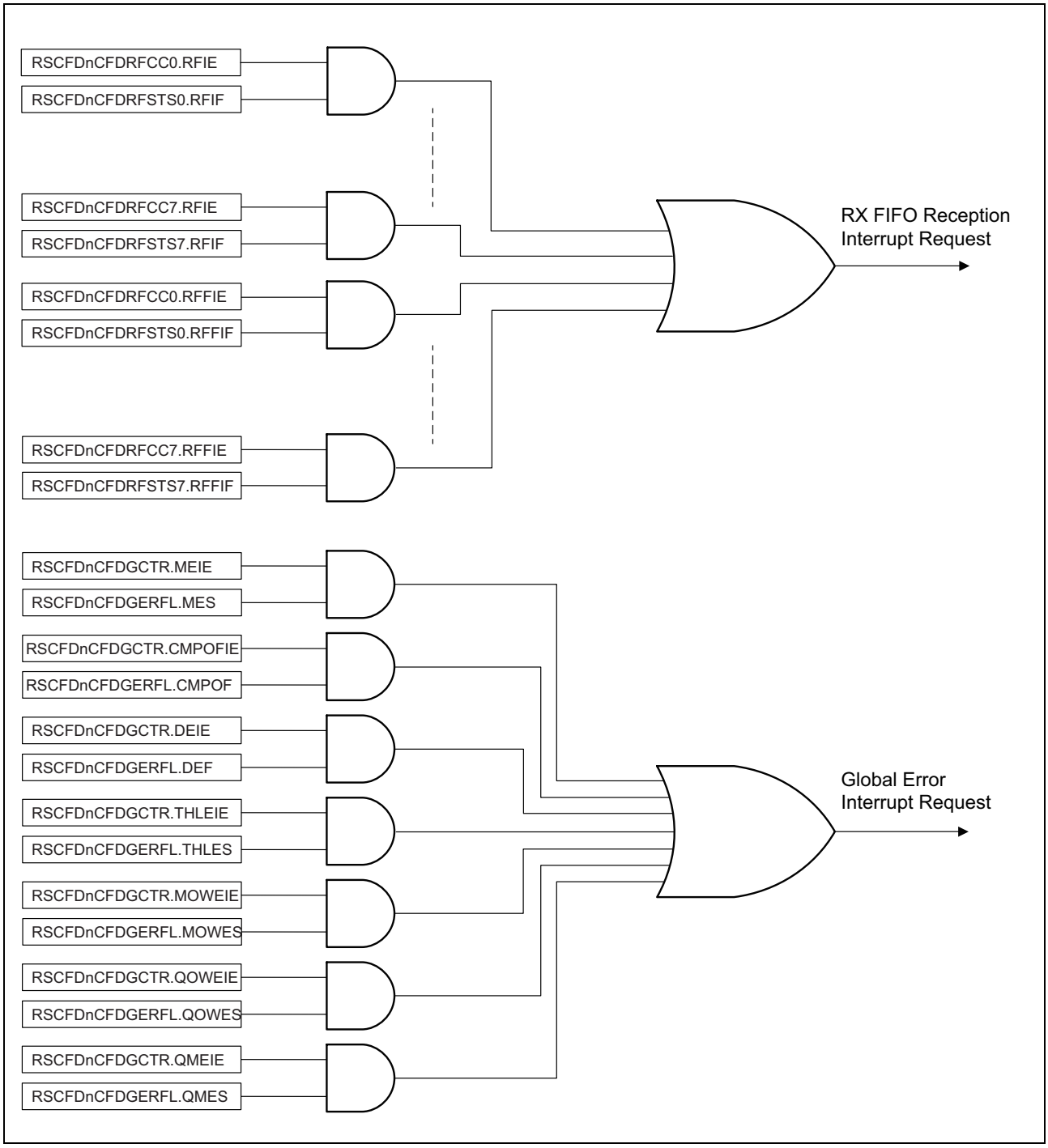


Figure 23.29 Global Interrupt Block Diagram

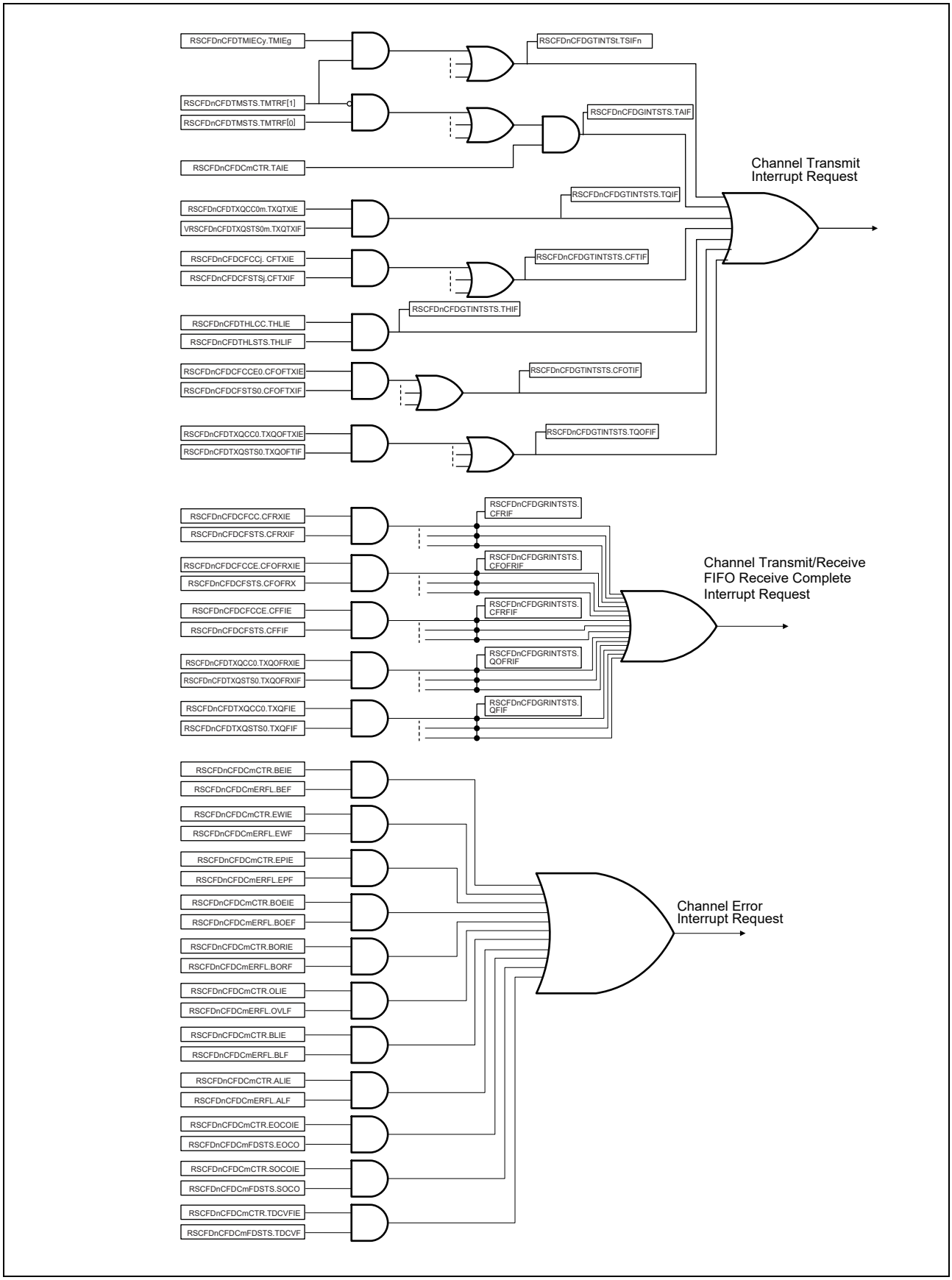


Figure 23.30 Channel Interrupt Block Diagram

23.8.2 DMA Transfer

The RS-CANFD module has some message buffer which can be associated with a DMA channels:

Reception DMA

- 8 RX FIFO Message Buffers
- 8 Transmit/Receive FIFO Message Buffers

Transmission DMA

- 2 x m TXQ Message Buffers (TXQ0, TXQ3)
- m Transmit/Receive FIFO Message Buffers

The figure below illustrates the potential DMA channels.

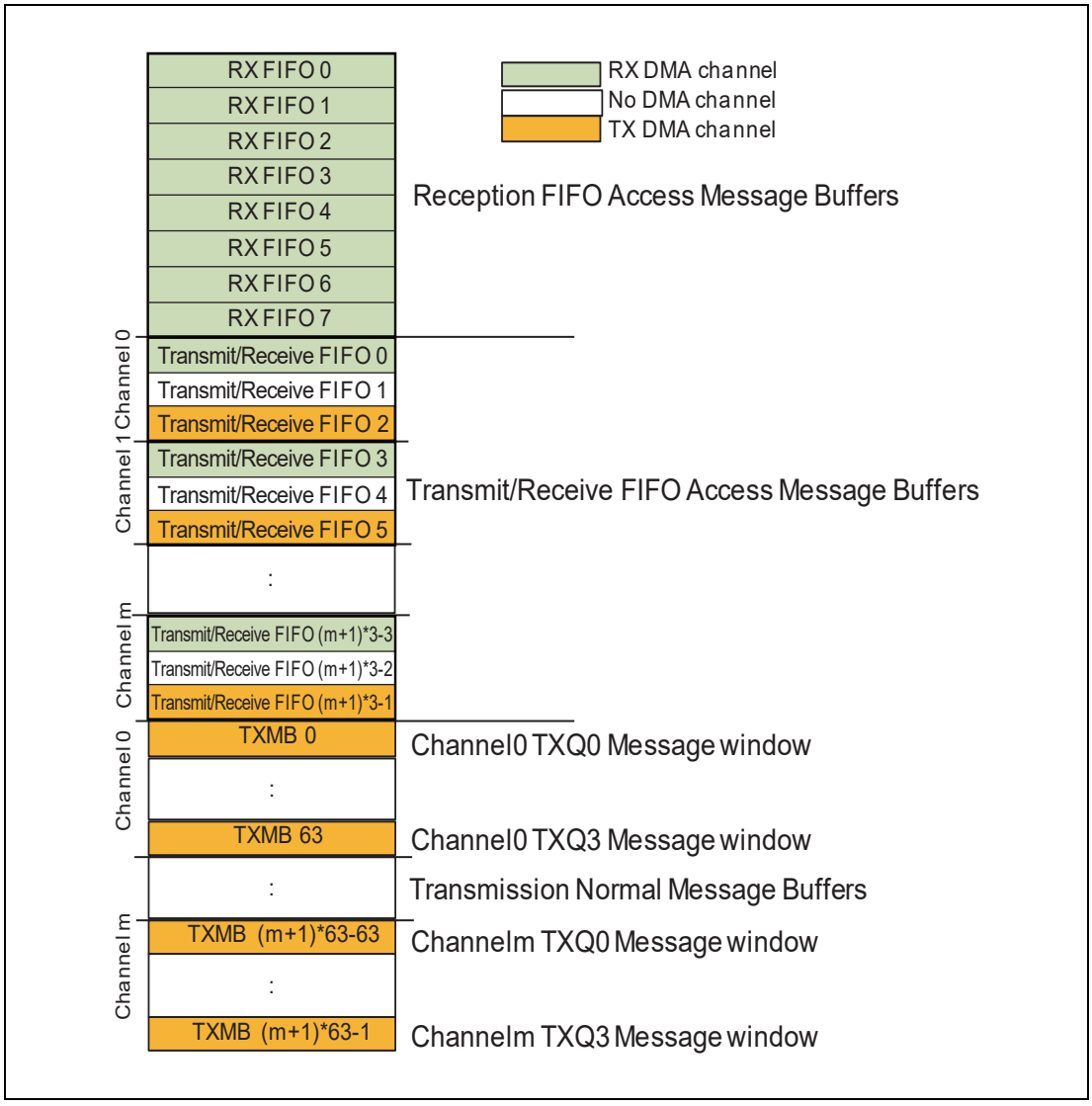


Figure 23.31 Message Buffer connectable to a DMA channel

A DMA channel transfer request will be generated for each FIFO entry to the DMAC when the related RSCFDnCFDCDTCT.RFDMAE or RSCFDnCFDCDTCT.CFDMAE is set to 1_B and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (RSCFDnCFDRFCCx.RFIE or RSCFDnCFDCFCCk.CFRXIE).

Users should use the regular start address for the DMA access window address. Refer to the table below.

Table 23.148 DMA channel access window address

b = Message Buffer Component Index	MBCP	Register	d	Regular Start Address m = [0…no_of_channels-1]
[0…no_of_RFMBCPs-1]	RFMBCPb	RFID	x	$6000_H + b\times0080_H$
		RFPTR	x	$6004_H + b\times0080_H$
		RFFDSTS	x	$6008_H + b\times0080_H$
		RFDFd	[0 to 15]	$600C_H + d\times0004_H + b\times0080_H$
[0…no_of_CFMBCPs_per_channel-1]	CFMBCPbm	CFID	x	$6400_H + b\times0080_H + m\times180_H$
		CFPTR	x	$6404_H + b\times0080_H + m\times180_H$
		CFFDCSTS	x	$6408_H + b\times0080_H + m\times180_H$
		RSCFDnCFDFd	[0 to 15]	$640C_H + d\times0004_H + b\times0080_H + m\times180_H$

DMA FIFO pointer decrement will be done automatically with reading the last configured data payload Byte (RSCFDnCFDRFCCx.RFPLS or RSCFDnCFDCFCCk.CFPLS).

Note: The DMA should read the exact length of the configured data payload size (RSCFDnCFDRFCCx.RFPLS or RSCFDnCFDCFCCk.CFPLS), no more, no less.

Users should not write to the FIFO and TXQ control registers when DMA is enabled.

The DMA enable of the particular DMA FIFO (RSCFDnCFDCDTCT.RFDMAE or RSCFDnCFDCDTCT.CFDMAE) can be set at any time, **Figure 23.32, DMA enable flow** is a configuration flow for an initial set-up.

When RSCFDnCFDCDTTCT.TQ0DMAE or RSCFDnCFDCDTTCT.TQ3DMAE or RSCFDnCFDCDTTCT.CFDMAE is set, the messages of the corresponding TXQ or Transmit/Receive FIFO can be handled by DMA controller.
Take the following procedure when the TXQ or the Transmit/Receive FIFO can be handled by DMA controller.

1. CPU checks the TXQ or the Transmit/Receive FIFO is not full.
2. When transmit data can be used, CPU enables DMA to set this data to Transmit/Receive FIFO or TXQ.
   When using Transmit/Receive FIFO, transmit data is write in RSCFDnCFDCFID, RSCFDnCFDCFPTR, RSCFDnCFDCFFDCSTS and RSCFDnCFDTMBCPbm register.
   When using TXQ, transmit data is write in RSCFDnCFDTMID, RSCFDnCFDTMPTR, RSCFDnCFDTMFDCTR and RSCFDnCFDTMDFd register.
3. In a case of the Transmit/Receive FIFO, Transmit/Receive FIFO pointer is incremented automatically when DMA controller writes the last data payload byte configured by RSCFDnCFDCFCCk.CFPLS.
   In the case of TXQ, if the data of 64 data payload is written, a TXQ pointer will increase automatically.
   When payload data is less than 64 byte, dummy data must be written in and 64 data payload size must be done.
   Note only 32-bit write-access can be possible on the DMA message handling.

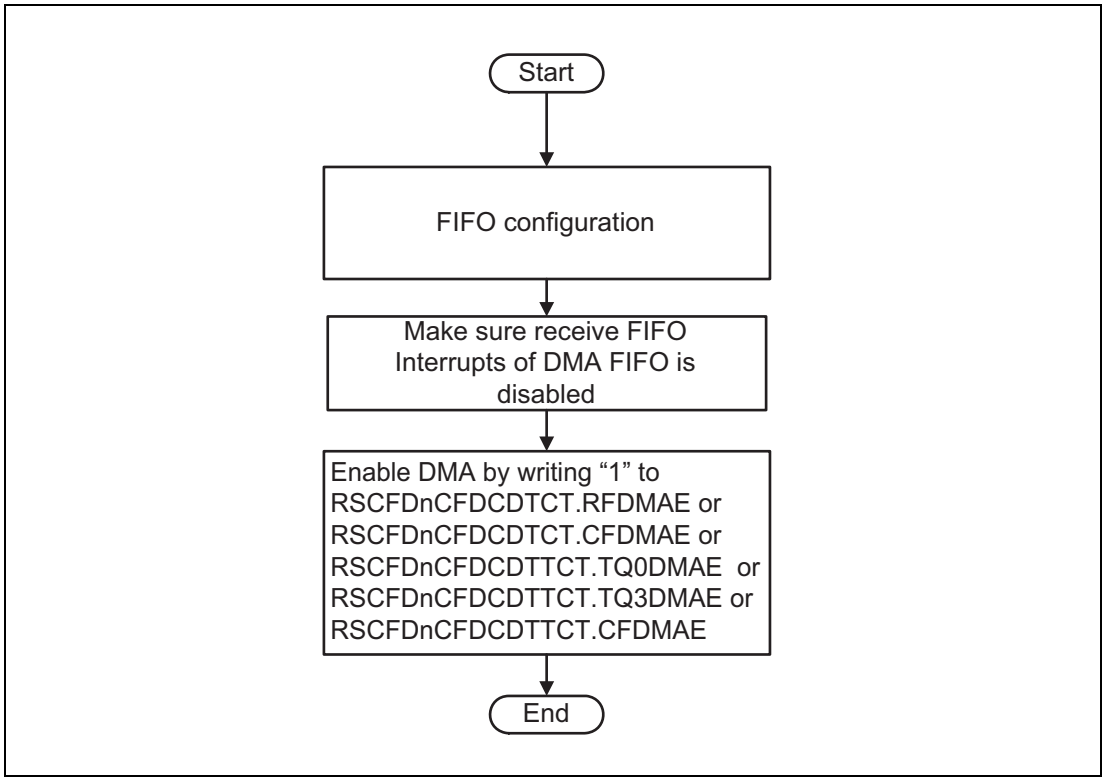


Figure 23.32 DMA enable flow

To disable a DMA transfer requested, disable the particular DMA enable bit (RSCFDnCFDCDTCT.RFDMAE or RSCFDnCFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer then this must be completed first before further action should be taken. The transfer status can be identified by the RSCFDnCFDCDTSTS.RFDMASTS or RSCFDnCFDCDTSTS.CFDMASTS. For reference see the flow below. When the DMA is disabled then consider what to do with the remaining or new incoming messages to this particular reception FIFO’s.

When the FIFO is not disabled then reception to the FIFO will continue.

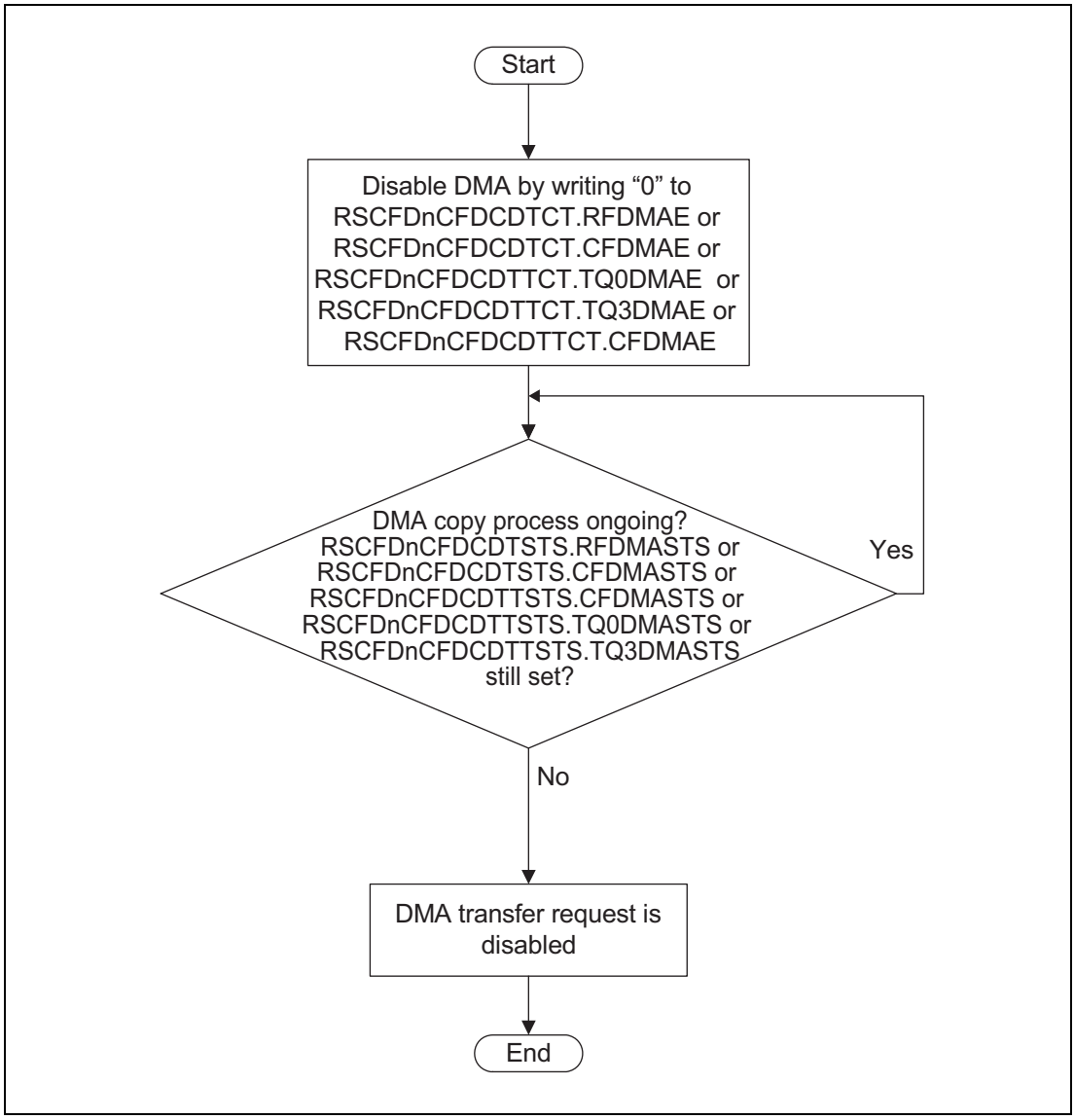


Figure 23.33 DMA disable flow

23.9 Reception and Transmission

23.9.1 Reception

In the RS-CANFD module, CAN messages received on any of the channels, will be stored in RX Message Buffers or in RX FIFO Buffers or Transmit/Receive FIFO Buffers configured in RX Mode or GW Mode depending upon the Acceptance Filter List entries:

- Up to ((m+1)×16) RX Message Buffers can be configured
- 8 RX FIFO Buffers available
- Up to ((m+1)×3) Transmit/Receive FIFO Buffers can be configured in RX mode or GW mode
- Up to ((m+1)×3) TX Queue can be configured in GW mode

23.9.1.1 Message storage in RX Message Buffers

When a message is successfully received and stored in a RX Message Buffer, the corresponding New data Flag is set in the RX Message Buffer New data Register.

The CAN Message can be read from the corresponding RX Message Buffer.

If a new message is stored into a RX Message Buffer before the previous message in this Message Buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX Message Buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

NOTES

1. Interrupts are not provided for the RX Message Buffers in the RS-CANFD module and hence the RX Message Buffer New data Register should be accessed periodically to check if a new message has been stored in the RX Message Buffers.
2. Unused Data Bytes will be filled with 0_H depending upon the DLC value.

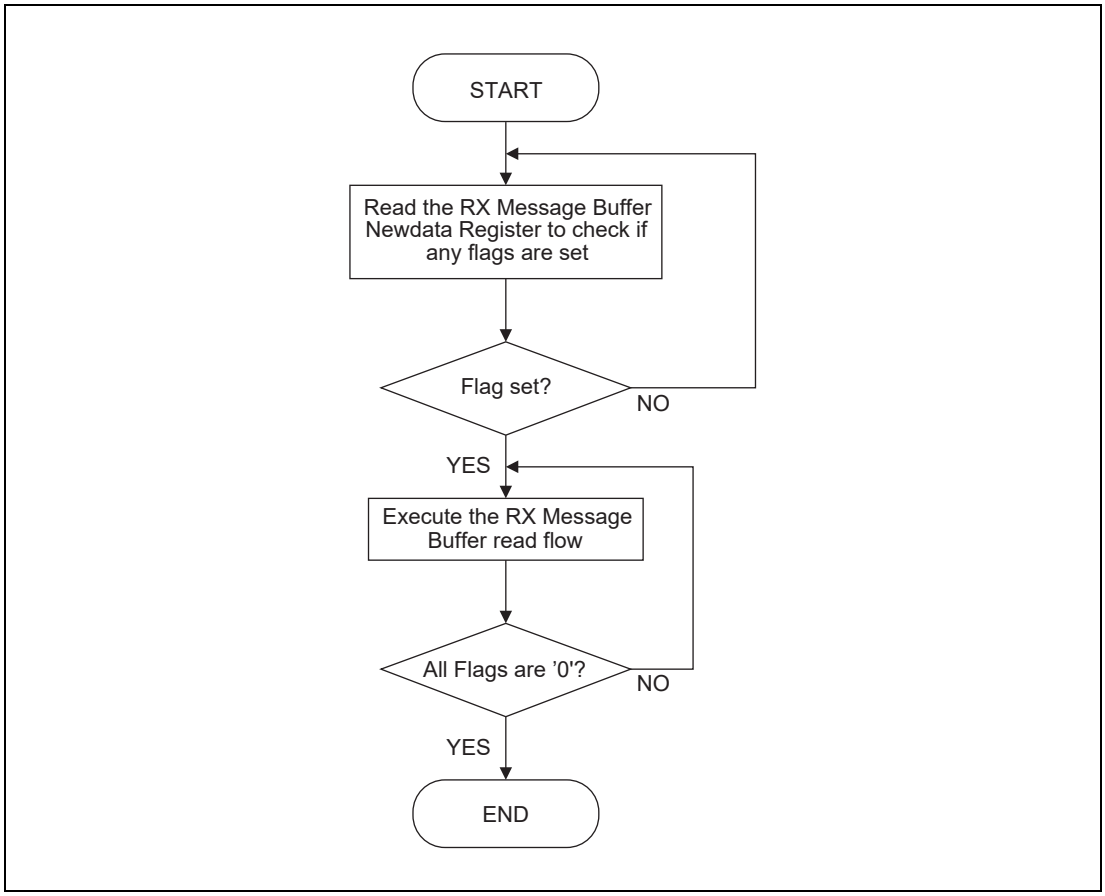


Figure 23.34 RX Message Buffer Message Access Flow

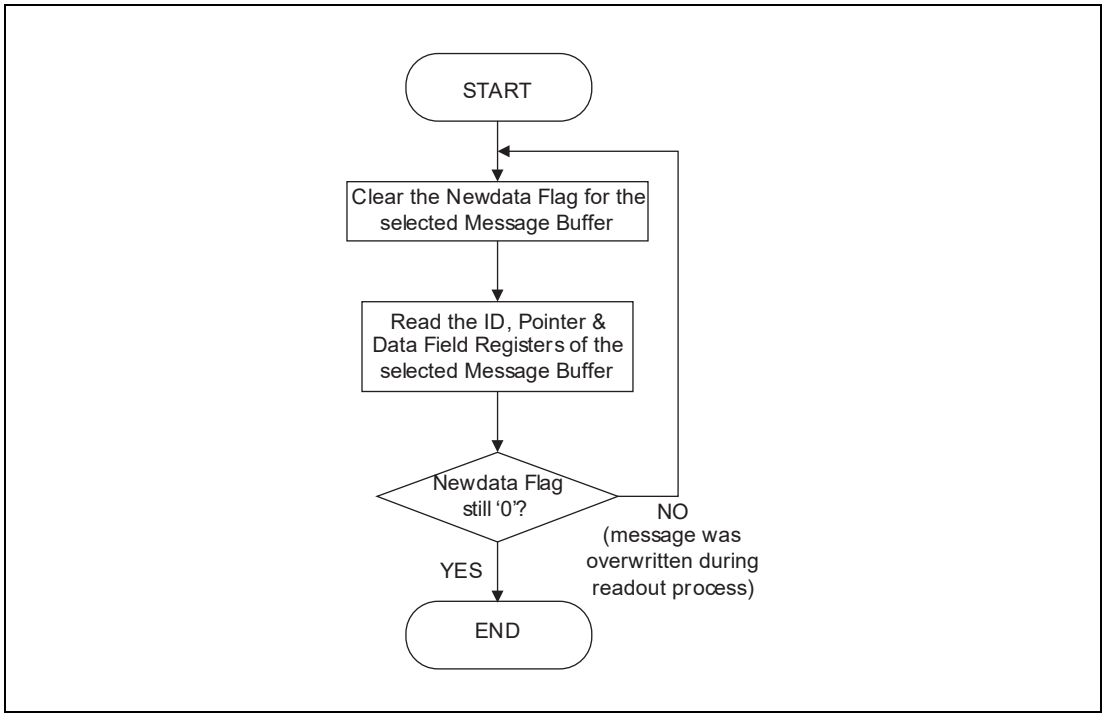


Figure 23.35 RX Message Buffer Read flow

23.9.1.2 Message storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Transmit/Receive FIFO Buffers configured in RX or GW Mode should be configured based on the requirements of the system.

The RSCFDnCFDGAFLP1j.GAFLFDP[31:0] field in the matching AFL entry selects the FIFO Buffers to which the related reception message will be stored.

When the received message is stored in one or more RX FIFO Buffers or Transmit/Receive FIFO Buffers configured in RX Mode or GW Mode, then the Message counter value is incremented in the corresponding RX FIFO Status Register or Transmit/Receive FIFO Status Register.

Depending upon the configuration of the FIFO Buffers, an Interrupt may also be generated.

The message can be read from the corresponding FIFO access registers.

NOTE

Since many messages can be stored in the FIFO Buffers, reading more than 1 message may be required to read the latest message stored in a FIFO Buffer.

If the Message count value matches the FIFO depth, then the FIFO Full Flag is set.

When the value FF_H is written to the corresponding FIFO Pointer Control Register, then the Message Count is decremented by 1.

Users should only write FF_H to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, then the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO Message count matches the FIFO Depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message will be lost (no overwrite of already stored messages will take place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a Message due to Overrun condition.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer will be overwritten with the message received or the message will be discarded. The behavior is determined by setting RSCFDnCFDCFCCEk.CFMOWM bit.

When RSCFDnCFDCFCCEk.CFMOWM = 0:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message will be discarded. And RSCFDnCFDCFSTSk.CFMLT bit is set to 1.

When RSCFDnCFDCFCCEk.CFMOWM = 1:

When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer will be overwritten with the received message. The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.
Then, RSCFDnCFDCFSTSk.CFMOW bit is set to 1, which notifies that the oldest message has

been overwritten with the received message.
In addition, in a case a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and re-transmission for the message is not performed. Then the read point moves to the next message automatically.

Users should not write change for this bit when the RSCFDnCFDCFCCk.CFE bit is 1_B.

Transmit/Receive FIFO can set interrupt, when CAN frame reception is completed.

Transmit/Receive FIFO can set interrupt, when FIFO is in full status in RX mode or GW mode.

NOTES

1. The Message Lost can be set only in RX or GW mode by CAN side, the flag will not be set when the CPU side is overloading the FIFO buffers.
2. When RSCFDnCFDGAFLP0j.GAFLSRDi (i = 0 to 2) is set and the RSCFDnCFDTXQCCim.TXQGWE (i = 0 to 2) is also set, a receiving frame is stored in the target TXQ as send data by routing.

The RX FIFO Buffers and the Transmit/Receive FIFO Buffers configured in RX or GW Mode can be disabled at any time by clearing the RSCFDnCFDRFCCx.RFE or RSCFDnCFDCFCCk.CFE bit in the RX FIFO Configuration / Control Register and the Transmit/Receive FIFO Configuration / Control Register.

When the RSCFDnCFDRFCCx.RFE or RSCFDnCFDCFCCk.CFE bit is cleared, then the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO Buffers will be lost and no further messages can be stored into the FIFO.

When the RX FIFO Buffers or Transmit/Receive FIFO Buffers configured in RX Mode is assigned as DMA channel then the SW should not access the FIFO Access Register of this FIFO buffer or write FF_H to the FIFO Pointer Control Register (RSCFDnCFDCFPCTRk.CFPC or RSCFDnCFDRFPCTRx.RFPC), because this could lead to unintended FIFO message decrement. The DMA channel will control the FIFO decrement by automatically.

NOTE

If the interrupt flag is set for a FIFO Buffer and then the FIFO is disabled, then the interrupt flag will not be cleared automatically. The interrupt flag should be cleared before disabling the FIFO.

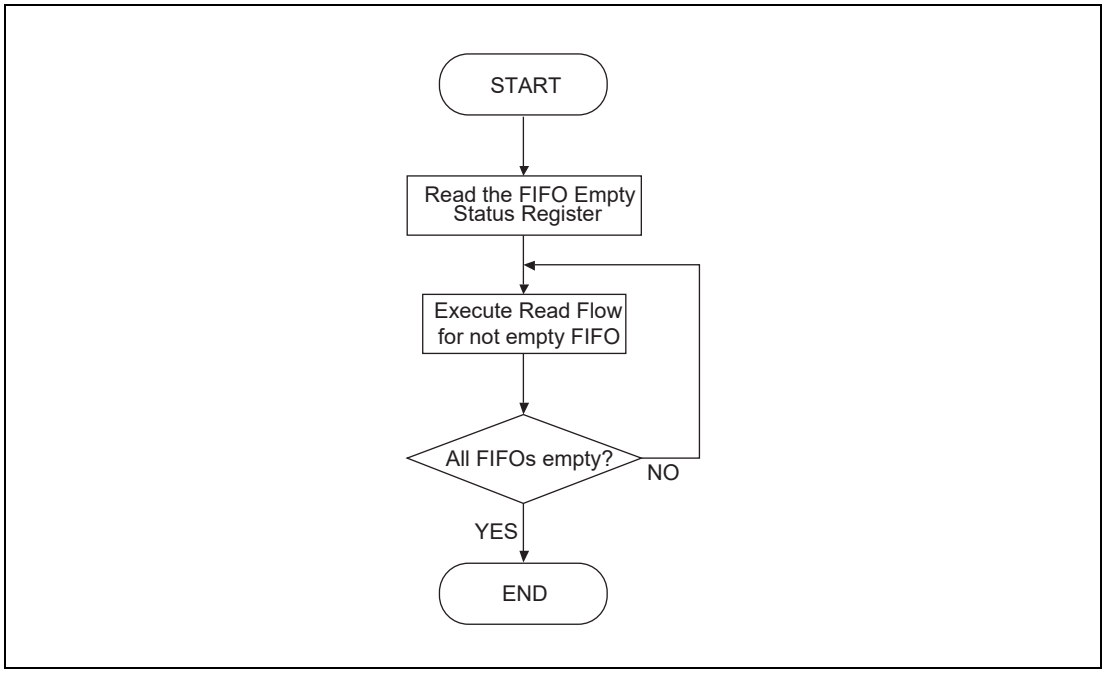


Figure 23.36 FIFO Buffer Message access Flow (example for polling case)

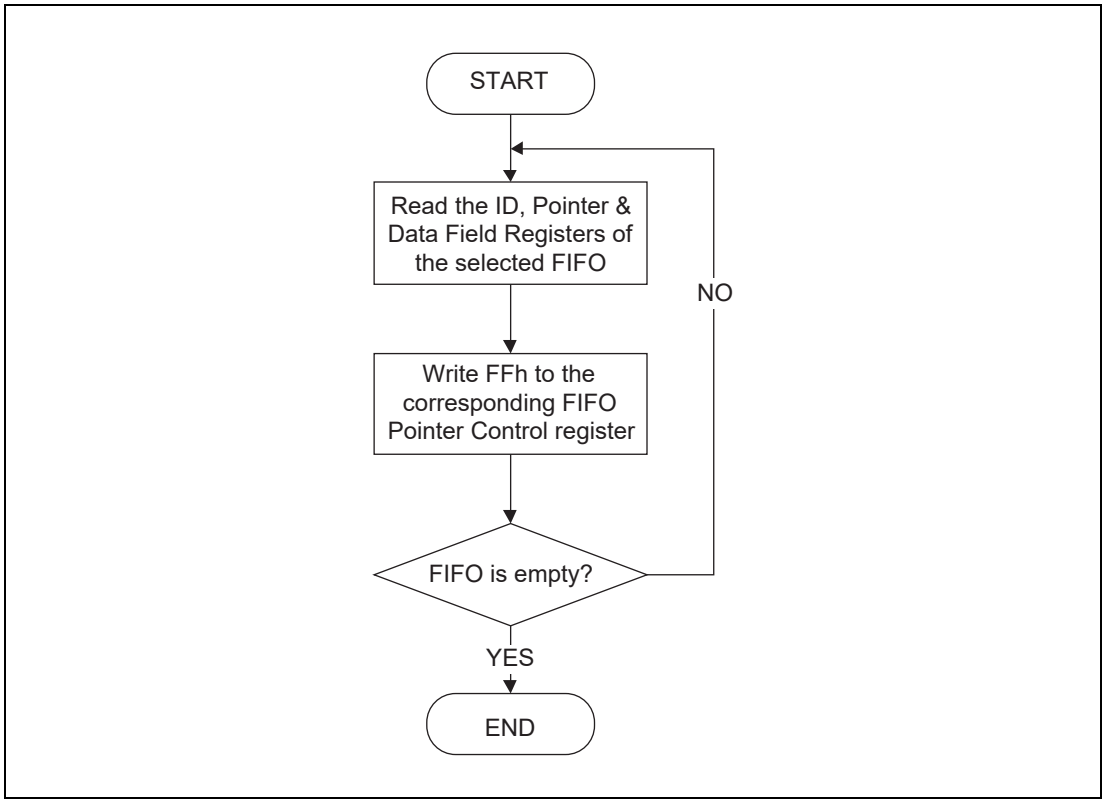


Figure 23.37 RX FIFO Buffer Read flow (example for polling case)

NOTE

When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even if it clears an "interruption flag" after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the deadline, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

23.9.1.3 Timestamp

The Timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value will be captured based on the RSCFDnCFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of Start of Frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception it is stored together with the message ID and Data into the target RX Message Buffer or RX/GW FIFO.

For transmit message the Timestamp counter value will be stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured via the RSCFDnCFDGCFG.TSSS bit of the Global Configuration Register. If it is 0_B, the peripheral clock is used. If it is 1_B, the selected CAN channel bit time clock is used.

The channel selection is done via the RSCFDnCFDGCFG.TSBTCS Bit of the Global Configuration Register.

Care has to be taken when using selected CAN channel bit time clock as clock source. In case of entering Channel Halt Mode or Channel Reset Mode, for this channel, the Timestamp counter is stopped. So, also for other CAN channels the Timestamp counter value will not be updated.

If peripheral clock is selected as Timestamp counter clock source Channel Modes are not influencing the Timestamp counter function.

The source clock for the Timestamp counter can be divided by a factor defined by the RSCFDnCFDGCFG.TSP bits (Timestamp Prescaler) in the Global Configuration Register.

The Timestamp counter can be reset to 0000_H via the RSCFDnCFDGCTR.TSRST bit (Timestamp Reset).

23.9.2 Transmission

There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- TX Queue transmission
  A fixed number of transmission Message Buffers 64 [for U2A-EVA, U2A16, U2A8] and 32 [for U2A6] TX Message Buffers are dedicated for each channel. These Message Buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue and/or Transmit/Receive FIFO in TX or GW mode can be configured in the following way (see **Figure 23.38, Channel Transmission Message Buffer Configuration** for an example for U2A-EVA, U2A16, U2A8):
  TX Queue: Up to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] transmission Message Buffers for one channel can be grouped to form a TX Queue with a common access window.
  Upper transmission Message Buffers are used to form the TXQ1 or TXQ3.
  Lower transmission Message Buffers are used to form the TXQ0 or TXQ2.
  Transmission control and status registers of these transmission Message Buffers should not be used.
  One Channel has four TX Queue.
  Each TXQ has each access window.
  - TXQ0 is transmission Message Buffer 0 of each channel.
  - TXQ1 is transmission Message Buffer 31 of each channel.
  - TXQ2 is transmission Message Buffer 32 of each channel.
  - TXQ3 is transmission Message Buffer 63 of each channel.

  When using TXQ1 and TXQ0 simultaneously, the sum of the depths of TXQ1 and TXQ0should not exceed 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6].When using TXQ3 and TXQ2 simultaneously, the sum of the depths of TXQ3 and TXQ2should not exceed 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6].
  Transmit/Receive FIFO (TX/GW mode): each Transmit/Receive FIFO in TX or GW mode is linked to a dedicated channel. Each channel has a fixed number of 3 Transmit/Receive FIFOs assigned to it. Within the channel, a Transmit/Receive FIFO configured in TX or GW mode, can be freely linked (assigned) between 32 and 63 [for U2A-EVA, U2A16, U2A8] and 47 [for U2A6] transmission Message Buffers (only one FIFO to one transmission Message Buffer).
  The Transmit/Receive FIFO Buffer then replaces the transmission Message Buffer linked to it.
  Transmission control and status registers of these transmission Message Buffers should not be used.

Refer to **Figure 23.27, Message Buffer Configuration** for information about Transmit/Receive FIFO Buffer assignment to related channels.

NOTE

Transmit/Receive FIFO buffers should not be linked to TX Message Buffers that are already part of a TX Queue.

Tx Message Buffer0	Tx Message Buffer0	Tx Queue 0	Tx Queue 0
Tx Message Buffer1	Tx Message Buffer1	Tx Queue 0	Tx Queue 0
Tx Message Buffer2	Tx Message Buffer2	Tx Queue 0	Tx Queue 0
Tx Message Buffer3	Tx Message Buffer3	Tx Queue 0	Tx Queue 0
Tx Message Buffer4	Tx Message Buffer4	Tx Queue 0	Tx Queue 0
Tx Message Buffer5	Tx Message Buffer5	Tx Queue 0	Tx Queue 0
Tx Message Buffer6	Tx Message Buffer6	Tx Queue 0	Tx Queue 0
Tx Message Buffer7	Tx Message Buffer7	Tx Queue 0	Tx Queue 0
Tx Message Buffer8	Tx Message Buffer8	Tx Queue 0	Tx Queue 0
Tx Message Buffer9	Tx Message Buffer9	Tx Queue 0	Tx Queue 0
Tx Message Buffer10	Tx Message Buffer10	Tx Queue 0	Tx Queue 0
Tx Message Buffer11	Tx Message Buffer11	Tx Queue 0	Tx Queue 0
Tx Message Buffer12	Tx Message Buffer12	Tx Queue 0	Tx Queue 0
Tx Message Buffer13	Tx Message Buffer13	Tx Queue 0	Tx Queue 0
Tx Message Buffer14	Tx Message Buffer14	Tx Queue 0	Tx Queue 0
Tx Message Buffer15	Tx Message Buffer15	Tx Queue 0	Tx Queue 0
Tx Message Buffer16	Tx Message Buffer16	Tx Message Buffer16	Tx Queue 0
Tx Message Buffer17	Tx Message Buffer17	Tx Message Buffer17	Tx Queue 0
Tx Message Buffer18	Tx Message Buffer18	Tx Message Buffer18	Tx Queue 0
Tx Message Buffer19	Tx Message Buffer19	Tx Message Buffer19	Tx Queue 0
Tx Message Buffer20	Tx Message Buffer20	Tx Message Buffer20	Tx Queue 0
Tx Message Buffer21	Tx Message Buffer21	Tx Message Buffer21	Tx Queue 0
Tx Message Buffer22	Tx Message Buffer22	Tx Message Buffer22	Tx Queue 0
Tx Message Buffer23	Tx Message Buffer23	Tx Message Buffer23	Tx Queue 0
Tx Message Buffer24	Tx Message Buffer24	Tx Queue 1	Tx Queue 0
Tx Message Buffer25	Tx Message Buffer25	Tx Queue 1	Tx Queue 0
Tx Message Buffer26	Tx Message Buffer26	Tx Queue 1	Tx Queue 0
Tx Message Buffer27	Tx Message Buffer27	Tx Queue 1	Tx Queue 0
Tx Message Buffer28	Tx Message Buffer28	Tx Queue 1	Tx Queue 0
Tx Message Buffer29	Tx Message Buffer29	Tx Queue 1	Tx Queue 0
Tx Message Buffer30	Tx Message Buffer30	Tx Queue 1	Tx Queue 0
Tx Message Buffer31	Tx Message Buffer31	Tx Queue 1	Tx Queue 0
Tx Message Buffer32	Tx Message Buffer32	Ch. Transmit/Receive FIFO 0	Tx Queue 2
Tx Message Buffer33	Ch. Transmit/Receive FIFO 0	Tx Message Buffer33	Tx Queue 2
Tx Message Buffer34	Tx Message Buffer34	Tx Message Buffer34	Tx Queue 2
Tx Message Buffer35	Ch. Transmit/Receive FIFO 1	Tx Message Buffer35	Tx Queue 2
Tx Message Buffer36	Tx Message Buffer36	Tx Message Buffer36	Tx Queue 2
Tx Message Buffer37	Tx Message Buffer37	Ch. Transmit/Receive FIFO 1	Tx Queue 2
Tx Message Buffer38	Tx Message Buffer38	Tx Message Buffer38	Tx Queue 2
Tx Message Buffer39	Tx Message Buffer39	Tx Message Buffer39	Tx Queue 2
Tx Message Buffer40	Tx Message Buffer40	Tx Message Buffer40	Tx Queue 2
Tx Message Buffer41	Tx Message Buffer41	Tx Message Buffer41	Tx Queue 2
Tx Message Buffer42	Tx Message Buffer42	Tx Message Buffer42	Tx Queue 2
Tx Message Buffer43	Tx Message Buffer43	Ch. Transmit/Receive FIFO 2	Tx Queue 2
Tx Message Buffer44	Tx Message Buffer44	Tx Message Buffer44	Tx Queue 2
Tx Message Buffer45	Tx Message Buffer45	Tx Message Buffer45	Tx Queue 2
Tx Message Buffer46	Tx Message Buffer46	Tx Message Buffer46	Tx Queue 2
Tx Message Buffer47	Tx Message Buffer47	Tx Message Buffer47	Tx Queue 2
Tx Message Buffer48	Tx Message Buffer48	Tx Message Buffer48	Tx Queue 3
Tx Message Buffer49	Tx Message Buffer49	Tx Message Buffer49	Tx Queue 3
Tx Message Buffer50	Tx Message Buffer50	Tx Message Buffer50	Tx Queue 3

Figure 23.38 Channel Transmission Message Buffer Configuration (1/2)

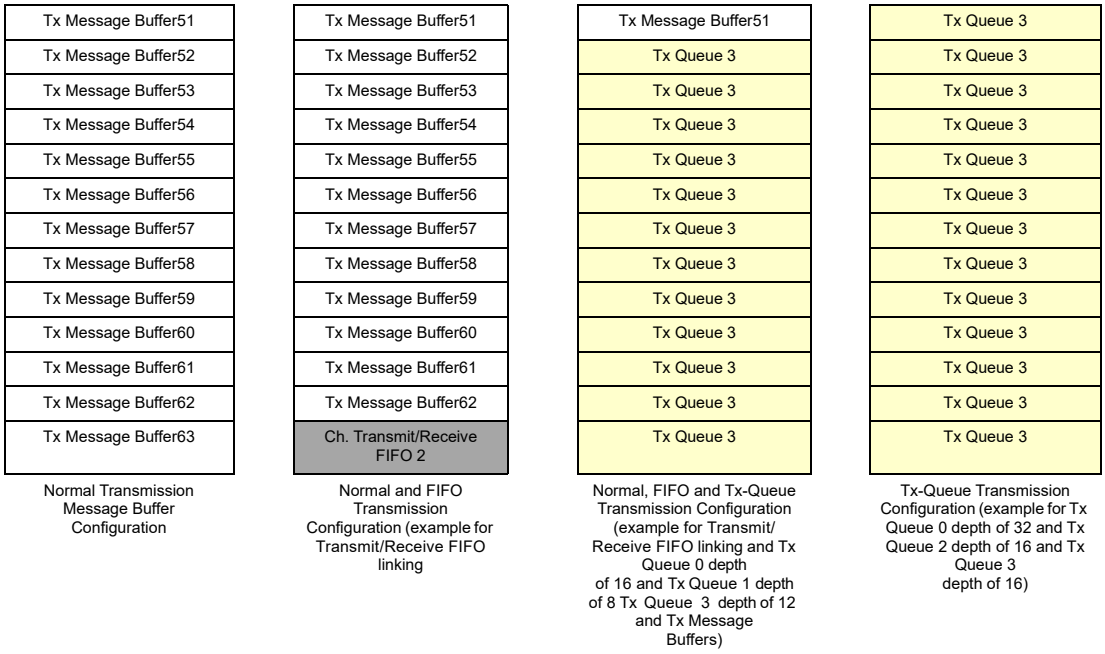


Figure 23.38 Channel Transmission Message Buffer Configuration (2/2)

23.9.2.1 Transmission Priority

If two or more transmission Message Buffers of a channel are configured for transmission, then the transmission priority in the RS-CANFD module can be selected from the following two modes:

- CAN ID priority
- Message Buffer number priority

The transmission priority mode is common for all Message Buffers and all CAN channels. It can be configured via the RSCFDnCFDGCFG.TPRI bit in the Global Configuration Register.

For Message Buffer number priority transmission, the smallest Message Buffer number with transmission request has the highest priority for transmission. This also includes the TX Message Buffers linked to the Transmit/Receive FIFO Buffers configured in TX mode or GW mode.

However, Message Buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 (2015) specification). All TX Message Buffers can enter the ID priority comparison for Message Buffers configured for transmission. This also includes the TX Message Buffers linked to the Transmit/Receive FIFO Buffers configured in TX mode or GW mode and includes the TX Queue Message Buffers.

If the ID of two or more Message Buffers is the same, then the smaller Message Buffer number will have higher priority for transmission.

Note: For Transmit/Receive FIFO Buffers configured in TX mode or GW mode, only the message currently being pointed to by the FIFO Read Pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending Message within the same FIFO will be considered in the transmission arbitration.

In contrast to this, all transmission Message Buffers of a TX Queue will participate in internal transmission arbitration.

Figure 23.39 below shows the transmission configuration flow.

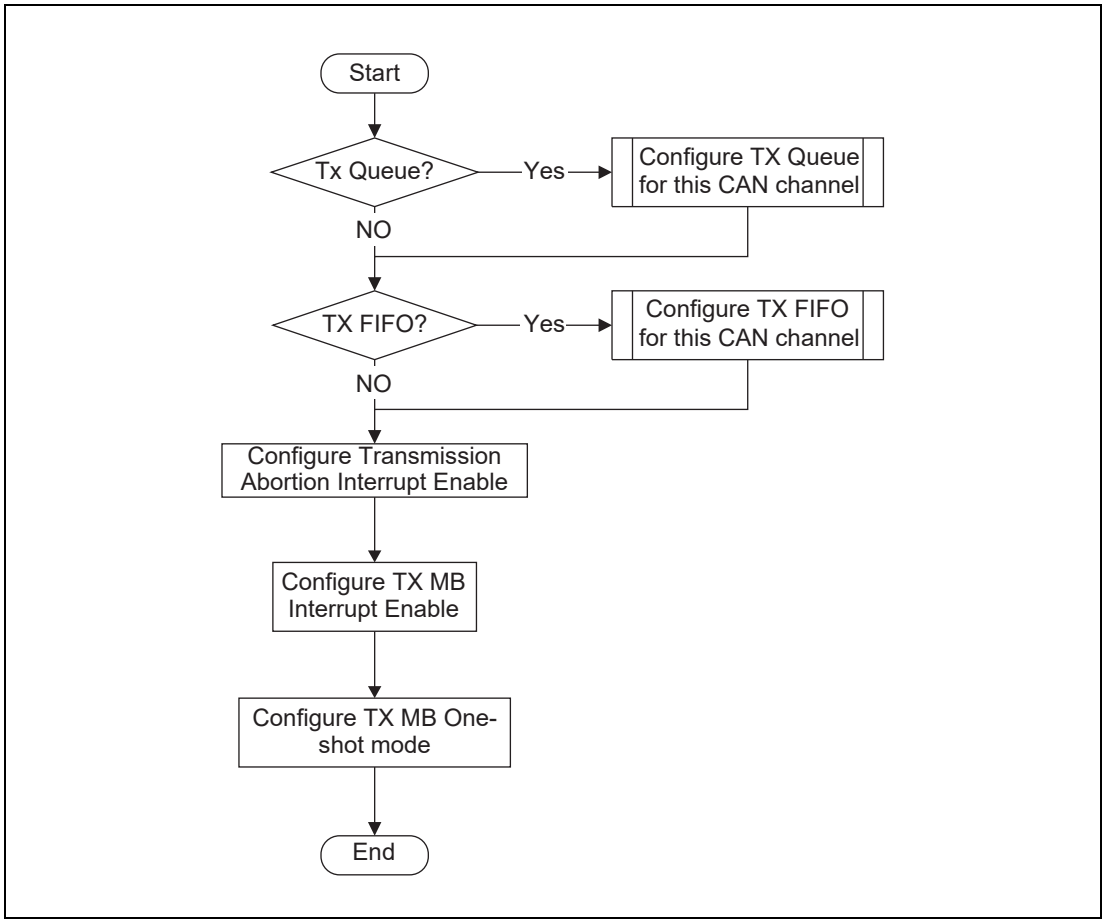


Figure 23.39 Transmission Configuration Flow

23.9.2.2 Normal Transmission

Each transmission Message Buffer has two modes of message transmission:

1. Regular Transmission Mode
   If the Message Buffer is placed in regular transmission mode, the data frame or remote frame set in that Message Buffer can be transmitted.
   Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (RSCFDnCFDTMSTSp.TMTRF) in the TX Message Buffer Status Register. These bits are set to 10_B or 11_B when the regular transmission is successful.
   When arbitration is lost or an error occurs, message transmission will be attempted further if no transmission abort request is set for this transmission Message Buffer.
   New internal transmission arbitration for this channel will be performed considering all Message Buffers with transmission request.

2. One Shot Transmission Mode
   When the RSCFDnCFDTMCp.TMOM bit of the TX Message Buffer Control Register is set for a transmission Message Buffer, then the Message Buffer is placed in one-shot transmission mode and attempts to transmit a message only once.
   Completion of one shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (RSCFDnCFDTMSTSp.TMTRF) in the TX Message Buffer Status Register. The RSCFDnCFDTMSTSp.TMTRF bits are set to 10_B or 11_B when the one shot transmission is successful.
   The RSCFDnCFDTMSTSp.TMTRF bits are set to 01_B when arbitration is lost or an error occurs

during the transmission of the related Message Buffer.
Further message transmission will not be attempted in this case.
The regular transmission request procedure after a configuration is shown in **Figure 23.40**.

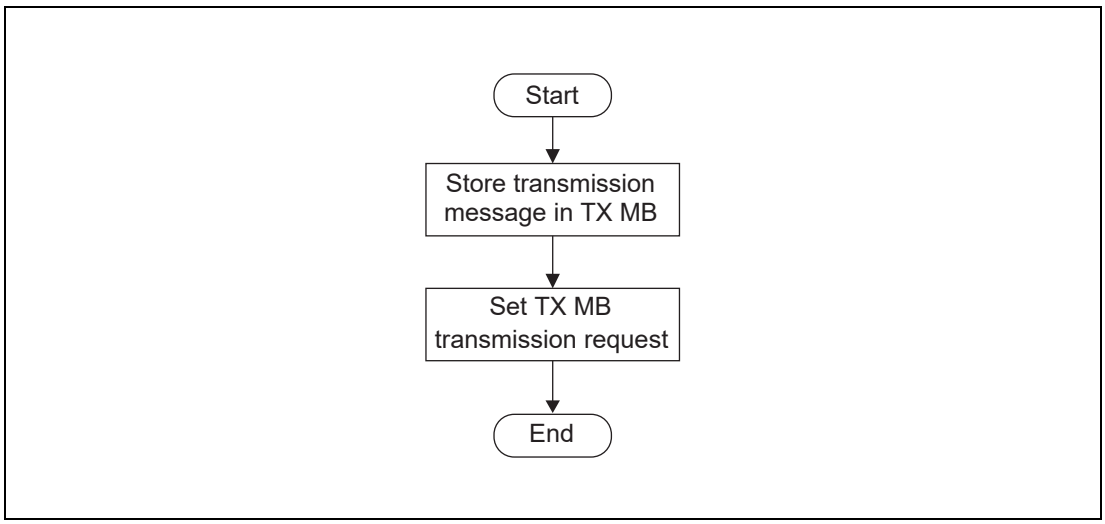


Figure 23.40 Transmission request procedure using normal TX Message Buffer mode

(1) TX Message Buffer control register setting

Table 23.149 shows configuration of the normal CAN transmission mode.

Table 23.149 Configuration of CAN transmission mode

Transmission Request RSCFDnCFDTMCp.TMTR	Transmission Abortion Request RSCFDnCFDTMCp.TMTAR	One Shot Enable RSCFDnCFDTMCp.TMOM	Communication activity
0	0	0	Message Buffer disabled
0	0	1	Message Buffer disabled
1	0	0	Configured as a transmission Message Buffer for a data frame or a remote frame
1	0	1	Configured as a one shot transmission Message Buffer for a data frame or a remote frame
1	1	0	Transmission abortion requested
1	1	1	One shot transmission abortion requested

The Configuration bits can be configured in the TX Message Buffer Control Register.

The **Figure 23.41** shows timings for successful transmission for two Message Buffers of one channel.

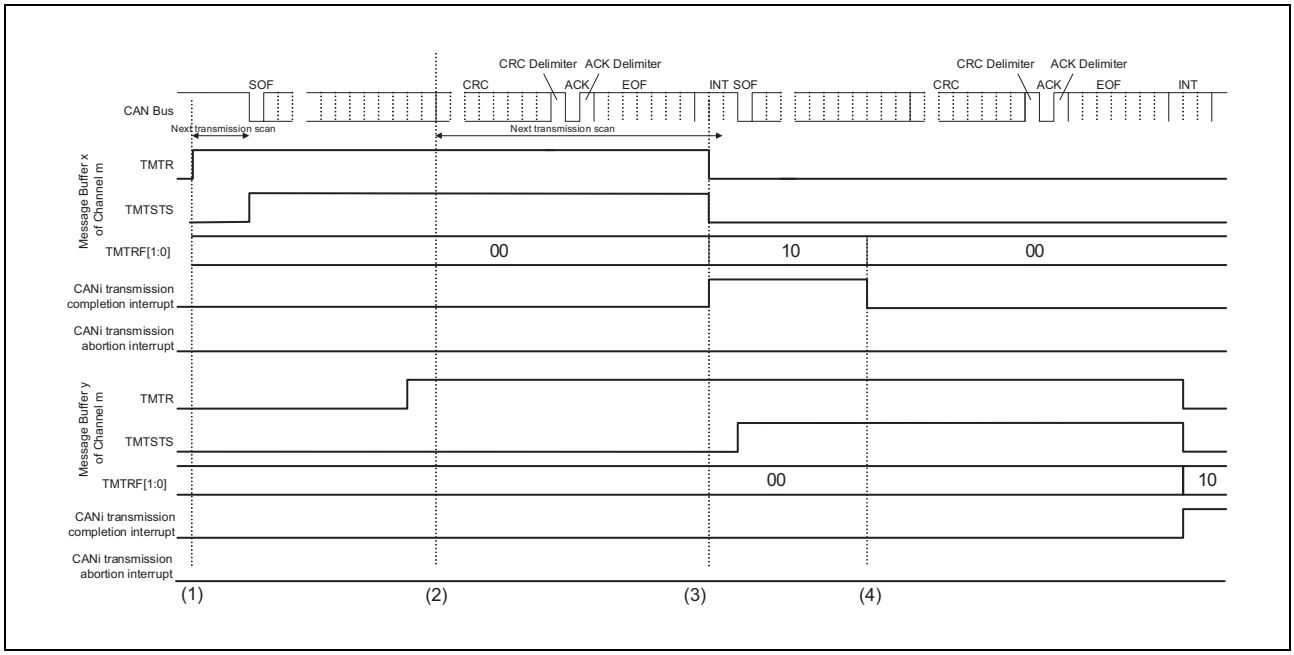


Figure 23.41 Timing of request and flag bits for successful transmission

(1) If the RSCFDnCFDTMCp.TMTR bit in the TX Message Buffer Control Register is set in the bus idle state, Message Buffer scanning procedure starts to decide the highest priority Message Buffer for transmission.
When the transmission Message Buffer is decided, the RSCFDnCFDTMSTSp.TMTSTS bit in the related TX Message Buffer Status Register is set (Transmitting/Transmitter), and CAN channel starts the transmission*1.

(2) At 1st Bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
The scan time could be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.

(3) If the message has been successfully transmitted, the RSCFDnCFDTMSTSp.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Register are set to 10_B and RSCFDnCFDTMSTSp.TMTSTS and the RSCFDnCFDTMCp.TMTR bits are cleared.
When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Register is set (Interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the RSCFDnCFDTMSTSp.TMTRF flag bits have to be cleared.

(4) Before starting the next transmission, clear the RSCFDnCFDTMSTSp.TMTRF bits. Load the next message in the transmission Message Buffer and set the RSCFDnCFDTMCp.TMTR bit again.
RSCFDnCFDTMCp.TMTR bit cannot be set again before RSCFDnCFDTMSTSp.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the RSCFDnCFDTMSTSp.TMTSTS bit is cleared.
The transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer from the beginning of the 1st CRC bit.
If an error occurs either during the transmission or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

NOTE

The setting point of RSCFDnCFDTMSTSp.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

The **Figure 23.42** shows timings for transmission abort for two Message Buffers of one channel.

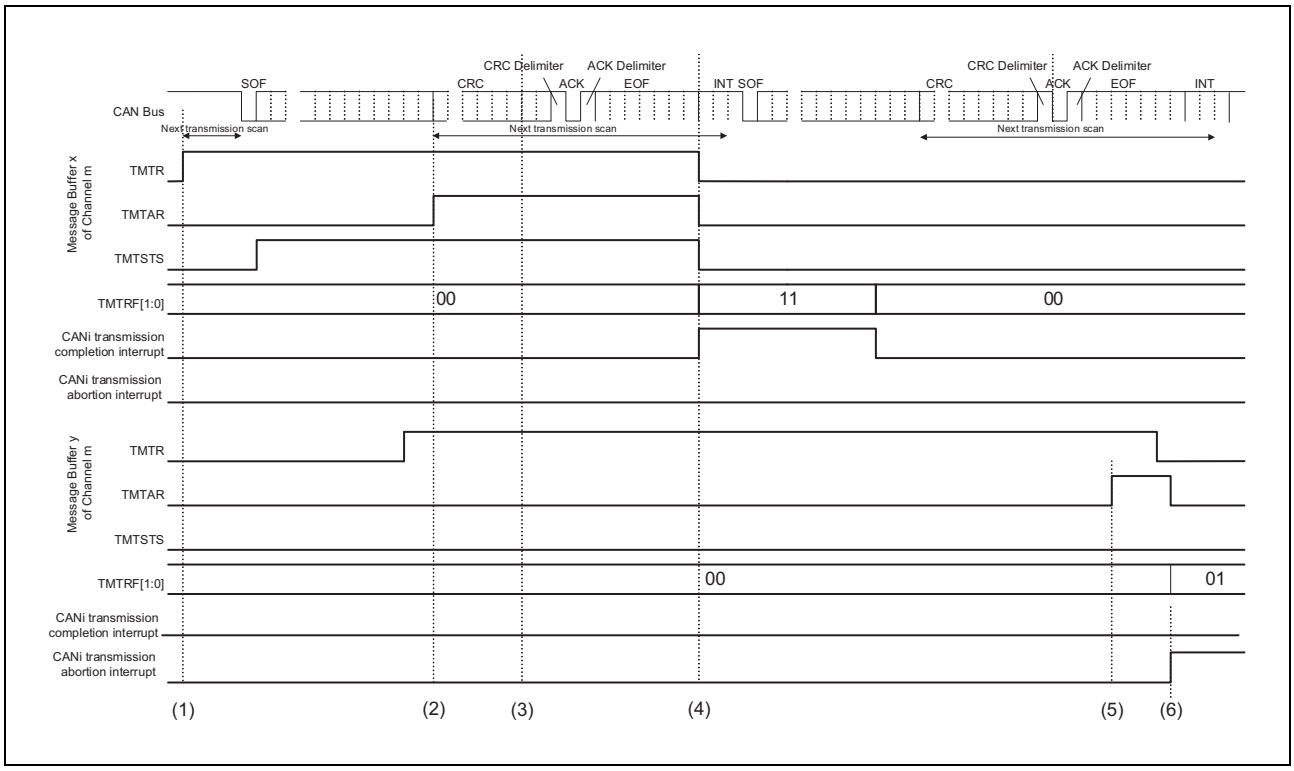


Figure 23.42 Timing of request and flag bits for transmission abort

(1) If the RSCFDnCFDTMCp.TMTR bit in the TX Message Buffer Control Register is set in the bus idle state, Message Buffer scanning procedure starts to decide the highest priority Message Buffer for transmission.
When the transmission Message Buffer is decided, the RSCFDnCFDTMSTSp.TMTSTS bit in the TX Message Buffer Status Register is set (Transmitting/Transmitter), and CAN channel starts the transmission*1.

(2) If the RSCFDnCFDTMCp.TMTAR bit is set when the related Message Buffer is already selected for transmission or currently transmitting then the message will not be aborted, if no error occurs or arbitration is lost.

(3) At 1st CRC bit, the transmission scanning procedure starts for the next transmission. In this example timing chart Message Buffer y is not selected as next transmission Message Buffer. The scan time could be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.

(4) If the message has been successfully transmitted, the RSCFDnCFDTMSTSp.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Register is set to 11_B and RSCFDnCFDTMSTSp.TMTSTS and the RSCFDnCFDTMCp.TMTR bits are cleared.
When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Register is set

(Interrupt enabled), the CAN successful transmission interrupt request is generated.
To clear the related interrupt line the RSCFDnCFDTMSTSp.TMTRF[1:0] bits has to be cleared.

(5) Another CAN node is transmitting on the CAN bus (RSCFDnCFDTMSTSp.TMTSTS not set!): if the RSCFDnCFDTMCp.TMTAR bit is set when the related channel is under transmission scan then the transmission request cannot be cleared.

(6) After internal processing time the transmission is aborted and the RSCFDnCFDTMSTSp.TMTRF[1:0] bits are set to 01_B.
If the Message Buffer is not transmitting or selected as next transmission Message Buffer or under transmit scan, then the abort is immediately accepted and the corresponding RSCFDnCFDTMSTSp.TMTRF[1:0] bits in the TX Message Buffer Status Register is set to 01_B.
In addition, RSCFDnCFDTMCp.TMTR, and RSCFDnCFDTMCp.TMTAR bits are cleared automatically.
When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort.
To clear the related interrupt line the RSCFDnCFDTMSTSp.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the RSCFDnCFDTMSTSp.TMTSTS is cleared.
The transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer from the beginning of the 1st CRC bit.
If an error occurs, either during the transmission, or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

23.9.2.3 TX FIFO or GW FIFO Transmission

Three Transmit/Receive FIFO buffers are assigned to each channel. The 3 FIFO buffers could be linked to any normal TX Message Buffer position for this channel by the RSCFDnCFDCFCCk.CFTML bits in the Transmit/Receive FIFO Configuration / Control Register if configured in TX or GW mode.

When the transmission scan starts and the FIFO Buffer corresponding to this TX Message Buffer is enabled, then the relevant message in the FIFO Buffer will participate in the transmission scan.

Configuration of a TX Message Buffer linked to a FIFO Buffer configured in TX or GW mode should not be done.

(1) TX FIFO Operation

CAN Messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value FFH is written into the corresponding FIFO Pointer Control Register, then the Message Count of the related FIFO is incremented by 1.

Users should only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers.

If the Message count matches the FIFO Depth, then the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding RS-CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the Message Count value is decremented by 1.

When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The Interrupt generation conditions for the TX FIFO buffers can be configured by configuring the RSCFDnCFDCFCCk.CFIM bit in the corresponding Transmit/Receive FIFO Configuration / Control Register.

If RSCFDnCFDCFCCk.CFIM bit is 0_B, then interrupt is generated when last message is successfully transmitted from the TX FIFO buffer.

If RSCFDnCFDCFCCk.CFIM bit is 1_B, then interrupt is generated for every successfully transmitted message from the TX FIFO Buffer.

Transmit/Receive FIFO can set interrupt, when CAN frame transmitted is completed.

The Transmit/Receive FIFO Buffers configured in TX Mode can be disabled by clearing the RSCFDnCFDCFCCk.CFE bit in the Transmit/Receive FIFO Configuration / Control Register. If this bit is cleared to 0_B, the FIFO empty flag is set as described below:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Test Mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission

NOTE

The Transmit/Receive FIFO buffer is considered as disabled after clearing the RSCFDnCFDCFCCk.CFE bit only when the Empty flag is set for the corresponding Transmit/ Receive FIFO Buffer.

Other possible messages pending from the TX FIFO are lost and their transmission needs to be requested again. Before RSCFDnCFDCFCCk.CFE is set again ensure that RSCFDnCFDCFSTSk.CFEMP bit is set and that there is no pending abort from the TX FIFO.

When the RSCFDnCFDCFCCk.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Hence, all messages in the FIFO buffers will be lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after a configuration is shown in **Figure 23.43**.

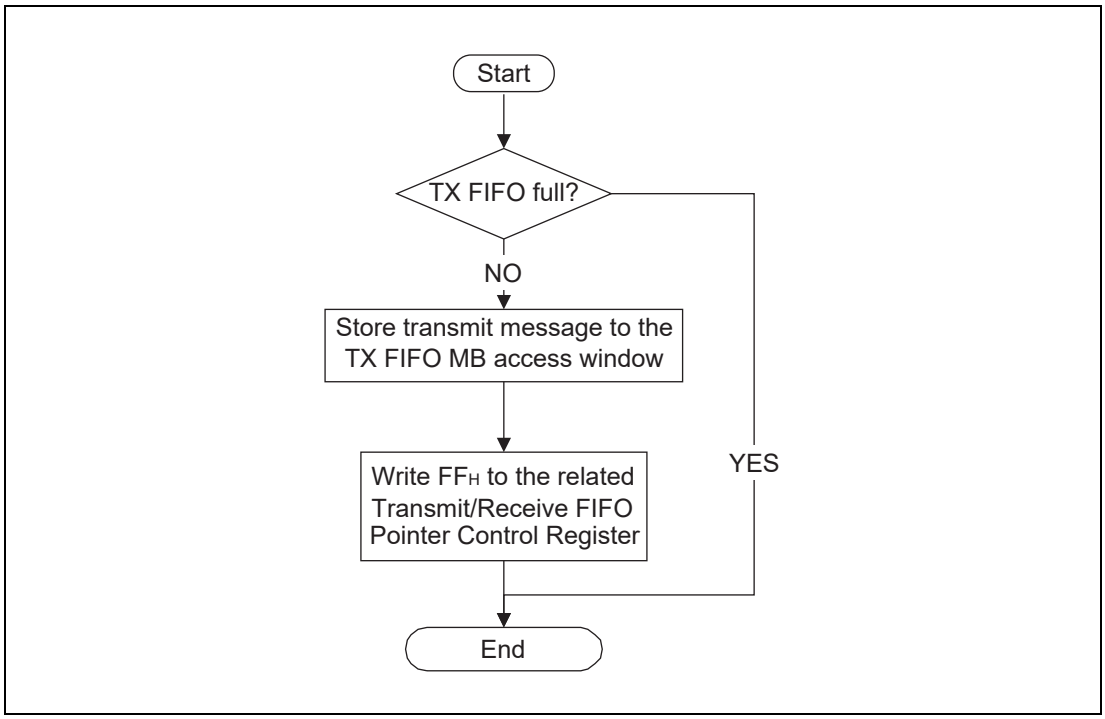


Figure 23.43 TX FIFO transmission request procedure

(2) GW FIFO Operation

The AFL entries for routing the received messages to GW FIFO buffers should be configured based on the requirements of the system. The matching AFL entry selects the GW FIFO Buffer for storage of a received message on any of the CAN channels.

When a message is successfully received and stored in a GW FIFO Buffer, then the FIFO Message Count in the corresponding FIFO Status Register is incremented by 1.

If the Message Count matches the FIFO Depth, then the FIFO Full flag is set.

The oldest message in the GW FIFO is included in the scan for transmission by the corresponding RS-CANFD module channel logic.

When a message is successfully transmitted from the GW FIFO, the Message Count value is decremented by 1.

When all the messages from the GW FIFO are transmitted, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO Message count matches the FIFO Depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message will be lost (no overwrite of already stored messages will take place).

The Interrupt generation conditions for the GW FIFO buffers can be configured by configuring the RSCFDnCFDCFCCk.CFIM bit in the corresponding Transmit/Receive FIFO Configuration / Control Register.

If RSCFDnCFDCFCCk.CFIM bit is 0_B, then RX interrupt flag is set when FIFO counter increments and reaches value configured by RSCFDnCFDCFCCk.CFIGCV and the TX interrupt flag is set when FIFO transmits the last message successfully.

If RSCFDnCFDCFCCk.CFIM bit is 1_B, then RX Interrupt flag is set at the end of storage of every received message and TX interrupt flag is set if a message is successfully transmitted from the FIFO.

Transmit/Receive FIFO can set interrupt, when CAN frame transmitted is completed.

Transmit/Receive FIFO can set interrupt, when CAN frame reception is completed.

Transmit/Receive FIFO can set interrupt, when FIFO is in full status in RX mode or GW mode.

In the case of RSCFDnCFDCFCCEk.CFBME = 1, it becomes FIFO buffering mode, send data is stored in Transmit/Receive FIFO, and transmission is stopped. Transmission will be started if it is set as RSCFDnCFDCFCCEk.CFBME = 0.

The Transmit/Receive FIFO Buffers configured in GW Mode can be disabled by clearing the RSCFDnCFDCFCCk.CFE bit in the Transmit/Receive FIFO Configuration / Control Register. If this bit is cleared, the GW FIFO becomes empty as described below:

- Immediately if the message from the GW FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Test Mode if the transmission from the GW FIFO is already scheduled for transmission or already in transmission

Other possible messages pending from the GW FIFO are lost.

Before RSCFDnCFDCFCCk.CFE is set again ensure that the RSCFDnCFDCFSTSk.CFEMP bit is set and that there is no pending abort from the GW FIFO.

When the RSCFDnCFDCFCCk.CFE bit is cleared and the RSCFDnCFDCFSTSk.CFEMP bit is set, the message read and write pointers of the GW FIFO are cleared and are no longer active . Hence all messages in the GW FIFO buffers will be lost and no further message can be stored into the GW FIFO.

In applications intended to be used as CAN-to-CAN gateways it would be useful if the Error State Indication (ESI) information of the routing messages is not replaced by the sending node Error State Indication. For this, each channel has the control function register RSCFDnCFDCmFDCFG.ESIC to replace their own ESI information by the routing ESI information.

NOTE

If the sending node is error passive then the ESI bit will be sent anyway as Error passive (ESI = 1).

(3) Interval Timer for FIFO Transmission

For each Transmit/Receive FIFO in TX or GW mode it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is, called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the RSCFDnCFDCFCCk.CFE bit is set.

When the Transmit/Receive FIFO in TX or GW mode is enabled, then the first message will be transmitted without considering this interval time.

The Interval Timer will stop counting when:

FIFO is disabled by clearing the RSCFDnCFDCFCCk.CFE bit.

CAN channel is in CH_RESET mode.

The interval time is specified by the RSCFDnCFDCFCCk.CFITT value in the Transmit/Receive FIFO Configuration / Control Register and can be specified from 0 to 255 timer units.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission a value of 0 should be selected.

The timer source can be selected by the configuration bit CFITSS in the Transmit/Receive FIFO Configuration / Control Register. For the timer source the CAN Bit Timing clock of the FIFO related channel or a global reference clock could be selected.

If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT or CH_RESET or CH_STOP mode, then the Interval Timer is stopped for that channel.

If peripheral clock is selected as Interval Timer clock source, then the Interval Timer is stopped only when the CAN channel is in CH_RESET or CH_STOP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value RSCFDnCFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

Refer to **Table 23.150** for RSCFDnCFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency/period.

Table 23.150 Configuration example for the FIFO interval timer reference clock

Reference clock Peripheral clock	1µs	100µs	500µs
16MHz / 62.5 ns	16	1600	8000
20MHz / 50 ns	20	2000	10000
32MHz / 31.25 ns	32	3200	16000
50MHz / 20 ns	50	5000	25000

Additionally the reference clock resolution can be specified by the Interval Timer Reference Clock Resolution Value RSCFDnCFDCFCCk.CFITR in the Transmit/Receive FIFO Configuration / Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10).

The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time. The whole range for the Separation Time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX/GW FIFO. Hence, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message will earliest be sent after this interval time.

The **Figure 23.44** shows an example timing of the internal processing.

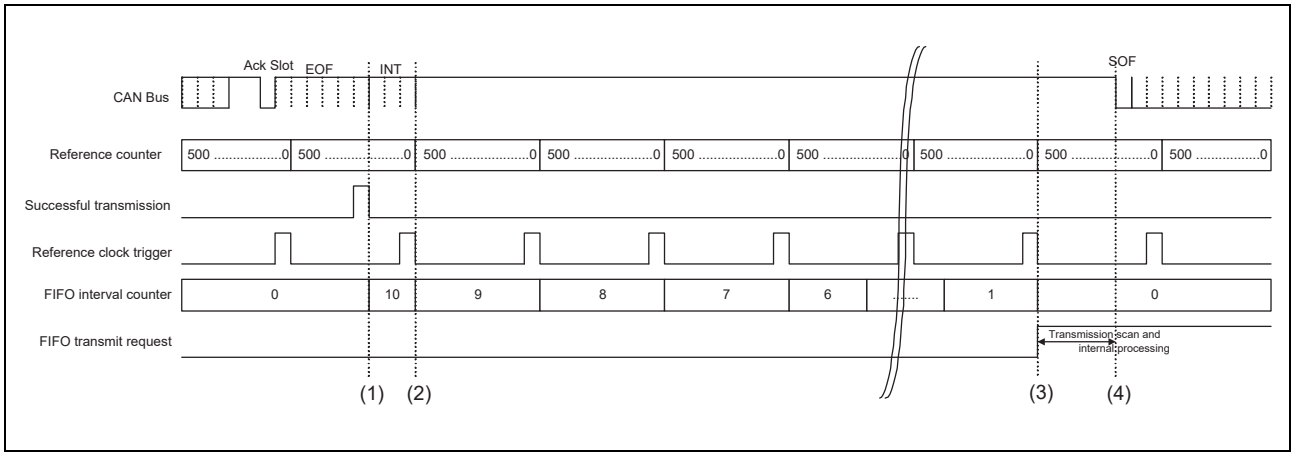


Figure 23.44 Example for interval processing time

The configuration for this timing above is following:

- Peripheral clock frequency = 50MHz
- Interval Timer Reference clock (RSCFDnCFDGCFG.ITRCP)= 500 times
- Reference clock due to the settings above = 10µs
- Transmit/Receive FIFO interval Timer Source Selection (RSCFDnCFDCFCCk.CFITSS)= 0
- Transmit/Receive FIFO Interval Timer Resolution (RSCFDnCFDCFCCk.CFITR)= 0
- Transmit/Receive FIFO Interval Transmission time (RSCFDnCFDCFCCk.CFITT)= 10 times
- Theoretical Message separation interval = 100 µs

(1) Internal FIFO interval timer is restarted with the occurrence of Successful transmission result. This restart is not synchronized to the Reference clock trigger. Therefore the first interval is counting less or equal to one Reference clock interval.

(2) With the next Reference clock trigger the FIFO interval timer is decremented and so on.

(3) When the FIFO interval timer reached the value “zero” the FIFO Transmit request is set.

(4) When the FIFO is selected for transmission then the transmission will start soon. Due to internal processing this usually takes less than 3 CAN bit time, between internal FIFO transmit request set (3) and actual transmission.

In worst case when multi events like reception scan, internal message routing, transmit scan on all channels happen, then it could take up to 1152 peripheral clock cycles.

As shown in **Figure 23.44**, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, users should configure RSCFDnCFDCFCCk.CFITT to required minimum value+1

If further TX Message Buffers or TX/GW FIFOs are configured for transmission for the same channel the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time due to higher priority message transmission from these TX Message Buffers or TX/GW FIFOs.

Figure 23.45 shows a block diagram of the FIFO interval time generation circuit.

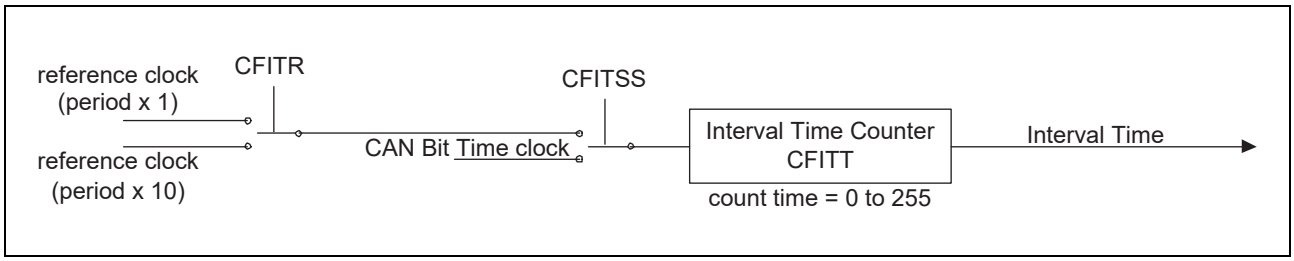


Figure 23.45 Block Diagram of FIFO interval timer

23.9.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] TX Message Buffers, which are accessed via one access window. One Channel has four TX Queue. One TX Queue can be configured with a depth of 3 up to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] buffer and it is using the TX Message Buffer No. 0 as access window (Referred to TXQ0). The second TX Queue can be configured with a depth of 3 up to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] buffer and it is using the TX Message Buffer No. 31 as access window (Referred to TXQ1). The third TX Queue can be configured with a depth of 3 up to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] buffer and it is using the TX Message Buffer No. 32 as access window (Referred to TXQ2). The fourth TX Queue can be configured with a depth of 3 up to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] buffer and it is using the TX Message Buffer No. 63 as access window (Referred to TXQ3). All the Message of TXQ0, TXQ1, TXQ2 and TXQ3 enter the priority comparison for the transmission, which should be only ID Priority (RSCFDnCFDGCFG.TPRI = 0_B). The registers for TXQ0 are RSCFDnCFDTXQCC0m, RSCFDnCFDTXQSTS0m, and RSCFDnCFDTXQPCTR0m. The registers for TXQ1 are RSCFDnCFDTXQCC1m, RSCFDnCFDTXQSTS1m, and RSCFDnCFDTXQPCTR1m. The registers for TXQ2 are RSCFDnCFDTXQCC2m, RSCFDnCFDTXQSTS2m, and RSCFDnCFDTXQPCTR2m. The registers for TXQ3 are RSCFDnCFDTXQCC3m, RSCFDnCFDTXQSTS3m, and RSCFDnCFDTXQPCTR3m. As access window TX Message Buffer No.63 (TXQ3) or TX Message Buffer No.32 (TXQ2) or TX Message Buffer No.31 (TXQ1) or TX Message Buffer No.0 (TXQ0) is used, refer to related access registers TX Message Buffer ID Register (TMID), TX Message Buffer Pointer Register (TMPTR), TX Message Buffer Data Field 0 Register (TMDF0) and TX Message Buffer Data Field 1 Register (TMDF1).

The depth of each TXQ0 Buffer can be configured by writing to the RSCFDnCFDTXQCC0m. TXQDC[4:0] bits of the TX Queue Configuration / Control Register. TXQ0can set from TXMB0 to TXMB31 [for U2A-EVA, U2A16, U2A8] and TXMB15 [for U2A6] as a queue buffer at the maximum.

The 31 [for U2A-EVA, U2A16, U2A8] and 15 [for U2A6] available options for depth configuration are:

00000_B: TX Queue disabled

00001_B: reserved

00010_B: 3 Messages

:

11101_B: 30 Messages

11110_B: 31 Messages

11111_B: 32 Messages

The depth of each TXQ1 Buffer can be configured by writing to the RSCFDnCFDTXQCC1m.TXQDC[4:0] bits of the TX Queue Configuration / Control Register.

TXQ1 can set from TXMB31 [for U2A-EVA, U2A16, U2A8] and TXMB15 [for U2A6] to TXMB0 as a queue buffer at the maximum.

The 31 [for U2A-EVA, U2A16, U2A8] and 15 [for U2A6] available options for depth configuration are:

00000_B: TX Queue disabled

00001_B: reserved

00010_B: 3 Messages

:

11101_B: 30 Messages

11110_B: 31 Messages

11111_B: 32 Messages

The depth of each TXQ2 Buffer can be configured by writing to the RSCFDnCFDTXQCC2m.TXQDC[4:0] bits of the TX Queue Configuration / Control Register.

TXQ2 can set from TXMB32 to TXMB63 [for U2A-EVA, U2A16, U2A8] and TXMB47 [for U2A6] as a queue buffer at the maximum.

The 31 [for U2A-EVA, U2A16, U2A8] and 15 [for U2A6] available options for depth configuration are:

00000_B: TX Queue disabled

00001_B: reserved

00010_B: 3 Messages

:

11101_B: 30 Messages

11110_B: 31 Messages

11111_B: 32 Messages

The depth of each TXQ3 Buffer can be configured by writing to the RSCFDnCFDTXQCC3m.TXQDC[4:0] bits of the TX Queue Configuration / Control Register.

TXQ3 can set from TXMB63 [for U2A-EVA, U2A16, U2A8] and TXMB47 [for U2A6] to TXMB32 as a queue buffer at the maximum.

The 31 [for U2A-EVA, U2A16, U2A8] and 15 [for U2A6] available options for depth configuration are:

00000_B: TX Queue disabled

00001_B: reserved

00010_B: 3 Messages

:

11101_B: 30 Messages

11110_B: 31 Messages

11111_B: 32 Messages

When using TXQ1 and TXQ0 simultaneously, the depth of TXQ is 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] or less in total.

When using TXQ3 and TXQ2 simultaneously, the depth of TXQ is 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] or less in total.

Users should not access all the TX Message Buffers forming the TX Queue directly (except TX Message Buffer No. 63, TX Message Buffer No. 32, TX Message Buffer No. 31 and TX Message Buffer No. 0, which acts as TX Queue access window).

When RSCFDnCFDGAFLP0j.GAFLSRD is set and the RSCFDnCFDTXQCC00m.TXQGWE. RSCFDnCFDTXQCC1m.TXQGWE. RSCFDnCFDTXQCC2m.TXQGWE is also set, a receiving frame is stored in the target TXQ as send data by routing.

When RSCFDnCFDTXQCCn.TXQOWE bit is 1_B, the TX queue is in TX queue overwrite mode. If the message of the same ID is stored in TX Queue when a frame is received and it is stored in TX Queue, an old message will be overwritten by a new message. Therefore, an old message is not transmitted. When the old message of the same ID is transmitting and a CAN bus error and an arbitration-lost occur, the message of old ID is not resent.

When users use the function in GW mode and TX queue overwrite mode, the depth of TXQ (RSCFDnCFDTXQCC0m.TXQDC) should be configured to the value which is the various number of ID which is used in the TX queue plus 3. If it accesses by routing in gateway mode when a TXQ buffer is full, RSCFDnCFDTXQSTS.TXQMLT will be set and send data will be thrown away. Then the function is valid for the standard ID frame and is invalid for the extended ID frame.

Explanation of operation of the TXQ same ID over-writing function in GW mode is shown below.

Figure 23.46 Explanation of operation of the TXQ same ID over-writing function in GW mode

When a system writes in TXQ, a system should write in send data, after checking the state of TXQ.

Users should also not access or configure the related TX Message Buffer Control Register.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full then no further access should be done to the Queue, until it is no longer full. If it accesses by software writing in when the buffer of TXQ is full, send data will be overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue empty flag is set as described below:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Test Mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission

NOTE

The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission needs to be requested again.

Before TXQE is set again ensure that the RSCFDnCFDTXQSTSm.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared all messages in the TX Queue buffers will be lost and no further message should be stored into the TX Queue.

When a message has been stored to the TX Queue, FF_H must be written in to the TX Queue Pointer Control Register. This will set the transmit request automatically and change the internal Message Buffer pointer to the next free Message Buffer location of the TX Queue.

NOTE

If two messages with the same Identifier are stored in the TX Queue, then the order of transmission of these messages could be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new Message with the same Identifier is stored in the TX Queue.

Or if TX queue overwrite mode is used, the frame of the same ID will be rewritten on a new frame.

For the TX Queue a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the RSCFDnCFDTXQCCn.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in **Figure 23.47**.

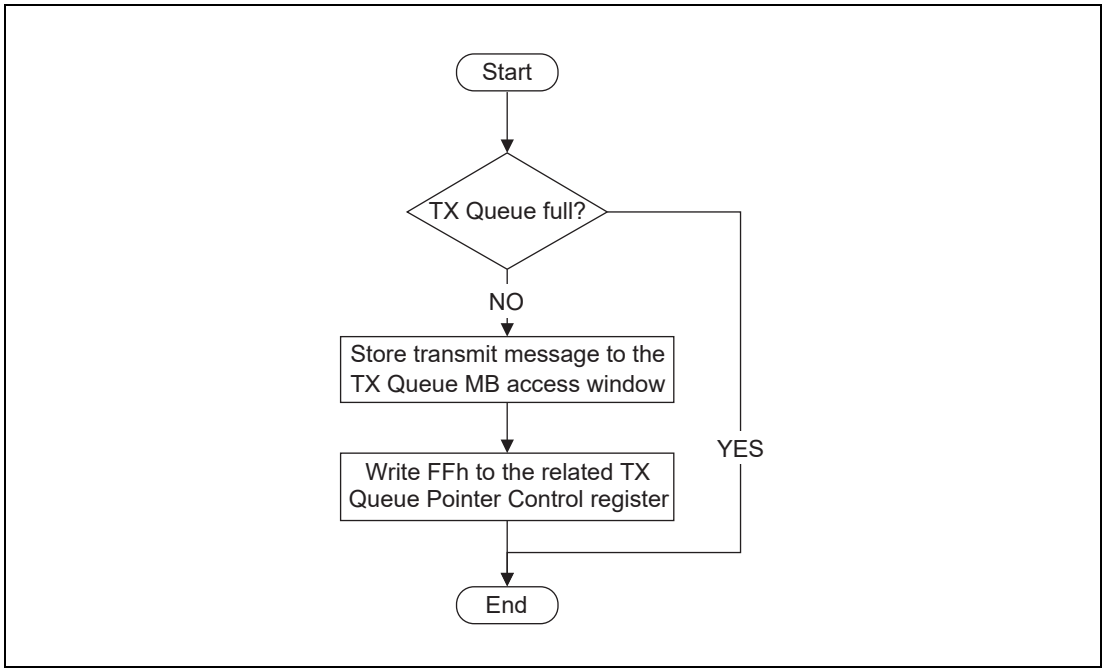


Figure 23.47 TX Queue transmission request

TXQ Name	Access window	Range width		Direction		HW Routing access point	CPU access point	DMA access point	Note	
		U2A-EVA, U2A16, U2A8	U2A6	U2A-EVA, U2A16, U2A8	U2A6				U2A-EVA, U2A16, U2A8	U2A6
TXQ0	TXMB0	0, 3-32	0, 3-16	TXMB0 -> TXMB31	TXMB0 -> TXMB15	Yes	Yes	Yes	When using both TXQ0/1, the number sum total of stages is 32 or less.	When using both TXQ0/1, the number sum total of stages is 16 or less.
TXQ1	TXMB31	0, 3-32	0, 3-16	TXMB31 -> TXMB0	TXMB15 -> TXMB0	Yes	Yes	No		
TXQ2	TXMB32	0, 3-32	0, 3-16	TXMB32 -> TXMB63	TXMB32 -> TXMB47	Yes	Yes	No	When using both TXQ2/3, the number sum total of stages is 32 or less.	When using both TXQ2/3, the number sum total of stages is 16 or less.
TXQ3	TXMB63	0, 3-32	0, 3-16	TXMB63 -> TXMB32	TXMB47 -> TXMB32	No	Yes	Yes		

TXQ0 can use HW Routing, CPU access, and DMA access.

HW Routing access, CPU access, and DMA access should not be used simultaneously.

The one access method is chosen.

23.9.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers for each CAN channel. Two THL Buffer are provided for each CAN channel and each THL Buffer can store up to 32 THL entries for a CAN channel.

The RSCFDnCFDTHLCCm.THLDTE bit of the TX History List Configuration / Control Register can be used to configure if only message information from TX FIFOs / TX Queue is stored or if all transmit message information from TX Queue, TX FIFO or normal TX Message Buffers should be stored in the TX History List for a CAN channel.

When a RSCFDnCFDTHLCCm.THLDGE bit is set up, the information on all the frames transmitted in GW mode is stored in THL.

Each transmit message can be individually configured for acceptance to the TX History List by the RSCFDnCFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted on that CAN channel.

Storing to the List is not synchronized with the status of RSCFDnCFDTMSTSp.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the List could happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE is configured to 1 or when the TX History List counter RSCFDnCFDTHLSTSm.THLMC[5:0] is increased.

The delay time is dependent on the number of channels due to internal processing.

Maximum delay time from setting the RSCFDnCFDTMSTSp.TMTRF to storing the TX History List data is 224 peripheral bus clock cycles.

The History list records following information of the transmitted message:

Buffer Type:

001: TX Message Buffer

010:TX FIFO

100:TX Queue

Buffer Number:

TX Message Buffer, TX Queue Message Buffer or TX Message Buffer Link for the Transmit/ Receive FIFO Buffer from which the transmission occurred. The number depends upon the Buffer Type, refer to **Table 23.151**.

Transmission ID:

Transmission Pointer stored in the transmission message

Transmit Timestamp:

Message timestamp captured at capture point as configured by RSCFDnCFDGFDCFG.TSCCFG.

Transmission Information Label:

Transmission information label stored in the transmission message.

Transmit Gateway Buffer indication:

In the case of the data transmitted from Gateway, RSCFDnCFDTHLACC0m.TGW bit is set to 1.

Table 23.151 TX History List Buffer Number entry (1/3)

Buffer Type / Buffer Number	001	010	100
000000	Message Buffer 0	Number shown corresponds to the Transmit/Receive FIFO TX Message Buffer Link CFTML of the related Transmit/Receive FIFO Configuration / Control Register	Number shown corresponds to the Message Buffer belonging to the TX Queue from which the frame was transmitted
000001	Message Buffer 1		
000010	Message Buffer 2		
000011	Message Buffer 3		
000100	Message Buffer 4		
000101	Message Buffer 5		
000110	Message Buffer 6		
000111	Message Buffer 7		
001000	Message Buffer 8		
001001	Message Buffer 9		
001010	Message Buffer 10		
001011	Message Buffer 11		
001100	Message Buffer 12		
001101	Message Buffer 13		
001110	Message Buffer 14		
001111	Message Buffer 15		
010000	Message Buffer 16		
010001	Message Buffer 17		
010010	Message Buffer 18		
010011	Message Buffer 19		
010100	Message Buffer 20		
010101	Message Buffer 21		
010110	Message Buffer 22		
010111	Message Buffer 23		
011000	Message Buffer 24		
011001	Message Buffer 25		
011010	Message Buffer 26		
011011	Message Buffer 27		
011100	Message Buffer 28		
011101	Message Buffer 29		
011110	Message Buffer 30		
011111	Message Buffer 31		
100000	Message Buffer 32		
100001	Message Buffer 33		
100010	Message Buffer 34		
100011	Message Buffer 35		
100100	Message Buffer 36		
100101	Message Buffer 37		
100110	Message Buffer 38		
100111	Message Buffer 39		
101000	Message Buffer 40		

Table 23.151 TX History List Buffer Number entry (2/3)

Buffer Type / Buffer Number	001	010	100
101001	Message Buffer 41	Number shown corresponds to the Transmit/Receive FIFO TX Message Buffer Link CFTML of the related Transmit/Receive FIFO Configuration / Control Register	Number shown corresponds to the Message Buffer belonging to the TX Queue from which the frame was transmitted
101010	Message Buffer 42		
101011	Message Buffer 43		
101100	Message Buffer 44		
101101	Message Buffer 45		
101110	Message Buffer 46		
101111	Message Buffer 47		
110000	Message Buffer 48		
110001	Message Buffer 49		
110010	Message Buffer 50		
110011	Message Buffer 51		
110100	Message Buffer 52		
110101	Message Buffer 53		
110110	Message Buffer 54		
110111	Message Buffer 55		
111000	Message Buffer 56		
111001	Message Buffer 57		
111010	Message Buffer 58		
111011	Message Buffer 59		
111100	Message Buffer 60		
111101	Message Buffer 61		
111110	Message Buffer 62		
111111	Message Buffer 63		
1000000	Message Buffer 64		
1000001	Message Buffer 65		
1000010	Message Buffer 66		
1000011	Message Buffer 67		
1000100	Message Buffer 68		
1000101	Message Buffer 69		
1000110	Message Buffer 70		
1000111	Message Buffer 71		
1001000	Message Buffer 72		
1001001	Message Buffer 73		
1001010	Message Buffer 74		
1001011	Message Buffer 75		
1001100	Message Buffer 76		
1001101	Message Buffer 77		

Table 23.151 TX History List Buffer Number entry (3/3)

Buffer Type / Buffer Number	001	010	100
1001110	Message Buffer 78	Number shown corresponds to the Transmit/Receive FIFO TX Message Buffer Link CFTML of the related Transmit/Receive FIFO Configuration / Control Register	Number shown corresponds to the Message Buffer belonging to the TX Queue from which the frame was transmitted
1001111	Message Buffer 79		
1010000	Message Buffer 80		
1010001	Message Buffer 81		
1010010	Message Buffer 82		
1010011	Message Buffer 83		
1010100	Message Buffer 84		
1010101	Message Buffer 85		
1010110	Message Buffer 86		
1010111	Message Buffer 87		
1011000	Message Buffer 88		
1011001	Message Buffer 89		
1011010	Message Buffer 90		
1011011	Message Buffer 91		
1011100	Message Buffer 92		
1011101	Message Buffer 93		
1011110	Message Buffer 94		
1011111	Message Buffer 95		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the RSCFDnCFDCFFDCSTS.CFPTR[15:0] part of the Transmit/Receive FIFO Access Pointer Register for a TX FIFO or to the RSCFDnCFDTMFDCTR.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window Message Buffer.

When the message is successfully transmitted then this identification number is stored together with the other message related information to the TX History List and can be read via the Transmission ID (TID) of the TX History List Access Register.

Also for normal TX Message Buffers, the RSCFDnCFDTMFDCTR.TMPTR[15:0] part of the TX Message Buffer Pointer Register will be stored in the Transmission History List. Information label is the same.

Figure 23.48 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register will be done for every single entry.

After reading one entry, FF_H has to be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 23.49 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the RSCFDnCFDTHLCCm.THLIM bit of the corresponding TX History List Configuration / Control Register and enabled with the RSCFDnCFDTHLCCm.THLIE bit of the same registers, either to

generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the RSCFDnCFDTHLSTSm.THLELT bit in the TX History List Status Register.

Status of this bit is also shown by the THLES bit in the Global Error Flag Register.

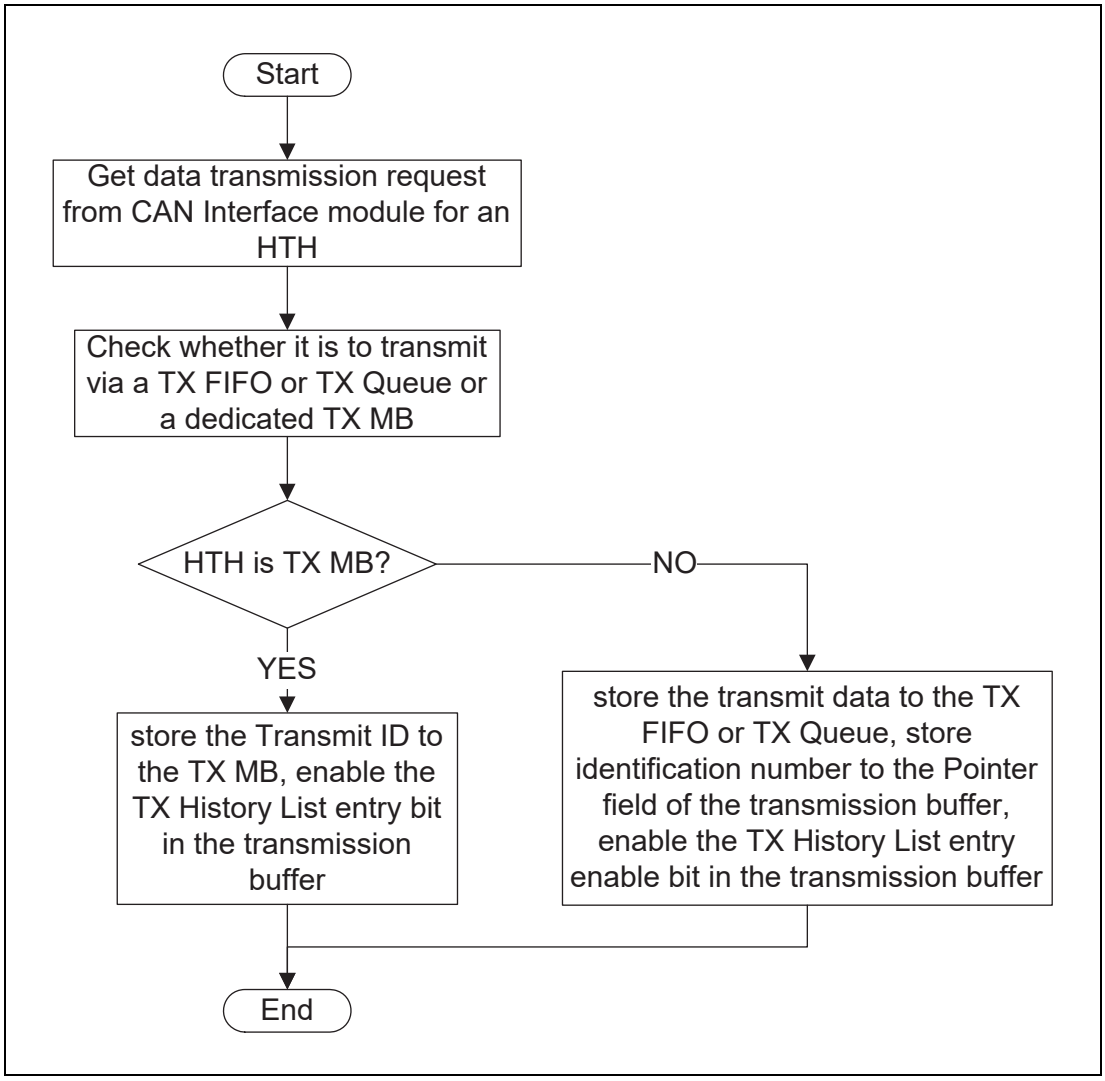


Figure 23.48 TX History List preparation flow

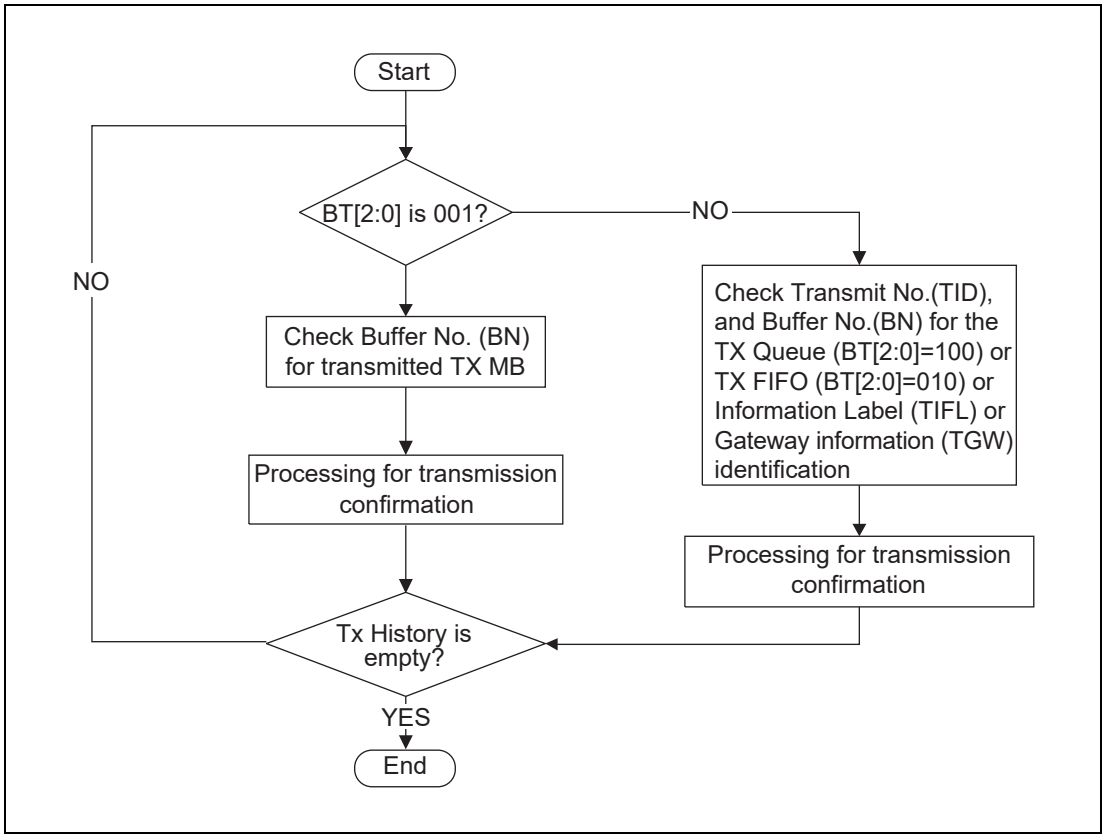


Figure 23.49 TX History List processing flow

23.9.2.6 TX data padding

If the data length code (DLC) of the transmitting message is higher number of data bytes than the buffer size. Then the data bytes beyond the restricted range will be replaced by bytes with the value of CC HEX.

This could happen for Transmit/Receive FIFOs configured as (TX or GW) when the transmit message DLC is higher than the RSCFDnCFDCFCCk.CFPLS.

This could also happens in CANFD only mode, if a Classical CAN Frame is configured with a DLC bigger than 8.

23.10 ECC Check

An ECC check mechanism is implemented in RS-CANFD to check consistency of data in the RAM.

Each memory location in the RAM consists of 39 bits. 32 bits are used for storage of user data and 7 bits are used for storage of ECC data. For each long-word (32 bits) memory location, ECC data is allocated. The value of this ECC data is calculated by the ECC generation logic.

The ECC check mechanism checks the data during read access from CAN side or CPU side to the RAM. If a single bit error is detected, then it is corrected and a flag is set in the ECC Status Register (Register is outside of CAN IP).

If multiple bit error is detected from CAN side read during the transmission scan, then the transmit request of the CAN channel which is under transmission scan will be suppressed and the corresponding RSCFDnCFDGERFL.EEFm bit is set. This is done to avoid transmission of corrupted data.

In general if multiple bit errors are detected then perform RAM test to check the RAM status. When no problem is found, make a transition to global reset mode and then make communication initialization again.

NOTE

If an error is detected by the ECC logic then the error address of the RAM access will be captured in the capture register which is part of the ECC macro (not RS-CANFD).

The captured RAM address will not match with the SFR address.

23.11 Test Mode

The RS-CANFD module can be configured into Test Modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the RS-CANFD module in the test modes.

Note that all Test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Users should not enable any combinations of the various Test modes specified in this Section.

The Test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes

23.11.1 Channel specific test modes

Each CAN channel can be configured into following test modes:

- Basic test mode
- Listen-only mode
- Self test mode 0 (External Loop back mode)
- Self test mode 1 (Internal Loop back mode)
- Restricted Operation Mode

23.11.1.1 Basic test mode

The Basic test mode is used for Restricted Operation Mode or CRC Error Test.

This mode can output value of CRC to CFDCnERFL.CRCREG, CFDCnFDCRC.CRCREG and CFDCnFDCRC.SCNT.

The Basic test mode should be used when a particular test setting needs to be enabled other than when in Listen-Only and Self-test modes

23.11.1.2 Listen-only mode

The ISO 11898-1 (2015) recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin will remain in recessive state.

This mode can be used for Baud Rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX Message Buffer or TX- / GW-FIFO of this channel.

NOTE

If a message is stored in GW FIFO or Routing TXQ then users should ensure that the transmitting channel is not in Listen-only mode so that transmission is not requested for this channel from the GW FIFO or Routing TXQ.

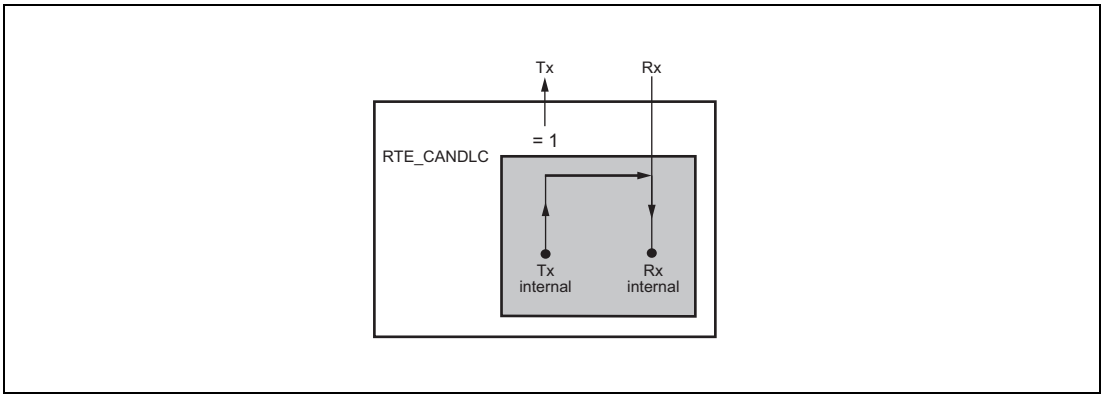


Figure 23.50 Listen-only mode

23.11.1.3 Self test mode 0 (External Loop back mode)

In Self Test Mode 0, the CAN engine treats its own transmitted messages as received messages via the CAN transceiver and can store them into its receive Message Buffers.

CAN node except Transceiver should be stop in this test.

So to receive its own transmitted messages, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests.

The Rx/TX pins should be connected to the transceiver.

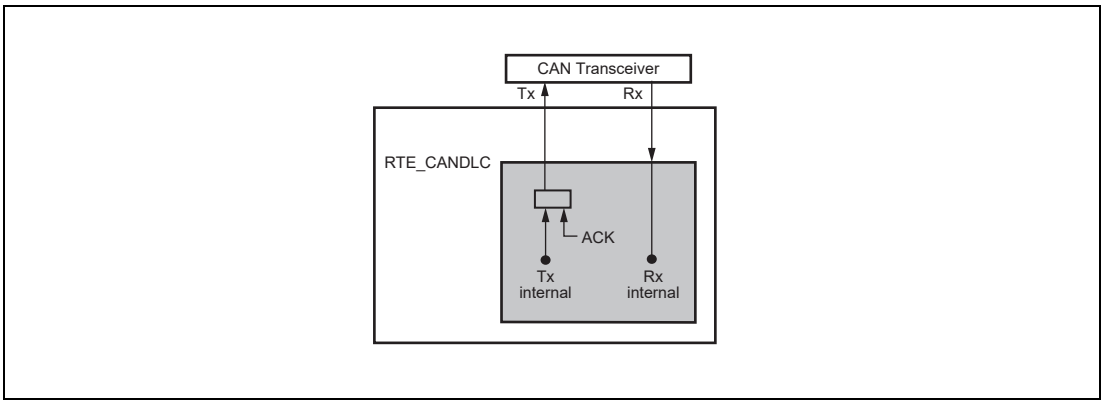


Figure 23.51 Self test mode 0 (External Loop back mode)

23.11.1.4 Self test mode 1 (Internal Loop back mode)

In Self Test Mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to Rx internal. The actual value of the external Rx input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits.

The Rx/TX pins do not need to be connected to the CAN bus or any external device.

NOTE

The channel Pins are also disconnected from the Internal CAN Bus Communication line.

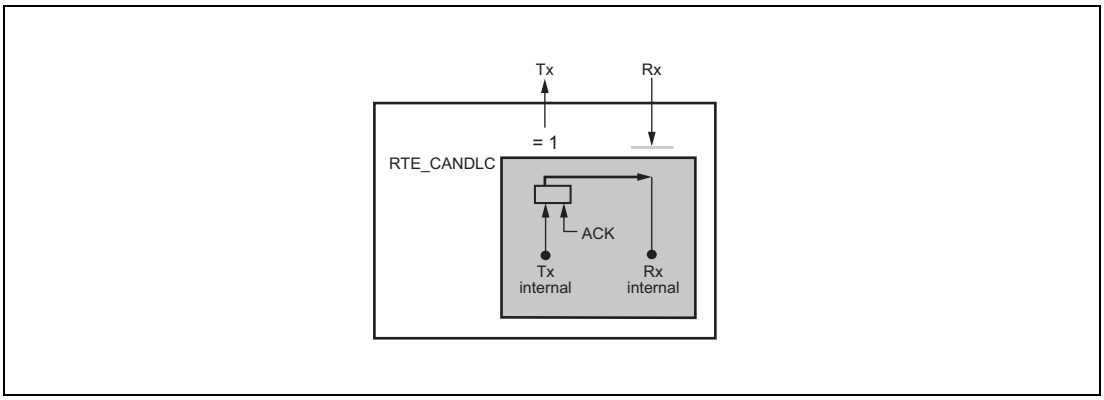


Figure 23.52 Self test mode 1 (Internal Loop back mode)

23.11.1.5 Restricted Operation Mode

In Restricted Operation Mode the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active Error and Overload frames cannot be transmitted instead it waits for the occurrence of bus idle condition to resynchronise itself to the CAN communication after an error or overload condition occurs

Moreover the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors.

The mode is specified as in ISO 11898-1 (2015); however it is permitted to set any transmit requested.

23.11.2 Global test modes

The RS-CANFD module can be configured into following test modes:

- RAM Test Mode
- Internal CAN Bus Communication Mode
- CRC Error Test

For following test modes are protected by a special SW procedure to enable the mode. This SW procedure enables the write access to the test mode by specific unlock Key, the related unlock key can be seen in the table below:

Table 23.152 Related Unlock Key

Test Mode	First Unlock key	Second Unlock key
RAM Test Mode	7575_H	$8A8A_H$

If the SW sequence of the two consecutive unlock key write accesses (word or long-word accesses) is interrupted by any other write access to the SFR or if incorrect data is written to the Global Lock Key Register then the corresponding Test mode cannot be set and the sequence should be re-started.

After the two unlock key write accesses, the next write access should be to set the corresponding Test mode Enable bit. If this is not followed, the unlock mechanism resets and the Test mode enable bit cannot be set and then the unlock sequence should be restarted.

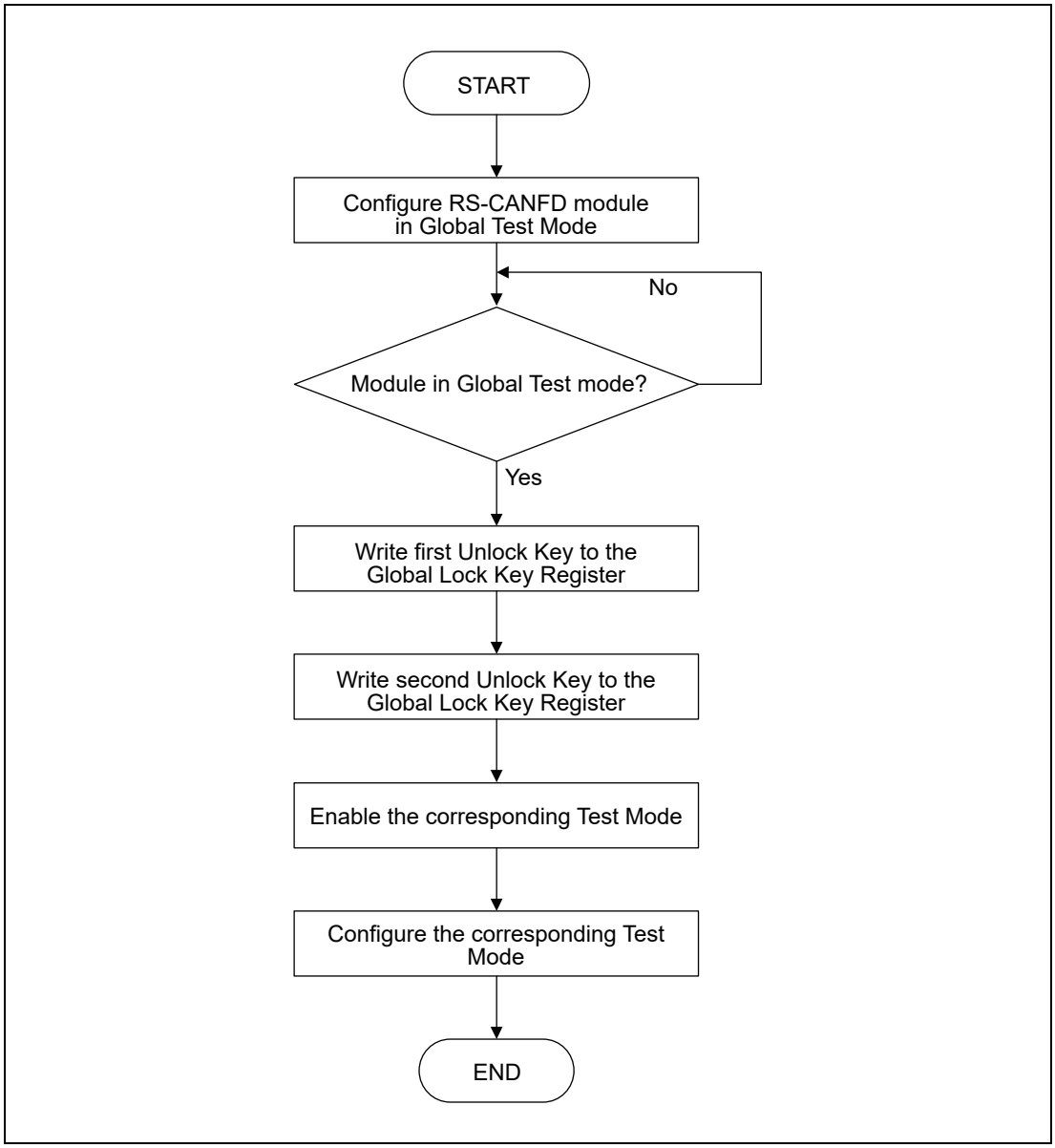


Figure 23.53 Unlock SW protection routine

23.11.2.1 RAM Test Mode

The RS-CANFD module can be configured in RAM Test Mode by setting the RSCFDnCFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding unlock key is written before. This is a special test mode, in which, the complete RAM area can be accessed.

NOTE

The actual RAM size is bigger than the RAM area initialized after HW Reset. Hence, ECC error flag (of the ECC macro) may be set if CPU reads data from this un-initialized RAM area while RS-CANFD module is in RAM Test Mode.

In this mode, the RAM area is split into number of pages (pn) of 256 Bytes each. Which can be accessed via RSCFDnCFDRPGACCr register.

The page should be selected for read / write access by writing to the RSCFDnCFDGTSTCFG.RTMPS[9:0] bits in the Global Test Control Register. Then, data can be read from or written in to the RAM Test Page Access Register.

Figure 23.54 shows the structure of the pages in the RAM when performing a RAM Test Mode.

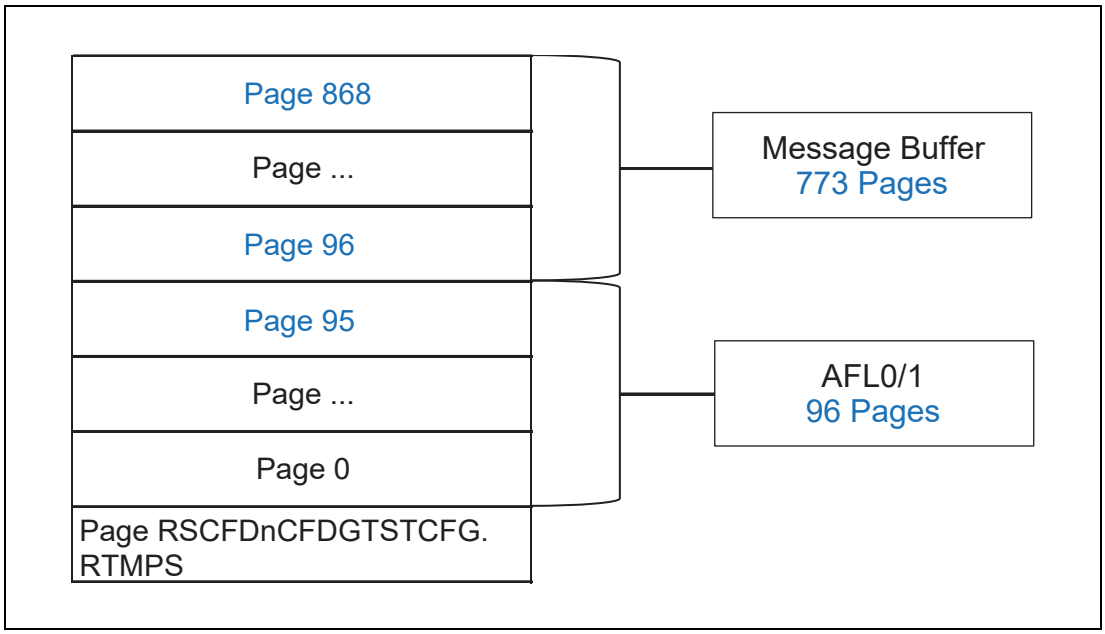


Figure 23.54 RAM page structure

For U2A-EVA, U2A16, U2A8:
The total available RAM size for 8 CAN channel version is 24576 bytes for the AFL RAM and 197888 bytes for the Message Buffer RAM.

For U2A6:
The total available RAM size for a 8 CAN channel version is 16384 bytes for the AFL RAM and 61696 bytes for the Message Buffer RAM.
The total available RAM size for a 4 CAN channel version is 8192 bytes for the AFL RAM and 30848 bytes for the Message Buffer RAM.

AFL RAM0/1 can treat RAM Test mode as one RAM.

The pn and RSCFDnCFDGTSTCFG.RTMPS[9:0] values for the AFL and MB RAMs are calculated in the following way:

pn = ceil(Total RAM size in Bytes / Number of Bytes per page)

AFL RAM:

pn = ceil(24576 / 256) = 96 Pages

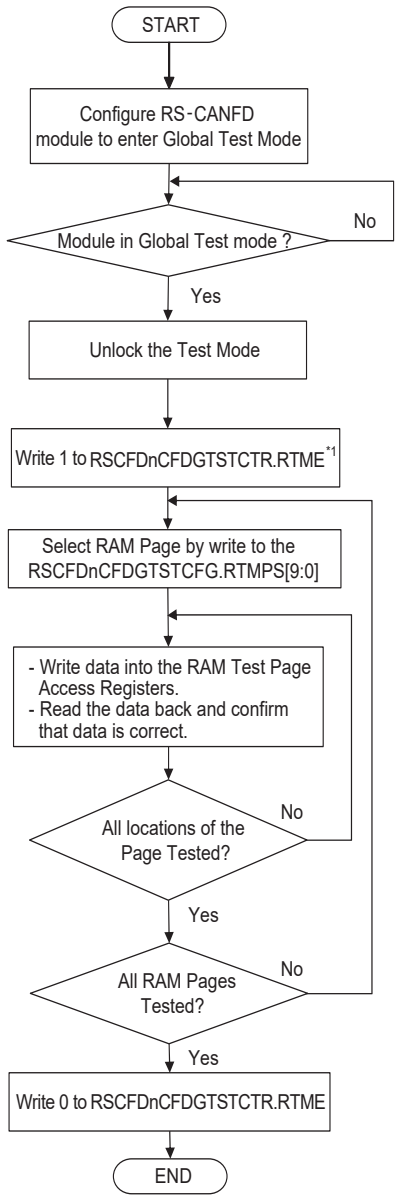
RSCFDnCFDGTSTCFG.RTMPS[9:0] = 0 to 95 (10'h5F) inclusive

MB RAM:

pn = ceil(197888 / 256) = 773 Pages

RSCFDnCFDGTSTCFG.RTMPS[9:0] = 96 to 868 (10'h364) inclusive

Figure 23.55 below shows the SW flow for RAM Test mode.



Note 1. Change into the following status before changing to RAMTEST.
Cancel of a request of transmission.
Disable of all the FIFO and TXQ.
Clear of the receiving flag of a receiving buffer.

Figure 23.55 RAM Test Mode SW flow

To exit this Test mode, the RSCFDnCFDGTSTCTR.RTME bit must be cleared. The RSCFDnCFDGTSTCTR.RTME bit is cleared by writing 0_B to it.

The RSCFDnCFDGTSTCTR.RTME bit is cleared automatically when the RS-CANFD module enters Global Reset mode from the Test mode.

23.11.2.2 Internal CAN Bus Communication Test Mode

The RS-CANFD module can be configured in Internal CAN Bus Communication Test Mode by setting the RSCFDnCFDGTSTCTR.ICBCTME bit in the Global Test Control Register. This is a special test mode, in which the CAN channels can be connected together internally to generate a CAN cluster within the RS-CANFD module.

Users should only use following sequence to enter Internal CAN Bus Communication Test Mode:

1. Configure all channels in Halt Mode and check that all channels have entered Halt mode (Global Test mode).
2. Write data into the Global Test Configuration Register to select the channels participating in the Internal CAN Bus Communication Test.
3. Set the RSCFDnCFDGTSTCTR.ICBCTME bit of the Global Test Control Register.
4. Check that RSCFDnCFDGTSTCTR.ICBCTME bit is set in the Global Test Control Register.

In this mode, the TxD outputs of the channels participating (configured) in Internal CAN Bus Communication Mode are connected together using AND gate. The output of the AND gate is connected to the RxD inputs of all participating channels to create a CAN cluster within the RS-CANFD module. The channels are isolated from the external CAN bus while the RS-CANFD module is in this test mode.

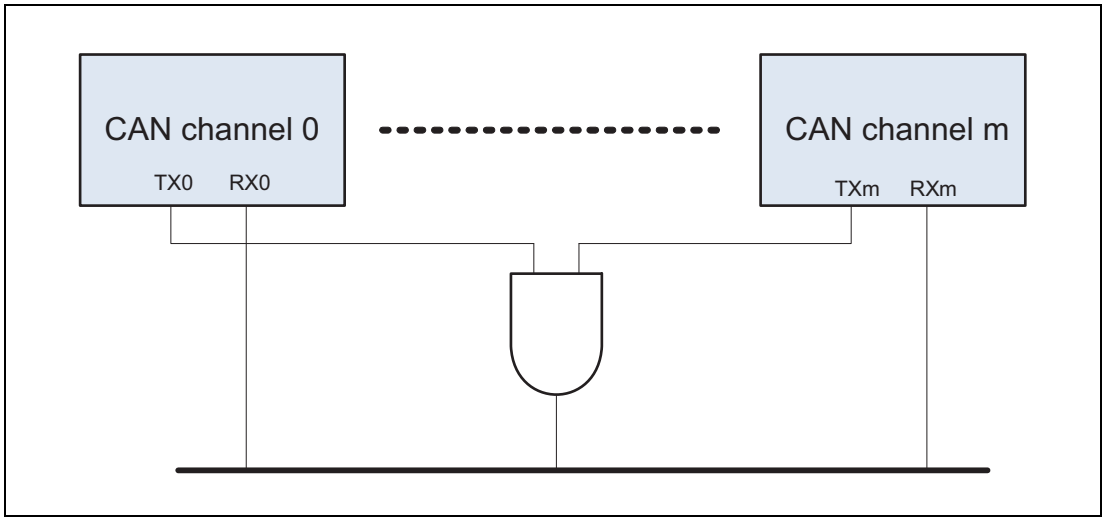


Figure 23.56 Internal CAN Bus connections

The AFL, Flat RX Message Buffers, FIFO Buffers, Flat TX Message Buffers and various registers can now be configured as normal to start communication between channels.

The channels not participating in Internal CAN Bus should only be configured in Halt Mode.

CRC Error Test

After the RS-CANFD module has been configured in Internal CAN Bus Communication Test Mode, the following sequence should be used to perform CRC Error testing. In the sequence below channel x is the reference transmitter RS-CANFD module and channel y is the receiver RS-CANFD module where (x, y = [0…m] and x ≠ y):

1. Configure channel x node to transmit 1 reference message

2. Set the RSCFDnCFDCyCTR.CRCT bit to 1_B, in order to invert the first bit of the incoming bit stream from channel x
3. Set the RSCFDnCFDTMCx.TMTR
4. Wait for the can_cherr_int[y] output signal to set to 1_B
5. Read either the RSCFDnCFDCyERFL.CRCREG or the RSCFDnCFDCyFDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from channel x.
6. Check that RSCFDnCFDCyERFL.CERR is 1_B

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC Error test.

23.12 RAM area configuration

The RAM area used in RS-CANFD can be split into the following groups as shown below in **Figure 23.57**:

- Rule Table area
- RX MB + FIFO Buffer area
- TX MB area
- THL area
- OTB area (One Time Buffer area is used for internal purpose)

Physically the RAM is split in two RAMs the AFL (Rule Table) RAM0 (RSCFDnCFDGAFLID and RSCFDnCFDGAFLP0), AFL (Rule Table) RAM1 (RSCFDnCFDGAFLM and RSCFDnCFDGAFLP1), and the Message Buffer RAM (RX, RX FIFO, Transmit/Receive FIFO, TX, THL and OTB).
The size of AFL RAM and MRAM changes with the number of channels.

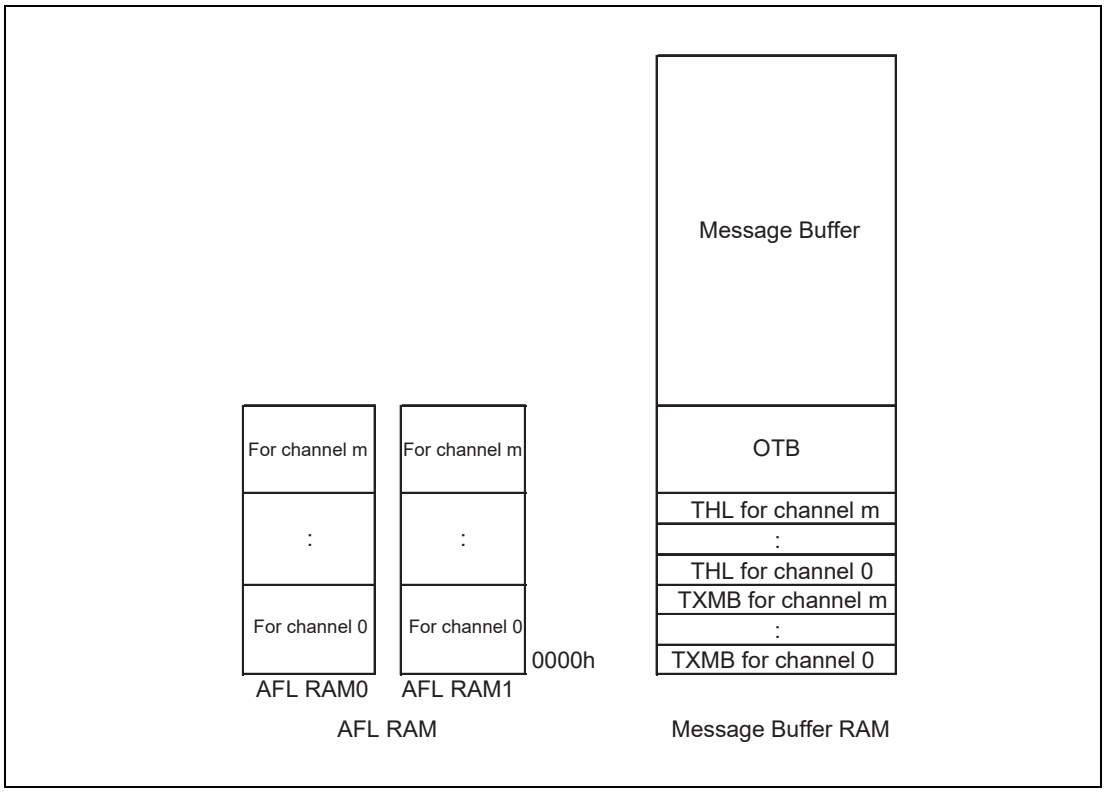


Figure 23.57 RAM area grouping

The Rule Table area always starts at AFL RAM address 0000_H and has a fixed size for a given number of channels.

The MRAM area starts with the TX-MB area at address 0000_H. The TX-MB area is followed immediately by the THL area which is then followed immediately by the OTB area. The size of the TX-MB, THL and OTB area is fixed for a given number of channels.

The OTB area is followed by the message buffer area. The massage buffer area size depends on the configuration of the flat RX MBs, RXFIFOs and CFIFOs. When all are configured the RX-MB area is followed by the RXFIFO area which is followed by the CFIFO area.

The configured MRAM area can be calculated then as follows.

MRAM_cfg = RXMB_MRAM_cfg + RXFIFO_MRAM_cfg + CFIFO_MRAM_cfg +
TXMB_MRAM_cfg +THL_MRAM_cfg + OTB_MRAM_cfg

RXMB_MRAM_cfg = (12 Bytes + RSCFDnCFDRMND.RMPLS) * RSCFDnCFDRMNB.NRXMB

RXFIFO_MRAM_cfg = SUM((12 Bytes + RSCFDnCFDRFCCx.RFPLS) * RSCFDnCFDRFCCx.RFDC)

CFIFO_MRAM_cfg = SUM((12 Bytes + RSCFDnCFDCFCCk.CFPLS) * RSCFDnCFDCFCCk.CFDC)

TXMB_MRAM_cfg = 4864 Bytes * m (For U2A-EVA, U2A16, U2A8) or 2432 Bytes * m (For U2A6)

THL_MRAM_cfg = 256 Bytes * m

OTB_MRAM_cfg = 160 Bytes * m

“x” means RX FIFO index = [0…no_of_RFIFOs-1]

“k” means Transmit/Receive FIFO index = [0 .. no_of_CFIFOs -1]

no_of_RFIFOs : Number of configured RX FIFOs

no_of_CFIFOs: Number of configured CFIFOs

NOTE

For RSCFDnCFDRFCCx.RFDC, RSCFDnCFDCFCCk.CFDC, RSCFDnCFDRMNB.RMPLS, RSCFDnCFDRMNB.NRXMB, RSCFDnCFDRFCCx.RFPLS, and RSCFDnCFDCFCCk.CFPLS the related number of bytes must be used.

The **Table 23.153** shows the calculation of the different RAM areas used for the AFL entries, OTB buffers, TX/RX message buffers and RX/Transmit/Receive FIFOs.

Table 23.153 AFL RAM and MRAM area calculation

RAM Name	RAM Property	RAM Area Calculation Method	RAM Values
	No. of Channels	(m+1)	8
AFL	Avg. rule entries per channel		192
	Max Rule entries	(m+1) * Avg. rule entries per channel	1536
	No. of Bytes in a rule entry	Fixed	16
	Number of Bytes AFL RAM	Max Rule entries * No. of Bytes in a rule entry	24576
TX MB	No. of TX MBs per channel	Fixed	64
	Max no. of TX MBs	(m+1) * No. of TX MBs per channel	512
	No. of Bytes needed for each TX MB	Fixed	76
	Number of Bytes in TX MB area	(m+1) * No. of TX MBs per channel * No. of Bytes needed for each TX MB	38912
THL	No. of entries in 1 THL buffer	Fixed	32
	Max no. of THL entries	(m+1) * No. of entries in 1 THL buffer	256
	No. of Bytes needed for each THL entry	Fixed	8
	Number of Bytes in THL area	Max no. of THL entries * No. of Bytes needed for each THL entry	2048
OTB	Avg. number of buffers for each channel		2
	Max no. of OTB entries	(m+1) * Avg. number of buffers for each channel	16
	No. of Bytes for OTB entry	Fixed	80
	Number of Bytes in OTB area	Max no. of OTB entries * No. of Bytes for OTB entry	1280
Message Buffer	No. of RX MBs per channel	Fixed	16
	Max no. of RX MBs	(m+1) * No. of RXMBs per channel	128
	No. of RX FIFOs	Fixed	8
	No. of Transmit/Receive FIFOs per channel	Fixed	3
	Max no. of Transmit/Receive FIFOs	(m+1) * No. of Transmit/Receive FIFOs per channel	24
	Avg. number of messages for RXMB and FIFO buffers for each channel		256
	No. of Bytes for each stored message	Fixed	-
	Average size of a Message Buffer in Bytes		76
	Number of Bytes in Message Pool area	(m+1) * Avg. number of messages for RXMB and FIFO buffers for each channel * (No. of Bytes for each stored message or Average size of a Message Buffer in Bytes based on CAN_FD_MODE)	155648
	Number of Bytes Message RAM	Number of Bytes in Message Pool area + Number of Bytes in OTB area + Number of Bytes in THL area	197888

23.12.1 Examples

The **Figure 23.58** below shows one possible configuration of a 8 channel version.

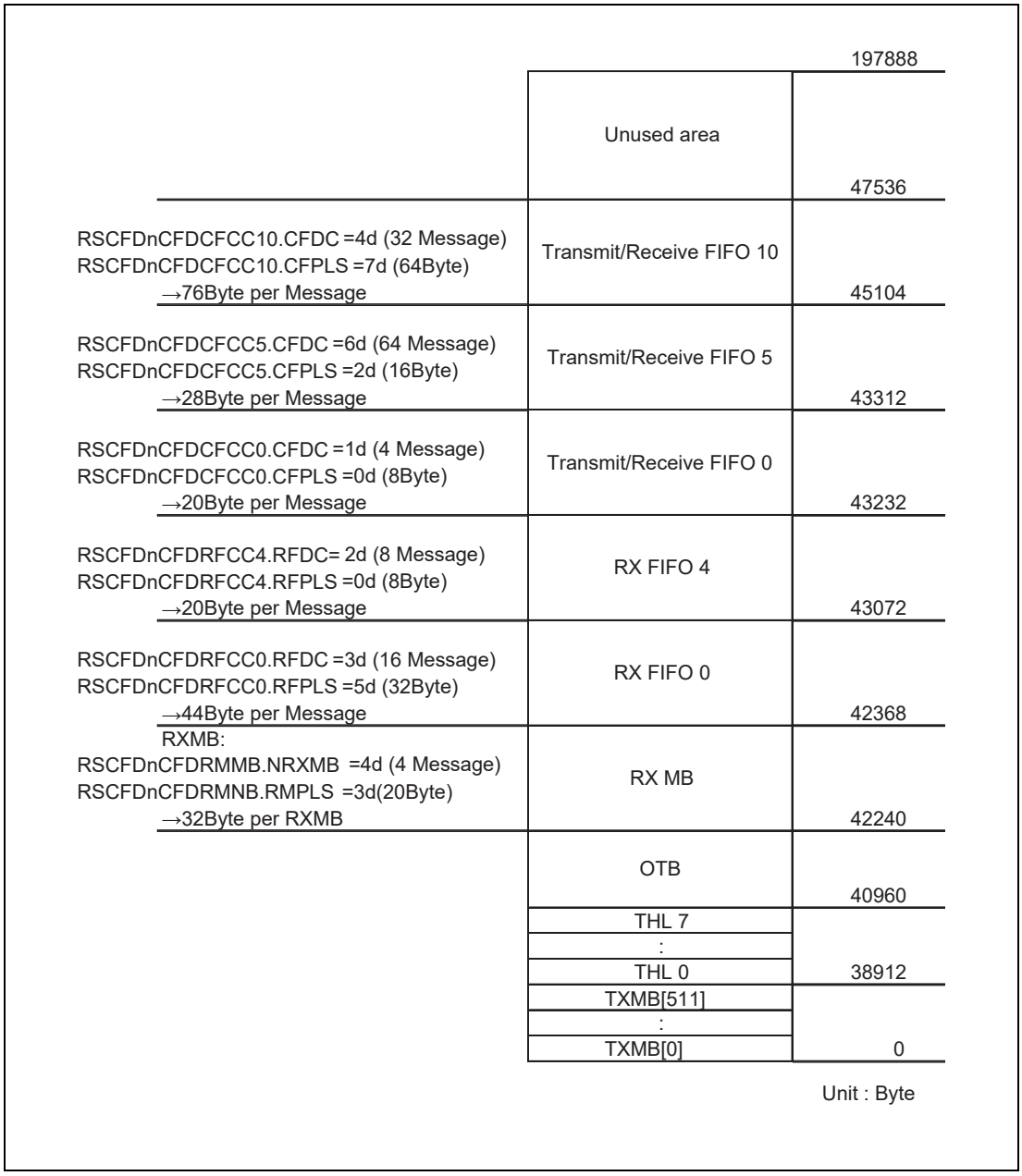


Figure 23.58 RX MB + FIFO buffers RAM area configuration examples of a RS-CANFD 8 channel version

23.12.2 num_ram_chan dependencies

Depending on the num_ram_chan[3:0] configuration the sizes of the AFLRAM and MRAM differs. Further the number of initialization cycles of the memory and the number of RAM test pages is different. Dedicated values can be found in the following table.

Table 23.154 Dedicated values (For U2A-EVA, U2A16, U2A8)

CH	num_ram_chan [3:0]	AFLRAM area size	MRAM area size	RAM initialization cycles (pclk cycle)	RAM Test RTMPS range*1	
1	0001_B	3072	24736	6186	00_H to $6C_H$	(108)
2	0010_B	6144	49472	12370	00_H to $D9_H$	(217)
3	0011_B	9216	74208	18554	00_H to 145_H	(325)
4	0100_B	12288	98944	24738	00_H to $1B2_H$	(434)
5	0101_B	15360	123680	30922	00_H to $21F_H$	(543)
6	0110_B	18432	148416	37106	00_H to $28B_H$	(651)
7	0111_B	21504	173152	43290	00_H to $2FB_H$	(760)
8	1000_B	24576	197888	49474	00_H to 364_H	(868)
8	0000_B	24576	197888	49474	00_H to 364_H	(868)

Note 1. 0001_B (1ch) : User should not access more than 160 Bytes in the last page
0010_B (2ch) : User should not access more than 64 Bytes in the last page
0011_B (3ch) : User should not access more than 224 Bytes in the last page
0100_B (4ch) : User should not access more than 128 Bytes in the last page
0101_B (5ch) : User should not access more than 32 Bytes in the last page
0110_B (6ch) : User should not access more than 192 Bytes in the last page
0111_B (7ch) : User should not access more than 96 Bytes in the last page
1000_B (8ch) : User can access all area in the last page
0000_B (8ch) : User can access all area in the last page
Other values are prohibited.

Table 23.155 Dedicated values (For U2A6)

CH	num_ram_chan [3:0]	AFLRAM area size	MRAM area size	RAM initialization cycles (pclk cycle)	RAM Test RTMPS range*1	
1	0001_B	2048	7712	(not support)	00_H to 26_H	(38)
2	0010_B	4096	15424	3858	00_H to $4C_H$	(76)
3	0011_B	6144	23136	(not support)	00_H to 72_H	(114)
4	0100_B	8192	30848	7714	00_H to 98_H	(152)
5	0101_B	10240	38560	(not support)	00_H to BE_H	(190)
6	0110_B	12288	46272	11570	00_H to $E4_H$	(228)
7	0111_B	14336	53984	(not support)	00_H to $10A_H$	(266)
8	1000_B	16384	61696	15426	00_H to 130_H	(304)
8	0000_B	16384	61696	15426	00_H to 130_H	(304)

Note 1. 0001_B (1ch) : User should not access more than 32 Bytes in the last page
0010_B (2ch) : User should not access more than 64 Bytes in the last page
0011_B (3ch) : User should not access more than 96 Bytes in the last page
0100_B (4ch) : User should not access more than 128 Bytes in the last page
0101_B (5ch) : User should not access more than 160 Bytes in the last page
0110_B (6ch) : User should not access more than 192 Bytes in the last page
0111_B (7ch) : User should not access more than 224 Bytes in the last page
1000_B (8ch) : User can access all area in the last page
0000_B (8ch) : User can access all area in the last page
Other values are prohibited.

23.13 Bus traffic measurement

The idle time of the CAN bus can be measured using the clkc clock or clk_xincan clock. Bus traffic can be calculated based on the measurement results.

23.13.1 How to count the CAN bus idle time

Below is the concept of measuring the idle time of the CAN bus.

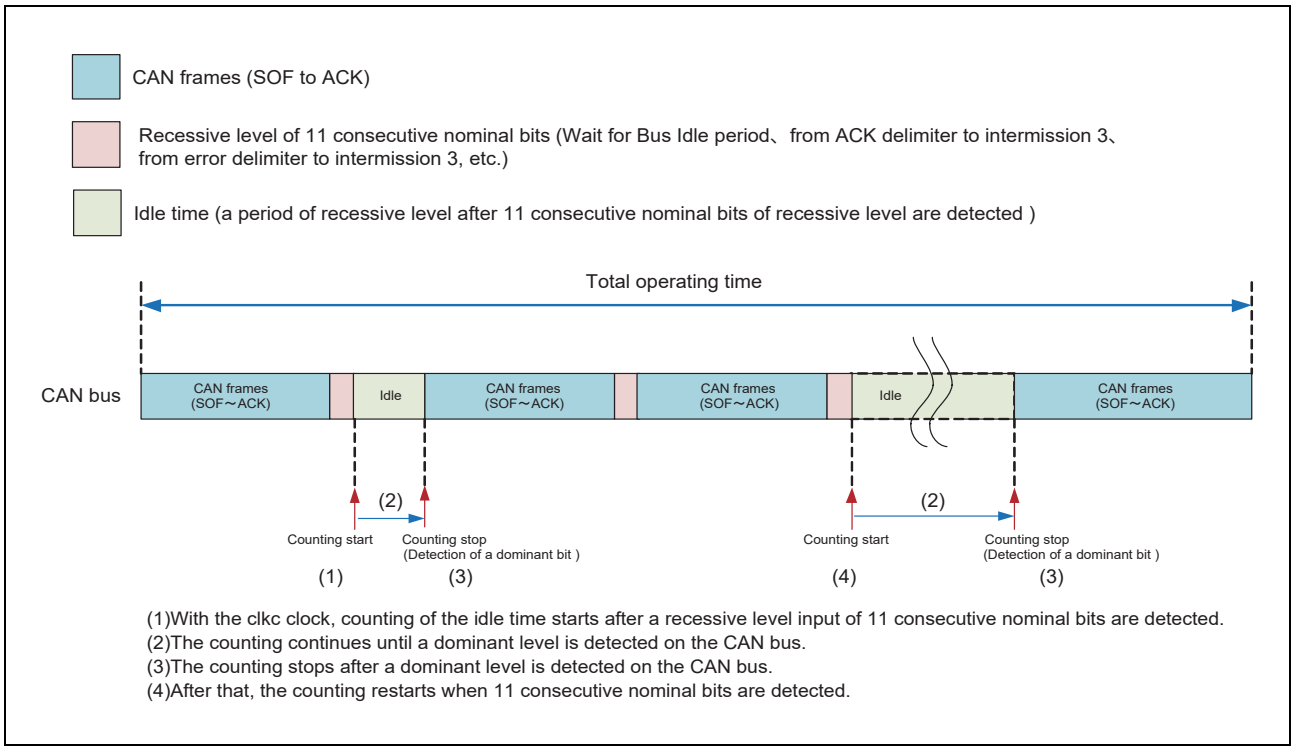


Figure 23.59 Concept of Measuring the Idle Time of the CAN Bus

23.13.2 Operations and measurement procedure

The following are the procedure for measuring the idle time of the CAN bus.

1. The channel to be measured transits to operation mode.
2. Write 1 to RSCFDnCFDCmBLCT.BLCE bit to set the measuring counter to operating mode.
3. Write 1 to RSCFDnCFDCmBLCT.BLCLD bit to clear the counter register.
4. RS-CAN-FD module detect a recessive level input of 11 consecutive nominal bits.
5. RS-CAN-FD module start counting the bus idle time.
6. RS-CAN-FD module detect a dominant level.
7. RS-CAN-FD module counting stops.
8. RS-CAN-FD module detect a recessive level input of 11 consecutive nominal bits.
9. RS-CAN-FD module counting starts.
10. Write 1 to RSCFDnCFDCmBLCT.BLCLD bit to clear the counter register and simultaneously load the counter value to RSCFDnCFDCmBLSTS.
11. Read the value of RSCFDnCFDCmBLSTS.

To stop the measurement counter, write 0 to RSCFDnCFDCmBLCT.BLCE bit.

To initialize the counter, write 1 to RSCFDnCFDCmBLCT.BLCLD bit.

This measurement will be enabled when the channel to be measured is in operation mode.

When the relevant channels are in reset mode, the counter will not operate.

Also, accurate measurements are not available in test mode.

Write 1 to RSCFDnCFDCmBLCT.BLCLD bit to clear the counter register and simultaneously load the value of the counter to RSCFDnCFDCmBLSTS.

The lower three bits of RSCFDnCFDCmBLSTS are fixed to 0.

Based on the values of the counter, software can calculate the CAN bus traffic according to the following formulas.

$$\frac{(\textit{Total operating time} - \textit{Total idle time})}{\textit{Total operating time}} = \frac{\textit{Bus operating time}}{\textit{Total operating time}} = \textit{Bus usage ratio}$$

Total idle time: a value read from RSCFDnCFDCmBLSTS × a clock cycle of clkc

Total operating time: a setting interval of RSCFDnCFDCmBLCT.BLCLD bit

Example) Below is a calculation example under the following conditions.

Conditions: nominal bit rate = 1Mbps

clkc clock = 40MHz (=25ns)

a setting interval of RSCFDnCFDCmBLCT.BLCLD bit = cycle of 1ms

a read value of RSCFDnCFDCmBLSTS register = $4E20_H$ (20000)

$$\frac{(\textit{Total operating time} - \textit{Total idle time})}{\textit{Total operating time}} = \frac{(\textit{1000000 ns} - \textit{20000} \times \textit{25 ns})}{\textit{1000000 ns}} = \textit{50 \%}$$

23.14 Flexible CAN mode

This is a mode in which it is possible to connect the CAN modules of 2 channels to a single CAN driver.

The pair of Channel in flexible CAN mode is as follows.

When RSCFDnCFDGFCMC.FLXC0 bit is set, Channel 0 and Channel 1 of a RS-CANFD module are Flexible CAN mode.

When RSCFDnCFDGFCMC.FLXC1 bit is set, Channel 2 and Channel 3 of a RS-CANFD module are Flexible CAN mode.

When RSCFDnCFDGFCMC.FLXC2 bit is set, Channel 4 and Channel 5 of a RS-CANFD module are Flexible CAN mode.

When RSCFDnCFDGFCMC.FLXC3 bit is set, Channel 6 and Channel 7 of a RS-CANFD module are Flexible CAN mode.

Channel m+1 uses TX/RX terminal of Channel m.

The TX/RX terminal of Channel m+1 cannot use.

In Flexible CAN mode, each channel performs communication processing independently.

However, when one of the channels transmits, the other channel will not return an acknowledge bit.

NOTE

When operating in Flexible CAN mode the error counters (TEC/REC) of the two CAN nodes are not synchronized with each other.

Flexible transmission buffer assignment configured in RSCFDnCFDGFTBAC register and Flexible CAN mode configured in RSCFDnCFDGFCMC register should not be used simultaneously.

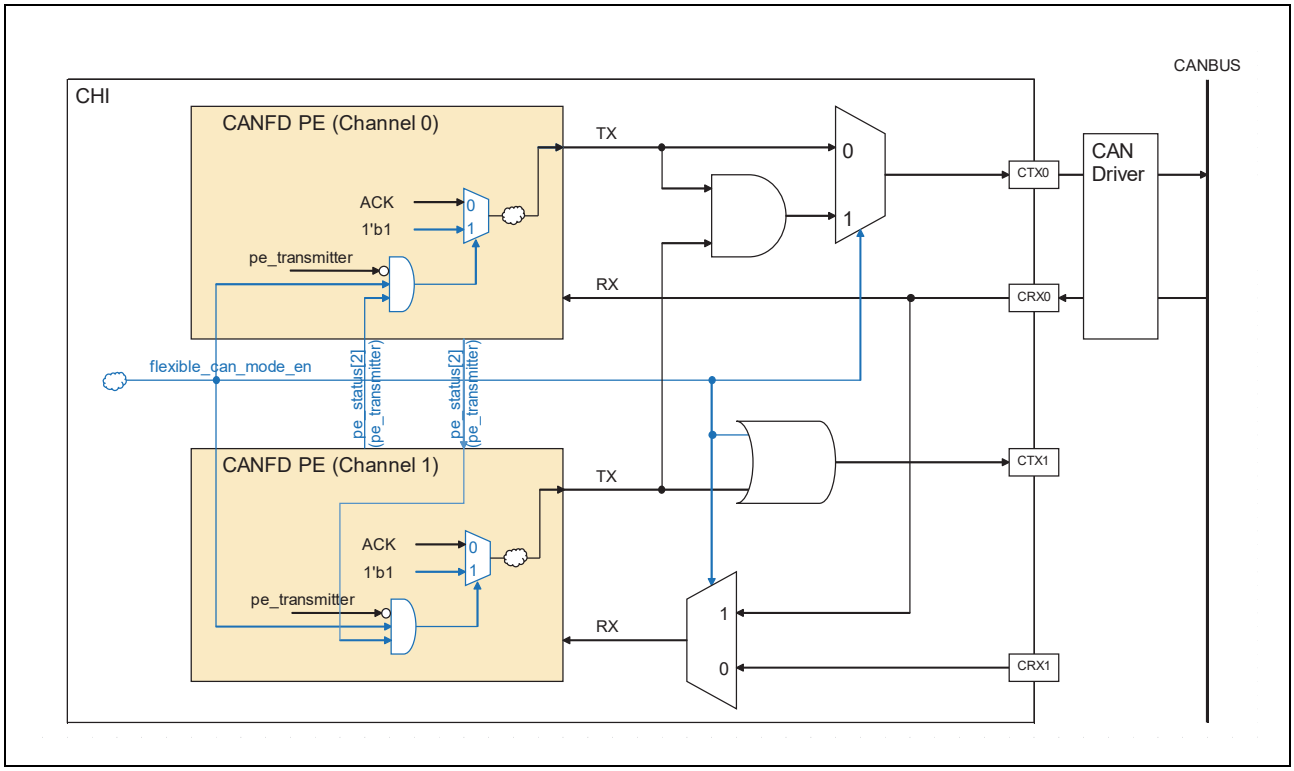


Figure 23.60 Diagram of the Flexible CAN

23.15 Flexible transmission buffer assignment

Each Channel has 64 [for U2A-EVA, U2A16, U2A8] and 32 [for U2A6] transmit buffers by exclusive use.

In order to correspond to an insufficient case by a TXMB, this Channel can rent a maximum of 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] transmit buffers from the next Channel.

The buffer which can be rented becomes every four units by 4 to 32 [for U2A-EVA, U2A16, U2A8] and 16 [for U2A6] buffers.

The pair of Channel in flexible transmission buffer assignment is as follows.

When RSCFDnCFDGFTBAC.FLXMB0 bit is set, Flexible transmission buffer assignment between Channel 0 and Channel 1.

When RSCFDnCFDGFTBAC.FLXMB1 bit is set, Flexible transmission buffer assignment between Channel 2 and Channel 3.

When RSCFDnCFDGFTBAC.FLXMB2 bit is set, Flexible transmission buffer assignment between Channel 4 and Channel 5.

When RSCFDnCFDGFTBAC.FLXMB3 bit is set, Flexible transmission buffer assignment between Channel 6 and Channel 7.

Flexible transmission buffer assignment configured in RSCFDnCFDGFTBAC register and Flexible CAN mode configured in RSCFDnCFDGFCMC register should not be used simultaneously.

Interrupt of the rented buffer is outputted to interrupt of the renting channel.

When using TXQ by the rented buffer, TXQ should only set within the rented range.

The TXMB lent out operates in the mode of the channel to be used.

For example, when the channel 1 is a reset mode, using rented TXMB, transmission of the channel 0 is possible.

Moreover, rented TXMB is subject to the influence of the transmitting status, TX SCAN PROCESS, or transmitting abortion of the rented channel.

In the case of message buffer number priority mode, the priority of TXMB0 is high and the priority of TXMB95 (case which rented 32 buffers) becomes low. Priority of rented TXMB becomes low.

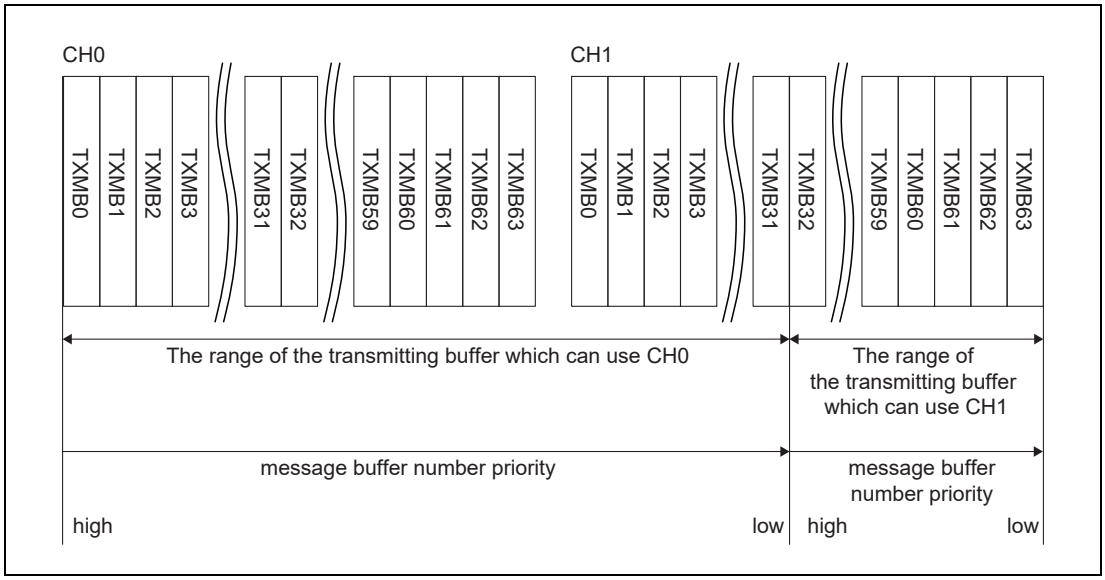


Figure 23.61 The priority of Message Buffer Number Priority Mode (Example of CH0:96 buf/CH1:32 buf)

The example of a rental channel 0 and channel 1 is shown below.

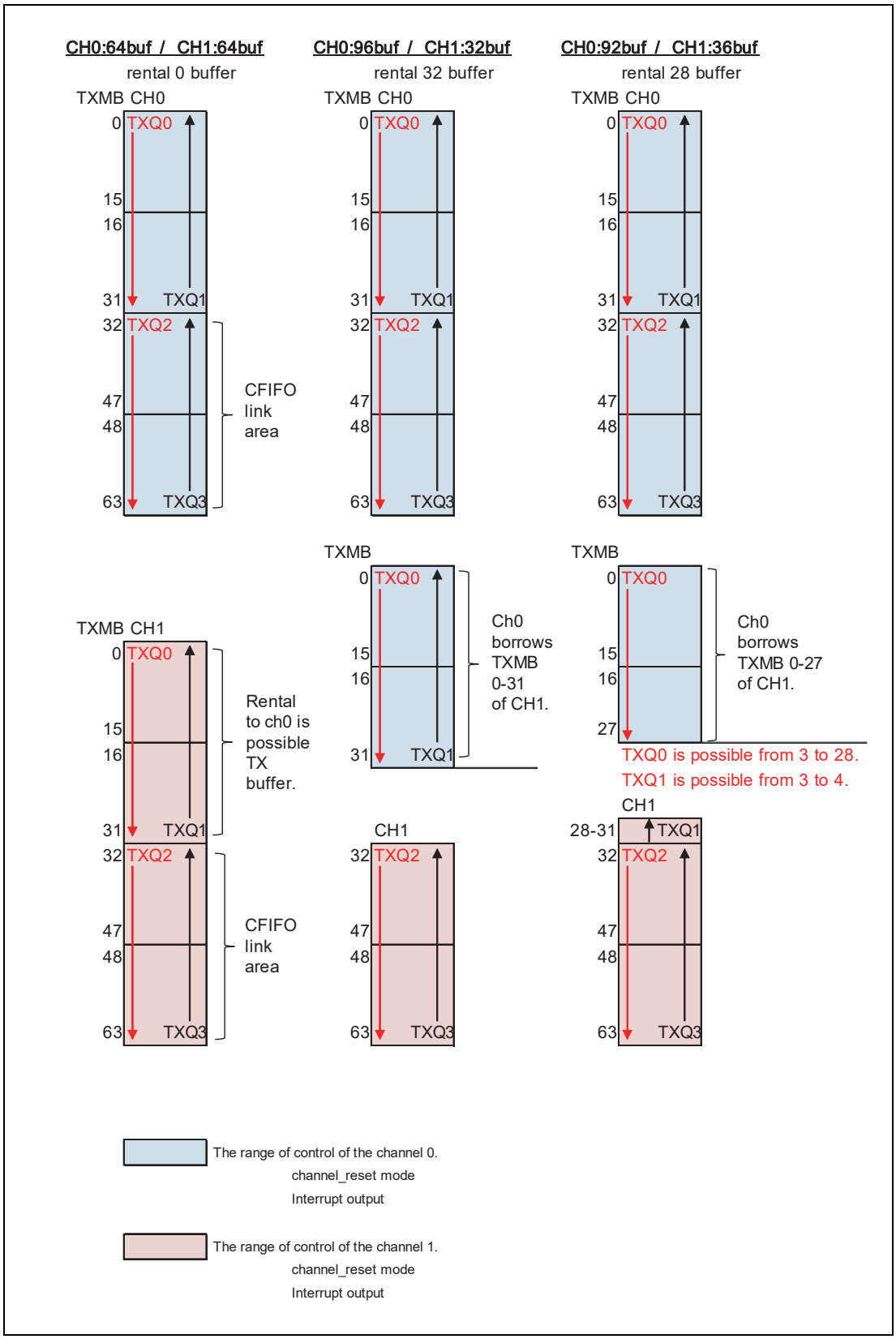


Figure 23.62 Flexible transmission buffer assignment (0, 32, 28 TXMB rental)

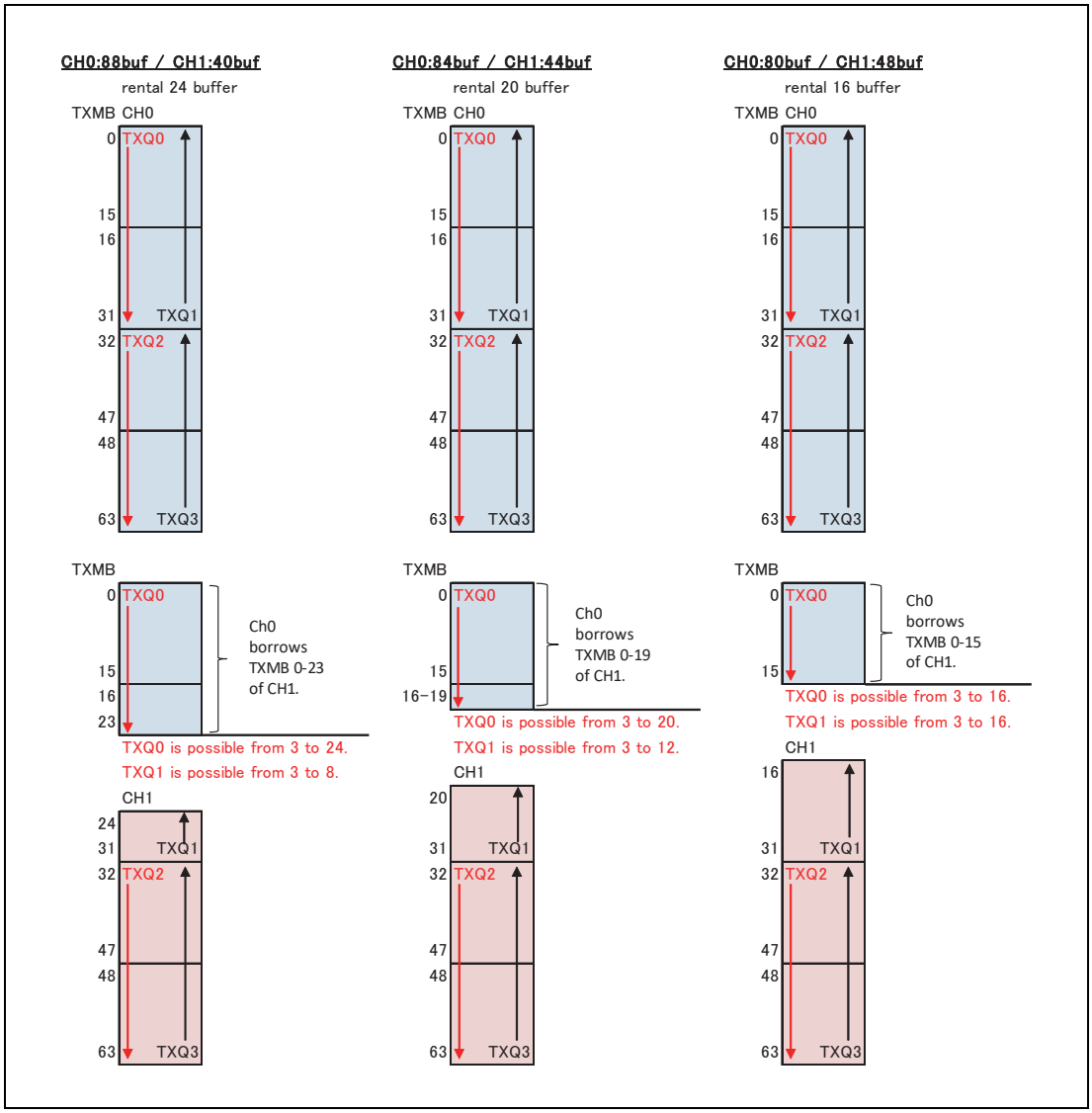


Figure 23.63 Flexible transmission buffer assignment (24, 20, 16 TXMB rental)

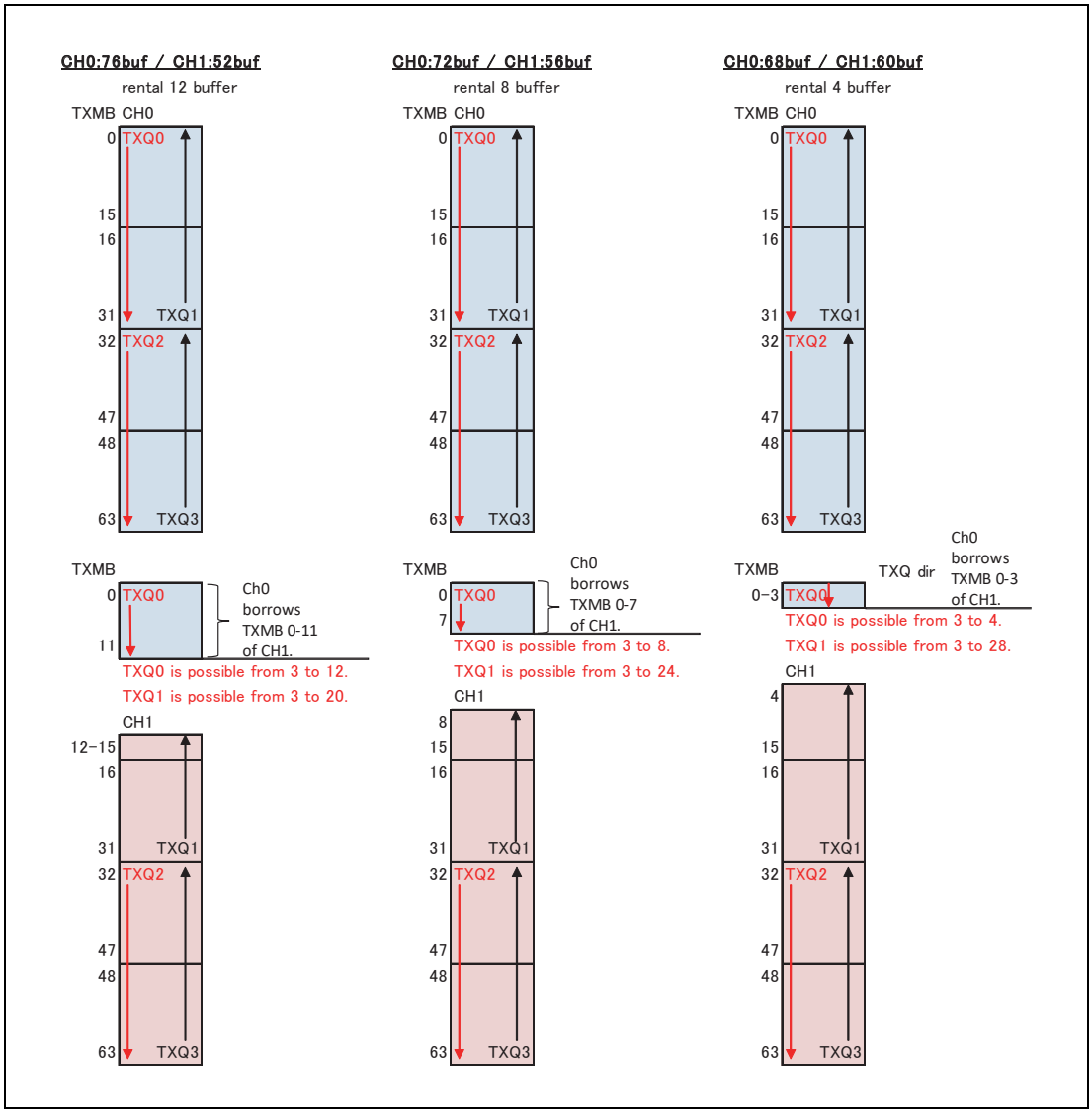


Figure 23.64 Flexible transmission buffer assignment (12, 8, 4 TXMB rental)

Section 24 FlexRay (FLXA)

The FlexRay IP-module performs communication according to the FlexRay protocol specification v2.1. With maximum specified sample clock the bitrate is 10 MBit/s. Additional bus driver (BD) hardware is required for connection to the physical layer.

24.1 Features FLXA for RH850/U2A-EVA

24.1.1 Number of Units

This microcontroller has the following number units of FlexRay.

Table 24.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	FLXAn									

Table 24.2 Index

Index	Description
n	The number of the FlexRay indicated by the letter “n” (n = 0,1).
m	The register number is identified by the index “m”, for example, FLXAnFRESIDm for the even Sync ID register.
p	The flag number is indicated by the letter “p” (p = (m – 1) × 32 to (m × 32 – 1)).

24.1.2 Register Base Addresses

All FlexRay register addresses are given as address offsets to the individual base address <FLXAn_base>.

The base address <FLXAn_base> of each FlexRay is listed in the following table:

Table 24.3 Register Base Addresses

Base Address Name	Base Addresses	Bus Group
<FLXA0_base>	$1002\ 0000_H$	H-Bus Group1
<FLXA1_base>	$1002\ 1000_H$	H-Bus Group2

24.1.3 Clock Supply

FlexRay provide one clock input.

Table 24.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name	Clock for Unit
FLXAn	Bus clock	CLK_HBUS	For H-Bus
	FlexRay sample clock	CLK_HSB	For FlexRay com

24.1.4 Interrupt Requests and Error Notifications

FlexRay can generate the following interrupt requests:

Table 24.5 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
FLXA0					
INTFLXA0LINE0	INTFLXAnLINE0 (n = 0)	Line 0 interrupt for FLXA0	623	—	—
INTFLXA0LINE1	INTFLXAnLINE1 (n = 0)	Line 1 interrupt for FLXA0	624	—	—
INTFLXA0TIM0	INTFLXAnTIM0 (n = 0)	Timer 0 interrupt for FLXA0	625	—	—
INTFLXA0TIM1	INTFLXAnTIM1 (n = 0)	Timer 1 interrupt for FLXA0	626	—	—
INTFLXA0TIM2	INTFLXAnTIM2 (n = 0)	Timer 2 interrupt for FLXA0	627	—	—
INTFLXA0FDA	INTFLXAnFDA (n = 0)	FIFO data available interrupt for FLXA0	628	—	—
INTFLXA0FW	INTFLXAnFW (n = 0)	FIFO transfer warning interrupt for FLXA0	629	—	—
INTFLXA0OW	INTFLXAnOW (n = 0)	Output transfer warning interrupt for FLXA0	630	—	—
INTFLXA0OT	INTFLXAnOT (n = 0)	Output transfer done interrupt for FLXA0	631	—	—
INTFLXA0IQF	INTFLXAnIQF (n = 0)	Input queue full interrupt for FLXA0	632	—	—
INTFLXA0IQE	INTFLXAnIQE (n = 0)	Input queue empty interrupt for FLXA0	633	—	—
FLXA1					
INTFLXA1LINE0	INTFLXAnLINE0 (n = 1)	Line 0 interrupt for FLXA1	634	—	—
INTFLXA1LINE1	INTFLXAnLINE1 (n = 1)	Line 1 interrupt for FLXA1	635	—	—
INTFLXA1TIM0	INTFLXAnTIM0 (n = 1)	Timer 0 interrupt for FLXA1	636	—	—
INTFLXA1TIM1	INTFLXAnTIM1 (n = 1)	Timer 1 interrupt for FLXA1	637	—	—
INTFLXA1TIM2	INTFLXAnTIM2 (n = 1)	Timer 2 interrupt for FLXA1	638	—	—
INTFLXA1FDA	INTFLXAnFDA (n = 1)	FIFO data available interrupt for FLXA1	639	—	—
INTFLXA1FW	INTFLXAnFW (n = 1)	FIFO transfer warning interrupt for FLXA1	640	—	—
INTFLXA1OW	INTFLXAnOW (n = 1)	Output transfer warning interrupt for FLXA1	641	—	—
INTFLXA1OT	INTFLXAnOT (n = 1)	Output transfer done interrupt for FLXA1	642	—	—
INTFLXA1IQF	INTFLXAnIQF (n = 1)	Input queue full interrupt for FLXA1	643	—	—
INTFLXA1IQE	INTFLXAnIQE (n = 1)	Input queue empty interrupt for FLXA1	644	—	—

This module has no error notifications.

24.1.5 Reset Sources

FLXAn reset sources are listed in the following table. FLXAn is initialized by these reset sources.

Table 24.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
FLXAn	All registers	√	√	√	√	√	√	—

24.1.6 External Input/Output Signals

Table 24.7 shows the pin information.

Table 24.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signals
FLXA0		
flx0_rxda_extfxr	Channel A receive data input	FLXA0RXDA
flx0_rxdb_extfxr	Channel B receive data input	FLXA0RXDB
flx0_fxr_txda	Channel A transmit data output	FLXA0TXDA
flx0_fxr_txdb	Channel B transmit data output	FLXA0TXDB
flx0_fxr_txena_n	Channel A transmit enable	FLXA0TXENA
flx0_fxr_txenb_n	Channel B transmit enable	FLXA0TXENB
flx0_stpwt_extfxr	Stop watch trigger input	FLXA0STPWT
FLXA1		
flx1_rxda_extfxr	Channel A receive data input	FLXA1RXDA
flx1_rxdb_extfxr	Channel B receive data input	FLXA1RXDB
flx1_fxr_txda	Channel A transmit data output	FLXA1TXDA
flx1_fxr_txdb	Channel B transmit data output	FLXA1TXDB
flx1_fxr_txena_n	Channel A transmit enable	FLXA1TXENA
flx1_fxr_txenb_n	Channel B transmit enable	FLXA1TXENB
flx1_stpwt_extfxr	Stop watch trigger input	FLXA1STPWT

24.1.7 Combinations of Pins and Ports

Combinations of FlexRay pins and ports are listed in the following table.

Table 24.8 Combinations of Pins and Ports

Function	Pin Name	Port Name	
		Group 1	Group 2
FLXA0	FLXA0RXDA	P20_14	P20_2
	FLXA0RXDB	P10_0	P20_3
	FLXA0TXDA	P10_4	P20_7
	FLXA0TXDB	P10_3	P20_6
	FLXA0TXENA	P10_2	P20_5
	FLXA0TXENB	P10_5	P20_8
	FLXA0STPWT	P10_1	P20_4
FLXA1	FLXA1RXDA	P21_7	—
	FLXA1RXDB	P21_2	—
	FLXA1TXDA	P21_5	—
	FLXA1TXDB	P21_3	—
	FLXA1TXENA	P21_6	—
	FLXA1TXENB	P21_1	—
	FLXA1STPWT	P17_5	—

n = 0: HBG92
n = 1: HBG93

24.2 Overview

24.2.1 Functional Overview

For communication on a FlexRay network, individual message buffers with up to 254 data bytes are configurable. The message buffer is a Message RAM that is configurable up to 128 message buffers. All functions concerning the handling of messages are implemented in the Message Handler. Those functions are the acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule as well as providing message status information.

The register set of the FlexRay IP-module can be accessed directly by an external Host via the module's Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, to access the Message RAM via Input / Output Buffer; and to control the data transfer between the Message RAM and the Local RAM/Cluster RAM.

The FlexRay IP-module supports the following features:

Item	Specification
Communication	Conformance with FlexRay protocol specification v2.1
Data transfer rate	Up to 10 M bit/s on each channel
FlexRay channels	2 (channels A and B)
Message buffer	Up to 128 message buffers are configurable. Message buffers are configurable with different payload length Each message buffer can be configured as a part of receive buffer, transfer buffer or receive FIFO. Filtering for slot counter, cycle counter or channel.
Message RAM	8-Kbyte message RAM can be configured as below. • 128 message buffers with up to 48-byte data section • 30 message buffers with 254-byte data section
FIFO	One configurable receive FIFO
Message Buffer Access	Access by the host CPU through input/output buffer Input buffer: a message transferred to the message RAM is retained. Output buffer: a message read from the message RAM is retained.
	Access by data transfer function Input transfer: the contents of the message buffer is transferred from the Local RAM/Cluster RAM to the message RAM by a request of the CPU Output transfer: the contents of the message buffer is transferred automatically to the Local RAM/Cluster RAM from the message RAM
Network management	Supported
Interrupts	Maskable module interrupts
Timer	Two absolute timers One relative timer One stop watch timer

24.2.2 Block Diagram

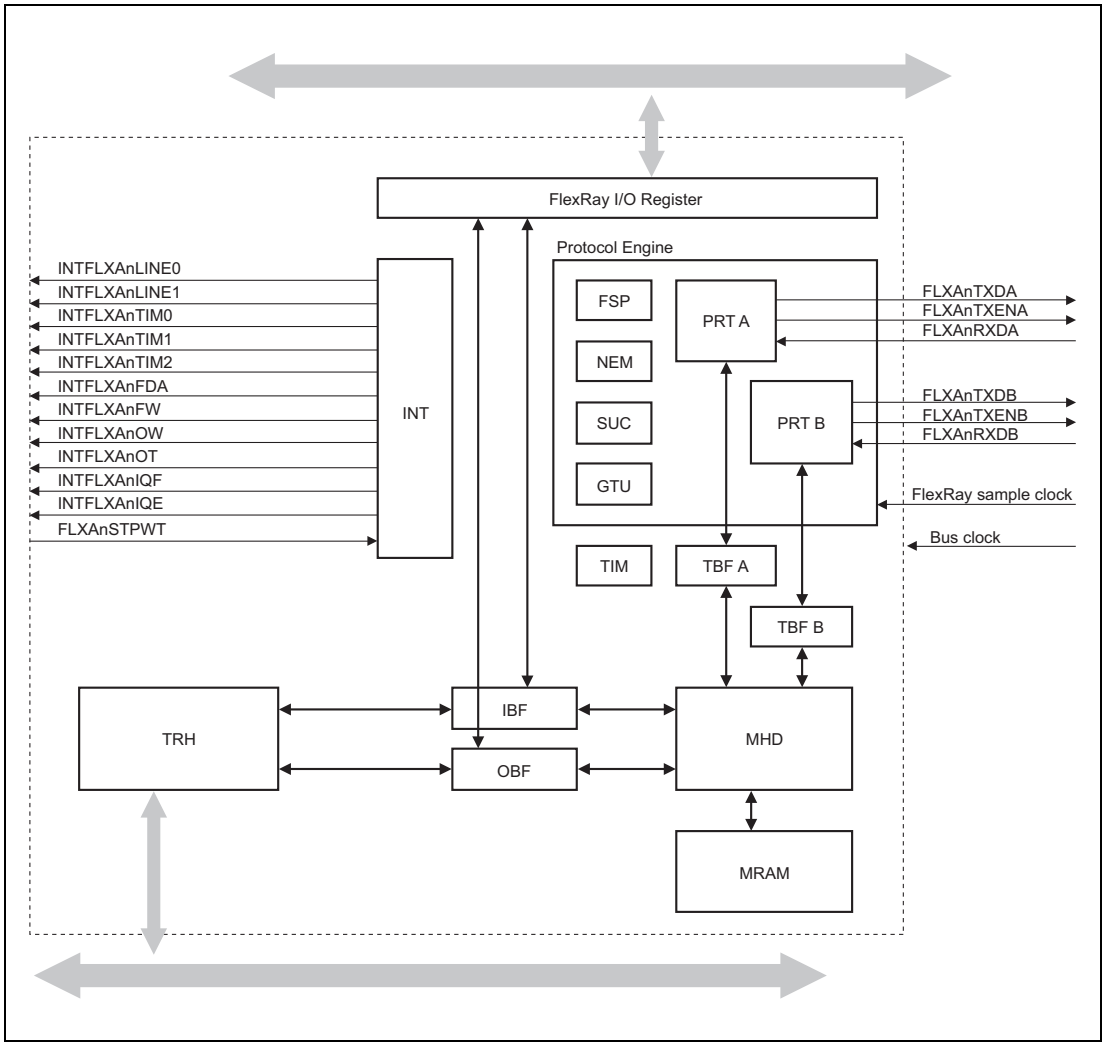


Figure 24.1 FlexRay IP Block Diagram

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The FlexRay Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- TBFRAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

TBFRAM (TBF A/B)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the TBFRAMs for intermediate message storage and to the physical layer via bus driver (BD).

They perform the following functionality:

- Control and check of bit timing
- Reception / transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC
- Interfacing to bus driver

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick (µT)
- Generation of macrotick (MT)
- Fault tolerant clock synchronization by FTM algorithm
  - rate correction
  - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Control controls the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

Frame and Symbol Processing (FSP)

The Frame and Symbol Processing controls the following functions:

- Checks the correct timing of frames and symbols
- Tests the syntactical and semantical correctness of received frames
- Sets the slot status flags

Network Management (NEM)

Handles the network management vector.

Interrupt Control (INT)

The Interrupt Controller performs the following functions:

- Provides error and status interrupt flags
- Enable / disable interrupt sources
- Assignment of interrupt sources to one of the two general module interrupt lines
- Enable / disable module interrupt lines

Timer (TIM)

The Timer module includes the following macrotick timer:

- two absolute timers
- one relative timer
- one stop watch timer

Transfer Handler (TRH)

Handles the data transfer between Local RAM/Cluster RAM and FlexRay module.

The Transfer Handler supports the following transfer types:

- Transfer of buffer configuration data from the Local RAM/Cluster RAM to the Message RAM
- Transfer of payload data for transmission buffers from the Local RAM/Cluster RAM to the Message RAM
- Transfer of buffer configuration data and payload data for transmission buffer from the Local RAM/Cluster RAM to the Message RAM
- Automatic transfer of payload data from receive buffer to the Local RAM/Cluster RAM upon frame reception
- Automatic transfer of payload data, buffer configuration data and message buffer status data from receive buffer to the Local RAM/Cluster RAM upon frame reception
- Automatic transfer of buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Cluster RAM in response to slot status update
- Manual transfer of payload data, buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Cluster RAM.

24.3 Registers

24.3.1 List of Registers

The FlexRay module allocates an address space as shown in **Table 24.9**.

Within this specification the “Values after reset” refers to the reset of the microcontroller. For registers in the address range <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ the “Value after reset” are also applicable when the software reset of the FlexRay module (using bit FLXAnFROC.OE) is applied.

The addresses in this specification are listed as offsets from a base address. The base address <FLXAn_base> must thus be added to the addresses.

For <FLXAn_base>, refer to **Section 24.1.2, Register Base Addresses**.

Table 24.9 List of Registers (1/4)

Register Name	Symbol	Value after Reset	Address	Access Size	Access Protection	
					HBG	Other
FlexRay Operation Control Register	FLXAnFROC	0000 0000_H	<FLXAn_base> + 0004_H	8, 16, 32	*1	—
FlexRay Operation Status Register	FLXAnFROS	0000 0000_H	<FLXAn_base> + $000C_H$	8, 16, 32	*1	—
FlexRay Lock Register	FLXAnFRLCK	0000 0000_H	<FLXAn_base> + $001C_H$	8, 16, 32	*1	—
FlexRay Error Interrupt Register	FLXAnFREIR	0000 0000_H	<FLXAn_base> + 0020_H	8, 16, 32	*1	—
FlexRay Status Interrupt Register	FLXAnFRSIR	0000 0000_H	<FLXAn_base> + 0024_H	8, 16, 32	*1	—
FlexRay Error Interrupt Line Select	FLXAnFREILS	0000 0000_H	<FLXAn_base> + 0028_H	8, 16, 32	*1	—
FlexRay Status Interrupt Line Select	FLXAnFRSILS	0303 $FFFF_H$	<FLXAn_base> + $002C_H$	8, 16, 32	*1	—
FlexRay Error Interrupt Enable Set Register	FLXAnFREIES	0000 0000_H	<FLXAn_base> + 0030_H	8, 16, 32	*1	—
FlexRay Error Interrupt Enable Reset Register	FLXAnFREIER	0000 0000_H	<FLXAn_base> + 0034_H	8, 16, 32	*1	—
FlexRay Status Interrupt Enable Set Register	FLXAnFRSIES	0000 0000_H	<FLXAn_base> + 0038_H	8, 16, 32	*1	—
FlexRay Status Interrupt Enable Reset Register	FLXAnFRSIER	0000 0000_H	<FLXAn_base> + $003C_H$	8, 16, 32	*1	—
FlexRay Interrupt Line Enable Register	FLXAnFRILE	0000 0000_H	<FLXAn_base> + 0040_H	8, 16, 32	*1	—
FlexRay Timer 0 Configuration Register	FLXAnFRT0C	0000 0000_H	<FLXAn_base> + 0044_H	8, 16, 32	*1	—
FlexRay Timer 1 Configuration Register	FLXAnFRT1C	0002 0000_H	<FLXAn_base> + 0048_H	8, 16, 32	*1	—
FlexRay Stop Watch Register 1	FLXAnFRSTPW1	0000 0000_H	<FLXAn_base> + $004C_H$	8, 16, 32	*1	—
FlexRay Stop Watch Register 2	FLXAnFRSTPW2	0000 0000_H	<FLXAn_base> + 0050_H	8, 16, 32	*1	—
FlexRay SUC Configuration Register 1	FLXAnFRSUCC1	0C40 1080_H	<FLXAn_base> + 0080_H	8, 16, 32	*1	—
FlexRay SUC Configuration Register 2	FLXAnFRSUCC2	0100 0504_H	<FLXAn_base> + 0084_H	8, 16, 32	*1	—
FlexRay SUC Configuration Register 3	FLXAnFRSUCC3	0000 0011_H	<FLXAn_base> + 0088_H	8, 16, 32	*1	—
FlexRay NEM Configuration Register	FLXAnFRNEMC	0000 0000_H	<FLXAn_base> + $008C_H$	8, 16, 32	*1	—
FlexRay PRT Configuration Register 1	FLXAnFRPRTC1	084C 0633_H	<FLXAn_base> + 0090_H	8, 16, 32	*1	—
FlexRay PRT Configuration Register 2	FLXAnFRPRTC2	0F2D $0A0E_H$	<FLXAn_base> + 0094_H	8, 16, 32	*1	—
FlexRay MHD Configuration Register	FLXAnFRMHDC	0000 0000_H	<FLXAn_base> + 0098_H	8, 16, 32	*1	—
FlexRay GTU Configuration Register 1	FLXAnFRGTUC1	0000 0280_H	<FLXAn_base> + $00A0_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 2	FLXAnFRGTUC2	0002 $000A_H$	<FLXAn_base> + $00A4_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 3	FLXAnFRGTUC3	0202 0000_H	<FLXAn_base> + $00A8_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 4	FLXAnFRGTUC4	0008 0007_H	<FLXAn_base> + $00AC_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 5	FLXAnFRGTUC5	0E00 0000_H	<FLXAn_base> + $00B0_H$	8, 16, 32	*1	—

Table 24.9 List of Registers (2/4)

Register Name	Symbol	Value after Reset	Address	Access Size	Access Protection	
					HBG	Other
FlexRay GTU Configuration Register 6	FLXAnFRGTUC6	0002 0000$_H$	<FLXAn_base> + 00B4$_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 7	FLXAnFRGTUC7	0002 0004$_H$	<FLXAn_base> + 00B8$_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 8	FLXAnFRGTUC8	0000 0002$_H$	<FLXAn_base> + 00BC$_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 9	FLXAnFRGTUC9	0000 0101$_H$	<FLXAn_base> + 00C0$_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 10	FLXAnFRGTUC10	0002 0005$_H$	<FLXAn_base> + 00C4$_H$	8, 16, 32	*1	—
FlexRay GTU Configuration Register 11	FLXAnFRGTUC11	0000 0000$_H$	<FLXAn_base> + 00C8$_H$	8, 16, 32	*1	—
FlexRay CC Status Vector Register	FLXAnFRCCSV	0010 4000$_H$	<FLXAn_base> + 0100$_H$	8, 16, 32	*1	—
FlexRay CC Error Vector Register	FLXAnFRCCEV	0000 0000$_H$	<FLXAn_base> + 0104$_H$	8, 16, 32	*1	—
FlexRay Slot Counter Value Register	FLXAnFRSCV	0000 0000$_H$	<FLXAn_base> + 0110$_H$	8, 16, 32	*1	—
FlexRay Macrotick and Cycle Counter Value Register	FLXAnFRMTCCV	0000 0000$_H$	<FLXAn_base> + 0114$_H$	8, 16, 32	*1	—
FlexRay Rate Correction Value Register	FLXAnFRRCV	0000 0000$_H$	<FLXAn_base> + 0118$_H$	8, 16, 32	*1	—
FlexRay Offset Correction Value Register	FLXAnFROCV	0000 0000$_H$	<FLXAn_base> + 011C$_H$	8, 16, 32	*1	—
FlexRay Sync Frame Status Register	FLXAnFRSFS	0000 0000$_H$	<FLXAn_base> + 0120$_H$	8, 16, 32	*1	—
FlexRay Symbol Window and NIT Status Register	FLXAnFRSWNIT	0000 0000$_H$	<FLXAn_base> + 0124$_H$	8, 16, 32	*1	—
FlexRay Aggregated Channel Status Register	FLXAnFRACS	0000 0000$_H$	<FLXAn_base> + 0128$_H$	8, 16, 32	*1	—
FlexRay Even Sync ID Register m (m = 1 to 15)	FLXAnFRESIDm (m = 1 to 15)	0000 0000$_H$	<FLXAn_base> + 0130$_H$ to <FLXAn_base> + 0168$_H$ (<FLXAn_base> + 0130$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Odd Sync ID Register m (m = 1 to 15)	FLXAnFROSIDm (m = 1 to 15)	0000 0000$_H$	<FLXAn_base> + 0170$_H$ to <FLXAn_base> + 01A8$_H$ (<FLXAn_base> + 0170$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Network Management Vector Register m (m = 1 to 3)	FLXAnFRNMVm (m = 1 to 3)	0000 0000$_H$	<FLXAn_base> + 01B0$_H$ to <FLXAn_base> + 01B8$_H$ (<FLXAn_base> + 01B0$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Message RAM Configuration Register	FLXAnFRMRC	0180 0000$_H$	<FLXAn_base> + 0300$_H$	8, 16, 32	*1	—
FlexRay FIFO Rejection Filter Register	FLXAnFRFRF	0180 0000$_H$	<FLXAn_base> + 0304$_H$	8, 16, 32	*1	—
FlexRay FIFO Rejection Filter Mask Register	FLXAnFRFRFM	0000 0000$_H$	<FLXAn_base> + 0308$_H$	8, 16, 32	*1	—
FlexRay FIFO Critical Level Register	FLXAnFRFCL	0000 0080$_H$	<FLXAn_base> + 030C$_H$	8, 16, 32	*1	—
FlexRay Message Handler Status Register	FLXAnFRMHDS	0000 0080$_H$	<FLXAn_base> + 0310$_H$	8, 16, 32	*1	—
FlexRay Last Dynamic Transmit Slot Register	FLXAnFRLDTS	0000 0000$_H$	<FLXAn_base> + 0314$_H$	8, 16, 32	*1	—
FlexRay FIFO Status Register	FLXAnFRFSR	0000 0000$_H$	<FLXAn_base> + 0318$_H$	8, 16, 32	*1	—
FlexRay Message Handler Constraints Flags Register	FLXAnFRMHDF	0000 0000$_H$	<FLXAn_base> + 031C$_H$	8, 16, 32	*1	—

Table 24.9 List of Registers (3/4)

Register Name	Symbol	Value after Reset	Address	Access Size	Access Protection	
					HBG	Other
FlexRay Transmission Register m (m= 1 to 4)	FLXAnFRTXRQm (m = 1 to 4)	0000 0000$_H$	<FLXAn_base> + 0320$_H$ to <FLXAn_base> + 032C$_H$ (<FLXAn_base> + 0320$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay New Data Register m (m = 1 to 4)	FLXAnFRNDATm (m = 1 to 4)	0000 0000$_H$	<FLXAn_base> + 0330$_H$ to <FLXAn_base> + 033C$_H$ (<FLXAn_base> + 0330$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Message Buffer Status Changed Register m (m = 1 to 4)	FLXAnFRMBSCm (m = 1 to 4)	0000 0000$_H$	<FLXAn_base> + 0340$_H$ to <FLXAn_base> + 034C$_H$ (<FLXAn_base> + 0340$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Write Data Section Register m (m = 1 to 64)	FLXAnFRWRDSm (m = 1 to 64)	0000 0000$_H$	<FLXAn_base> + 0400$_H$ to <FLXAn_base> + 04FC$_H$ (<FLXAn_base> + 0400$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Write Header Section Register 1	FLXAnFRWRHS1	0000 0000$_H$	<FLXAn_base> + 0500$_H$	8, 16, 32	*1	—
FlexRay Write Header Section Register 2	FLXAnFRWRHS2	0000 0000$_H$	<FLXAn_base> + 0504$_H$	8, 16, 32	*1	—
FlexRay Write Header Section Register 3	FLXAnFRWRHS3	0000 0000$_H$	<FLXAn_base> + 0508$_H$	8, 16, 32	*1	—
FlexRay Input Buffer Command Mask Register	FLXAnFRIBCM	0000 0000$_H$	<FLXAn_base> + 0510$_H$	8, 16, 32	*1	—
FlexRay Input Buffer Command Request Register	FLXAnFRIBCR	0000 0000$_H$	<FLXAn_base> + 0514$_H$	8, 16, 32	*1	—
FlexRay Read Data Section Register m (m = 1 to 64)	FLXAnFRRDDSm (m = 1 to 64)	0000 0000$_H$	<FLXAn_base> + 0600$_H$ to <FLXAn_base> + 06FC$_H$ (<FLXAn_base> + 0600$_H$ + (m–1) × 4)	8, 16, 32	*1	—
FlexRay Read Header Section Register 1	FLXAnFRRDHS1	0000 0000$_H$	<FLXAn_base> + 0700$_H$	8, 16, 32	*1	—
FlexRay Read Header Section Register 2	FLXAnFRRDHS2	0000 0000$_H$	<FLXAn_base> + 0704$_H$	8, 16, 32	*1	—
FlexRay Read Header Section Register 3	FLXAnFRRDHS3	0000 0000$_H$	<FLXAn_base> + 0708$_H$	8, 16, 32	*1	—
FlexRay Message Buffer Status Register	FLXAnFRMBS	0000 0000$_H$	<FLXAn_base> + 070C$_H$	8, 16, 32	*1	—
FlexRay Output Buffer Command Mask Register	FLXAnFROBCM	0000 0000$_H$	<FLXAn_base> + 0710$_H$	8, 16, 32	*1	—
FlexRay Output Buffer Command Request Register	FLXAnFROBCR	0000 0000$_H$	<FLXAn_base> + 0714$_H$	8, 16, 32	*1	—
FlexRay Input Transfer Configuration Register	FLXAnFRITC	0000 0000$_H$	<FLXAn_base> + 0800$_H$	8, 16, 32	*1	—
FlexRay Output Transfer Configuration Register	FLXAnFROTC	0000 0000$_H$	<FLXAn_base> + 0804$_H$	8, 16, 32	*1	—
FlexRay Input pointer table Base Address Register	FLXAnFRIBA	0000 0000$_H$	<FLXAn_base> + 0808$_H$	8, 16, 32	*1	—
FlexRay FIFO pointer table Base Address Register	FLXAnFRFBA	0000 0000$_H$	<FLXAn_base> + 080C$_H$	8, 16, 32	*1	—
FlexRay Output pointer table Base Address Register	FLXAnFROBA	0000 0000$_H$	<FLXAn_base> + 0810$_H$	8, 16, 32	*1	—
FlexRay Input Queue Control Register	FLXAnFRIQC	0000 0000$_H$	<FLXAn_base> + 0814$_H$	8, 16, 32	*1	—

Table 24.9 List of Registers (4/4)

Register Name	Symbol	Value after Reset	Address	Access Size	Access Protection	
					HBG	Other
FlexRay User Input transfer Request Register	FLXAnFRUIR	0000 0000_H	<FLXAn_base> + 0818_H	8, 16, 32	*1	—
FlexRay User Output transfer Request Register	FLXAnFRUOR	0000 0000_H	<FLXAn_base> + $081C_H$	8, 16, 32	*1	—
FlexRay Input Transfer Status Register	FLXAnFRITS	0000 0000_H	<FLXAn_base> + 0820_H	8, 16, 32	*1	—
FlexRay Output Transfer Status Register	FLXAnFROTS	0000 0000_H	<FLXAn_base> + 0824_H	8, 16, 32	*1	—
FlexRay Access Error Status Register	FLXAnFRAES	0000 0000_H	<FLXAn_base> + 0828_H	8, 16, 32	*1	—
FlexRay Access Error Address Register	FLXAnFRAEA	0000 0000_H	<FLXAn_base> + $082C_H$	8, 16, 32	*1	—
FlexRay Message Data Available Register m (m = 0 to 3)	FLXAnFRDAm (m = 0 to 3)	0000 0000_H	<FLXAn_base> + 0830_H to <FLXAn_base> + $083C_H$ (<FLXAn_base> + 0830_H + (m × 4))	8, 16, 32	*1	—
FlexRay Timer 2 Configuration Register	FLXAnFRT2C	0000 0000_H	<FLXAn_base> + 0844_H	8, 16, 32	*1	—

Note 1. n = 0: HBG92
n = 1: HBG93

24.3.2 FlexRay Operation register

24.3.2.1 FLXAnFROC — FlexRay Operation Control Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IE	T1IE	T0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OEP	—	—	—	—	—	BEC	OE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 24.10 FLXAnFROC Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IE	Timer 2 interrupt enable Bit 0: Disabled 1: Enabled
17	T1IE	Timer 1 interrupt enable Bit 0: Disabled 1: Enabled
16	T0IE	Timer 0 interrupt enable Bit 0: Disabled 1: Enabled
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	OEP	Operation Enable bit Protection Bit 0: OE is unprotected 1: OE is protected
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	BEC	Byte Endian Control Bit 0: Little endian 1: Big endian
0	OE	Operation Enable Bit 0: Operation disabled, software reset of the FlexRay module 1: Operation enabled

(1) FLXAnFROC.T2IE

Timer 2 interrupt enable bit

This bit controls the timer 2 interrupt.

0: Disabled

No interrupt will be requested and the timer 2 interrupt line will be released if pending.

1: Enabled

Timer 2 interrupt will be asserted when FLXAnFROS.T2IS is 1.

(2) FLXAnFROC.T1IE

Timer 1 interrupt enable bit

The user should only set this bit to 1 when timer 1 interrupt is not enabled in the FlexRay Status interrupt enable register (bit FLXAnFRSIES.TI1E should be 0).

This bit controls the timer 1 interrupt.

0: Disabled

No interrupt will be requested and the timer 1 interrupt line will be released if pending.

1: Enabled

Timer 1 interrupt will be asserted when bit FLXAnFROS.T1IS is 1.

(3) FLXAnFROC.T0IE

Timer 0 interrupt enable bit

The user should only set this bit to 1 when timer 0 interrupt is not enabled in the FlexRay Status interrupt enable register (bit FLXAnFRSIES.TI0E should be 0).

This bit controls the timer 0 interrupt.

0: Disabled

No interrupt will be requested and the timer 0 interrupt line will be released if pending.

1: Enabled

Timer 0 interrupt will be asserted when bit FLXAnFROS.T0IS is 1.

(4) FLXAnFROC.OEP

Operation enable bit protection bit

This bit protects against unintended write access to the OE bit.

0: OE bit is unprotected

Write access to the OE bit is enabled

1: OE bit is protected

Write access to the OE bit is disabled

(5) FLXAnFROC.BEC

Byte endian control bit

The user should only change this bit when FLXAnFROS.OS is '1'.

This bit controls the byte order on reading and writing the FlexRay network management vector register (FLXAnFRNMVx), FlexRay write data section (FLXAnFRWRDSx) and FlexRay read data section (FLXAnFRRDDSx). This bit also controls the byte order when reading or writing FlexRay payload data using the data transfer function.

For details about the byte alignment, refer to **Section 24.4.17, Byte Alignment**.

0: Little endian

Byte alignment in FLXAnFRNMVx, FLXAnFRWRDSx and FLXAnFRRDDSx is in little endian style.

1: Big endian

Byte alignment in FLXAnFRNMVx, FLXAnFRWRDSx and FLXAnFRRDDSx is in big endian style.

(6) FLXAnFROC.OE

Operation enable bit

The user can only write to this bit when bit FLXAnFROC.OEP is 0.

The user should only write this bit with 0 when bit FLXAnFROS.OS is 1.

The user should only write this bit with 1 when bit FLXAnFROS.OS is 0 and the FlexRay sample clock is enabled.

This bit controls the operation state and serves the software reset of the FlexRay module. The operation status bit (FLXAnFROS.OS) indicates whether the FlexRay module is in reset state or not.

0: Operation disabled, software reset of the FlexRay module

Forcibly moves the FlexRay module to its reset state, whatever the state of the FlexRay module is.

1: Operation enabled

Reset state of the FlexRay module is released.

24.3.2.2 FLXAnFROS — FlexRay Operation Status Register

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $000C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IS	T1IS	T0IS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.11 FLXAnFROS Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IS	Timer 2 Interrupt Status Bit 0: Timer 2 has not matched the conditions configured in the FLXAnFRT2C register 1: Timer 2 matched the conditions configured in the FLXAnFRT2C register
17	T1IS	Timer 1 Interrupt Status Bit 0: Timer 1 has not matched the conditions configured in the FLXAnFRT1C register 1: Timer 1 matched the conditions configured in the FLXAnFRT1C register
16	T0IS	Timer 0 Interrupt Status Bit 0: Timer 0 has not matched the conditions configured in the FLXAnFRT0C register 1: Timer 0 matched the conditions configured in the FLXAnFRT0C register
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OS	Operation Status Bit 0: Operation disabled, reset state 1: Operation enabled

(1) FLXAnFROS.T2IS

Timer 2 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 2 has matched the state configured in the FLXAnFRT2C register.

When the FLXAnFROC.T2IE bit is enabled and the FLXAnFROS.T2IS bit is set to 1, the timer 2 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T2IS bit.
- This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state matches the state configured in the FLXAnFRT2C register.

(2) FLXAnFROS.T1IS

Timer 1 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 1 has matched the state configured in the FLXAnFRT1C register.

When the FLXAnFROC.T1IE bit is enabled and the FLXAnFROS.T1IS bit is set to 1, the timer 1 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T1IS bit.
  This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state becomes the state configured in the FLXAnFRT1C register.

(3) FLXAnFROS.T0IS

Timer 0 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the timer 0 has matched the state configured in the FLXAnFRT0C register.

When the FLXAnFROC.T0IE bit is enabled and the FLXAnFROS.T0IS bit is set to 1, the timer 0 interrupt is generated.

[Clearing condition]

- This bit is cleared by writing 1 to the FLXAnFROS.T0IS bit.
- This bit is cleared when the FLXAnFROS.OS bit changes from 1 to 0.

[Setting condition]

- This bit is set when the state becomes the state configured in the FLXAnFRT0C register.

(4) FLXAnFROS.OS

Operation Status Bit

This bit represents if the FlexRay module is in the reset or the operation state.

When bit FLXAnFROS.OS is 0 the FlexRay module gets initialized and registers mapped to the address area <FLXAn_base>+ 0010_H to <FLXAn_base> + $0FFF_H$ cannot be accessed; read access from these registers will return undefined data.

When bit FLXAnFROS.OS is 1 it is possible to access to the address area <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ and to perform FlexRay communication.

When bit FLXAnFROS.OS changes from 0 to 1 all registers in the address range <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ are set to the “Values after reset”.

[Clearing condition]

- When bit FLXAnFROC.OE is set to 0. It takes up to two peripheral bus clock cycles until bit FLXAnFROS.OS is set to 0.

[Setting condition]

- When bit FLXAnFROC.OE is set to 1 it takes up to four peripheral clock cycles of the clock with the lower frequency out of the FlexRay sample clock and peripheral bus clock until bit FLXAnFROS.OS is set to 1.

24.3.3 Special Registers

24.3.3.1 FLXAnFRLCK — FlexRay Lock Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $001C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.12 FLXAnFRLCK Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CLK[7:0]	Configuration Lock Key Bit

(1) FLXAnFRLCK.CLK

Configuration Lock Key Bit

The Lock Register is write-only. Reading the register will return $0000\ 0000_H$.

To leave CONFIG state by writing bits FLXAnFRSUCC1.CMD[3:0] (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the FLXAnFRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: Bits FLXAnFRLCK.CLK[7:0] = "$1100\ 1110_B$" (CE_H)

Second write: Bits FLXAnFRLCK.CLK[7:0] = "$0011\ 0001_B$" (31_H)

Third write: Bits FLXAnFRSUCC1.CMD[3:0]

CAUTION

In case that the Host uses 8/16-bit accesses to write the listed bit fields, the user has to ensure that no "dummy accesses" e.g. to the remaining register bytes / words are inserted by the compiler.

24.3.4 Interrupt Registers

24.3.4.1 FLXAnFREIR — FlexRay Error Interrupt Register

Do not rewrite this register using bit manipulation instructions.

The flags are set when the CC detects one of the listed error conditions. The flags remain set until the Host clears them.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0020_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABB	LTVB	EDB	—	—	—	—	—	TABA	LTVA	EDA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHF	IOBA	IIBA	EFA	RFO	AERR	CCL	CCF	SFO	SFBM	CNA	PEMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.13 FLXAnFREIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABB	Transmission Across Boundary Channel B Flag 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B
25	LTVB	Latest Transmit Violation Channel B Flag 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B
24	EDB	Error Detected on Channel B Flag 0: No error detected on channel B 1: Error detected on channel B
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABA	Transmission Across Boundary Channel A Flag 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A
17	LTVA	Latest Transmit Violation Channel A Flag 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A
16	EDA	Error Detected on Channel A Flag 0: No error detected on channel A 1: Error detected on channel A
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHF	Message Handler Constraints Flag 0: No Message Handler failure detected 1: Message Handler failure detected

Table 24.13 FLXAnFREIR Register Contents (2/2)

Bit Position	Bit Name	Function
10	IOBA	Illegal Output buffer Access Flag 0: No illegal Host access to Output Buffer occurred 1: Illegal Host access to Output Buffer occurred
9	IIBA	Illegal Input Buffer Access Flag 0: No illegal Host access to Input Buffer occurred 1: Illegal Host access to Input Buffer occurred
8	EFA	Empty FIFO Access Flag 0: No Host access to empty FIFO occurred 1: Host access to empty FIFO occurred
7	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has occurred
6	AERR	Access error flag Flag 0: Access error is not detected. 1: Access error is detected.
5	CCL	CHI Command Locked Flag 0: CHI command accepted 1: CHI command not accepted
4	CCF	Clock Correction Failure Flag 0: No clock correction error 1: Clock correction failed
3	SFO	Sync Frame Overflow Flag 0: Number of received sync frames ≤ the FLXAnFRGTUC2.SNM bit 1: More sync frames received than configured by the FLXAnFRGTUC2.SNM bit
2	SFBM	Sync Frames Below Minimum Flag 0: Sync node: 1 or more sync frames received Non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received
1	CNA	Command Not Accepted Flag 0: CHI command accepted 1: CHI command not accepted
0	PEMC	POC Error Mode Changed Flag 0: Error mode has not changed 1: Error mode has changed

(1) FLXAnFREIR.TABB

Transmission Across Boundary Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when 1 is written.

The flag signals to the Host that a transmission across a slot boundary occurred for channel B.

(2) FLXAnFREIR.LTVB

Latest Transmit Violation Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a latest transmit violation on channel B to the Host.

(3) FLXAnFREIR.EDB

Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This bit is set whenever one of bits FLXAnFRACS.SEDB, FLXAnFRACS.CEDB, FLXAnFRACS.CIB, and FLXAnFRACS.SBVB changes from 0 to 1.

(4) FLXAnFREIR.TABA

Transmission Across Boundary Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals to the Host that a transmission across a slot boundary occurred for channel A.

(5) FLXAnFREIR.LTVA

Latest Transmit Violation Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a latest transmit violation on channel A to the Host.

(6) FLXAnFREIR.EDA

Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This bit is set whenever one of bits FLXAnFRACS.SEDA, FLXAnFRACS.CEDA, FLXAnFRACS.CIA, and FLXAnFRACS.SBVA changes from 0 to 1.

(7) FLXAnFREIR.MHF

Message Handler Constraints Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals a Message Handler constraints violation condition. It is set whenever one of the bits FLXAnFRMHDF.SNUA, FLXAnFRMHDF.SNUB, FLXAnFRMHDF.FNFA, FLXAnFRMHDF.FNFB, FLXAnFRMHDF.TBFA, FLXAnFRMHDF.TBFB, and FLXAnFRMHDF.WAHP changes from 0 to 1.

(8) FLXAnFREIR.IOBA

Illegal Output buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while bit FLXAnFROBCR.OBSYS is set to 1.

(9) FLXAnFREIR.IIBA

Illegal Input Buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies:

1. The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the
   - Header section of message buffer 0, 1 if configured for transmission in key slot
   - Header section of static message buffers with buffer number < FLXAnFRMRC.FDB[7:0] while bits FLXAnFRMRC.SEC[1:0] = “01”
   - Header section of any static or dynamic message buffer while bits FLXAnFRMRC.SEC[1:0] = “1x”
   - Header and / or data section of any message buffer belonging to the receive FIFO
2. The Host writes to any register of the Input Buffer while bit FLXAnFRIBCR.IBSYH is set to 1.

(10) FLXAnFREIR.EFA

Empty FIFO Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty.

(11) FLXAnFREIR.RFO

Receive FIFO Overrun Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. The actual state of the FIFO is monitored in register FLXAnFRFSR.

(12) FLXAnFREIR.AERR

Access error flag Flag

Writing 0 in this bit has no effect.

This bit is cleared when writing 1 to it.

Notifies of an access error.

When bit FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, or FLXAnFRMHDS.ATBF2 changes from 0 to 1, this bit is set to 1.

(13) FLXAnFREIR.CCL

CHI Command Locked Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits was not successful because the execution of the previous CHI command has not yet completed. In this case bit FLXAnFREIR.CNA is also set to 1.

(14) FLXAnFREIR.CCF

Clock Correction Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set at the end of the communication cycle whenever one of the following errors occurred:

- Missing offset and / or rate correction
- Clock correction limit reached

The clock correction status is monitored in registers FLXAnFRCCEV and FLXAnFRSFS. A failure may occur during startup, therefore bit FLXAnFREIR.CCF should be set to 0 after the CC entered NORMAL_ACTIVE state.

(15) FLXAnFREIR.SFO

Sync Frame Overflow Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

It is set to 1 when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last double cycle exceeds the maximum number of sync frames as defined by the FLXAnFRGTUC2.SNM[3:0] bits.

(16) FLXAnFREIR.SFBM

Sync Frames Below Minimum Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 at the end of a cycle if the number of sync frames received during the last communication cycle was below the limit required for rate or offset correction term calculation (i.e. missing offset and / or missing rate correction). The clock correction status is monitored in FLXAnFRCCEV and FLXAnFRSFS.

This flag may be set to 1 during startup. Therefore this flag should be set to 0 by the Host after the CC entered NORMAL_ACTIVE state.

(17) FLXAnFREIR.CNA

Command Not Accepted Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

The flag signals that the write access to the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (FLXAnFREIR.CCL = 1).

(18) FLXAnFREIR.PEMC

POC Error Mode Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 whenever the error mode signaled by the FLXAnFRCCEV.ERRM[1:0] bits has changed.

24.3.4.2 FLXAnFRSIR — FlexRay Status Interrupt Register

Do not rewrite this register using bit manipulation instructions.

The flags are set when the CC detects one of the listed events. The flags remain set until the Host clears them.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSB	WUPB	—	—	—	—	—	—	MTSA	WUPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.14 FLXAnFRSIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSB	MTS Received on Channel B Flag (vSS!ValidMTSB) 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
24	WUPB	Wakeup Pattern Channel B Flag 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSA	MTS Received on Channel A Flag (vSS!ValidMTSA) 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
16	WUPA	Wakeup Pattern Channel A Flag 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A
15	SDS	Start of Dynamic Segment Flag 0: Dynamic segment not yet started 1: Dynamic segment started
14	MBSI	Message Buffer Status Interrupt Flag 0: No message buffer status change of message buffer with MBI = 1 1: Message buffer status of at least one message buffer with MBI = 1 has changed
13	SUCS	Startup Completed Successfully Flag 0: No startup completed successfully 1: Startup completed successfully
12	SWE	Stop Watch Event Flag 0: No Stop Watch Event 1: Stop Watch Event occurred
11	TOBC	Transfer Output Buffer Completed Flag 0: No transfer completed 1: Transfer between Message RAM and Output Buffer completed

Table 24.14 FLXAnFRSIR Register Contents (2/2)

Bit Position	Bit Name	Function
10	TIBC	Transfer Input Buffer Completed Flag 0: No transfer completed 1: Transfer between Input Buffer and Message RAM completed
9	TI1	Timer Interrupt 1 Flag 0: No timer interrupt 1 1: Timer interrupt 1 occurred
8	TI0	Timer Interrupt 0 Flag 0: No timer interrupt 0 1: Timer interrupt 0 occurred
7	NMVC	Network Management Vector Changed Flag 0: No change in the network management vector 1: Network management vector changed
6	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
5	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty
4	RXI	Receive Interrupt Flag 0: No ND flag of a receive buffer with MBI = 1 has been set to 1 1: At least one ND flag of a receive buffer with MBI = 1 has been set to 1
3	TXI	Transmit Interrupt Flag 0: No frame transmitted from a transmit buffer with MBI = 1 1: At least one frame was transmitted from a transmit buffer with MBI = 1
2	CYCS	Cycle Start Interrupt Flag 0: No communication cycle started 1: Communication cycle started
1	CAS	Collision Avoidance Symbol Flag 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received
0	WST	Wakeup Status Flag 0: Wakeup status unchanged 1: Wakeup status changed

(1) FLXAnFRSIR.MTSB

MTS Received on Channel B Flag (vSS!ValidMTSB)

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Media Access Test symbol received on channel B during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(2) FLXAnFRSIR.WUPB

Wakeup Pattern Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel B in either of the following states:

- WAKEUP
- READY
- STARTUP

(3) FLXAnFRSIR.MTSA

MTS Received on Channel A Flag (vSS!ValidMTSA)

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Media Access Test symbol received on channel A during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(4) FLXAnFRSIR.WUPA

Wakeup Pattern Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel A in either of the following states:

- WAKEUP
- READY
- STARTUP

(5) FLXAnFRSIR.SDS

Start of Dynamic Segment Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the dynamic segment starts.

(6) FLXAnFRSIR.MBSI

Message Buffer Status Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the message buffer status FLXAnFRMBS register has changed and if bit MBI of that message buffer is 1 (see **Table 24.106**).

(7) FLXAnFRSIR.SUCS

Startup Completed Successfully Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

(8) FLXAnFRSIR.SWE

Stop Watch Event Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set after a stop watch activation when the actual cycle counter and macrotick value are stored in the Stop Watch register (see **Section 24.3.5.4, FLXAnFRSTPW1 — FlexRay Stop Watch Register 1**).

(9) FLXAnFRSIR.TOBC

Transfer Output Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and bit FLXAnFROBCR.OBSYS has been reset by the Message Handler.

(10) FLXAnFRSIR.TIBC

Transfer Input Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever a transfer from Input Buffer to the Message RAM has completed and bit FLXAnFRIBCR.IBSYS has been reset by the Message Handler.

(11) FLXAnFRSIR.TI1

Timer 1 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever timer 1 matches the conditions configured in register FLXAnFRT1C.

FlexRay timer 1 interrupt is generated when bit FLXAnFROC.T1IE is effective.

(12) FLXAnFRSIR.TI0

Timer 0 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set whenever timer 0 matches the conditions configured in register FLXAnFRT0C.

FlexRay timer 0 interrupt is generated when bit FLXAnFROC.T0IE is effective.

(13) FLXAnFRSIR.NMVC

Network Management Vector Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This is set when a change in the Network Management Vector occurs.

(14) FLXAnFRSIR.RFCL

Receive FIFO Critical Level Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set when the receive FIFO fill level indicated by bits FLXAnFRFSR.RFFL[7:0] is equal or greater than the critical level as configured by bit FLXAnFRFCL.CL.

(15) FLXAnFRSIR.RFNE

Receive FIFO Not Empty Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The actual state of the receive FIFO is monitored in register FLXAnFRFSR.

(16) FLXAnFRSIR.RXI

Receive Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see **Section 24.3.9.6, FLXAnFRNDATm — FlexRay New Data Register m (m = 1 to 4)**, and if bit MBI of that message buffer is set to 1 (see **Table 24.106**)

(17) FLXAnFRSIR.TXI

Transmit Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC at the end of frame transmission if bit MBI in the respective message buffer is set to 1 (see **Table 24.106**).

(18) FLXAnFRSIR.CYCS

Cycle Start Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when a communication cycle starts.

(19) FLXAnFRSIR.CAS

Collision Avoidance Symbol Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received.

(20) FLXAnFRSIR.WST

Wakeup Status Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set when flags FLXAnFRCCSV.WSV[2:0] change to a value other than UNDEFINED.

24.3.4.3 FLXAnFREILS — FlexRay Error Interrupt Line Select Register

This register assigns an interrupt generated by a specific error interrupt flag from register FLXAnFREIR to one of the two modules interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0028_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBL	LTVBL	EDBL	—	—	—	—	—	TABAL	LTVAL	EDAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFL	IOBAL	IIBAL	EFAL	RFOL	AERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.15 FLXAnFREILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBL	Transmission Across Boundary Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
25	LTVBL	Latest Transmit Violation Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	EDBL	Error Detected on Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAL	Transmission Across Boundary Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
17	LTVAL	Latest Transmit Violation Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	EDAL	Error Detected on Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFL	Message Handler Constraints Flag Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
10	IOBAL	Illegal Output Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	IIBAL	Illegal Input Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 24.15 FLXAnFREILS Register Contents (2/2)

Bit Position	Bit Name	Function
8	EFAL	Empty FIFO Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	RFOL	Receive FIFO Overrun Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	AERRL	Access Error Interrupt Output Select Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	CCLL	CHI Command Locked Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	CCFL	Clock Correction Failure Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	SFOL	Sync Frame Overflow Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	SFBML	Sync Frames Below Minimum Interrupt Line Bit 0: Interrupt assigned to FlexRay 0interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CNAL	Command Not Accepted Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	PEMCL	POC Error Mode Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

24.3.4.4 FLXAnFRSILS — FlexRay Status Interrupt Line Select Register

This register assigns an interrupt generated by a specific status interrupt flag from register FLXAnFRSIR to one of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $002C_H$

Value after reset: $0303\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBL	WUPBL	—	—	—	—	—	—	MTSAL	WUPAL
Value after reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.16 FLXAnFRSILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBL	Media Access Test Symbol Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	WUPBL	Wakeup Pattern Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAL	Media Access Test Symbol Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	WUPAL	Wakeup Pattern Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15	SDSL	Start of Dynamic Segment Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
14	MBSIL	Message Buffer Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
13	SUCSL	Startup Completed Successfully Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
12	SWEL	Stop Watch Event Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
11	TOBCL	Transfer Output Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 24.16 FLXAnFRSILS Register Contents (2/2)

Bit Position	Bit Name	Function
10	TIBCL	Transfer Input Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	TI1L	Timer 1 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
8	TI0L	Timer 0 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	NMVCL	Network Management Vector Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	RFCLL	Receive FIFO Critical Level Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	RFNEL	Receive FIFO Not Empty Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	RXIL	Receive Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	TXIL	Transmit Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	CYCSL	Cycle Start Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CASL	Collision Avoidance Symbol Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	WSTL	Wakeup Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

24.3.4.5 FLXAnFREIES — FlexRay Error Interrupt Enable Set Register

The settings in the FlexRay Error Interrupt Enable Set (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBE	LTVBE	EDBE	—	—	—	—	—	TABAE	LTVAE	EDAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFE	IOBAE	IIBAE	EFAE	RFOE	AERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.17 FLXAnFREIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBE	Transmission Across Boundary Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBE	Latest Transmit Violation Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBE	Error Detected on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAE	Transmission Across Boundary Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAE	Latest Transmit Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
16	EDAE	Error Detected on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 24.17 FLXAnFREIES Register Contents (2/2)

Bit Position	Bit Name	Function
11	MHFE	Message Handler Constraints Flag Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAE	Illegal Output Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAE	Illegal Input Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAE	Empty FIFO Access Interrupt Enable3 Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOE	Receive FIFO Overrun Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRE	Access Error Interrupt Enable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLE	CHI Command Locked Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFE	Clock Correction Failure Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOE	Sync Frame Overflow Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBME	Sync Frames Below Minimum Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAE	Command Not Accepted Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCE	POC Error Mode Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

24.3.4.6 FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register

The settings in the FlexRay Error Interrupt Enable Set (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBD	LTVBD	EDBD	—	—	—	—	—	TABAD	LTVAD	EDAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFD	IOBAD	IIBAD	EFAD	RFOD	AERRD	CCLD	CCFD	SFOD	SFBMD	CNAD	PEMCD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.18 FLXAnFREIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBD	Transmission Across Boundary Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBD	Latest Transmit Violation Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBD	Error Detected on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAD	Transmission Across Boundary Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAD	Latest Transmit Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	EDAD	Error Detected on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 24.18 FLXAnFREIER Register Contents (2/2)

Bit Position	Bit Name	Function
11	MHFD	Message Handler Constraints Flag Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAD	Illegal Output Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAD	Illegal Input Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAD	Empty FIFO Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOD	Receive FIFO Overrun Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRD	Access Error Interrupt Disable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLD	CHI Command Locked Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFD	Clock Correction Failure Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOD	Sync Frame Overflow Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBMD	Sync Frames Below Minimum Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAD	Command Not Accepted Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCD	POC Error Mode Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

24.3.4.7 FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register

The settings in the FlexRay Status Interrupt Enable Set (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBE	WUPBE	—	—	—	—	—	—	MTSAE	WUPAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.19 FLXAnFRSIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBE	MTS Received on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBE	Wakeup Pattern Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAE	MTS Received on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAE	Wakeup Pattern Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSE	Start of Dynamic Segment Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
14	MBSIE	Message Buffer Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSE	Startup Completed Successfully Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 24.19 FLXAnFRSIES Register Contents (2/2)

Bit Position	Bit Name	Function
12	SWEE	Stop Watch Event Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCE	Transfer Output Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCE	Transfer Input Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1E	Timer 1 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0E	Timer 0 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCE	Network Management Vector Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLE	Receive FIFO Critical Level Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNEE	Receive FIFO Not Empty Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXIE	Receive Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXIE	Transmit Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSE	Cycle Start Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASE	Collision Avoidance Symbol Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTE	Wakeup Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

24.3.4.8 FLXAnFRSIER — FlexRay Status Interrupt Enable Reset Register

The settings in the FlexRay Status Interrupt Enable Set (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $003C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBD	WUPBD	—	—	—	—	—	—	MTSAD	WUPAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSD	MBSID	SUCSD	SWED	TOBCD	TIBCD	TI1D	TI0D	NMVCD	RFCLD	RFNED	RXID	TXID	CYCSD	CASD	WSTD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.20 FLXAnFRSIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBD	MTS Received on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBD	Wakeup Pattern Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAD	MTS Received on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAD	Wakeup Pattern Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSD	Start of Dynamic Segment Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
14	MBSID	Message Buffer Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSD	Startup Completed Successfully Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 24.20 FLXAnFRSIER Register Contents (2/2)

Bit Position	Bit Name	Function
12	SWED	Stop Watch Event Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCD	Transfer Output Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCD	Transfer Input Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1D	Timer Interrupt 1 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0D	Timer Interrupt 0 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCD	Network Management Vector Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLD	Receive FIFO Critical Level Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNED	Receive FIFO Not Empty Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXID	Receive Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXID	Transmit Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSD	Cycle Start Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASD	Collision Avoidance Symbol Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTD	Wakeup Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

24.3.4.9 FLXAnFRILE — FlexRay Interrupt Line Enable Register

Each of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt) can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EINT1	EINT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.21 FLXAnFRILE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	EINT1	Enable FlexRay 1 Interrupt Line Bit 0: FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt enabled
0	EINT0	Enable FlexRay 0 Interrupt Line Bit 0: FlexRay 0 interrupt disabled 1: FlexRay 0 interrupt enabled

24.3.5 FlexRay Timer Registers

24.3.5.1 FLXAnFRT0C — FlexRay Timer 0 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 0 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 0 passes, bits FLXAnFRSIR.TI0 and FLXAnFROS.T0IS are set to 1. A timer 0 interrupt then occurs while bit FLXAnFROC.T0IE is effective.

CAUTION

The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T0CC[6:0]							—	—	—	—	—	—	T0MS	T0RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.22 FLXAnFRT0C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T0MO[13:0]	Timer 0 Macrotick Offset Bit Timer 0 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T0CC[6:0]	Timer 0 Cycle Code Bit Timer 0 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T0MS	Timer 0 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T0RC	Timer 0 Run Control Bit 0: Timer 0 halted 1: Timer 0 running

(1) FLXAnFRT0C.T0MO

Timer 0 Macrotick Offset Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.T0RC to 0.

Configures the macrotick offset from the beginning of the communication cycle where the interrupt is to occur. The FlexRay timer 0 interrupt occurs at this offset for each cycle of the cycle set.

(2) FLXAnFRT0C.T0CC

Timer 0 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.T0RC to 0.

The 7-bit timer 0 cycle code determines the cycle set used for generation of the FlexRay timer 0 interrupt. For details about the configuration of the cycle code see **Section 24.4.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT0C.T0MS

Timer 0 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT0C.T0RC to 0.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT0C.T0RC

Timer 0 Run Control Bit

Timer 0 can be activated (set bit FLXAnFRT0C.T0RC to 1) when the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

24.3.5.2 FLXAnFRT1C — FlexRay Timer 1 Configuration Register

This register is a relative timer. After the specified number of macroticks (MT) has expired, a FlexRay timer 1 interrupt is asserted. When the FlexRay timer 1 passes, bits FLXAnFRSIR.TI1 and FLXAnFROS.T1IS are set to 1. A timer 1 interrupt then occurs while bit FLXAnFROC.T1IE is effective.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0048_H

Value after reset: $0002\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T1MC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	T1MS	T1RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.23 FLXAnFRT1C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T1MC[13:0]	Specify timer 1 macrotick count value.
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T1MS	Timer 1 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T1RC	Timer 1 Run Control Bit 0: Timer 1 halted 1: Timer 1 running

(1) FLXAnFRT1C.T1MC

Timer 1 Macrotick Count Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT1C.T1RC to 0.

Valid values are 2 to 16383 MT in continuous mode

Valid values are 1 to 16383 MT in single-shot mode

When the configured macrotick count is reached the FlexRay timer 1 interrupt is generated.

(2) FLXAnFRT1C.T1MS

Timer 1 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing bit FLXAnFRT1C.T1RC to 0.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(3) FLXAnFRT1C.T1RC

Timer 1 Run Control Bit

Timer 1 can be activated (set bit FLXAnFRT1C.T1RC to 1) as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

24.3.5.3 FLXAnFRT2C — FlexRay Timer 2 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 2 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 2 expires, FLXAnFROS.T2IS are set to '1'. A timer 2 interrupt then occurs while the FLXAnFROC.T2IE bit is effective.

FlexRay timer 2 has the same absolute timer features as FlexRay timer 0.

CAUTION

The configuration of timer 2 is compared against the macrotick counter value, there is no separate counter for timer 2.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0844_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T2MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T2CC[6:0]							—	—	—	—	—	—	T2MS	T2RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.24 FLXAnFRT2C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T2MO[13:0]	Timer 2 Macrotick Offset Bit Timer 2 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T2CC[6:0]	Timer 2 Cycle Code Bit Timer 2 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T2MS	Timer 2 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T2RC	Timer 2 Run Control Bit 0: Timer halted 1: Timer running

(1) FLXAnFRT2C.T2MO

Timer 2 Macrotick Offset Bit

Stop the timer by writing 0 to the FLXAnFRT2C.T2RC bit before changing the setting of the timer.

The T2MO bits are used to set the timing for generation of the timer 2 interrupt as an offset value in MT units from the position where the transfer cycle starts. That is, the timer 2 interrupt is generated at the offset position specified by the setting for the number of MT cycles.

(2) FLXAnFRT2C.T2CC

Timer 2 Cycle Code Bit

Stop the timer by writing 0 to bit FLXAnFRT2C.T2RC before changing the setting of the timer.

The T2CC bits are used to make the cycle setting for generation of the timer 2 interrupt as a 7-bit timer 2 cycle code. For details, see **Section 24.4.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT2C.T2MS

Timer 2 Mode Select Bit

Stop the timer by writing 0 to bit FLXAnFRT2C.T2RC before changing the setting of the timer.

Set the execution mode of the timer. In single shot mode, the timer stops when the timer setting matched with the MT value of the cycle counter.

(4) FLXAnFRT2C.T2RC

Timer 2 Run Control Bit

The timer 2 can operate only when POC is in NORMAL_ACTIVE state or NORMAL_PASSIVE state (bit FLXAnFRT2C.T2RC is set to 1).

The timer 2 stops when it transits to other state except transition between NORMAL_ACTIVE state and NORMAL_PASSIVE state.

24.3.5.4 FLXAnFRSTPW1 — FlexRay Stop Watch Register 1

The stop watch is activated by the following trigger events.

- Input of a rising edge or falling edge to the FLXAnSTPWT pin
- FlexRay 0 interrupt or FlexRay 1 interrupt
- Writing bit FLXAnFRSTPW1.SSWT to 1

With the macrotick counter increment following next to the stop watch activation the actual cycle counter and macrotick values are captured in register FLXAnFRSTPW1 while the slot counter values for channel A and B are captured in register FLXAnFRSTPW2.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $004C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SMTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SCCV[5:0]						—	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.25 FLXAnFRSTPW1 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	SMTV[13:0]	Stop Watch Event Occurrence Macrotick Value
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	SCCV[5:0]	Stop Watch Event Occurrence Cycle Counter Value
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	EINT1	FlexRay Interrupt 1 Trigger Enable 0: Stop watch trigger by FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt event triggers stop watch
5	EINT0	FlexRay 0 Interrupt Trigger Enable 0: Stop watch trigger by FlexRay 0 interrupt disabled 1: FlexRay interrupt 0 event triggers stop watch
4	EETP	External Trigger Pin Enable 0: Trigger by the FLXAnSTPWT pin is disabled. 1: Trigger by the FLXAnSTPWT pin is enabled.
3	SSWT	Software Stop Watch Trigger 0: Software trigger reset 1: Stop watch activated by software trigger
2	EDGE	Stop Watch Trigger Edge Select 0: Falling edge 1: Rising edge

Table 24.25 FLXAnFRSTPW1 Register Contents (2/2)

Bit Position	Bit Name	Function
1	SWMS	Stop Watch Mode Select 0: Single-shot mode 1: Continuous mode
0	ESWT	Hardware Stop Watch Trigger Enable 0: Stop watch trigger disabled 1: Stop watch trigger enabled

(1) FLXAnFRSTPW1.SMTV

Stop Watch Captured Macrotick Value

State of the macrotick counter when the stop watch event occurred.

(2) FLXAnFRSTPW1.SCCV

Stop Watch Captured Cycle Counter Value

State of the cycle counter when the stop watch event occurred.

(3) FLXAnFRSTPW1.EINT1

Enable FlexRay 1 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 1 interrupt when bit FLXAnFRSTPW1.ESWT is 1.

(4) FLXAnFRSTPW1.EINT0

Enable FlexRay 0 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 0 interrupt when bit FLXAnFRSTPW1.ESWT = 1.

(5) FLXAnFRSTPW1.EETP

External Trigger Pin Enable Bit

When bit FLXAnFRSTPW1.ESWT is set to 1, the FLXAnSTPWT pin event is treated as a stop watch trigger.

(6) FLXAnFRSTPW1.SSWT

Software Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

Writing 1 in this bit activates the stop watch. This bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

(7) FLXAnFRSTPW1.EDGE

Stop Watch Trigger Edge Select Bit

(8) FLXAnFRSTPW1.SWMS

Stop Watch Mode Select Bit

(9) FLXAnFRSTPW1.ESWT

Enable Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

If enabled, any of an external trigger pin input event, a FlexRay 0 interrupt event, or a FlexRay 1 interrupt event activates the stop watch.

In single-shot mode, this bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

24.3.5.5 FLXAnFRSTPW2 — FlexRay Stop Watch Register 2

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SSCVB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SSCVA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.26 FLXAnFRSTPW2 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SSCVB[10:0]	Stop Watch Captured Slot Counter Value Channel B
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SSCVA[10:0]	Stop Watch Captured Slot Counter Value Channel A

(1) FLXAnFRSTPW2.SSCVB

Stop Watch Captured Slot Counter Value Channel B

State of the slot counter for channel B when the stop watch event occurred.

(2) FLXAnFRSTPW2.SSCVA

Stop Watch Captured Slot Counter Value Channel A

State of the slot counter for channel A when the stop watch event occurred.

24.3.6 CC Control Registers

This section describes the registers provided by the CC (Communication Controller) to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered from reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to execute the lock release sequence as described in **Section 24.3.3.1, FLXAnFRLCK — FlexRay Lock Register**.

24.3.6.1 FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0080_H

Value after reset: 0C40 1080_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CCHB	CCHA	MTSB	MTSA	HCSE	TSM	WUCS	PTA[4:0]				
Value after reset	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[4:0]					—	TXSY	TXST	PBSY	—	—	—	CMD[3:0]			
Value after reset	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.27 FLXAnFRSUCC1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	CCHB	Connected to Channel B Bit Configures pChannels 0: Not connected to channel B 1: Node connected to channel B (value after reset)
26	CCHA	Connected to Channel A Bit Configures pChannels 0: Not connected to channel A 1: Node connected to channel A (value after reset)
25	MTSB	Select Channel B for MTS Transmission Bit 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission
24	MTSA	Select Channel A for MTS Transmission Bit 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission
23	HCSE	Halt due to Clock Sync Error Bit Configures pAllowHaltDueToClock 0: CC will enter / remain in NORMAL_PASSIVE 1: CC will enter HALT state
22	TSM	Transmission Slot Mode Bit Configures pSingleSlotEnabled 0: ALL Slot Mode 1: SINGLE Slot Mode (value after reset)

Table 24.27 FLXAnFRSUCC1 Register Contents (2/2)

Bit Position	Bit Name	Function
21	WUCS	Wakeup Channel Select Bit Configures pWakeupChannel 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B
20 to 16	PTA[4:0]	Passive to Active Bit Configures pAllowPassiveToActive
15 to 11	CSA[4:0]	Cold Start Attempts Bit Configures gColdStartAttempts
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TXSY	Transmit Sync Frame in Key Slot Bit Configures pKeySlotUsedForSync 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node
8	TXST	Transmit Startup Frame in Key Slot Bit Configures pKeySlotUsedForStartup 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter
7	PBSY	POC Busy Flag 0: POC not busy, bit FLXAnFRSUCC1.CMD writeable 1: POC is busy, bit FLXAnFRSUCC1.CMD locked
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	CMD[3:0]	CHI Command Vector Bit 0000: command_not_accepted 0001: CONFIG 0010: READY 0011: WAKEUP 0100: RUN 0101: ALL_SLOTS 0110: HALT 0111: FREEZE 1000: SEND_MTS 1001: ALLOW_COLDSTART 1010: RESET_STATUS_INDICATORS 1100: CLEAR_RAMS others: reserved

(1) FLXAnFRSUCC1.CCHB

Connected to Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel B (pChannels).

(2) FLXAnFRSUCC1.CCHA

Connected to Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel A (pChannels).

(3) FLXAnFRSUCC1.MTSB

Select Channel B for MTS Transmission Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The FLXAnFRSUCC1.MTSB bit may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 24.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing 1000_B to the FLXAnFRSUCC1.CMD[3:0] bits.

The bit selects channel B for MTS symbol transmission.

(4) FLXAnFRSUCC1.MTSA

Select Channel A for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS[5:0] bits are DEFAULT_CONFIG or CONFIG.

Bit FLXAnFRSUCC1.MTSA may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 24.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1 an MTS symbol will be transmitted on both channels when requested by writing 1000_B to the FLXAnFRSUCC1.CMD[3:0] bits.

The bit selects channel A for MTS symbol transmission.

(5) FLXAnFRSUCC1.HCSE

Halt due to Clock Sync Error Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Controls the transition to HALT state due to a clock synchronization error (pAllowHaltDueToClock).

(6) FLXAnFRSUCC1.TSM

Transmission Slot Mode Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] is DEFAULT_CONFIG or CONFIG.

Selects the value after transmission slot mode reset (pSingleSlotEnabled).

In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 respectively message buffers 0 and 1 depending on bit FLXAnFRMRC.SPLM.

In case FLXAnFRSUCC1.TSM = 1, message buffer 0 respectively message buffers 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots.

FLXAnFRSUCC1.TSM is a configuration bit which can only be set / reset by the Host.

The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing 0101_B to bits FLXAnFRSUCC1.CMD[3:0] in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by bits FLXAnFRCCSV.SLM[1:0].

(7) FLXAnFRSUCC1.WUCS

Wakeup Channel Select Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

With this bit the Host selects the channel on which the CC sends the Wakeup pattern (pWakeupChannel).

(8) FLXAnFRSUCC1.PTA

Passive to Active Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 31 even / odd cycle pairs.

Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state (pAllowPassiveToActive).

If set to "00000_B" the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(9) FLXAnFRSUCC1.CSA

Cold Start Attempts Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Must be identical in all nodes of a cluster.

Valid values are 2 to 31.

Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node (gColdStartAttempts).

(10) FLXAnFRSUCC1.TXSY

Transmit Sync Frame in Key Slot Bit

The user can only write to these bits when the FLXAnFRCCSV.POCS[5:0] bits is DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit sync frames (pKeySlotUsedForSync).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(11) FLXAnFRSUCC1.TXST

Transmit Startup Frame in Key Slot Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit startup frames (pKeySlotUsedForStartup).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(12) FLXAnFRSUCC1.PBSY

POC Busy Flag

Signals that the POC is busy and cannot accept a command from the Host. Bits FLXAnFRSUCC1.CMD[5:0] are locked against write accesses.

Set to 1 after reset during initialization of internal RAM blocks.

(13) FLXAnFRSUCC1.CMD

CHI Command Vector Bit

The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXAnFRSUCC1.CMD[3:0] bits will be reset to 0000_B = command_not_accepted, and flag FLXAnFREIR.CNA will be set to 1.

In general the Host must check FLXAnFRSUCC1.PBSY before writing a new CHI command.

In case the previous CHI command has not yet completed, flag FLXAnFREIR.CCL is set to 1 together with FLXAnFREIR.CNA; the CHI command needs to be repeated.

Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will flag FLXAnFREIR.CNA be set.

Reading bits FLXAnFRSUCC1.CMD[3:0] show whether the last CHI command was accepted. The actual POC state is monitored by bits FLXAnFRCCSV.POCS[5:0].

- command_not_accepted

  Bits FLXAnFRSUCC1.CMD[3:0] are reset to 0000_B due to one of the following conditions:

  – Illegal command applied by the Host
  – Host applied command to leave CONFIG state without preceding config lock key
  – Host applied new command while execution of the previous Host command has not completed
  – Host writes command_not_accepted

When bits FLXAnFRSUCC1.CMD[3:0] are reset to 0000_B, FLXAnFREIR.CNA is set to 1, and if enabled an interrupt is generated. Commands which are not accepted are not executed.

CONFIG command

Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, or READY. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to 0000_B = command_not_accepted.

READY command

Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

WAKEUP command

Go to POC state WAKEUP when called in POC state READY. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

RUN command

Go to POC state STARTUP when called in POC state READY. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

ALL_SLOTS command

Leave SINGLE slot mode and go to ALL-SLOTS mode after successful startup / integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

HALT command

Set halt request FLXAnFRCCSV.HRQ to 1 and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

FREEZE command

Set the freeze status indicator FLXAnFRCCSV.FSI flag to 1 and go to POC state HALT immediately. Can be called from any state.

SEND_MTS command

Send single MTS symbol during the next following symbol window on the channel configured by the FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB bits, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (the FLXAnFRCCSV.SLM[1:0] bits = "11"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

ALLOW_COLDSTART command

The command resets the FLXAnFRCCSV.CSI flag to enable the node to become leading coldstarter. When called in states DEFAULT_CONFIG, CONFIG, or HALT, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted. To become leading coldstarter it is also required that both FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY bits are set.

RESET_STATUS_INDICATORS command

Resets status flags FLXAnFRCCSV.CSNI, FLXAnFRCCSV.CSAI and FLXAnFRCCSV.WSV to their values after reset. May be called in POC states READY and STARTUP. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

CLEAR_RAMS command

Sets bit FLXAnFRMHDS.CRAM to 1 when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to "0000_B" = command_not_accepted.

Bit FLXAnFRMHDS.CRAM is also set to 1 when the CC leaves reset. By setting bit FLXAnFRMHDS.CRAM all internal RAM blocks are initialized to zero. During the initialization of the RAMs, bit RSUCC1.PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS.

The initialization of the internal message RAM requires 2048 bus clock cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after reset or after assertion of CHI command CLEAR_RAMS.

Before asserting CHI command CLEAR_RAMS, the Host should make sure that no transfer between Message RAM and IBF / OBF or the TBFRAMs is ongoing and that the data transfer handler has no effect (bits FLXAnFRITS.ITS and FLXAnFROTS.OTS are 0). This command also resets the Message Buffer Status registers FLXAnFRMHDS, FLXAnFRLDTS, FLXAnFRFSR, FLXAnFRMHDF, FLXAnFRTXRQ1 to FLXAnFRTXRQ4, FLXAnFRNDAT1 to FLXAnFRNDAT14, and FLXAnFRMBSC1 to FLXAnFRMBSC4.

CAUTIONS

1. **All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the FlexRay domain after at most 8 cycles of the slower of the two clocks "bus clock" and "FlexRay sample clock", assumed that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXAnFRCCSV will show data that is additionally delayed by synchronization from the FlexRay domain to the bus clock domain. The maximum additional delay is 12 cycles of the slower of the two clocks 'bus clock' and "FlexRay sample clock".**
2. **When transfer is stopped by a FREEZE or READY command and then restarted as a leading ColdStart node, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. Such cases arise when the startup frame is set in a slot 1 to slot 7. This does not occur in a ColdStart after the reset of the microcontroller. Even if the above situation arises, the ColdStart will succeed on the second trial. Repetition prolongs the overall ColdStart process, but the ColdStart will not be obstructed by the earlier situation. To avoid this effect, allocate startup and sync frames to static slot 8 or a slot with a higher number.**

Table 24.28 below references the CHI commands from the *FlexRay Protocol Specification* (*Section 2.1.1.1, Table 2.2*) to the FlexRay CHI command vector FLXAnFRSUCC1.CMD[3:0] bits.

Table 24.28 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI command	Where processed (POC States)	CHI Command Vector CMD
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active, POC: normal passive	HALT
READY	All except POC: default config, POC: config, POC: ready, POC: halt	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

24.3.6.2 FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0084_H

Value after reset: $0100\ 0504_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LTN[3:0]				—	—	—	LT[20:16]				
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LT[15:0]															
Value after reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.29 FLXAnFRSUCC2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	LTN[3:0]	Listen Timeout Noise Bit Configures (gListenNoise - 1)
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	LT[20:0]	Listen Timeout Bit Configures pdListenTimeout

(1) FLXAnFRSUCC2.LTN

Listen Timeout Noise Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The range for gListenNoise is 2 to 16.

FLXAnFRSUCC2.LTN must be configured identical in all nodes of a cluster.

Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout.

CAUTION

The wakeup / startup noise timeout is calculated as follows:
pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT × (FLXAnFRSUCC2.LTN + 1)

(2) FLXAnFRSUCC2.LT

Listen Timeout Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The range for pdListenTimeout is 1284 to 1283846 µT.

Configures wakeup / startup listen time out in µT.

24.3.6.3 FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0088_H

Value after reset: $0000\ 0011_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WCF[3:0]				WCP[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.30 FLXAnFRSUCC3 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	WCF[3:0]	Maximum Without Clock Correction Fatal Bit (transition to HALT state) Configures gMaxWithoutClockCorrectionFatal
3 to 0	WCP[3:0]	Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state) Configures gMaxWithoutClockCorrectionPassive

(1) FLXAnFRSUCC3.WCF

Maximum Without Clock Correction Fatal Bit (transition to HALT state)

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

CAUTION

The transition to HALT state is prevented if the FLXAnFRSUCC1.HCSE bit is not set.

(2) FLXAnFRSUCC3.WCP

Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state)

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15 cycle pairs.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction terms that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state.

24.3.6.4 FLXAnFRNEMC — FlexRay NEM Configuration Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $008C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	NML[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.31 FLXAnFRNEMC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	NML[3:0]	Network Management Vector Length Bit Configures gNetworkManagementVectorLength

(1) FLXAnFRNEMC.NML

Network Management Vector Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 12 bytes.

The configured length must be identical in all nodes of a cluster.

These bits configure the length of the NM vector.

24.3.6.5 FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0090_H

Value after reset: $084C\ 0633_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RWP[5:0]						—	RXW[8:0]								
Value after reset	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[1:0]		SPP[1:0]		—	CASM[6:0]							TSST[3:0]			
Value after reset	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.32 FLXAnFRPRTC1 Register Contents

Bit Position	Bit Name	Function
31 to 26	RWP[5:0]	Repetitions of Tx Wakeup Pattern Bit Configures pWakeupPattern
25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 16	RXW[8:0]	Wakeup Symbol Receive Window Length Bit Configures gdWakeupSymbolRxWindow
15, 14	BRP[1:0]	Baud Rate Prescaler Bit Configures gdSampleClockPeriod and pSamplesPerMicrotick 00 = 10 Mbps 01 = 5 Mbps 10 = 2.5 Mbps 11 = 2.5 Mbps
13, 12	SPP[1:0]	Strobe Point Position Bit Configures Strobe point position 00 = Sample 5 01 = Sample 4 10 = Sample 6 11 = Sample 5
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 4	CASM[6:0]	Collision Avoidance Symbol Max Bit Configures gdCASRxLowMax
3 to 0	TSST[3:0]	Transmission Start Sequence Transmitter Bit Configures gdTSSTransmitter

(1) FLXAnFRPRTC1.RWP

Repetitions of Tx Wakeup Pattern Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63.

Configures the number of repetitions (sequences) of the Tx wakeup symbol.

(2) FLXAnFRPRTC1.RXW

Wakeup Symbol Receive Window Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] DEFAULT_CONFIG or CONFIG.

Valid values are 76 to 301 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the received wakeup pattern.

(3) FLXAnFRPRTC1.BRP

Baud Rate Prescaler Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock set to 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

00 = 10 MBit/s

gdSampleClockPeriod = 12.5 ns = 1 × “sample clock”

pSamplesPerMicrotick = 2 (1 µT = 25 ns)

01 = 5 MBit/s

gdSampleClockPeriod = 25 ns = 2 × “sample clock”

pSamplesPerMicrotick = 1 (1 µT = 25 ns)

10, 11 = 2.5 MBit/s

gdSampleClockPeriod = 50 ns = 4 × “sample clock”

pSamplesPerMicrotick = 1 (1 µT = 50 ns)

(4) FLXAnFRPRTC1.SPP

Strobe Point Position Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

These bits specify the sample-counter position at which the Checker Core strobes in response to received bits.

The strobed bit value is determined by sampling at the timing specified by bits FLXAnFRPRTC1.SPP[1:0].

CAUTION

The current revision 2.1 of the FlexRay protocol requires that the FLXAnFRPRTC1.SPP[1:0] bits are 00. The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

(5) FLXAnFRPRTC1.CASM

Collision Avoidance Symbol Max Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

CASM6 is fixed to 1.

Valid values are 67 to 99 bit times.

Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS).

(6) FLXAnFRPRTC1.TSST

Transmission Start Sequence Transmitter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are is DEFAULT_CONFIG or CONFIG.

Valid values are 3 to 15 bit times.

Must be identical in all nodes of a cluster.

Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 µT = 100ns @ 10Mbps).

24.3.6.6 FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0094_H

Value after reset: $0F2D\ 0A0E_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TXL[5:0]						TXI[7:0]							
Value after reset	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXL[5:0]						—	—	RXI[5:0]					
Value after reset	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.33 FLXAnFRPRTC2 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	TXL[5:0]	Wakeup Symbol Transmit Low Bit Configures gdWakeupSymbolTxLow
23 to 16	TXI[7:0]	Wakeup Symbol Transmit Idle Bit Configures gdWakeupSymbolTxIdle
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RXL[5:0]	Wakeup Symbol Receive Low Bit Configures gdWakeupSymbolRxLow
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RXI[5:0]	Wakeup Symbol Rx Idle Bit Configures gdWakeupSymbolRxIdle

(1) FLXAnFRPRTC2.TXL

Wakeup Symbol Transmit Low Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 15 to 60 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol.

(2) FLXAnFRPRTC2.TXI

Wakeup Symbol Transmit Idle Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 45 to 180 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol.

(3) FLXAnFRPRTC2.RXL

Wakeup Symbol Receive Low Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 55 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol.

(4) FLXAnFRPRTC2.RXI

Wakeup Symbol Rx Idle Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 59 bit times.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol.

24.3.6.7 FLXAnFRMHDC — FlexRay MHD Configuration Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0098_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SLT[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SFDL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.34 FLXAnFRMHDC Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	SLT[12:0]	Start of Latest Transmit Bit Configures pLatestTx
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	SFDL[6:0]	Static Frame Data Length Bit Configures gPayloadLengthStatic

(1) FLXAnFRMHDC.SLT

Start of Latest Transmit Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7981 minislots.

Configures the maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXAnFRMHDC.SLT is set to zero.

(2) FLXAnFRMHDC.SFDL

Static Frame Data Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 127.

The payload length must be identical in all nodes of a cluster.

Configures the cluster-wide payload length for all frames sent in the static segment in double bytes.

24.3.6.8 FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00A0_H$

Value after reset: $0000\ 0280_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UT[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UT[15:0]															
Value after reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.35 FLXAnFRGTUC1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	UT[19:0]	Setting of Communication Cycle in Microticks Bit Configures pMicroPerCycle

(1) FLXAnFRGTUC1.UT

Setting of Communication Cycle in Microticks Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 640 to 640000 µT.

Configures the duration of the communication cycle in microticks.

24.3.6.9 FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00A4_H$

Value after reset: $0002\ 000A_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SNM[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MPC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.36 FLXAnFRGTUC2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	SNM[3:0]	Sync Node Max Bit
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MPC[13:0]	Setting of Communication Cycle in Macrotick Bit Configures gMacroPerCycle

(1) FLXAnFRGTUC2.SNM

Sync Node Max Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 15.

Must be identical in all nodes of a cluster.

Maximum number of frames within a cluster with sync frame indicator bit SYN set to 1.

(2) FLXAnFRGTUC2.MPC

Setting of Communication Cycle in Macrotick Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS [5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 16000 MT.

The cycle length must be identical in all nodes of a cluster.

Configures the duration of one communication cycle in macroticks.

24.3.6.10 FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00A8_H$

Value after reset: $0202\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MIOB[6:0]							—	MIOA[6:0]						
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UIOB[7:0]								UIOA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.37 FLXAnFRGTUC3 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MIOB[6:0]	Macrotick Initial Offset Channel B Bit Configures pMacroInitialOffset[B]
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MIOA[6:0]	Macrotick Initial Offset Channel A Bit Configures pMacroInitialOffset[A]
15 to 8	UIOB[7:0]	Microtick Initial Offset Channel B Bit Configures pMicroInitialOffset[B]
7 to 0	UIOA[7:0]	Microtick Initial Offset Channel A Bit Configures pMicroInitialOffset[A]

(1) FLXAnFRGTUC3.MIOB

Macrotick Initial Offset Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(2) FLXAnFRGTUC3.MIOA

Macrotick Initial Offset Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(3) FLXAnFRGTUC3.UIOB

Microtick Initial Offset Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 µT.

Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [B] and therefore has to be set for each channel independently.

(4) FLXAnFRGTUC3.UIOA

Microtick Initial Offset Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS [5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 µT.

Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [A] and therefore has to be set for each channel independently.

24.3.6.11 FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4

For details about configuration of FLXAnFRGTUC4.NIT and FLXAnFRGTUC4.OCS see **Section 24.4.2.5, Configuration of NIT Start and Offset Correction Start**.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00AC_H$

Value after reset: $0008\ 0007_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OCS[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	NIT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.38 FLXAnFRGTUC4 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	OCS[13:0]	Offset Correction Start Bit Configures (gOffsetCorrectionStart - 1)
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	NIT[13:0]	Network Idle Time Start Bit Configures (gMacroPerCycle -gdNIT - 1)

(1) FLXAnFRGTUC4.OCS

Offset Correction Start Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 8 to 15998 MT.

For cluster consisting of E-Ray implementations only, it is sufficient to program FLXAnFRGTUC4.OCS = FLXAnFRGTUC4.NIT + 1.

Must be identical in all nodes of a cluster.

Determines the start of the offset correction within the NIT phase, calculated from start of cycle.

(2) FLXAnFRGTUC4.NIT

Network Idle Time Start Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 7 to 15997 MT.

Must be identical in all nodes of a cluster.

Configures the starting point of the Network Idle Time NIT at the end of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if Macrotick = gMacroPerCycle -gdNIT -1 and the increment pulse of Macrotick is set.

24.3.6.12 FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00B0_H$

Value after reset: $0E00\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEC[7:0]								—	—	—	CDD[4:0]				
Value after reset	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCB[7:0]								DCA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.39 FLXAnFRGTUC5 Register Contents

Bit Position	Bit Name	Function
31 to 24	DEC[7:0]	Decoding Correction Bit Configures pDecodingCorrection
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	CDD[4:0]	Cluster Drift Damping Bit Configures pClusterDriftDamping
15 to 8	DCB[7:0]	Delay Compensation Channel B Bit Configures pDelayCompensation[B]
7 to 0	DCA[7:0]	Delay Compensation Channel A Bit Configures pDelayCompensation[A]

(1) FLXAnFRGTUC5.DEC

Decoding Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 143 µT.

Configures the decoding correction value in microticks used to determine the primary time reference point.

(2) FLXAnFRGTUC5.CDD

Cluster Drift Damping Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 20 µT.

Configures the cluster drift damping value in microticks used in clock synchronization to minimize accumulation of rounding errors.

(3) FLXAnFRGTUC5.DCB

Delay Compensation Channel B Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 µT.

Used to compensate for reception delays on channel B. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

(4) FLXAnFRGTUC5.DCA

Delay Compensation Channel A Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 µT.

Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

24.3.6.13 FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00B4_H$

Value after reset: $0002\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MOD[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASR[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.40 FLXAnFRGTUC6 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MOD[10:0]	Maximum Oscillator Drift Bit Configures pdMaxDrift
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	ASR[10:0]	Accepted Startup Range Bit Configures pdAcceptedStartupRange

(1) FLXAnFRGTUC6.MOD

Maximum Oscillator Drift Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 µT.

Maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT.

(2) FLXAnFRGTUC6.ASR

Accepted Startup Range Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 1875 µT.

Number of microticks constituting the expanded range of measured deviation for startup frames during integration.

24.3.6.14 FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00B8_H$

Value after reset: $0002\ 0004_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NSS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SSL[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.41 FLXAnFRGTUC7 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	NSS[9:0]	Number of Static Slots Bit Configures gNumberOfStaticSlots
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	SSL[9:0]	Static Slot Length Bit Configures gdStaticSlot

(1) FLXAnFRGTUC7.NSS

Number of Static Slots Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1023.

The number of static slots must be identical in all nodes of a cluster.

Configures the number of static slots in a cycle.

(2) FLXAnFRGTUC7.SSL

Static Slot Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 4 to 659 MT.

The static slot length must be identical in all nodes of a cluster.

Configures the length of a static slot in macroticks.

24.3.6.15 FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00BC_H$

Value after reset: $0000\ 0002_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NMS[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.42 FLXAnFRGTUC8 Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	NMS[12:0]	Number of Minislots Bit Configures gNumberOfMinislots
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	MSL[5:0]	Minislot Length Bit Configures gdMinislot

(1) FLXAnFRGTUC8.NMS

Number of Minislots Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7986.

The number of minislots must be identical in all nodes of a cluster.

Configures the number of minislots within the dynamic segment of a cycle.

(2) FLXAnFRGTUC8.MSL

Minislot Length Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63 MT.

The minislot length must be identical in all nodes of a cluster.

Configures the length of a minislot in macroticks.

24.3.6.16 FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00C0_H$

Value after reset: $0000\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSI[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MAPO[4:0]					—	—	APO[5:0]					
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.43 FLXAnFRGTUC9 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DSI[1:0]	Dynamic Slot Idle Phase Bit Configures gdDynamicSlotIdlePhase
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	MAPO[4:0]	Minislot Action Point Offset Bit Configures gdMinislotActionPointOffset
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	APO[5:0]	Action Point Offset Bit Configures gdActionPointOffset

(1) FLXAnFRGTUC9.DSI

Dynamic Slot Idle Phase Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 2 Minislot.

Must be identical in all nodes of a cluster.

Configures the duration of the dynamic slot idle phase in the number of minislots. The duration has to be greater or equal than the idle detection time.

(2) FLXAnFRGTUC9.MAPO

Minislot Action Point Offset Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 31 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within the minislots of the dynamic segment.

(3) FLXAnFRGTUC9.APO

Action Point Offset Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 63 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within static slots and symbol window.

24.3.6.17 FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00C4_H$

Value after reset: $0002\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MOC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.44 FLXAnFRGTUC10 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MRC[10:0]	Maximum Rate Correction Bit Configures pRateCorrectionOut
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MOC[13:0]	Maximum Offset Correction Bit Configures pOffsetCorrectionOut

(1) FLXAnFRGTUC10.MRC

Maximum Rate Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 µT.

Holds the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks only the internal rate correction value against the maximum rate correction value (absolute value).

(2) FLXAnFRGTUC10.MOC

Maximum Offset Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 5 to 15266 µT.

Holds the maximum permitted offset correction value (absolute value) to be applied by the internal clock synchronization algorithm (absolute value). The CC checks only the internal offset correction value against the maximum offset correction value.

24.3.6.18 FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $00C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ERC[2:0]			—	—	—	—	—	EOC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ERCC[1:0]		—	—	—	—	—	—	EOCC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.45 FLXAnFRGTUC11 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 24	ERC[2:0]	External Rate Correction Bit Configures pExternRateCorrection
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	EOC[2:0]	External Offset Correction Bit Configures pExternOffsetCorrection
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	ERCC[1:0]	External Rate Correction Control Bit Configures vExternRateControl 00: External rate correction is prohibited. 01: External rate correction is prohibited. 10: Subtract 11: Add
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	EOCC[1:0]	External Offset Correction Control Bit Configures vExternOffsetControl 00: External offset correction is prohibited. 01: External offset correction is prohibited. 10: Subtract 11: Add

(1) FLXAnFRGTUC11.ERC

External Rate Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 µT.

Holds the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated rate correction value. The value is applied during NIT.

(2) FLXAnFRGTUC11.EOC

External Offset Correction Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 µT.

Holds the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated offset correction value. The value is applied during NIT.

(3) FLXAnFRGTUC11.ERCC

External Rate Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

Writing the following values enables the external rate correction.

00 = External rate correction is prohibited.

01 = External rate correction is prohibited.

10 = Subtract

External rate correction value subtracted from calculated rate correction value

11 = Add

External rate correction value added to calculated rate correction value

(4) FLXAnFRGTUC11.EOCC

External Offset Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

Writing the following values enables the external offset correction.

00 = External offset correction is prohibited.

01 = External offset correction is prohibited.

10 = Subtract

External offset correction value subtracted from calculated offset correction value

11 = Add

External offset correction value added to calculated offset correction value

24.3.7 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses).

24.3.7.1 FLXAnFRCCSV — FlexRay CC Status Vector Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0100_H

Value after reset: $0010\ 4000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PSL[5:0]						RCA[4:0]					WSV[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CSI	CSAI	CSNI	—	—	SLM[1:0]		HRQ	FSI	POCS[5:0]					
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.46 FLXAnFRCCSV Register Contents (1/2)

Bit	Symbol	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 24	PSL[5:0]	POC Status Log Flag Status of FLXAnFRCCSV.POCS immediately before entering HALT state.
23 to 19	RCA[4:0]	Remaining Coldstart Attempts Flag Indicates vRemainingColdstartAttempts
18 to 16	WSV[2:0]	Wakeup Status Flag Indicates vPOC!WakeupStatus 000: UNDEFINED 001: RECEIVED_HEADER 010: RECEIVED_WUP 011: COLLISION_HEADER 100: COLLISION_WUP 101: COLLISION_UNKNOWN 110: TRANSMITTED 111: Reserved
15	Reserved	When read, the value after reset is returned.
14	CSI	Cold Start Inhibit Flag Indicates vColdStartInhibit 0: Cold starting of node enabled 1: Cold starting of node disabled
13	CSAI	Coldstart Abort Indicator Flag
12	CSNI	Coldstart Noise Indicator Flag Indicates vPOC!ColdstartNoise
11, 10	Reserved	When read, the value after reset is returned.
9, 8	SLM[1:0]	Slot Mode Flag Indicates vPOC!SlotMode 00: SINGLE 01: reserved 10: ALL_PENDING 11: ALL

Table 24.46 FLXAnFRCCSV Register Contents (2/2)

Bit	Symbol	Function
7	HRQ	Halt Request Flag Indicates vPOC!CHIHaltRequest
6	FSI	Freeze Status Indicator Flag Indicates vPOC!Freeze
5 to 0	POCS[5:0]	Protocol Operation Control Status Flag

(1) FLXAnFRCCSV.PSL

POC Status Log Flag

Set the value of bits FLXAnFRCCSV.POCS[5:0] immediately before entering HALT state.

Set to HALT when FREEZE command is applied during HALT state and FLXAnFRCCSV.FSI is not already set i.e. the HALT state was not reached by FREEZE command.

Reset to "000000_B" when leaving HALT state.

(2) FLXAnFRCCSV.RCA

Remaining Coldstart Attempts Flag

Indicates the number of remaining coldstart attempts (vRemainingColdstartAttempts).

The value after a reset during CONFIG and DEFAULT_CONFIG state is also bits FLXAnFRSUCC1.CSA[4:0].

The RUN command resets this counter to the maximum number of coldstart attempts as configured by the FLXAnFRSUCC1.CSA [4:0] bits.

(3) FLXAnFRCCSV.WSV

Wakeup Status Flag

Indicates the status of the current wakeup attempt (vPOC!WakeupStatus).

Reset to 0 when entering Wakeup state, by CHI command RESET_STATUS_INDICATORS, or by transition from DEFAULT_CONFIG to CONFIG state

000_B = UNDEFINED

Wakeup not yet executed by the CC.

001_B = RECEIVED_HEADER

Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.

010_B = RECEIVED_WUP

Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

011_B = COLLISION_HEADER

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.

100_B = COLLISION_WUP

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.

101_B = COLLISION_UNKNOWN

Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.

110_B = TRANSMITTED

Set when the CC has successfully completed the transmission of the wakeup pattern.

111_B = reserved

(4) FLXAnFRCCSV.CSI

Cold Start Inhibit Flag

Indicates that the node is disabled from cold starting (vColdStartInhibit).

The flag is set to 1 whenever the POC enters READY state due to CHI command READY.

The flag has to be reset under control of the Host by CHI command ALLOW_COLDSTART (bits FLXAnFRSUCC1.CMD[3:0] are 1001_B).

(5) FLXAnFRCCSV.CSAI

Coldstart Abort Indicator Flag

Indicates that a coldstart attempt was aborted.

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(6) FLXAnFRCCSV.CSNI

Coldstart Noise Indicator Flag

Indicates that the cold start procedure occurred under noisy conditions (vPOC!ColdstartNoise).

Reset by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(7) FLXAnFRCCSV.SLM

Slot Mode Flag

Indicates the actual slot mode of the POC (vPOC!SlotMode) in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE.

Default value is SINGLE. Changes to ALL, depending on FLXAnFRSUCC1.TSM.

In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL.

Set FLXAnFRSUCC1.TSM to SINGLE except for NORMAL_ACTIVE or NORMAL_PASSIVE.

(8) FLXAnFRCCSV.HRQ

Halt Request Flag

Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle (vPOC!CHIHaltRequest).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(9) FLXAnFRCCSV.FSI

Freeze Status Indicator Flag

Indicates that the POC has entered the HALT state due to CHI command, FREEZE (bits FLXAnFRSUCC1.CMD[3:0] are 0111_B) or due to an error condition requiring an immediate POC halt (vPOC!Freeze).

Reset by transition from HALT to DEFAULT_CONFIG state.

(10) FLXAnFRCCSV.POCS

Protocol Operation Control Status Flag

Indicates the actual state of operation of the CC Protocol Operation Control

$00\ 0000_B$ = DEFAULT_CONFIG state

$00\ 0001_B$ = READY state

$00\ 0010_B$ = NORMAL_ACTIVE state

$00\ 0011_B$ = NORMAL_PASSIVE state

$00\ 0100_B$ = HALT state

$00\ 1111_B$ = CONFIG state

Indicates the actual state of operation of the POC in the wakeup path

$01\ 0000_B$ = WAKEUP_STANDBY state

$01\ 0001_B$ = WAKEUP_LISTEN state

$01\ 0010_B$ = WAKEUP_SEND state

$01\ 0011_B$ = WAKEUP_DETECT state

Indicates the actual state of operation of the POC in the startup path

$10\ 0000_B$ = STARTUP_PREPARE state

$10\ 0001_B$ = COLDSTART_LISTEN state

$10\ 0010_B$ = COLDSTART_COLLISION_RESOLUTION state

$10\ 0011_B$ = COLDSTART_CONSISTENCY_CHECK state

$10\ 0100_B$ = COLDSTART_GAP state

$10\ 0101_B$ = COLDSTART_JOIN State

$10\ 0110_B$ = INTEGRATION_COLDSTART_CHECK state

$10\ 0111_B$ = INTEGRATION_LISTEN state

$10\ 1000_B$ = INTEGRATION_CONSISTENCY_CHECK state

$10\ 1001_B$ = INITIALIZE_SCHEDULE state

$10\ 1010_B$ = ABORT_STARTUP state

$10\ 1011_B$ = STARTUP_SUCCESS state

others = reserved

24.3.7.2 FLXAnFRCCEV — FlexRay CC Error Vector Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0104_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PTAC[4:0]					ERRM[1:0]		—	—	CCFC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.47 FLXAnFRCCEV Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12 to 8	PTAC[4:0]	Passive to Active Count Flag Indicates vAllowPassiveToActive
7, 6	ERRM[1:0]	Error Mode Flag Indicates vPOC!ErrorMode 00: ACTIVE 01: PASSIVE 10: COMM_HALT 11: reserved
5, 4	Reserved	When read, the value after reset is returned.
3 to 0	CCFC[3:0]	Clock Correction Failed Counter Indicates vClockCorrectionFailed

(1) FLXAnFRCCEV.PTAC

Passive to Active Count Flag

Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when this bit equals to the value of bits FLXAnFRSUCC1.PTA[4:0] –1.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(2) FLXAnFRCCEV.ERRM

Error Mode Flag

Indicates the actual error mode of the POC (vPOC!ErrorMode).

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(3) FLXAnFRCCEV.CCFC

Clock Correction Failed Counter

Indicates the clock correction failed counter of the POC (vClockCorrectionFailed).

The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active.

The Clock Correction Failed Counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active.

The Clock Correction Failed Counter stops at 15.

Reset by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

24.3.7.3 FLXAnFRSCV — FlexRay Slot Counter Value Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0110_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SCCB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCCA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.48 FLXAnFRSCV Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SCCB[10:0]	Slot Counter Channel B Indicates vSlotCounter[B]
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SCCA[10:0]	Slot Counter Channel A Indicates vSlotCounter[A]

(1) FLXAnFRSCV.SCCB

Slot Counter Channel B

Current slot counter value on channel B (vSlotCounter[B]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSCV.SCCA

Slot Counter Channel A

Current slot counter value on channel A (vSlotCounter[A]). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.4 FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0114_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	CCV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.49 FLXAnFRMTCCV Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	CCV[5:0]	Cycle Counter Value Indicates vCycleCounter
15, 14	Reserved	When read, the value after reset is returned.
13 to 0	MTV[13:0]	Macrotick Value Indicates vMacrotick

(1) FLXAnFRMTCCV.CCV

Cycle Counter Value

Current cycle counter value (vCycleCounter). The value is incremented by the CC at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRMTCCV.MTV

Macrotick Value

Current macrotick value (vMacrotick). The value is incremented by the CC and reset at the start of a communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.5 FLXAnFRRCV — FlexRay Rate Correction Value Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0118_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RCV[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.50 FLXAnFRRCV Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11 to 0	RCV[11:0]	Rate Correction Value Flag Indicates vRateCorrection

(1) FLXAnFRRCV.RCV

Rate Correction Value Flag

Indicates internal rate correction value (vRateCorrection/ two's complement) before limitation. If the value of this bit exceeds the limits defined by bits FLXAnFRGTUC10.MRC[10:0], flag FLXAnFRSFS.RCLR is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external rate correction value is added to the limited rate correction value.

24.3.7.6 FLXAnFROCV — FlexRay Offset Correction Value Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $011C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OCV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.51 FLXAnFROCV Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 0	OCV[18:0]	Offset Correction Value Flag Indicates vOffsetCorrection

(1) FLXAnFROCV.OCV

Offset Correction Value Flag

Indicates offset correction value (vOffsetCorrection/ two's complement) before limitation. If the value of this bit exceeds the limits defined by bits FLXAnFRGTUC10.MOC[10:0], flag FLXAnFRSFS.OCLR is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

CAUTION

The external offset correction value is added to the limited offset correction value.

24.3.7.7 FLXAnFRSFS — FlexRay Sync Frame Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0120_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RCLR	MRCS	OCLR	MOCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSBO[3:0]				VSBE[3:0]				VSAO[3:0]				VSAE[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.52 FLXAnFRSFS Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19	RCLR	Rate Correction Limit Reached Flag 0: Rate correction below limit 1: Rate correction limit reached
18	MRCS	Missing Rate Correction Signal Flag 0: Rate correction signal valid 1: Missing rate correction signal
17	OCLR	Offset Correction Limit Reached Flag 0: Offset correction below limit 1: Offset correction limit reached
16	MOCS	Missing Offset Correction Signal Flag 0: Offset correction signal valid 1: Missing offset correction signal
15 to 12	VSBO[3:0]	Valid Sync Frames Channel B, odd communication cycle
11 to 8	VSBE[3:0]	Valid Sync Frames Channel B, even communication cycle
7 to 4	VSAO[3:0]	Valid Sync Frames Channel A, odd communication cycle
3 to 0	VSAE[3:0]	Valid Sync Frames Channel A, even communication cycle

(1) FLXAnFRSFS.RCLR

Rate Correction Limit Reached Flag

The Rate Correction Limit Reached flag signals to the Host, that the rate correction value has exceeded its limit as defined by bits FLXAnFRGTUC10.MRC[10:0]. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(2) FLXAnFRSFS.MRCS

Missing Rate Correction Signal Flag

The Missing Rate Correction flag signals to the Host, that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(3) FLXAnFRSFS.OCLR

Offset Correction Limit Reached Flag

The Offset Correction Limit Reached flag signals to the Host, that the offset correction value has exceeded its limit as defined by bits FLXAnFRGTUC10.MOC[10:0]. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(4) FLXAnFRSFS.MOCS

Missing Offset Correction Signal Flag

The Missing Offset Correction flag signals to the Host, that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(5) FLXAnFRSFS.VSBO

Valid Sync Frames Channel B, odd communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHB is 1.

Holds the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(6) FLXAnFRSFS.VSBE

Valid Sync Frames Channel B, even communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHB is 1.

Holds the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(7) FLXAnFRSFS.VSAO

Valid Sync Frames Channel A, odd communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHA is 1.

Holds the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(8) FLXAnFRSFS.VSAE

Valid Sync Frames Channel A, even communication cycle

These bits are only valid when bit FLXAnFRSUCC1.CCHA is 1.

Holds the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by bit FLXAnFRSUCC1.TXSY, the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

24.3.7.8 FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register

Symbol window related status information is updated by the CC at the end of the symbol window for each channel. NIT related status information is updated by the CC at the end of the NIT for each channel.

During startup the status data is not updated.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0124_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.53 FLXAnFRSWNIT Register Contents (1/2)

Bit	Symbol	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	SBNB	Slot Boundary Violation during NIT Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B
10	SENB	Syntax Error during NIT Channel B Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel B
9	SBNA	Slot Boundary Violation during NIT Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A
8	SENA	Syntax Error during NIT Channel A Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel A
7	MTSB	MTS Received on Channel B Flag 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
6	MTSA	MTS Received on Channel A Flag 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
5	TCSB	Transmission Conflict in Symbol Window Channel B Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B
4	SBSB	Slot Boundary Violation in Symbol Window Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B
3	SESB	Syntax Error in Symbol Window Channel B Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel B

Table 24.53 FLXAnFRSWNIT Register Contents (2/2)

Bit	Symbol	Function
2	TCSA	Transmission Conflict in Symbol Window Channel A Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A
1	SBSA	Slot Boundary Violation in Symbol Window Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A
0	SESA	Syntax Error in Symbol Window Channel A Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel A

(1) FLXAnFRSWNIT.SBNB

Indicates a Slot Boundary Violation during NIT Channel B Flag (vSS!BViolationB).

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSWNIT.SENB

Indicates a Syntax Error during NIT Channel B Flag (vSS!SyntaxErrorB).

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRSWNIT.SBNA

Indicates a Slot Boundary Violation during NIT Channel A Flag (vSS!BViolationA).

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRSWNIT.SENA

Indicates a Syntax Error during NIT Channel A Flag (vSS!SyntaxErrorA).

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRSWNIT.MTSB

Indicates a MTS Received on Channel B Flag (vSS!ValidMTSB).

Media Access Test symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also bit FLXAnFRSIR.MTSB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRSWNIT.MTSA

Indicates a MTS Received on Channel A Flag (vSS!ValidMTSA).

Media Access Test symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also bit FLXAnFRSIR.MTSA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRSWNIT.TCSB

Indicates a Transmission Conflict in Symbol Window Channel B Flag (vSS!TxConflictB).

Reset when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRSWNIT.SBSB

Indicates a Slot Boundary Violation in Symbol Window Channel B Flag (vSS!BViolationB).

Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRSWNIT.SESB

Indicates a Syntax Error in Symbol Window Channel B Flag (vSS!SyntaxErrorB).

Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRSWNIT.TCSA

Indicates a Transmission Conflict in Symbol Window Channel A Flag (vSS!TxConflictA).

Reset when leaving CONFIG state or when entering STARTUP state.

(11) FLXAnFRSWNIT.SBSA

Indicates a Slot Boundary Violation in Symbol Window Channel A Flag (vSS!BViolationA).

Reset when leaving CONFIG state or when entering STARTUP state.

(12) FLXAnFRSWNIT.SESA

Indicates a Syntax Error in Symbol Window Channel A Flag (vSS!SyntaxErrorA).

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.9 FLXAnFRACS — FlexRay Aggregated Channel Status Register

Do not rewrite this register using bit manipulation instructions.

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception.

The aggregated channel status also includes status data from the symbol window and the network idle time.

The status data is updated (set) after each slot and aggregated until it is reset by the Host.

During startup the status data is not updated.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0128_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SBVB	CIB	CEDB	SEDB	VFRB	—	—	—	SBVA	CIA	CEDA	SEDA	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 24.54 FLXAnFRACS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SBVB	Slot Boundary Violation on Channel B Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B
11	CIB	Communication Indicator Channel B Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication
10	CEDB	Content Error Detected on Channel B Flag 0: No frame with content error received 1: Frame(s) with content error received on channel B
9	SEDB	Syntax Error Detected on Channel B Flag 0: No syntax error observed 1: Syntax error(s) observed on channel B
8	VFRB	Valid Frame Received on Channel B Flag 0: No valid frame received 1: Valid frame(s) received on channel B
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SBVA	Slot Boundary Violation on Channel A Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A

Table 24.54 FLXAnFRACS Register Contents (2/2)

Bit Position	Bit Name	Function
3	CIA	Communication Indicator Channel A Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication
2	CEDA	Content Error Detected on Channel A Flag 0: No frame with content error received 1: Frame(s) with content error received on channel A
1	SEDA	Syntax Error Detected on Channel A Flag 0: No syntax error observed 1: Syntax error(s) observed on channel A
0	VFRA	Valid Frame Received on Channel A Flag 0: No valid frame received 1: Valid frame(s) received on channel A

(1) FLXAnFRACS.SBVB

Slot Boundary Violation on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRACS.CIB

Communication Indicator Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel B in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had When any combination of either syntax error OR content error OR slot boundary violation.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

The FLXAnFRACS.CIB flag is also set when the slot boundary at the end of the slot is reached during the channel idle recognition phase while there is only a single frame in the slot.

(3) FLXAnFRACS.CEDB

Content Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRACS.SEDB

Syntax Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDB is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRACS.VFRB

Valid Frame Received on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel B in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRACS.SBVA

Slot Boundary Violation on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRACS.CIA

Communication Indicator Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

When this flag changes from 0 to 1, the FLXAnFREIR.EDA bit is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

The FLXAnFRACS.CIA flag is also set when the slot boundary at the end of the slot is reached during the channel idle recognition phase while there is only a single frame in the slot.

(8) FLXAnFRACS.CEDA

Content Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRACS.SEDA

Syntax Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.

When this flag changes from 0 to 1, bit FLXAnFREIR.EDA is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRACS.VFRA

Valid Frame Received on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

One or more valid frames were received on channel A in any static or dynamic slot during the observation period.

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.10 FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)

Registers FLXAnFRESID1 to FLXAnFRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXAnFRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFRESID1.RXEA, FLXAnFRESID1.RXEB are set. The value is updated during the NIT of each even communication cycle.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0130_H to <FLXAn_base> + 0168_H (<FLXAn_base> + 0130_H+(n-1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXEB	RXEA	—	—	—	—	EID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.55 FLXAnFRESIDn (n=1 to 15) Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXEB	Received / Configured Even Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames
14	RXEA	Received / Configured Even Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	EID[9:0]	Even Sync ID Flag (vsSyncIDListA,B even)

(1) FLXAnFRESIDn.RXEB

Received / Configured Even Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = bits FLXAnFRESID1.EID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRESIDn.RXEA

Received / Configured Even Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = bits FLXAnFRESID1.EID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRESIDn.EID

Even Sync ID Flag (vsSyncIDListA,B even)

Sync frame ID even communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.11 FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)

Registers FLXAnFROSID1 to FLXAnFROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXAnFROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFROSID1.RXOA, FLXAnFROSID1.RXOB are set. The value is updated during the NIT of each odd communication cycle.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0170_H to <FLXAn_base> + $01A8_H$ (<FLXAn_base> + 0170_H+ (n-1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXOB	RXOA	—	—	—	—	OID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.56 FLXAnFROSIDn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXOB	Received / Configured Odd Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames
14	RXOA	Received / Configured Odd Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	OID[9:0]	Odd Sync ID Flag (vsSyncIDListA,B odd)

(1) FLXAnFROSIDn.RXOB

Received / Configured Odd Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = bits FLXAnFROSID1.OID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFROSIDn.RXOA

Received / Configured Odd Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = bits FLXAnFROSID1.OID[9:0].

Reset when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFROSIDn.OID

Odd Sync ID Flag (vsSyncIDListA,B odd)

Sync frame ID odd communication cycle.

Reset when leaving CONFIG state or when entering STARTUP state.

24.3.7.12 FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)

The three network management registers hold the accrued NM vector (see **Section 24.4.7, Network Management**).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

FLXAnFRNMVm-bytes exceeding the configured NM vector length are not valid.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $01B0_H$ to <FLXAn_base> + $01B8_H$ (<FLXAn_base> + $01B0_H$ + (m-1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.57 FLXAnFRNMVn Register Contents

Bit Position	Bit Name	Function
31 to 0	NM[31:0]	NM Vector

(1) FLXAnFRNMVm.NM

NM Vector

The three network management vector registers hold the accrued NM vector (configurable 0 to 12 bytes). The NM vector to be held is generated by bit-wise logic OR for each NM vector received on each channel (valid static frames with PPI = 1) (see **Section 24.4.7, Network Management**).

FLXAnFRNMVm-bytes exceeding the configured NM vector length are not valid.

The Register Contents are updated at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

These bits are cleared when leaving CONFIG state or when entering STARTUP state.

24.3.8 Message Buffer Control Registers

24.3.8.1 FLXAnFRMRC — FlexRay Message RAM Configuration Register

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO.

The Message RAM can be divided into up three different areas; Static Buffer area, Static and Dynamic Buffer area, FIFO area. If present, the Static Buffer area is starting at Message Buffer 0.

The start of the Static and Dynamic Buffer area is configured by bits FLXAnFRMRC.FDB[7:0]. Bits FLXAnFRMRC.FDB[7:0] define the end of the Static Buffer area. If no Static Buffer area is present, the Static and Dynamic Buffer area starts at Message Buffer 0.

The start of the FIFO area is configured by bits FLXAnFRMRC.FFB[7:0]. Bits FLXAnFRMRC.FFB[7:0] define the end of the previous area, which can be either the Static Buffer area or the Static and Dynamic Buffer area. If no Static Buffer area and no Static and Dynamic Buffer area is present, the FIFO area starts at Message Buffer 0.

With bits FLXAnFRMRC.LCB[7:0], the end of the last configured area is configured which can be the Static Buffer area, the Static and Dynamic Buffer area or the FIFO area.

Figure 24.2 shows an example configuration of the Message RAM where all there area are configured.

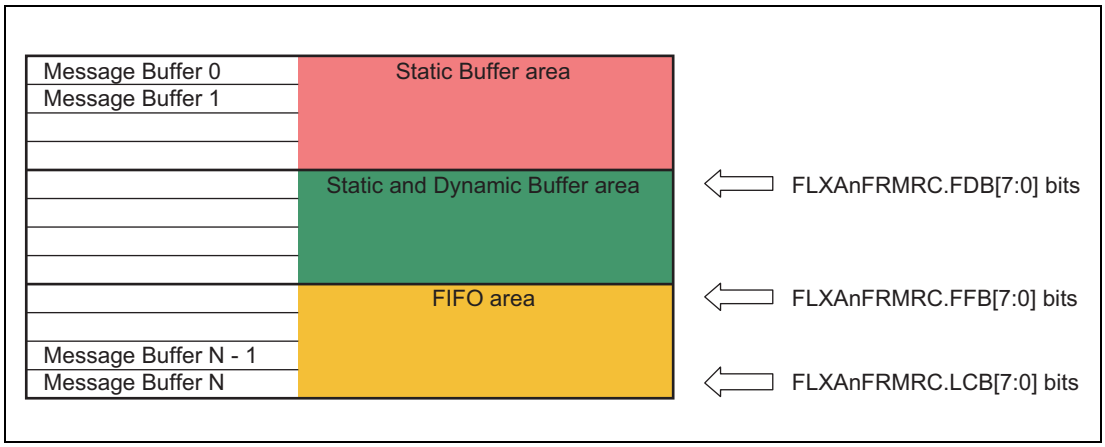


Figure 24.2 Message RAM Organization

CAUTIONS

1. **In case the node is configured as sync node (bit FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (bit FLXAnFRSUCC1.TSM = 1), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 resp. 1 is treated like all other message buffers.**
2. **The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see Section 24.4.13, Message RAM.**
3. **In case two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the “Static Buffers” or at the beginning of the “Static + Dynamic Buffers” section.**

4. **The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.**
5. **The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via the FLXAnFRWRHS2.PLC[6:0] and FLXAnFRWRHS3.DP[6:0] bits. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.**

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0300_H

Value after reset: $0180\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPLM	SEC[1:0]		LCB[7:0]							
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FFB[7:0]								FDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.58 FLXAnFRMRC Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	SPLM	Sync Frame Payload Multiplex Bit 0: Only message buffer 0 locked against reconfiguration 1: Both message buffers 0 and 1 are locked against reconfiguration
25, 24	SEC[1:0]	Secure Buffers Bit 00: all buffers unlocked 01: static buffers locked, FIFO locked, limited transmission 10: all buffers locked 11: all buffers locked, limited transmission
23 to 16	LCB[7:0]	Last Configured Buffer Bit 0 to 127: Number of message buffers is FLXAnFRMRC.LCB + 1 128: No message buffer assigned to the FIFO
15 to 8	FFB[7:0]	First Buffer of FIFO Bit 0: All message buffers assigned to the FIFO 1 to 127: Message buffers from FLXAnFRMRC.FFB to FLXAnFRMRC.LCB assigned to the FIFO 128: No message buffer configured
7 to 0	FDB[7:0]	First Dynamic Buffer Bit 0: No group of message buffers exclusively for the static segment configured 1 to 127: Message buffers 0 to FLXAnFRMRC.FDB - 1 reserved for static segment 128: No dynamic message buffers configured

(1) FLXAnFRMRC.SPLM

Sync Frame Payload Multiplex Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

This bit is only evaluated if the node is configured as sync node (bit FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (bit FLXAnFRSUCC1.TSM = 1).

When this bit is set to 1 message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B.

When this bit is set to 0, sync frames are transmitted from message buffer 0 with the same payload data on all channels configured. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen according to this bit setting.

(2) FLXAnFRMRC.SEC

Secure Buffers Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state.

For temporary unlocking, see **Section 24.4.13.4, Host Handling of Access Errors**.

00_B = all buffers unlocked

Reconfiguration of message buffers enabled with numbers < FLXAnFRMRC.FFB enabled

Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if bit FLXAnFRMRC.SPLM = 1, also message buffer 1) is always locked

01_B = static buffers locked, FIFO locked, limited transmission

Reconfiguration of message buffers with numbers < FLXAnFRMRC.FDB and with numbers ≥ FLXAnFRMRC.FFB locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

10_B = all buffers locked

Reconfiguration of all message buffers locked

11_B = all buffers locked, limited transmission

Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

(3) FLXAnFRMRC.LCB

Last Configured Buffer Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (bits FLXAnFRMRC.FDB[7:0] < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FDB.

When a FIFO area is configured (bits FLXAnFRMRC.FFB[7:0] < 128), the user should configure bits FLXAnFRMRC.LCB[7:0] ≥ bits FLXAnFRMRC.FFB[7:0].

(4) FLXAnFRMRC.FFB

First Buffer of FIFO Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (bits FLXAnFRMRC.FDB[7:0] < 128), the user should configure bits FLXAnFRMRC.FFB[7:0] > bits FLXAnFRMRC.FDB[7:0].

(5) FLXAnFRMRC.FDB

First Dynamic Buffer Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

24.3.8.2 FLXAnFRFRF — FlexRay FIFO Rejection Filter Register

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0304_H

Value after reset: $0180\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNF	RSS	CYF[6:0]						
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FID[10:0]											CH[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.59 FLXAnFRFRF Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	RNF	Reject Null Frames Bit 0: Null frames are stored in the FIFO 1: Reject all null frames
23	RSS	Reject in Static Segment Bit 0: FIFO also used for static segment 1: Reject messages in static segment
22 to 16	CYF[6:0]	Cycle Counter Filter Bit
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	FID[10:0]	Frame ID Filter Bit 0 to 2047: Frame ID filter values
1, 0	CH[1:0]	Channel Filter Bit 00: receive on both channels 01: receive only on channel B 10: receive only on channel A 11: no reception

(1) FLXAnFRFRF.RNF

Reject Null Frames Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, received null frames are not stored in the FIFO.

(2) FLXAnFRFRF.RSS

Reject in Static Segment Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, the FIFO is used only for the dynamic segment.

(3) FLXAnFRFRF.CYF

Cycle Counter Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected. For details about the configuration of the cycle counter filter see **Section 24.4.8.2, Cycle Counter Filtering**.

(4) FLXAnFRFRF.FID

Frame ID Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXAnFRFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When bits FLXAnFRFRFM.MFID[10:0] are zero, a frame ID filter value of zero means that no frame ID is rejected.

(5) FLXAnFRFRF.CH

Channel Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

24.3.8.3 FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register

The FlexRay FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set to 1, it indicates that the corresponding bit in the FLXAnFRFRF register will not be considered for rejection filtering.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0308_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MFID[10:0]											—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 24.60 FLXAnFRFRFM Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	MFID[10:0]	Mask Frame ID Filter Bit 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRFRFM.MFID

Mask Frame ID Filter Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

24.3.8.4 FLXAnFRFCL — FlexRay FIFO Critical Level Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $030C_H$

Value after reset: $0000\ 0080_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.61 FLXAnFRFCL Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CL[7:0]	Critical Level Bit Critical Level

(1) FLXAnFRFCL.CL

Critical Level Bit

The user can only write to these bits when bits FLXAnFRCCSV.POCS[5:0] are DEFAULT_CONFIG or CONFIG.

When the receive FIFO fill level FLXAnFRFSR.RFFL is equal or greater than the critical level configured by bits FLXAnFRFCL.CL[7:0], the receive FIFO critical level flag FLXAnFRFSR.RFCL is set to 1.

If FLXAnFRFCL.CL is programmed to values > 128, bit FLXAnFRFSR.RFCL is never set to 1.

24.3.9 Message Buffer Status Registers

24.3.9.1 FLXAnFRMHDS — FlexRay Message Handler Status Register

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0310_H

Value after reset: $0000\ 0080_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MBU[6:0]							—	MBT[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	FMB[6:0]							CRAM	MFMB	FMBD	ATBF2	ATBF1	AMR	—	—
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Table 24.62 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MBU[6:0]	Message Buffer Updated Flag
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MBT[6:0]	Message Buffer Transmitted Flag
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	FMB[6:0]	Faulty Message Buffer Number Flag
7	CRAM	Clear all internal RAM's Flag 0: No execution of the CHI command CLEAR_RAMS 1: Execution of the CHI command CLEAR_RAMS ongoing
6	MFMB	Multiple Faulty Message Buffer Detection Flag 0 = No additional faulty message buffer. 1 = Additional faulty message buffer was detected while the FMBD flag is set to 1.
5	FMBD	Faulty Message Buffer Detection Flag 0 = No faulty message buffer. 1 = Message buffer referenced by bit FLXAnFRMHDS.FMB holds faulty data with a parity error.
4	ATBF2	TBFRAM B Access Error Flag 0 = No access error 1 = Access error occurred when reading the RAM B.
3	ATBF1	TBFRAM A Access Error Flag 0 = No access error. 1 = Access error occurred when reading the RAM A.
2	AMR	Message RAM Access Error Flag 0 = No access error 1 = Access error occurred when reading the Message RAM.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRMHDS.MBU

Message Buffer Updated Flag

Number of message buffer that was updated last by the CC. For this message buffer the respective ND and / or MBC flag in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers and the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers are also set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDS.MBT

Message Buffer Transmitted Flag

Number of last successfully transmitted message buffer.

If the message buffer is configured for single-shot mode, the respective TXR flag in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers was reset to 0.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDS.FMB

Faulty Message Buffer Number Flag

This flag indicates that an access error occurred when reading from the message buffer referenced by FLXAnFRMHDS.FMB.

The value of this flag is only valid when one of flags FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, FLXAnFRMHDS.ATBF2, and FLXAnFRMHDS.FMBD is set to 1.

This flag is not updated while the FLXAnFRMHDS.FMBD flag is 1.

This flag is cleared by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDS.CRAM

Internal RAM Clear Flag

This flag indicates that the CHI command CLEAR_RAMS is ongoing (all bits of the message RAM, input buffer, output buffer and TBF are written to 0).

This flag is set by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDS.MFMB

Multiple Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that an additional faulty message buffer was detected while the FLXAnFRMHDS.FMBD flag is set.

This bit is cleared by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDS.FMBD

Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that the message buffer referenced by FLXAnFRMHDS.FMB holds faulty data due to an access error.

This bit is cleared by the CHI command CLEAR_RAMS.

(7) FLXAnFRMHD.ATBF2

TBFRAM B Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the TBFRAM B.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHD.ATBF1

TBFRAM A Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the TBFRAM A.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDS.AMR

Message RAM Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the Message RAM.

CAUTION

When this flag changes from 0 to 1, the FLXAnFREIR.AERR bit is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

24.3.9.2 FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0314_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	LDTB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LDTA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.63 FLXAnFRLDTS Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	LDTB[10:0]	Last Dynamic Transmission Channel B Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	LDTA[10:0]	Last Dynamic Transmission Channel A Flag

(1) FLXAnFRLDTS.LDTB

Last Dynamic Transmission Channel B Flag

Stores the value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS of channel B.

(2) FLXAnFRLDTS.LDTA

Last Dynamic Transmission Channel A Flag

Stores the value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS of channel A.

24.3.9.3 FLXAnFRFSR — FlexRay FIFO Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0318_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFFL[7:0]								—	—	—	—	—	RFO	RFCL	RFNE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.64 FLXAnFRFSR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	RFFL[7:0]	Receive FIFO Fill Level Flag
7 to 3	Reserved	When read, the value after reset is returned.
2	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
1	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
0	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty

(1) FLXAnFRFSR.RFFL

Receive FIFO Fill Level Flag

Indicates the number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRFSR.RFO

Receive FIFO Overrun Flag

The flag is set to 1 by the CC when a receive FIFO overrun is detected.

When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, the FLXAnFREIR.RFO flag is set to 1.

The flag is cleared by the next FIFO read access issued by the Host.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRFSR.RFCL

Receive FIFO Critical Level Flag

This flag is set to 1 when the value of the receive FIFO fill level FLXAnFRFSR.RFFL[7:0] is equal or greater than the critical level as configured by FLXAnFRFCL.CL.

When this bit changes from 0 to 1 bit FLXAnFRSIR.RFCL is set to 1, and if enabled, an interrupt is generated.

The flag is cleared by the CC as soon as the value of the FLXAnFRFSR.RFFL[7:0] bits drops below the value set in the FLXAnFRFCL.CL bit.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRFSR.RFNE

Receive FIFO Not Empty Flag

This flag is set to 1 by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, the FLXAnFRSIR.RFNE bit is set to 1.

The bit is reset to 0 after the Host has read all messages from the FIFO.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

24.3.9.4 FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register

Do not rewrite this register using bit manipulation instructions.

Some constraints exist for the Message Handler regarding bus clock frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FLXAnFRMHDF.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $031C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.65 FLXAnFRMHDF Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	WAHP	Write Attempt to Header Partition Flag 0: No write attempt to header partition 1: Write attempt to header partition
7	TNSB	Transmission Not Started Channel B Flag 0: No transmission not started on channel B 1: Transmission not started on channel B
6	TNSA	Transmission Not Started Channel A Flag 0: No transmission not started on channel A 1: Transmission not started on channel A
5	TBFB	TBFB Access Failure B Flag 0: No TBFB access failure 1: TBFB access failure
4	TBFA	Temporary buffer Access Failure A Flag 0: No TBFA access failure 1: TBFA access failure
3	FNFB	Find Sequence Not Finished Channel B Flag 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B
2	FNFA	Find Sequence Not Finished Channel A Flag 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A
1	SNUB	Status Not Updated Channel B Flag 0: No overload condition occurred when updating MBS (FLXAnFRMBS) for channel B 1: Message buffer status (FLXAnFRMBS) for channel B not updated
0	SNUA	Status Not Updated Channel A Flag 0: No overload condition occurred when updating MBS (FLXAnFRMBS) for channel A 1: Message buffer status (FLXAnFRMBS) for channel A not updated

(1) FLXAnFRMHDF.WAHP

Write Attempt to Header Partition Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

Outside DEFAULT_CONFIG and CONFIG state this flag is set to 1 by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.

When this flag changes from 0 to 1, the FLXAnFREIR.MHF flag is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDF.TNSB

Transmission Not Started Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDF.TNSA

Transmission Not Started Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDF.TBFB

TBFB Access Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when a read or write access to TBFB requested by PRT (Protocol controller) B could not complete within the available time.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDF.TBFA

TBFA Access Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when a read or write access to TBFA requested by PRT A could not complete within the available time.

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDF.FNFB

Find sequence Not Finished flag, channel B

Writing 0 has no effect on the bit's value.

Writing a 1 to the bit clears it.

This flag is set to 1 by the CC when the message handler, due to being overloaded, was not able to finish a find sequence (scan of message RAM for a matching message buffer).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is also set to 1.

The value is reset on leaving the CONFIG state or entering the STARTUP state.

The value is also reset by the CLEAR_RAMS CHI command.

(7) FLXAnFRMHDF.FNFA

Find sequence Not Finished flag, channel A

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHDF.SNUB

Status Not Updated Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDF.SNUA

Status Not Updated Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing a 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1, flag FLXAnFREIR.MHF is set to 1.

Reset when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

24.3.9.5 FLXAnFRTXRQm — FlexRay Transmission Request m (m = 1 to 4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are evaluated for transmit buffers only. If the number of configured message buffers is less than 128, the remaining TXR flags have no meaning.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0320_H to <FLXAn_base> + $032C_H$ (<FLXAn_base> + 0320_H + (m–1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXRp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXRp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.66 FLXAnFRTXRQm Register Contents

Bit Position	Bit Name	Function
31 to 0	TXRp[31:0]	Transmission Request Flag p

(1) FLXAnFRTXRQm.TXRp (p = (m – 1) × 32 to (m × 32 – 1))

Transmission Request Flag p

If the flag is set to 1, the respective message buffer is ready for transmission respectively transmission of this message buffer is in progress.

In single-shot mode the flags are reset to 0 after transmission has completed.

This bit is cleared by the CHI command CLEAR_RAMS.

24.3.9.6 FLXAnFRNDATm — FlexRay New Data Register m (m = 1 to 4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, the remaining ND flags have no meaning.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0330_H to <FLXAn_base> + $033C_H$ (<FLXAn_base> + 0330_H + (m-1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NDp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.67 FLXAnFRNDATm Register Contents

Bit Position	Bit Name	Function
31 to 0	NDp[31:0]	New Data Flag p

(1) FLXAnFRNDATm.NDp (p = (m – 1) × 32 to (m × 32 – 1))

New Data Flag p

The flags are set to 1 when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer.

The flags are not set to 1 after reception of null frames except for message buffers belonging to the receive FIFO.

An ND flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMS.

24.3.9.7 FLXAnFRMBSCm — FlexRay Message Buffer Status Changed Register m (m = 1 to 4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, the remaining MBC flags have no meaning.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0340_H to <FLXAn_base> + $034C_H$ (<FLXAn_base> + 0340_H + (m–1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBCp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBCp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.68 FLXAnFRMBSCm Register Contents

Bit Position	Bit Name	Function
31 to 0	MBCp[31:0]	Message Buffer Status Changed Flag p

(1) FLXAnFRMBSCm.MBCp (p = (m – 1) × 32 to (m × 32 – 1))

Message Buffer Status Changed Flag p

Indicates whether the Message Handler has changed one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, FTB in the header section (see **Section 24.3.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** and **Section 24.4.13.1, Header Partition**) of the respective message buffer.

An MBC flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Reset when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMS.

24.3.10 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in **Section 24.3.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via bits FLXAnFRWRHS2.PLC[6:0] and bits FLXAnFRWRHS3.DP[10:0]. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in **Section 24.4.12.2, Host access to Message RAM**.

These registers cannot be written when the input data transfer function shown in **Section 24.4.16.1, Input Data Transfer** is used and the FLXAnFRITS.ITS bit is 1.

24.3.10.1 FLXAnFRWRDSm — FlexRay Write Data Section Register m (m = 1 to 64)

This register holds the data words to be transferred to the data section of the specified message buffer. The number of data words written to the Message RAM is defined by the payload length configured in bits FLXAnFRWRHS2.PLC[6:0].

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0400_H to <FLXAn_base> + $04FC_H$(<FLXAn_base> + 0400_H+(m–1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.69 FLXAnFRWRDSm Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data Bit

(1) FLXAnFRWRDSm.MD

Message Data Bit

Reset by the CHI command CLEAR_RAMS.

CAUTIONS

1. In case the FLXAnFRWRHS2.PLC[6:0] bits specifies an odd payload length, the remaining message data bytes are unused.
2. When writing to the FLXAnFRWRDSm register, all bytes have to be written by one 32-bit access, two consecutive 16-bit accesses, or four consecutive 8- bit accesses before the transfer from the Input Buffer to the Message RAM is started. If not all bytes of a 32- bit word have been written by the Host (8/16-bit access only), the FLXAnFRWRDSm register holds partly undefined data.

24.3.10.2 FLXAnFRWRHS1 — FlexRay Write Header Section Register 1

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0500_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.70 FLXAnFRWRHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MBI	Message Buffer Interrupt Bit 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled
28	TXM	Transmission Mode Setting Bit 0: Continuous mode 1: Single-shot mode
27	PPIT	Payload Preamble Indicator Transmit Bit 0: Payload Preamble Indicator is set to 0 1: Payload Preamble Indicator is set to 1
26	CFG	Message Buffer Direction Configuration Bit 0: The corresponding buffer is configured as Receive Buffer 1: The corresponding buffer is configured as Transmit Buffer
25,24	CH[1:0]	Channel Filter Control Bit
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CYC[6:0]	Cycle Code Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	FID[10:0]	Frame ID Bit

(1) FLXAnFRWRHS1.MBI

Message Buffer Interrupt Enable Bit

This bit enables the message buffer interrupt.

After a dedicated receive buffer has been updated by the Message Handler, bit FLXAnFRSIR.RXI and /or bit FLXAnFRSIR.MBSI are set to 1. After a transmission has completed bit FLXAnFRSIR.TXI is set to 1.

(2) FLXAnFRWRHS1.TXM

Transmission Mode Setting Bit

This bit selects transmit mode of the corresponding message buffer. For transmit mode, see **Section 24.4.9.3, Transmit Buffers**.

(3) FLXAnFRWRHS1.PPIT

Payload Preamble Indicator Transmit Bit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames of the corresponding message buffer.

If the bit is set to 1 in a static message buffer, the respective message buffer holds network management information.

If the bit is set to 1 in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the Host.

(4) FLXAnFRWRHS1.CFG

Message Buffer Direction Configuration Bit

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated.

When an unused 32-bit or larger area is not allocated to the head of the data partition, set the data section of the message buffer that is allocated immediately after the header partition (of the last buffer) as a transmit buffer by setting this bit to 1.

(5) FLXAnFRWRHS1.CH

Channel Filter Control Bit

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

CH[1:0]	Transmit Buffer transmit frame on	Receive Buffer store frame received from
00	No transmission	Ignore frame
01	Channel A	Channel A
10	Channel B	Channel B
11	Both channels (static segment only)	Channel A or B (store first semantically valid frame; static segment only)

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CH = “00_B”)

(6) FLXAnFRWRHS1.CYC

Cycle Code Bit

The 7-bit cycle code determines the cycle set used for cycle counter filtering.

For details about the configuration of the cycle code **Section 24.4.8.2, Cycle Counter Filtering**.

(7) FLXAnFRWRHS1.FID

Frame ID Bit

Frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message.

Message buffers with frame ID = 0 are considered as not valid.

24.3.10.3 FLXAnFRWRHS2 — FlexRay Write Header Section Register 2

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0504_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.71 FLXAnFRWRHS2 Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	PLC[6:0]	Payload Length Configured Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	CRC[10:0]	Header CRC Bit (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC is configured

(1) FLXAnFRWRHS2.PLC

Payload Length Configured Bit

Length of data section (number of 2-byte words) as configured by the Host.

During static segment the static frame payload length as configured by bits FLXAnFRMHDC.SFDL[6:0] defines the payload length for all static frames. If the payload length configured by bits FLXAnFRWRHS2.PLC[6:0] is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is "0000_H" (see **Section 24.4.9.3, Transmit Buffers**).

(2) FLXAnFRWRHS2.CRC

Header CRC Bit (vRF!Header!HeaderCRC)

Setting of the receive buffer is not required.

Transmitting of the message buffer needs the header CRC calculation and setting.

For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by bits FLXAnFRMHDC.SFDL[6:0].

24.3.10.4 FLXAnFRWRHS3 — FlexRay Write Header Section Register 3

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0508_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.72 FLXAnFRWRHS3 Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	DP[10:0]	Data Pointer Bit

(1) FLXAnFRWRHS3.DP

Data Pointer Bit

Configures the pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

24.3.10.5 FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register

Configures how the message buffer in the Message RAM selected by register FLXAnFRIBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits FLXAnFRIBCM.LHSH, FLXAnFRIBCM.LDSH, and FLXAnFRIBCM.STXRH are swapped with bits FLXAnFRIBCM.LHSS, FLXAnFRIBCM.LDSS, and FLXAnFRIBCM.STXRS.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0510_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRS	LDSS	LHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRH	LDSH	LHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24.73 FLXAnFRIBCM Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	STXRS	Set Transmission Request Shadow Flag 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished)
17	LDSS	Load Data Section Shadow Flag 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
16	LHSS	Load Header Section Shadow Flag 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	STXRH	Set Transmission Request Host Bit 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission
1	LDSH	Load Data Section Host Bit 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM
0	LHSH	Load Header Section Host Bit 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM

(1) FLXAnFRIBCM.STXRS

Set Transmission Request Shadow Flag

(2) FLXAnFRIBCM.LDSS

Load Data Section Shadow Flag

(3) FLXAnFRIBCM.LHSS

Load Header Section Shadow Flag

(4) FLXAnFRIBCM.STXRH

Set Transmission Request Host Bit

If this bit is set to 1, the TXR flag for the selected message buffer is set in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed.

TXR is evaluated for transmit buffers only.

(5) FLXAnFRIBCM.LDSH

Set Load Data Section Host Bit

(6) FLXAnFRIBCM.LHSH

Set Load Header Section Host Bit

24.3.10.6 FLXAnFRIBCR — FlexRay Input Buffer Command Request Register

When the Host writes the number of the target message buffer in the Message RAM to bits FLXAnFRIBCR.IBRH[6:0], IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and bits FLXAnFRIBCR.IBRS[6:0] are also swapped (see **Section 24.4.12.2, (1) Data Transfer from Input Buffer to Message RAM**).

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by bits FLXAnFRIBCR.IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to bits FLXAnFRIBCR.IBRH[6:0].

If a write access to bits FLXAnFRIBCR.IBRH[6:0] occurs while bit FLXAnFRIBCR.IBSYS is 1, bit FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, bit FLXAnFRIBCR.IBSYH is reset to 0. Bit FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and bits FLXAnFRIBCR.IBRS[6:0] are also swapped.

Any write access to an Input Buffer register while both bit FLXAnFRIBCR.IBSYS and bit FLXAnFRIBCR.IBSYH are set to 1 will cause the error flag FLXAnFREIR.IIBA to be set to 1.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0514_H

Value after reset: 00000000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IBSYS	—	—	—	—	—	—	—	—	IBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IBSYH	—	—	—	—	—	—	—	—	IBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.74 FLXAnFRIBCR Register Contents (1/2)

Bit Position	Bit Name	Function
31	IBSYS	Input Buffer Busy Shadow Flag 0: Transfer between IBF Shadow and Message RAM completed 1: Transfer between IBF Shadow and Message RAM in progress
30 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IBRS[6:0]	Input Buffer Request Shadow Flag
15	IBSYH	Input Buffer Busy Host Flag 0: No request pending 1: Request while transfer between IBF Shadow and Message RAM in progress

Table 24.74 FLXAnFRIBCR Register Contents (2/2)

Bit Position	Bit Name	Function
14 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	IBRH[6:0]	Input Buffer Request Host Bit

(1) FLXAnFRIBCR.IBSYS

Input Buffer Busy Shadow Flag

Set to 1 after writing bits FLXAnFRIBCR.IBRH[6:0].

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

When the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0.

(2) FLXAnFRIBCR.IBRS

Input Buffer Request Shadow Flag

Number of the target message buffer actually updated / lately updated.

(3) FLXAnFRIBCR.IBSYH

Input Buffer Busy Host Flag

Set to 1 by writing bits FLXAnFRIBCR.IBRH[6:0] while bit FLXAnFRIBCR.IBSYS is still 1.

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

After the ongoing transfer between IBF Shadow and the Message RAM has completed, the FLXAnFRIBCR.IBSYH is set back to 0.

(4) FLXAnFRIBCR.IBRH

Input Buffer Request Host Bit

Selects the target message buffer in the Message RAM for data transfer from Input Buffer.

24.3.11 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in **Section 24.4.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

These registers cannot be written when the output data transfer function shown in **Section 24.4.16.2, Output Data Transfer**, in Output Data Transfer is used and the FLXAnFROTS.OTS bit is 1.

24.3.11.1 FLXAnFRRDDSm — FlexRay Read Data Section Register m (m = 1 to 64)

Holds the data words read from the data section of the addressed message buffer. This register holds the data wards to be transferred to the data section of the specified message buffer. The number of data words (DWn) read from the Message RAM is defined by the payload length configured in bits FLXAnFRRDHS2.PLC[6:0].

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0600_H to <FLXAn_base> + $06FC_H$ (<FLXAn_base> + 0600_H + (m–1) × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.75 FLXAnFRRDDSm Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data

(1) FLXAnFRRDDSm.MD

Message Data Flag

Cleared by the CHI command CLEAR_RAMS.

CAUTION

In case the FLXAnFRWRHS2.PLC[6:0] bits specifies an odd payload length, the remaining message data bytes are unused.

24.3.11.2 FLXAnFRRDHS1 — FlexRay Read Header Section Register 1

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0700_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.76 FLXAnFRRDHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	MBI	Message Buffer Interrupt Flag
28	TXM	Transmission Mode Flag
27	PPIT	Payload Preamble Indicator Transmit Flag
26	CFG	Message Buffer Direction Configuration Flag
25, 24	CH[1:0]	Channel Filter Control Flag
23	Reserved	When read, the value after reset is returned.
22 to 16	CYC[6:0]	Cycle Code
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	FID[10:0]	Frame ID

(1) FLXAnFRRDHS1.MBI

Message Buffer Interrupt Flag

Values as configured by the Host via bit FLXAnFRWRHS1.MBI.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(2) FLXAnFRRDHS1.TXM

Transmission Mode Flag

Values as configured by the Host via bit FLXAnFRWRHS1.TXM.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(3) FLXAnFRRDHS1.PPIT

Payload Preamble Indicator Transmit Flag

Values as configured by the Host via bit FLXAnFRWRHS1.PPIT.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(4) FLXAnFRRDHS1.CFG

Message Buffer Direction Configuration Flag

Values as configured by the Host via bit FLXAnFRWRHS1.CFG.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(5) FLXAnFRRDHS1.CH

Channel Filter Control Flag

Values as configured by the Host via bit FLXAnFRWRHS1.CH.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(6) FLXAnFRRDHS1.CYC

Cycle Code

Values as configured by the Host via bit FLXAnFRWRHS1.CYC.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(7) FLXAnFRRDHS1.FID

Frame ID

Values as configured by the Host via bit FLXAnFRWRHS1.FID.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are holding the received frame ID.

24.3.11.3 FLXAnFRRDHS2 — FlexRay Read Header Section Register 2

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0704_H

Value after reset: $0000\ 0000_H$

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the FLXAnFRRDHS2 register is updated from data frames only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLR[6:0]							—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.77 FLXAnFRRDHS2 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 24	PLR[6:0]	Payload Length Received Flag (vRF!Header!Length)
23	Reserved	When read, the value after reset is returned.
22 to 16	PLC[6:0]	Payload Length Configuration Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	CRC[10:0]	Header CRC Flag (vRF!Header!HeaderCRC)

(1) FLXAnFRRDHS2.PLR

Payload Length Received Flag (vRF!Header!Length)

Payload length (vRF!Header!Length) value updated from received data frames (exception: if message buffer belongs to the receive FIFO FLXAnFRRDSH2.PLR is also updated from received null frames).

(2) FLXAnFRRDHS2.PLC

Payload Length Configuration Flag

Length of data section (number of 2-byte words) as configured by the Host.

(3) FLXAnFRRDHS2.CRC

Header CRC Flag (vRF!Header!HeaderCRC)

Receive Buffer: Header CRC (vRF!Header!HeaderCRC) updated from received data frames

Transmit Buffer: Header CRC configured by the Host

(4) Data storage

When a message is stored into a message buffer the following behavior with respect to payload length received and payload length configured is implemented:

Bits FLXAnFRRDHS2.PLR[6:0] > Bits FLXAnFRRDHS2.PLC[6:0]:

The payload data stored in the message buffer is truncated to the payload length configured if bits FLXAnFRRDHS2.PLC[6:0] even or else truncated to bits FLXAnFRRDHS2.PLC[6:0] + 1.

Bits FLXAnFRRDHS2.PLR[6:0] <= Bits FLXAnFRRDHS2.PLC[6:0]:

The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by bits FLXAnFRRDHS2.PLC[6:0] are filled with undefined data.

Bits FLXAnFRRDHS2.PLR[6:0] = zero:

The message buffer’s data section is filled with undefined data

Bits FLXAnFRRDHS2.PLC[6:0] = zero:

Message buffer has no data section configured. No data is stored into the message buffer's data section.

CAUTIONS

1. The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is the FLXAnFRRDHS2.PLC[6:0] bits rounded to the next even value.
2. The FLXAnFRRDHS2.PLC[6:0] bits should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.
   For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the FLXAnFRRDHS2 register is updated from data frames only.

24.3.11.4 FLXAnFRRDHS3 — FlexRay Read Header Section Register 3

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area, the FLXAnFRRDHS3 register is updated from data frames only.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0708_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RES	PPI	NFI	SYN	SFI	RCI	—	—	RCC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.78 FLXAnFRRDHS3 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RES	Reserved Bit Indicator Flag (vRF!Header!Reserved)
28	PPI	Payload Preamble Indicator (vRF!Header!PPIndicator)
27	NFI	Null Frame Indicator Flag (vRF!Header!NFIndicator) 0: Up to now no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer
26	SYN	Sync Frame Indicator Flag (vRF!Header!SyFIndicator) 0: The received frame is not a sync frame 1: The received frame is a sync frame
25	SFI	Startup Frame Indicator Flag (vRF!Header!SuFIndicator) 0: The received frame is not a startup frame 1: The received frame is a startup frame
24	RCI	Received on Channel Indicator Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	RCC[5:0]	Receive Cycle Counter (vRF!Header!CycleCount)
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	DP[10:0]	Data Pointer Flag

(1) FLXAnFRRDHS3.RES

Reserved Bit Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

(2) FLXAnFRRDHS3.PPI

Payload Preamble Indicator Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

0 = The payload segment of the received frame does not contain a network management vector nor a message ID

1 = Static segment: Network management vector in the first part of the payload
Dynamic segment: Message ID in the first part of the payload

(3) FLXAnFRRDHS3.NFI

Null Frame Indicator Flag (vRF!Header!NFIndicator)

Is set to 1 after storage of the first received data frame.

(4) FLXAnFRRDHS3.SYN

Sync Frame Indicator Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

(5) FLXAnFRRDHS3.SFI

Startup Frame Indicator Flag (vRF!Header!SuFIndicator)

A startup frame is marked by the startup frame indicator.

(6) FLXAnFRRDHS3.RCI

Received on Channel Indicator Flag (vSS!Channel)

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

(7) FLXAnFRRDHS3.RCC

Receive Cycle Counter (vRF!Header!CycleCount)

Cycle counter value updated from received data frame.

(8) FLXAnFRRDHS3.DP

Data Pointer Flag

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

The bit value is the same as that set in bit FLXAnFRWRHS3.DP

24.3.11.5 FLXAnFRMBS — FlexRay Message Buffer Status Register

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer.

The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated.

The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all FLXAnFRMBS flags are reset to zero independent of which FLXAnFRIBCM bits are set or not.

For details about receive / transmit filtering see **Section 24.4.8, Filtering and Masking**, **Section 24.4.9, Transmit Process** and **Section 24.4.10, Receive Process**.

Whenever the Message Handler changes one of the flags FLXAnFRMBS.VFRA, FLXAnFRMBS.VFRB, FLXAnFRMBS.SEOA, FLXAnFRMBS.SEOB, FLXAnFRMBS.CEOA, FLXAnFRMBS.CEOB, FLXAnFRMBS.SVOA, FLXAnFRMBS.SVOB, FLXAnFRMBS.TCIA, FLXAnFRMBS.TCIB, FLXAnFRMBS.ESA, FLXAnFRMBS.ESB, FLXAnFRMBS.MLST, FLXAnFRMBS.FTA, FLXAnFRMBS.FTB the respective message buffer's MBC flag in registers FLXAnFRMBSC1 to FLXAnFRMBSC4 is set.

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $070C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	—	—	CCS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.79 FLXAnFRMBS Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RESS	Reserved Bit Status Flag (vRF!Header!Reserved)
28	PPIS	Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator) 0: PPI indicator set to 0 1: PPI indicator set to 1
27	NFIS	Null Frame Indicator Status Flag (vRF!Header!NFIndicator) 0: Received frame is a null frame 1: Received frame is not a null frame
26	SYNS	Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator) 0: No sync frame received 1: The received frame is a sync frame

Table 24.79 FLXAnFRMBS Register Contents (2/2)

Bit Position	Bit Name	Function
25	SFIS	Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator) 0: No startup frame received 1: The received frame is a startup frame
24	RCIS	Received on Channel Indicator Status Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	CCS[5:0]	Cycle Count Status Flag
15	FTB	Frame Transmitted on Channel B Flag 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B
14	FTA	Frame Transmitted on Channel A Flag 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A
13	Reserved	When read, the value after reset is returned.
12	MLST	Message Lost Flag 0: No message lost 1: Unprocessed message was overwritten
11	ESB	Empty Slot Channel B Flag 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B
10	ESA	Empty Slot Channel A Flag 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A
9	TCIB	Transmission Conflict Indication Channel B Flag (vSS!TxConflictB) 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B
8	TCIA	Transmission Conflict Indication Channel A Flag (vSS!TxConflictA) 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A
7	SVOB	Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB) 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B
6	SVOA	Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA) 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A
5	CEOB	Content Error Observed on Channel B Flag (vSS!ContentErrorB) 0: No content error observed on channel B 1: Content error observed on channel B
4	CEOA	Content Error Observed on Channel A Flag (vSS!ContentErrorA) 0: No content error observed on channel A 1: Content error observed on channel A
3	SEOB	Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB) 0: No syntax error observed on channel B 1: Syntax error observed on channel B
2	SEOA	Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA) 0: No syntax error observed on channel A 1: Syntax error observed on channel A
1	VFRB	Valid Frame Received on Channel B (vSS!ValidFrameB) 0: No valid frame received on channel B 1: Valid frame received on channel B
0	VFRA	Valid Frame Received on Channel A Flag (vSS!ValidFrameA) 0: No valid frame received on channel A 1: Valid frame received on channel A

(1) FLXAnFRMBS.RESS

Reserved Bit Status Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(2) FLXAnFRMBS.PPIS

Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

0 = PPI indicator set to 0
The payload segment of the received frame does not contain a network management vector or a message ID

1 = PPI indicator set to 1
Static segment: Network management vector at the beginning of the payload
Dynamic segment: Message ID at the beginning of the payload

(3) FLXAnFRMBS.NFIS

Null Frame Indicator Status Flag (vRF!Header!NFIndicator)

If set to 0 the payload segment of the received frame contains no usable data.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(4) FLXAnFRMBS.SYNS

Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(5) FLXAnFRMBS.SFIS

Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator)

The startup frame indicator specifies a startup frame.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(6) FLXAnFRMBS.RCIS

Received on Channel Indicator Status Flag (vSS!Channel)

Indicates the channel on which the frame was received.

For receive buffers (bit FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(7) FLXAnFRMBS.CCS

Cycle Count Status Flag

Actual cycle count when status was updated.

(8) FLXAnFRMBS.FTB

Frame Transmitted on Channel B Flag

Indicates that this node has transmitted a data frame in the configured slot on channel B.

CAUTION

The FlexRay protocol specification requires that this bit can only be reset by the Host. Therefore, the Cycle Count Status FLXAnFRMBS.CCS bit for this bit is only valid for the cycle where the bit is set to 1.

(9) FLXAnFRMBS.FTA

Frame Transmitted on Channel A Flag

Indicates that this node has transmitted a data frame in the configured slot on channel A.

CAUTION

The FlexRay protocol specification requires that this bit can only be reset by the Host. Therefore the Cycle Count Status FLXAnFRMBS.CCS bit for this bit is only valid for the cycle where this bit is set to 1.

(10) FLXAnFRMBS.MLST

Message Lost Flag

The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame.

Not affected by reception of null frames except for message buffers belonging to the receive FIFO. Bits FLXAnFRNDATm.NDp is reset to 0 by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset to 0 by reading out the message buffer via OBF.

(11) FLXAnFRMBS.ESB

Empty Slot Channel B Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(12) FLXAnFRMBS.ESA

Empty Slot Channel A Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(13) FLXAnFRMBS.TCIB

Transmission Conflict Indication Channel B Flag (vSS!TxConflictB)

A transmission conflict indication is set to 1 if a transmission conflict has occurred on channel B.

(14) FLXAnFRMBS.TCIA

Transmission Conflict Indication Channel A Flag (vSS!TxConflictA)

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

(15) FLXAnFRMBS.SVOB

Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel B.

(16) FLXAnFRMBS.SVOA

Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on channel A.

(17) FLXAnFRMBS.CEOB

Content Error Observed on Channel B Flag (vSS!ContentErrorB)

A content error was observed in the assigned slot on channel B.

(18) FLXAnFRMBS.CEOA

Content Error Observed on Channel A Flag (vSS!ContentErrorA)

A content error was observed in the assigned slot on channel A.

(19) FLXAnFRMBS.SEOB

Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB)

A syntax error was observed in the assigned slot on channel B.

(20) FLXAnFRMBS.SEOA

Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA)

A syntax error was observed in the assigned slot on channel A.

(21) FLXAnFRMBS.VFRB

Valid Frame Received on Channel B Flag (vSS!ValidFrameB)

A valid frame indication is set if a valid frame was received on channel B.

(22) FLXAnFRMBS.VFRA

Valid Frame Received on Channel A Flag (vSS!ValidFrameA)

A valid frame indication is set if a valid frame was received on channel A.

24.3.11.6 FLXAnFROBCM — FlexRay Output Buffer Command Mask Register

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by bits FLXAnFROBCR.OBRS[6:0].

Mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to the register internal storage when a Message RAM transfer is requested by bit FLXAnFROBCR.REQ.

When OBF Host and OBF Shadow are swapped, mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 24.4.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag ND of the selected message buffer in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers is cleared.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0710_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSH	RHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSS	RHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.80 FLXAnFROBCM Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RDSH	Read Data Section Host Flag 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
16	RHSH	Read Header Section Host Flag 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RDSS	Read Data Section Shadow Bit 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
0	RHSS	Read Header Section Shadow Bit 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer

(1) FLXAnFROBCM.RDSH

Read Data Section Host Flag

(2) FLXAnFROBCM.RHSH

Read Header Section Host Flag

(3) FLXAnFROBCM.RDSS

Read Data Section Shadow Bit

(4) FLXAnFROBCM.RHSS

Read Header Section Shadow Bit

24.3.11.7 FLXAnFROBCR — FlexRay Output Buffer Command Request Register

After setting bit FLXAnFROBCR.REQ to 1 while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1, bits FLXAnFROBCR.OBRS[6:0] is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the message buffer selected by bits FLXAnFROBCR.OBRS[6:0] from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCM.OBSYS back to 0.

By setting bit FLXAnFROBCR.VIEW to 1 while FLXAnFROBCR.OBSYS is 0, OBF Host and OBF Shadow are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register FLXAnFROBCM internal storage to keep them attached to the respective Output Buffer transfer. bits FLXAnFROBCR.OBRH[6:0] signals the number of the message buffer currently accessible by the Host.

If bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are set to 1 with the same write access while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Any write access to bits FLXAnFROBCR[15:8] while FLXAnFROBCR.OBSYS is set to 1 will cause bit FLXAnFREIR.IOBA to be set to 1. n this case, this write access has no effect and the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 24.4.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0714_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	OBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OBSYS	—	—	—	—	—	REQ	VIEW	—	OBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.81 FLXAnFROBCR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	OBRH[6:0]	Output Buffer Request Host Flag
15	OBSYS	Output Buffer Busy Shadow Flag 0: No transfer in progress 1: Transfer between Message RAM and OBF Shadow in progress
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	REQ	Request Message RAM Transfer Bit 0: No request 1: Transfer to OBF Shadow requested
8	VIEW	View Shadow Buffer Bit 0: No action 1: Swap OBF Shadow and OBF Host
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	OBRS[6:0]	Output Buffer Request Shadow Bit

(1) FLXAnFROBCR.OBRH

Output Buffer Request Host Flag

Number of message buffer currently accessible by the Host via the FLXAnFRRDHS1 to FLXAnFRRDHS3, FLXAnFRMBS, and FLXAnFRRDDS1 to FLXAnFRRDDS64 registers.

By writing bit FLXAnFROBCR.VIEW to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host.

(2) FLXAnFROBCR.OBSYS

Output Buffer Busy Shadow Flag

Set to 1 after setting bit FLXAnFROBCR.REQ. When the transfer between the Message RAM and OBF Shadow has completed, FLXAnFROBCR.OBSYS is set back to 0.

(3) FLXAnFROBCR.REQ

Request Message RAM Transfer Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Requests transfer of message buffer addressed by bits FLXAnFROBCR.OBRS[6:0] from Message RAM to OBF Shadow.

(4) FLXAnFROBCR.VIEW

View Shadow Buffer Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Toggles between OBF Shadow and OBF Host.

(5) FLXAnFROBCR.OBRS

Output Buffer Request Shadow Bit

Only writeable while bit FLXAnFROBCR.OBSYS = 0.

Number of source message buffer to be transferred from the Message RAM to OBF Shadow.

If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see **Section 24.4.11, FIFO Function**) to OBF Shadow.

24.3.12 Data Transfer Control Register

24.3.12.1 FLXAnFRITC — FlexRay Input Transfer Configuration Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0800_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	ITM[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IQEIE	IQFIE	—	—	—	—	—	—	IQHR	ITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.82 FLXAnFRITC Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	ITM[6:0]	Input queue Table Max Bit These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIE	Input Queue Empty Interrupt Enable Bit 0: Disabled 1: Enabled
8	IQFIE	Input Queue Full Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IQHR	Input Queue Halt Request Bit 0: Input queue run request 1: Input queue halt request
0	ITE	Input Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFRITC.ITM

Input queue Table Max Bit

The user can only write to this bit when bit FLXAnFRITS.ITS is 0.

These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.

Valid values are 00_H (1 queue entry) to $7F_H$ (128 queue entries).

Note that each entry requires two long words in the input pointer table.

(2) FLXAnFRITC.IQEIE

Input Queue Empty Interrupt Enable Bit

This bit controls the input queue empty interrupt.

0: Disabled
No interrupt will be generated and the input queue empty interrupt line will be released.

1: Enabled
Input queue empty interrupt will be generated when bit FLXAnFRITS.IQEIS is 1.

(3) FLXAnFRITC.IQFIE

Input Queue Full Interrupt Enable Bit

This bit controls the input queue full interrupt.

0: Disabled
No interrupt will be generated and the input queue full interrupt line will be released.

1: Enabled
Input queue full interrupt will be generated when bit FLXAnFRITS.IQFIS is 1.

(4) FLXAnFRITC.IQHR

Input Queue Halt Request Bit

The IQHR bit should not be set to 1 when FLXAnFRITS.ITS is 0.

This bit requests a halt of the input queue.

The status of the halt request is shown in the FLXAnFRITS.IQH register.

Refer to **Section 24.4.16.1, (5) Halting the input queue** about usage of this bit.

0: Input queue run request
The input queue resumes their operation.

1: Input queue halt request
The input queue gets halted. An active input transfer will be completed but no further transfer request will start.

(5) FLXAnFRITC.ITE

Input Transfer Enable Bit

The user should only set this bit to 1 when bit FLXAnFRIBCR.IBSYS is 0.

The user should only set this bit to 0 when bit FLXAnFRITC.IQHR is 0. Otherwise committed input transfers get lost.

This bit controls the operation mode of the input transfer queue.

The operation status of the input transfer queue function is shown in bit FLXAnFRITS.ITS.

Refer to **Section 24.4.16.1 (1), Activation and deactivation** about usage of this bit.

0: Operation Disable request
The input transfer queue is disabled when it becomes empty.

1: Operation Enable request
The input transfer queue is enabled. Input data structures are transferred to the FlexRay internal message RAM.

24.3.12.2 FLXAnFROTC — FlexRay Output Transfer Configuration Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0804_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	FTM[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FWIE	OWIE	FIE	OIE	—	—	—	—	—	—	OTCS	OTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.83 FLXAnFROTC Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FTM[4:0]	FIFO Table Max Bit Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Cluster RAM.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIE	FIFO transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
10	OWIE	Output transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
9	FIE	FIFO transfer Interrupt Enable Bit 0: Disabled 1: Enabled
8	OIE	Output transfer Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OTCS	Output Transfer Condition Select Bit 0: New data only mode 1: New data and status changed mode
0	OTE	Output Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFROTC.FTM

FIFO Table Max Bit

The user can only write to these bits when bit FLXAnFROTS.OTS is 0.

Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Cluster RAM.

Valid values are 00_H (1 FIFO entry) to $1F_H$ (32 FIFO entries).

(2) FLXAnFROTC.FWIE

FIFO transfer Warning Interrupt Enable Bit

This bit controls the FIFO transfer warning interrupt.

0: Disabled
No interrupt will be generated and the FIFO transfer warning interrupt line will be released.

1: Enabled
FIFO transfer warning interrupt will be generated when bit FLXAnFROTS.FWIS is 1.

(3) FLXAnFROTC.OWIE

Output transfer Warning Interrupt Enable Bit

This bit controls the output transfer warning interrupt.

0: Disabled
No interrupt will be generated and the output transfer warning interrupt line will be released.

1: Enabled
Output transfer warning interrupt will be generated when bit FLXAnFROTS.OWIS is 1.

(4) FLXAnFROTC.FIE

FIFO transfer Interrupt Enable Bit

This bit controls the FIFO transfer interrupt.

0: Disabled
No interrupt will be generated and the FIFO transfer interrupt line will be released.

1: Enabled
FIFO transfer interrupt will be generated when bit FLXAnFROTS.FIS is 1.

(5) FLXAnFROTC.OIE

Output transfer Interrupt Enable Bit

This bit controls the output transfer interrupt.

0: Disabled
No interrupt will be generated and the output transfer interrupt line will be released.

1: Enabled
Output transfer interrupt will be generated when bit FLXAnFROTS.OTIS is 1.

(6) FLXAnFROTC.OTCS

Output Transfer Condition Select Bit

The user can only write to this bit when bit FLXAnFROTS.OTS is 0.

This bit controls the output transfer condition.

0: New data only mode
Bits FLXAnFRNDATm.NDp are used to detect a transfer condition for dedicated receive buffer.

1: New data and status changed mode
Bits FLXAnFRNDATm.NDp and FLXAnFRMBSCm. are used to detect a transfer condition for dedicated transmit and receive buffer.

(7) FLXAnFROTC.OTE

Output Transfer Enable Bit

The user should only set this bit to 1 when bit FLXAnFROBCR.OBSYS is 0.

This bit controls the operation mode of the output transfer function.

The operation status of the output buffer transfer function is shown in FLXAnFROTS.OTS.

Refer to **Section 24.4.16.2, (1) Activation and deactivation** about usage of this bit.

0: Operation Disable request
The output buffer transfer gets disabled.
An active message buffer transfer will be completed but no further transfer will start.

1: Operation Enable request
The output buffer transfer gets enabled. Message buffers are transferred from the FlexRay internal message RAM to output data structures.

The user should not change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

24.3.12.3 FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0808_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ITA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 24.84 FLXAnFRIBA Register Contents

Bit Position	Bit Name	Function
31 to 0	ITA[31:0]	Input Table Base Address Bit These bits configure the base address of the input pointer table.

(1) FLXAnFRIBA.ITA

Input Table Base Address Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRIBA.ITA[1:0] are always 0.

These bits configure the base address of the input pointer table.

The table is used for the input transfer queue transferring message buffers from the Local RAM/Cluster RAM into the FlexRay internal message RAM.

The size of the input queue is configured in bits FLXAnFRITC.ITM[6:0].

Note that each entry requires two long words in the input pointer table.

24.3.12.4 FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $080C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 24.85 FLXAnFRFBA Register Contents

Bit Position	Bit Name	Function
31 to 0	FTA[31:0]	FIFO pointer Table Base Address Bit These bits configure the base address of the FIFO pointer table.

(1) FLXAnFRFBA.FTA

FIFO pointer Table Base Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRFBA.FTA[1:0] are always 0.

These bits configure the base address of the FIFO pointer table.

The table is used for message buffers transferred from the FlexRay internal FIFO to the Local RAM/ Cluster RAM.

The size of the FIFO is configured in bits FLXAnFROTC.FTM[4:0].

24.3.12.5 FLXAnFROBA — FlexRay Output Pointer Table Base Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0810_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 24.86 FLXAnFROBA Register Contents

Bit Position	Bit Name	Function
31 to 0	OTA[31:0]	Output pointer Table Base Address Bit These bits configure the base address of the output pointer table.

(1) FLXAnFROBA.OTA

Output pointer Table Base Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFROBA.OTA[1:0] are always 0.

These bits configure the base address of the output pointer table.

The table is used for message buffers transferred from the FlexRay internal message RAM to the Local RAM/Cluster RAM.

The size of the table depends on the utilization of the FlexRay internal message RAM and can have up to 128 entries.

24.3.12.6 FLXAnFRIQC — FlexRay Input Queue Control Register

Access: This register is a write-only register that can be written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0814_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W

Table 24.87 FLXAnFRIQC Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When writing, write the value after reset.
6 to 0	IMBNR[6:0]	Input Message Buffer NumbeR Bit Message buffer number added to the input queue

(1) FLXAnFRIQC.IMBNR

Input Message Buffer NumbeR Bit

The user can only write to this bit when FLXAnFRITS.IQFP is 0.

The user should not write to this register when FLXAnFRITS.ITS is 0 or when FLXAnFRITC.ITE is 0.

These bits are read as 0.

This value specifies the message buffer added to the input queue.

The number has to be identical to FLXAnFRWRHS4.IMBNR[6:0] (see **Section 24.4.16.1, (3) Input pointer table**) of the input pointer table.

The address to the input data structure has to be provided in the input pointer table at the put index (bit FLXAnFRITS.IPIDX[6:0]) before writing to this register.

Writing to this register increments the input put index (bit FLXAnFRITS.IPIDX[6:0]).

24.3.12.7 FLXAnFRUIR — FlexRay User Input Transfer Request Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0818_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.88 FLXAnFRUIR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	UIDX[7:0]	User requested Input inDeX Bit Input pointer table index requested for input transfer

(1) FLXAnFRUIR.UIDX

User requested Input inDeX Bit

The user can only write to this bit when bit FLXAnFRITS.UIRP is 0.

The user should not write to this register when bit FLXAnFRITS.ITS is 0.

The user should not write to this register when bit FLXAnFRITS.UIRP is 1.

The user should not write to this register when bit FLXAnFRITS.IQH is 1.

The user should only write bits FLXAnFRITC.ITM[6:0] +1 to this register.

This value specifies the input pointer table index for the requested input transfer.

The address to the input data structure has to be provided in the input pointer table at the index FLXAnFRUIR.UIDX[7:0] before writing to this register.

When writing to this register, the requested input data structure will be transferred from input data structure position to the FlexRay internal message RAM.

In opposite to queued input transfers the related FLXAnFRDA.DA flag is not influenced by the user input transfer.

24.3.12.8 FLXAnFRUOR — FlexRay User Output Transfer Request Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $081C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	URDS	—	—	UMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.89 FLXAnFRUOR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	URDS	User request Read Data Section Bit 0: Data section is not transferred 1: Data section is transferred
8, 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	UMBNR[6:0]	User requested output Message Buffer NumbeR Bit Message buffer number requested for output transfer

(1) FLXAnFRUOR.URDS

User request Read Data Section Bit

The user can only write to this bit when bit FLXAnFROTS.UORP is 0.

The user should not write to this register when bit FLXAnFROTS.OTS is 0.

The user should not write to this register when bit FLXAnFROTS.UORP is 1.

0: Data section is not transferred

The data section of the message buffer selected by the bits FLXAnFRUOR.UMBNR[6:0] is not requested

1: Data section is transferred

The data section of the message buffer selected by the bits FLXAnFRUOR.UMBNR[6:0] is requested

(2) FLXAnFRUOR.UMBNR

User requested output Message Buffer NumbeR Bit

The user can only write to this bit when bit FLXAnFROTS.UORP is 0.

The user should not write to this register when bit FLXAnFROTS.OTS is 0.

The user should not write to this register when bit FLXAnFROTS.UORP is 1.

The user should restrict this bit to dedicated receive and transmit buffers when the FlexRay module is not in the CONFIG state.

When writing to this register, the contents of the message buffer will be transferred to a given area. The destination will be the area specified by the value in the output pointer table.

The data to be transferred can be specified by setting the URDS bit whether to transfer the header section only or to include the data section.

24.3.13 Data Transfer Status Register

24.3.13.1 FLXAnFRITS — FlexRay Input Transfer Status Register

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0820_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IGIDX[6:0]							—	IPIDX[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IQFP	—	—	IQEIS	IQFIS	—	—	—	—	—	UIRP	IQH	ITS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.90 FLXAnFRITS Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	IGIDX[6:0]	Input queue Get InDeX Bit Represents the get index of the input pointer table
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IPIDX[6:0]	Input queue Put InDeX Bit Represents the put index of the input pointer table
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	IQFP	Input Queue Full condition Pending Bit 0: Entries in the input queue are available 1: All entries in the input queue are occupied
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIS	Input Queue Empty Interrupt Status Bit 0: No input queue empty condition detected 1: Input queue empty condition detected
8	IQFIS	Input Queue Full Interrupt Status Bit 0: No input queue full condition detected 1: Input queue full condition detected
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	UIRP	User Input transfer Request Pending Bit 0: No user input transfer request pending 1: User input transfer request pending
1	IQH	Input Queue Halted Bit 0: Input queue not halted 1: Input queue halted
0	ITS	Input Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFRITS.IGIDX

Input queue Get InDeX Bit

These bits are only valid when bit FLXAnFRITS.IQH is 1

These bits represent the input pointer index the input queue handler will transfer next.

Valid values are 00_H to FLXAnFRITC.ITM.

The get index is incremented when the input data structure has been transferred from the Local RAM/ Cluster RAM and the related FLXAnFRDA.DA flag is cleared.

The index is set to 00_H when bit FLXAnFRITS.ITS changes from 0 to 1.

(2) FLXAnFRITS.IPIDX

Input queue Put InDeX Bit

These bits represent the index where the next input data structure pointer in the input pointer table should be stored.

Valid values are 00_H to FLXAnFRITC.ITM.

After reaching the maximum value the put index continues from 00_H.

The index is incremented when writing to bits FLXAnFRIQC.IMBNR[6:0].

The index is set to 00_H when bit FLXAnFRITS.ITS changes from 0 to 1.

(3) FLXAnFRITS.IQFP

Input Queue Full condition Pending Bit

This bit represents that the input queue is full.

There should be no further input transfer requests, by writing to bits FLXAnFRIQC.IMBNR[6:0], as long as this bit is 1.

[Clearing condition]

- This bit is cleared when there is one free entry in the input queue.

[Setting condition]

- This bit is set when all entries in the input queue are occupied.

(4) FLXAnFRITS.IQEIS

Input Queue Empty Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFRITC.IQEIE, the input queue empty interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRITS.IQEIS.
- This bit is cleared when bit FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

- This bit is set when all pending input transfers have been processed and consequently the input queue becomes empty.

(5) FLXAnFRITS.IQFIS

Input Queue Full Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFRITC.IQFIE the input queue full interrupt is generated when FLXAnFRITS.IQFIS is 1.

This flag is intended as interrupt status flag. It does not represent the current input queue status; for this status refer to bit FLXAnFRITS.IQFP.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRITS.IQFIS.
- This bit is cleared when bit FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

- This bit is set when all entries in the input queue are occupied.

(6) FLXAnFRITS.UIRP

User Input transfer Request Pending Bit

This bit represents that a user input transfer is still pending.

There should be no further write access to bits FLXAnFRUIR.UIDX[7:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when the user input transfer request is processed by the input transfer handler.

[Setting condition]

- This bit is set when writing to bits FLXAnFRUIR.UIDX[7:0].

(7) FLXAnFRITS.IQH

Input Queue Halted Bit

This bit represents the status of the input queue.

There should be no further write access to bits FLXAnFRUIR.UIDX[7:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when bit FLXAnFRITC.IQHR is set to 0.

[Setting condition]

- This bit is set immediately when bit FLXAnFRITC.IQHR is set to 1 and there is no ongoing input transfer.
- This bit is set only after an ongoing input transfer has been completed and bit FLXAnFRITC.IQHR is set to 1.

(8) FLXAnFRITS.ITS

Input Transfer Status Bit

This bit represents the status of the input queue handler.

While this bit is 1, there should be no read or write access to the address area <FLXAn_base> + 0400_H to <FLXAn_base> + $05FF_H$ and there should be no CLEAR_RAMS command applied to bits FLXAnFRSUCC1.CMD[3:0].

The input transfer queue indexes and related status flags are set to 0 when bit FLXAnFRITS.ITS changes from 0 to 1.

[Clearing condition]

- This bit is cleared immediately when bit FLXAnFRITC.ITE is set to 0 and there are no pending input transfers
- This bit is cleared after all pending requests have been processed and bit FLXAnFRITC.ITE is 0.

[Setting condition]

- This bit is set when bit FLXAnFRITC.ITE is set to 1.

24.3.13.2 FLXAnFROTS — FlexRay Output Transfer Status Register

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0824_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	FFL[5:0]						—	—	—	FGIDX[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWP	OWP	FDA	—	FWIS	OWIS	FIS	OTIS	—	—	—	—	—	UORP	—	OTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.91 FLXAnFROTS Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	FFL[5:0]	FIFO Fill Level Bit Represent the number of unprocessed output FIFO structures
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FGIDX[4:0]	FIFO Get InDeX Bit Represent the get index in the FIFO pointer table
15	FWP	FIFO transfer Warning condition Pending Bit 0: No FIFO transfer warning condition pending 1: FIFO transfer warning condition pending
14	OWP	Output transfer Warning condition Pending 0: No output transfer warning condition pending 1: Output transfer warning condition pending
13	FDA	FIFO Data Available Bit 0: No available FIFO structures 1: FIFO structures available at current FLXAnFROTS.FGIDX index
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIS	FIFO transfer Warning Interrupt Status Bit 0: No FIFO transfer warning condition detected 1: FIFO transfer warning condition detected
10	OWIS	Output transfer Warning Interrupt Status Bit 0: No output transfer warning condition detected 1: Output transfer warning condition detected
9	FIS	FIFO transfer Interrupt Status Bit 0: No FIFO structure updated in Local RAM/Cluster RAM 1: FIFO structure updated in Local RAM/Cluster RAM
8	OTIS	Output transfer Interrupt Status Bit 0: No output structure updated in Local RAM/Cluster RAM 1: Output structure updated in Local RAM/Cluster RAM
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 24.91 FLXAnFROTS Register Contents (2/2)

Bit Position	Bit Name	Function
2	UORP	User Output transfer Request Pending Bit 0: No user output transfer request pending 1: User output transfer request pending
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OTS	Output Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFROTS.FFL

FIFO Fill Level Bit

These bits represent the number of available output FIFO structures in the Local RAM/Cluster RAM.

Valid values are 00_H to FLXAnFROTC.FTM+1.

The value 00_H represents that the FIFO is empty.

The value FLXAnFROTC.FTM+1 represents that the FIFO is full and no further FIFO transfers will be done.

The FIFO fill level is incremented when a FIFO data structure has been transferred from the FlexRay internal FIFO into the Local RAM/Cluster RAM.

The FIFO fill level is decremented when the user releases a FIFO data structure in the Local RAM/ Cluster RAM by writing 1 to bit FLXAnFROTS.FDA.

The FIFO fill level is set to 00_H when bit FLXAnFROTS.OTS changes from 0 to 1.

(2) FLXAnFROTS.FGIDX

FIFO Get InDeX Bit

These bits represent the index where the current output data structure pointer in the FIFO pointer table is available for reading.

Valid values are 00_H to FLXAnFROTC.FTM.

After reaching the maximum value the get index continues from 00_H.

The index is incremented when a FIFO data structure is released by writing 1 to bit FLXAnFROTS.FDA.

The index is set to 00_H when bit FLXAnFROTS.OTS changes from 0 to 1.

(3) FLXAnFROTS.FWP

FIFO transfer Warning condition Pending Bit

This bit represents the FIFO transfer warning condition.

[Clearing condition]

- This bit is cleared when there are free output data structures (FLXAnFROTS.FFL ≤ FLXAnFROTC.FTM).

This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1).

(4) FLXAnFROTS.OWP

Output transfer Warning condition Pending Bit

This bit represents the output transfer warning condition.

[Clearing condition]

- This bit is cleared, when all output structure pointers that have a pending output handler transfer condition detected, are released (for dedicated transmit and receive message buffers or a user output transfer request).

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).
- This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(5) FLXAnFROTS.FDA

FIFO Data Available Bit

Writing 0 has no effect on the bit value.

When this bit is 1, the next valid output data structure is available.

The related data structure pointer is in the FIFO pointer table at FLXAnFROTS.FGIDX.

Writing 1 to FLXAnFROTS.FDA

- increments FLXAnFROTS.FGIDX[4:0] and
- decrements the FIFO fill level (FLXAnFROTS.FFL)

If there are still unprocessed data structures FLXAnFROTS.FDA remains 1.

[Clearing condition]

- This bit is cleared when writing 1 to FLXAnFROTS.FDA and the FIFO fill level becomes 00_H.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when there is at least one FIFO data structure available in the Local RAM/Cluster RAM.

(6) FLXAnFROTS.FWIS

FIFO transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.FWIE, the FIFO transfer warning interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFROTS.FWIS.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1).

(7) FLXAnFROTS.OWIS

Output transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.OWIE, the FIFO transfer warning interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to this bit.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).
- This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(8) FLXAnFROTS.FIS

FIFO transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.FIE, the FIFO transfer interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to this bit.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when a FIFO data structure is updated by the transfer handler or bits FLXAnFROTS.FFL[5:0] changes from 00_H to 01_H.

(9) FLXAnFROTS.OTIS

Output transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in bit FLXAnFROTC.OIE, the output transfer interrupt is generated when this bit is 1.

[Clearing condition]

- This bit is cleared when writing a 1 to bit FLXAnFROTS.OTIS.
- This bit is cleared when bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when an output data structure is updated by the transfer handler (from a dedicated transmit or receive message buffer or by a user output transfer request).

(10) FLXAnFROTS.UORP

User Output transfer Request Pending Bit

This bit represents that a user output transfer is still pending.

There should be no further write access to bits FLXAnFRUOR.UMBNR[6:0] when this bit is 1.

[Clearing condition]

- This bit is cleared when the user output transfer request is processed by the output transfer handler.
- This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

- This bit is set when writing to bit FLXAnFRUOR.UMBNR.

(11) FLXAnFROTS.OTS

Output Transfer Status Bit

This bit represents the status of the output transfer handler.

While this bit is 1, there should be no read or write access to the address area <FLXAn_base> + 0600_H to <FLXAn_base> + $07FF_H$ and there should be no CLEAR_RAMS command applied to bits FLXAnFRSUCC1.CMD[3:0].

While this bit is 1, the user should not change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

The output handler transfer indexes and related status flags are set to 0 when this bit changes from 0 to 1.

[Clearing condition]

- This bit is cleared immediately when bit FLXAnFROTC.OTE is set to 0 and there are no ongoing output transfers.
- This bit is cleared after an ongoing transfer has been completed and bit FLXAnFROTC.OTE is 0.

[Setting condition]

- This bit is set when bit FLXAnFROTC.OTE is set to 1.

24.3.13.3 FLXAnFRAES — FlexRay Access Error Status Register

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0828_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MAE	FAE	OAE	IAE	EIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.92 FLXAnFRAES Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MAE	Multiple Access Errors Bit 0: No multiple access errors occurred 1: Multiple access errors occurred
10	FAE	FIFO transfer Access Error Bit 0: No access error occurred during FIFO transfer 1: Access error occurred during FIFO transfer
9	OAE	Output transfer Access Error Bit 0: No access error occurred during output transfer 1: Access error occurred during output transfer
8	IAE	Input transfer Access Error Bit 0: No access error occurred during input transfer 1: Access error occurred during input transfer
7 to 0	EIDX[7:0]	Error InDeX Bit Data structure pointer index number

(1) FLXAnFRAES.MAE

Multiple Access Errors Bit

Writing 0 has no effect on the bit value.

This bit represents that there were multiple access errors during a data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.MAE.

[Setting condition]

- This bit is set when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is set and one of the conditions below is met:
  - an access to an protected address occurred during a FIFO data transfer or
  - an access to an protected address occurred during an output data transfer or

– an access to an protected address occurred during an input data transfer

(2) FLXAnFRAES.FAE

FIFO transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during a FIFO data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.FAE.

[Setting condition]

- This bit is set when a Local RAM/Cluster RAM access error was detected during a FIFO transfer and bits FLXAnFRAES.OAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(3) FLXAnFRAES.OAE

Output transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during a output data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.OAE.

[Setting condition]

- This bit is set when a Local RAM/Cluster RAM access error was detected during an output transfer and bits FLXAnFRAES.FAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(4) FLXAnFRAES.IAE

Input transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit represent that there was an access error during an input data transfer.

[Clearing condition]

- This bit is cleared when writing a 1 to FLXAnFRAES.IAE.

[Setting condition]

- This bit is set when a Local RAM/Cluster RAM access error was detected during an input transfer and bits FLXAnFRAES.OAE, FLXAnFRAES.FAE and FLXAnFRAES.MAE are 0.

(5) FLXAnFRAES.EIDX

Error InDeX Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is 1.

When bit FLXAnFRAES.FAE is 1, FLXAnFRAES.EIDX holds the used FIFO put index when the access error has occurred.

When the bit FLXAnFRAES.OAE is '1', FLXAnFRAES.EIDX holds the used output table entry (which is related to message buffer number) when the access error has occurred.

When the bit FLXAnFRAES.IAE is '1', FLXAnFRAES.EIDX holds the used input pointer table get index when the access error has occurred during the input transfer or user requested input transfer.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is changing from 0 to 1.

24.3.13.4 FLXAnFRAEA — FlexRay Access Error Address Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + $082C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AEA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.93 FLXAnFRAEA Register Contents

Bit Position	Bit Name	Function
31 to 0	AEA[31:0]	Access Error Address Bit Address in the Local RAM/Cluster RAM when an access error has occurred

(1) FLXAnFRAEA.AEA

Access Error Address Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is 1.

These bits represent the address of the access error indicated in the FLXAnFRAES register.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE and FLXAnFRAES.IAE is changing from 0 to 1.

24.3.13.5 FLXAnFRDAm — FlexRay message Data Available Register m (m = 0 to 3)

Do not rewrite this register using bit manipulation instructions.

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <FLXAn_base> + 0830_H to <FLXAn_base> + $083C_H$ (<FLXAn_base> + 0830_H + m × 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAp[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAp[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.94 FLXAnFRDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	DAp[31:0]	Data Available Bit p 0: No data available for destination 1: Data available for destination

(1) FLXAnFRDAm.DAp (p = m × 32 to ((m + 1) × 32) – 1)

Data Available Bit p

The user should not write a 1 to bits that are 0.

To maintain the status of input transfers, the user should not clear bits related to input transfers.

This register is used for input and output transfers.

Each flag corresponds to a FlexRay message buffer.

[Clearing condition]

- Input transfer:
  This bit is cleared when the input data structure has been transferred from the Local RAM/Cluster RAM. The data structure and the data structure pointer can be changed when the related flag is 0.
- Output transfer:
  This bit is cleared when writing a 1 to it.

[Setting condition]

- Input transfer:
  This bit is set when the corresponding message buffer number has been written to bits FLXAnFRIQC.IMBNR[6:0].
  As long as this bit is 1, the input data structure and the data structure pointer corresponding to this input transfer request should not be changed.
- Output transfer:
  This bit is set when the output data structure corresponding to this message buffer has been

updated.
As long as this bit is 1, the data structure is stable; no further update of the data structure will be done by the output handler. While this bit is 1, the application is allowed to change the output data structure pointer in the output pointer table for this message buffer number.

24.4 Operation

This chapter describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification.

24.4.1 FlexRay Module Operation Control

24.4.1.1 FlexRay Module Enable

After a reset or after the FlexRay module has been disabled (following **Section 24.4.1.2, FlexRay Module Disable**) the FlexRay module is in the reset state (bit FLXAnFROS.OS is 0) and the clocks of the FlexRay core module are disabled.

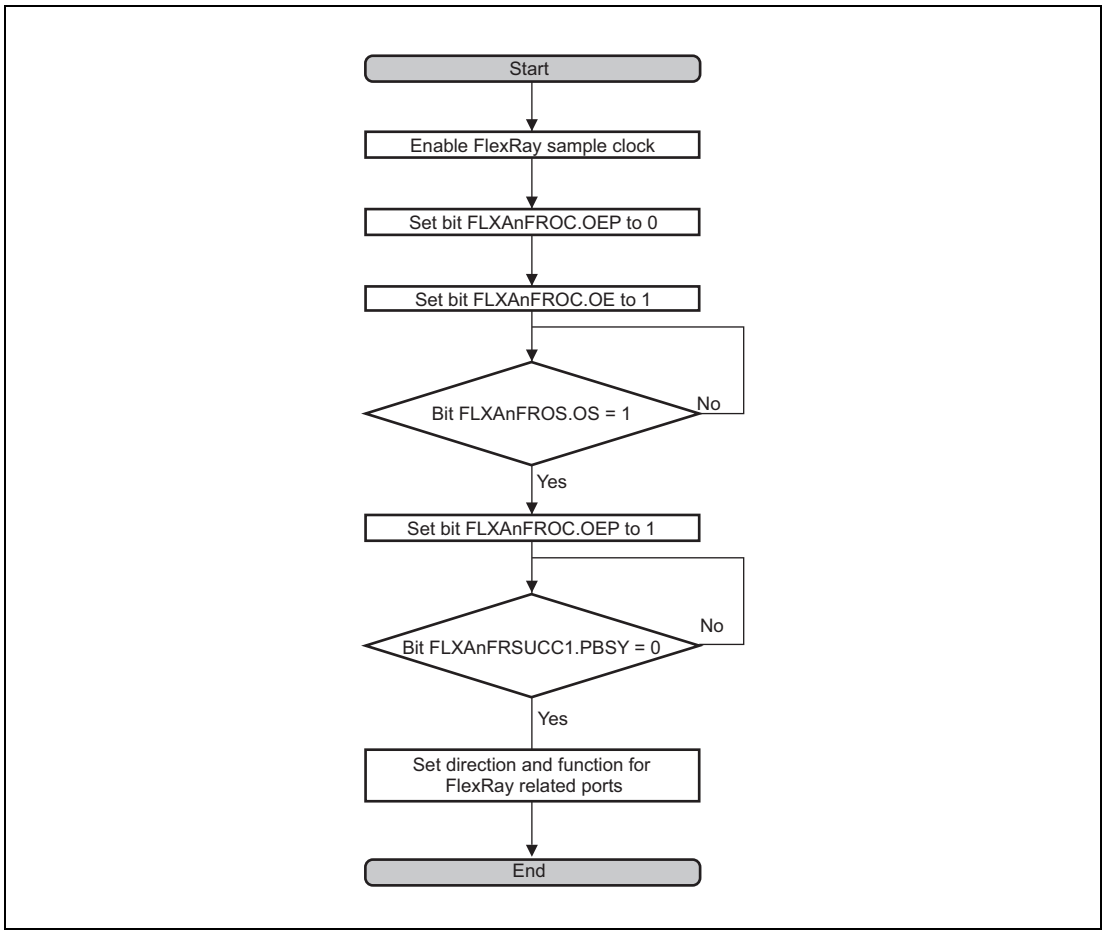


Figure 24.3 FlexRay Enable Processing Flow

24.4.1.2 FlexRay Module Disable

The FlexRay module can be disabled at any time. However, it is recommended to disable the FlexRay module using bit FLXAnFROC.OE only when the FlexRay module is in HALT, CONFIG or DEFAULT_CONFIG state. Resetting the FlexRay module in any other state will terminate any ongoing FlexRay communication.

If the data transfer function is used, it is also required to disable this function before disabling the FlexRay module (see **Section 24.4.16.1, (1) Activation and deactivation** for suspending input transfer function and **Section 24.4.16.2, (1) Activation and deactivation** for suspending output transfer).

The following flow should be executed to disable the FlexRay module.

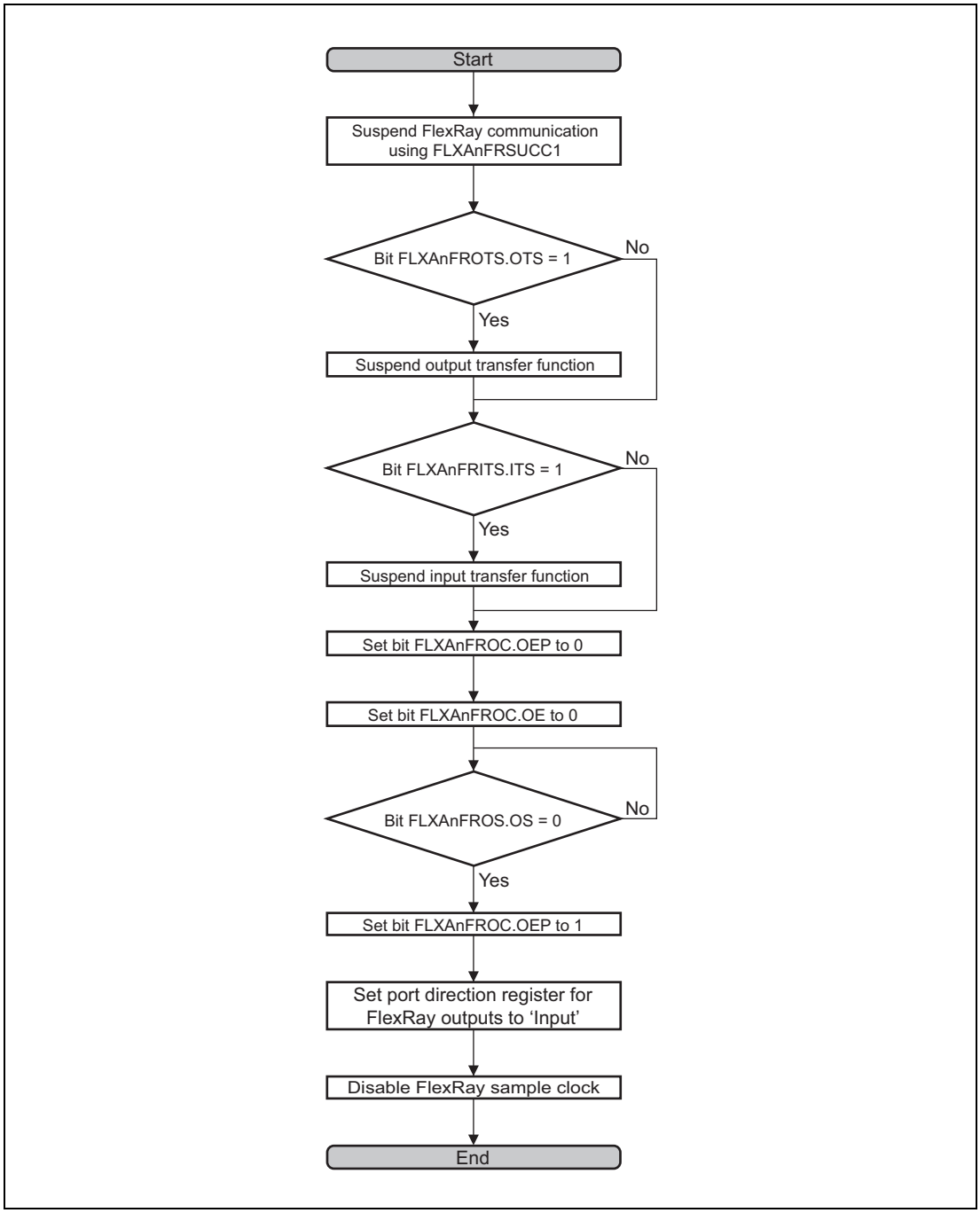


Figure 24.4 FlexRay Disable Processing Flow

24.4.2 Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

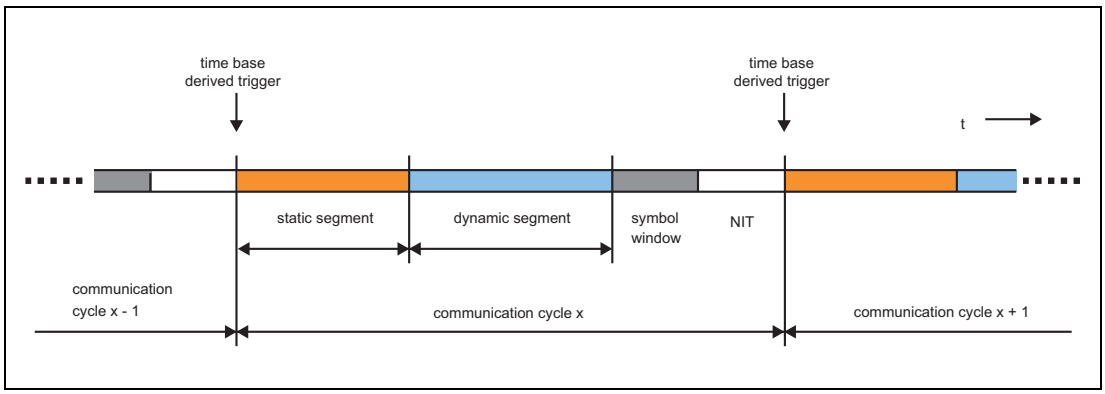


Figure 24.5 Structure of Communication Cycle

24.4.2.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters:

Number of Static Slots (FLXAnFRGTUC7.NSS[9:0])

Static Slot Length (FLXAnFRGTUC7.SSL[9:0])

Payload Length Static (FLXAnFRMHDC.SFDL[6:0])

Action Point Offset (FLXAnFRGTUC9.APO[5:0])

24.4.2.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters:

Number of Minislots (FLXAnFRGTUC8.NMS[12:0])

Minislot Length (FLXAnFRGTUC8.MSL[5:0])

Minislot Action Point Offset (FLXAnFRGTUC9.MAPO[4:0])

Start of Latest Transmit (last minislot) (FLXAnFRMHDC.SLT[12:0])

24.4.2.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are send in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters:

Symbol Window Action Point Offset (FLXAnFRGTUC9.APO[4:0]) (same as for static slots)

Network Idle Time Start (FLXAnFRGTUC4.NIT[13:0])

24.4.2.4 Network Idle Time (NIT)

During network idle time the CC has to perform the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters:

Network Idle Time Start (FLXAnFRGTUC4.NIT[13:0])

Offset Correction Start (FLXAnFRGTUC4.OCS[13:0])

24.4.2.5 Configuration of NIT Start and Offset Correction Start

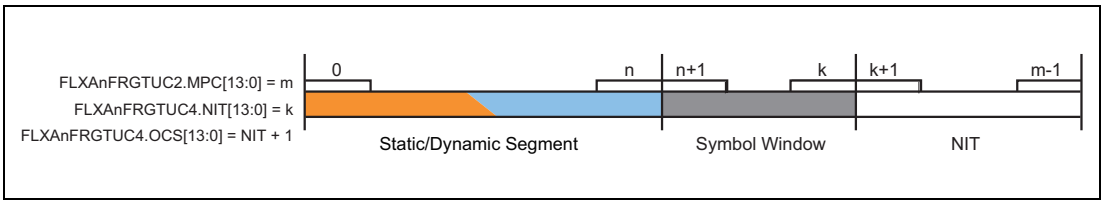


Figure 24.6 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming FLXAnFRGTUC2.MPC[13:0] = m.

The static / dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset + dynamic segment length – 1MT

= gNumberOfStaticSlots × gdStaticSlot + dynamic segment offset + gNumberOfMinislots × gdMinislot – 1 MT

The static segment length is configured by FLXAnFRGTUC7.SSL[9:0] and FLXAnFRGTUC7.NSS[9:0].

The dynamic segment length is configured by FLXAnFRGTUC8.MSL[5:0] and FLXAnFRGTUC8.NMS[12:0].

The dynamic segment offset is:

If gdActionPointOffset ≤ gdMinislotActionPointOffset:

dynamic segment offset = 0 MT

Else if gdActionPointOffset > gdMinislotActionPointOffset:

dynamic segment offset = gdActionPointOffset -gdMinislotActionPointOffset

The NIT starts with macrotick k+1 and ends with the last macrotick of cycle m-1. It has to be configured by setting FLXAnFRGTUC4.NIT[13:0] = k.

For the FlexRay module the offset correction start is required to be FLXAnFRGTUC4.OCS[13:0] ≥ FLXAnFRGTUC4.NIT[13:0] + 1 = k + 1.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by the number of macroticks (k – n).

24.4.3 Communication Modes

The FlexRay Protocol Specification defines the Time-Triggered Distributed (TT-D) mode.

24.4.3.1 Time-triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- Pure static: Minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

24.4.4 Clock Synchronization

In TT-D mode a distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

24.4.4.1 Global Time

Activities in a FlexRay node, individual nodes independently operate. Operations including communication, are based on the concept of a global time. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values; the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster specific:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (µT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

24.4.4.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (µT).

Node specific:

- Oscillator clock → prescaler → microtick (µT)
- µT = basic unit of time measurement in a CC, clock correction is done in units of µTs
- Cycle counter + macrotick counter = nodes local view of the global time

24.4.4.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXAnFRGTUC2.SNM[3:0]) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see *FlexRay protocol specification v2.1, chapter 8*.

(1) Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of µTs
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened / shortened)

(2) Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of µTs
- Distributed over macroticks comprising the next even / odd cycle pair (MTs lengthened / shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit FLXAnFRMRC.SPLM has to be programmed to 1.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames, bit FLXAnFRSUCC1.TXSY must be set to 1.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is not checked against configured limits

24.4.5 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FLXAnFREIR.PEMC to 1 and may trigger an interrupt to the Host if enabled. The actual error mode is signaled by FLXAnFRCCEV.ERRM[1:0].

Table 24.95 Error Modes of the POC (Degradation Model)

Error Mode	Activity
ACTIVE	Full operation, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
PASSIVE	Reduced operation, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
COMM_HALT	Operation halted, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. The bus drivers are disabled.

24.4.5.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the “maximum without clock correction passive” limit defined by FLXAnFRSUCC3.WCP[3:0], the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the “maximum without clock correction fatal” limit defined by FLXAnFRSUCC3.WCF[3:0], it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter (FLXAnFRCCEV.CCFC[3:0]) allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FLXAnFRSFS.MOCS or the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FLXAnFRSFS.MOCS nor the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value FLXAnFRSUCC3.WCF is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

CAUTION

The transition to HALT state is prevented if the FLXAnFRSUCC1.HCSE bit is not set to 1.

24.4.5.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FLXAnFRSUCC1.PTA[4:0] defines the number of consecutive valid even/odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FLXAnFRSUCC1.PTA[4:0] is set to 0 the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

24.4.5.3 HALT Command

In case the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing 0110 to bits FLXAnFRSUCC1.CMD[3:0]. In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to assure that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from flags FLXAnFRCCSV.PSL[5:0].

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state, bits FLXAnFRSUCC1.CMD[3:0] will be reset to “0000” = command_not_accepted and bit FLXAnFREIR.CNA is set to 1. If enabled an interrupt to the Host is generated.

24.4.5.4 FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing 0111 to bits FLXAnFRSUCC1.CMD[3:0]. The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from flags FLXAnFRCCSV.PSL[5:0].

CAUTION

When transfer is stopped by a FREEZE or READY command and then restarted as a leading ColdStart node, the startup frame may not be transmitted in cycle 0. This depends on the internal state of the FlexRay module. Such cases arise when the startup frame is set in a slot 1 to slot 7. This does not occur in a ColdStart after the reset of the microcontroller. Even if the above situation arises, the ColdStart will succeed on the second trial. Repetition prolongs the overall ColdStart process, but the ColdStart will not be obstructed by the earlier situation. To avoid this effect, allocate startup and sync frames to static slot 8 or a slot with a higher number.

24.4.6 Communication Controller States

24.4.6.1 Communication Controller State Diagram

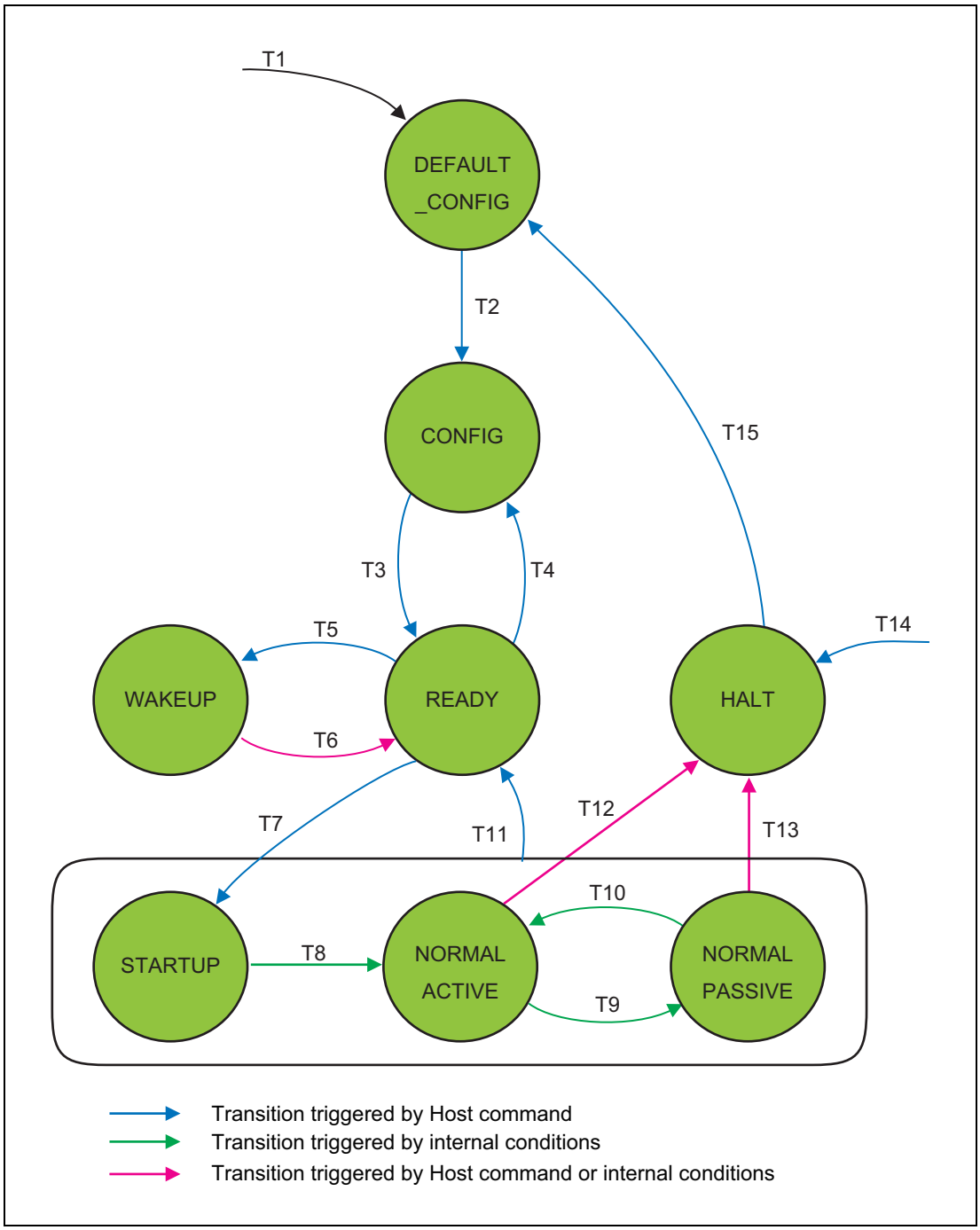


Figure 24.7 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by reset, rxda_extfxr, rxdb_extfxr, by the POC state machine, and by the CHI Command Vector FLXAnFRSUCC1.CMD[3:0].

The CC transits from all states to HALT state after application of the FREEZE command (FLXAnFRSUCC1.CMD[3:0] = “0111”).

Table 24.96 State Transitions of FlexRay Overall State Machine

T#	Condition	From	To
1	Reset	All States	DEFAULT_CONFIG
2	Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = “0001”	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command READY: FLXAnFRSUCC1.CMD[3:0] = “0010”	CONFIG	READY
4	Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = “0001”	READY	CONFIG
5	Command WAKEUP: FLXAnFRSUCC1.CMD[3:0] = “0011”	READY	WAKEUP
6	Complete transmission of wakeup pattern, received WUP or received frame header, wakeup collision or command READY: FLXAnFRSUCC1.CMD[3:0] = “0010”	WAKEUP	READY
7	Command RUN: FLXAnFRSUCC1.CMD[3:0] = “0100”	READY	STARTUP
8	Successful STARTUP	STARTUP	NORMAL_ACTIVE
9	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by FLXAnFRSUCC3.WCP[3:0]	NORMAL_ACTIVE	NORMAL_PASSIVE
10	Number of valid correction terms reached the Passive to Active limit configured by FLXAnFRSUCC1.PTA[4:0]	NORMAL_PASSIVE	NORMAL_ACTIVE
11	Command READY: FLXAnFRSUCC1.CMD[3:0] = “0010”	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
12	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF[3:0] when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD[3:0] = “0110”	NORMAL_ACTIVE	HALT
13	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF[3:0] when bit FLXAnFRSUCC1.HCSE set to 1, or command HALT: FLXAnFRSUCC1.CMD[3:0] = “0110”	NORMAL_PASSIVE	HALT
14	Command FREEZE: FLXAnFRSUCC1.CMD[3:0] = “0111”	All States	HALT
15	Command CONFIG: FLXAnFRSUCC1.CMD[3:0] = “0001”	HALT	DEFAULT_CONFIG

24.4.6.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When the reset is applied (the reset of the microcontroller or the software reset of the FlexRay module)
- When exiting from HALT state

To leave DEFAULT_CONFIG state, the Host has to write 0001 to bits FLXAnFRSUCC1.CMD[3:0]. The CC then transits to CONFIG state.

24.4.6.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyze status information and configuration. Before leaving CONFIG state, the Host has to assure that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described is **Section 24.3.3.1, FLXAnFRLCK — FlexRay Lock Register**. Directly after unlocking the CONFIG state the Host has to write bits FLXAnFRSUCC1.CMD[3:0] to enter the next state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1 to FLXAnFRTXRQ4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (bus clock and sample clock). To do this the Host has to assure that all Message RAM transfers have finished before turning off the clocks.

24.4.6.4 READY State

After unlocking CONFIG state and writing 0010_B to FLXAnFRSUCC1.CMD[3:0], the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing 0010_B to FLXAnFRSUCC1.CMD[3:0] (READY command).

The CC exits from this state

- To CONFIG state by writing 0001_B to FLXAnFRSUCC1.CMD[3:0] (CONFIG command)
- To WAKEUP state by writing 0011_B to FLXAnFRSUCC1.CMD[3:0] (WAKEUP command)
- To STARTUP state by writing 0100_B to FLXAnFRSUCC1.CMD[3:0] (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1 to FLXAnFRTXRQ4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

24.4.6.5 WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

- When exiting from READY state by writing 0011_B to bits FLXAnFRSUCC1.CMD[3:0] (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern (WUP)
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing the FLXAnFRSUCC1.WUCS bit. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag FLXAnFRSIR.WST. The wakeup status vector can be read from FLXAnFRCCSV.WSV[2:0]. If a valid wakeup pattern was received also either flag FLXAnFRSIR.WUPA or flag FLXAnFRSIR.WUPB is set to 1.

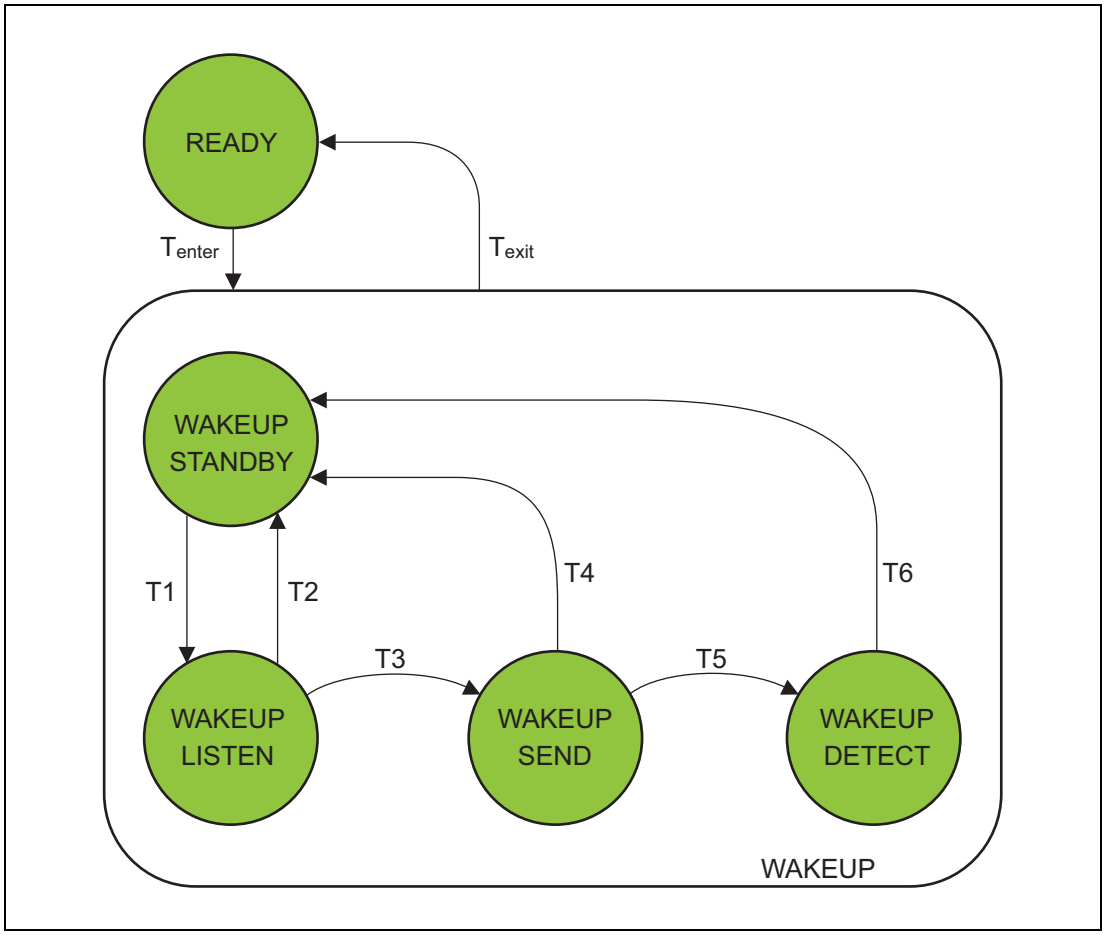


Figure 24.8 Structure of POC State WAKEUP

Table 24.97 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing 0011 to FLXAnFRSUCC1.CMD[3:0] (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired or WUP detected on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header received on either available	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) or Host commands change to READY state by writing 0010 to FLXAnFRSUCC1.CMD[3:0] (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the parameters Listen Timeout (bits FLXAnFRSUCC2.LT[20:0]) and Listen Timeout Noise (bits FLXAnFRSUCC2.LTN[3:0]). Listen Timeout enables a fast cluster wakeup in case of a noise free environment, while Listen Timeout Noise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by bits FLXAnFRSUCC2.LT[20:0]. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification recommends that two different CCs shall awake the two channels.

(1) Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signaled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the CC in CONFIG state
  - Select wakeup channel by programming bit FLXAnFRSUCC1.WUCS
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing 0011 to bits FLXAnFRSUCC1.CMD[3:0]
  - CC enters WAKEUP
  - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
  - In a dual channel cluster wait for WUP on the other channel
  - Reset coldstart inhibit flag FLXAnFRCCSV.CSI by writing 1001_B to bits FLXAnFRSUCC1.CMD[3:0] (ALLOW_COLDSTART command)
- Command CC to enter startup by writing 0100_B to bits FLXAnFRSUCC1.CMD[3:0] (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing 0100 to bits FLXAnFRSUCC1.CMD[3:0] (RUN command)

(2) Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FLXAnFRPRTC1 and FLXAnFRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by bits FLXAnFRPRTC2.TXL[5:0]
- Wakeup symbol idle time used to listen for activity on the bus, configured by bits FLXAnFRPRTC2.TXI[7:0]
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by bits FLXAnFRPRTC1.RWP[5:0] (2 to 63 repetitions)
- Wakeup symbol receive window length configured by bits FLXAnFRPRTC1.RXW[8:0]
- Wakeup symbol receive low time configured by bits FLXAnFRPRTC2.RXL[5:0]
- Wakeup symbol receive idle time configured by bits FLXAnFRPRTC2.RXI[5:0]

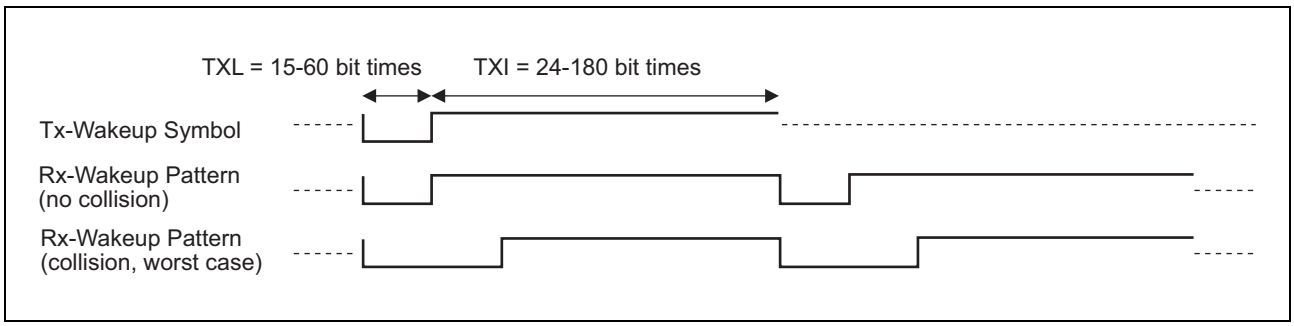


Figure 24.9 Timing of Wakeup Pattern

24.4.6.6 STARTUP State

The description below is intended to help configuring startup for the FlexRay module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the *FlexRay protocol specification v2.1*, *section 7.2*.

Any node entering STARTUP state that has coldstart capability should assure that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is specified for initial synchronization of all nodes. In general, a node may enter NORMAL_ACTIVE state via (see **Figure 24.10**):

- Coldstart path initiating the schedule synchronization (LeadingColdstart node)
- Coldstart path joining other coldstart nodes (FollowingColdstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY set to 1. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set to 1.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by bits FLXAnFRSUCC1.CSA[4:0].

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

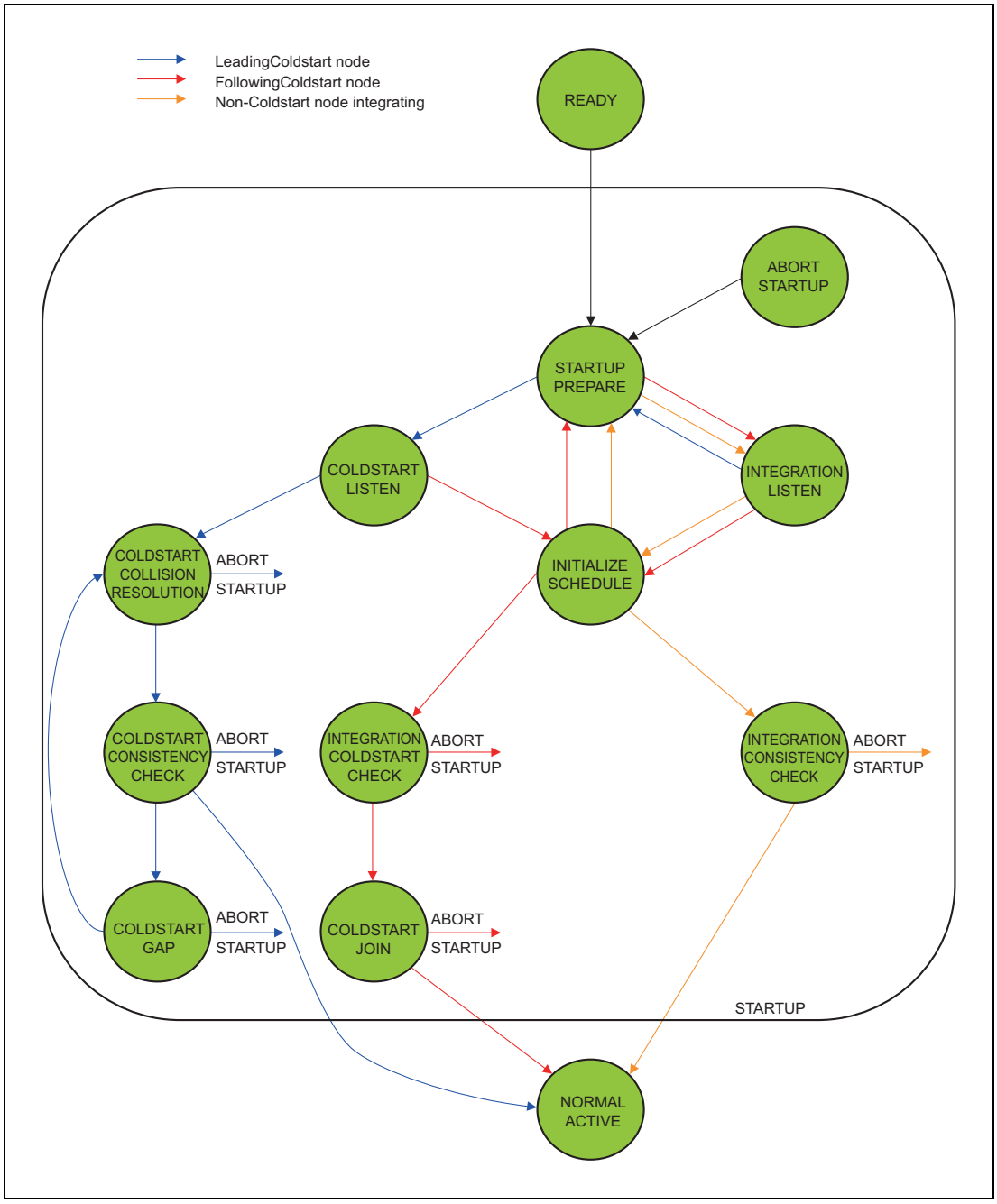


Figure 24.10 State Diagram Time-Triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit FLXAnFRCCSV.CSI is set to 1, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXAnFRCCSV.CSI is set to 1 whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (bits FLXAnFRSUCC1.CMD[3:0] = "1001_B")

(2) Startup Timeouts

The CC supplies two different µT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

CAUTION

The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXAnFRSUCC2.LT[20:0] and FLXAnFRSUCC2.LTN[3:0].

(a) Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming bits FLXAnFRSUCC2.LT[20:0] (see **Section 24.3.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup timeout is:

pdListenTimeout = FLXAnFRSUCC2.LT[20:0]

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

(b) Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming bits FLXAnFRSUCC2.LTN[3:0] (see **Section 24.3.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup noise timeout:

pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT[20:0] × (FLXAnFRSUCC2.LTN[3:0] + 1)

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FLXAnFRSUCC1.CSA[4:0]. The number of remaining coldstarts attempts can be read from FLXAnFRCCSV.RCA[4:0]. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is assured that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still available. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signaled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

24.4.6.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even / odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command, at the end of the current cycle)
- HALT state by writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

24.4.6.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command, at the end of the current cycle)
- HALT state by writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE.
  The transition takes place when bits FLXAnFRCCEV.PTAC[4:0] equals bits FLXAnFRSUCC1.PTA[4:0] - 1
- To READY state by writing 0010_B to bits FLXAnFRSUCC1.CMD[3:0] (READY command)

24.4.6.9 HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the “maximum without clock correction fatal” limit and FLXAnFRSUCC1.HCSE is set to 1

The CC exits from this state to DEFAULT_CONFIG state

- By writing 0001_B to bits FLXAnFRSUCC1.CMD[3:0] (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analyzing purposes.

When the Host writes 0110_B to bits FLXAnFRSUCC1.CMD[3:0] (HALT command), the CC sets flag FLXAnFRCCSV.HRQ to 1 and enters HALT state at the next end of cycle.

When the Host writes 0111_B to bits FLXAnFRSUCC1.CMD[3:0] (FREEZE command), the CC enters HALT state immediately and sets flag FLXAnFRCCSV.FSI to 1.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL[5:0].

24.4.7 Network Management

The accrued Network Management (NM) vector can be read from registers FLXAnFRNMV1 to FLXAnFRNMV3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by bits FLXAnFRNEMC.NML[3:0]. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set as 1, bit PPIT in the header section of the respective transmit buffer has to be set to 1 via bit FLXAnFRWRHS1.PPIT. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

CAUTIONS

1. **In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by the FLXAnFRNEMC.NML[3:0] bits.**
2. **When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FLXAnFRNMV1 to FLXAnFRNMV3 holds the value from the cycle before.**

24.4.8 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance / transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

CAUTION

For the FIFO the acceptance filter is configured by the FIFO Rejection Filter (FLXAnFRFRF) and the FIFO Rejection Filter Mask (FLXAnFRFRFM).

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

24.4.8.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID and channel ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

24.4.8.2 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 resp. 1 is configured to hold the startup / sync frame or the single slot frame by bits FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, cycle counter filtering for message buffer 0 resp. 1 shall be disabled.

CAUTION

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in **Table 24.98**.

Table 24.98 Definition of cycle set

Cycle Code	Matching Cycle Counter Values
$000000x_B$	all Cycles
$000001c_B$	every second Cycle at (Cycle Count) mod2 = c
$00001cc_B$	every fourth Cycle at (Cycle Count) mod4 = cc
$0001ccc_B$	every eighth Cycle at (Cycle Count) mod8 = ccc
$001cccc_B$	every sixteenth Cycle at (Cycle Count) mod16 = cccc
$01ccccc_B$	every thirty-second Cycle at (Cycle Count) mod32 = ccccc
$1cccccc_B$	every sixty-fourth Cycle at (Cycle Count) mod64 = cccccc

Table 24.99 below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 24.99 Examples for valid cycle sets

Cycle Code	Matching Cycle Counter Values
0000011_B	1-3-5-7-…. -63
0000100_B	0-4-8-12-…. -60
0001110_B	6-14-22-30-…. -62
0011000_B	8-24-40-56
0100011_B	3-35
1001001_B	9

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Channel ID and frame ID must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Channel ID and frame ID must also be met.

24.4.8.3 Channel ID Filtering

There is a 2-bit channel filtering field (CH) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see **Table 24.100**).

Table 24.100 Channel filtering configuration

CH[1:0]	**Transmit Buffer** transmit frame	**Receive Buffer store valid receive frame**
00	no transmission	ignore frame
01	on channel A	received on channel A
10	on channel B	received on channel B
11	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CH = "11_B").

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CH = "11").

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted resp. received frames are ignored (same function as CH = "00").

24.4.8.4 FIFO Filtering

For FIFO filtering registers FLXAnFRFRF and FLXAnFRFRFM are used. The FIFO filter consists of channel filter bits FLXAnFRFRF.CH[1:0], frame ID filter bits FLXAnFRFRF.FID[10:0], and cycle counter filter bits FLXAnFRFRF.CYF[6:0]. Registers FLXAnFRFRF and FLXAnFRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

24.4.9 Transmit Process

24.4.9.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

24.4.9.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by bits FLXAnFRMHDC.SLT[12.0] defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

24.4.9.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit CFG in the header section of the respective message buffer to 1 via FLXAnFRWRHS1.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B
- Dynamic segment: channel A or channel B

Message buffer 0 resp. 1 is dedicated to hold the startup frame, the sync frame, or the designated single slot frame as configured by FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of bits FLXAnFRMRC.SEC[1.0] (see **Section 24.4.12.1, Reconfiguration of Message Buffers**). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the PPIT bit in the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer (see **Section 24.4.7, Network Management**).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by

FLXAnFRMHDC.SFDL[6.0], the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is "0000_H".

CAUTION

In case of an odd payload length (PLC = 1, 3, 5, so on) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is 0000_H.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode the CC resets the respective TXR flag to 0 after transmission has completed. Now the Host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR to 0 after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset to 0 by the Host by writing the respective message buffer number to the FLXAnFRIBCR register while bit FLXAnFRIBCM.STXRH is set to 0.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

24.4.9.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via registers FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write the data section of the transmit buffer via register FLXAnFRWRDSm
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR
- If configured in register FLXAnFRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = 0) in the FLXAnFRTRXQ1 to FLXAnFRTRXQ4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FLXAnFRTXRQ1 to FLXAnFRTXRQ4 registers is reset to 0 (single-shot mode), and, if bit MBI in the header section of the message buffer is set to 1, bit FLXAnFRSIR.TXI is set to 1. If enabled, an interrupt is generated.

24.4.9.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag to 1 before transmit time, the CC transmits a null frame with the null frame indication bit set to 0 and the payload data set to zero.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (TXR = 0) to 1.
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status FLXAnFRMBS is updated.

Null frames are not transmitted in the dynamic segment.

24.4.10 Receive Process

24.4.10.1 Dedicated Receive Buffers

A portion of the FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to 0 via FLXAnFRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of bits FLXAnFRMRC.SEC[1.0] (see **Section 24.4.12.1, Reconfiguration of Message Buffers**). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

24.4.10.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via registers FLXAnFRWRHS1 to FLXAnFRWRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FLXAnFRNDAT1 to FLXAnFRNDAT4 registers is set to 1, and, if bit MBI in the header section of that message buffer is set to 1, bit FLXAnFRSIR.RXI is set to 1. If enabled, an interrupt is generated.

In case that bit ND was already set to 1 when the Message Handler updates the message buffer, bit FLXAnFRMBS.MLST of the respective message buffer is set to 1 and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status FLXAnFRMBS is updated.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1 to FLXAnFRMBSC4 registers is set to 1, and if bit MBI in the header section of that message buffer is set, bit FLXAnFRSIR.MBSI is set to 1. If enabled an interrupt is generated.

If the payload length of a received frame PLR is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in **Section 24.4.12.2, (2) Data Transfer from Message RAM to Output Buffer**.

CAUTION

The ND and MBC flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

24.4.10.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status FLXAnFRMBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1 to FLXAnFRMBSC4 register is set to 1, and if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.MBSI is set to 1. If enabled, an interrupt is generated.

24.4.11 FIFO Function

24.4.11.1 Description

A portion of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXAnFRMRC.FFB[7.0] and ending with the message buffer referenced by FLXAnFRMRC.LCB[7.0]. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status FLXAnFRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. When Flag FLXAnFRSIR.RFNE is set to 1, it shows that the FIFO is not empty. When bit FLXAnFRSIR.RFCL is set to 1, it shows that the receive FIFO fill level (FLXAnFRFSR.RFFL[7.0]) is equal or greater than the critical level as configured by FLXAnFRFCL.CL. When bit FLXAnFREIR.RFO is set to 1, it shows that a FIFO overrun has been detected. If interrupts are enabled, interrupt requests are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXAnFREIR.RFO to 1.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag FLXAnFRSIR.RFNE is set to 1. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in **Figure 24.11** for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FLXAnFRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXAnFRFRF.RSS is set to 1, all messages received in the static segment are rejected by the FIFO. If bit FLXAnFRFRF.RNF is set to 1, received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FLXAnFRFRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

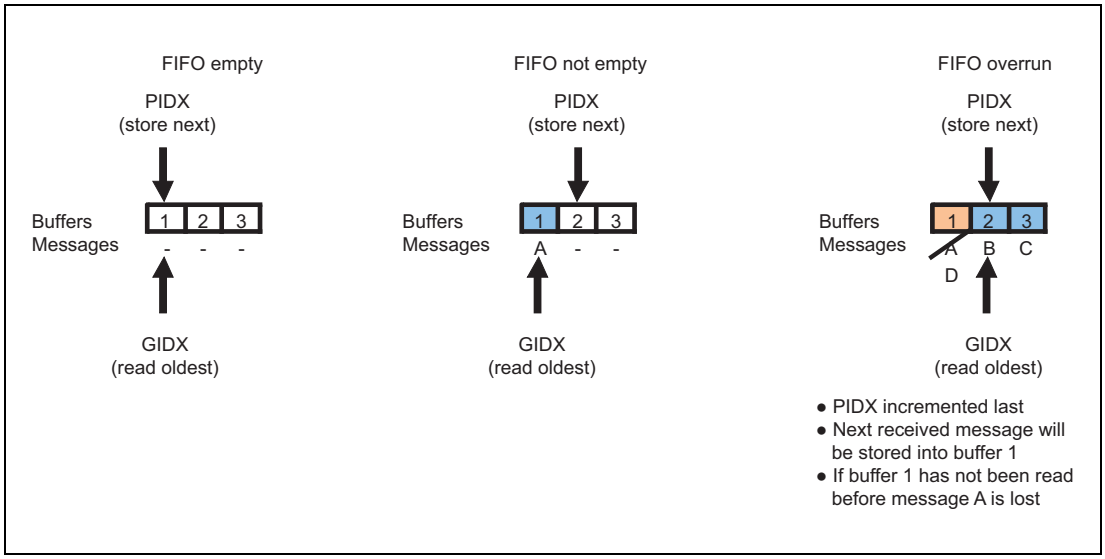


Figure 24.11 FIFO Status: Empty, Not Empty, Overrun

24.4.11.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via bits FLXAnFRWRHS2.PLC[6.0]. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via bits FLXAnFRWRHS3.DP[10.0].

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

CAUTIONS

1. **It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via FLXAnFRWRHS1.MBI to avoid generation of RX interrupts.**
2. **If the payload length of a received frame is longer than the value programmed by FLXAnFRWRHS2.PLC in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.**

24.4.11.3 Access to the FIFO

(1) When the output buffer is used:

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FLXAnFRMRC.FFB[7.0]) to the register FLXAnFROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

(2) When the data transfer function is used:

The message received in FIFO can be transferred to the Local RAM/Cluster RAM by using the output data transfer function. For the output data transfer function, see **Section 24.4.16.2, Output Data Transfer**, Output Data Transfer.

24.4.12 Message Handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two TBFs.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to bits FLXAnFRGTUC7.NSS[9:0]. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from bits FLXAnFRGTUC7.NSS[9:0] + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

Access of the Host to the message buffer contents using the input or output buffer function is described in this subsection. Access to the message buffer contents using the data transfer function is mentioned in **Section 24.4.16, Usage of Data Transfer**.

24.4.12.1 Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers FLXAnFRWRHS1 to FLXAnFRWRHS3.

Reconfiguration has to be enabled via control bit FLXAnFRMRC.SEC in the Message RAM Configuration register.

If a message buffer has not been transmitted / updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission / reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted / updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to **Table 24.101** below.

Table 24.101 Scan of Message RAM

Start of Scan in Slot	Scan for Slots
1	2 to 15, 1 (next cycle)
8	16 to 23, 1 (next cycle)
16	24 to 31, 1 (next cycle)
24	32 to 39, 1 (next cycle)
....	....

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by bits FLXAnFRMRC.FDB[7:0]. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by bit FLXAnFRMRC.FDB.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the “Static Buffers”, it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the “Static + Dynamic Buffers”, it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

CAUTION

Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted / updated from a received frame.

24.4.12.2 Host access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to FLXAnFRIBCR or FLXAnFROBCR register.

The FLXAnFRIBCM and FLXAnFROBCM registers can be used to write / read header and data section of the selected message buffer separately.

If bit FLXAnFRIBCM.STXR is set to 1, the transmission request flag TXR of the selected message buffer is automatically set to 1 after the message buffer has been updated. If bit FLXAnFRIBCM.STXR is reset to 0, the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are build up as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host), while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

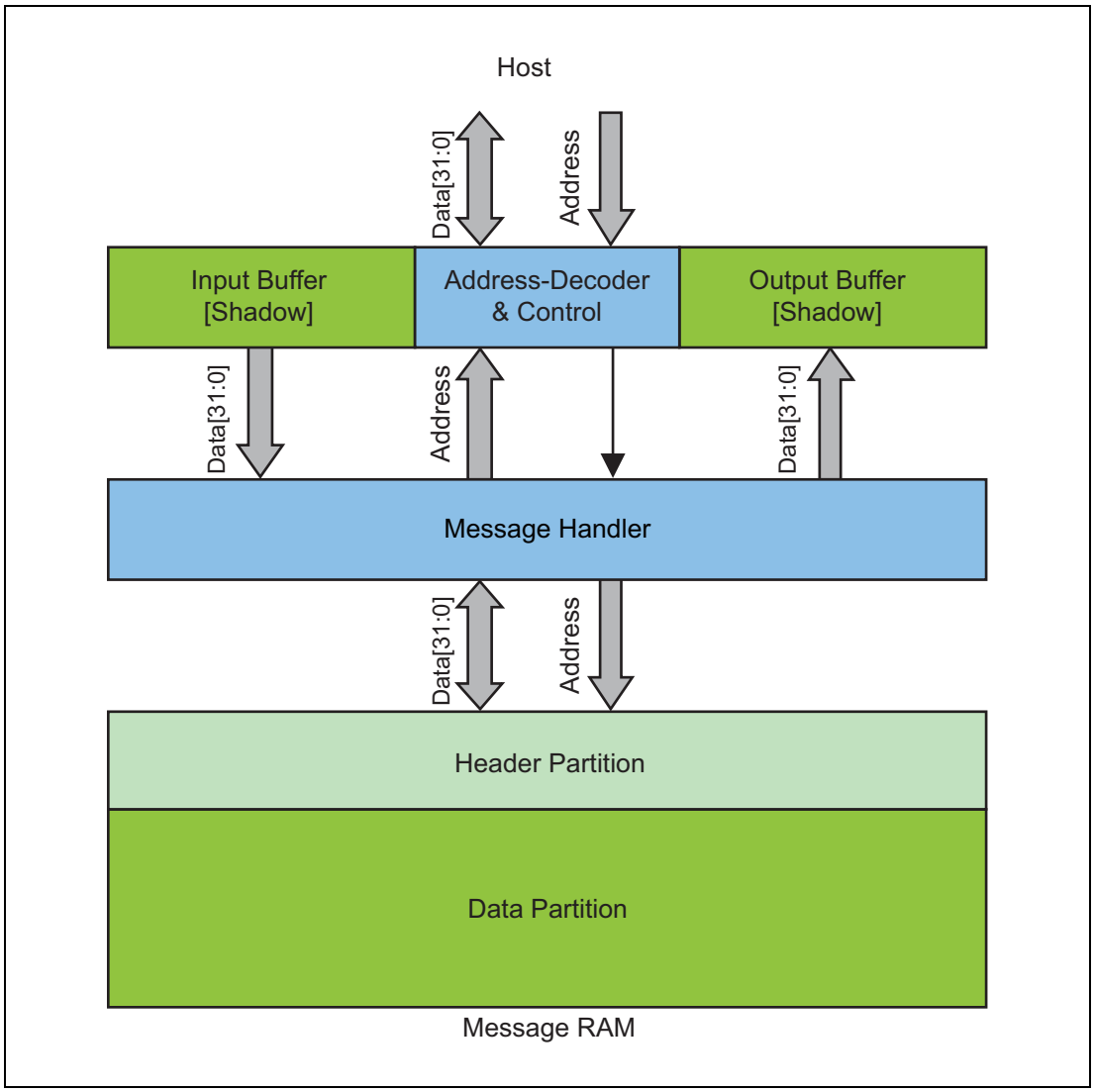


Figure 24.12 Host access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to FLXAnFRWRDSm and the header to FLXAnFRWRHS1 to FLXAnFRWRHS3. The specific action is selected by configuring the FlexRay Input Buffer Command Mask FLXAnFRIBCM.

When the Host writes the number of the target message buffer in the Message RAM to bits FLXAnFRIBCR.IBRH[6:0], IBF Host and IBF Shadow are swapped (see **Figure 24.13**).

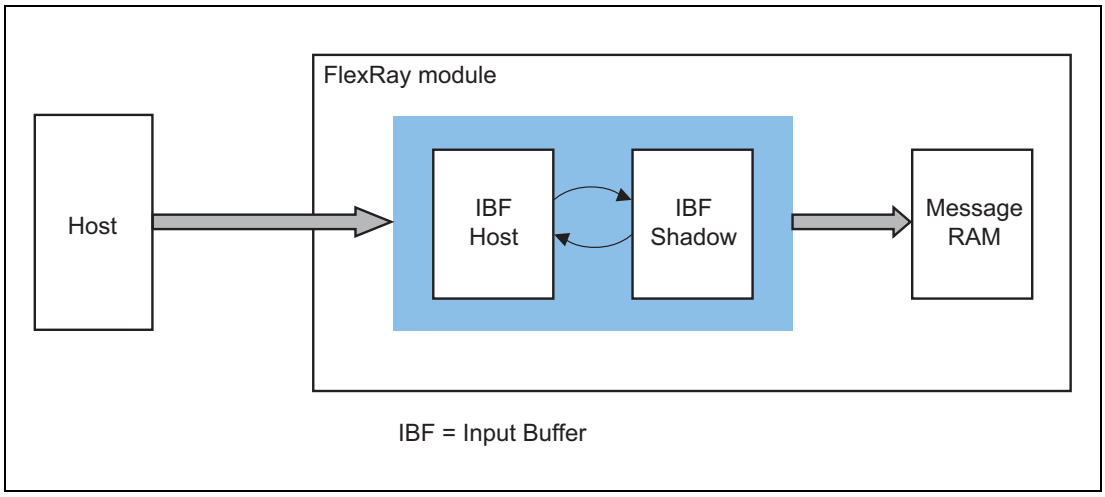


Figure 24.13 Double Buffer Structure Input Buffer

In addition the bits in the FLXAnFRIBCM and FLXAnFRIBCR registers are also swapped to keep them attached to the respective IBF section (see **Figure 24.14**).

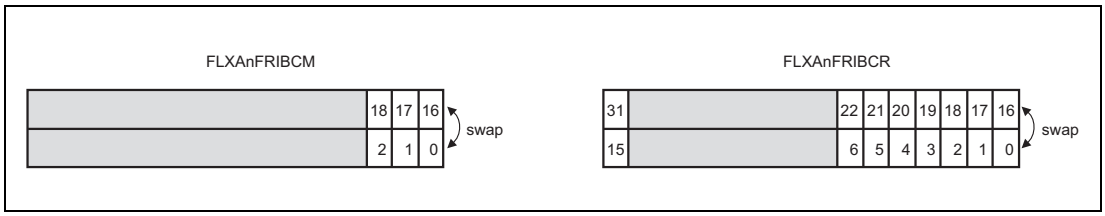


Figure 24.14 Swapping of FLXAnFRIBCM and FLXAnFRIBCR bits

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by bits FLXAnFRIBCR.IBRS[6:0].

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to bits FLXAnFRIBCR.IBRH[6:0].

If a write access to bits FLXAnFRIBCR.IBRH[6:0] occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0, FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under bits FLXAnFRIBCR.IBRH[6:0] and FLXAnFRIBCR.IBRS[6:0] and the command mask flags are also swapped.

Example of a 8/16/32-bit Host access sequence:

Configure / update k-th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSm
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH[6:0]

Configure / update (k+1)th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSm
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: set bits FLXAnFRIBCR.IBRH[6:0]

CAUTION

Any write access to IBF while bit FLXAnFRIBCR.IBSYH is 1 will set bit FLXAnFREIR.IIBA to 1. In this case the write access has no effect.

Table 24.102 Assignment of FLXAnFRIBCM Bits

Pos.	Access	Bit	Function
18	R	STXRS	Set Transmission Request Shadow ongoing or finished
17	R	LDSS	Load Data Section Shadow ongoing or finished
16	R	LHSS	Load Header Section Shadow ongoing or finished
2	R/W	STXRH	Set Transmission Request Host
1	R/W	LDSH	Load Data Section Host
0	R/W	LHSH	Load Header Section Host

Table 24.103 Assignment of FLXAnFRIBCR Bits

Pos.	Access	Bit	Function
31	R	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22 to 16	R	IBRS[6:0]	IBF Request Shadow, number of message buffer currently / lately updated
15	R	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH[6:0]
6 to 0	R/W	IBRH[6:0]	IBF Request Host, number of message buffer to be updated next

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FLXAnFROBCR to trigger the data transfer as configured in FLXAnFROBCM. After the transfer has completed, the Host can read the transferred data from FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS.

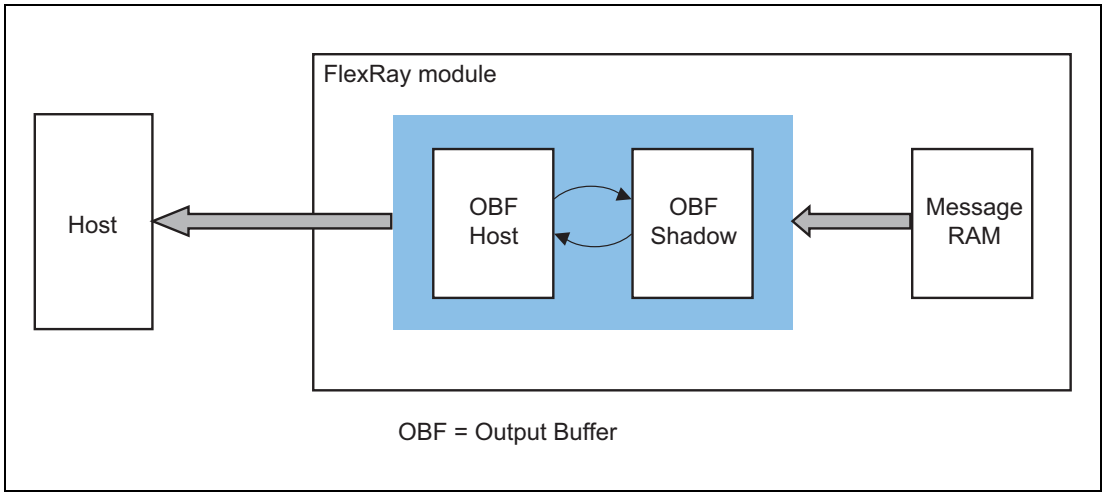


Figure 24.15 Double buffer structure Output Buffer

OBF Host and OBF Shadow as well as bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS, FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH and bits FLXAnFROBCR.OBRS[6:0], FLXAnFROBCR.OBRH[6:0] are swapped under control of bits FLXAnFROBCR.VIEW and FLXAnFROBCR.REQ.

Writing bit FLXAnFROBCR.REQ to 1 copies bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS and bits FLXAnFROBCR.OBRS[6:0] to an internal storage (see **Figure 24.16**).

After setting bit FLXAnFROBCR.REQ to 1, FLXAnFROBCR.OBSYS is set to 1, and the transfer of the message buffer selected by bits FLXAnFROBCR.OBRS[6:0] from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, bit FLXAnFROBCR.OBSYS is set back to 0. Bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW can only be set to 1 while bit FLXAnFROBCR.OBSYS is 0.

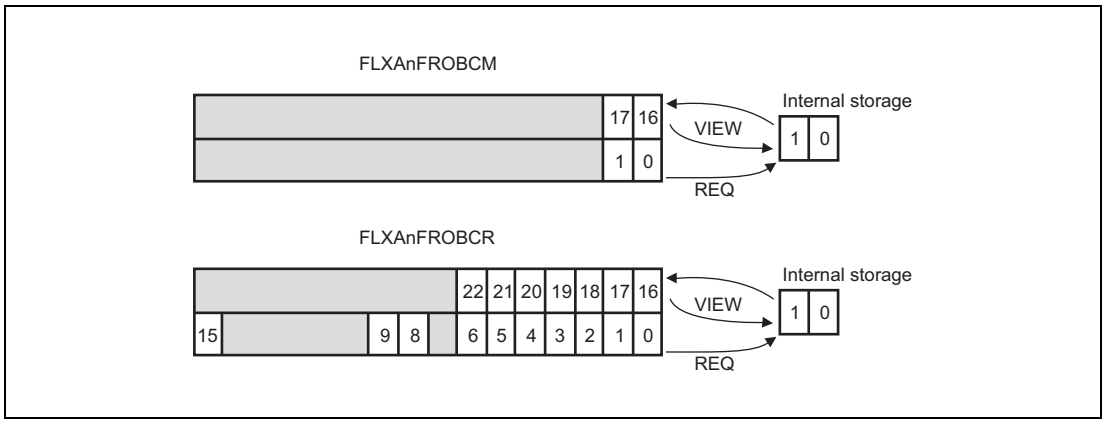


Figure 24.16 Swapping of FLXAnFROBCM and FLXAnFROBCR bits

OBF Host and OBF Shadow are switched by setting bit FLXAnFROBCR.VIEW to 1 while bit FLXAnFROBCR.OBSYS is 0 (see **Figure 24.15**).

In addition, bits FLXAnFROBCR.OBRH[6:0] and bits FLXAnFROBCM.RHSH and FLXAnFROBCM.RDSH are switched with the registers internal storage thus assuring that the message buffer number stored in bits FLXAnFROBCR.OBRH[6:0] and the mask configuration stored in FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH match the transferred data stored in OBF Host (see **Figure 24.16**).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits REQ and VIEW are set to 1 with the same write access while FLXAnFROBSYS is 0, FLXAnFROBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards bits FLXAnFROBCR.OBRS[6:0] are copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Example of an 8/16/32-bit Host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are necessary:

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS
- Request transfer of message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS[6:0] and FLXAnFROBCR.REQ (in case of and 8-bit Host interface, FLXAnFROBCR.OBRS[6:0] have to be written before FLXAnFROBCR.REQ).
- Wait until FLXAnFROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing FLXAnFROBCR.VIEW = 1
- Read out transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Example of an 8/16/32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing bits FLXAnFROBCR.OBRS[6:0] and FLXAnFROBCR.REQ (in case of accessing to the FLXAnFROBCR register in 8-bit units, bits FLXAnFROBCR.OBRS[6:0] have to be written before FLXAnFROBCR.REQ).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing FLXAnFROBCR.OBRS[6:0] of 2nd message buffer, bits FLXAnFROBCR.REQ, and FLXAnFROBCR.VIEW (in case of and 8-bit access to register FLXAnFROBCR, bits FLXAnFROBCR.OBRS[6:0] have to be written before bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW).
- Read out 1st transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Demand access to last requested message buffer without request of another message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Demand access to last transferred message buffer by writing FLXAnFROBCR.VIEW
- Read out last transferred message buffer by reading FLXAnFRRDDSm, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Table 24.104 Assignment of FLXAnFROBCM bits

Pos.	Access	Bit	Function
17	R	RDSH	Data Section available for Host access
16	R	RHSH	Header Section available for Host access
1	R/W	RDSS	Read Data Section Shadow
0	R/W	RHSS	Read Header Section Shadow

Table 24.105 Assignment of FLXAnFROBCR bits

Pos.	Access	Bit	Function
22 to 16	R	OBRH[6:0]	OBF Request Host, number of message buffer available for Host access
15	R	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	R/W	REQ	Request Transfer from Message RAM to OBF Shadow
8	R/W	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6 to 0	R/W	OBRS[6:0]	OBF Request Shadow, number of message buffer for next request

24.4.12.3 FlexRay Protocol Controller Access to Message RAM

The two TBFs (A,B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each TBF is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If e.g. the Message Handler writes the next message to be sent to TBF Tx, the FlexRay Channel Protocol Controller can access TBF Rx to store the message it is actually receiving. During transmission of the message stored in TBFTx, the Message Handler transfers the last received message stored in TBF Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the TBFs and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

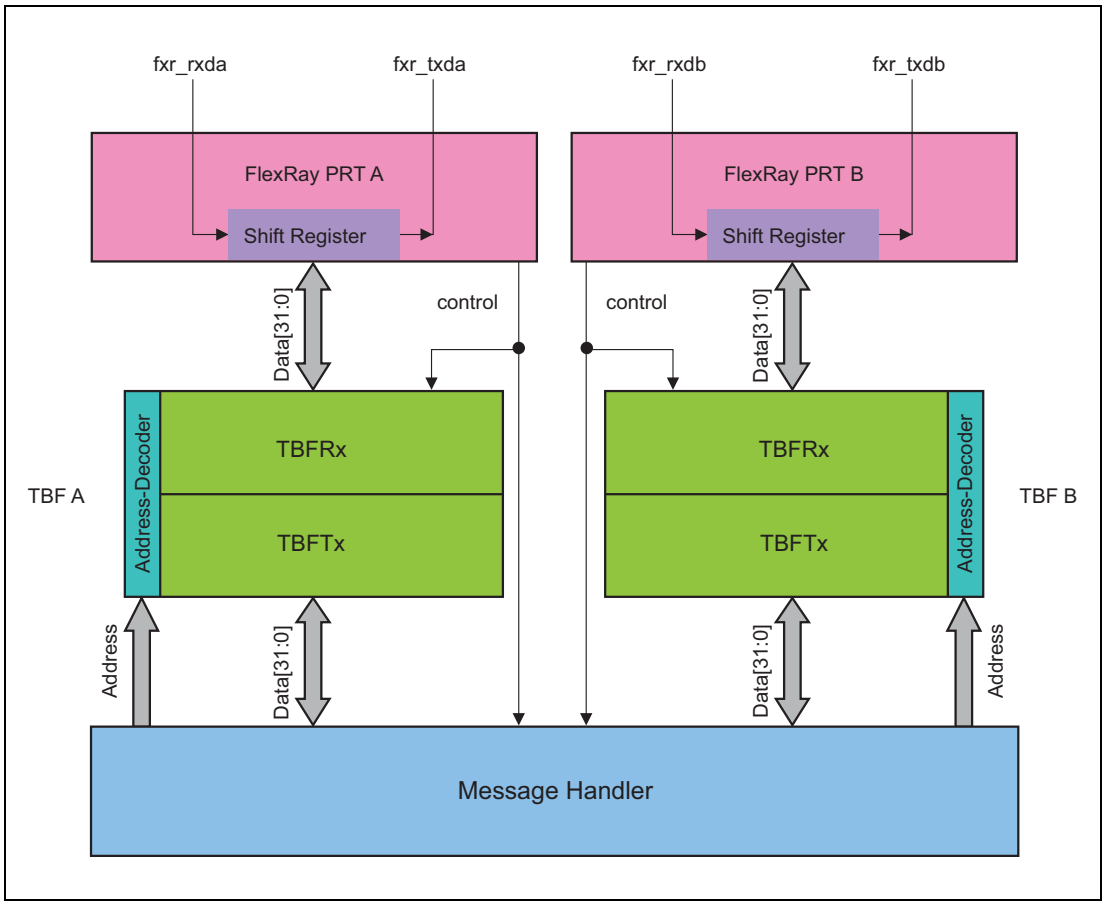


Figure 24.17 Access to TBFs

24.4.13 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is able to store up to 2048 32-bit words. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in **Figure 24.18**.

When a message buffer of the data section to be allocated immediately after to the header partition is set as a reception buffer (bit FLXAnFRWRHS1.CFG to 0) or reception FIFO buffer, set a 32-bit unused area (minimum) at the beginning of the data section. In this case, the data partition is allowed to start at Message RAM word number: ((bits FLXAnFRMRC.LCB[7:0] + 1) × 4) + 1.

When a message buffer of the data section to be allocated immediately after to the header partition is set as a transmit buffer (bit FLXAnFRWRHS1.CFG to 1), the data partition is allowed to start at Message RAM word number: (bits FLXAnFRMRC.LCB[7:0] + 1) × 4.

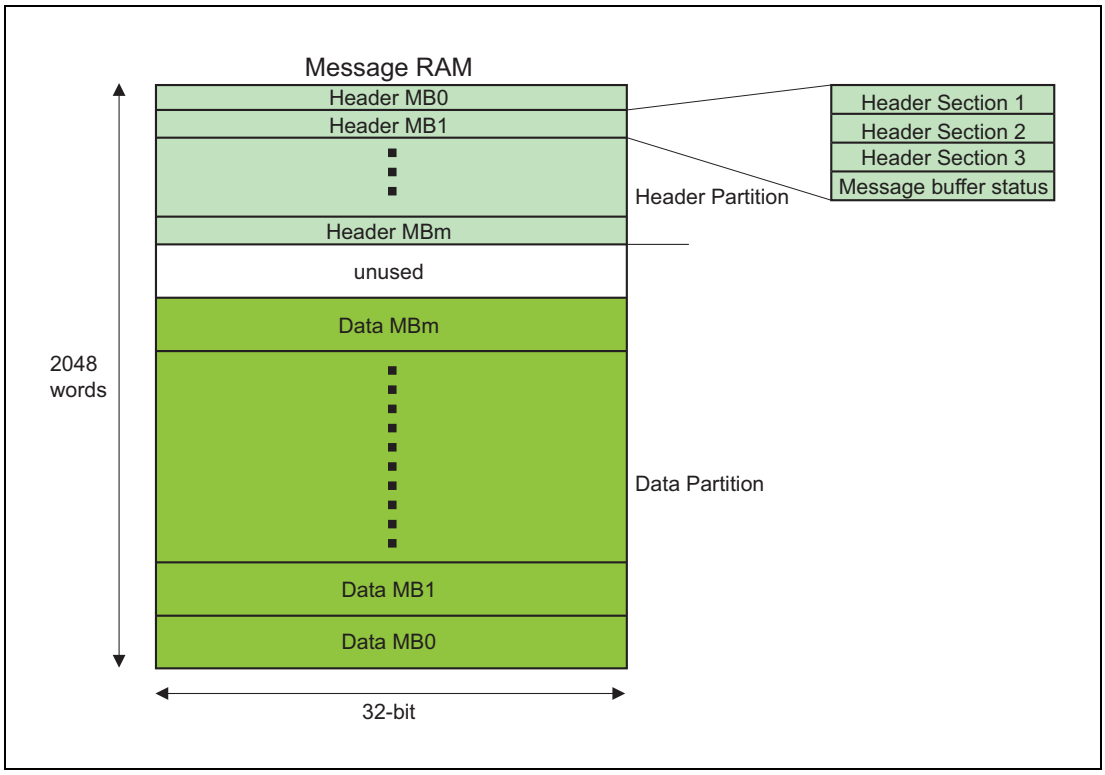


Figure 24.18 Configuration Example of Message Buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

CAUTION

header partition + data partition may not occupy more than 2048 32-bit words.

24.4.13.1 Header Partition

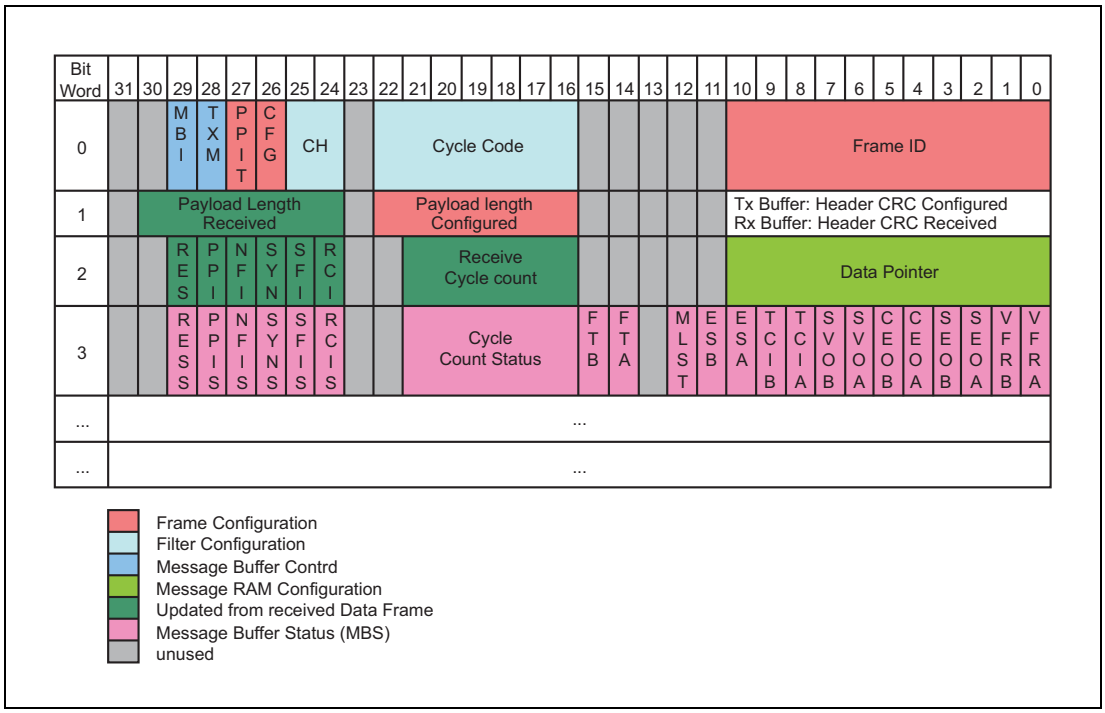
The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in **Table 24.106** below. Configuration of the header sections of the message buffers is done via IBF (FLXAnFRWRHS1 to FLXAnFRWRHS3). Read access to the header sections is done via OBF (FLXAnFRRDHS1 to FLXAnFRRDHS3 + FLXAnFRMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 32-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received PLR, Receive Cycle Count RCC, Received on Channel Indicator RCI, Startup Frame Indicator SFI, Sync Frame Indicator SYN, Null Frame Indicator NFI, Payload Preamble Indicator PPI, and Reserved Bit RES are updated from received valid data frames only.

Table 24.106 Header Section of a Message Buffer in the Message RAM



(1) Header section 1 (word 0)

Write access via FLXAnFRWRHS1, read access via FLXAnFRRDHS1:

- Frame ID
  - Slot counter filtering configuration
- Cycle Code
  - Cycle counter filtering configuration
- CH
  - Channel filtering configuration
- CFG
  - Message buffer direction configuration: receive / transmit
- PPIT
  - Payload Preamble Indicator Transmit
- TXM
  - Transmit mode configuration: single-shot / continuous
- MBI
  - Message buffer receive / transmit interrupt enable

(2) Header section 2 (word 1)

Write access via FLXAnFRWRHS2, read access via FLXAnFRRDHS2:

- Header CRC
  - Transmit Buffer: Configured by the Host (calculated from frame header)
  - Receive Buffer: Updated from received frame
- Payload Length Configured
  - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
  - Length of payload segment (2-byte words) stored from received frame

(3) Header section 3 (word 2)

Write access via FLXAnFRWRHS3, read access via FLXAnFRRDHS3:

- Data Pointer
  - Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXAnFRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count
  - Cycle count from received frame
- RCI
  - Received on Channel Indicator

- SFI
  - Startup Frame Indicator
- SYN
  - Sync Frame Indicator
- NFI
  - Null Frame Indicator
- PPI
  - Payload Preamble Indicator
- RES
  - Reserved bit

(4) Message Buffer Status FLXAnFRMBS (word 3)

Read access via FLXAnFRMBS, updated by the CC at the end of the configured slot.

- VFRA
  - Valid Frame Received on channel A
- VFRB
  - Valid Frame Received on channel B
- SEOA
  - Syntax Error Observed on channel A
- SEOB
  - Syntax Error Observed on channel B
- CEOA
  - Content Error Observed on channel A
- CEOB
  - Content Error Observed on channel B
- SVOA
  - Slot boundary Violation Observed on channel A
- SVOB
  - Slot boundary Violation Observed on channel B
- TCIA
  - Transmission Conflict Indication channel A
- TCIB
  - Transmission Conflict Indication channel B
- ESA
  - Empty Slot Channel A

- ESB
  - Empty Slot Channel B
- MLST
  - Message LoST
- FTA
  - Frame Transmitted on Channel A
- FTB
  - Frame Transmitted on Channel B
- Cycle Count Status
  - Actual cycle count when status was updated
- RCIS
  - Received on Channel Indicator Status
- SFIS
  - Startup Frame Indicator Status
- SYNS
  - Sync Frame Indicator Status
- NFIS
  - Null Frame Indicator Status
- PPIS
  - Payload Preamble Indicator Status
- RESS
  - Reserved bit Status

24.4.13.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the user has to assure that the data pointers point to addresses within the data partition. **Table 24.107** below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32bit word are unused (see **Table 24.107** below).

Table 24.107 Example for Structure of the Data Partition in the Message RAM

Bit Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	unused								unused								unused								unused							
...	unused								unused								unused								unused							
...	MBn Data3								MBn Data2								MBn Data1								MBn Data0							
...	...								...								...								...							
...	...								...								...								...							
...	MBn Data(m)								MBn Data(m-1)								MBn Data(m-2)								MBn Data(m-3)							
...	...								...								...								...							
...	...								...								...								...							
...	...								...								...								...							
...	MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
...	...								...								...								...							
2046	MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
2047	unused								unused								MB0 Data5								MB0 Data4							

24.4.13.3 Message Data Integrity Check

There is a data integrity checking mechanism implemented in the FlexRay core to assure the integrity of the data stored in the related RAM. Each RAM has a checksum generator / checker attached as shown in **Figure 24.19**.

When data is written to a RAM, the local checksum generator generates the checksum. The checksum is stored together with the respective data word. The checksum is checked each time a data word is read from a RAM.

If a checksum error is detected, the respective access error flag is set. The access error flags (bits AMR, ATBF1, and ATBF2) and the faulty message buffer indicators (bits FMBD, MFMB, and FMB) are located in the FlexRay Message Handler Status register (FLXAnFRMHDS). These single access error flags control the error interrupt flag (bit FLXAnFREIR.AERR).

Figure 24.19 shows the data paths between the Input Buffer, TBF and Message RAM.

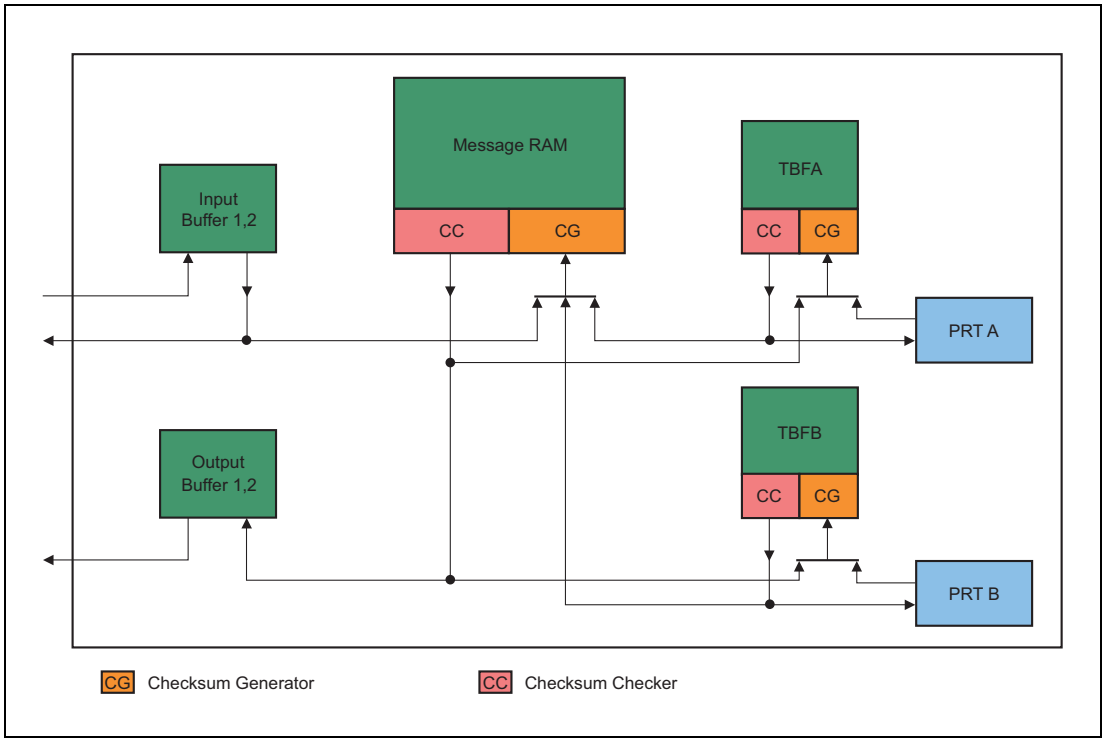


Figure 24.19 Checksum Generation and Check

When an access error has been detected the following actions will be performed:

In all cases:

- The respective access error flag in FLXAnFRMHDS register is set
- The access error flag FLXAnFREIR.AERR is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases:

(1) Access error during data transfer from Input Buffer 1, 2 to Message RAM when reading header section of respective message buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.
- Transmit buffer: Transmission request for the respective message buffer is not set.

(2) Access error during scan of header sections in Message RAM:

- FLXAnFRMHDS.AMR is set.
- Bit FLXAnFRMHDS.FMBD is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- Ignore message buffer (message buffer is skipped).

(3) Access error during data transfer from Message RAM to TBFA and TBFB:

- FLXAnFRMHDS.AMR is set.
- Bit FLXAnFRMHDS.FMBD is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero.

(4) Access error during data transfer from TBFA and TBFB to Message RAM when reading header section of respective message buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.

(5) Access error during data transfer from Message RAM to Output Buffer:

- The FLXAnFRMHDS.AMR bit is set to 1.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.

(6) Access error during a data transfer from TBFA and TBFB to Protocol Controller 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- Frames already in transmission are invalidated by setting the frame CRC to zero.

(7) Access error during data transfer from TBFA and TBFB to Message RAM when reading TBFA and TBFB:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that bits FLXAnFRMHDS.FMB[6:0] point to a faulty message buffer.
- Bits FLXAnFRMHDS.FMB[6:0] indicate the number of the faulty message buffer.

(8) Access error during data read of TBFA and TBFB:

When an access error occurs while the Message Handler read a frame with network management information (PPI = 1) from the TBFA and TBFB, the corresponding network management vector registers FLXAnFRNMV1 to 3 are not updated from this frame.

24.4.13.4 Host Handling of Access Errors

Access error caused by bit flips can be fixed by:

(1) Self-healing

Access errors located in the Data Section of Message RAM, TBFA or TBFB are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

(2) CLEAR_RAMS Command

The POC command CLEAR_RAMS initializes the message RAM to zero, when called in the DEFAULT_CONFIG or CONFIG state.

(3) Temporary Unlocking of Header Section

An access error in the header section of a locked message buffer can be fixed by a transfer from the Input Buffer to the locked buffer Header Section. For this transfer, the write access to the FLXAnFRIBCR register (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see **Section 24.3.3.1, FLXAnFRLCK — FlexRay Lock Register**).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by FLXAnFRMRC.SEC, and will be updated with new data.

24.4.14 Interrupts

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable / disable controls for each individual interrupt source separately.

An interrupt may be triggered when

- An error was detected
- A status flag is set to 1
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A message transfer from the Local RAM/Cluster RAM to Message RAM or from Message RAM to Local RAM/Cluster RAM has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers FLXAnFREIR, FLXAnFRSIR, FLXAnFROS, FLXAnFROTS and FLXAnFRITS.

The general purpose interrupt lines to the Host, FlexRay Line 0 Interrupt, and FlexRay Line 1 Interrupt are controlled by the interrupts enabled by the FLXAnFREIES and FLXAnFRSIES registers. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

The input data transfer interrupt lines to the Host, FlexRay input queue empty interrupt, FlexRay input queue full interrupt, are controlled by the enabled interrupts in FLXAnFRITS. In addition, each of the input data transfer interrupts can be enabled or disabled separately by setting the related bits in FLXAnFRITC.

The output data transfer interrupt lines to the Host, FlexRay FIFO transfer warning interrupt, FlexRay output transfer warning interrupt, FlexRay FIFO transfer interrupt, FlexRay output transfer interrupt, are controlled by the enabled interrupts in FLXAnFROTS. In addition each of the output data transfer interrupts can be enabled or disabled separately by setting the related bits in FLXAnFROTC.

The three timer interrupts lines to the Host are controlled by the enabled interrupts in FLXAnFROS. In addition each of the interrupt lines can be enabled or disabled separately by setting bit FLXAnFROC.T0IE, FLXAnFROC.T1IE and FLXAnFROC.T2IE.

When a transfer between IBF/OBF and the Message RAM has completed bit FLXAnFRSIR.TIBC or FLXAnFRSIR.TOBC is set to 1.

An stop watch event is generated by the stpwt_extfxr pin input.

24.4.15 Assignment of FlexRay Configuration Parameters

Table 24.108 FlexRay configuration parameters (1/2)

Parameter	Bit (field)
pKeySlotusedForStartup	FLXAnFRSUCC1.TXST
pKeySlotUsedForSync	FLXAnFRSUCC1.TXSY
gColdStartAttempts	FLXAnFRSUCC1.CSA[4:0]
pAllowPassiveToActive	FLXAnFRSUCC1.PTA[4:0]
pWakeupChannel	FLXAnFRSUCC1.WUCS
pSingleSlotEnabled	FLXAnFRSUCC1.TSM
pAllowHaltDueToClock	FLXAnFRSUCC1.HCSE
pChannels	FLXAnFRSUCC1.CCHA, CCHB
pdListenTimeOut	FLXAnFRSUCC2.LT[20:0]
gListenNoise	FLXAnFRSUCC2.LTN[3:0]
gMaxWithoutClockCorrectionPassive	FLXAnFRSUCC3.WCP[3:0]
gMaxWithoutClockCorrectionFatal	FLXAnFRSUCC3.WCF[3:0]
gNetworkManagementVectorLength	FLXAnFRNEMC.NML[3:0]
gdTSSTransmitter	FLXAnFRPRTC1.TSST[3:0]
gdCASRxLowMax	FLXAnFRPRTC1.CASM[6:0]
gdSampleClockPeriod	FLXAnFRPRTC1.BRP[1:0]
pSamplesPerMicrotick	FLXAnFRPRTC1.BRP[1:0]
gdWakeupSymbolRxWindow	FLXAnFRPRTC1.RXW[8:0]
pWakeupPattern	FLXAnFRPRTC1.RWP[5:0]
gdWakeupSymbolRxIdle	FLXAnFRPRTC2.RXI[5:0]
gdWakeupSymbolRxLow	FLXAnFRPRTC2.RXL[5:0]
gdWakeupSymbolTxIdle	FLXAnFRPRTC2.TXI[5:0]
gdWakeupSymbolTxLow	FLXAnFRPRTC2.TXL[5:0]
gPayloadLengthStatic	FLXAnFRMHDC.SFDL[6:0]
pLatestTx	FLXAnFRMHDC.SLT[12:0]
pMicroPerCycle	FLXAnFRGTUC1.UT[19:0]
gMacroPerCycle	FLXAnFRGTUC2.MPC[13:0]
gSyncNodeMax	FLXAnFRGTUC2.SNM[3:0]
pMicroInitialOffset[A]	FLXAnFRGTUC3.UIOA[7:0]
pMicroInitialOffset[B]	FLXAnFRGTUC3.UIOB[7:0]
pMacroInitialOffset[A]	FLXAnFRGTUC3.MIOA[6:0]
pMacroInitialOffset[B]	FLXAnFRGTUC3.MIOB[6:0]
gdNIT	FLXAnFRGTUC4.NIT[13:0]
gOffsetCorrectionStart	FLXAnFRGTUC4.OCS[13:0]
pDelayCompensation[A]	FLXAnFRGTUC5.DCA[7:0]
pDelayCompensation[B]	FLXAnFRGTUC5.DCB[7:0]
pClusterDriftDamping	FLXAnFRGTUC5.CDD[4:0]
pDecodingCorrection	FLXAnFRGTUC5.DEC[7:0]
pdAcceptedStartupRange	FLXAnFRGTUC6.ASR[10:0]
pdMaxDrift	FLXAnFRGTUC6.MOD[10:0]
gdStaticSlot	FLXAnFRGTUC7.SSL[9:0]
gNumberOfStaticSlots	FLXAnFRGTUC7.NSS[9:0]

Table 24.108 FlexRay configuration parameters (2/2)

Parameter	Bit (field)
gdMinislot	FLXAnFRGTUC8.MSL[5:0]
gNumberOfMinislots	FLXAnFRGTUC8.NMS[12:0]
gdActionPointOffset	FLXAnFRGTUC9.APO[5:0]
gdMinislotActionPointOffset	FLXAnFRGTUC9.MAPO[4:0]
gdDynamicSlotIdlePhase	FLXAnFRGTUC9.DSI[1:0]
pOffsetCorrectionOut	FLXAnFRGTUC10.MOC[13:0]
pRateCorrectionOut	FLXAnFRGTUC10.MRC[10:0]
pExternOffsetCorrection	FLXAnFRGTUC11.EOC[2:0]
pExternRateCorrection	FLXAnFRGTUC11.ERC[2:0]

24.4.16 Usage of Data Transfer

The data transfer function is for the transfer of FlexRay messages between the FlexRay module's internal message RAM and areas of user RAM such as Local RAM/Cluster RAM.

Data transfer from an area of user RAM to the internal message RAM (input) is started by software. Input transfer is used to set up a message buffer or to update data for transmission.

Data transfer from the internal message RAM to an area of user RAM (output) starts in response to the following events.

- Data are stored in the reception message buffer or the internal FIFO buffer of the FlexRay module.
- The state of a slot changes.
- The user issues a transfer request.

An area of the user RAM needs to be secured for the data to be transferred. The data are transferred in a defined structure.

Access to the input and output buffers by the CPU is prohibited while this function is in use.

24.4.16.1 Input Data Transfer

When the automatic input data transfer function is enabled, committed input data structures are transferred from the Local RAM/Cluster RAM to the FlexRay internal message RAM with minimum CPU support.

(1) Activation and deactivation

The input data transfer function should be activated before usage. The activation of the input transfer handler initializes the input queue put index (FLXAnFRITS.IPIDX[6:0]) and get index (FLXAnFRITS.IGIDX[6:0]) to zero. Also the interrupt status bits in the FLXAnFRITS register (IQEIS and IQFIS) are set to 0.

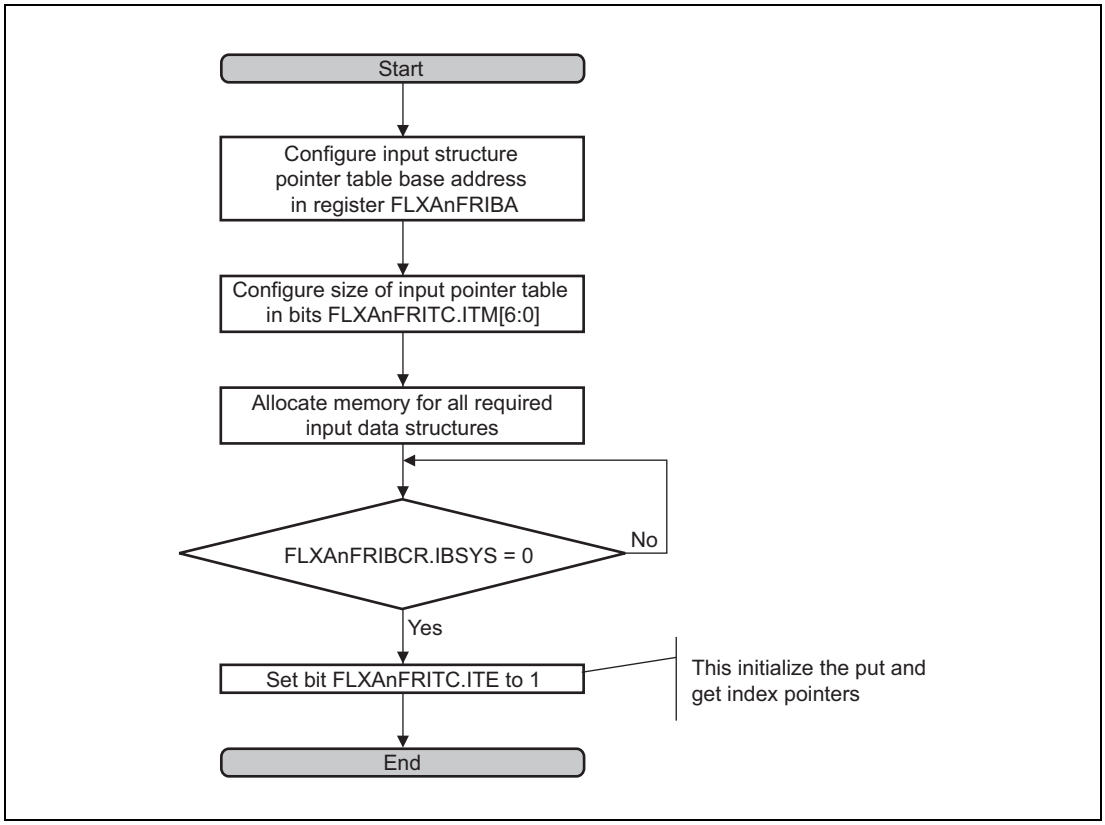


Figure 24.20 Input Transfer Enable Processing Flow

A deactivation request of the input transfer function can be made at any time. The input queue put index and the input queue status are maintained independently from the input transfer function state.

Before the transfer function gets disabled (bit FLXAnFRITS.ITS = 0), user requested input transfers and all committed input transfers will be completed.

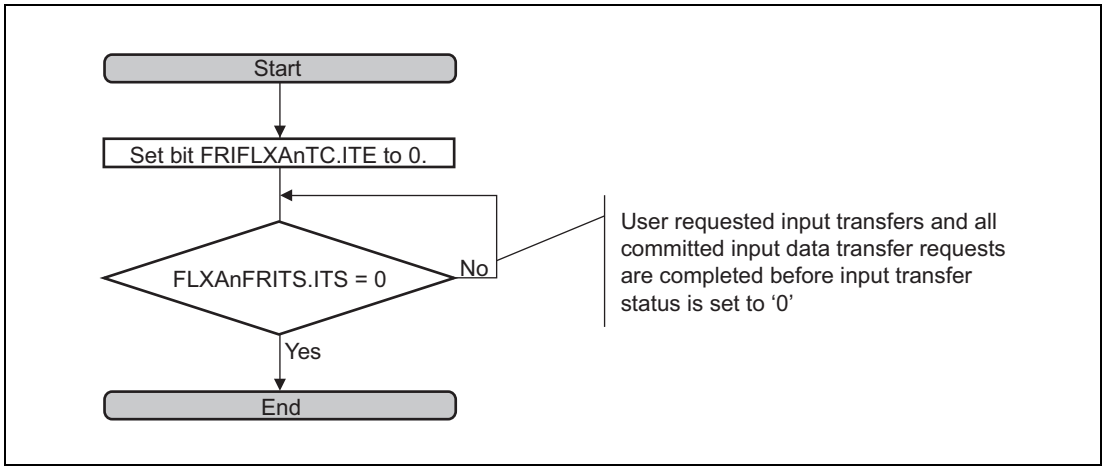


Figure 24.21 Input Transfer Disable Processing Flow

(2) Input data structure

The application has to reserve a location in the Local RAM/Cluster RAM to provide the content for message buffer configuration (input data structure).

The location of this input data structure needs to be defined by an input data structure pointer also located in the Local RAM/Cluster RAM.

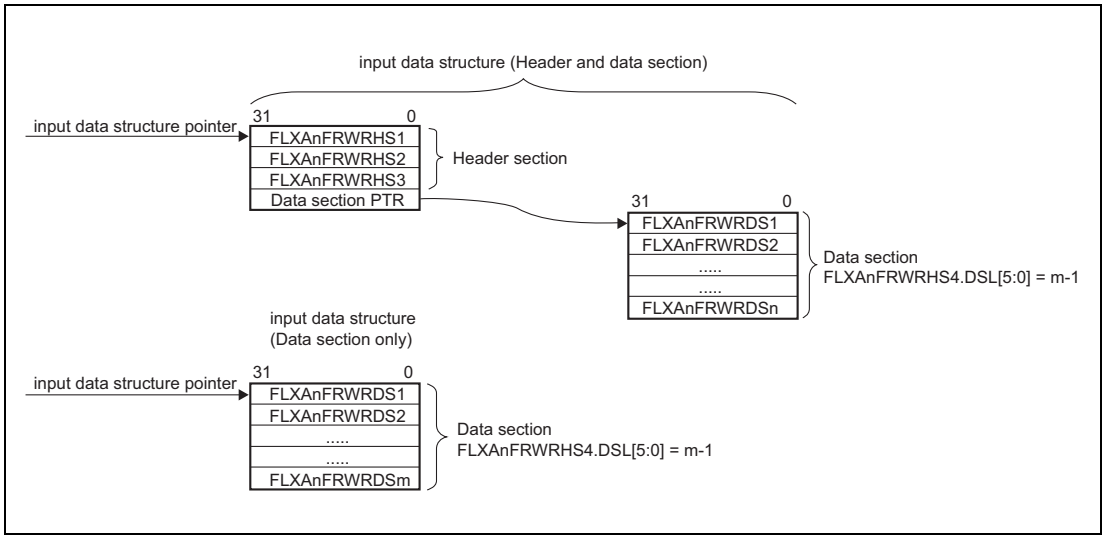


Figure 24.22 Input data structure

In general the input data structure consists of two sections, the header and the data section.

The header section consists of FLXAnFRWRHS1, FLXAnFRWRHS2, FLXAnFRWRHS3 and the data section pointer.

For bit alignment and bit function in the header section, see **Section 24.4.13.1, Header Partition**.

Depending on the settings in the control field (FLXAnFRWRHS4) located in the input pointer table, the data structure pointer is a reference to the address of FLXAnFRWRHS1 or FLXAnFRWRDS1. The data structure pointer has to be aligned to a 32 bit address.

If bit FLXAnFRWRHS4.LHS is set to 1, it is required to provide a valid header section. In this case FLXAnFRWRHS1 is the first element of the data structure.

If bit FLXAnFRWRHS4.LHS is set to 0, a header section is not required. In this case FLXAnFRWRDS1 is the first element of the data structure.

If bit FLXAnFRWRHS4.LDS is set to 1, it is required to provide a valid data section. The pointer to the data section is a reference to the address of the first payload long word (FLXAnFRWRDS1) and has to be aligned to a 32 bit address.

If bit FLXAnFRWRHS4.LDS is set to 0, a data section is not required. The data section pointer is not evaluated by the input handler.

The length of the data section and the size to be allocated in the Local RAM/Cluster RAM depends on the configuration of bits FLXAnFRWRHS4.DSL[5:0].

For the transfer into the FlexRay core internal message RAM the number of 16 bit words configured by bits FLXAnFRWRHS2.PLC[6:0] is used. The application has to ensure, that a proper number of data words is provided in the Local RAM/Cluster RAM. In case the buffer is configured by bits FLXAnFRWRHS2.PLC[6:0] to hold an odd payload length, the application has to write zero to the last 16 bit of the payload section to ensure that the padding data is all zero.

(3) Input pointer table

To transfer data from the input data structures located in the Local RAM/Cluster RAM to the FlexRay internal message RAM the related input data structure pointer and control field needs to be added to the input pointer table which is located in the Local RAM/Cluster RAM.

The location of the first element of this table is identified by the input pointer table base address (FLXAnFRIBA.ITA[31:0]). This base address has to be aligned to a 32 bit address.

The maximum number of input requests that can be queued is defined by the Input queue Table Max register (FLXAnFRITC.ITM[6:0]).

Each Input pointer table entry requires two long words. The required address range of the input pointer table for the queued transfer requests can be calculated by

Input pointer table size (byte) = (((FLXAnFRTTC.ITM[6:0] + 1) × 2) × 4)

Equation 1

The input pointer entry for the user requested input transfer should be added after the end of the input pointer table.

The pointer table index related to this entry and hence the number to be written to FLXAnFRUIR.UIDX[7:0], is FLXAnFRITC.ITM[6:0] + 1. The address in the input pointer table related to the user requested input transfer (user input address) can be calculated by

User input address = FLXAnFRIBA.ITA[31:0] + Input pointer table size

Equation 2

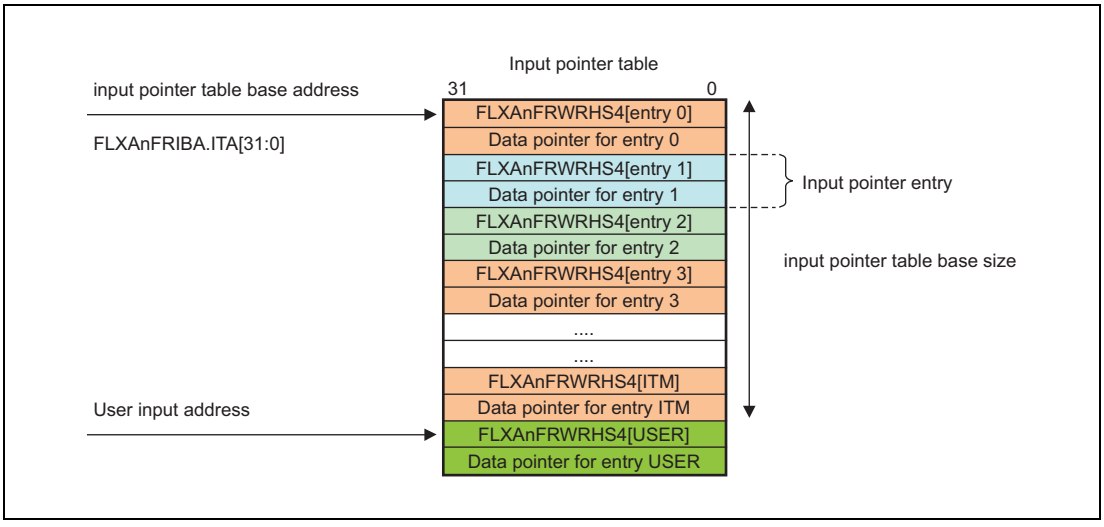


Figure 24.23 Input pointer table

The input pointer table holds the control field FLXAnFRWRHS4 and the pointers to the Local RAM/ Cluster RAM location where the message buffer content (header section and/or data section) is stored.

The application has to write FLXAnFRWRHS4 and the input data structure pointer at the addresses in the input pointer table related to the put index position before a transfer request is initiated.

FLXAnFRWRHS4:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	DSL[5:0]					
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	INV	STR	LDS	LHS	—	IMBNR[6:0]						
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 24.109 FLXAnFRWRHS4 Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, an undefined value is read. When writing, always write 0.
21 to 16	DSL[5:0]	Data Section Length Bit Specifies the length of the data section in terms of 32 bit values.
15 to 12	Reserved	When read, an undefined value is read. When writing, always write 0.
11	INV	Invalidate entry Bit 0: The data structure is valid and will be transferred to the FlexRay internal message RAM. 1: The data structure is invalid. FlexRay internal message RAM is not updated using this input pointer entry.
10	STR	Set transmission request Bit 0: The bit FLXAnFRTXRQm.TXRp for the message buffer selected by the bits IMBNR[6:0] is set to 0. No data from this message buffer is transmitted. 1: The bit FLXAnFRTXRQm.TXRp for the message buffer selected by the bits IMBNR[6:0] is set to 1 to release the message buffer for transmission The application should not set the bit STR to 1 for receive buffers.
9	LDS	Load data section Bit 0: No update of data section. 1: Data section for the message buffer selected by the bits IMBNR[6:0] is updated.
8	LHS	Load header section Bit 0: No update of header section. 1: Header section for the message buffer selected by the bits IMBNR[6:0] is updated.
7	Reserved	When read, an undefined value is read. When writing, always write 0.
6 to 0	IMBNR[6:0]	Message buffer number to be updated Bit Selects the target message buffer number in the FlexRay internal message RAM for transfer

Note that the LHS bit should not be set for protected message buffers.

The bit LDS defines if the data section of the message buffer selected by the bits IMBNR[6:0] should be updated.

If LDS is set to 1 (DSL[5:0] + 1) 32 bit words of payload data are transferred from the Local RAM/ Cluster RAM to the message buffer selected by the bits IMBNR[6:0].

If LDS is set to 0 no payload data is transferred from the Local RAM/Cluster RAM.

Note that the payload transferred is independent from the configured payload length (bits FLXAnFRWRHS2.PLC[6:0]).

The bit INV can be used to invalidate a committed data structure. This bit should be only used to cancel the transfer of committed data structures when the input queue is halted (see **Section 24.4.16.2, (5) Transfer function of dedicated message buffers**).

When this bit is set to 1 the message buffer number (IMBNR[6:0]) is not updated. When the bit is set to 0 the message buffer number (IMBNR[6:0]) is updated.

(4) Transfer function of input data structure

To use the input data structure transfer function the input transfer has to be activated (see **Section 24.4.16.1, (1) Activation and deactivation**). The activation process requires the setup of the input pointer table (see **Section 24.4.16.1, (3) Input pointer table**) in order to specify the source location (input data structures) for the data structures to be transferred. When the input transfer gets enabled the get index pointer is initialized to zero.

All FlexRay internal message buffers can be updated using the input transfer queue which is built in the input pointer table. The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. For that purpose the application has to maintain a put index for the input pointer table that indicates where the pointer has to be written to.

To commit this table entry to the input handler, the application has to write the target message buffer number to the input queue control register (bits FLXAnFRIQC.IMBNR[6:0]). Afterwards the application has to increment the application internal put index.

By writing to the input queue control register the data available bits (FLXAnFRDAm.DA[31:0]) are automatically set to 1. The input transfer handler also maintains the put index pointer in the status register (FLXAnFRITS.IPIDX[6:0]).

In case the input queue gets full (number of queued input transfer requests is equal to the input queue table size) FLXAnFRITS.IQFP and FLXAnFRITS.IQFIS are set to 1. The input queue full condition pending flag (FLXAnFRITS.IQFP) changes from 1 to 0 when there are entries in the input queue available, whereby the input queue full interrupt status flag (FLXAnFRITS.IQFIS) needs to be cleared by the application.

The application should not make any further write access to bits FLXAnFRIQC.IMBNR[6:0] as long as bit FLXAnFRITS.IQFP is 1.

In case the input queue gets empty (number of queued input transfer requests changes to zero) FLXAnFRITS.IQEIS is set to 1. The input queue empty interrupt status flag (FLXAnFRITS.IQEIS) needs to be cleared by the application.

The transfer of the input data structures to the FlexRay message RAM is controlled by a get index pointer which is handled inside the FlexRay module and flagged in bits FLXAnFRITS.IGIDX[6:0]. Note that the index is referring to the input entry and not the address offset in the input pointer table.

If the input queue is not empty, the transfer handler reads out the input pointer table entry of the transfer queue and starts the transfer of the input data structure from the address the input pointer is referring to. When all required data words are transferred to the FlexRay module, the data available flag for the transferred message buffer number is set to 0 and the get index in the transfer handler is incremented by one.

In case of an invalidated data structure (see **Section 24.4.16.1, (5) Halting the input queue**) no FlexRay internal message buffer is updated and the related data available flag is automatically set to 0. The change of the data available flag can be used to confirm the cancellation a transmit request.

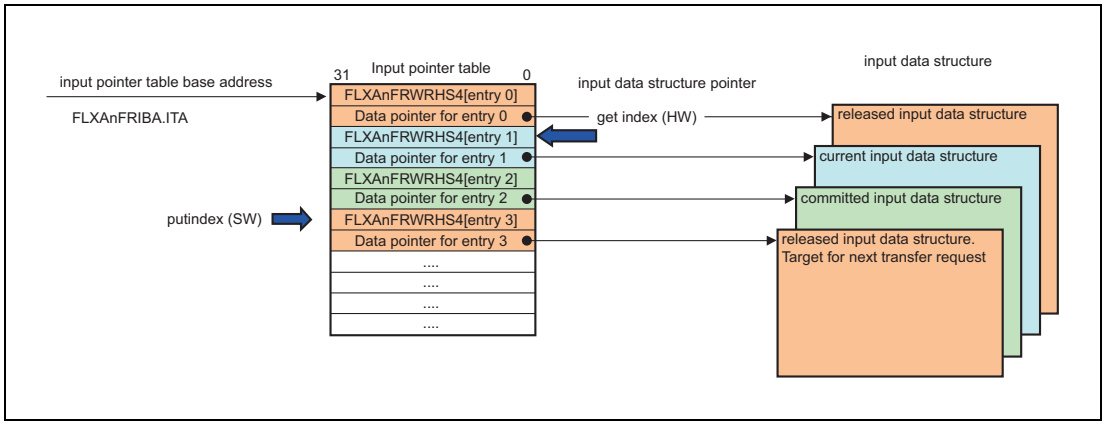


Figure 24.24 Input Pointer Table

Receive message buffers can be also configured using the input data transfer by setting up the required header sections and mark only the header section (FLXAnFRWRHS4.LDS = 0, FLXAnFRWRHS4.LHS = 1) to be updated in the FlexRay module.

(5) Halting the input queue

Committed data structures cannot be removed, but can be invalidated or updated when the input queue is halted.

To cancel data structures already committed to the input queue, the queue can be halted by writing a 1 to FLXAnFRITC.IQHR.

After the ongoing input transfer has been completed the queue is halted and FLXAnFRITS.IQH changes from 0 to 1.

To invalidate an entry of the input queue FLXAnFRWRHS4.INV has to be set to 1. All other bits in FWRHS4 should be unchanged.

The following processing flow shall be used to analyze whether a committed message has been already transferred to the FlexRay internal message RAM or not.

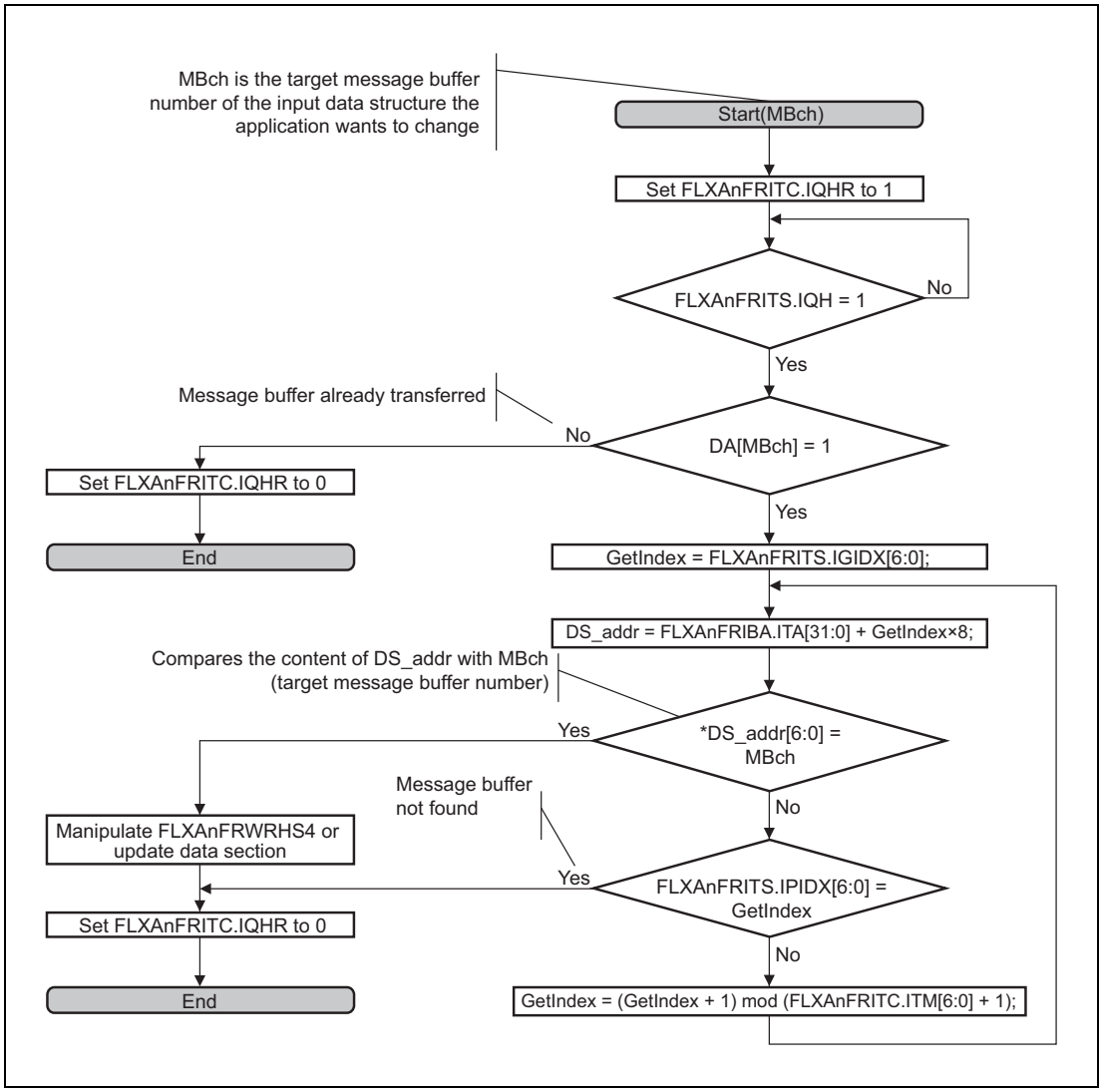


Figure 24.25 Input Table Analysis

In case the message buffer was already transferred to the FlexRay internal message RAM, the user input transfer request can be used to bypass the actual queue and update the required message buffer (see **Section 24.4.16.1, (6) Transfer function of user requested input transfers**).

(6) Transfer function of user requested input transfers

To use this function the input transfer has to be activated (see **Section 24.4.16.1, (1) Activation and deactivation**).

The application is capable, by using FLXAnFRUIR.UIDX[7:0], to request a transfer of an input data structure. The user input transfer request is serviced first.

The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. The table entry for the user input transfer request should be added after the end of the input pointer table (see **Section 24.4.16.1, (3) Input pointer table**).

To commit this table entry to the input handler, the application has to write the index (FLXAnFRITC.ITM[6:0] +1) to the user input transfer request register (FLXAnFRUIR.UIDX[7:0]).

By writing to the user input transfer request register, the user input transfer request pending flag (FLXAnFRITS.UIRP) is automatically set to 1.

As long this flag is 1 the application should not make any further user input transfer requests.

The user input transfer request pending flag (FLXAnFRITS.UIRP) changes from 1 to 0 when the requested input transfer is completed. As next the pending transfers are processed.

24.4.16.2 Output Data Transfer

When the output data transfer function is enabled, received messages (either in dedicated message buffers or in the FlexRay receive FIFO) are transferred to the Local RAM/Cluster RAM by the output data handler. The output data handler can also transfer the message buffer content to the Local RAM/ Cluster RAM on application request. When enabled the output handler is also capable to initiate a transfer when the message buffer status has changed.

(1) Activation and deactivation

The output data transfer function should be activated before usage. The activation of the output transfer handler will initialize the FIFO put and get index pointer and FIFO fill level (FLXAnFROTS.FGIDX[4:0] and FLXAnFROTS.FFL[5:0]) to zero, set the bits FLXAnFROTS.FDA, FLXAnFROTS.OWP, FLXAnFROTS.FWP and FLXAnFROTS.UORP to 0. Also the interrupt status flags FLXAnFROTS.OTIS, FLXAnFROTS.FIS, FLXAnFROTS.OWIS and FLXAnFROTS.FWIS) are set to 0.

The activation has no influence to the data available flags (FLXAnFRDAm.DA[31:0]) which are related to the dedicated buffers; these flags have to be cleared by the application.

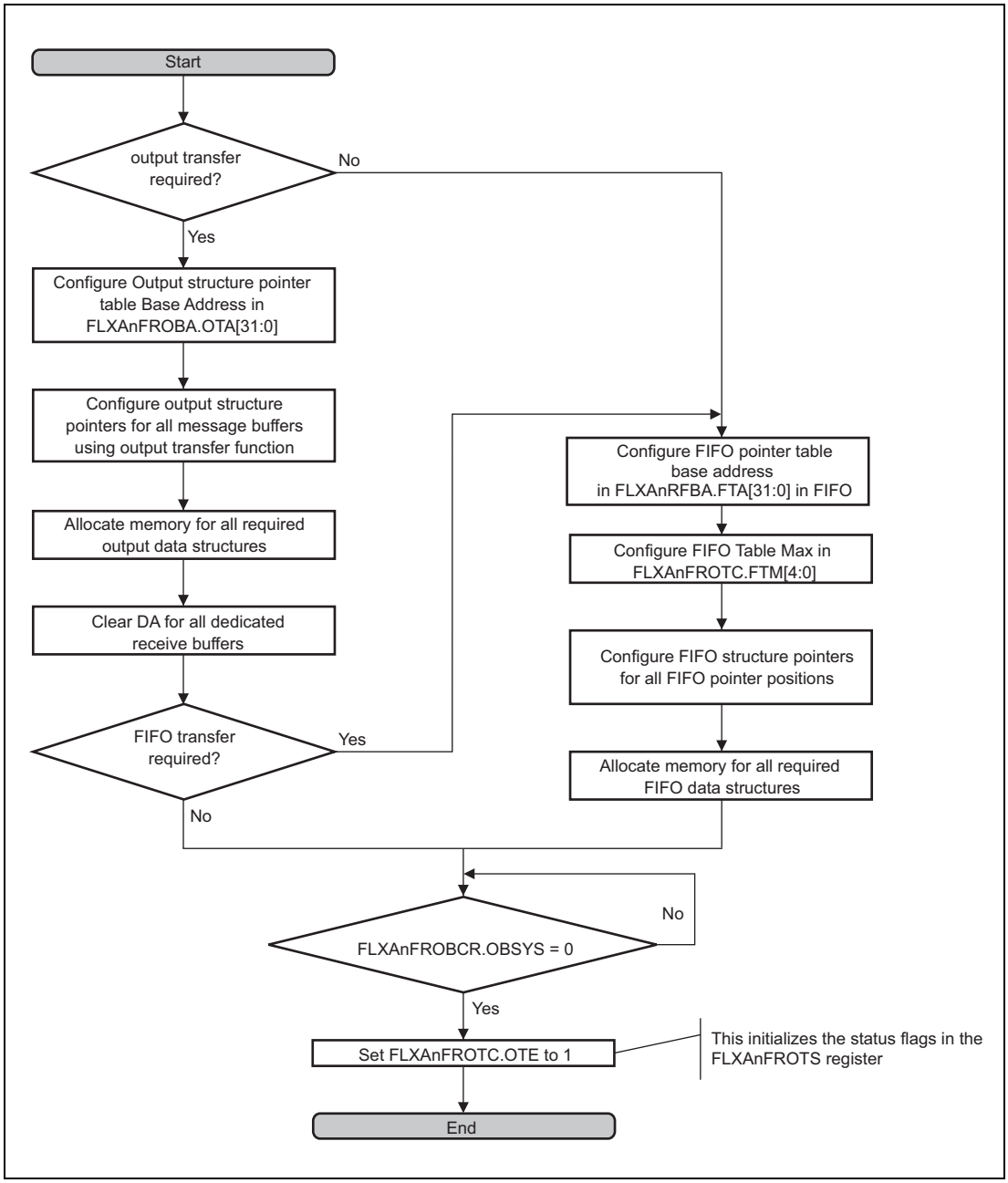


Figure 24.26 Output Transfer Enable Processing Flow

A deactivation request of the output data transfer function can be made at any time. An ongoing transfer will be completed and the completion of this transfer will be flagged. During this time FLXAnFROTS.OTS remains 1.

When FLXAnFROTS.OTS changes from 1 to 0, the output transfer function is deactivated. The data available status flags and the FIFO get index are still retained when the output transfer function is disabled.

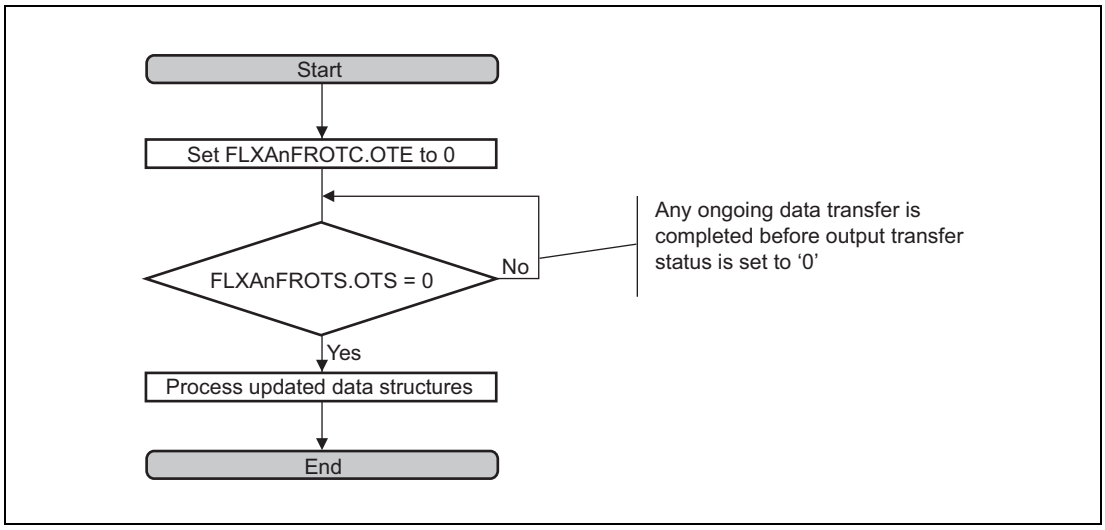


Figure 24.27 Output Transfer Disable Processing Flow

(2) Output transfer data structure

The data in the Local RAM/Cluster RAM is stored in an output data structure. The location of the output data structures are determined by output data structure pointers also located in the Local RAM/ Cluster RAM. The output data structure and indexing is visualized in **Figure 24.28**.

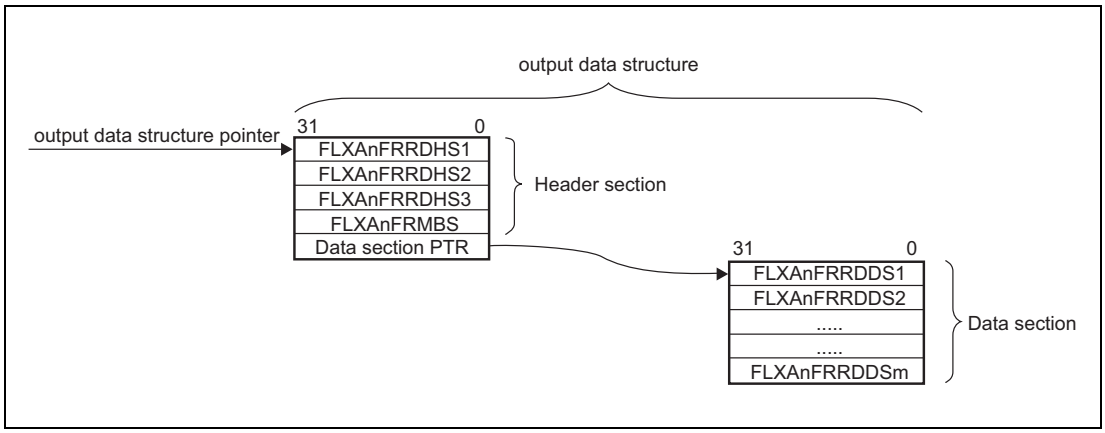


Figure 24.28 Output Data Structure

The output data structure consists of two sections, the header and data section. The header section consists of FLXAnFRRDHS1, FLXAnFRRDHS2, FLXAnFRRDHS3, FLXAnFRMBS and the data section pointer. FLXAnFRRDHS1 is the first element of the structure and has to be aligned to a 32 bit address. The data structure pointer is a reference to the address of FLXAnFRRDHS1. For information about the bit alignment and bit function within the header section refer to **Section 24.4.13.1, Header Partition**.

FLXAnFRRDDS1 is the first element of the data section. The data section pointer needs to be aligned with a 32-bit address corresponding to the FLXAnFRRDDS1 address.

The length of the data section as well as the total structure size to be allocated in the Local RAM/ Cluster RAM depends on the configured payload length (bits FLXAnFRRDHS2.PLC[6:0]) of the related message buffer. In case the configured payload length is an odd number of words or the received payload length (bits FLXAnFRRDHS2.PLR[6:0]) is smaller than the configured payload length, the remaining data words in the Local RAM/Cluster RAM are unused and should not be used by the application.

The output data structure is identical for all three kinds of output transfers. In case only the header section is transferred the data section pointer is not evaluated by the output handler and the data section remains unchanged.

(3) Output pointer table

For the output data transfer function the application needs to setup an output pointer table in the Local RAM/Cluster RAM. The location of the first element of this table should be programmed into the output pointer table base address (bits FLXAnFROBA.OTA[6:0]). This base address has to be aligned to a 32 bit address.

The size of the output pointer table is defined by the maximum of: the last configured dedicated message buffer and the highest message buffer number which will be used for the user output transfer request.

The output pointer table holds pointers (output data structure pointers) to the Local RAM/Cluster RAM. location where a memory space is reserved for the target message buffer content (header section and data section).

There is a fixed linear relationship between the address of the entries in the output pointer table and the number of the related message buffers (see **Figure 24.29**): the output pointer table starts with the entry for message buffer number 0 at the address configured in bits FLXAnFROBA.OTA[31:0] and continues in ascending order for each following message buffer number, by 32 bit aligned address (e.g. message buffer 1 at output pointer table address FLXAnFROBA.OTA[31:0] + 4, message buffer 2 at output pointer table address FLXAnFROBA.OTA[31:0] + 8, etc.) for all possible message buffers.

When a set of bit FLXAnFRNDATm.NDp is the only transfer condition (FLXAnFROTC.OTCS is set to 0), only message buffers configured as a dedicated receive buffer or that will be used for user output transfer requests need have valid pointer entries.

When a set of bit FLXAnFRNDATm.NDp or FLXAnFRMBSCm.MBCp is the transfer condition (FLXAnFROTC.OTCS is set to 1), all dedicated receive buffer and dedicated transmit buffers need to have valid pointer entries.

(4) FIFO output pointer table

The FlexRay module internal FIFO can be extended by a queued buffer structure in the Local RAM/ Cluster RAM.

If the FlexRay module internal FIFO is used the application needs to setup the FIFO output pointer table. The location of the first element of this table is identified by the FIFO pointer table base address (bits FLXAnFRFBA.FTA[31:0]). This base address has to be aligned to a 32 bit address.

The size of the FIFO pointer table and hence the maximum number of messages that can be added to the queue, is defined by FIFO Table Max (bits FLXAnFROTC.FTM[4:0]).

The FIFO pointer table holds pointers (output data structure pointers) to the Local RAM/Cluster RAM location where a memory space is reserved the target message buffer content (header section and data section). For each table entry a data pointer shall be configured in this table.

(5) Transfer function of dedicated message buffers

To use this transfer function the output transfer has to be activated (see **Section 24.4.16.2, (1) Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 24.4.16.2, (3) Output pointer table**) in order to specify the destination location (output data structures) for the data to transfer. **Figure 24.29** shows how the output pointer table references the output data structures.

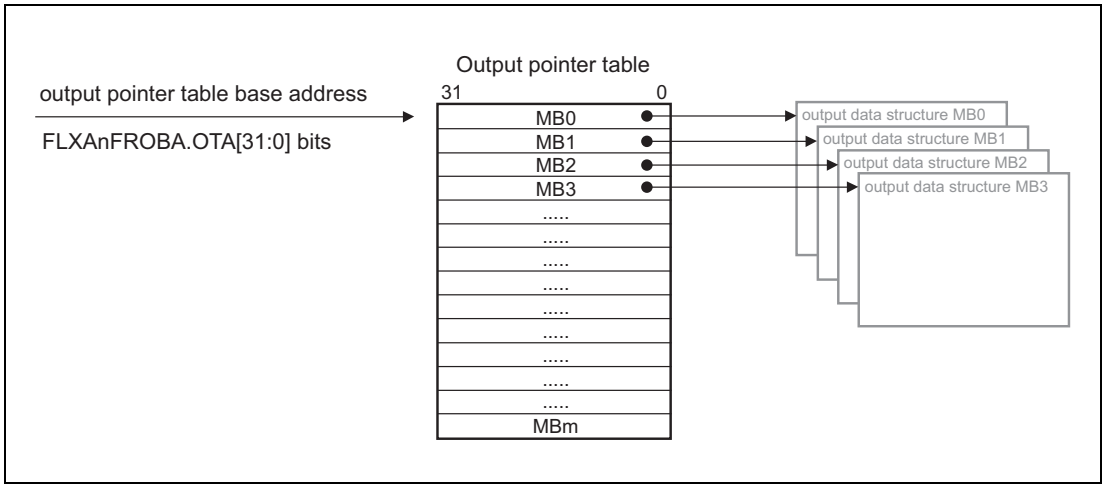


Figure 24.29 Output Data Structure and Indexing

With FLXAnFROTC.OTCS the output transfer condition can be selected between the 'New data only mode' and the 'New data and status changed mode'.

In the 'New data only mode' an output data transfer is initiated when a valid FlexRay data frame has been stored into a dedicated receive buffer which causes the related FLXAnFRNDATm.NDp flag to set. The FLXAnFRNDATm.NDp flag is automatically set to 0 during the transfer procedure. The header section is also transferred and hence the FLXAnFRMBSCm.MBCp flag is set to 0.

In the 'New data and status changed mode' an output data transfer is initiated as described in the 'New data only mode'. In addition an output data transfer is initiated when only the message buffer status has been changed which causes the related FLXAnFRMBSCm.MBCp flag to be set. In this case only the header section is transferred. The FLXAnFRMBSCm.MBCp flag is automatically set to 0 during the transfer procedure.

After transferring the message buffer data from the FlexRay internal message RAM to the output data structure the corresponding data available flag in the FLXAnFRDAm (m = 0 to 3) registers is set to 1. The update of the output data structure is also flagged by the setting of the output transfer interrupt status flag (FLXAnFROTS.OTIS).

As long as the data available flag remains 1 the corresponding output data structure will not be updated.

In the case

- the data available flag is 1 and a valid received message was stored or
- when FLXAnFROTC.OTCS is 1 and the message buffer status was updated,

the output transfer warning interrupt flag (FLXAnFROTS.OWIS) is set to 1 notifying the application that new data is available but the output data structure transfer cannot be processed. In addition FLXAnFROTS.OWP is set to 1 that continuously flags that status of the output transfer warning condition.

If a valid receive message in the FlexRay internal message RAM is overwritten by an additional receive message, the message lost flag (FLXAnFRMBS.MLST) is set to 1. This flag can be evaluated after the message buffer has been transferred into an output data structure.

Following sections are giving a guidance how output data structures can be handled.

(a) Data section copy method

One option is to copy the information from the output data structure to a different location of the Local RAM/Cluster RAM and then release the output data structure by clearing the related data available flag. The application should use the copied information for further processing.

(b) Data structure pointer method

A different option is to modify the output data structure pointer in the output pointer table and to release the output data structure by clearing the related data available flag. The changed output data pointer should refer to a free data structure. The application should use the old data structure for further processing.

(c) Data section pointer method

A third option is to modify the data section pointer in the output data structure and to release the output data structure by clearing the related data available flag. The changed data section pointer should refer to a free memory area. The application should use the old data section for further processing by forwarding the data section pointer.

(6) Transfer function of FIFO message buffers

To use this buffer transfer function the output transfer has to be activated (see **Section 24.4.16.2, (1) Activation and deactivation**). The activation process requires the setup of the FIFO pointer table (see **Section 24.4.16.2, (4) FIFO output pointer table**) in order to specify a location in the Local RAM/Cluster RAM reserved for the storage of the required output data structures.

A FIFO data transfer is initiated when a valid FlexRay data frame has been stored in the FlexRay internal FIFO.

After transfers from the internal FIFO to the output data structure, the FIFO interrupt status flag (FLXAnFROTS.FIS) and FIFO data available bit (FLXAnFROTS.FDA) are set to 1. The bit FLXAnFROTS.FIS can be used as an interrupt source. The bit FLXAnFROTS.FDA indicates that the FIFO is not empty.

The size of the output structure is up to the maximum number of entries in the FIFO table (as set by the FLXAnFROTC.FTM[4:0] bits).

The transfer to the extended FIFO buffer structure is controlled by index pointers. This put index is controlled by the FIFO transfer handler and is incremented after transferring a message to the output data structure.

The FIFO reception handler also maintains a get index which is indicated in FLXAnFROTS.FGIDX[4:0]. The value of this get index is known by the application by either reading the status or maintaining a software variable. The get index (the value after a reset is 00000_B) is incremented by one when the application releases the oldest entry of the FIFO queue by writing 1 to FLXAnFROTS.FDA. By comparing the put index and the get index the FIFO handler knows about the current fill level of the queued buffer structure.

The current FIFO fill level is flagged in FLXAnFROTS.FFL[5:0]. When FLXAnFROTS.FDA is 1, there is at least one entry in the FIFO queue.

In case the queued buffer structure in the Local RAM/Cluster RAM is full (FLXAnFROTS.FFL[5:0] = FLXAnFROTC.FTM[4:0] + 1), no further transfers are initialized, new messages remain in the FlexRay internal FIFO and the FIFO transfer warning interrupt status flag (FLXAnFROTS.FWIS) is set to 1.

In case the FlexRay internal FIFO structure becomes full, messages in the FlexRay internal FIFO structure may be overwritten. The related status flags and configuration registers of the FlexRay core module can be used to generate desired warning notifications.

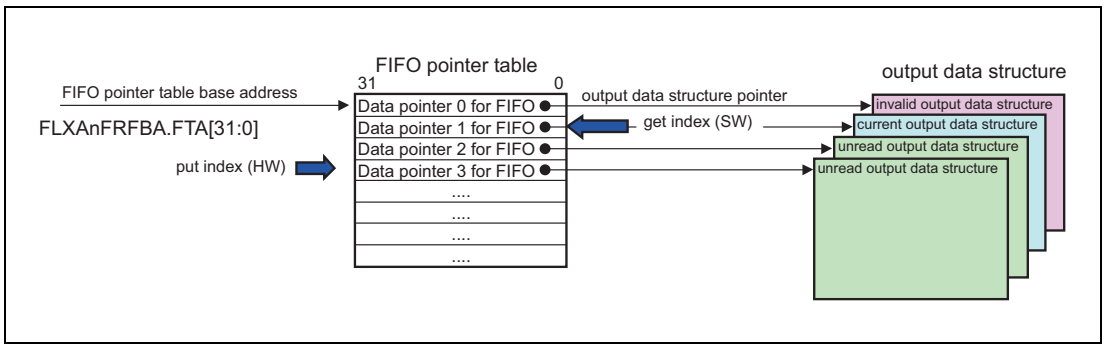


Figure 24.30 FIFO Pointer Table

(7) Transfer function of user output transfer requests

To use this transfer function the output transfer has to be activated (see **Section 24.4.16.2, (1) Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 24.4.16.2, (3) Output pointer table**) in order to specify the location in the Local RAM/ Cluster RAM reserved for the transfer of the data (output data structures).

The application is capable, by using FLXAnFRUOR.UMBNR[6:0], to request a transfer of dedicated message buffer to an output data structure. Except in CONFIG state, message buffers which are part of the FlexRay internal FIFO should not be requested.

The header section is always transferred to the output data structure. The transfer of the data section can be enabled by setting FLXAnFRUOR.URDS to 1. The selected message buffer content is stored in the output data structure location determined by the pointers in the output pointer table.

The data available status and transfer blocking by bits FLXAnFRDAm.DA[31:0] is also used for the user requested transfers. Therefore bits FLXAnFRDAm.DA[31:0] related to the requested buffer number (FLXAnFRUOR.UMBNR[6:0]) should be released before making the transfer request.

After writing to FLXAnFRUOR.UMBNR[6:0], the bit FLXAnFROTS.UORP is set to 1 to indicate that there is a pending user transfer request. When the transfer has been processed the bit FLXAnFROTS.UORP is set to 0, the bit FLXAnFROTS.OTIS is set to 1 and bits FLXAnFRDAm.DA[31:0] related to the requested buffer number (FLXAnFRUOR.UMBNR[6:0]) are set to 1.

User output transfer requests cannot be queued. The application should check the bit FLXAnFROTS.UORP before writing to FLXAnFRUOR.UMBNR[6:0].

User output transfer requests should not be made for message buffers which are pending in the input transfer queue.

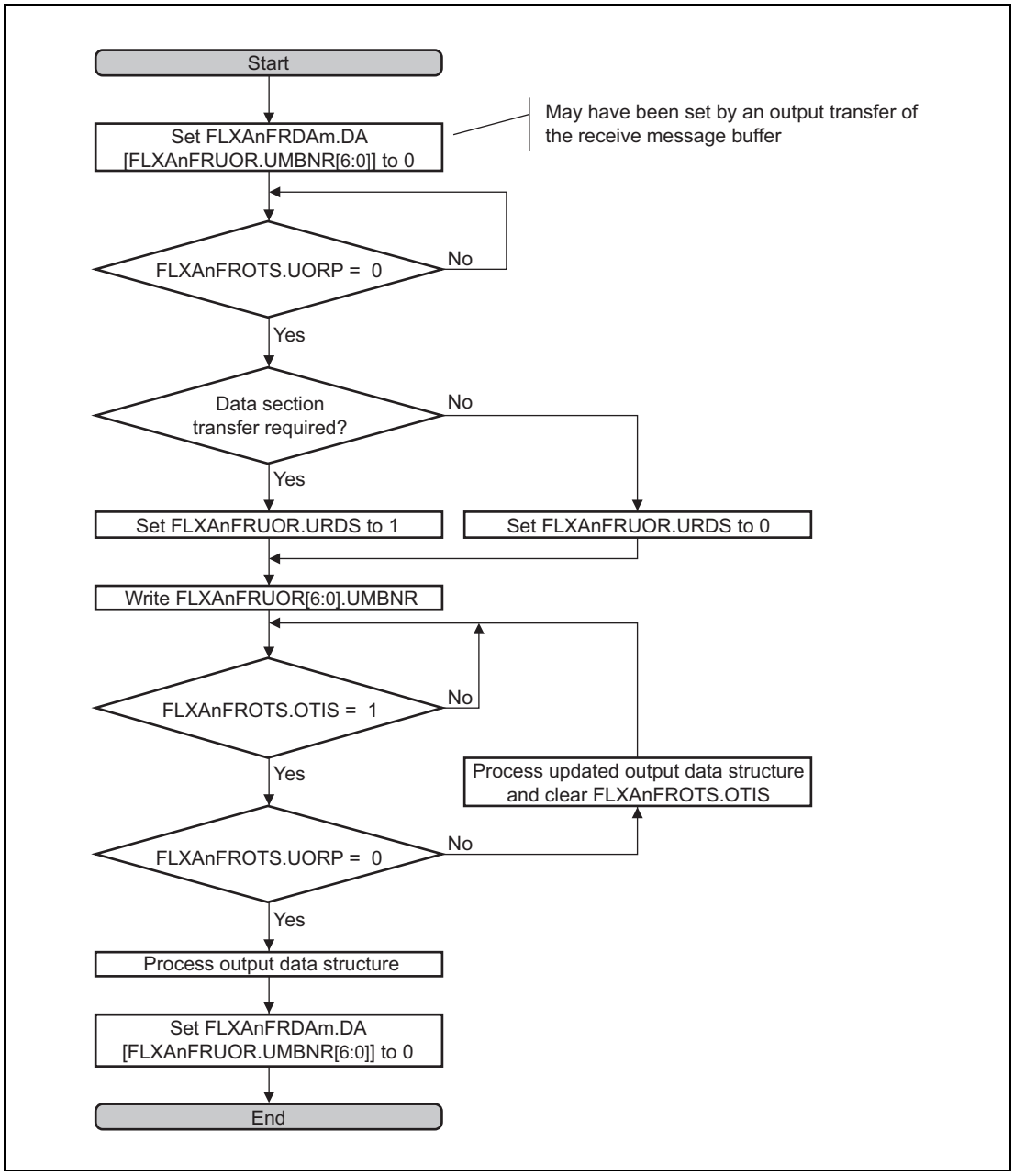


Figure 24.31 User Output Transfer Request Processing Flow

Note that it may be possible that the data structure addressed by a user request is being updated due to a receive message buffer update (which causes bits FLXAnFRDAm.DA[31:0] being set). This set FLXAnFRDAm.DA flags inhibits the user output transfer request. Therefore polling FLXAnFROTS.UORP is not a secure method to identify when the transfer of a requested message buffer has been completed. The bits FLXAnFROTS.OTIS or bits FLXAnFRDAm.DA[31:0] can be used instead. The exact flow depends on the software architecture.

24.4.16.3 Data Structure Transfer Scheduling

Cyclically the different types of transfer requests are checked. In order to guarantee a certain transfer time the different types of transfers have different priorities.

Use requested input transfers have highest priority followed by the transfer of data structures committed into the active input transfer queue. No new output transfer will be started as long as there is a pending input transfer request.

The three output transfer request types are checked in a specific order:

(1) All dedicated message buffers in ascending order

When FLXAnFROTC.OTCS is set to 0, set flags FLXAnFRNDATm.NDp are causing a transfer of the message buffer to the output data structure if the destination area is free (bits FLXAnFRDAm.DA[31:0] are 0).

When FLXAnFROTC.OTCS is set to 1, set flags FLXAnFRNDATm.NDp or set flags in the FLXAnFRMBSCm.MBC[31:0] register are causing a transfer of the message buffer to the output data structure if the destination area is free (bits FLXAnFRDAm.DA[31:0] are 0).

(2) FlexRay internal FIFO

When the FlexRay internal FIFO is not empty and there is a free destination area, one FIFO message is transferred into the output data structure specified by the FIFO pointer table.

(3) User output request

If there is a pending user output transfer request, one message buffer is transferred into the corresponding output data structure.

The check sequence is suspended when an input transfer occurs.

24.4.16.4 Behavior in Case of Data Transfer Access Error

The memory areas accessed by the data transfer function may be protected by a memory protection unit (MPU). When the MPU flags an access to a protected address caused by an input or output transfer, an access error event is generated and the related bit in the FLXAnFRAES register is set.

The ongoing transfer is immediately terminated but succeeding transfers are processed and may generate further access errors. Any following access errors are only flagged in FLXAnFRAES.MAE. The other status flags are not updated.

(1) Access error during input transfer

When an access error occurs during an input transfer:

- The ongoing transfer is immediately terminated. The FlexRay internal message RAM will not be updated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.IAE is set to 1
- The input pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- In case of an normal input transfer the to the transfer related bits FLXAnFRDAm.DA[31:0] is set to 0
- In case of a user input transfer request FLXAnFRITS.UIRP is set to 0

With the given status information the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the input access error flag (FLXAnFRAES.IAE).

(2) Access error during output transfer

When an access error occurs during an output transfer:

- The ongoing transfer is immediately terminated but the update of the data structure may have started.
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.OAE is set to 1
- The output pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- In case of an normal output transfer the to the transfer related bits FLXAnFRDAm.DA[31:0] remains 0 and no output transfer interrupt is generated
- In case of a user output transfer request FLXAnFROTS.UORP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. The data structure in the Local RAM/Cluster RAM cannot be treated as valid.

In addition the application needs to clear the output access error flag (FLXAnFRAES.OAE).

The FlexRay module internal transfer of the message buffer is completed before the Local RAM/ Cluster RAM access error is detected. The output transfer will not be re-initiated. To avoid loss of data, the application can perform a user output transfer request of this message buffer to a correct Local RAM/Cluster RAM location.

(3) Access error during FIFO transfer

When an access error occurs during an FIFO transfer:

- The ongoing transfer is immediately terminated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.FAE is set to 1
- The FIFO pointer table index is flagged in FLXAnFRAES.EIDX[7:0]
- The FIFO index pointer are not changed and hence the FIFO status flags are unchanged

With the given status information the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the FIFO access error flag (FLXAnFRAES.FAE).

The data in the Local RAM/Cluster RAM cannot be treated as valid and is not released to the application. The message cannot be recovered.

24.4.16.5 Behaviors in Case of RAM Read Errors

The FlexRay internal message RAM has an ECC checking mechanism. In case an uncorrectable RAM read error occurs, the application has to analyze the status in the FLXAnFRMHDS register and react as described in **Section 24.4.16.3, Data Structure Transfer Scheduling**. The input and output transfer handler reacts also on these errors detected in the message RAM when the error is related to an active transfer.

In addition, the TBFA and TBFB have an ECC checking mechanism as well. An uncorrectable RAM read errors does not impact the data transfer functionality but have to be handled as described in **Section 24.4.13.1, (4) Message Buffer Status FLXAnFRMBS (word 3)**.

In all cases, data causing a read error is never transferred to the Local RAM/Cluster RAM. If there is no recovery available in the application, the message is lost.

(1) Read error during transfer from TBF to MBF

This internal transfer is done for each valid FlexRay message received.

A read error can only occur when reading the header section in the FlexRay Message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

For dedicated receive message buffers, the related flags FRNDATm.ND[31:0] will not get set. Consequently the affected message buffer will not be transferred to the output data structure.

For the FlexRay internal FIFO buffers, flags FRNDATm.ND[31:0] are not set but the FlexRay internal FIFO put index is incremented. Due to this, a transfer procedure from the FlexRay internal FIFO buffer to the output buffer is started. However, if the read error is still present in the header section, updating of the output data structure will not start (see **Section 24.4.16.5, (2) Read error during transfer from MBF to OBF**); thus the data in the Local RAM/Cluster RAM remains correct.

Note that the correction or any other reconfiguration of FIFO related to message buffers while there are pending FIFO transfers may result in incorrect data in the Local RAM/Cluster RAM. It is strongly recommended to deactivate the output data transfer before starting the reconfiguration and flush the FlexRay internal FIFO before reactivation of the output data transfer.

(2) Read error during transfer from MBF to OBF

This internal transfer is done for every output data transfer (dedicated reception, FIFO, user requested).

A read error can occur in the header and data section (see read error flags in FLXAnFRMHDS). In both cases the message gets lost. If the error is located in the header section, the message buffer needs to be re-configured. If the error is located in the data section, the error is corrected with the next data section update.

When a read error occurs during the transfer from the message RAM to the output buffer, the output data structure will not be updated and the data available will not be set to 1. The FIFO put index and the FIFO fill level are not changed also.

In case of user output transfer request, FLXAnFROTS.UORP is set to 0 even if there was no update of the output data structure.

(3) Read error during transfer from IBF to MBF

This internal transfer is done for every input data transfer.

A read error can occur only when there is no update of the header section requested (the bit LHS in FLXAnFRWRHS4 is set to 0) due to the reading of the header section from the message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

When a read error occurs during the input data transfer, the actually transferred message in the input queue gets lost.

(4) Message RAM read errors

Read errors when reading the header section are flagged in the FLXAnFRMHDS register.

Depending on the buffer type and set buffer protection, a reconfiguration of the message buffer may not be possible.

The input transfer function cannot be used to reconfigure a locked message buffer using the method described in **Section 24.4.13.4, (3) Temporary Unlocking of Header Section**.

Before reconfiguring a locked buffer, the user should disable the input transfer function and the output transfer function.

24.4.17 Byte Alignment

The alignment of the bytes received by the FlexRay protocol and the alignment of the bytes required by the application may be different. The FlexRay module provides with FLXAnFROC.BEC a byte alignment function to support different byte ordering styles.

Figure 24.32 shows the payload byte alignment in a FlexRay frame.

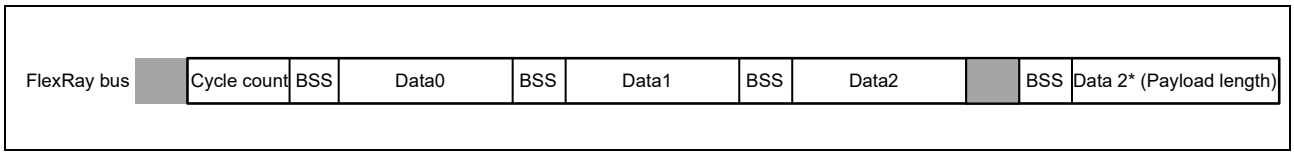


Figure 24.32 Byte Alignment on the FlexRay Bus

24.4.17.1 Little Endian Alignment

When FLXAnFROC.BEC is 0, the byte alignment is set to Little Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as follows.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXAn FRNMV1	Data 3								Data 2								Data 1								Data 0							
FLXAn FRNMV2	Data 7								Data 6								Data 5								Data 4							
FLXAn FRNMV3	Data 11								Data 10								Data 9								Data 8							

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as follows.

- FLXAnFRWRDSx.MD[7:0] = Data4x-4
- FLXAnFRWRDSx.MD[15:8] = Data4x-3
- FLXAnFRWRDSx.MD[23:16] = Data4x-2
- FLXAnFRWRDSx.MD[31:24] = Data4x-1

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[7:0], FLXAnFRWRDSx.MD [15:8], FLXAnFRWRDSx.MD [23:16], FLXAnFRWRDSx.MD [31:24] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure are as follows.

- FLXAnFRRDDSx.MD[7:0] = Data4x-4
- FLXAnFRRDDSx.MD[15:8] = Data4x-3
- FLXAnFRRDDSx.MD[23:16] = Data4x-2
- FLXAnFRRDDSx.MD[31:24] = Data4x-1

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[7:0], FLXAnFRRDDSx.MD[15:8], FLXAnFRRDDSx.MD[23:16], FLXAnFRRDDSx.MD[31:24] with the most significant bit (MSB) received first.

24.4.17.2 Big Endian Alignment

When FLXAnFROC.BEC is '1', the byte alignment is set to Big Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as follows.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLXAn FRNMV1	Data 0								Data 1								Data 2								Data 3							
FLXAn FRNMV2	Data 4								Data 5								Data 6								Data 7							
FLXAn FRNMV3	Data 8								Data 9								Data 10								Data 11							

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure are as follows.

- FLXAnFRWRDSx.MD[7:0] = Data4x-1
- FLXAnFRWRDSx.MD[15:8] = Data4x-2
- FLXAnFRWRDSx.MD[23:16] = Data4x-3
- FLXAnFRWRDSx.MD[31:24] = Data4x-4

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[31:24], FLXAnFRWRDSx.MD[23:16], FLXAnFRWRDSx.MD[15:8], and FLXAnFRWRDSx.MD[7:0] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure are as follows.

- FLXAnFRRDDSx.MD[7:0] = Data4x-1
- FLXAnFRRDDSx.MD[15:8] = Data4x-2
- FLXAnFRRDDSx.MD[23:16] = Data4x-3
- FLXAnFRRDDSx.MD[31:24] = Data4x-4

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[31:24], FLXAnFRRDDSx.MD [23:16], FLXAnFRRDDSx.MD[15:8], FLXAnFRRDDSx.MD[7:0] with the most significant bit (MSB) received first.

Section 25 Ethernet AVB (ETNB)

This section contains a generic description of the Ethernet AVB (ETNB).

The first part in this section describes properties specific to the RH850/U2A-EVA, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ETNB.

CAUTION

Read URAM in this manual as Local RAM/Global RAM/Retention RAM.

Read AXI BUS in this manual as AHB BUS.

25.1 Features of ETNB for RH850/U2A-EVA

25.1.1 Number of Units and Channels

This microcontroller has the following number of ETNB units.

Table 25.1 Number of Units (100 Mbps Ether)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	—	1
Name	ETNBn (n = 0)									

Table 25.2 Number of Units (1 Gbps Ether)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	—	—	—	—
Name	ETNBn (n = 1)									

Table 25.3 Index

Index	Description
n	Throughout this section, the individual ETNB units of Fast Ethernet and Gigabit Ethernet are identified by the index “n”. ETNB0 is for Fast Ethernet and ETNB1 is for Gigabit Ethernet.

25.1.2 Register Base Addresses

The ETNB base address is listed in the following table.

The ETNB register addresses are given as offsets from the base address.

Table 25.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ETNB0_base>	FF0A 2000_H	P-Bus Group 9
<ETNB1_base>	FF0A 4000_H	P-Bus Group 9

25.1.3 Clock Supply

The ETNB clock supply is shown in the following table.

Table 25.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
ETNBn	clk_axi	CLK HBUS
	clk_chi	CLK HBUS
	Register access clock	CLK HBUS

CAUTION

Minimum frequency of the clock clk_axi and clk_chi is:

- **62.5 MHz when E-MAC works in 1000 Mbps mode (ETNBnGECMR.SPEED = 1)**
- **12.5 MHz when E-MAC works in 100 Mbps mode (ETNBnGECMR.SPEED = 0)**

25.1.4 Interrupt Requests and Error Notifications

The ETNB interrupt requests are listed in the following table.

Table 25.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
ETNB0			
INTETNB0DATA	Data related interrupt	645	—
INTETNB0ERR	Error related interrupt	646	—
INTETNB0MNG	Management related interrupt	647	—
INTETNB0MAC	MAC interrupt	648	—
ETNB1			
INTETNB1DATA	Data related interrupt	649	—
INTETNB1ERR	Error related interrupt	650	—
INTETNB1MNG	Management related interrupt	651	—
INTETNB1MAC	MAC interrupt	652	—
INTETNB1STA	SGMII interrupt	653	—

This module has no error notifications.

25.1.5 Reset Sources

The ETNB reset sources are listed in the following table. ETNB is initialized by these reset sources.

Table 25.7 Reset Sources (RH850/U2A-EVA)

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ETNB0	All registers	√	√	√	√	√	√	—
ETNB1	ETNB1SGSDS ETNB1SGCLKSEL ETNB1SGRCIE	√	√	√	√	√	—	—
	Other than above	√	√	√	√	√	√*1	—

Note 1. When reset by module reset.
Need to wait module reset assertion until turning off the power of SGMII SerDes.
Power status can be checked by ETNB1SGSDS.PWS bit.

25.1.6 External Input/Output Signals

External input/output signals of ETNB are listed below.

Table 25.8 ETNBn Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal
ETNB0			
AVB_TX_CLK	I	MII transmit clock signal	ETNB0TXCLK
AVB_RX_CLK	I	MII receive clock signal	ETNB0RXCLK
AVB_TX_EN	O	MII/RMII transmit data enable signal	ETNB0TXEN
AVB_TXD[3:0](MII)/ AVB_TXD[1:0](RMII)	O	MII/RMII transmit data signal	ETNB0TXD[3:0](MII)/ ETNB0TXD[1:0](RMII)
AVB_TX_ER	O	MII transmit error signal	ETNB0TXER
AVB_RX_DV	I	MII receive data valid signal	ETNB0RXDV
AVB_RXD[3:0](MII)/ AVB_RXD[1:0](RMII)	I	MII/RMII receive data signal	ETNB0RXD[3:0](MII)/ ETNB0RXD[1:0](RMII)
AVB_RX_ER	I	MII/RMII receive error signal	ETNB0RXER
RMII_REF50CK	I	RMII reference clock signal	ETNB0REFCLK
RMII_CRS_DV	I	RMII receive data valid signal	ETNB0CRS_DV
AVB_MDC	O	PHY management clock signal	ETNB0MDC
AVB_MDIO	I/O	PHY management transfer data signal	ETNB0MDIO
AVB_LINK	I	PHY link status signal	ETNB0LINKSTA
AVB_MAGIC	O	Wake-On-LAN. This signal indicates that a Magic Packet™*1 has been received	ETNB0WOL
ETNB1			
TX_DATAN	O	SGMII Tx serial data outputs (negative)	TX_DATAN
TX_DATAP	O	SGMII Tx serial data outputs (positive)	TX_DATAP
RX_CLKN*2	I	SGMII Rx ddr clock inputs (negative)	RX_CLKN
RX_CLKP*2	I	SGMII Rx ddr clock inputs (positive)	RX_CLKP
RX_DATAN	I	SGMII Rx serial data inputs (negative)	RX_DATAN
RX_DATAP	I	SGMII Rx serial data inputs (positive)	RX_DATAP
REFCLK	I	SGMII Reference clock input (25 MHz)	ETNB1REFCLK
AVB_TX_CLK	I	MII transmit clock signal	ETNB1TXCLK
AVB_RX_CLK	I	MII receive clock signal	ETNB1RXCLK
AVB_TX_EN	O	MII transmit data enable signal	ETNB1TXEN
AVB_TXD[3:0]	O	MII transmit data signal	ETNB1TXD[3:0]
AVB_TX_ER	O	MII transmit error signal	ETNB1TXER
AVB_RX_DV	I	MII receive data valid signal	ETNB1RXDV
AVB_RXD[3:0]	I	MII receive data signal	ETNB1RXD[3:0]
AVB_RX_ER	I	MII receive error signal	ETNB1RXER
AVB_MDC	O	PHY management clock signal	ETNB1MDC
AVB_MDIO	I/O	PHY management transfer data signal	ETNB1MDIO
AVB_LINK*3	I	PHY link status signal	ETNB1LINKSTA
AVB_PHY_INT	I	PHY interrupt information	ETNB1PHYINT
AVB_MAGIC	O	Wake-On-LAN. This signal indicates that a Magic Packet™*1 has been received	ETNB1WOL

Note 1. Magic Packet™ is a trademark of Advanced Micro Devices, inc.

Note 2. Use for RH850/U2A-EVA (BGA-516) only.

Note 3. Supported by RH850/U2A-EVA (BGA-516) and RH850/U2A16 (BGA-516) only.

25.2 Overview

25.2.1 Functional Overview

Table 25.9 lists the specifications of the ETNB.

Table 25.9 Specifications of ETNBn

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
Communication interface*3	For ETNB0: • RMII (Reduced Media Independent Interface) • MII (Media Independent Interface) For ETNB1: • SGMII (Serial Gigabit Media Independent Interface) • MII*1 (Media Independent Interface)
Transfer speed	For ETNB0: • 100 Mbps • 10 Mbps For ETNB1: • 1000 Mbps • 100 Mbps • 10 Mbps
Transfer mode	Full-duplex mode
AVB function	• Conforming with the following standards stipulated for IEEE 802.1BA. – IEEE 802.1AS (time synchronization protocol) – IEEE 802.1Qav (real-time transfer) – IEEE 802.1Qat (stream reservation protocol) * IEEE 802.1Qat should be supported by software. • Descriptor management system • IEEE 1722 (AVTP presentation timestamp)
Transmit/Receive FIFO	For transmission: 16 Kbytes For reception: 8 Kbytes
Magic Packet™	Detection of Magic Packet™*2 and Wake-On-LAN signal output

Note 1. RH850/U2A16 (373 pins), RH850/U2A16 (292 pins), RH850/U2A8 (373 pins) and RH850/U2A8 (292 pins) do not support MII.

Note 2. Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

Note 3. For ETNB0 MII/RMII select and ETNB1 MII/SGMII select, refer to **Section 51.12.14, OPBT8 — Option Byte 8**.
For SGMII transmitter polarity inversion, SGMII receiver polarity inversion, SGMII output amplitude control and SGMII receiver termination selectors, refer to **Section 51.12.20, OPBT14 — Option Byte 14**.

25.2.2 Block Diagram

25.2.2.1 Block Diagram of ETNB0 (Fast Ethernet)

Figure 25.1 is a block diagram of the ETNB0.

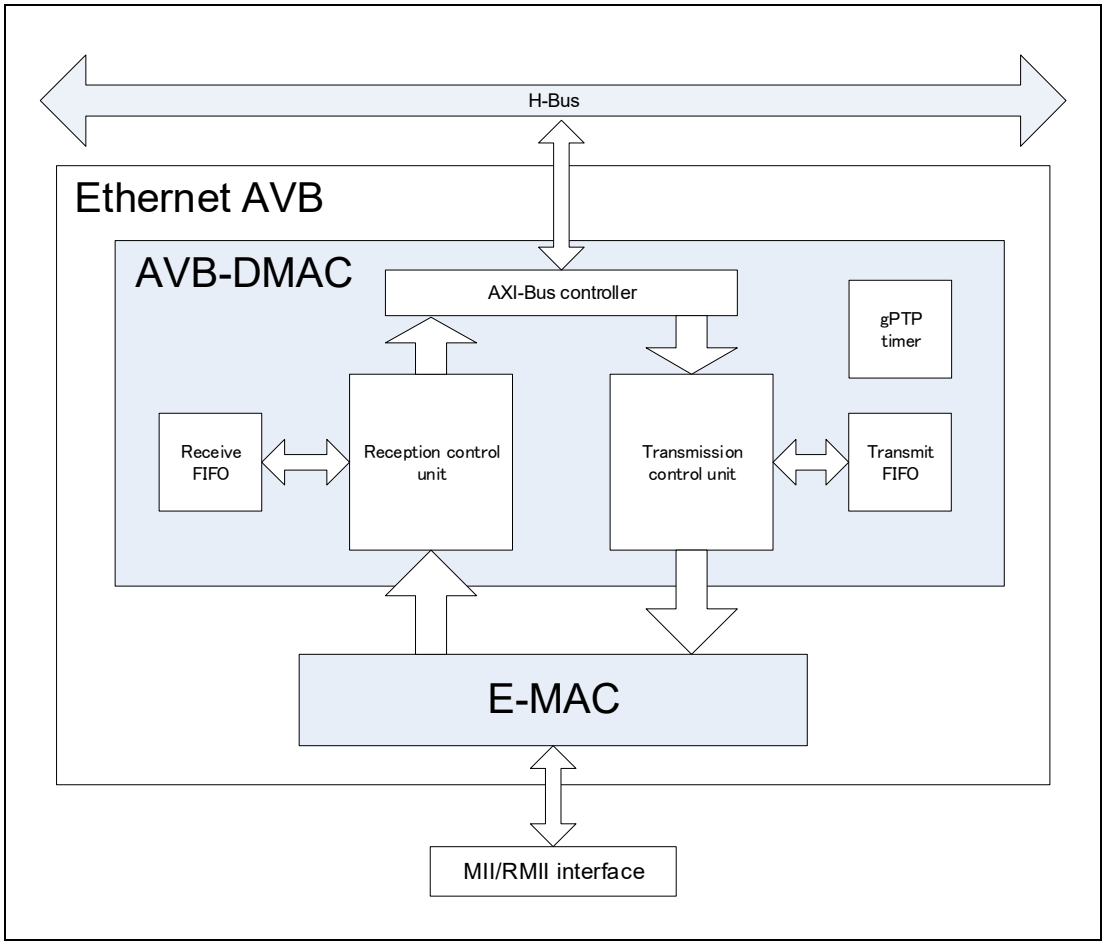


Figure 25.1 Block Diagram of ETNB0

25.2.2.2 Block Diagram of ETNB1 (Gigabit Ethernet)

Figure 25.2 is a block diagram of the ETNB1.

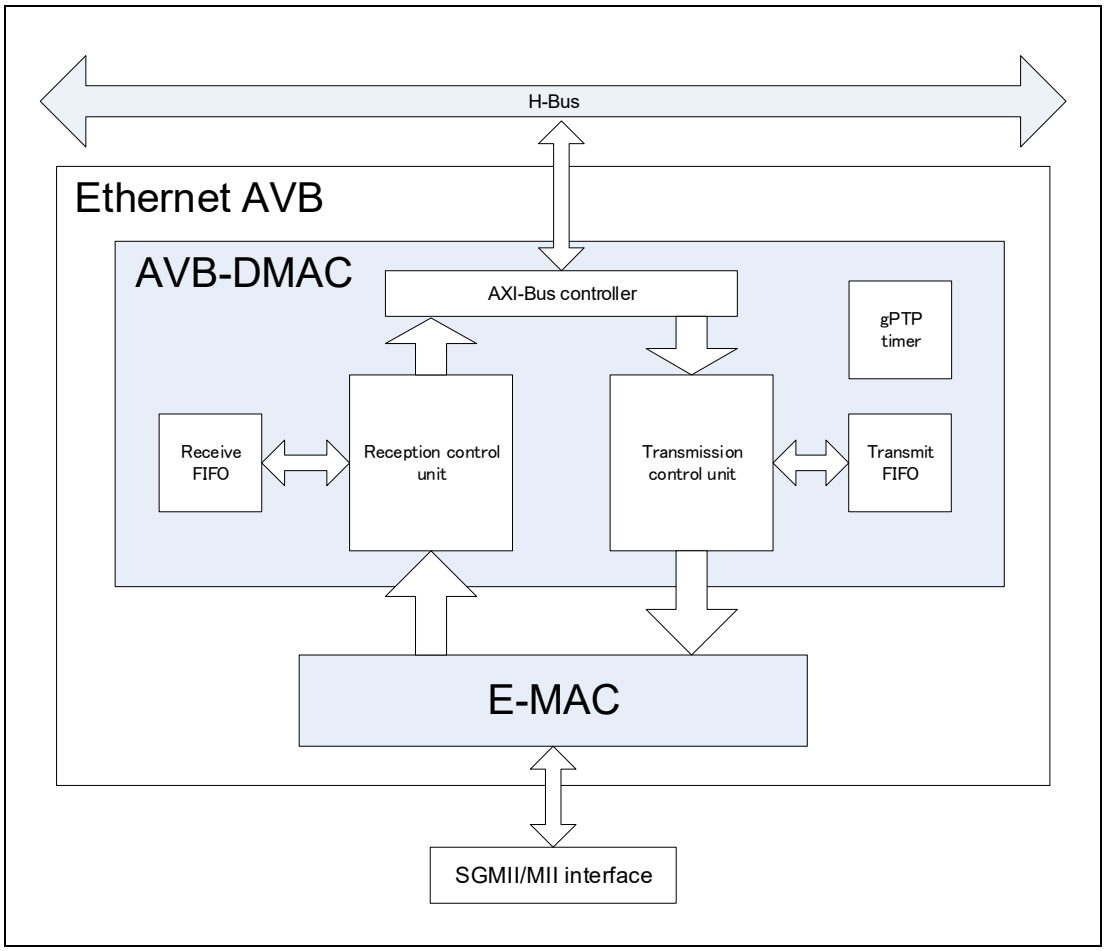


Figure 25.2 Block Diagram of ETNB1

25.3 Register Descriptions

25.3.1 List of Registers

The following table lists the ETNB registers.

For details about <ETNBn_base>, see **Section 25.1.2, Register Base Addresses**.

Table 25.10 List of Registers (1/3)

Unit name	Register Name	Abbreviation	Address	Access Protection	
				PBG	Other
Ethernet AVB Registers					
	AVB-DMAC mode register	ETNBnCCC	<ETNBn_base> + 0000_H	*1	—
	Descriptor base address table register	ETNBnDBAT	<ETNBn_base> + 0004_H	*1	—
	Descriptor base address load request register	ETNBnDLR	<ETNBn_base> + 0008_H	*1	—
	AVB-DMAC status register	ETNBnCSR	<ETNBn_base> + $000C_H$	*1	—
	Current descriptor address register q (q = 0 to 21)	ETNBnCDARq	<ETNBn_base> + 0010_H + q × 4_H	*1	—
	Error status register	ETNBnESR	<ETNBn_base> + 0088_H	*1	—
	RMII interface mode register	ETNBnRIMR	<ETNBn_base> + $008C_H$	*1	—
	Receive configuration register	ETNBnRCR	<ETNBn_base> + 0090_H	*1	—
	Receive queue configuration register i (i = 0 to 4)	ETNBnRQCi	<ETNBn_base> + 0094_H + i × 4_H	*1	—
	Receive padding configuration register	ETNBnRPC	<ETNBn_base> + $00B0_H$	*1	—
	Unread frame counter warning level register	ETNBnUFCW	<ETNBn_base> + $00BC_H$	*1	—
	Unread frame counter stop level configuration register	ETNBnUFCS	<ETNBn_base> + $00C0_H$	*1	—
	Unread frame counter register i (i = 0 to 4)	ETNBnUFCVi	<ETNBn_base> + $00C4_H$ + i × 4_H	*1	—
	Unread frame counter decrement register i (i = 0 to 4)	ETNBnUFCDi	<ETNBn_base> + $00E0_H$ + i × 4_H	*1	—
	Separation filter offset register	ETNBnSFO	<ETNBn_base> + $00FC_H$	*1	—
	Separation filter pattern register i (i = 0 to 31)	ETNBnSFPi	<ETNBn_base> + 0100_H + i × 4_H	*1	—
	Separation filter value register i (i = 0, 1)	ETNBnSFVi	<ETNBn_base> + $01B8_H$ + i × 4_H	*1	—
	Separation filter mask register i (i = 0, 1)	ETNBnSFMi	<ETNBn_base> + $01C0_H$ + i × 4_H	*1	—
	Separation filter load register	ETNBnSFL	<ETNBn_base> + $01C8_H$	*1	—
	Payload CRC register	ETNBnPCRC	<ETNBn_base> + $01CC_H$	*1	—
	Transmit configuration register	ETNBnTGC	<ETNBn_base> + 0300_H	*1	—
	Transmit configuration control register	ETNBnTCCR	<ETNBn_base> + 0304_H	*1	—
	Transmit status register	ETNBnTSR	<ETNBn_base> + 0308_H	*1	—
	Timestamp FIFO access register 0	ETNBnTFA0	<ETNBn_base> + 0310_H	*1	—
	Timestamp FIFO access register 1	ETNBnTFA1	<ETNBn_base> + 0314_H	*1	—
	Timestamp FIFO access register 2	ETNBnTFA2	<ETNBn_base> + 0318_H	*1	—
	Version and release register	ETNBnVRR	<ETNBn_base> + $031C_H$	*1	—
	CBS increment value register c (c = 0, 1)	ETNBnCIVRc	<ETNBn_base> + 0320_H + c × 4_H	*1	—
	CBS decrement value register c (c = 0, 1)	ETNBnCDVRc	<ETNBn_base> + 0328_H + c × 4_H	*1	—

Table 25.10 List of Registers (2/3)

Unit name	Register Name	Abbreviation	Address	Access Protection	
				PBG	Other
Ethernet AVB Registers					
	CBS upper limit register c (c = 0, 1)	ETNBnCULc	<ETNBn_base> + 0330_H + c × 4_H	*1	—
	CBS lower limit register c (c = 0, 1)	ETNBnCLLc	<ETNBn_base> + 0338_H + c × 4_H	*1	—
	Descriptor interrupt control register	ETNBnDIC	<ETNBn_base> + 0350_H	*1	—
	Descriptor interrupt status register	ETNBnDIS	<ETNBn_base> + 0354_H	*1	—
	Error interrupt control register	ETNBnEIC	<ETNBn_base> + 0358_H	*1	—
	Error interrupt status register	ETNBnEIS	<ETNBn_base> + $035C_H$	*1	—
	Receive interrupt control register 0	ETNBnRIC0	<ETNBn_base> + 0360_H	*1	—
	Receive interrupt status register 0	ETNBnRIS0	<ETNBn_base> + 0364_H	*1	—
	Receive interrupt control register 1	ETNBnRIC1	<ETNBn_base> + 0368_H	*1	—
	Receive interrupt status register 1	ETNBnRIS1	<ETNBn_base> + $036C_H$	*1	—
	Receive interrupt control register 2	ETNBnRIC2	<ETNBn_base> + 0370_H	*1	—
	Receive interrupt status register 2	ETNBnRIS2	<ETNBn_base> + 0374_H	*1	—
	Transmit interrupt control register	ETNBnTIC	<ETNBn_base> + 0378_H	*1	—
	Transmit interrupt status register	ETNBnTIS	<ETNBn_base> + $037C_H$	*1	—
	Interrupt summary status register	ETNBnISS	<ETNBn_base> + 0380_H	*1	—
	Common interrupt enable register	ETNBnCIE	<ETNBn_base> + 0384_H	*1	—
	Receive interrupt control register 3	ETNBnRIC3	<ETNBn_base> + 0388_H	*1	—
	Receive interrupt status register 3	ETNBnRIS3	<ETNBn_base> + $038C_H$	*1	—
	Receive interrupt enable register 0	ETNBnRIE0	<ETNBn_base> + 0460_H	*1	—
	Receive interrupt enable register 1	ETNBnRIE1	<ETNBn_base> + 0468_H	*1	—
	Receive interrupt enable register 2	ETNBnRIE2	<ETNBn_base> + 0470_H	*1	—
	Receive interrupt enable register 3	ETNBnRIE3	<ETNBn_base> + 0488_H	*1	—
	Receive interrupt disable register 0	ETNBnRID0	<ETNBn_base> + 0464_H	*1	—
	Receive interrupt disable register 1	ETNBnRID1	<ETNBn_base> + $046C_H$	*1	—
	Receive interrupt disable register 2	ETNBnRID2	<ETNBn_base> + 0474_H	*1	—
	Receive interrupt disable register 3	ETNBnRID3	<ETNBn_base> + $048C_H$	*1	—
	gPTP configuration control register	ETNBnGCCR	<ETNBn_base> + 0390_H	*1	—
	Transmit interrupt enable register	ETNBnTIE	<ETNBn_base> + 0478_H	*1	—
	Transmit interrupt disable register	ETNBnTID	<ETNBn_base> + $047C_H$	*1	—
	gPTP maximum transit time register	ETNBnGMTT	<ETNBn_base> + 0394_H	*1	—
	gPTP presentation time comparison register	ETNBnGPTC	<ETNBn_base> + 0398_H	*1	—
	gPTP timer increment register	ETNBnGTI	<ETNBn_base> + $039C_H$	*1	—
	gPTP timer offset register i (i = 0 to 2)	ETNBnGTOi	<ETNBn_base> + $03A0_H$ + i × 4_H	*1	—
	gPTP interrupt control register	ETNBnGIC	<ETNBn_base> + $03AC_H$	*1	—
	gPTP interrupt status register	ETNBnGIS	<ETNBn_base> + $03B0_H$	*1	—
	gPTP timer capture register i (i = 0 to 2)	ETNBnGCTi	<ETNBn_base> + $03B8_H$ + i × 4_H	*1	—
	gPTP status register	ETNBnGSR	<ETNBn_base> + $03C4_H$	*1	—
	gPTP presentation time FIFO register i (i = 0 to 3)	ETNBnGPTFi	<ETNBn_base> + $03E0_H$ + i × 4_H	*1	—
	E-MAC mode register	ETNBnECMR	<ETNBn_base> + 0500_H	*1	—

Table 25.10 List of Registers (3/3)

Unit name	Register Name	Abbreviation	Address	Access Protection	
				PBG	Other
Ethernet AVB Registers					
	Receive frame length register	ETNBnRFLR	<ETNBn_base> + 0508_H	*1	—
	E-MAC status register	ETNBnECSR	<ETNBn_base> + 0510_H	*1	—
	E-MAC interrupt permission register	ETNBnECSIPR	<ETNBn_base> + 0518_H	*1	—
	PHY interface register	ETNBnPIR	<ETNBn_base> + 0520_H	*1	—
	PHY LINK status register	ETNBnPLSR	<ETNBn_base> + 0528_H	*1	—
	PHY interrupt polarity register	ETNBnPIPR	<ETNBn_base> + $052C_H$	*1	—
	Auto PAUSE frame time parameter register	ETNBnAPFTP	<ETNBn_base> + 0554_H	*1	—
	Manual PAUSE frame register	ETNBnMPR	<ETNBn_base> + 0558_H	*1	—
	PAUSE frame transmit counter	ETNBnPFTCR	<ETNBn_base> + $055C_H$	*1	—
	PAUSE frame receive counter	ETNBnPFRCR	<ETNBn_base> + 0560_H	*1	—
	Automatic PAUSE frame retransmit count register	ETNBnTPAUSER	<ETNBn_base> + 0564_H	*1	—
	PAUSE frame transmit times counter	ETNBnPFTTCR	<ETNBn_base> + 0568_H	*1	—
	E-MAC mode register 2	ETNBnGECMR	<ETNBn_base> + $05B0_H$	*1	—
	MAC address high register	ETNBnMAHR	<ETNBn_base> + $05C0_H$	*1	—
	MAC address low register	ETNBnMALR	<ETNBn_base> + $05C8_H$	*1	—
	CRC error frame receive counter register	ETNBnCEFCR	<ETNBn_base> + 0740_H	*1	—
	Frame receive error counter register	ETNBnFRECR	<ETNBn_base> + 0748_H	*1	—
	Too-short frame receive counter register	ETNBnTSFRCR	<ETNBn_base> + 0750_H	*1	—
	Too-long frame receive counter register	ETNBnTLFRCR	<ETNBn_base> + 0758_H	*1	—
	Residual-bit frame receive counter register	ETNBnRFCR	<ETNBn_base> + 0760_H	*1	—
	Multicast address frame receive counter register	ETNBnMAFCR	<ETNBn_base> + 0778_H	*1	—
SGMII Interface Related Registers					
	SGMII operation mode configuration register	ETNB1SGOPMC	<ETNB1_base> + 1000_H	PBG90#7	—
	SGMII operation mode status register	ETNB1SGOPMS	<ETNB1_base> + 1004_H	PBG90#7	—
	SGMII software reset register	ETNB1SGSRST	<ETNB1_base> + 1008_H	PBG90#7	—
	SGMII interrupt status register	ETNB1SGINTS	<ETNB1_base> + $100C_H$	PBG90#7	—
	SGMII interrupt mask register	ETNB1SGINTM	<ETNB1_base> + 1010_H	PBG90#7	—
	SGMII link timer value configuration register	ETNB1SGLTVC	<ETNB1_base> + 1014_H	PBG90#7	—
	SGMII code error counter register	ETNB1SGCECT	<ETNB1_base> + 1018_H	PBG90#7	—
	SGMII running disparity error counter register	ETNB1SGRECT	<ETNB1_base> + $101C_H$	PBG90#7	—
	SGMII SerDes status register	ETNB1SGSDS	<ETNB1_base> + 1800_H	PBG90#7	—
	SGMII reference clock select register	ETNB1SGCLKSEL	<ETNB1_base> + 1804_H	PBG90#7	—
	SGMII reference clock input enable register	ETNB1SGRCIE	<ETNB1_base> + 1808_H	PBG90#7	—

Note 1. n = 0: PBG90#6
n = 1: PBG90#7

25.3.2 Ethernet AVB Registers

25.3.2.1 ETNBnCCC — AVB-DMAC Mode Register

The ETNBnCCC register specifies the operating mode of the AVB-DMAC.

Access: ETNBnCCC can be read or written in 32-bit units.
ETNBnCCCL and ETBnCCCH can be read or written in 16-bit units.
ETNBnCCCLL, ETBnCCCLH, ETNBnCCCHL, ETNBnCCCHH can be read or written in 8-bit units.

Address: ETNBnCCC: <ETNBn_base> + 0000_H
ETNBnCCCL: <ETNBn_base> + 0000_H
ETNBnCCCH: <ETNBn_base> + 0002_H
ETNBnCCCLL: <ETNBn_base> + 0000_H
ETNBnCCCLH: <ETNBn_base> + 0001_H
ETNBnCCCHL: <ETNBn_base> + 0002_H
ETNBnCCCHH: <ETNBn_base> + 0003_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	FCE	LBME	—	—	—	BOC	—	—	CSEL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTSR	GAC	—	—	—	—	—	OPC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W

Table 25.11 ETNBnCCC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	FCE	Flow Control Enable 0: Flow control disabled 1: Flow control enabled
24	LBME	Loopback Mode Enable 0: Normal operation 1: Loopback mode is enabled.
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	BOC	Byte Order Configuration 0: First Ethernet byte in URAM[7:0] 1: First Ethernet byte in URAM[31:24]
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	CSEL[1:0]	gPTP Clock Select 00_B: gPTP is not in use. 01_B: Peripheral bus clock (clk_chi) 10_B: Ethernet transmission clock 11_B: Setting prohibited
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	DTSR	Data Transfer Suspend Request 0: Normal operation 1: Requests suspension

Table 25.11 ETNBnCCC Register Contents (2/2)

Bit Position	Bit Name	Function
7	GAC	gPTP Active in Config 0: Normal operation 1: gPTP support active in configuration mode
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	OPC[1:0]	Operating Mode Configuration 00_B: Reset mode 01_B: Configuration mode 10_B: Operation mode 11_B: Standby mode

FCE: Flow Control Enable Bit

This bit enables the flow control support of MAC.

When flow control is enabled, the MAC gets informed about the Rx-FIFO level (Rx-FIFO fill level reached ETNBnRCR.RFCL).

LBME: Loopback Mode Enable Bit

This bit enables loopback mode.

In loopback mode, the transmission lines are internally connected to the reception lines. When loopback mode is to be used, the Ethernet transmission clock or the reference clock must be supplied to the MII/RMII/SGMII interface. A received clock signal is not required for the MII interface. Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

Data for transmission are still output normally. To eliminate effects on external modules, pin control should be applied to block the output of data. For details about the pin control, see Section 2, Pin Functions.

BOC: Byte Order Configuration Bit

This bit specifies the assignment of the first byte of received Ethernet frames when it is allocated to the URAM.

This setting of this register does not affect the descriptor format and filter parameters of the URAM.

Writing to this bit is only possible when the current operating mode is configuration mode.

The CPU can only write 0 to this bit.

Figure 25.3 and **Figure 25.4** show how data from frames received via the Ethernet connection are stored in the URAM.

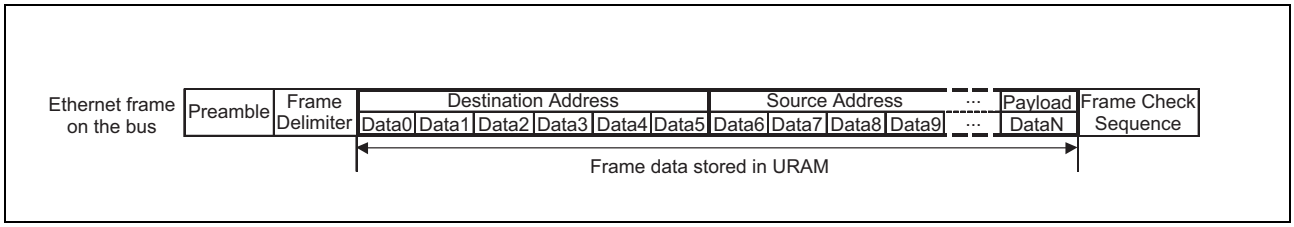


Figure 25.3 Data of Ethernet Frame Received

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Data3								Data2								Data1								Data0							
DPTR+4	Data7								Data6								Data5								Data4							
DPTR+8	Data11								Data10								Data9								Data8							

Figure 25.4 When ETNBnCCC.BOC = 0

CSEL[1:0]: gPTP Clock Select Bits

These bits select the clock source for the gPTP timer.
Writing to these bits is only possible when the current operating mode is configuration mode and ETNBnCCC.GAC is 0, or when the current operating mode is reset mode and writing 1 to ETNBnCCC.GAC and writing 01_B to ETNBnCCC.OPC by same write access.

DTSR: Data Transfer Suspend Request Bit

This bit can suspend access to the URAM.

The access is suspended on completion of the transfer of the frame currently being transferred.

This function disables access to the URAM without affecting normal operation of the AVB-DMAC. Use this bit when exclusive control over the contents of the URAM is necessary, for example, in checking its integrity.

CAUTION

The transmission and reception queues are not processed while access is suspended. Change neither the AVB-DMAC settings nor the mode until the suspend request is given by ETNBnCCC.DTSR and later the value of ETNBnCSR.DTS is updated.

GAC: gPTP Active in Config Bit

These bit enables the gPTP support of the AVB-DMAC in configuration mode. Function of gPTP support in operation mode and standby mode is not influenced by this bit. When gPTP support is active in configuration mode, ETNBnCCC.CSEL defines the timer clock source.

Writing 1 to this bit is only possible when the current operating mode is reset mode and 01_B is written to ETNBnCCC.OPC by the same write access.

OPC[1:0]: Operating Mode Configuration Bits

These bits specify the operating mode.

For the operating modes, see **Section 25.4.1.1, Operating Modes**.

Writing to this bit is possible in any of the operating modes, but should not be done after the application system has issued a Power Off request.

Do not write 11_B to these bits when ETNBnCCC.GAC is 1.

CAUTION

In RMII mode, after entering reset mode by writing 00_B to OPC[1:0], it is necessary to assert Module Reset by SWMRESA_ETNB.SWMRESA_ETNB0 to initialize RMII circuit. This caution just applies for U2A-EVA.

25.3.2.2 ETNBnDBAT — Descriptor Base Address Table Register

The ETNBnDBAT register is used to set the base address of the descriptor table.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.12 ETNBnDBAT Register Contents

Bit Position	Bit Name	Function
31 to 0	TA[31:0]	Descriptor Base Table Address Base address of the descriptor table in the URAM

CAUTION

The setting of this bit must be a multiple of four (Bit 0 and bit 1 must be set to “0”).

TA[31:0]: Descriptor Base Table Address Bits

These bits specify the base address of the descriptor table in the URAM.

For the structure of the descriptor base address table, see **Section 25.4.3, Descriptors**.

Writing to this bit is only possible when the current operating mode is configuration mode.

25.3.2.3 ETNBnDLR — Descriptor Base Address Load Request Register

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0008_H

Value after reset: $003F\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	LBA21	LBA20	LBA19	LBA18	LBA17	LBA16
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBA15	LBA14	LBA13	LBA12	LBA11	LBA10	LBA9	LBA8	LBA7	LBA6	LBA5	LBA4	LBA3	LBA2	LBA1	LBA0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.13 ETNBnDLR Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	LBA21	Base Address Load Request (Rx17: Stream 15) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
20	LBA20	Base Address Load Request (Rx16: Stream 14) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
19	LBA19	Base Address Load Request (Rx15: Stream 13) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
18	LBA18	Base Address Load Request (Rx14: Stream 12) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
17	LBA17	Base Address Load Request (Rx13: Stream 11) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
16	LBA16	Base Address Load Request (Rx12: Stream 10) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
15	LBA15	Base Address Load Request (Rx11: Stream 9) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 25.13 ETNBnDLR Register Contents (2/3)

Bit Position	Bit Name	Function
14	LBA14	Base Address Load Request (Rx10: Stream 8) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
13	LBA13	Base Address Load Request (Rx9: Stream 7) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
12	LBA12	Base Address Load Request (Rx8: Stream 6) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
11	LBA11	Base Address Load Request (Rx7: Stream 5) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
10	LBA10	Base Address Load Request (Rx6: Stream 4) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
9	LBA9	Base Address Load Request (Rx5: Stream 3) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
8	LBA8	Base Address Load Request (Rx4: Stream 2) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
7	LBA7	Base Address Load Request (Rx3: Stream 1) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
6	LBA6	Base Address Load Request (Rx2: Stream 0) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
5	LBA5	Base Address Load Request (Rx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
4	LBA4	Base Address Load Request (Rx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 25.13 ETNBnDLR Register Contents (3/3)

Bit Position	Bit Name	Function
3	LBA3	Base Address Load Request (Tx3: Stream Class A) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
2	LBA2	Base Address Load Request (Tx2: Stream Class B) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
1	LBA1	Base Address Load Request (Tx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
0	LBA0	Base Address Load Request (Tx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

LBAq (q = 0 to 21): Base Address Load Request Bits

Each bit is used to issue requests to load base addresses and to indicate that a base address is currently being loaded.

Setting this bit to 1 issues a request for loading the descriptor base address for the queue q.

If transfer is currently in progress, loading is executed on completion of transfer for the current frame.

Completion of loading leads to automatic setting of the corresponding bit to 0.

For transmission queues, base address load requests are executed even while fetching is in progress (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is 1). Therefore, be sure to check that fetching is not in progress before issuing a request.

Writing to a bit of this register is only possible when the current operating mode is operation mode.

Only 1 can be written to this bit.

25.3.2.4 ETNBnCSR — AVB-DMAC Status Register

The ETNBnCSR register is used to indicate the operating mode in which the AVB-DMAC is running and communications states.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $000C_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	RPO	TPO3	TPO2	TPO1	TPO0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DTS	—	—	—	—	OPS[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.14 ETNBnCSR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20	RPO	Receive Process Status 0: Normal operation 1: Reception is in progress.
19	TPO3	Transmit Process Status 3 (Stream Class A) 0: Normal operation 1: Transmission is in progress.
18	TPO2	Transmit Process Status 2 (Stream Class B) 0: Normal operation 1: Transmission is in progress.
17	TPO1	Transmit Process Status 1 (Network Control) 0: Normal operation 1: Transmission is in progress.
16	TPO0	Transmit Process Status 0 (Best Effort) 0: Normal operation 1: Transmission is in progress.
15 to 9	Reserved	When read, the value after reset is returned.
8	DTS	Data Transmission Suspended Status 0: Normal operation 1: Transmission is suspended.
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	OPS[3:0]	Operating Mode Status 0001_B: Reset mode 0010_B: Configuration mode 0100_B: Operation mode 1000_B: Standby mode Other settings are prohibited.

RPO: Receive Process Status Bit

This bit indicates whether a reception queue contains an unread received frame.

This bit being set to 1 indicates that a received frame is yet to be stored in the URAM.

- [Clearing conditions]
  - The current operating mode changes from operation mode.
  - Received frames in the reception FIFO all being stored in the URAM.
- [Setting condition]
  - A received frame being stored in the reception FIFO (but not yet in the URAM)

TPO3: Transmit Process Status 3 Bit

This bit indicates whether a class A stream is being transmitted.

When this bit is set to 1, it indicates that the AVB-DMAC is fetching data for transmission from the URAM or E-MAC is transmitting data.

- [Clearing conditions]
  - The current operating mode changes from operation mode.
  - Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ3) is set to 0)
- [Setting condition]
  - Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ3))

TPO2: Transmit Process Status 2 Bit

This bit indicates whether a class B stream is being transmitted.

When this bit is set to 1, it indicates that the AVB-DMAC is fetching data for transmission from the URAM or E-MAC is transmitting data.

- [Clearing conditions]
  - The current operating mode changes from operation mode.
  - Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ2) is set to 0)
- [Setting condition]
  - Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ2))

TPO1: Transmit Process Status 1 Bit

This bit indicates whether a network control is being transmitted.

When this bit is set to 1, the AVB-DMAC is fetching data in the URAM or E-MAC is transmitting data.

- [Clearing conditions]
  - The current operating mode changes from operation mode.
  - Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ1) is set to 0)
- [Setting condition]
  - Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ1))

TPO0: Transmit Process Status 0 Bit

This bit indicates whether a best effort is being transmitted.

When this bit is set to 1, the AVB-DMAC is fetching data in the URAM or E-MAC is transmitting data.

- [Clearing conditions]
  - The current operating mode changes from operation mode.
  - Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ0) is set to 0)
- [Setting condition]
  - Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQ0))

DTS: Data Transmission Suspend Status Bit

This bit indicates whether access to the URAM is enabled.

- [Clearing condition]
  - The current operating mode is not operation mode.
  - The data transmission suspend request bit in the AVB-DMAC mode register (ETNBnCCC.DTSR) being 0.
- [Setting condition]
  - Access to the URAM not proceeding while the data transmission suspend request bit (ETNBnCCC.DTSR) in the AVB-DMAC mode register is 1 (if the URAM is being accessed, this bit is set to 1 on completion of access).

OPS[3:0]: Operating Mode Status Bits

These bits indicate the current operating mode.

For the operating modes, see **Section 25.4.1.1, Operating Modes**.

25.3.2.5 ETNBnCDARq — Current Descriptor Address Register q (q = 0 to 21)

The ETNBnCDARq register indicates the current descriptor address.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0010_H + q × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.15 ETNBnCDARq Register Contents

Bit Position	Bit Name	Function
31 to 0	CDA[31:0]	Current Descriptor Address The address of the current descriptors for the transmission/receive queues q

CDA[31:0]: Current Descriptor Address Bits

ETNBnCDAR0 to ETNBnCDAR3 indicate the addresses of the current descriptors for the corresponding transmission queues while ETNBnCDAR4 to ETNBnCDAR21 indicate the addresses of the current descriptors for the corresponding reception queues.

If the operating mode is changed to operation mode, contents of this bits are changed to (ETNBnDBAT + q × 8).

Also, when the descriptor base address load request register (ETNBnDLR) issues a load request, the contents of the corresponding register are set in the descriptor base address table register (ETNBnDBAT).

Conditions for updating:

These bits are set to 0 when the operating mode is not operation mode.

This register is updated when a descriptor to a queue is processed.

25.3.2.6 ETNBnESR — Error Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0088_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	EIL	ET[3:0]				—	—	—	EQN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.16 ETNBnESR Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12	EIL	Error Information Lost 0: No loss of error information 1: Lost of error information detected
11 to 8	ET[3:0]	Error Type 0000_B: Read descriptor from URAM 0001_B: Write descriptor to URAM 0010_B: Interpret read descriptor 0011_B: Tx-Buffer is corrupted 0100_B: Read data from URAM 0101_B: Write data or timestamp to URAM 0110_B: Reading from received FIFO 0111_B: Received FIFO is corrupted 1000_B: Frame size error during reception detected 1001_B: Frame size error during transmission detected 1010_B: Tx-Buffer overflow 1011_B: AVTP FIFO error
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	EQN[4:0]	Error Queue Number

EIL: Error Information Lost Bit

This bit indicates that error information detected by Ethernet AVB is lost because the previous reported error has not been processed by CPU.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 0 when CPU writes 0 to the queue error flag (ETNBnEIS.QEF) of the error interrupt status register.
- This bit is set to 1 when the set condition of the queue error flag (ETNBnEIS.QEF) is fulfilled while the queue error flag (ETNBnEIS.QEF) is 1.

ET[3:0]: Error Type Bits

These bits indicate details about the transfer stage which was handled when Ethernet AVB has detected an error.

When a fault relates to the read descriptor (ETNBnESR.ET = 0000_B or 0010_B), the CPU needs to correct the faulty descriptor so that the related queues can continue processing. If a queue halts at a faulty descriptor, ETNBnCDARq.CDA (q = ETNBnESR.EQN) directly checks the faulty descriptor.

When the fault is related to descriptor writing (ETNBnESR.ET = 0001_B), CPU needs recognize the not-updated or incorrectly updated descriptor in queue ETNBnESR.EQN. The write problem is not influencing how Ethernet AVB processes the descriptor chain.

When the fault is related to the Tx-Buffer (ETNBnESR.ET = 0011_B) CPU needs to clean-up the Tx-Buffer to correct the buffer control configuration.

All other errors are transient in nature and may be corrected by continuation of HW or SW operation; so there is no strong demand on CPU interaction. For details of the error processing, **Section 25.4.2.3, Checking Integrity**.

The CPU can only evaluate these bits when the queue error flag (ETNBnEIS.QEF) is 1.

[Changing condition]

– These bits are updated when the queue error flag (ETNBnEIS.QEF) setting conditions are satisfied and ETNBnEIS.QEF is set to 0.

EQN[4:0]: Error Queue Number Bits

These bits indicate the queue number which was handled when Ethernet AVB has detected an error.

A fault reported for ETNBnESR.EQN = 0 to 3 is related to transmit queue t = 0 to 3.
From ETNBnESR.EQN = 4 the fault is related to receive queue r = ETNBnESR.EQN - 4.

The CPU can only evaluate these bits when the queue error flag (ETNBnEIS.QEF) is 1.

The CPU can not evaluate these bits when the error type bit (ETNBnESR.ET) of the error status register is 0011_B, 0111_B, or 1011_B.

[Changing condition]

– These bits are updated when the set condition of the queue error flag (ETNBnEIS.QEF) is fulfilled and the queue error flag (ETNBnEIS.QEF) is 0.

25.3.2.7 ETNBnRIMR - RMII Interface Mode Register

The ETNBnRIMR register specifies the operating mode for the RMII interface.
This register is only valid in RMII mode.
The setting in this register must not be changed while transmission or reception is enabled.

Access: This register can be read or written in 32-bit units.
Address: <ETNBn_base> + $008C_H$
Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.17 ETNBnRIMR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SPEED	RMII Transfer Rate 0: 10 Mbps 1: 100 Mbps

25.3.2.8 ETNBnRCR — Receive Configuration Register

The ETNBnRCR register is used to configure receive-relating settings of the AVB-DMAC.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0090_H

Value after reset: $1800\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RFCL[12:0]												
Value after reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ETS2	ETS0	ESF[1:0]		ENCF	EFFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.18 ETNBnRCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	RFCL[12:0]	Receive FIFO Warning Level Recommended value: 1800_H
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	ETS2	Timestamp Enable (Stream) 0: Timestamp is disabled. 1: Timestamp is enabled. Recommended value: 0
4	ETS0	Timestamp Enable (Best Effort) 0: Timestamp is disabled. 1: Timestamp is enabled. Recommended value: 0
3, 2	ESF[1:0]	Stream Filtering Select Settings for reception queues 2 to 17 00_B: Filtering is disabled. Frames are processed in queue 0 (best effort). 01_B: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching stream frames are processed in queue 0 (best effort). 10_B: The separation filter is enabled for AVB stream frames. Non-matching stream frames are discarded. 11_B: The separation filter is enabled for AVB stream frames. Non-matching stream frames are handled with queue 0 (best effort). Recommended value: 10 or 11
1	ENCF	Network Control Filtering Enable Setting for reception queue 1 (network control) 0: Network control is disabled. 1: Network control is enabled.
0	EFFS	Error Frame Enable 0: Error frames are disabled. 1: Error frames are enabled. Recommended value: 0

RFCL[12:0]: Receive FIFO Warning Level Bits

These bits set the caution level for the reception FIFO and are used to maintain the priority order of the storage of received data and the fetching of data for transmission.

If the reception FIFO contains less data than this level, processing of both transmission and reception queues becomes pending.

If the reception FIFO contains more data than this level, only data in the reception queue are transferred, and processing of the transmission queue becomes pending.

Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

In the case of this LSI chip, set these bits to 1800_H.

ETS2: Timestamp Enable (Stream) Bit

This bit enables timestamp information included in reception queues 2 to 17.

Writing to this bit is only possible when the current operating mode is configuration mode.

Set this bit to 1 when use the extended descriptor.

ETS0: Timestamp Enable (Best Effort) Bit

This bit enables timestamp information included in reception queue 0.

Writing to this bit is only possible when the current operating mode is configuration mode.

Set this bit to 1 when use the extended descriptor.

ESF[1:0]: Stream Filtering Select Bits

These bits select separation filtering to reception queues 2 to 17.

The queue-dependent separation filter can be used in combination with the identification of AVB stream frames.

When the value is 00_B, filtering is disabled and frames from streams are processed in reception queue 0 (best effort).

When the value is 01_B, the separation filter is enabled for both AVB stream frames and non-AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

When the value is 10_B, the separation filter is enabled for AVB stream frames; frames from non-matching streams are discarded.

When the value is 11_B, the separation filter is enabled for AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

For separation filtering, see **Section 25.4.4.1(1), Separation Filtering**.

Writing to this bit is only possible when the current operating mode is configuration mode.

ENCF: Enable Network Control Filtering Bit

Enables the AVB network control frame for reception queue 1.

When reception queue 1 is disabled, a received frame is stored in reception queue 0 (best effort).

Writing to this bit is only possible when the current operating mode is configuration mode.

EFFS: Enable Error Frame Bit

Enables or disables the reception of frames that have been classified as error frames by the E-MAC.

Received error frames are stored in reception queue 0 (best effort).

An indicator of error detection by the E-MAC during reception is stored in the descriptor (DESCR.MSC).

Writing to this bit is only possible when the current operating mode is configuration mode.

25.3.2.9 ETNBnRQCi — Receive Queue Configuration Register i (i = 0 to 4)

The ETNBnRQCi register configures the settings of receive queue (r = 0+i × 4 to 3+i × 4).*1

Access: ETNBnRQCi can be read or written in 32-bit units.
ETNBnRQCiL and ETNBnRQCiH can be read or written in 16-bit units.*2
ETNBnRQCiLL, ETNBnRQCiLH, ETNBnRQCiHL, ETNBnRQCiHH can be read or written in 8-bit units.*2

Address: ETNBnRQCi: <ETNBn_base> + 0094_H + i × 4_H
ETNBnRQCiL: <ETNBn_base> + 0094_H + i × 4_H
ETNBnRQCiH: <ETNBn_base> + 0094_H + i × 4_H + 2_H
ETNBnRQCiLL: <ETNBn_base> + 0094_H + i × 4_H
ETNBnRQCiLH: <ETNBn_base> + 0094_H + i × 4_H + 1_H
ETNBnRQCiHL: <ETNBn_base> + 0094_H + i × 4_H + 2_H
ETNBnRQCiHH: <ETNBn_base> + 0094_H + i × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PIAr (r = 3 + i × 4)*3	UFCCr[1:0] (r = 3 + i × 4)*3		—	—	RSMr[1:0] (r = 3 + i × 4)*3		—	PIAr (r = 2 + i × 4)*3	UFCCr[1:0] (r = 2 + i × 4)*3		—	—	RSMr[1:0] (r = 2 + i × 4)*3	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIAr (r = 1 + i × 4)*3	UFCCr[1:0] (r = 1 + i × 4)*3		—	—	RSMr[1:0] (r = 1 + i × 4)*3		—	PIAr (r = 0 + i × 4)*3	UFCCr[1:0] (r = 0 + i × 4)*3		—	—	RSMr[1:0] (r = 0 + i × 4)*3	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Note 1. The ETNBnRQC4 register corresponds to reception queues from 16 and 17.

Note 2. The ETNBnRQC4 register corresponds to ETNBnRQC4, ETNBnRQC4L, ETNBnRQC4LL, and ETNBnRQC4LH only.

Note 3. The ETNBnRQC4 register corresponds to RSM16[1:0], UFCC16[1:0], RSM17[1:0], and UFCC17[1:0] only and bits from 16 to 31 are reserved bits.

Table 25.19 ETNBnRQCi Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	PIAr	Packed Incremental Data Area (Receive Queue 3 + i × 4) This bit defines how addresses inside the incremental data area of reception queue 3 + i × 4 are handled.
29, 28	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 3 + i × 4) These bits set the unread frame counter used in reception queue 3 + i × 4.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 3 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	PIAr	Packed Incremental Data Area (Receive Queue 2 + i × 4) This bit defines how addresses inside the incremental data area of reception queue 2 + i × 4 are handled.
21, 20	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 2 + i × 4) These bits set the unread frame counter used in reception queue 2 + i × 4.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 25.19 ETNBnRQCi Register Contents (2/2)

Bit Position	Bit Name	Function
17, 16	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 2 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	PIAr	Packed Incremental Data Area (Receive Queue 1 + i × 4) This bit defines how addresses inside the incremental data area of reception queue 1 + i × 4 are handled.
13, 12	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 1 + i × 4) These bits set the unread frame counter used in reception queue 1 + i × 4.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 1 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIAr	Packed Incremental Data Area (Receive Queue 0 + i × 4) This bit defines how addresses inside the incremental data area of reception queue 0 + i × 4 are handled.
5, 4	UFCCr[1:0]	Unread Frame Counter Configuration (Receive Queue 0 + i × 4) These bits set the unread frame counter used in reception queue 0 + i × 4.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	RSMr[1:0]	Receive Synchronous Mode (Receive Queue 0 + i × 4) 00_B: Mode with write-back Other than 00_B: Setting prohibited

PIAr (r = 0 to 17): Packed Incremental Data Area Bit

This bit defines how addresses inside the incremental data area of queue r are handled.
When this bit is 0, frame data in incremental data area starts always at a 32 bit address. Depending on received frame length there are up to 3 undefined bytes between two frames in URAM.
When this bit is 1, there will no gaps generated between two frames in incremental data area.
Writing to the bit is only possible when the current operating mode is configuration mode.

UFCCr[1:0] (r = 0 to 17): Unread Frame Counter Configuration Bits

These bits set the unread frame counter for reception queue r.
With the AVB-DMAC, four patterns of settings are available for the unread frame counter. Use the unread frame counter stop level configuration register (ETNBnUFCS) to set the warning level and stop level of the unread frame counter.
Set the pattern number (0 to 3) set in the unread frame counter stop level configuration register (ETNBnUFCS) in these bits.
When the value is 00_B, the stop function is disabled.
Writing to the bits is only possible when the current operating mode is configuration mode.

RSMr[1:0] (r = 0 to 17): Receive Synchronous Mode Bits

These bits set receive synchronous mode.
Set 00_B in these bits.
For receive synchronous mode, see **Section 25.4.4.3(3), Mode with Write-Back**.
Writing to the bits is only possible when the current operating mode is configuration mode.

25.3.2.10 ETNBnRPC — Receive Padding Configuration Register

The ETNBnRPC register is used to set padding for received frames.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $00B0_H$

Value after reset: $0000\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DCNT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 25.20 ETNBnRPC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	DCNT[7:0]	Stored Data Counter These bits specify the amount of data to be stored with the descriptor. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	PCNT[2:0]	Stored Padding Counter These bits indicate the amount of padding to be stored in data areas for descriptors. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

Padding can be used to extend frame lengths, but frame lengths should not exceed 4 Kbytes.

DCNT[7:0]: Stored Data Counter Bits

These bits specify the amount of the frame data (1 to 255) to be stored following the padding. Counting by one indicates one word (4 bytes). For example, when these bits are set to 47, the amount of data is 47 words (= 188 bytes).

When these bits are 0, all received data is stored following the initial padding.

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see **Section 25.4.4.3(2)(c), Padding**.

PCNT[2:0]: Stored Padding Counter Bits

These bits specify the amount of padding (1 to 7) to be appended to the URAM. Counting by one indicates one word (4 bytes). For example, when these bits are set to 1, the amount of padding is one word (= 4 bytes).

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see **Section 25.4.4.3(2)(c), Padding**.

25.3.2.11 ETNBnUFCW — Unread Frame Counter Warning Level Register

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $00BC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WL3[5:0]						—	—	WL2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	WL1[5:0]						—	—	WL0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.21 ETNBnUFCW Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	WL3[5:0]	Warning Level 3 Number of unread frames to reach warning level.
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	WL2[5:0]	Warning Level 2 Number of unread frames to reach warning level.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	WL1[5:0]	Warning Level 1 Number of unread frames to reach warning level.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	WL0[5:0]	Warning Level 0 Number of unread frames to reach warning level.

WLj[5:0]: Warning Level j (j = 0 to 3)

These bits define a warning level which can be used by the unread frame counters of receive queues. Each receive queue can select one of the 4 common warning level configurations.

When these bits are 0, in configuration j the warning level function is disabled.

The CPU can only write to these bits if ETNBnCSR.OPS is CONFIG.

The CPU should not write 63 to these bits.

25.3.2.12 ETNBnUFCS — Unread Frame Counter Stop Level Configuration Register

The ETNBnUFCS register sets the stop levels for unread frames.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $00C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SL3[5:0]						—	—	SL2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SL1[5:0]						—	—	SL0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 25.22 ETNBnUFCS Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	SL3[5:0]	Stop Level 3 Unread frame count stop level 3
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	SL2[5:0]	Stop Level 2 Unread frame count stop level 2
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	SL1[5:0]	Stop Level 1 Unread frame count stop level 1
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	SL0[5:0]	Stop Level 0 Unread frame count stop level 0 The write value should be 0.

SL0 to SL3[5:0]: Stop Level 0 to 3 Bits

These bits set the stop levels for unread frames.

One of the four stop levels from 0 to 3 can be set for each reception queue. When these bits are set to 0, the stop function is disabled. The level to be used is specified by the receive queue configuration register i (ETNBnRQCi) (i = 0 to 4).

Writing to the bits is only possible when the current operating mode is configuration mode.

25.3.2.13 ETNBnUFCVi — Unread Frame Counter Register i (i = 0 to 4)

The ETNBnUFCVi register indicates the number of unread frames in reception queues r (r=0 + i × 4 to 3 + i × 4).*1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $00C4_H$ + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CVr[5:0](r = 3 + i × 4)*2						—	—	CVr[5:0](r = 2 + i × 4)*2					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CVr[5:0](r = 1 + i × 4)*2						—	—	CVr[5:0](r = 0 + i × 4)*2					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The ETNBnUFCV4 register corresponds to reception queues from 16 to 17.

Note 2. The ETNBnUFCV4 register corresponds to CV16[5:0] and CV17[5:0] only and bits from 16 to 31 are reserved bits.

Table 25.23 ETNBnUFCVi Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 24	CVr[5:0]	Unread Frame Count 3 + 4 × i Number of unread frames in reception queue 3 + 4 × i
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	CVr[5:0]	Unread Frame Count 2 + 4 × i Number of unread frames in reception queue 2 + 4 × i
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	CVr[5:0]	Unread Frame Count 1 + 4 × i Number of unread frames in reception queue 1 + 4 × i
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	CVr[5:0]	Unread Frame Count 0 + 4 × i Number of unread frames in reception queue 0 + 4 × i

CVr[5:0] (r = 0 to 17): Unread Frame Count r Bits

These bits indicate the number of unread frames in reception queue r.

The number of unread frames is decremented by the value that is written to the unread frame counter decrement register i (ETNBnUFCDi).

For a description of how to use unread frames, refer to **Section 25.4.4.4, Unread Frame Counters**.

Conditions for updating:

The bits are set to 0 when the operating mode is not operation mode and when the descriptor base address load request register (ETNBnDLR) issues a base address load request.

The number is incremented when data received in reception queue r are stored normally. The maximum increment is $3F_H$. If the value exceeds $3F_H$, incrementation will not proceed.)

The number is decremented by the value written to the unread frame counter decrement register i (ETNBnUFCDi).

25.3.2.14 ETNBnUFCDi — Unread Frame Counter Decrement Register i (i = 0 to 4)

The ETNBnUFCDi register is used to decrement unread counters of reception queues r (r = 0 + i × 4 to 3 + i × 4).*1

Access: ETNBnUFCDi can be read or written in 32-bit units.
ETNBnUFCDiL and ETNBnUFCDiH can be read or written in 16-bit units.*2
ETNBnUFCDiLL, ETNBnUFCDiLH, ETNBnUFCDiHL, ETNBnUFCDiHH can be read or written in 8-bit units.*2

Address: ETNBnUFCDi: <ETNBn_base> + $00E0_H$ + i × 4_H
ETNBnUFCDiL: <ETNBn_base> + $00E0_H$ + i × 4_H
ETNBnUFCDiH: <ETNBn_base> + $00E0_H$ + i × 4_H + 2_H
ETNBnUFCDiLL: <ETNBn_base> + $00E0_H$ + i × 4_H
ETNBnUFCDiLH: <ETNBn_base> + $00E0_H$ + i × 4_H + 1_H
ETNBnUFCDiHL: <ETNBn_base> + $00E0_H$ + i × 4_H + 2_H
ETNBnUFCDiHH: <ETNBn_base> + $00E0_H$ + i × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	DVr[5:0] (r = 3 + i × 4) *3						—	—	DVr[5:0] (r = 2 + i × 4)*3					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DVr[5:0] (r = 1 + i × 4)*3						—	—	DVr[5:0] (r = 0 + i × 4)*3					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The ETNBnUFCD4 register corresponds to reception queues from 16 to 17.

Note 2. The ETNBnUFCD4 register corresponds to ETNBnUFCD4, ETNBnUFCD4L, ETNBnUFCD4LL, and ETNBnUFCD4LH only.

Note 3. The ETNBnUFCD4 register corresponds to DV16[5:0] and DV17[5:0] only and bits from 16 to 31 are reserved bits.

Table 25.24 ETNBnUFCDi Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	DVr[5:0]	Unread Frame Decrement Value 3 + 4 × i Unread frame decrement value for reception queue 3 + 4 × i
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	DVr[5:0]	Unread Frame Decrement Value 2 + 4 × i Unread frame decrement value for reception queue 2 + 4 × i
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	DVr[5:0]	Unread Frame Decrement Value 1 + 4 × i Unread frame decrement value for reception queue 1 + 4 × i
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	DVr[5:0]	Unread Frame Decrement Value 0 + 4 × i Unread frame decrement value for reception queue 0 + 4 × i

DVr[5:0] (r = 0 to 17): Unread Frame Decrement Value r Bits

These bits set the decrement value for unread frames in reception queue r. The value of an unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) is decremented by the value set in the corresponding bits of this register.

Write $3F_H$ to these bits to reset the unread counters in reception queue r.

These bits are always read as 0.

25.3.2.15 ETNBnSFO — Separation Filter Offset Register

The ETNBnSFO register sets an offset into frames for use by the separation filter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $00FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FBP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.25 ETNBnSFO Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FBP[5:0]	First Byte Position Position of the first byte of Ethernet frames in the separation filter

FBP[5:0]: First Byte Position Bits

These bits set the position of the first byte of Ethernet frames in the separation filter.

When these bits are 0, the separation filter starts from the top of each Ethernet frame (first byte of the destination address). For bytes in Ethernet frames, see **Figure 25.3, Data of Ethernet Frame Received**, in **Section 25.3.2, Ethernet AVB Registers**.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 25.4.4.1(1), Separation Filtering**.

[Condition for Changing]

These bits are updated to the value of ETNBnSFVi.LV[5:0] when ETNBnSFL.LC changes from 30 (Load ETNBnSFO.FBP) to 31 while the current operating mode is operation mode.

CAUTION

Received frames having fewer bytes than the setting of these bits + 8 bytes do not match the separation filter. In this case, the data will either be sorted into a reception queue or discarded in accord with the setting of the separation filtering select bits in the receive configuration register (ETNBnRCR.ESF).

25.3.2.16 ETNBnSFPi — Separation Filter Pattern Register i (i = 0 to 31)

The ETNBnSFPi register configures the pattern of the separation filter used for reception queues 2 to 17.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0100_H + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FPs[31:16] *2(s = reception queue r - 2)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPs[15:0]*1 (s = reception queue r - 2)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. In this case, i is an even number. When i is an odd number, FPs[47:32]

Note 2. In this case, i is an even number. When i is an odd number, FPs[63:48]

Table 25.26 ETNBnSFPi Register Contents

Bit Position	Bit Name	Function
31 to 0	FPs[63:0]	Separation Filter Pattern These bits set the pattern of the separation filter. The 64-bit filter pattern is set for each queue.

FPs[63:0] (s = 0 to 15): Separation Filter Pattern Bits

These bits set the pattern for a separation filter to be used with reception queues 2 to 17 (for streams 0 to 15).

Each queue has space of 64-bit. Reception queue 2 (Stream 0) uses ETNBnSFP0 and ETNBnSFP1. Reception queue 17 (Stream 15) uses ETNBnSFP30 and ETNBnSFP31.

The separation filter passes a frame when, after masking by the mask value set in the separation filter mask register (ETNBnSFMi), data from received frames match the value defined in these bits.

ETNBnSFPi.FPs[7:0] (where i is an even number) are used for the byte of Ethernet frame data specified by the separation filter offset register, while ETNBnSFPi.FPs[63:56] (where i is the odd number) are used for the byte at the address specified by the separation filter offset register (ETNBnSFO) + 7.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 25.4.4.1(1), Separation Filtering**.

[Condition for Changing]

These bits are updated to the value of ETNBnSFVi.LV[63:0] when ETNBnSFL.LC changes from s (Load ETNBnSFPi.FPs) to 31 while the current operating mode is operation mode.

25.3.2.17 ETNBnSFVi — Separation Filter Value register i (i = 0 or 1)

The ETNBnSFVi register sets the 64-bit value to be loaded for separation filter registers.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $01B8_H$ + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.27 ETNBnSFVi Register Contents

Bit Position	Bit Name	Function
31 to 0	LV[31:0]	Load Value Value used to update separation filtering during operation mode

LV[31:0]: Load Value Bits

These bits set the 64-bit value to be loaded into separation filter registers (ETNBnSFO.FBP, ETNBnSFMi.CFM, ETNBnSFPi.FPs). The value is loaded when writing to ETNBnSFL.LC. When loading the First Byte Position (ETNBnSFO.FBP), ETNBnSFVi.LV[63:6] are ignored. Writing to the bit is only possible when the current operating mode is operation mode, and the current value of ETNBnSFL.LC is 31.

These bits are set to 0 when leaving operation mode.

25.3.2.18 ETNBnSFMi — Separation Filter Mask Register i (i = 0 or 1)

The ETNBnSFMi register sets the mask value of the separation filter used for reception queue 2 to 17.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $01C0_H$ + i × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFM[31 + 32 × i: 16 + 32 × i]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFM[15 + 32 × i: 0 + 32 × i]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.28 ETNBnSFMi Register Contents

Bit Position	Bit Name	Function
31 to 0	CFM[63:0]	Separation Filter Mask These bits set the mask value for the separation filter.

CFM[63:0]: Separation Filter Mask Bits

These bits set the mask value for the separation filter for use with the corresponding reception queue 2 to 17 (stream 0 to 15).

ETNBnSFM0.CFM[7:0] are used for bytes of Ethernet frame data specified by the separation filter offset register, while ETNBnSFM1.CFM[63:56] are used for the separation filter offset register (ETNBnSFO) + 7.

Frame data at the positions of mask bits that are set to 0 are masked; that is, they do not affect pattern-matching by the separation filter.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 25.4.4.1(1), Separation Filtering**.

[Condition for Changing]

These bits are updated to the value of ETNBnSFVi.LV[63:0] when ETNBnSFL.LC from 29 (Load ETNBnSFMi.CFM) to 31 while the current operating mode is operation mode.

25.3.2.19 ETNBnSFL — Separation Filter Load Register

The ETNBnSFL register controls update of separation filter configuration while the operating mode is operation mode.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $01C8_H$

Value after reset: $0000\ 001F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	LC[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 25.29 ETNBnSFL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	LC[4:0]	Load Command 0 to 15: Load of ETNBnSFPi.FPs with s = ETNBnSFL.LC 16 to 28: Invalid 29: Load of ETNBnSFMi.CFM 30: Load of ETNBnSFO.FBP 31: No pending load request

LC[4:0]: Load Command Bits

These bits control update of separation filter configuration while the operating mode is operation mode.
By writing 30 to these bits CPU triggers an update of First Byte Position (ETNBnSFO.FBP) value.
By writing 29 to these bits CPU triggers an update of Common Filter Mask (ETNBnSFMi.CFM) value.
By writing number between 0 to 15 to these bits CPU triggers an update of Filter Pattern s (ETNBnSFPi.FPs) value. The number written to ETNBnSFL.LC defines the index s of the updated Filter Pattern.

Writing to the bit is only possible when the current operating mode is operation mode, and the current value of ETNBnSFL.LC is 31.
These bits are set to 31 when the requested update has been processed, or when leaving operation mode.

25.3.2.20 ETNBnPCRC — Payload CRC Register

The ETNBnPCRC register is used to set the number of the first data byte in received frame to check a potential payload CRC.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $01CC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CAS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.30 ETNBnPCRC Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	CAS[9:0]	Crc Area Start Number of 1st byte included in payload CRC calculation

CAS[9:0]: CRC Area Start Bits

These bits are used to set the number of the first data byte in received frame to check a potential payload CRC. For details, see **Section 25.4.4.5, Payload CRC support**.
Writing to the bit is only possible when the current operating mode is operation mode.
The value of CAS must be in the range 0 to 255.
These bits are set to 0 when leaving operation mode.

25.3.2.21 ETNBnTGC — Transmit Configuration Register

The ETNBnTGC register configures the transmission-relating settings of the AVB-DMAC.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0300_H

Value after reset: $0022\ 2200_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	TBD3[1:0]		—	—	TBD2[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TBD1[1:0]		—	—	TBD0[1:0]		—	—	TQP[1:0]		TSM3	TSM2	TSM1	TSM0
Value after reset	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.31 ETNBnTGC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	TBD3[1:0]	Transmit FIFO Size (Stream Class A) Number of frames to be fetched from transmission queue 3 (for stream class A) **CAUTION** Write 10_B to these bits.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	TBD2[1:0]	Transmit FIFO Size (Stream Class B) Number of frames to be fetched from transmission queue 2 (for stream class B) **CAUTION** Write 10_B to these bits.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	TBD1[1:0]	Transmit FIFO Size (Network Control) Number of frames to be fetched from transmission queue 1 (for network control) **CAUTION** Write 10_B to these bits.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TBD0[1:0]	Transmit FIFO Size (Best Effort) Number of frames to be fetched from transmission queue 0 (for best effort) **CAUTION** Write 10_B to these bits.

Table 25.31 ETNBnTGC Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	TQP[1:0]	Transmit Queue Priority 00: Non-AVB mode 01: AVB mode 1 10: Setting prohibited 11: AVB mode 2
3	TSM3	Transmit Synchronous Mode (Stream Class A) 0: With write-back 1: Setting prohibited
2	TSM2	Transmit Synchronous Mode (Stream Class B) 0: With write-back 1: Setting prohibited
1	TSM1	Transmit Synchronous Mode (Network Control) 0: With write-back 1: Setting prohibited
0	TSM0	Transmit Synchronous Mode (Best Effort) 0: With write-back 1: Setting prohibited

TBDt[1:0] (t = 0 to 3):
Transmit FIFO Size (Stream Class A/ Stream Class B/Network Control/Best Effort) Bits

These bits set the sizes of the transmission FIFO buffers for use with each of the transmission queues.

Writing to these bits is only possible when the current operating mode is configuration mode.

Set these bits to 2 ("10_B").

TQP[1:0]: Transmit Queue Priority Bits

These bits set the priority of the transmission queues.

00_B: Non-AVB mode: Q3→Q2→Q1→Q0

01_B: AVB mode 1: Q3 (CBS)→Q2 (CBS)→Q1→Q0

10_B: Setting prohibited

11_B: AVB mode 2: Q1→Q3 (CBS)→Q2 (CBS)→Q0

For the credit-based shaping (CBS) algorithm, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

The CBS algorithm is invalidated in non-AVB mode (i.e. when the value is "00_B").

Writing to the bits is only possible when the current operating mode is configuration mode.

TSM0 to TSM3: Transmit Synchronous Mode Bits

These bits set Transmit synchronous mode.

Writing to these bits is only possible when the current operating mode is configuration mode.

Set these bits to 0.

25.3.2.22 ETNBnTCCR — Transmit Configuration Control Register

The ETNBnTCCR register controls transmission by the AVB-DMAC and is used to make related settings.

Access: ETNBnTCCR can be read or written in 32-bit units.
ETNBnTCCRL can be read or written in 16-bit units.
ETNBnTCCRLL, ETNBnTCCRLH can be read or written in 8-bit units.

Address: ETNBnTCCR: <ETNBn_base> + 0304_H
ETNBnTCCRL: <ETNBn_base> + 0304_H
ETNBnTCCRLL: <ETNBn_base> + 0304_H
ETNBnTCCRLH: <ETNBn_base> + 0304_H + 1_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MFR	MFEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFR	TFEN	—	—	—	—	TSRQ3	TSRQ2	TSRQ1	TSRQ0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.32 ETNBnTCCR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MFR	MAC status FIFO Release 0: No request to MAC status FIFO 1: Release oldest entry of MAC status FIFO
16	MFEN	MAC status FIFO Enable 0: Disabled 1: Enabled
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TFR	Timestamp FIFO Release 0: (Not operating) 1: Releases the oldest entry in the timestamp FIFO.
8	TFEN	Timestamp FIFO Enable 0: Recording of transmission timestamps in the timestamp FIFO is disabled. 1: Recording of transmission timestamps in the timestamp FIFO is enabled.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	TSRQ3	Transmit Start Request (Queue 3 (Stream Class A)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.
2	TSRQ2	Transmit Start Request (Queue 2 (Stream Class B)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.
1	TSRQ1	Transmit Start Request (Queue 1 (Network Control)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.

Table 25.32 ETNBnTCCR Register Contents

Bit Position	Bit Name	Function
0	TSRQ0	Transmit Start Request (Queue 0 (Best Effort)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Wait for transmission process to fetch is performed.

TFR: Timestamp FIFO Release Bit

This bit makes the timestamp FIFO release the oldest entry.

For a description of how to use the timestamp FIFO, see **Section 25.4.5.4, Timestamping in Transmission**.

TFEN: Timestamp FIFO Enable Bit

This bit enables storage in the timestamp FIFO.

When it is set, timestamp information is stored for descriptors with DESCR.TSR set to 1 (for DESCR.TSR, see **Section 25.4.5.2(2), Configuration of Transmission Frame Data Descriptors**).

When 0 is set in this bit, all timestamp FIFO entries are invalidated.

For a description of how to use the timestamp FIFO, see **Section 25.4.5.4, Timestamping in Transmission**.

TSRQt (t = 0 to 3): Transmit Start Request (Queue t) Bit

This bit issues a request to start transmission for transmission queue t.

When read, this bit being set to 1 indicates that transmission queue t has a frame that has not yet been fetched to the transmission FIFO.

Frame transmission by the E-MAC is processed independently from fetching to the transmission FIFO. The timing of transmission from a queue depends on the priority order of transmission.

For the scheduling of transmission queues, see **Section 25.4.5.1, Transmission Modes**.

Writing to this bit is only possible when the current operating mode is Operation mode.

Only 1 can be written to the bit. Writing 0 to the bit has no effect.

Conditions for updating:

The bit is set to 0 when the operating mode is not operation mode, when a descriptor of type EEMPTY, FEMPTY or LEMPTY (no usable data) is processed, when an EOS descriptor is processed, or when a descriptor with defective data is processed.

25.3.2.23 ETNBnTSR — Transmit Status Register

The ETNBnTSR register indicates the state of transmission by the AVB-DMAC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0308_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MFFL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFFL[2:0]			—	—	—	—	CCS1[1:0]		CCS0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.33 ETNBnTSR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 16	MFFL[4:0]	MAC status FIFO Fill Level Number of entries stored in the MAC status FIFO.
15 to 11	Reserved	When read, the value after reset is returned.
10 to 8	TFFL[2:0]	Timestamp FIFO Count Number of timestamp FIFOs
7 to 4	Reserved	When read, the value after reset is returned.
3, 2	CCS1[1:0]	CBS Counter Status 1 (Class A) 00_B: The current credit value is within the range. 01_B: The current credit value is less than or equal to the lower limit. 10_B: The current credit value is greater than or equal to the upper limit. 11_B: (Reserved)
1, 0	CCS0[1:0]	CBS Counter Status 0 (Class B) 00_B: The current credit value is within the range. 01_B: The current credit value is less than or equal to the lower limit. 10_B: The current credit value is greater than or equal to the upper limit. 11_B: (Reserved)

MFFL[4:0]: MAC status FIFO Fill Level

These bits indicate the number of entries available in the MAC status FIFO.

If this value is 0, the FIFO is empty. If this value is 16, the FIFO is full. The values 17 to 31 are reserved.

Conditions for updating:

- These bits are set to 0 when leaving OPERATION mode.
- These bits are set to 0 when ETNBnTCCR.MFEN is 0.
- This value is incremented when frame with DESCR.MSR has been transmitted by MAC, ETNBnTCCR.MFEN is 1 and ETNBnTSR.MFFL is not 16.
- This value is incremented when MAC detects an error during transmission, ETNBnTCCR.MFEN

is 1 and ETNBnTSR.MFFL is not 16.

- This value is decremented when 1 is written to ETNBnTCCR.MFR and ETNBnTSR.MFFL is not 0.

TFFL[2:0]: Timestamp FIFO Count Bits

These bits indicate the number of timestamps in the timestamp FIFO.

The value 0 indicates it is empty and the value 3 indicates it is full (values 4 to 7 are reserved).

Conditions for updating:

- The bits are set to 0 when the operating mode is not operation mode or when the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) = 0.
- When the timestamp FIFO enable bit (ETNBnTCCR.TFEN) is 1 and these bits are not 3, the value of these bits is incremented after a frame with DESCR.TSR set has been transmitted by the E-MAC (for DESCR.TSR, see **Section 25.4.5.2(2), Configuration of Transmission Frame Data Descriptors**).

  The value of these bits is decremented if it is not 0 when 1 is written to the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR).

CCS0 and CCS1[1:0]: CBS Counter Status 0 and 1 Bits

These bits indicate the CBS (credit-based shaping) state of stream data transmission queues 0 and 1. If the calculated credit value is outside the range specified by CBS upper limit register c (ETNBnCULc) and CBS lower limit register c (ETNBnCLLc), it falls outside the range for CBS.

Conditions for updating:

- The bits are set to 00_B when the operating mode is not operation mode.
- When the determined credit value is within the range, 00_B is set.
- If the determined credit value is lower than the CBS lower limit register c (ETNBnCLLc), 01_B is set.
- If the determined credit value is higher than the upper limit register c (ETNBnCULc), 10_B is set.

25.3.2.24 ETNBnTFA0 — Timestamp FIFO Access Register 0

ETNBnTFA0 indicates the nano seconds portion of the timestamp value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0310_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.34 ETNBnTFA0 Register Contents

Bit Position	Bit Name	Function
31 to 0	TSV[31:0]	Timestamp Value

TSV[79:0]: Timestamp Value Bits

These 80 bits consist of ETNBnTFA0.TSV[31:0], ETNBnTFA1.TSV[63:32], and ETNBnTFA2.TSV[79:64], which together indicate the oldest timestamp value stored in the timestamp FIFO.

Once the timestamp FIFO is full, no further timestamp values are stored.

Conditions for updating:

- The bits are set to $0000\ 0000_H$ when the operating mode is not operation mode.
- The register is updated whenever a value is stored in the timestamp FIFO (when the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) changes from 0 to 1).
- The register is updated when the oldest entry is released (when the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR) is set to 1).

25.3.2.25 ETNBnTFA1 — Timestamp FIFO Access Register 1

The ETNBnTFA1 register indicates the lower seconds portion of the timestamp value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0314_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSV[63:48]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.35 ETNBnTFA1 Register Contents

Bit Position	Bit Name	Function
31 to 0	TSV[63:32]	Timestamp Value

TSV[63:32]: Timestamp Value Bits

For details, see **Section 25.3.2.24, ETNBnTFA0 — Timestamp FIFO Access Register 0**.

25.3.2.26 ETNBnTFA2 — Timestamp FIFO Access Register 2

The ETNBnTFA2 register indicates the timestamp tag and the higher seconds portion of the timestamp value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0318_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	TST[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSV[79:64]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.36 ETNBnTFA2 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25 to 16	TST[9:0]	Timestamp Tag
15 to 0	TSV[79:64]	Timestamp Value

TST[9:0]: Timestamp Tag Bits

These bits indicate the contents of the DESCR.TAG bit within the descriptor for frame transmission. These values are used to check the correlation between frames within the transmission queue and the timestamp values (timestamp FIFO access register 0 to 2 (ETNBnTFA0 to 2)) which can be placed in the FIFO.

For transmit frame tagging, **Section 25.4.5.4, Timestamping in Transmission**.

Conditions for updating:

- The bits are set to 000_H when the operating mode is not operation mode.
- The bits are updated when a value is stored in the timestamp FIFO (when the value of the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) changes from 0 to 1).
- The bits are updated when the oldest entry has been released (1 is set in the timestamp FIFO release bit in the transmit configuration control register (ETNBnTCCR.TFR)).

TSV[79:64]: Timestamp Value Bits

For details, see **Section 25.3.2.24, ETNBnTFA0 — Timestamp FIFO Access Register 0**.

25.3.2.27 ETNBnVRR — AVB-DMAC Version Register

The ETNBnVRR register indicates the version of the AVB-DMAC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $031C_H$

Value after reset: $0000\ E300_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC[15:0]															
Value after reset	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.37 ETNBnVRR Register Contents

Bit Position	Bit Name	Function
31 to 0	VC[31:0]	Version Code Version code of AVB-DMAC

VC[31:0]: Version Code Bits

These bits indicates the version of the AVB-DMAC.

$0000\ 0000_H$: RAVBES1/2

$0000\ E300_H$: RAVBES3

25.3.2.28 ETNBnCIVRc — CBS Increment Value Register c (c = 0 or 1)

The ETNBnCIVR0 register sets the increment in the CBS algorithm for transmission queue 2 (for stream class B) and the ETNBnCIVR1 register sets the increment in the CBS algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0320_H + c × 4_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CIV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.38 ETNBnCIVRc Register Contents

Bit Position	Bit Name	Function
31 to 0	CIV[31:0]	CBS Increment Value Setting value: 1 to 65535 ($0000\ 0001_H$ to $0000\ FFFF_H$)

CIV[31:0]: CBS Increment Value Bits

These bits set the increment for the CBS algorithm.

Set a value in the range from 1 to 65535 ($0000\ 0001_H$ to $0000\ FFFF_H$).

The value to be written to these bits depends on the Ethernet bit rate and clk_chi (peripheral bus clock). For details, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

25.3.2.29 ETNBnCDVRc — CBS Decrement Value Register c (c = 0 or 1)

The ETNBnCDVR0 register sets the decrement in the CBS algorithm for transmission queue 2 (for stream class B).

The ETNBnCDVR1 register sets the decrement in the CBS algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0328_H + c × 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDV[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.39 ETNBnCDVRc Register Contents

Bit Position	Bit Name	Function
31 to 0	CDV[31:0]	CBS Decrement Value Setting value: -1 to -65536 (FFFF $FFFF_H$ to FFFF 0000_H)

CDV[31:0]: CBS Decrement Value Bits

These bits set the decrement for the CBS algorithm.

Set a negative value from -1 to -65536 (FFFF $FFFF_H$ to FFFF 0000_H).

The value to be written to these bits depends on the Ethernet bit rate and clk_chi (peripheral bus clock). For details, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

25.3.2.30 ETNBnCULc — CBS Upper Limit Register c (c = 0 or 1)

The ETNBnCUL0 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The ETNBnCUL1 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0330_H + c × 4_H

Value after reset: $7FFF\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULV[31:16]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ULV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.40 ETNBnCULc Register Contents

Bit Position	Bit Name	Function
31 to 0	ULV[31:0]	CBS Upper Limit Upper limit on CBS values

ULV[31:0]: CBS Upper Limit Bits

These bits set the upper limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a positive value to these bits.

For details, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

25.3.2.31 ETNBnCLLc — CBS Lower Limit Register c (c = 0 or 1)

The ETNBnCLL0 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The ETNBnCLL1 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0338_H + c × 4_H

Value after reset: $8000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LLV[31:16]															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.41 ETNBnCLLc Register Contents

Bit Position	Bit Name	Function
31 to 0	LLV[31:0]	CBS Lower Limit Lower limit on CBS values

LLV[31:0]: CBS Lower Limit Bits

These bits set the lower limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a negative value to these bits.

For details, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

25.3.2.32 ETNBnDIC — Descriptor Interrupt Control Register

The ETNBnDIC register is used to control descriptor interrupts 1 to 15.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0350_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPE15	DPE14	DPE13	DPE12	DPE11	DPE10	DPE9	DPE8	DPE7	DPE6	DPE5	DPE4	DPE3	DPE2	DPE1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 25.42 ETNBnDIC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	DPE15	Descriptor Interrupt Enable 15 0: Disabled 1: Enabled
14	DPE14	Descriptor Interrupt Enable 14 0: Disabled 1: Enabled
13	DPE13	Descriptor Interrupt Enable 13 0: Disabled 1: Enabled
12	DPE12	Descriptor Interrupt Enable 12 0: Disabled 1: Enabled
11	DPE11	Descriptor Interrupt Enable 11 0: Disabled 1: Enabled
10	DPE10	Descriptor Interrupt Enable 10 0: Disabled 1: Enabled
9	DPE9	Descriptor Interrupt Enable 9 0: Disabled 1: Enabled
8	DPE8	Descriptor Interrupt Enable 8 0: Disabled 1: Enabled
7	DPE7	Descriptor Interrupt Enable 7 0: Disabled 1: Enabled
6	DPE6	Descriptor Interrupt Enable 6 0: Disabled 1: Enabled
5	DPE5	Descriptor Interrupt Enable 5 0: Disabled 1: Enabled

Table 25.42 ETNBnDIC Register Contents (2/2)

Bit Position	Bit Name	Function
4	DPE4	Descriptor Interrupt Enable 4 0: Disabled 1: Enabled
3	DPE3	Descriptor Interrupt Enable 3 0: Disabled 1: Enabled
2	DPE2	Descriptor Interrupt Enable 2 0: Disabled 1: Enabled
1	DPE1	Descriptor Interrupt Enable 1 0: Disabled 1: Enabled
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

DPEi (i = 1 to 15): Descriptor Interrupt Enable Bits

When interrupts are allowed, and if an interrupt source occurs (descriptor interrupt status bit of the descriptor interrupt status register (ETNBnDIS 1 to 15) = 1), an interrupt is generated.

25.3.2.33 ETNBnDIS — Descriptor Interrupt Status Register

The ETNBnDIS register indicates the state of descriptor interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0354_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPF15	DPF14	DPF13	DPF12	DPF11	DPF10	DPF9	DPF8	DPF7	DPF6	DPF5	DPF4	DPF3	DPF2	DPF1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 25.43 ETNBnDIS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	DPF15	Descriptor Interrupt Status 15 0: The interrupt is not pending. 1: The interrupt is pending.
14	DPF14	Descriptor Interrupt Status 14 0: The interrupt is not pending. 1: The interrupt is pending.
13	DPF13	Descriptor Interrupt Status 13 0: The interrupt is not pending. 1: The interrupt is pending.
12	DPF12	Descriptor Interrupt Status 12 0: The interrupt is not pending. 1: The interrupt is pending.
11	DPF11	Descriptor Interrupt Status 11 0: The interrupt is not pending. 1: The interrupt is pending.
10	DPF10	Descriptor Interrupt Status 10 0: The interrupt is not pending. 1: The interrupt is pending.
9	DPF9	Descriptor Interrupt Status 9 0: The interrupt is not pending. 1: The interrupt is pending.
8	DPF8	Descriptor Interrupt Status 8 0: The interrupt is not pending. 1: The interrupt is pending.
7	DPF7	Descriptor Interrupt Status 7 0: The interrupt is not pending. 1: The interrupt is pending.
6	DPF6	Descriptor Interrupt Status 6 0: The interrupt is not pending. 1: The interrupt is pending.
5	DPF5	Descriptor Interrupt Status 5 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.43 ETNBnDIS Register Contents (2/2)

Bit Position	Bit Name	Function
4	DPF4	Descriptor Interrupt Status 4 0: The interrupt is not pending. 1: The interrupt is pending.
3	DPF3	Descriptor Interrupt Status 3 0: The interrupt is not pending. 1: The interrupt is pending.
2	DPF2	Descriptor Interrupt Status 2 0: The interrupt is not pending. 1: The interrupt is pending.
1	DPF1	Descriptor Interrupt Status 1 0: The interrupt is not pending. 1: The interrupt is pending.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

DPFi (i = 1 to 15): Descriptor Interrupt Status Bits

When DESCR.DIE is 1 to 15 (0001_B to 1111_B), the corresponding bit indicates completion of the processing of a descriptor within the reception or transmission queue.

When DESCR.DIE is 0, the descriptor interrupt is not generated.

Only 0 can be written to these bits.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode.

– The bit is set to 1 after a descriptor with DESCR.DIE set to the corresponding number from 1 to 15 (0001_B to 1111_B) is processed.

25.3.2.34 ETNBnEIC — Error Interrupt Control Register

The ETNBnEIC register controls the AVB-DMAC-related error interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0358_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TBFE	MFFE	TFFE	CULE1	CULE0	CLLE1	CLLE0	SEE	QEE	MTEE	MREE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.44 ETNBnEIC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	TBFE	Tx-Buffer Full Interrupt Enable 0: Disabled 1: Enabled
9	MFFE	MAC Status FIFO Full Interrupt Enable 0: Disabled 1: Enabled
8	TFFE	Timestamp FIFO Full-Error Interrupt Enable 0: Disabled 1: Enabled
7	CULE1	CBS Upper Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
6	CULE0	CBS Upper Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
5	CLLE1	CBS Lower Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
4	CLLE0	CBS Lower Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
3	SEE	Separation Error interrupt Enable 0: Disabled 1: Enabled
2	QEE	Queue Error interrupt Enable 0: Disabled 1: Enabled
1	MTEE	MAC Transmission Error interrupt Enable 0: Disabled 1: Enabled
0	MREE	MAC Reception Error interrupt Enable 0: Disabled 1: Enabled

TBFE: Tx-Buffer Full Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnEIS.TBFF is 1.

MFFE: MAC status FIFO Full Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnEIS.MFFF is 1.

TFFE: Timestamp FIFO Full-Error Interrupt Enable Bits

When the timestamp FIFO is full (the timestamp FIFO full error interrupt status bits of the error interrupt status register (ETNBnEIS.TFFF) = 1) and the interrupt is enabled, the interrupt is issued.

CULE1: CBS Upper Limit Error Interrupt Enable Bit (Class A)

When the CBS of Class A reached the upper limit value (CBS upper limit error interrupt status bit (Class A) (ETNBnEIS.CULF1) = 1) and the interrupt is enabled, an interrupt occurs.

CULE0: CBS Upper Limit Error Interrupt Enable Bit (Class B)

When the CBS of Class B reached the upper limit value (CBS upper limit error interrupt status bit (Class B) (ETNBnEIS.CULF0) = 1) and the interrupt is enabled, an interrupt occurs.

CLLE1: CBS Lower Limit Error Interrupt Enable Bit (Class A)

When the CBS of Class A reached the lower limit value (CBS lower limit error interrupt status bit (Class A) (ETNBnEIS.CLLF1) = 1) and the interrupt is enabled, an interrupt occurs.

CLLE0: CBS Lower Limit Error Interrupt Enable Bit (Class B)

When the CBS of Class B reached the lower limit value (CBS lower limit error interrupt status bit (Class B) (ETNBnEIS.CLLF0)=1) and the interrupt is enabled, an interrupt occurs.

SEE: Separation Error interrupt Enable Bit

When the separation error flag (ETNBnEIS.SEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

QEE: Queue Error interrupt Enable Bit

When the queue error flag (ETNBnEIS.QEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

MTEE: MAC Transmission Error interrupt Enable Bit

When the MAC transmission error flag (ETNBnEIS.MTEF) is set to 1 and the interrupt is enabled, an interrupt occurs.

MREE: MAC Reception Error interrupt Enable Bit

When the MAC reception error flag (ETNBnEIS.MREF) is set to 1 and the interrupt is enabled, an interrupt occurs.

25.3.2.35 ETNBnEIS — Error Interrupt Status Register

The ETNBnEIS register indicates the states of AVB-DMAC-related error interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $035C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TBFF	MFFF	TFFF	CULF1	CULF0	CLLF1	CLLF0	SEF	QEF	MTEF	MREF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.45 ETNBnEIS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	QFS	Queue Full Summary 0: The interrupt is not pending. 1: The interrupt is pending.
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	TBFF	Tx-Buffer Full Flag 0: The interrupt is not pending. 1: The interrupt is pending.
9	MFFF	MAC Status FIFO Full Flag 0: The interrupt is not pending. 1: The interrupt is pending.
8	TFFF	Timestamp FIFO Full Flag 0: The interrupt is not pending. 1: The interrupt is pending.
7	CULF1	CBS Upper Limit reached Flag 1 (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
6	CULF0	CBS Upper Limit reached Flag 0 (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
5	CLLF1	CBS Lower Limit reached Flag 1 (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
4	CLLF0	CBS Lower Limit reached Flag 0 (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
3	SEF	Separation Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
2	QEF	Queue Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.45 ETNBnEIS Register Contents (2/2)

Bit Position	Bit Name	Function
1	MTEF	MAC Transmission Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
0	MREF	MAC Reception Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.

QFS: Queue Full Error Status Bit

With the interrupts enabled, this bit indicates that a queue is full (the receive queue r full interrupt status bit (QFFr) or the receive FIFO full interrupt status bit (RFFF) in receive interrupt status register 2 (ETNBnRIS2) = 1).

[Conditions for Changing]

– If the receive queue r full interrupt status bit (ETNBnRIS2.QFFr) and the receive queue r full interrupt enable bit in the receive interrupt control register 2 (ETNBnRIC2.QFEr) are updated, this bit is also updated.

– If the receive FIFO full interrupt status bit (ETNBnRIS2.RFFF) and the receive FIFO full interrupt enable bit (ETNBnRIC2.RFFE) are updated, this bit is also updated.

TBFF: Tx-Buffer Full Flag

This bit indicates that fetching transmit data was delayed because the Tx-Buffer has no sufficient storage available to store fetched frame. It depends on configuration of ETNBnTGC.TBDt if a full Tx-Buffer is expected during normal operation.

No frame gets lost but transmission priority may be influenced.

The CPU can only write 0 to this bit.

[Conditions for Changing]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 1 when frame data has been fetched from URAM but there is no storage available in Tx-Buffer.

MFFF: MAC status FIFO Full Flag

This bit indicates that a transmission MAC status is overwritten because the MAC status FIFO is full (overwrite condition).

The CPU can only write 0 to this bit.

[Conditions for Changing]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 1 when a frame with DESCR.MSR has been transmitted by MAC, ETNBnTCCR.MFEN is 1 and ETNBnTSR.MFFL is 16.

This bit is set to 1 when MAC detects an error during transmission and ETNBnTSR.MFFL is 16.

TFFF: Timestamp FIFO Full-Error Interrupt Status Bit

This bit indicates that a new transmission timestamp has been discarded due to the timestamp FIFO being full (i.e. has reached the overrun state).

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when a frame with DESCR.TSR set is transmitted while the timestamp FIFO enable bit in the transmit configuration control register (ETNBnTCCR.TFEN) is set to 1 and the timestamp FIFO count bit in the transmit status register (ETNBnTSR.TFFL) is set to 3.

CULF1: CBS Upper Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has exceeded the set upper limit (ETNBnCUL1.ULV in the CBS upper limit register c (ETNBnCULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

- This bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 1 when the value of the CBS counter status 1 (Class A) bits in the transmit status register (ETNBnTSR.CCS1) change from 00_B (within the range) to 10_B (indicating a value equal to or higher than the upper limit).

CULF0: CBS Upper Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has exceeded the set upper limit (ETNBnCUL0.ULV in the CBS upper limit register c (ETNBnCULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 0 (Class B) (ETNBnTSR.CCS0) bit in the transmit status register changes from 00_B (within the range) to 10_B (indicating a value over the upper limit).

CLLF1: CBS Lower Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has fallen below the set lower limit (ETNBnCLL1.LLV in CBS lower limit register c (ETNBnCLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 1 (Class A) bit in the transmit status register (ETNBnTSR.CCS1) changes from 00_B (within the range) to 01_B (indicating a value less than the lower limit).

CLLF0: CBS Lower Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has fallen below the set lower limit (ETNBnCLL0.LLV in the CBS lower limit register c (ETNBnCLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register (ETNBnTSR.CCS0) changes from 00_B (within the range) to 01_B (indicating a value less than the lower limit).

SEF: Separation Error Flag

This bit indicates that a received frame was discarded because it has not matched any configured separation filter for AVB stream data frames.

The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 1 when a valid AVB stream data frame was received by MAC but discarded because the separation filter select bits in the receive configuration register (ETNBnRCR.ESF) are 10_B and no separation filter has matched.

QEF: Queue Error Flag

This bit indicates that an error has been detected while processing reception or transmit queue.

Detail about the detected error is indicated by the error status register (ETNBnESR).

For details of the error processing, **Section 25.4.2.3, Checking Integrity**.

The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when the operating mode changes from operation mode.
- This bit is set to 1 when a state of error is detected.

MTEF: MAC Transmission Error Flag

This bit indicates that the MAC has detected a fault during transmission.

For details, the MAC registers have to be checked.

The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when the operating mode changes from operation mode.
- This bit is set to 1 when MAC detects an error during frame transmission.

MREF: MAC Reception Error Flag

This bit indicates that the MAC has detected a fault during reception.

For details, the MAC registers have to be checked.

NOTE

When the storage of faulty received frames (ETNBnRCR.EFFS) is enabled, the MAC error code (DESCR.MSC) is stored in the descriptor. By evaluating this information CPU can identify corrupted frames in URAM.

The CPU can only write 0 to this bit.

[Changing condition]

- This bit is set to 0 when leaving OPERATION mode.
- This bit is set to 1 when E-MAC detects an error during frame reception.

25.3.2.36 ETNBnRIC0 — Receive Interrupt Control Register 0

The ETNBnRIC0 register controls the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0360_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRE17	FRE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRE15	FRE14	FRE13	FRE12	FRE11	FRE10	FRE9	FRE8	FRE7	FRE6	FRE5	FRE4	FRE3	FRE2	FRE1	FRE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.46 ETNBnRIC0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	FRE17	Receive Frame Interrupt Enable 17 (Stream 15) 0: Disabled 1: Enabled
16	FRE16	Receive Frame Interrupt Enable 16 (Stream 14) 0: Disabled 1: Enabled
15	FRE15	Receive Frame Interrupt Enable 15 (Stream 13) 0: Disabled 1: Enabled
14	FRE14	Receive Frame Interrupt Enable 14 (Stream 12) 0: Disabled 1: Enabled
13	FRE13	Receive Frame Interrupt Enable 13 (Stream 11) 0: Disabled 1: Enabled
12	FRE12	Receive Frame Interrupt Enable 12 (Stream 10) 0: Disabled 1: Enabled
11	FRE11	Receive Frame Interrupt Enable 11 (Stream 9) 0: Disabled 1: Enabled
10	FRE10	Receive Frame Interrupt Enable 10 (Stream 8) 0: Disabled 1: Enabled
9	FRE9	Receive Frame Interrupt Enable 9 (Stream 7) 0: Disabled 1: Enabled
8	FRE8	Receive Frame Interrupt Enable 8 (Stream 6) 0: Disabled 1: Enabled
7	FRE7	Receive Frame Interrupt Enable 7 (Stream 5) 0: Disabled 1: Enabled

Table 25.46 ETNBnRIC0 Register Contents (2/2)

Bit Position	Bit Name	Function
6	FRE6	Receive Frame Interrupt Enable 6 (Stream 4) 0: Disabled 1: Enabled
5	FRE5	Receive Frame Interrupt Enable 5 (Stream 3) 0: Disabled 1: Enabled
4	FRE4	Receive Frame Interrupt Enable 4 (Stream 2) 0: Disabled 1: Enabled
3	FRE3	Receive Frame Interrupt Enable 3 (Stream 1) 0: Disabled 1: Enabled
2	FRE2	Receive Frame Interrupt Enable 2 (Stream 0) 0: Disabled 1: Enabled
1	FRE1	Receive Frame Interrupt Enable 1 (Network Control) 0: Disabled 1: Enabled
0	FRE0	Receive Frame Interrupt Enable 0 (Best Effort) 0: Disabled 1: Enabled

FREr (r = 0 to 17): Receive Frame Interrupt Enable Bits

When interrupts are allowed, and if an interrupt source occurs (receive interrupt status bit in the receive interrupt status register (ETNBnRIS0.FRF0 to 17) = 1), an interrupt is generated.

[Changing condition]

– This bit is set to 0 when writing 1 to ETNBnRID0.FRDr.

– This bit is set to 1 when writing 1 to ETNBnRIE0.FRSr.

25.3.2.37 ETNBnRIS0 — Receive Interrupt Status Register 0

The ETNBnRIS0 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0364_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRF17	FRF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRF15	FRF14	FRF13	FRF12	FRF11	FRF10	FRF9	FRF8	FRF7	FRF6	FRF5	FRF4	FRF3	FRF2	FRF1	FRF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.47 ETNBnRIS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	FRF17	Receive Frame Interrupt Status 17 (Stream 15) 0: The interrupt is not pending. 1: The interrupt is pending.
16	FRF16	Receive Frame Interrupt Status 16 (Stream 14) 0: The interrupt is not pending. 1: The interrupt is pending.
15	FRF15	Receive Frame Interrupt Status 15 (Stream 13) 0: The interrupt is not pending. 1: The interrupt is pending.
14	FRF14	Receive Frame Interrupt Status 14 (Stream 12) 0: The interrupt is not pending. 1: The interrupt is pending.
13	FRF13	Receive Frame Interrupt Status 13 (Stream 11) 0: The interrupt is not pending. 1: The interrupt is pending.
12	FRF12	Receive Frame Interrupt Status 12 (Stream 10) 0: The interrupt is not pending. 1: The interrupt is pending.
11	FRF11	Receive Frame Interrupt Status 11 (Stream 9) 0: The interrupt is not pending. 1: The interrupt is pending.
10	FRF10	Receive Frame Interrupt Status 10 (Stream 8) 0: The interrupt is not pending. 1: The interrupt is pending.
9	FRF9	Receive Frame Interrupt Status 9 (Stream 7) 0: The interrupt is not pending. 1: The interrupt is pending.
8	FRF8	Receive Frame Interrupt Status 8 (Stream 6) 0: The interrupt is not pending. 1: The interrupt is pending.
7	FRF7	Receive Frame Interrupt Status 7 (Stream 5) 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.47 ETNBnRIS0 Register Contents (2/2)

Bit Position	Bit Name	Function
6	FRF6	Receive Frame Interrupt Status 6 (Stream 4) 0: The interrupt is not pending. 1: The interrupt is pending.
5	FRF5	Receive Frame Interrupt Status 5 (Stream 3) 0: The interrupt is not pending. 1: The interrupt is pending.
4	FRF4	Receive Frame Interrupt Status 4 (Stream 2) 0: The interrupt is not pending. 1: The interrupt is pending.
3	FRF3	Receive Frame Interrupt Status 3 (Stream 1) 0: The interrupt is not pending. 1: The interrupt is pending.
2	FRF2	Receive Frame Interrupt Status 2 (Stream 0) 0: The interrupt is not pending. 1: The interrupt is pending.
1	FRF1	Receive Frame Interrupt Status 1 (Network Control) 0: The interrupt is not pending. 1: The interrupt is pending.
0	FRF0	Receive Frame Interrupt Status 0 (Best Effort) 0: The interrupt is not pending. 1: The interrupt is pending.

FRFr (r = 0 to 17): Receive Frame Interrupt Status Bits

Each bit indicates that a corresponding frame has been stored normally in reception queues 0 to 17 and that data is queued for CPU processing.

Only 0 can be written to the bit.

[Conditions for Changing]

- A bit is set to 0 when the operating mode is not operation mode.
- A bit is set to 0 when a value is written to the unread frame counter decrement register i (ETNBnUFCDi) (i = 0 to 4), and this decrements the value of unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) to 0.
- When a frame is stored normally in a reception queue, the corresponding bit is set to 1.

25.3.2.38 ETNBnRIC1 — Receive Interrupt Control Register 1

The ETNBnRIC1 register controls AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0368_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWE	—	—	—	—	—	—	—	—	—	—	—	—	—	RWE17	RWE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWE15	RWE14	RWE13	RWE12	RWE11	RWE10	RWE9	RWE8	RWE7	RWE6	RWE5	RWE4	RWE3	RWE2	RWE1	RWE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.48 ETNBnRIC1 Register Contents

Bit Position	Bit Name	Function
31	RFWE	Receive FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RWEr (r = 17 to 0)	Reception Warning Interrupt Enable 0: Disabled 1: Enabled

RFWE: Receive FIFO Warning Interrupt Enable Bit

While this bit is 1 an interrupt will be generated when ETNBnRIS1.RFWF is 1.

[Conditions for Changing]

– This bit is set to 0 when writing 1 to ETNBnRID1.RFWD.

– This bit is set to 1 when writing 1 to ETNBnRIE1.RFWS.

RWEr: Receive Warning Interrupt Enable Bit (r = 0 to 17)

While this bit is 1 an interrupt will be generated when ETNBnRIS1.RWFr is 1.

[Conditions for Changing]

– This bit is set to 0 when writing 1 to ETNBnRID1.RWDr.

– This bit is set to 1 when writing 1 to ETNBnRIE1.RWSr.

25.3.2.39 ETNBnRIS1 — Receive Interrupt Status Register 1

The ETNBnRIS1 register indicates the states of AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $036C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWF	—	—	—	—	—	—	—	—	—	—	—	—	—	RWF17	RWF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWF15	RWF14	RWF13	RWF12	RWF11	RWF10	RWF9	RWF8	RWF7	RWF6	RWF5	RWF4	RWF3	RWF2	RWF1	RWF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.49 ETNBnRIS1 Register Contents (1/2)

Bit Position	Bit Name	Function
31	RFWF	Receive FIFO Warning Flag 0: The interrupt is not pending. 1: The interrupt is pending.
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RWF17	Receive Warning Flag 17 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
16	RWF16	Receive Warning Flag 16 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
15	RWF15	Receive Warning Flag 15 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
14	RWF14	Receive Warning Flag 14 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
13	RWF13	Receive Warning Flag 13 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
12	RWF12	Receive Warning Flag 12 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
11	RWF11	Receive Warning Flag 11 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
10	RWF10	Receive Warning Flag 10 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
9	RWF9	Receive Warning Flag 9 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
8	RWF8	Receive Warning Flag 8 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.

Table 25.49 ETNBnRIS1 Register Contents (2/2)

Bit Position	Bit Name	Function
7	RWF7	Receive Warning Flag 7 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
6	RWF6	Receive Warning Flag 6 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
5	RWF5	Receive Warning Flag 5 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
4	RWF4	Receive Warning Flag 4 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
3	RWF3	Receive Warning Flag 3 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
2	RWF2	Receive Warning Flag 2 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
1	RWF1	Receive Warning Flag 1 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.
0	RWF0	Receive Warning Flag 0 (Stream) 0: The interrupt is not pending. 1: Unread frame counter warning level reached.

RFWF: Receive FIFO Warning Flag

This bit indicates that the reception FIFO has reached configured critical warning level.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when a complete frame is received in the reception FIFO and the fill level goes above the configured critical warning level during reception of this frame (ETNBnRCR.RFCL).

RWFr: Reception Warning Flag (r = 0 to 17)

This bit indicates that the unread frame counter of receive queue r has reached the configured warning level.

Only 0 can be written to the bit.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode.
– The bit is set to 0 when CPU request a reload of the base address (ETNBnDLR.LBAq is 1 q = r +4).
– This bit is set to 0 when CPU writes to ETNBnUFCDi.DVr and ETNBnUFCVi.CVr is decremented to a value less than the selected warning level ETNBnUFCW.WLj.
– This bit is set to 1 when a frame is stored in queue r and the unread frame counter value r is incremented to the selected warning level j (ETNBnUFCVi.CVr +1 == ETNBnUFCW.WLj)

25.3.2.40 ETNBnRIC2 — Receive Interrupt Control Register 2

The ETNBnRIC2 register controls AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0370_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFE	—	—	—	—	—	—	—	—	—	—	—	—	—	QFE17	QFE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFE15	QFE14	QFE13	QFE12	QFE11	QFE10	QFE9	QFE8	QFE7	QFE6	QFE5	QFE4	QFE3	QFE2	QFE1	QFE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.50 ETNBnRIC2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	RFFE	Receive FIFO Full Interrupt Enable 0: Disabled 1: Enabled
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	QFE17	Receive Queue 17 (Stream 15) Full Interrupt Enable 0: Disabled 1: Enabled
16	QFE16	Receive Queue 16 (Stream 14) Full Interrupt Enable 0: Disabled 1: Enabled
15	QFE15	Receive Queue 15 (Stream 13) Full Interrupt Enable 0: Disabled 1: Enabled
14	QFE14	Receive Queue 14 (Stream 12) Full Interrupt Enable 0: Disabled 1: Enabled
13	QFE13	Receive Queue 13 (Stream 11) Full Interrupt Enable 0: Disabled 1: Enabled
12	QFE12	Receive Queue 12 (Stream 10) Full Interrupt Enable 0: Disabled 1: Enabled
11	QFE11	Receive Queue 11 (Stream 9) Full Interrupt Enable 0: Disabled 1: Enabled
10	QFE10	Receive Queue 10 (Stream 8) Full Interrupt Enable 0: Disabled 1: Enabled
9	QFE9	Receive Queue 9 (Stream 7) Full Interrupt Enable 0: Disabled 1: Enabled
8	QFE8	Receive Queue 8 (Stream 6) Full Interrupt Enable 0: Disabled 1: Enabled

Table 25.50 ETNBnRIC2 Register Contents (2/2)

Bit Position	Bit Name	Function
7	QFE7	Receive Queue 7 (Stream 5) Full Interrupt Enable 0: Disabled 1: Enabled
6	QFE6	Receive Queue 6 (Stream 4) Full Interrupt Enable 0: Disabled 1: Enabled
5	QFE5	Receive Queue 5 (Stream 3) Full Interrupt Enable 0: Disabled 1: Enabled
4	QFE4	Receive Queue 4 (Stream 2) Full Interrupt Enable 0: Disabled 1: Enabled
3	QFE3	Receive Queue 3 (Stream 1) Full Interrupt Enable 0: Disabled 1: Enabled
2	QFE2	Receive Queue 2 (Stream 0) Full Interrupt Enable 0: Disabled 1: Enabled
1	QFE1	Receive Queue 1 (Network Control) Full Interrupt Enable 0: Disabled 1: Enabled
0	QFE0	Receive Queue 0 (Best Effort) Full Interrupt Enable 0: Disabled 1: Enabled

RFFE: Receive FIFO Full Interrupt Enable Bit

When interrupts are allowed, and if the reception FIFO becomes full (the reception FIFO full interrupt status bit (ETNBnRIS2.RFFF) = 1 in receive interrupt status register 2), an interrupt is generated.

[Changing condition]

– This bit is set to 0 when writing 1 to ETNBnRID2.RFFD.

– This bit is set to 1 when writing 1 to ETNBnRIE2.RFFS.

QFEr (r = 0 to 17): Receive Queue r Full Interrupt Enable Bits

When interrupts are allowed, and if a receive queue (0 to 17) becomes full (the receive queue r full interrupt status bit (ETNBnRIS2.QFF0 to 17) = 1 in receive interrupt status register 2), an interrupt is generated.

[Changing condition]

– This bit is set to 0 when writing 1 to ETNBnRID2.QFDr.

– This bit is set to 1 when writing 1 to ETNBnRIE2.QFSr.

25.3.2.41 ETNBnRIS2 — Receive Interrupt Status Register 2

The ETNBnRIS2 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0374_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFF	—	—	—	—	—	—	—	—	—	—	—	—	—	QFF17	QFF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFF15	QFF14	QFF13	QFF12	QFF11	QFF10	QFF9	QFF8	QFF7	QFF6	QFF5	QFF4	QFF3	QFF2	QFF1	QFF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.51 ETNBnRIS2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	RFFF	Receive FIFO Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	QFF17	Receive Queue 17 (Stream 15) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
16	QFF16	Receive Queue 16 (Stream 14) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
15	QFF15	Receive Queue 15 (Stream 13) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
14	QFF14	Receive Queue 14 (Stream 12) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
13	QFF13	Receive Queue 13 (Stream 11) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
12	QFF12	Receive Queue 12 (Stream 10) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
11	QFF11	Receive Queue 11 (Stream 9) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
10	QFF10	Receive Queue 10 (Stream 8) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
9	QFF9	Receive Queue 9 (Stream 7) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
8	QFF8	Receive Queue 8 (Stream 6) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.51 ETNBnRIS2 Register Contents (2/2)

Bit Position	Bit Name	Function
7	QFF7	Receive Queue 7 (Stream 5) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
6	QFF6	Receive Queue 6 (Stream 4) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
5	QFF5	Receive Queue 5 (Stream 3) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
4	QFF4	Receive Queue 4 (Stream 2) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
3	QFF3	Receive Queue 3 (Stream 1) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
2	QFF2	Receive Queue 2 (Stream 0) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
1	QFF1	Receive Queue 1 (Network Control) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
0	QFF0	Receive Queue 0 (Best Effort) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

RFFF: Receive FIFO Full Interrupt Status Bit

This bit indicates that a frame was received but storing it was not possible due to the reception FIFO being full.

When receiving a frame is not possible, the frame will be discarded.

Other information regarding discarded frames is not retained. Even if the frame is not discarded, this bit may also be set to 1 if the E-MAC determines that the frame is an error frame

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the reception FIFO cannot hold received frame data.

QFFr (r = 0 to 17): Receive r Full Interrupt Status Bits

These bits indicate that reception queue r did not have space for storing a received frame.

A reception queue is treated as full when it has no descriptors (descriptor type (DESCR.DT) = FEMPTY, FEMPTY_IS, FEMPTY_IC, or FEMPTY_ND) available or reaches the set level for stopping.

CAUTION

If no FEMPTY descriptors or no empty space for descriptors remains in the queue during storing of a divided frame (see **Section 25.4.4.3(b), Storing Frame Data as Divided Frames**), an error frame is stored in the queue. Such error frames are treated as descriptor sequence errors.

Reading descriptor may be one or more storage elements ahead from actual storage in URAM. ETNBnCDARq.CDA provides information where queue has been stopped.

The CPU can only write 0 to this bit.

[Conditions for Changing]

- A bit is set to 0 when the operating mode is not operation mode.
- A bit is set to 1 when reception queue r has no space available for storage.
- A bits is set to 1 when a SW defined stop point (EOS descriptor) reached within a split frame.
- A bit is set to 1 when the unread frame counter (unread frame counter register i (ETNBnUFCVi) (i = 0 to 4)) reaches the set level for stopping.

25.3.2.42 ETNBnTIC — Transmit Interrupt Control Register

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0378_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	TDPE3	TDPE2	TDPE1	TDPE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MFWE	MFUE	TFWE	TFUE	—	—	—	—	FTE3	FTE2	FTE1	FTE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.52 ETNBnTIC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	TDPE3	Transmit Descriptor Processed Interrupt Enable 3 0: Disabled 1: Enabled
18	TDPE2	Transmit Descriptor Processed Interrupt Enable 2 0: Disabled 1: Enabled
17	TDPE1	Transmit Descriptor Processed Interrupt Enable 1 0: Disabled 1: Enabled
16	TDPE0	Transmit Descriptor Processed Interrupt Enable 0 0: Disabled 1: Enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MFWE	MAC Status FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
10	MFUE	MAC Status FIFO Update Interrupt Enable 0: Disabled 1: Enabled
9	TFWE	Timestamp FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
8	TFUE	Timestamp FIFO Update Interrupt Enable 0: Disabled 1: Enabled
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FTE3	Frame Transmitted Interrupt Enable 3 0: Disabled 1: Enabled
2	FTE2	Frame Transmitted Interrupt Enable 2 0: Disabled 1: Enabled

Table 25.52 ETNBnTIC Register Contents (2/2)

Bit Position	Bit Name	Function
1	FTE1	Frame Transmitted Interrupt Enable 1 0: Disabled 1: Enabled
0	FTE0	Frame Transmitted Interrupt Enable 0 0: Disabled 1: Enabled

TDPEt: Transmit Descriptor Processed Interrupt Enable t (t = 0 to 3)

While this bit is 1 an interrupt will be generated when ETNBnTIS.TDPFt is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.TDPDt.
- The bit is set to 1 when writing 1 to ETNBnTIE.TDPSt.

MFWE: MAC Status FIFO Warning Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnTIS.MFWF is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.MFWD.
- The bit is set to 1 when writing 1 to ETNBnTIE.MFWS.

MFUE: MAC Status FIFO Updated Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnTIS.MFUF is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.MFUD.
- The bit is set to 1 when writing 1 to ETNBnTIE.MFUS.

TFWE: Timestamp FIFO Warning Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnTIS.TFWF is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.TFWD.
- The bit is set to 1 when writing 1 to ETNBnTIE.TFWS.

TFUE: Timestamp FIFO Update Interrupt Enable

While this bit is 1 an interrupt will be generated when ETNBnTIS.TFUF is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.TFUD.
- The bit is set to 1 when writing 1 to ETNBnTIE.TFUS.

FTEt: Frame Transmitted Interrupt Enable t (t = 0 to 3)

While this bit is 1 an interrupt will be generated when ETNBnTIS.FTFt is 1.

[Conditions for Changing]

- The bit is set to 0 when writing 1 to ETNBnTID.FTDt.
- The bit is set to 1 when writing 1 to ETNBnTIE.FTSt.

25.3.2.43 ETNBnTIS — Transmit Interrupt Status Register

The ETNBnTIS register indicates the states of the AVB-DMAC transmit interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $037C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	TDPF3	TDPF2	TDPF1	TDPF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MFWF	MFUF	TFWF	TFUF	—	—	—	—	FTF3	FTF2	FTF1	FTF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.53 ETNBnTIS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	TDPF3	Transmit Descriptor Processed Flag 3 0: The interrupt is not pending. 1: The descriptor interrupt is pending.
18	TDPF2	Transmit Descriptor Processed Flag 2 0: The interrupt is not pending. 1: The descriptor interrupt is pending.
17	TDPF1	Transmit Descriptor Processed Flag 1 0: The interrupt is not pending. 1: The descriptor interrupt is pending.
16	TDPF0	Transmit Descriptor Processed Flag 0 0: The interrupt is not pending. 1: The descriptor interrupt is pending.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MFWF	MAC status FIFO Warning Flag 0: The interrupt is not pending. 1: The Tx Status FIFO has reached the warning level.
10	MFUF	MAC status FIFO Update Flag 0: The interrupt is not pending. 1: The Tx Status FIFO has been updated.
9	TFWF	Timestamp FIFO Warning Flag 0: The interrupt is not pending. 1: The timestamp FIFO has reached the warning level.
8	TFUF	Timestamp FIFO Update Flag 0: The interrupt is not pending. 1: The timestamp FIFO has been updated.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FTF3	Frame Transmitted Flag 3 0: The interrupt is not pending. 1: Frame transmitted by MAC.

Table 25.53 ETNBnTIS Register Contents (2/2)

Bit Position	Bit Name	Function
2	FTF2	Frame Transmitted Flag 2 0: The interrupt is not pending. 1: Frame transmitted by MAC.
1	FTF1	Frame Transmitted Flag 1 0: The interrupt is not pending. 1: Frame transmitted by MAC.
0	FTF0	Frame Transmitted Flag 0 0: The interrupt is not pending. 1: Frame transmitted by MAC.

TDPFt: Transmit Descriptor Processed Flag t (t = 0 to 3)

This bit indicates that a descriptor in transmit queue t has been processed where DESCR.DIE is 0001_B.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- The bit is set to 1 when the descriptor in transmit queue t has been processed where DESCR.DIE is 1.

NOTE

The descriptor with DESCR.DIE is 1, which sets this bit, sets in addition the universal descriptor interrupt ETNBnDIS.DPF1.

MFWF: MAC Status FIFO Warning Flag

This bit indicates that the warning level of the MAC status FIFO (12 out of 16 entries) has been reached.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 0 when ETNBnTCCR.MFEN is 0.
- This bit is set to 0 when writing 1 to ETNBnTCCR.MFR.
- This bit is set to 1 when a frame with DESCR.MSR has been transmitted by MAC and the MAC Status FIFO contains already 11 entries (ETNBnTSR.MFFL is 11).
- The bits is set to 1 when MAC detects an error during transmission and the MAC Status FIFO contains already 11 entries (ETNBnTSR.MFFL is 11).

MFUF: MAC Status FIFO Updated Flag

This bit indicates that the MAC status FIFO has been updated after the MAC has transmitted a frame.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 0 when ETNBnTCCR.MFEN is 0.
- This bit is set to 0 when writing 1 to ETNBnTCCR.MFR.
- This bit is set to 1 when a frame with DESCR.MSR has been transmitted by MAC and ETNBnTCCR.MFEN is 1.
- The bits is set to 1 when MAC detects an error during transmission and ETNBnTCCR.MFEN is 1.

TFWF: Timestamp FIFO Warning Flag

This bit indicates that the warning level of the transmission timestamp FIFO (2 out of 3 entries) has been reached.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 0 when ETNBnTCCR.TFEN is 0.
- This bit is set to 0 when writing 1 to ETNBnTCCR.TFR.
- This bit is set to 1 when a frame with DESCR.TSR has been transmitted by MAC and the timestamp FIFO contains already 1 entries (ETNBnTSR.TFFL is 1).

TFUF: Timestamp FIFO Update Flag

This bit indicates that the transmission timestamp FIFO has been updated.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 0 when ETNBnTCCR.TFEN is 0.
- This bit is set to 0 when writing 1 to ETNBnTCCR.TFR.
- This bit is set to 1 when a frame with DESCR.TSR has been transmitted by MAC and the ETNBnTCCR.TFEN is 1.

FTFt: Frame Transmitted Flag t (t = 0 to 3)

This bit indicates that from transmit queue a frame is transmitted by MAC.

Only 0 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation mode.
- This bit is set to 1 when a frame from transmit queue t has been transmitted by the MAC.

NOTE

This interrupt flag refers to the end of frame is transmission by MAC whereas the descriptor interrupt (ETNBnDIS.DPFi) refer to the end of processing storage element.

25.3.2.44 ETNBnISS — Interrupt Summary Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0380_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DPM15	DPM14	DPM13	DPM12	DPM11	DPM10	DPM9	DPM8	DPM7	DPM6	DPM5	DPM4	DPM3	DPM2	DPM1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CGIM	RFWM	MFWM	MFUM	TFWM	TFUM	MM	EM	—	—	—	FTM	RWM	FRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.54 ETNBnISS Register Contents (1/2)

Bit Position	Bit Name	Function
31	DPM15	Descriptor Processed Mirror 15 0: The interrupt is not pending. 1: The interrupt is pending.
30	DPM14	Descriptor Processed Mirror 14 0: The interrupt is not pending. 1: The interrupt is pending.
29	DPM13	Descriptor Processed Mirror 13 0: The interrupt is not pending. 1: The interrupt is pending.
28	DPM12	Descriptor Processed Mirror 12 0: The interrupt is not pending. 1: The interrupt is pending.
27	DPM11	Descriptor Processed Mirror 11 0: The interrupt is not pending. 1: The interrupt is pending.
26	DPM10	Descriptor Processed Mirror 10 0: The interrupt is not pending. 1: The interrupt is pending.
25	DPM9	Descriptor Processed Mirror 9 0: The interrupt is not pending. 1: The interrupt is pending.
24	DPM8	Descriptor Processed Mirror 8 0: The interrupt is not pending. 1: The interrupt is pending.
23	DPM7	Descriptor Processed Mirror 7 0: The interrupt is not pending. 1: The interrupt is pending.
22	DPM6	Descriptor Processed Mirror 6 0: The interrupt is not pending. 1: The interrupt is pending.
21	DPM5	Descriptor Processed Mirror 5 0: The interrupt is not pending. 1: The interrupt is pending.
20	DPM4	Descriptor Processed Mirror 4 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.54 ETNBnISS Register Contents (2/2)

Bit Position	Bit Name	Function
19	DPM3	Descriptor Processed Mirror 3 0: The interrupt is not pending. 1: The interrupt is pending.
18	DPM2	Descriptor Processed Mirror 2 0: The interrupt is not pending. 1: The interrupt is pending.
17	DPM1	Descriptor Processed Mirror 1 0: The interrupt is not pending. 1: The interrupt is pending.
16 to 14	Reserved	When read, the value after reset is returned.
13	CGIM	Combined gPTP Interrupt Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
12	RFWM	Receive FIFO Warning Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
11	MFWM	MAC Status FIFO Warning Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
10	MFUM	MAC Status FIFO Updated Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
9	TFWM	Timestamp FIFO Warning Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
8	TFUM	Timestamp FIFO Update Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
7	MM	MAC Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
6	EM	Error Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
5 to 3	Reserved	When read, the value after reset is returned.
2	FTM	Frame Transmitted Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
1	RWM	Receive Warning Mirror 0: The interrupt is not pending. 1: The interrupt is pending.
0	FRM	Frame Received Mirror 0: The interrupt is not pending. 1: The interrupt is pending.

DPMi: Descriptor Processed Mirror i (i = 1 to 15)

[Conditions for Changing]

This bit is set when any matching pair of ETNBnDIC.DPEi enable and ETNBnDIS.DPFi flag are both 1.

CGIM: Combined gPTP Interrupt Mirror

This bit indicates that at least one enabled gPTP interrupt (ETNBnGIC and ETNBnGIS) is 1.

[Conditions for Changing]
This bit is updated when ETNBnGIC or ETNBnGIS changes.

RFWM: Receive FIFO Warning Mirror

This bit indicates that ETNBnRIS1.RFWF is 1 and ETNBnRIC1.RFWE is 1.

[Conditions for Changing]
This bit is updated ETNBnRIS1.RFWF or ETNBnRIC1.RFWE changes.

MFWM: MAC Status FIFO Warning Mirror

This bit indicates that ETNBnTIS.MFWF is 1 and ETNBnTIC.MFWE is 1.

[Conditions for Changing]
This bit is updated when ETNBnTIS.MFWF or ETNBnTIC.MFWE changes.

MFUM: MAC Status FIFO Updated Mirror

This bit indicates that ETNBnTIS.MFUF is 1 and ETNBnTIC.MFUE is 1.

[Conditions for Changing]
This bit is updated ETNBnTIS.MFUF or ETNBnTIC.MFUE changes.

TFWM: Timestamp FIFO Warning Mirror

This bit indicates that ETNBnTIS.TFWF is 1 and ETNBnTIC.TFWE is 1.

[Conditions for Changing]
This bit is updated when ETNBnTIS.TFWF or ETNBnTIC.TFWE changes.

TFUM Timestamp FIFO Update Mirror

This bit indicates that ETNBnTIS.TFUF is 1 and ETNBnTIC.TFUE is 1.

[Conditions for Changing]
This bit is updated ETNBnTIS.TFUF or ETNBnTIC.TFUE changes.

MM: MAC Mirror

This bit indicates that there is a pending interrupt request issued by MAC.

For details about MAC interrupt sources, interrupt enabling and timing refer to **Section 25.4.12.1, Flow of E-MAC Initialization**.

NOTE

The MAC is able to assert interrupt also in CONFIG and STANDBY mode.

EM: Error Mirror

This bit indicates that at least one valid flag in ETNBnEIS is 1 or ETNBnEIS.QFS is 1.

[Conditions for Changing]
This bit is updated when ETNBnEIS or ETNBnEIC changes.

FTM: Frame Transmitted Mirror

[Conditions for Changing]

- This bit is set when any matching pair of ETNBnTIC.FTEt enable and ETNBnTIS.FTFt flag are both 1.
- This bit is set when any matching pair of ETNBnTIC.TDPEt enable and ETNBnTIS.TDPFt flag are both 1.

RWM: Reception Warning Mirror

[Conditions for Changing]
This bit is set when any matching pair of ETNBnRIC1.RWEr enable and ETNBnRIS1.RWFr flag are both 1.

FRM: Frame Received Mirror

[Conditions for Changing]

- This bit is set when any matching pair of ETNBnRIC0.FREr enable and ETNBnRIS0.FRFr flag are both 1.
- This bit is set when any matching pair of ETNBnRIC3.RDPEr enable and ETNBnRIS3.RDPFr flag are both 1.

25.3.2.45 ETNBnCIE — Common Interrupt Enable Register

This register is not available for RH850/U2A.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0384_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RFFL	RFWL	CL0M	RQFM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTIE	—	—	—	—	—	—	—	CRIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 25.55 ETNBnCIE Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	RFFL	Rx-FIFO Full interrupt Line select 0: Interrupt line A used for this notification 1: Interrupt line B used for this notification
18	RFWL	Rx-FIFO Warning interrupt Line select 0: Interrupt line A used for this notification 1: Interrupt line B used for this notification
17	CL0M	Common Line 0 Mode 0: Use common interrupt line 0 1: Use queue specific interrupt line 0
16	RQFM	Reception Queue Full Mode 0: Use for error interrupt line 1: Use for queue specific interrupt line
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CTIE	Common Transmit Interrupt Enable 0: Disabled 1: Enabled
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CRIE	Common Receive Interrupt Enable 0: Disabled 1: Enabled

RFFL: Rx-FIFO Full interrupt Line select

This bit selects the interrupt line of notification flagged by ETNBnRIS2.RFFF.

It is recommended to disable interrupt before change the line.

RFWL: Rx-FIFO Warning interrupt Line select

This bit selects the interrupt line of notification flagged by ETNBnRIS1.RFWF.

It is recommended to disable interrupt before change the line.

CL0M: Common Line 0 Mode

This bit controls if all data related interrupts using a common interrupt line or if queue specific interrupt lines for data related interrupts are used. The selected mode has no influence to internal flagging in RAVBES SFR (e.g. ETNBnISS).

To prevent interrupt notifications on different lines, it is recommended to set all ETNBnCIE.CTIE and ETNBnCIE.CRIE to 0 when this bit is 1.

RQFM: Reception Queue Full Mode

This bit controls if queue full notification is mapped to error interrupt line or to queue data interrupt line. The selected mode has no influence to internal flagging in RAVBES SFR (e.g. ETNBnISS).

CTIE: Common Transmit Interrupt Enable

This bit controls transmit related interrupt line outputs of RAVBES. It has no influence to internal flagging in RAVBES SFR (e.g. ETNBnISS).

These lines are controlled by this bit:
Line0_Tx[3:0] -- individual transmit queue interrupts

Transmission interrupts mapped on other lines are not influenced.

CRIE: Common Receive Interrupt Enable

This bit controls receive related interrupt line outputs of RAVBES. It has no influence to internal flagging in RAVBES SFR (e.g. ETNBnISS).

These lines are controlled by this bit:
Line0_Rx[17:0] -- individual receive queue interrupts

Receive interrupts mapped on other lines are not influenced.

25.3.2.46 ETNBnRIC3 — Receive Interrupt Control Register 3

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0388_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDPE 17	RDPE 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDPE 15	RDPE 14	RDPE 13	RDPE 12	RDPE 11	RDPE 10	RDPE9	RDPE8	RDPE7	RDPE6	RDPE5	RDPE4	RDPE3	RDPE2	RDPE1	RDPE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.56 ETNBnRIC3 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RDPEr (r = 17 to 0)	Receive Descriptor Processed Interrupt Enable r 0: Disabled 1: Enabled

RDPEr: Receive Descriptor Processed Interrupt Enable r (r = 0 to 17)

While this bit is 1 an interrupt will be generated when ETNBnRIS3.RDPFr is 1.

[Conditions for Changing]

– This bit is set to 0 when writing 1 to ETNBnRID3.RDPDr.

– This bit is set to 1 when writing 1 to ETNBnRIE3.RDPSr.

25.3.2.47 ETNBnRIS3 — Receive Interrupt Status Register 3

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $038C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDPF17	RDPF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDPF15	RDPF14	RDPF13	RDPF12	RDPF11	RDPF10	RDPF9	RDPF8	RDPF7	RDPF6	RDPF5	RDPF4	RDPF3	RDPF2	RDPF1	RDPF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.57 ETNBnRIS3 Register Contents (1/2)

Bit Position	Bit Name	Function
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RDPF17	Receive Descriptor Processed Flag 17 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
16	RDPF16	Receive Descriptor Processed Flag 16 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
15	RDPF15	Receive Descriptor Processed Flag 15 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
14	RDPF14	Receive Descriptor Processed Flag 14 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
13	RDPF13	Receive Descriptor Processed Flag 13 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
12	RDPF12	Receive Descriptor Processed Flag 12 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
11	RDPF11	Receive Descriptor Processed Flag 11 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
10	RDPF10	Receive Descriptor Processed Flag 10 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
9	RDPF9	Receive Descriptor Processed Flag 9(Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
8	RDPF8	Receive Descriptor Processed Flag 8 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
7	RDPF7	Receive Descriptor Processed Flag 7 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.

Table 25.57 ETNBnRIS3 Register Contents (2/2)

Bit Position	Bit Name	Function
6	RDPF6	Receive Descriptor Processed Flag 6 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
5	RDPF5	Receive Descriptor Processed Flag 5 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
4	RDPF4	Receive Descriptor Processed Flag 4 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
3	RDPF3	Receive Descriptor Processed Flag 3 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
2	RDPF2	Receive Descriptor Processed Flag 2 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
1	RDPF1	Receive Descriptor Processed Flag 1 (Network Control) 0: The interrupt is not pending. 1: The interrupt is pending.
0	RDPF0	Receive Descriptor Processed Flag 0 (Best Effort) 0: The interrupt is not pending. 1: The interrupt is pending.

RDPFr: Receive Descriptor Processed Flag r (r = 0 to 17)

This bit indicates that descriptor in reception queue r has been processed where DESCR.DIE is 0001_B.

Only 0 can be written to the bit.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode.

– The bit is set to 1 when the descriptor in reception queue r has been processed where DESCR.DIE is 1.

NOTE

The descriptor with DESCR.DIE is 1, which sets this bit, sets in addition the universal descriptor interrupt ETNBnDIS.DPF1.

25.3.2.48 ETNBnGCCR — gPTP Configuration Control Register

The ETNBnGCCR register is used to set and control the gPTP (generalized precision time protocol).

Access: ETNBnGCCR can be read or written in 32-bit units.
ETNBnGCCRL,ETNBnGCCRH can be read or written in 16-bit units.
ETNBnGCCRLL, ETNBnGCCRLH, ETNBnGCCRHL can be read or written in 8-bit units.

Address: ETNBnGCCR: <ETNBn_base> + 0390_H
ETNBnGCCRL: <ETNBn_base> + 0390_H
ETNBnGCCRLL: <ETNBn_base> + 0390_H
ETNBnGCCRLH: <ETNBn_base> + 0391_H

Value after reset: $0000\ 002C_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	LI[2:0]			SPC	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TCSS[1:0]		—	—	LMTT	LPTC	LTI	LTO	TCR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.58 ETNBnGCCR register contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	LI[2:0]	Load Index Index number of compare unit to be loaded.
19	SPC	Start Periodic Comparison 0: Absolute comparison value updated by ETNBnGCCR.LPTC request 1: Periodic comparison value updated by ETNBnGCCR.LPTC request
18 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	TCSS[1:0]	Timer Capture Source Select 00_B: gPTP timer value 01_B: Adjusted gPTP timer value 10_B: AVTP presentation time 11_B: No change
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LMTT	Maximum Transit Time Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
4	LPTC	Presentation Time Compare Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
3	LTI	Timer Increment Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.

Table 25.58 ETNBnGCCR register contents (2/2)

Bit Position	Bit Name	Function
2	LTO	Timer Offset Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
1, 0	TCR[1:0]	Timer Control Request 00_B: Timer control is not requested. 01_B: gPTP/AVTP presentation time reset 10_B: Continuous capture of AVTP to ETNBnGCTi.CTV 11_B: Captures the value set in the TCSS bit.

LI[2:0]: Load Index bits

These bits define the target unit to be updated by ETNBnGCCR.LPTC.
Writing to these bits is only possible when ETNBnGCCR.LPTC is 0.

SPC: Start periodic Comparison bit

This bit defines if the absolute comparison value or the periodicity value is updated to the value of ETNBnGPTC.PTCV on ETNBnGCCR.LPTC request.

Periodic comparison starts when updating the periodic comparison value; periodic comparison stops when updating the absolute comparison value.

Writing to these bits is only possible when ETNBnGCCR.LPTC is 0.

TCSS[1:0]: Timer Capture Source Selection Bit

These bits select the source used for updating the captured timer register (gPTP timer capture register (ETNBnGCTi.CTV)).

When writing 11_B to these bit, the write to ETNBnGCCR.TCSS is ignored and ETNBnGCCR.TCSS keeps its old value. This allows changing other parts of ETNBnGCCR without effecting an ongoing capture sequence.
Writing 00_B, 01_B, 10_B to these bits is only possible when ETNBnGCCR.TCR is 00_B.

LMTT: Maximum Transit Time Configuration Request Bit

This bit issues requests for configuring the gPTP maximum transit time configuration register (ETNBnGMTT).

Only 1 can be written to the bit.

[Conditions for Changing]

- The bit is set to 0 when the operating mode is not operation and ETNBnCCC.GAC is 0.
- The bit is set to 0 when the value of the gPTP maximum transit time configuration register (ETNBnGMTT) is loaded.

LPTC: Presentation Time Compare Value Configuration Request Bit

This bit issues requests for configuring the gPTP presentation time comparison register (ETNBnGPTC).

Only 1 can be written to the bit.

Writing to this bit is only possible when the current operating mode is configuration mode and ETNBnCCC.GAC is 0.

Do not write 1 to this bit when AVTP FIFO of selected compare unit is in use.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode and ETNBnCCC.GAC is 0.
– The bit is set to 0 when the value of the gPTP presentation time comparison register (ETNBnGPTC) is loaded.

LTI: Timer Increment Value Configuration Request Bit

This bit issues requests for configuring the gPTP timer increment configuration register (ETNBnGTI).

Only 1 can be written to the bit.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode and ETNBnCCC.GAC is 0.
– The bit is set to 0 when the value of the gPTP timer increment configuration register (ETNBnGTI) is loaded.

LTO: Timer Offset Value Configuration Request Bit

This bit issues requests for configuring gPTP timer offset configuration register i (ETNBnGTOi).

Only 1 can be written to the bit.

[Conditions for Changing]

– The bit is set to 0 when the operating mode is not operation mode and ETNBnCCC.GAC is 0.
– The bit is set to 0 when the value of gPTP timer offset configuration register i (ETNBnGTOi) is loaded.

TCR[1:0]: Timer Control Request Bits

These bits issue requests for controlling the gPTP timer.

The source selection (ETNBnGCCR.TCSS) can be done by same write access.

Writing to the bits is only possible when the current operating mode is operation mode, or when the current operation mode is configuration mode and ETNBnCCC.GAC is 1.

Do not write to the bits when the gPTP timer clock select bit in the AVB-DMAC mode register is 00_B.

Writing to the bits is only possible when ETNBnGCCR.TCR is 00_B or 10_B.

Do not write other values than 11_B to the bits when ETNBnGCCR.TCR is 10_B.

Continuous capture (10_B) can be used only for AVTP presentation time value (ETNBnGCCR.TCSS is 10_B).

[Conditions for Changing]

– The bits are set to 00_B when the operating mode is not operation mode and ETNBnCCC.GAC is 0.
– The bits are set to 00_B on completion of the requested processing.

25.3.2.49 ETNBnRIE0 — Reception Interrupt Enable register 0

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0460_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRS17	FRS16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRS15	FRS14	FRS13	FRS12	FRS11	FRS10	FRS9	FRS8	FRS7	FRS6	FRS5	FRS4	FRS3	FRS2	FRS1	FRS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.59 ETNBnRIE0 register contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	FRSr	Frame Received interrupt Set r 0: No change of ETNBnRIC0.FREr 1: Set ETNBnRIC0.FREr to 1

FRSr: Frame Received interrupt Set r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC0.FREr.
This bit is always read as 0.

25.3.2.50 ETNBnRID0 — Reception Interrupt Disable register 0

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0464_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRD17	FRD16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRD15	FRD14	FRD13	FRD12	FRD11	FRD10	FRD9	FRD8	FRD7	FRD6	FRD5	FRD4	FRD3	FRD2	FRD1	FRD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.60 ETNBnRID0 register contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	FRDr	Frame Received interrupt Disable r 0: No change of ETNBnRIC0.FREr 1: Set ETNBnRIC0.FREr to 0

FRDr: Frame Received interrupt Disable r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC0.FREr.
This bit is always read as 0.

25.3.2.51 ETNBnRIE1 — Reception Interrupt Enable register 1

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0468_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWS	—	—	—	—	—	—	—	—	—	—	—	—	—	RWS17	RWS16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.61 ETNBnRIE1 register contents

Bit Position	Bit Name	Function
31	RFWS	Rx-FIFO Warning interrupt Set 0: No change of ETNBnRIC1.RFWE 1: Set ETNBnRIC1.RFWE to 1
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RWSr	Reception Warning interrupt Set r 0: No change of ETNBnRIC1.RWEr 1: Set ETNBnRIC1.RWEr to 1

RFWS: Rx-FIFO Warning interrupt Set

This bit supports interrupt enable. It controls set of ETNBnRIC1.RFWE.
This bit is always read as 0.

RWSr: Reception Warning interrupt Set r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC1.RWEr.
This bit is always read as 0.

25.3.2.52 ETNBnRID1 — Reception Interrupt Disable register 1

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $046C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFWD	—	—	—	—	—	—	—	—	—	—	—	—	—	RWD17	RWD16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWD15	RWD14	RWD13	RWD12	RWD11	RWD10	RWD9	RWD8	RWD7	RWD6	RWD5	RWD4	RWD3	RWD2	RWD1	RWD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.62 ETNBnRID1 register contents

Bit Position	Bit Name	Function
31	RFWD	Rx-FIFO Warning interrupt Disable 0: No change of ETNBnRIC1.RFWE 1: Set ETNBnRIC1.RFWE to 0
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RWDr	Reception Warning interrupt Set r 0: No change of ETNBnRIC1.RWEr 1: Set ETNBnRIC1.RWEr to 0

RFWD: Rx-FIFO Warning interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnRIC1.RFWE.
This bit is always read as 0.

RWDr: Reception Warning interrupt Disable r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC1.RWEr.
This bit is always read as 0.

25.3.2.53 ETNBnRIE2 — Reception Interrupt Enable register 2

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0470_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFS	—	—	—	—	—	—	—	—	—	—	—	—	—	QFS17	QFS16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFS15	QFS14	QFS13	QFS12	QFS11	QFS10	QFS9	QFS8	QFS7	QFS6	QFS5	QFS4	QFS3	QFS2	QFS1	QFS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.63 ETNBnRIE2 register contents

Bit Position	Bit Name	Function
31	RFFS	Rx-FIFO Full interrupt Set 0: No change of ETNBnRIC2.RFFE 1: Set ETNBnRIC2.RFFE to 1
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	QFSr	Queue Full interrupt Set r 0: No change of ETNBnRIC2.QFEr 1: Set ETNBnRIC2.QFEr to 1

RFFS: Rx-FIFO Full interrupt Set

This bit supports interrupt enable. It controls set of ETNBnRIC2.RFFE.
This bit is always read as 0.

QFSr: Queue Full interrupt Set r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC2.QFEr.
This bit is always read as 0.

25.3.2.54 ETNBnRID2 — Reception Interrupt Disable register 2

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0474_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFFD	—	—	—	—	—	—	—	—	—	—	—	—	—	QFD17	QFD16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	QFD15	QFD14	QFD13	QFD12	QFD11	QFD10	QFD9	QFD8	QFD7	QFD6	QFD5	QFD4	QFD3	QFD2	QFD1	QFD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.64 ETNBnRID2 register contents

Bit Position	Bit Name	Function
31	RFFD	Rx-FIFO Full interrupt Disable 0: No change of ETNBnRIC2.RFFE 1: Set ETNBnRIC2.RFFE to 0
30 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	QFDr	Queue Full interrupt Disable r 0: No change of ETNBnRIC2.QFEr 1: Set ETNBnRIC2.QFEr to 0

RFFD: Rx-FIFO Full interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnRIC2.RFFE.
This bit is always read as 0.

QFDr: Queue Full interrupt Disable r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC2.QFEr.
This bit is always read as 0.

25.3.2.55 ETNBnRIE3 — Reception Interrupt Enable register 3

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0488_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDPS17	RDPS16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDPS15	RDPS14	RDPS13	RDPS12	RDPS11	RDPS10	RDPS9	RDPS8	RDPS7	RDPS6	RDPS5	RDPS4	RDPS3	RDPS2	RDPS1	RDPS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.65 ETNBnRIE3 register contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RDPSr	Receive Descriptor Processed interrupt Set r 0: No change of ETNBnRIC3.RDPEr 1: Set ETNBnRIC3.RDPEr to 1

RDPSr: Receive Descriptor Processed interrupt Set

This bit supports interrupt enable. It controls set of ETNBnRIC3.RDPEr.
This bit is always read as 0.

25.3.2.56 ETNBnRID3 — Reception Interrupt Enable register 3

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $048C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDPD 17	RDPD 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDPD 15	RDPD 14	RDPD 13	RDPD 12	RDPD 11	RDPD 10	RDPD9	RDPD8	RDPD7	RDPD6	RDPD5	RDPD4	RDPD3	RDPD2	RDPD1	RDPD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.66 ETNBnRID3 register contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RDPDr	Receive Descriptor Processed interrupt Disable r 0: No change of ETNBnRIC3.RDPEr 1: Set ETNBnRIC3.RDPEr to 0

RDPDr: Receive Descriptor Processed interrupt Disable r (r = 0 to 17)

This bit supports interrupt enable. It controls set of ETNBnRIC3.RDPEr.
This bit is always read as 0.

25.3.2.57 ETNBnTIE — Transmission Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0478_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	TDPS3	TDPS2	TDPS1	TDPS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MFWS	MFUS	TFWS	TFUS	—	—	—	—	FTS3	FTS2	FTS1	FTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.67 ETNBnTIE Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	TDPSt (t = 3 to 0)	Transmit Descriptor Processed Interrupt Set t 0: No change of ETNBnTIC.TDPEt 1: Set ETNBnTIC.TDPEt to 1
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MFWS	MAC Status FIFO Warning Interrupt Set 0: No change of ETNBnTIC.MFWE 1: Set ETNBnTIC.MFWE to 1
10	MFUS	MAC Status FIFO Updated Interrupt Set 0: No change of ETNBnTIC.MFUE 1: Set ETNBnTIC.MFUE to 1
9	TFWS	Transtamp FIFO Warning Interrupt Set 0: No change of ETNBnTIC.TFWE 1: Set ETNBnTIC.TFWE to 1
8	TFUS	Transtamp FIFO Updated Interrupt Set 0: No change of ETNBnTIC.TFUE 1: Set ETNBnTIC.TFUE to 1
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	FTSt[3:0] (t = 3 to 0)	Frame Transmitted Interrupt Set t 0: No change of ETNBnTIC.FTEt 1: Set ETNBnTIC.FTEt to 1

TDPSt: Transmit Descriptor Processed interrupt Set t (t = 0 to 3)

This bit supports interrupt enable. It controls set of ETNBnTIC.TDPEt.

This bit is always read as 0.

MFWS: MAC status FIFO Warning interrupt Set

This bit supports interrupt enable. It controls set of ETNBnTIC.MFWE.

This bit is always read as 0.

MFUS: MAC status FIFO Updated interrupt Set

This bit supports interrupt enable. It controls set of ETNBnTIC.MFUE.

This bit is always read as 0.

TFWS: Timestamp FIFO Warning interrupt Set

This bit supports interrupt enable. It controls set of ETNBnTIC.TFWE.

This bit is always read as 0.

TFUS: Timestamp FIFO Updated interrupt Set

This bit supports interrupt enable. It controls set of ETNBnTIC.TFUE.

This bit is always read as 0.

FTSt: Frame Transmitted interrupt Set t (t = 0 to 3)

This bit supports interrupt enable. It controls set of ETNBnTIC.FTEt.

This bit is always read as 0.

25.3.2.58 ETNBnTID — Transmission Interrupt Disable Register

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $047C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	TDPD3	TDPD2	TDPD1	TDPD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MFWD	MFUD	TFWD	TFUD	—	—	—	—	FTD3	FTD2	FTD1	FTD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.68 ETNBnTID Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	TDPDt (t = 3 to 0)	Transmit Descriptor Processed interrupt Disable t 0: No change of ETNBnTIC.TDPEt 1: Set ETNBnTIC.TDPEt to 0
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MFWD	MAC status FIFO Warning interrupt Disable 0: No change of ETNBnTIC.MFWE 1: Set ETNBnTIC.MFWE to 0
10	MFUD	MAC status FIFO Updated interrupt Disable 0: No change of ETNBnTIC.MFUE 1: Set ETNBnTIC.MFUE to 0
9	TFWD	Timestamp FIFO Warning interrupt Disable 0: No change of ETNBnTIC.TFWE 1: Set ETNBnTIC.TFWE to 0
8	TFUD	Timestamp FIFO Updated interrupt Disable 0: No change of ETNBnTIC.TFUE 1: Set ETNBnTIC.TFUE to 0
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	FTDt (t = 3 to 0)	Frame Transmitted interrupt Disable t 0: No change of TIC.FTEt 1: Set ETNBnTIC.FTEt to 0

TDPDt: Transmit Descriptor Processed interrupt Disable t (t = 0 to 3)

This bit supports interrupt enable. It controls set of ETNBnTIC.TDPEt.

This bit is always read as 0.

MFWD: MAC status FIFO Warning interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnTIC.MFWE.

This bit is always read as 0.

MFUD: MAC status FIFO Updated interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnTIC.MFUE.

This bit is always read as 0.

TFWD: Timestamp FIFO Warning interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnTIC.TFWE.

This bit is always read as 0.

TFUD: Timestamp FIFO Updated interrupt Disable

This bit supports interrupt enable. It controls set of ETNBnTIC.TFUE.

This bit is always read as 0.

FTDt: Frame Transmitted interrupt Disable t (t = 0 to 3)

This bit supports interrupt enable. It controls set of ETNBnTIC.FTEt.

This bit is always read as 0.

25.3.2.59 ETNBnGMTT — gPTP Maximum Transit Time Configuration Register

The ETNBnGMTT register sets the maximum time for transitions of the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0394_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MTTV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MTTV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.69 ETNBnGMTT register contents

Bit Position	Bit Name	Function
31 to 0	MTTV[31:0]	Maximum Transit Time The maximum transition time for addition to the presentation time

MTTV[31:0]: Maximum Transit Time Bits

These bits set the maximum transition time for use in calculating AVTP presentation times.

Write the desired setting to the bits, then issue the configuration request by setting the maximum transit time configuration request bit in the gPTP configuration control register (ETNBnGCCR.LMTT) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the maximum transit time configuration request bit (ETNBnGCCR.LMTT) is 1.

Do not write a value to these bits when the gPTP Active in Config bit (ETNBnCCC.GAC) is 1 and the maximum transit time configuration request bit (ETNBnGCCR.LMTT) is 1.

25.3.2.60 ETNBnGPTC — gPTP Presentation Time Comparison Register

The ETNBnGPTC register sets a value for comparison with presentation times in the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0398_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PTCV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.70 ETNBnGPTC register contents

Bit Position	Bit Name	Function
31 to 0	PTCV[31:0]	Presentation Time Comparison Value Value for comparison with the gPTP presentation times

PTCV[31:0]: Presentation Time Comparison Value Bits

These bits set a value for comparison with AVTP timer values to which a maximum transit time is not appended.

Write the desired setting to the bits, then issue the configuration request by setting the presentation time comparison value configuration request bit in the gPTP configuration control register (ETNBnGCCR.LPTC) to 1.

CAUTION

Do not write a value to these bits when the presentation time comparison value configuration request bit (ETNBnGCCR.LPTC) is 1. Do not write the range of 0 to 3FFF FFFF$_H$ to the bits when they are defining the period value of AVTP comparison unit (controlled by ETNBnGCCR.SPC).

25.3.2.61 ETNBnGTI — gPTP Timer Increment Configuration Register

The ETNBnGTI register sets the increment for the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $039C_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TIV[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.71 ETNBnGTI register contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 0	TIV[27:0]	gPTP Timer Increment Value Increment for the gPTP timer

TIV[27:0]: Bits (gPTP Timer Increment Value)

When the gPTP clock select bits in the AVB-DMAV mode register (ETNBnCCC.CSEL) are selecting a clock signal, these bits set the value by which the timer is incremented each time a cycle of that clock signal elapses.

Write the desired setting to the bits, then issue the configuration request by setting the timer increment value configuration request bit in the gPTP configuration control register (ETNBnGCCR.LTI) to 1.

CAUTIONS

1. **Do not write a value to these bits when the operating mode is operation mode and the timer increment value configuration request bit (ETNBnGCCR.LTI) is 1. Do not write 0 to the bits.**
2. **Do not write a value to these bits when the gPTP active in config bit (ETNBnCCC.GAC) is 1 and the timer increment value configuration request bit (ETNBnGCCR.LTI) is 1.**
3. **Do not program increment values less than $0010\ 0000_H$ (For calculation, refer to Section 25.4.5.1, Transmission Modes).**

25.3.2.62 ETNBnGTOi — gPTP Timer Offset Configuration Register i (i = 0 to 2)

The ETNBnGTOi register sets an offset value for the gPTP timer.

The offset value is added to the combination of bits 0 to 31 in ETNBnGTO0, 32 to 63 in ETNBnGTO1, and 64 to 79 in ETNBnGTO2, which together make up the gPTP timer.

Access: This register can be read or written in 32-bit units.

Address: ETNBnGTO0: <ETNBn_base> + $03A0_H$
ETNBnGTO1: <ETNBn_base> + $03A4_H$
ETNBnGTO2: <ETNBn_base> + $03A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOV[31 + 32 × i:16 + 32 × i]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOV[15 + 32 × i:0 + 32 × i]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ETNBnGTO2 register corresponds to TOV[79:64] only and bits from 16 to 31 are reserved bits.

Table 25.72 ETNBnGTOi register contents

Bit Position	Bit Name	Function
31 to 0	TOV[95:0]	Timer Offset Value Offset value for the gPTP timer

TOV[79:0]: Timer Offset Value Bits

This is an 80-bit value consisting of the settings in ETNBnGTO0.TOV[31:0], ETNBnGTO1.TOV[63:32], and ETNBnGTO2.TOV[79:64], and is used to set an offset for adding to the value of the gPTP timer.

Write the desired setting to the bits, then issue the configuration request by setting the timer offset value configuration request bit in the gPTP configuration control register (ETNBnGCCR.LTO) to 1.

CAUTIONS

1. **Do not write a value to these bits when the operating mode is operation mode and the timer offset value configuration request bit (ETNBnGCCR.LTO) is 1.**
2. **Write 0000_H to ETNBnGTO2.TOV[95:80].**
3. **Set a value in the range from 0 to 10^9-1 ($0000\ 0000_H$ to $3B9A\ C9FF_H$) in ETNBnGTO0.TOV[31:0].**
4. **Do not write to these bits when ETNBnGCCR.LTO is 1 while ETNBnCCC.GAC is 1.**

25.3.2.63 ETNBnGIC — gPTP Interrupt Control Register

The ETNBnGIC register is used to control gPTP-related interrupts.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $03AC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PTME7	PTME6	PTME5	PTME4	PTME3	PTME2	PTME1	PTME0	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.73 ETNBnGIC register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	PTME7	Presentation Time Match Interrupt Enable 7 0: Disabled 1: Enabled
8	PTME6	Presentation Time Match Interrupt Enable 6 0: Disabled 1: Enabled
7	PTME5	Presentation Time Match Interrupt Enable 5 0: Disabled 1: Enabled
6	PTME4	Presentation Time Match Interrupt Enable 4 0: Disabled 1: Enabled
5	PTME3	Presentation Time Match Interrupt Enable 3 0: Disabled 1: Enabled
4	PTME2	Presentation Time Match Interrupt Enable 2 0: Disabled 1: Enabled
3	PTME1	Presentation Time Match Interrupt Enable 1 0: Disabled 1: Enabled
2	PTME0	Presentation Time Match Interrupt Enable 0 0: Disabled 1: Enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

PTMEi (i = 0 to 7) Presentation Time Match Interrupt Enable Bit

When this bit is 1, setting of the presentation time match interrupt flag in the gPTP interrupt status register (ETNBnGIS.PTMFi) to 1 leads to generation of that interrupt.

25.3.2.64 ETNBnGIS — gPTP Interrupt Status Register

The ETNBnGIS register indicates the state of the gPTP-related interrupt.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $03B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PTMF7	PTMF6	PTMF5	PTMF4	PTMF3	PTMF2	PTMF1	PTMF0	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 25.74 ETNBnGIS register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	PTMF7	Presentation Time Match Interrupt Flag 7 0: The interrupt is not pending. 1: The interrupt is pending.
8	PTMF6	Presentation Time Match Interrupt Flag 6 0: The interrupt is not pending. 1: The interrupt is pending.
7	PTMF5	Presentation Time Match Interrupt Flag 5 0: The interrupt is not pending. 1: The interrupt is pending.
6	PTMF4	Presentation Time Match Interrupt Flag 4 0: The interrupt is not pending. 1: The interrupt is pending.
5	PTMF3	Presentation Time Match Interrupt Flag 3 0: The interrupt is not pending. 1: The interrupt is pending.
4	PTMF2	Presentation Time Match Interrupt Flag 2 0: The interrupt is not pending. 1: The interrupt is pending.
3	PTMF1	Presentation Time Match Interrupt Flag 1 0: The interrupt is not pending. 1: The interrupt is pending.
2	PTMF0	Presentation Time Match Interrupt Flag 0 0: The interrupt is not pending. 1: The interrupt is pending.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

PTMFi (i = 0 to 7) Presentation Time Match Interrupt Flag Bit

This bit indicates that the AVTP timer value has matched the configured comparison value of AVTP comparator i. Only 0 can be written to the bit.

[Conditions for Changing]

– This bit is set to 0 when the current operating mode changes from operation mode and ETNBnCCC.GAC is 0.
– This bit is set to 1 when the AVTP timer value reaches or exceeds the comparison value of AVTP comparator i.

25.3.2.65 ETNBnGCTi — gPTP Timer Capture Register i (i = 0 to 2)

The ETNBnGCTi registers form an 80-bit register that captures the gPTP timer value.

Access: This register can be read or written in 32-bit units.

Address: ETNBnGCT0: <ETNBn_base> + $03B8_H$
ETNBnGCT1: <ETNBn_base> + $03BC_H$
ETNBnGCT2: <ETNBn_base> + $03C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CTV[31 + 32 × i:16 + 32 × i]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTV[15 + 32 × i:0 + 32 × i]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ETNBnGCT2 register corresponds to CTV[79:64] only and bits from 16 to 31 are reserved bits.

Table 25.75 ETNBnGCTi register contents

Bit Position	Bit Name	Function
31 to 0	CTV[95:0]	gPTP Timer Capture Value Captured timer value

CTV[79:0]: gPTP Timer Capture Value Bits

These 80 bits consist of ETNBnGCT0.CTV[31:0], ETNBnGCT1.CTV[63:32] and ETNBnGCT2.CTV[79:64], which together indicate captured timer values.

When 00_B (value of the gPTP timer) or 01_B (adjusted gPTP timer value) is selected by the timer capture source select bits in the gPTP configuration control register (ETNBnGCCR.TCSS), the corresponding 80-bit values are stored in these bits.

When 10_B (AVTP presentation time) is selected by the timer capture source select bits (ETNBnGCCR.TCSS), the corresponding 32-bit values are stored in these bits (ETNBnGCT0.CTV[31:0]).

In case of continuous update (ETNBnGCCR.TCR is 10_B) CPU should only use 32-bit read access to get consistent value.

Do not read the value while the value of the timer control request bits (ETNBnGCCR.TCR) is 11_B because this indicates that storage is still in progress.

[Condition for Changing]
These bits are updated to the selected timer value with the capture request has been processed.

CAUTION

Write 0000_H to ETNBnGCT2.CTV[95:80].

25.3.2.66 ETNBnGSR — gPTP Status Register

The ETNBnGSR register indicates the status of the gPTP.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $03C4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	AFU3	AFU2	AFU1	AFU0	AFFL3[3:0]				AFFL2[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFFL1[3:0]				AFFL0[3:0]				PCM7	PCM6	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.76 ETNBnGSR register contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned.
27	AFU3	AVTP FIFO Update 3 0: No update of AVTP FIFO pending 1: Update of AVTP FIFO with value from ETNBnGPTFi.PTFV is ongoing
26	AFU2	AVTP FIFO Update 2 0: No update of AVTP FIFO pending 1: Update of AVTP FIFO with value from ETNBnGPTFi.PTFV is ongoing
25	AFU1	AVTP FIFO Update 1 0: No update of AVTP FIFO pending 1: Update of AVTP FIFO with value from ETNBnGPTFi.PTFV is ongoing
24	AFU0	AVTP FIFO Update 0 0: No update of AVTP FIFO pending 1: Update of AVTP FIFO with value from ETNBnGPTFi.PTFV is ongoing
23 to 20	AFFL3	AVTP FIFO Fill Level 3 FIFO fill level of AVTP compare unit 3
19 to 16	AFFL2	AVTP FIFO Fill Level 2 FIFO fill level of AVTP compare unit 2
15 to 12	AFFL1	AVTP FIFO Fill Level 1 FIFO fill level of AVTP compare unit 1
11 to 8	AFFL0	AVTP FIFO Fill Level 0 FIFO fill level of AVTP compare unit 0
7	PCM7	Periodic Comparison Mode 7 0: Single shot comparison 1: Periodic comparison
6	PCM6	Periodic Comparison Mode 6 0: Single shot comparison 1: Periodic comparison
5	PCM5	Periodic Comparison Mode 5 0: Single shot comparison 1: Periodic comparison
4	PCM4	Periodic Comparison Mode 4 0: Single shot comparison 1: Periodic comparison

Table 25.76 ETNBnGSR register contents (2/2)

Bit Position	Bit Name	Function
3	PCM3	Periodic Comparison Mode 3 0: Single shot comparison 1: Periodic comparison
2	PCM2	Periodic Comparison Mode 2 0: Single shot comparison 1: Periodic comparison
1	PCM1	Periodic Comparison Mode 1 0: Single shot comparison 1: Periodic comparison
0	PCM0	Periodic Comparison Mode 0 0: Single shot comparison 1: Periodic comparison

AFUi (i = 0 to 3): AVTP FIFO Update Bit

This bit indicates an ongoing update of the AVTP-FIFO of compare unit i. When this bit is 0, the AVTP-FIFO is able to process adding new values.

[Clearing conditions]

- The current operating mode changes from operation mode and ETNBnCCC.GAC is 0.
- AVTP compare value is added to the AVTP-FIFO.

[Setting condition]

- CPU writes a new value to ETNBnGPTFi.PTFV and ETNBnGSR.AFFLi is not 15.

AFFLi[3:0] (i = 0 to 3): AVTP FIFO Fill Level Bits

These bits indicate the number of AVTP timestamps pending in AVTP-FIFO of compare unit i. The timestamps are provided by CPU and taken by compare unit when no comparison on AVTP comparator i is active.

[Clearing conditions]

- The current operating mode changes from operation mode and ETNBnCCC.GAC is 0.
- AVTP compare value is added to the AVTP-FIFO.

[Conditions for Updating]

- This value is incremented when a new value is added to AVTP-FIFO (ETNBnGSR.AFUi changes to 0).
- This value is decremented when a timestamp value from FIFO has been loaded to the AVTP comparator.

PCMi (i = 0 to 7): Periodic Comparison Mode Bit

This bit indicates if single shot or periodic comparison mode is active on AVTP comparator unit i.

[Clearing conditions]

- The current operating mode changes from operation mode and ETNBnCCC.GAC is 0.
- The last match has happen after CPU has requested entering single shot mode.

[Setting condition]

- A period value for the AVTP comparator is loaded.

25.3.2.67 ETNBnGPTFi — gPTP Presentation Time FIFO Register i (i = 0 to 3)

The ETNBnGPTFi register is used to control the gPTP FIFO.

Access: This register can be read or written in 32-bit units.

Address: ETNBnGPTF0: <ETNBn_base> + $03E0_H$
ETNBnGPTF1: <ETNBn_base> + $03E4_H$
ETNBnGPTF2: <ETNBn_base> + $03E8_H$
ETNBnGPTF3: <ETNBn_base> + $03EC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PTFV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTFV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.77 ETNBnGPTFi Register Contents

Bit Position	Bit Name	Function
31 to 0	PTFV[31:0]	Presentation Time FIFO Value

PTFV[31:0]: Presentation Time FIFO Value Bits

These bits define a value to be compared with the AVTP timer value (without Max Transit Time added). When writing to this register the value is added to AVTP-FIFO of comparator unit i.
These bits are always read as 0.
The CPU cannot write to these bits if the current operating mode is configuration mode and ETNBnCCC.GAC is 0.

The CPU should not write to these bits when ETNBnGSR.PCMi is 1.
The CPU should not write to these bits when ETNBnGSR.AFUi is 1.
The CPU should not write to these bits when ETNBnGSR.AFFLi is 15.

25.3.2.68 ETNBnECMR — E-MAC Mode Register

ETNBnECMR is used to specify the operating mode of the E-MAC. The settings in this register are normally made in the initialization process following a reset.

The operating mode settings must not be changed while the transmission and reception functions are enabled (i.e. while the RE or TE bit in this register is 1).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0500_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MPDE	—	—	RE	TE	—	—	—	DM	PRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R/W

Table 25.78 ETNBnECMR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TRCCM	Counter Clear Mode 0: Writing to the counter register leads to the register being cleared to 0. 1: Reading from the counter register leads to the register being cleared to 0.
25, 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	RCSC	Checksum Calculation 0: Checksums are not automatically calculated. 1: Checksums are automatically calculated.
22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	DPAD	Data Padding 0: Padding to make up 60 bytes is inserted in data for transmission when fewer than 60 bytes are to be transmitted. 1: Padding is not inserted in data for transmission when fewer than 60 bytes are to be transmitted and the data are transmitted without being changed.
20	RZPF	PAUSE Frame Reception with Time = 0 0: Reception of PAUSE frames with the TIME parameter value 0 is disabled. 1: Reception of PAUSE frames with the TIME parameter value 0 is enabled.
19	ZPF	PAUSE Frame Usage with TIME = 0 Enable • PAUSE frame usage with TIME = 0 enable (in full-duplex mode) 0: Control in response to and for the sending of PAUSE frames with the TIME parameter value 0 is disabled. 1: Control in response to and for the sending of PAUSE frames with the TIME parameter value is 0 is enabled.
18	PFR	PAUSE Frame Receive Mode 0: PAUSE frames are not transferred to the AVB-DMAC. 1: PAUSE frames are transferred to the AVB-DMAC.

Table 25.78 ETNBnECMR Register Contents (2/2)

Bit Position	Bit Name	Function
17	RXF	Reception Flow Control Operation Mode 0: Flow control for the receiving port (reception of PAUSE frame) is disabled. 1: Flow control for the receiving port (reception of PAUSE frame) is enabled.
16	TXF	Transmission Flow Control Operation Mode 0: Flow control for the transmitting port is disabled (PAUSE frames are not automatically transmitted). 1: Flow control for the transmitting port is enabled (PAUSE frames are automatically transmitted as required).
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	MPDE	Magic Packet™ Detection Enable 0: Magic Packet™ detection is not enabled. 1: Magic Packet™ detection is enabled.
8, 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	RE	Reception Enable 0: Reception is disabled. 1: Reception is enabled.
5	TE	Transmission Enable 0: Transmission is disabled. 1: Transmission is enabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DM	Mode selection 0: Half-duplex mode 1: Full-duplex mode Setting this bit to 0 is prohibited
0	PRM	Promiscuous Mode 0: Normal operation 1: Promiscuous mode operation

TRCCM: Counter Clear Mode Bit

This bit sets the method for clearing the counter register. Refer to the descriptions of the counter registers.

RCSC: Checksum Calculation Bit

Setting this bit to 1 enables automatic calculation of checksums for the data field in received frames.

Only the data field of an Ethernet frame is in the scope of checksum calculation. Specifically, the checksum is calculated from the data field, which follows the length/type field and is followed by the CRC field. Calculation only involves 16-bit addition; it does not involve bit inversion.

DPAD: Data Padding Control Bit

This bit specifies padding or non-padding of data when less than 60 bytes are to be transmitted.

When this bit is set to 1, data are transmitted without padding; when it is set to 0, data are padded to make up 60-byte units for transmission.

RZPF: PAUSE Frame Reception with Time = 0 Bit

This bit is set to 0, received PAUSE frames with the Timer value 0 are discarded.

This bit is set to 1, release from the transmission wait state follows reception of a PAUSE frame with the Timer value 0.

ZPF: PAUSE Frame Usage with TIME = 0 Enable

- PAUSE frame usage with TIME = 0 enable (In full-duplex mode)

When this bit is set to 0, the next frame to be transmitted is not transmitted until the time specified by the Timer value has elapsed.

Received PAUSE frames with the Timer value 0 are discarded.

When this bit is set to “1”, if the amount of data in the reception FIFO becomes less than the setting of the receive FIFO warning level bits in the receive configuration register (ETNBnRCR.RFCL) before the time specified by the Timer value elapses, a PAUSE frame with a Timer value of 0 is automatically transmitted. If the interface is in the transmission wait state, it is released from that state on receiving a PAUSE frame with a Timer value of 0.

PFR: PAUSE Frame Receive Mode Bit

This bit specifies whether PAUSE frames are transferred to the AVB-DMAC.

RXF: Operating Mode for Flow Control in Reception Bit

This bit is set to 1 and a PAUSE frame is received, a next frame to be transmitted is not transmitted until the time indicated by the Timer value in the PAUSE frame has elapsed. However, the transmission of a current frame is continued. The number of received PAUSE frames is also counted. For details, see **Section 25.3.2.77, ETNBnPFRCR — PAUSE Frame Receive Counter**.

Setting this bit to 0 disables PAUSE frame detection.

TXF: Operating Mode for Flow Control in Transmission Bit

This bit enables or disables flow control in transmission.

Setting this bit to 0 disables PAUSE frame detection.

MPDE: Magic Packet™ Detection Enable Bit

Then MPDE bit enables or disables Magic Packet™ detection by hardware to allow activation via the Ethernet connection.

RE: Reception Enable Bit

When this bit is switched from the receive function enabled (RE = 1) to the receive function disabled (RE = 0), and if there is a frame is being received, the receive function remains enabled until the reception of the frame finishes.

TE: Transmission Enable Bit

When this bit is switched from the transmit function enabled (TE = 1) to the transmit function disabled (TE = 0), and if there is a frame being transmitted, the transmit function remains enabled until the transmission of the frame finishes.

DM: Duplex Mode Bit

This bit selects full- or half-duplex operation.

PRM: Promiscuous Mode Bit

This bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.).

25.3.2.69 ETNBnRFLR — Receive Frame Length Register

The ETNBnRFLR register specifies the maximum length (in bytes) of frames that can be received by this LSI.

This register must not be changed while reception is enabled (while the reception enable bit (ETNBnECMR.RE) in the E-MAC mode register is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0508_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.79 ETNBnRFLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	RFL[17:0]	Receive Frame Length Setting value Check value 00000_H: 1,518 bytes : : $005EE_H$: 1,518 bytes $005EF_H$: 1,519 bytes $005F0_H$: 1,520 bytes : : $1FFFF_H$: 131,071 bytes 20000_H: 131,072 bytes : : $3FFFF_H$: 131,072 bytes

RFL[17:0]: Receive Frame Length Bits

Frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data are not included in the transfer. When more data than the specified number of bytes are received, the portion of data that exceeds the specified value is discarded.

CAUTIONS

The prepared descriptor data size is just the specified value (ETNBnRFLR.RFL[17:0]). Therefore descriptor data size must be more than ETNBnRFLR.RFL[17:0] if you will receive such the Too-Long Frame.

25.3.2.70 ETNBnECSR — E-MAC Status Register

The ETNBnECSR register indicates the state of the E-MAC. The CPU can be notified of the state. For bits that generate an interrupt, the interrupt can be enabled or disabled by the corresponding bit in the E-MAC Interrupt Permission Register (ETNBnECSIPR) described in **Section 25.3.2.71, ETNBnECSIPR — E-MAC Interrupt Permission Register**.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0510_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PHYI	LCHNG	MPD	ICD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.80 ETNBnECSR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PHYI	PHY interrupt terminal state bit 0: PHY interrupt terminal (AVB_PHY_INT) is not asserted. 1: PHY interrupt terminal (AVB_PHY_INT) is asserted.
2	LCHNG	Link signal change bit 0: Change of Link status signal (AVB_LINK) is not detected. 1: Change of Link status signal (AVB_LINK) is detected.
1	MPD	Magic Packet™ Detection 0: Magic Packet™ has not been detected. 1: Magic Packet™ has been detected.
0	ICD	Illegal Carrier Detection 0: PHY-LSI has not detected an illegal carrier on the line. 1: PHY-LSI has detected an illegal carrier on the line.

PHYI: PHY Interrupt Pin State Bit

This bit indicates the PHY interrupt pin (AVB_PHY_INT), which is input from the PHY-LSI.

LCHNG: Link Signal Change Bit

This bit indicates a transition of the link status signal (AVB_LINK) input from the PHY-LSI from high to low or low to high.

However, signal changes may also be detected at times when the link status signal (AVB_LINK) function is selected.

To check the current link state, refer to the link status pin state bit in the PHY status register (ETNBnPLSR.LINK).

Writing 1 to this bit clears it to 0.

MPD: Magic Packet™ Detection Bit

This bit indicates that a Magic Packet™ has been detected on the line.

Writing 1 to this bit clears it to 0.

ICD: Illegal Carrier Detection Bit

This bit indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used.

Writing 1 to this bit clears it to 0.

25.3.2.71 ETNBnECSIPR — E-MAC Interrupt Permission Register

The ETNBnECSIPR register instructs permission of interrupt sources reported by the ETNBnECSR register. The bits can permit interrupts corresponding to the ETNBnECSR bits.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0518_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	PHYIM	LINKIM	MPDIP	ICDIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.81 ETNBnECSIPR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PHYIM	PHY Interrupt Mask 0: Interrupts by the PHYI setting prohibited. 1: Interrupts by the PHYI setting permitted.
2	LINKIM	LINK Interrupt Mask 0: Interrupts by the LINKI setting prohibited. 1: Interrupts by the LINKI setting permitted.
1	MPDIP	Magic Packet™ Detect Interrupt Enable 0: Interrupts on setting of the MPD bit is disabled. 1: Interrupts on setting of the MPD bit is enabled.
0	ICDIP	False Carrier Detect Interrupt Enable 0: Interrupts by the ICD bit setting prohibited. 1: Interrupt by the ICD bit setting permitted.

PHYIM: PHY Interrupt Mask Bit

When this bit is set to “1”, an interrupt occurs.

LINKIM: Link Signal Change Interrupt Enable Bit

When this bit is set to “1” and the link signal change bit (ETNBnECSR.LCHNG) in the E-MAC status register is set to “1”, an interrupt occurs.

MPDIP: Magic Packet™ Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the Magic Packet™ detection bit (ETNBnECSR.MPD) in the E-MAC status register to 1.

ICDIP: Illegal Carrier Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the illegal carrier detection bit (ETNBnECSR.ICD) in the E-MAC status register to 1.

25.3.2.72 ETNBnPIR — PHY Interface Register

The ETNBnPIR register provides a means of access to the PHY-LSI internal registers via the MII.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0520_H

Value after reset: 0000 $000X_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	—	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 25.82 ETNBnPIR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	MDI	MII Management Data-In Indicates the level of the ETNBnMDIO pin.
2	MDO	MII Management Data-Out Stores data to output from the ETNBnMDIO pin.
1	MMD	MII Management Mode 0: Read direction is specified. 1: Write direction is specified.
0	MDC	MII Management Data Clock The value set in this bit is output from the ETNBnMDC pin, which supplies the management data clock for the MII.

MDI: MII Management Data-In Bit

This bit indicates the level of the ETNBnMDIO pin.

MDO: MII Management Data-Out Bit

This bit stores data to output from the ETNBnMDIO pin.

The ETNBnMDIO pin outputs data when the MMD bit is set to 1 (to specify writing as the direction). Data are not output while the MMD bit is set to 0 (to specify reading as the direction).

MMD: MII Management Mode Bit

This bit specifies the direction for data through MDIO (reading or writing).

MDC: MII Management Data Clock Bit

Values set in this bit are output on the ETNBnMDC pin to supply the MII with the management data clock. For the method of access to the MII registers, see **Section 25.4.13, Connection to PHY-LSI**.

25.3.2.73 ETNBnPLSR — PHY LINK Status Register

The ETNBnPLSR register is used to check the status of PHY LINK pins.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0528_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LINK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.83 ETNBnPLSR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	LINK	When read, this bit returns the state of the ETNBnLINKSTA pin.

25.3.2.74 ETNBnAPFTP — Auto PAUSE Frame Time Parameter Register

The ETNBnAPFTP register is used to set the value for the TIME parameter of automatically generated PAUSE frame.

When a PAUSE frame is automatically transmitted, the value set in this register is used as its TIME parameter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0554_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APFTP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.84 ETNBnAPFTP Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	APFTP[15:0]	Automatic PAUSE Frame Time Parameter Set a time parameter value of the Automatic PAUSE Frame.*1*2 0000_H: Setting prohibited 0001_H: 1 × 512 bit-period 0002_H: 2 × 512 bit-period : : $FFFF_H$: 65535 × 512 bit-period

Note 1. The bit-period changes relative to the transfer speed.
100 Mbps: 1 bit-period = 10ns
10 Mbps: 1 bit-period = 100ns

Note 2. When setting the Transmission flow control operation mode bit (ETNBnECMR.TXF) in the E-MAC mode register to “1”, set this register value other than $0000\ 0000_H$.

APFTP[15:0]: Auto Pause Frame Time Parameter

These bit configure the time parameter value for the transmit of Auto Pause Frame.

The unit of the setting value is 512 bit-period.

25.3.2.75 ETNBnMPR — Manual PAUSE Frame Register

The ETNBnMPR register is used to set the value for the TIME parameter of manually generated PAUSE frames. When a PAUSE frame is manually transmitted, the value set in this register is used as its TIME parameter.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0558_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.85 ETNBnMPR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MP[15:0]	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame.*1 0000_H: Setting prohibited 0001_H: 1 × 512 bit-period 0002_H: 2 × 512 bit-period :: $FFFF_H$: 65535 × 512 bit-period

Note 1. The bit-period changes relative to the transfer speed.
100 Mbps: 1 bit-period = 10ns
10 Mbps: 1 bit-period = 100ns

MP[15:0]: Manual PAUSE Bits

These bits set the value of the TIME parameter in manually generated PAUSE frames.

The unit for the setting is 512 bit periods.

25.3.2.76 ETNBnPFTCR — PAUSE Frame Transmit Counter

The ETNBnPFTCR register is a counter that indicates the number of times PAUSE frames have been transmitted.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + $055C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFTXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.86 ETNBnPFTCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	PFTXC[15:0]	PAUSE Frame Transmit Counter These bits indicates the number of transmitted PAUSE frames.

PFTXC[15:0]: PAUSE Frame Transmit Counter Bits

These bits indicate the total number of PAUSE frames that have been transmitted (both manually and automatically).

The bits are cleared to 0 when they are read.

If counting up and clearing of the counter coincide, clearing the counter takes priority.

25.3.2.77 ETNBnPFRCR — PAUSE Frame Receive Counter

The ETNBnPFRCR register is a counter that indicates the number of times PAUSE frames have been received.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0560_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFRXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.87 ETNBnPFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	PFRXC[15:0]	PAUSE Frame Receive Counter Counter for counting the number of received PAUSE frames

PFRXC[15:0]: PAUSE Frame Receive Counter Bits

These bits indicate the number of PAUSE frames that have been received when flow control in reception is enabled (the RXF bit in ETNBnECMR = 1).

The bits are cleared to 0 when they are read.

If counting up and clearing the counter coincide, clearing the counter takes priority.

25.3.2.78 ETNBnMAHR — MAC Address High Register

The ETNBnMAHR register specifies the 32 higher-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

This register must not be changed while transmission or reception is enabled (E-MAC mode register reception enable bit (ETNBnECMR.RE) is set to “1” or transmission enable bit (ETNBnECMR.TE) is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $05C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.88 ETNBnMAHR Register Contents

Bit Position	Bit Name	Function
31 to 0	MA[47:16]	MAC Address Bits 47 to 16 These bits are used to set the 32 higher-order bits of the MAC address.

MA[47:16]: MAC Address Bits 47 to 16

These bits are used to set the 32 higher-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set $0123\ 4567_H$ in the this register.

25.3.2.79 ETNBnMALR — MAC Address Low Register

The ETNBnMALR register specifies the 16 lower-order bits of the 48-bit MAC address. The settings in this register are normally made in the initialization process after a reset.

This register must not be changed while transmission or reception is enabled (E-MAC mode register reception enable bit (ETNBnECMR.RE) is set to “1” or transmission enable bit (ETNBnECMR.TE) is set to “1”).

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $05C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.89 ETNBnMALR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MA[15:0]	MAC Address Bits 15 to 0 These bits are used to set the 16 lower-order bits of the MAC address.

MA[15:0]: MAC Address Bits 15 to 0

These bits are used to set the 16 lower-order bits of the MAC address.

For example, if the MAC address is 01-23-45-67-89-AB (hexadecimal), set $89AB_H$ in the ETNBnMALR register.

25.3.2.80 ETNBnCEFCR — CRC Error Frame Receive Counter Register

The ETNBnCEFCR register is a counter that indicates the number of times frames with CRC errors were received. Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0740_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CEFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.90 ETNBnCEFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CEFC[15:0]	CRC Error Frame Counter These bits indicate the number of CRC error frames received.

CEFC[15:0]: CRC Error Frame Counter Bits

These bits indicate the number of received frames having CRC errors.

The bits are cleared to “0” when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to “1”.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.81 ETNBnFRECR — Frame Receive Error Counter Register

The ETNBnFRECR register is a counter that indicates the number of frame receive errors were generated by input on the ETNBnRXER pin from the PHY-LSI. Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0748_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FREC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.91 ETNBnFRECR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	FREC[15:0]	Frame Receive Error Counter These bits indicate the number of errors during frame reception.

FREC[15:0]: Frame Receive Error Counter Bits

These bits indicate the number of errors during frame reception.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.82 ETNBnTSFRCR — Too-Short Frame Receive Counter Register

The ETNBnTSFRCR register is a counter that indicates the number of received frames that were fewer than 64 bytes in length. Counting stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0750_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSFRC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.92 ETNBnTSFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	TSFRC[15:0]	Too-Short Frame Receive Counter These bits indicate the number of frames received with a length of less than 64 bytes.

TSFRCR[15:0]: Too-Short Frame Receive Counter Bits

These bits indicate the number of received frames that were fewer than 64 bytes in length.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.83 ETNBnTLFRCR — Too-Long Frame Receive Counter Register

The ETNBnTLFRCR register is a counter that indicates the number of received frames that were longer than the value specified in the receive frame length register (ETNBnRFLR). Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0758_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TLFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.93 ETNBnTLFRCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	TLFC[15:0]	Too-Long Frame Receive Counter These bits indicate the number of frames received with a length exceeding the value in ETNBnRFLR.

TLFRCR[15:0]: Too-Long Frame Receive Counter Bits

These bits indicate the number of received frames that were longer than the value in ETNBnRFLR.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.84 ETNBnRFCR — Residual-Bit Frame Receive Counter Register

The ETNBnRFCR register is a counter that indicates the number of received frames containing "residual bits" (trailing bits not making up an 8-bit unit). Counting up stops when the value in this register reaches 0000 $FFFF_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0760_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.94 ETNBnRFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	RFC[15:0]	Residual-Bit Frame Receive Counter These bits indicate the number of received frames containing residual bits.

RFC[15:0]: Residual-Bit Frame Receive Counter Bits

These bits indicate the number of received frames containing residual bits.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.85 ETNBnMAFCR — Multicast Address Frame Receive Counter Register

The ETNBnMAFCR register is a counter that indicates the number of received frames for which a multicast address was specified. Counting up stops when the value in this register reaches 0000 FFFF$_H$.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0778$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.95 ETNBnMAFCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	MAFC[15:0]	Multicast Address Frame Counter These bits indicate the number of multicast frames that have been received.

MAFC[15:0]: Multicast Address Frame Counter Bits

These bits indicate the number of multicast frames that have been received.

The bits are cleared to 0 when they are read while the counter clear mode bit (ETNBnECMR.TRCCM) in the E-MAC mode register is set to 1.

When ETNBnECMR.TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

25.3.2.86 ETNBnPIPR — PHY Interrupt Polarity Register

The ETNBnPIPR register is used to set the active sense of AVB_PHY_INT pin.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $052C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PHYIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.96 ETNBnPIPR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PHYIP	PHY Interrupt Input Pin Polarity This bit sets the active sense of the PHY interrupt pin (AVB_PHY_INT). For the active sense, refer to the specifications of the PHY-LSI to be connected. 0: PHY interrupt pin (AVB_PHY_INT) is active low (the low level triggers the interrupt state) 1: PHY interrupt pin (AVB_PHY_INT) is active high (the high level triggers the interrupt state)

NOTE

Only ETNB1 (Gigabit Ethernet) supports this function.

25.3.2.87 ETNBnTPAUSER — Automatic PAUSE Frame Retransmit Count Register

The ETNBnTPAUSER register is used to set the value for upper limit number of auto generated PAUSE frames.

The setting in this register must not be changed while transmission is enabled.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + 0564_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFRTULMT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.97 ETNBnTPAUSER Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PFRTULMT[15:0]	Pause Frame Retry Upper Limit These bits set the value for upper limit time of auto generated PAUSE frames while Receive FIFO caution. For the active sense, refer to the specifications of the PHY-LSI to be connected. 0000_B: unlimited 0001_B: 1 times 0002_B: 2 times : $FFFF_B$: 65535 times

25.3.2.88 ETNBnPFTTCR — PAUSE Frame Retransmit Counter Register

The ETNBnPFTTCR register is counter that indicates the number of times auto PAUSE frames have been transmit.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ETNBn_base> + 0568_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFRTC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.98 ETNBnPFTTCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	PFRTC[15:0]	Pause Frame Retry Counter These bits indicate the number of auto PAUSE frames that have been transmit while Receive FIFO caution. The bits are cleared to 0 when they are read and receive FIFO caution start.

25.3.2.89 ETNBnGECMR — E-MAC Mode Register 2

The ETNBnGECMR register specifies the operating mode for the E-MAC.

The setting in the ETNBnGECMR register must not be changed while transmission or reception is enabled.

Access: This register can be read or written in 32-bit units.

Address: <ETNBn_base> + $05B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.99 ETNBnGECMR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SPEED	Transfer Speed Setting This bit sets the transfer rate. 0: Transfer is at 100 Mbps/10 Mbps 1: Transfer is at 1000 Mbps For ETNB0, setting this bit to 1 is prohibited.

NOTE

For ETNB0, setting SPEED = 1 is not allowed.

25.3.3 SGMII Interface Related Registers

25.3.3.1 ETNB1SGOPMC — Operation Mode Configuration Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + 1000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SPEED[1:0]		DUPLEX	MODE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 25.100 ETNB1SGOPMC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	SPEED[1:0]	Transfer Rate 00_B: 10 Mbps 01_B: 100 Mbps 10_B: 1000 Mbps 11_B: Setting prohibited The value of these bits is valid only when MODE = 1.
1	DUPLEX	Transfer Mode 0: Half-duplex mode (Setting prohibited) 1: Full-duplex mode The value of this bit is valid only when MODE = 1.
0	MODE	Operation Mode 0: Auto-Negotiation process with PHY-LSI is not bypassed. 1: Auto-Negotiation process with PHY-LSI is bypassed.

25.3.3.2 ETNB1SGOPMS — Operation Mode Status Register

Access: This register is a read-only register that can be read in 32-bit units

Address: <ETNB1_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XMIT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	LINK	SPEED[1:0]		DUPLEX	MODE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.101 ETNB1SGOPMS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17, 16	XMIT[1:0]	Internal Operation State 00_B: Idle state 01_B: Configuration state 10_B: Data Communication state 11_B: Reserved SGMII interface can perform data communication only when XMIT[1:0] = 10_B (Data communication state). When ETNB1SGOPMC.MODE = 1, these bits are fixed to 10_B. When ETNB1SGOPMC.MODE = 0, these bits change to 10_B after Auto-Negotiation with PHY-LSI is completed.
15 to 5	Reserved	When read, the value after reset is returned.
4	LINK	Link State This bit indicates the link state reported from the PHY-LSI. 0: Link-down 1: Link-up This bit is fixed to 1 when ETNB1SGOPMC.MODE = 1.
3, 2	SPEED[1:0]	Transfer Rate 00_B: 10 Mbps 01_B: 100 Mbps 10_B: 1000 Mbps 11_B: Reserved When ETNB1SGOPMC.MODE = 1, these bits indicate the value set in ETNB1SGOPMC.SPEED[1:0]. When ETNB1SGOPMC.MODE = 0, these bits indicate the value determined by Auto-Negotiation with PHY-LSI.
1	DUPLEX	Transfer Mode 0: Half-duplex mode (Setting prohibited) 1: Full-duplex mode When ETNB1SGOPMC.MODE = 1, these bits indicate the value set in ETNB1SGOPMC.DUPLEX. When ETNB1SGOPMC.MODE = 0, these bits indicate the value determined by Auto-Negotiation with PHY-LSI
0	MODE	Operation Mode 0: Auto-Negotiation process with PHY-LSI is not bypassed. 1: Auto-Negotiation process with PHY-LSI is bypassed. This bit indicates the value set in ETNB1SGOPMC.MODE.

25.3.3.3 ETNB1SGSRST — Software Reset Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + 1008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 25.102 ETNB1SGSRST Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRST	Software Reset for SGMII Interface Related Registers 0: Reset Release 1: Reset This bit does not reset this register itself nor the ETNB1SGSDS, ETNB1SGCLKSEL, ETNB1SGRCIE registers.

25.3.3.4 ETNB1SGINTS — Interrupt Status Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + $100C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URDYI	RDYI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.103 ETNB1SGINTS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	URDYI	Un-ready Interrupt 0: No interrupt 1: Indicates that the SGMII interface has changed a state in which it can perform data communication to one in which it cannot. (The value of ETNB1SGOPMS.XMIT[1:0] has changed from 10_B to another value.) This bit is cleared to 0 by writing 1 to this bit.
0	RDYI	Ready Interrupt 0: No interrupt 1: Indicates that the SGMII interface has changed a state in which it cannot perform data communication to one in which it can. (The value of ETNB1SGOPMS.XMIT[1:0] has changed from a value other than 10_B to 10_B.) This bit is cleared to 0 by writing 1 to this bit.

NOTE

This Register is valid only when ETNB1SGOPMC.MODE = 0.

25.3.3.5 ETNB1SGINTM — Interrupt Mask Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + 1010_H

Value after reset: $0000\ 0003_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	URDYIM	RDYIM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 25.104 ETNB1SGINTM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	URDYIM	Un-ready Interrupt Mask 0: Enables un-ready interrupt (URDYI) 1: Disables un-ready interrupt (URDYI)
0	RDYIM	Ready Interrupt Mask 0: Enables ready interrupt (RDYI) 1: Disables ready interrupt (RDYI)

NOTE

This Register is valid only when ETNB1SGOPMC.MODE = 0.

25.3.3.6 ETNB1SGLTVC — Link Timer Value Configuration Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + 1014_H

Value after reset: $0003\ 0D40_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LINK_TIMER [17:16]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LINK_TIMER[15:0]															
Value after reset	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.105 ETNB1SGLTVC Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17 to 0	LINK_TIMER [17:0]	LINK_TIMER Specifies the number of cycles to ensure the LINK_TIMER period specified as standard in units of 8 ns (the period for one cycle at 125 MHz). Standard value: 1.6 ms (8 ns × initial value of 200,000 cycles) These bits should be changed only for debugging purposes.

NOTE

This Register is valid only when ETNB1SGOPMC.MODE = 0.

25.3.3.7 ETNB1SGCECT — Code Error Counter Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + 1018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CER_CNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.106 ETNB1SGCECT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CER_CNT[15:0]	Code Error Counter These bits are used to count the number of code errors. These bits are initialized by writing 0000_H to these bits.

25.3.3.8 ETNB1SGRECT — Running Disparity Error Counter Register

Access: This register can be read or written in 32-bit units.

Address: <ETNB1_base> + $101C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RER_CNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.107 ETNB1SGRECT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	RER_CNT[15:0]	Running Disparity Error Counter These bits are used to count the number of running disparity errors. These bits are initialized by writing 0000_H to these bits.

25.3.3.9 ETNB1SGSDS — SGMII SerDes Status Register

The ETNB1SGSDS register indicates the status of SGMII SerDes.

Access: This register is a read-only register that can be read in 8-bit units

Address: <ETNB1_base> + 1800_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SUC	PWS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 25.108 ETNB1SGSDS Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	SUC	Startup Completion Flag 0: Startup not completed 1: Startup completed
1, 0	PWS[1:0]	Power Status 00_B: Off 01_B: During power-on sequence 10_B: During power-off sequence 11_B: On

25.3.3.10 ETNB1SGCLKSEL — SGMII Reference Clock Select Register

The ETNB1SGCLKSEL register is used to select the SGMII reference clock.

Access: This register can be read or written in 8-bit units

Address: <ETNB1_base> + 1804_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SEL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 25.109 ETNB1SGCLKSEL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEL	Reference Clock Select 0: External clock (25 MHz) 1: Internal MOSC clock (20 MHz) This setting is allowed only when MOSC is 20 MHz.

25.3.3.11 ETNB1SGRCIE — SGMII Reference Clock Input Enable Register

The ETNB1SGRCIE register is used to enable the SGMII reference clock input.

Access: This register can be read or written in 8-bit units

Address: <ETNB1_base> + 1808_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 25.110 ETNB1SGRCIE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RCIE	Reference Clock Input Enable 0: Disable 1: Enable The CPU can only write 1 to this bit. This bit is automatically cleared in SGMII SerDes power-off sequence.

25.4 Operation

The Ethernet AVB consists of the following functional units:

- DMA transfer controller (AVB-DMAC): Handles DMA transfer between the data storage areas for reception and transmission in the URAM and the reception and transmission FIFO buffers
- MAC controller (E-MAC): Handles transfer between the reception and transmission FIFO buffers and the MII/RMII/SGMII

Using its direct memory access (DMA) function, the AVB-DMAC handles DMA transfer of frame data between the destinations for storing Ethernet frame data for transmission and reception in the URAM and the FIFO buffers for reception and transmission. Data cannot be directly read from or written to the FIFO buffers.

To handle DMA transfer, the AVB-DMAC requires information that includes the addresses for storage of data for transmission and received data. The information is referred to as descriptors. The AVB-DMAC reads data for transmission from the storage area for data to be transmitted and writes received data to the storage area for received data according to the information described in descriptors. The descriptors are placed in the URAM. Arranging multiple descriptors in descriptor lists allows the continuous reception or transmission of multiple Ethernet frames.

The E-MAC supports a MII/RMII/SGMII, which provides an interface format for the externally connected PHY-LSI. The E-MAC constructs Ethernet frames from data written to the transmission FIFO and transmits these frames to the MII/RMII/SGMII. It also performs CRC checking of Ethernet frames received from the MII/RMII/SGMII and writes the frames to the reception FIFO.

25.4.1 AVB-DMAC Operating Modes

Figure 25.5 illustrates the operating modes of the AVB-DMAC.

Transitions of AVB-DMAC operating mode are under the control of the items listed below.

- CPU operating mode (hardware reset)
- Configuration of the operating mode configuration bits (ETNBnCCC.OPC) in the AVB-DMAC mode register

The current operating mode can be determined by reading the operating mode status bits in the AVB-DMAC status register (ETNBnCSR.OPS).

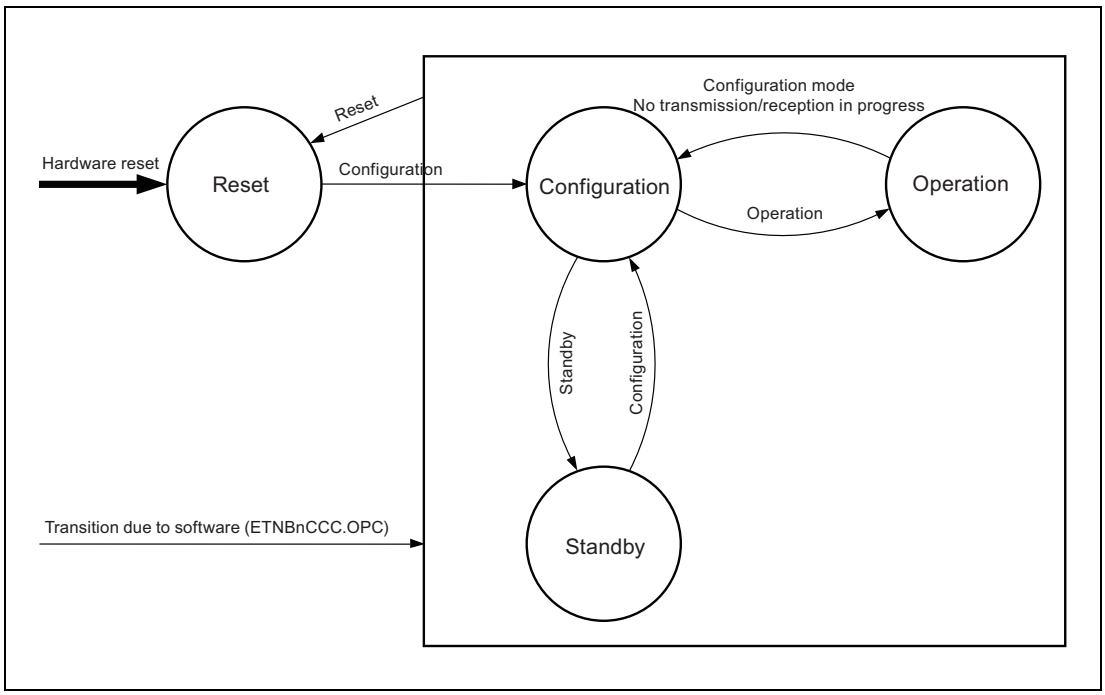


Figure 25.5 Operating Mode of AVB-DMAC

25.4.1.1 Operating Modes

(1) Reset mode

After a hardware reset, the AVB-DMAC enters reset mode.
In reset mode, only the AVB-DMAC operating mode control function is controllable and other functions are all stopped. This mode is designed for reduced power when the Ethernet function is not necessary.

(2) Configuration mode

In configuration mode, various settings for the AVB-DMAC can be made.
The operating functions are stopped and all status registers are initialized to their reset values.
The E-MAC functions in this mode.
By ETNBnCCC.GAC it is possible to enable gPTP support already in this mode.

(3) Operation mode

In operation mode, all functions of the AVB-DMAC can operate.
Ethernet communications can only proceed in this mode.
In operation mode, do not set E-MAC again.

(4) Standby mode

In standby mode, the E-MAC can only be used to control the operating mode. Other functions cannot be used.
When ETNBnCCC.GAC is 1, CPU should not enter this mode.

25.4.1.2 How to Set the Operating Mode

Set the operating mode configuration bits in the AVB-DMAC mode register (ETNBnCCC.OPC) to select the operating mode. Furthermore, the current operating mode can be checked by reading the operating mode status bits in the AVB-DMAC status register (ETNBnCSR.OPS).

Transitions other than from operation mode to configuration mode are made after the value is written to the operating mode configuration bits (ETNBnCCC.OPC) (**Figure 25.6**).

For transitions from operation mode to configuration mode, follow the procedure in **Figure 25.7** because any transmission and reception in progress will be executed before the transition to configuration mode.

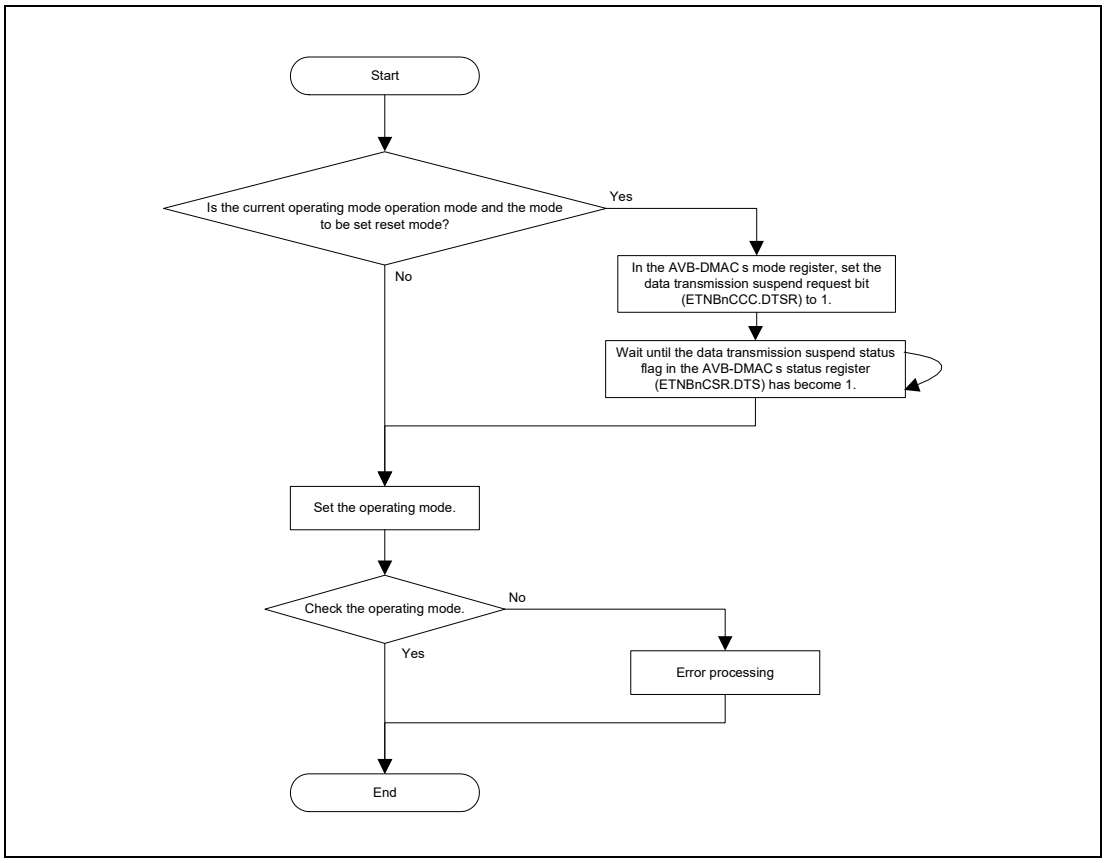


Figure 25.6 Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode)

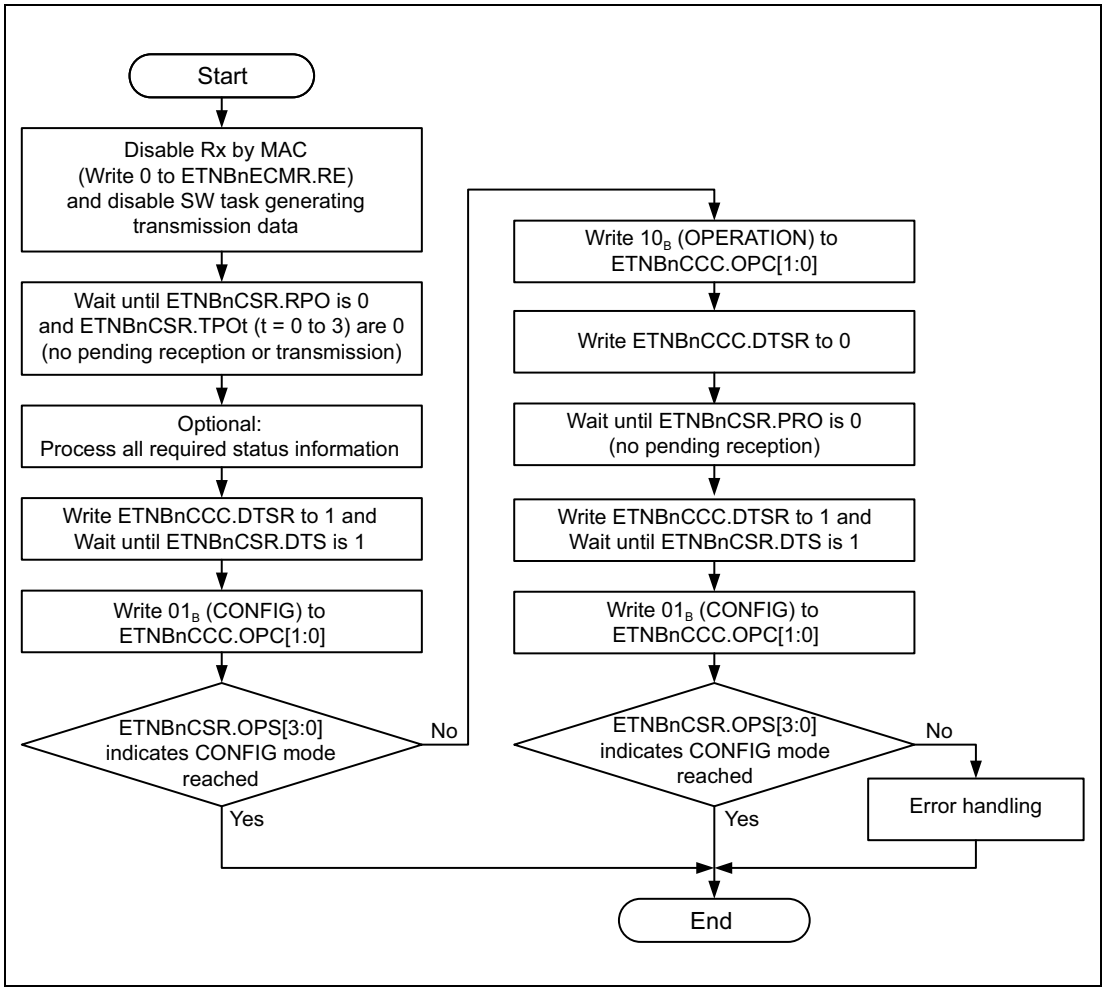


Figure 25.7 Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)

In the transition from operation mode to configuration mode, the AVB-DMAC executes the following operations before the transition is completed. Read the operating mode status bits in the AVB-DMAC status register (ETNBnCSR.OPS) to check that the transition to configuration mode has been completed.

- If the transfer of a frame between the reception FIFO and URAM is in progress, this is completed (other received frames remaining in the FIFO and any frames that are subsequently received by the E-MAC are discarded).
- If the transfer of a frame is in progress between the transmission FIFO and URAM, this is completed
  (frames for transmission remaining in the URAM will not be transmitted).
- All frames for transmission in the transmission FIFO are transferred to the E-MAC.

Notes:

When the operating mode shifts to configuration mode, all status registers are cleared.

We recommend following the procedure below in the case of this transition.

1. Disable reception.
2. Since reception actually stopping after being disabled requires time, wait for an interval equivalent to that for reception of a maximum length packet.

3. Stop the software task that is generating data for transmission.
4. Wait until the receive process status bit (ETNBnCSR.RPO) and the transmit process status bits (ETNBnCSR.TPO0 to 3) in the AVB-DMAC status register are set to 0.
5. Capture all of the required status information.
6. Set the operating mode configuration bits in the AVB-DMAC mode register (ETNBnCCC.OPC) to initiate the transition to configuration mode.

25.4.1.3 Operating Mode Transitions Due to Hardware

The following hardware factors can also initiate transitions of the AVB-DMAC operating mode.

(1) Hardware reset

Resetting of the LSI chip leads to resetting of the entire EthernetAVB module. The operating mode shifts to reset mode.

25.4.2 Common Control for Transmission and Reception

25.4.2.1 Initialization Procedure

Figure 25.8 shows the overall initialization procedure in outline.

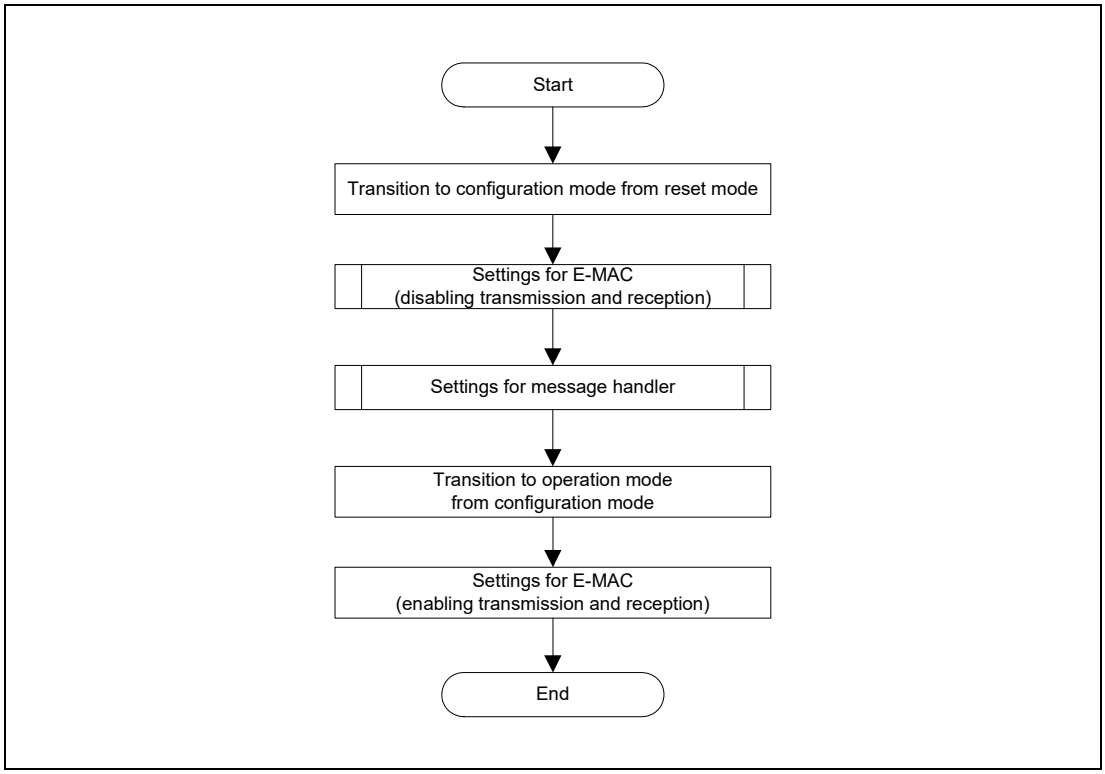


Figure 25.8 Outline of the Initialization Procedure

(1) Initializing the Receiver Section

Before starting reception, follow the procedure below.

Set the operating mode to operation mode or standby mode, and do not enable reception until the settings for the AVB-DMAC are completed.

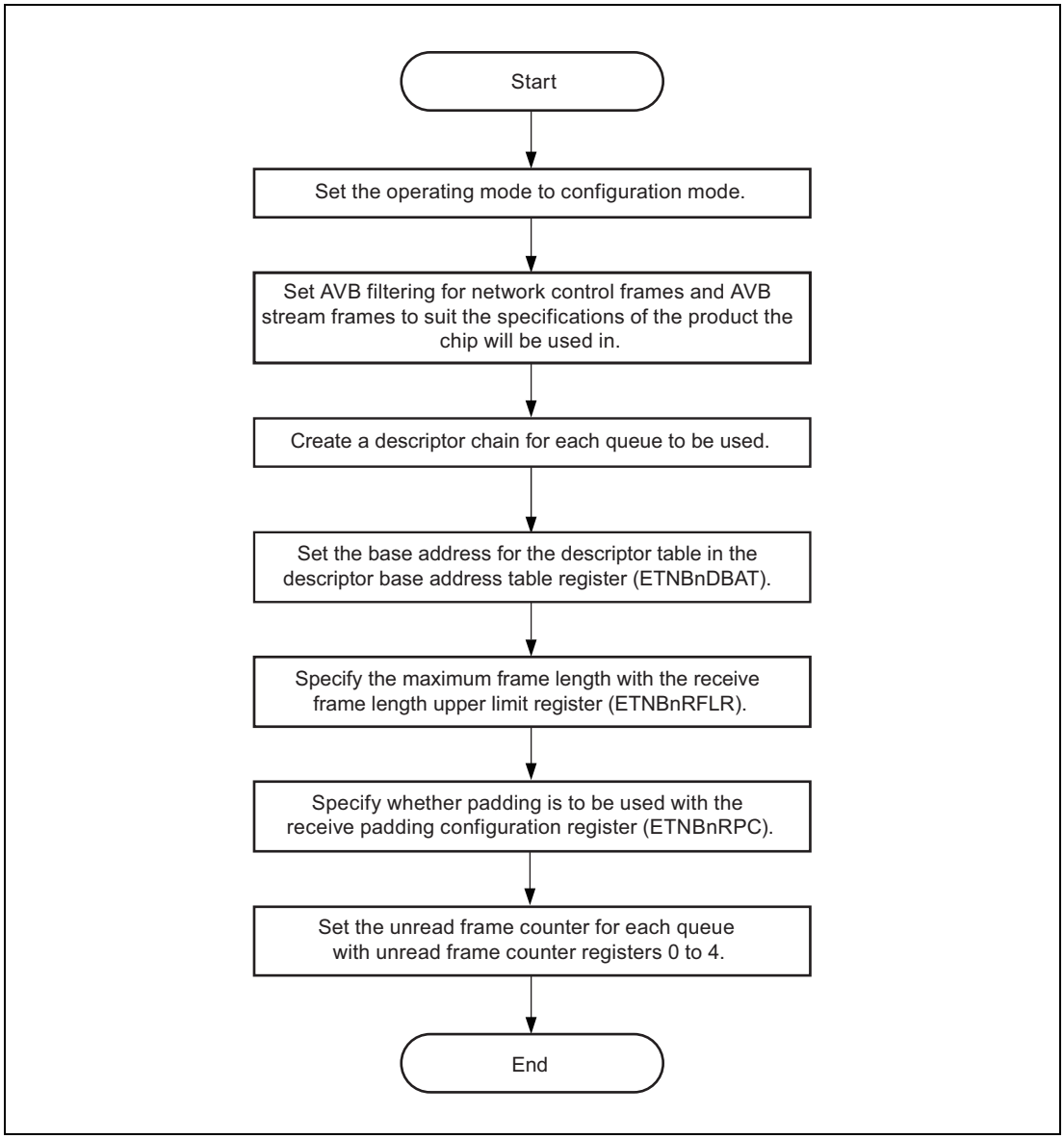


Figure 25.9 Procedure for Initializing the Receiver Section

(2) Initializing the Transmitter Section

Figure 25.10 illustrates initialization of the transmitter section.

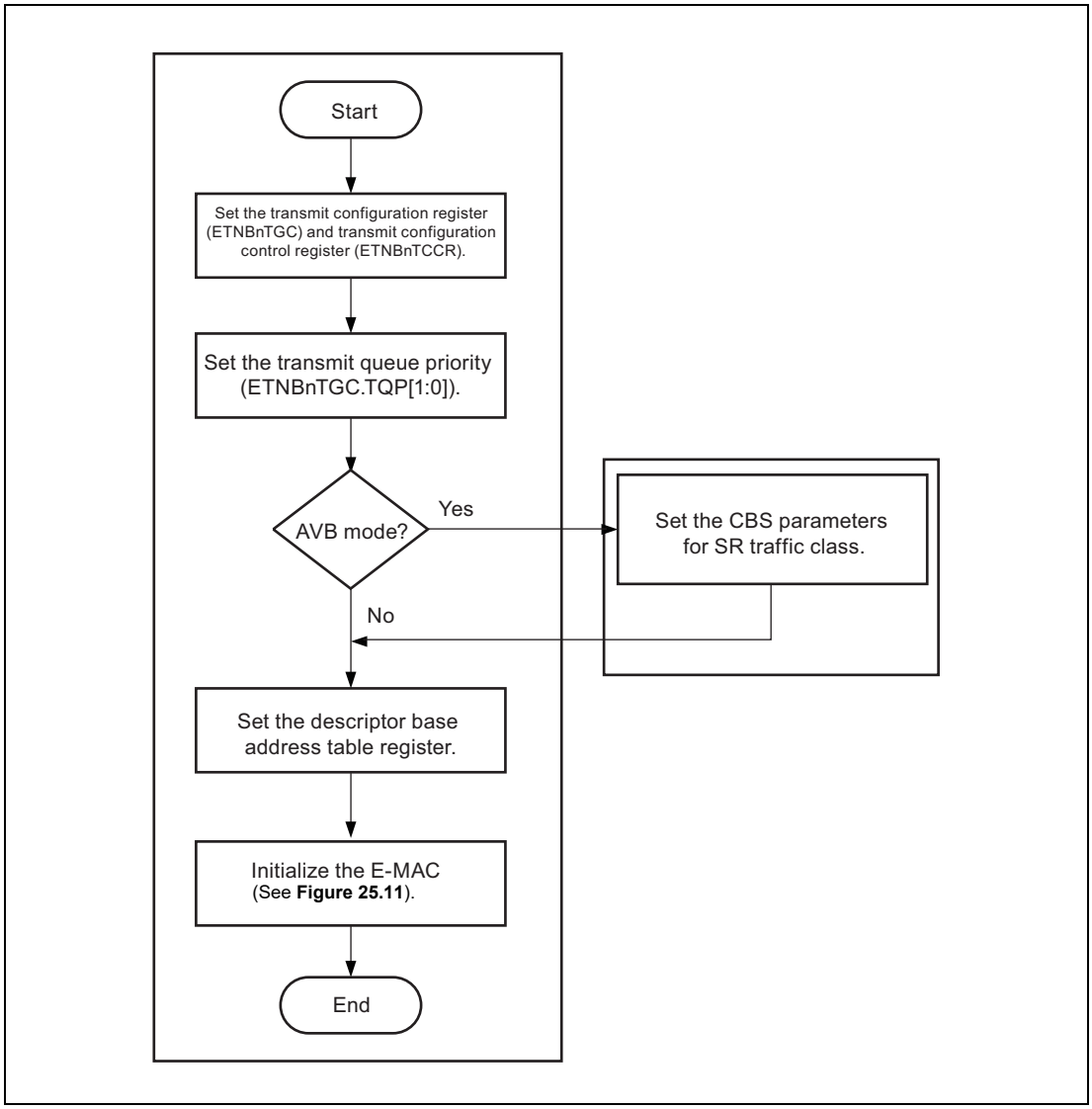


Figure 25.10 Procedure for Initializing the Transmitter Section

(3) Configuration the E-MAC Block

Figure 25.11 illustrates configuration of the E-MAC block.

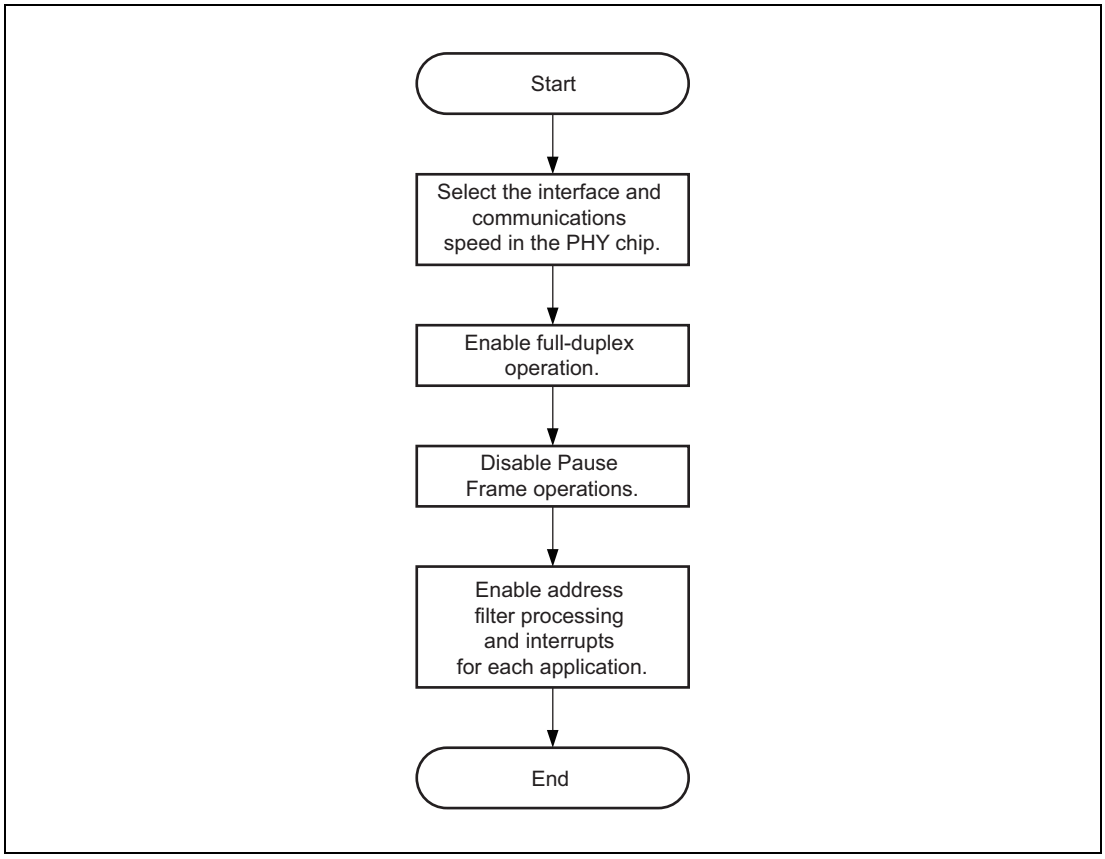


Figure 25.11 Procedure for Configuration the E-MAC Block

(4) Configuration the Message Handler Section

Figure 25.12 illustrates configuration the message handler section.

For a description of how to set up the descriptors and the CBS traffic shaping parameters, see **Section 25.4.3, Descriptors**, and **Section 25.4.6, CBS (Credit-Based Shaping)**.

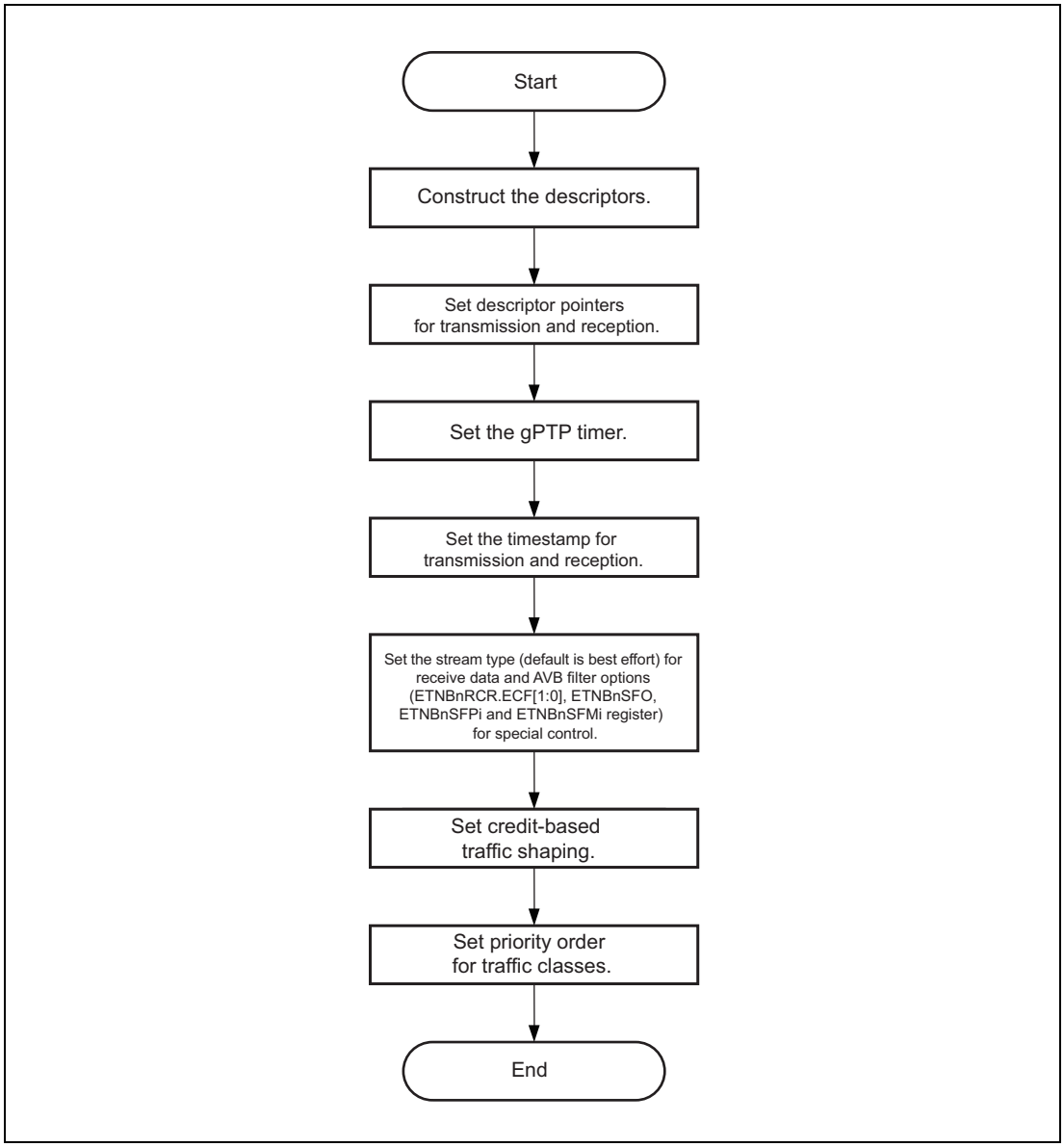


Figure 25.12 Configuration the Message Handler Section

25.4.2.2 Scheduling Reception and Transmission

The AVB-DMAC normally has independent buses for transmission and reception. Furthermore, the four processes of fetching, storing, transmission and reception are basically independent of one another. Fetching and storing, however, share the same bus master so cannot be executed simultaneously. Access to the bus master is controlled by the scheduler.

Figure 25.13 is a schematic view of AVB-DMAC operations in transmission and reception.

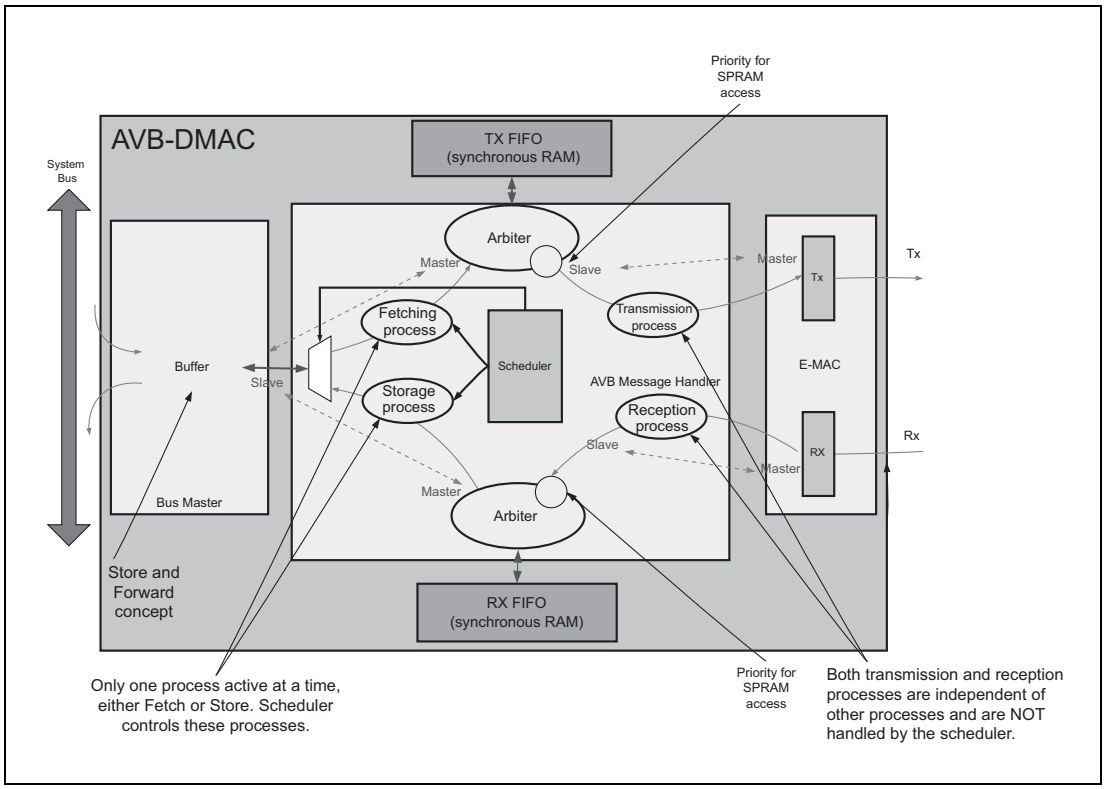


Figure 25.13 Schematic View of AVB-DMAC Operations in Transmission and Reception

Storing and fetching are alternately performed. When the number of frames held by the reception FIFO reaches the warning level, storing takes precedence over fetching.

(1) Relationship between Transmission Queue Numbers and Traffic Classes

In fetching, the relationships between the transmission queues and traffic classes are fixed, so the priority specified by the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) has no effect.

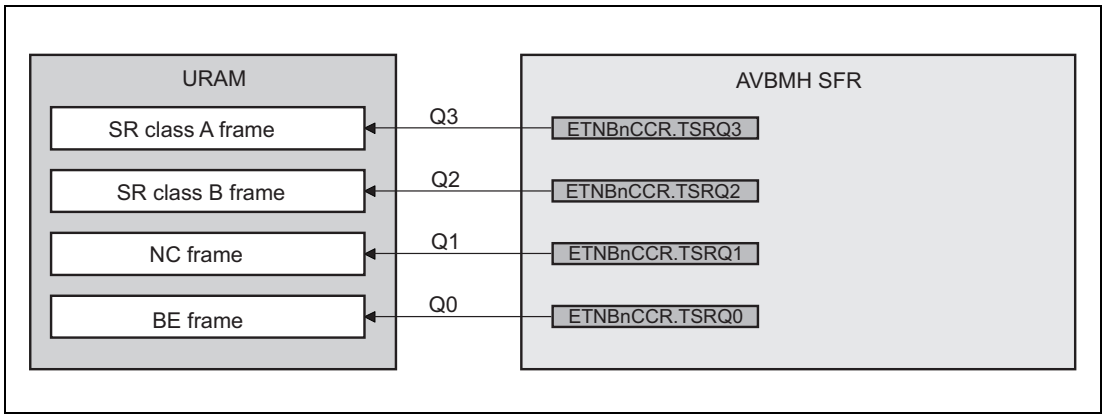


Figure 25.14 Class Associations of Queues for the Scheduler

In fetching, the credit values for stream classes A and B are not taken into account. Behavior depends on the setting of the transfer FIFO size configuration bits in the transfer configuration registers (ETNBnTGC.TBDt) and on the frame size that can be fetched to the transmission FIFO.

When the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) are 00_B or 01_B, the priority order is Q3 → Q 2 → Q1 → Q0.

When the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP) are 11_B, the priority order is Q1 → Q3 → Q2 → Q0.

25.4.2.3 Checking Integrity

The AVB-DMAC is capable of detecting and identifying errors produced in the processing of Ethernet frames and in the transfer of frame data for transmission and reception.

(1) Concept of Integrity Checking in Reception

The purpose of receive integrity check is to prevent error frames from being stored in the URAM.

If an error frame is stored, information to identify the frame as an error frame is appended to the data from the frame in the URAM.

CAUTION

If a special descriptor chain is to be used for header/data separation on reception, an error that breaks the sequence may lead to storage space for synchronization running out. In such cases, software interaction or re-synchronization via the EOS descriptor is required.

(2) Concept of Integrity Checking in Transmission

The purpose of integrity checking in transmission is to prevent the transmission of broken frames.

Since transmission of a frame by the E-MAC can neither be stopped nor disabled once it has started, this check involves intensive monitoring for problems that can arise during fetching.

(3) Items for Monitoring in Both Reception and Transmission

(a) Errors in access to the URAM for reading of descriptors

The same descriptor may be processed again because the current descriptor address (ETNBnCDARq.CDA) is not changed.

If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.
- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

Errors in access to read descriptors from the URAM are detected from the response signal of the AXI-Bus.

(b) Illegal configuration of a descriptor by an application

The same descriptor may be processed again because the current descriptor address (ETNBnCDARq.CDA) is not changed.

If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.

- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

(c) Errors in access to the URAM for writing of descriptors

As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated.

As DESCR.DT was not updated, hardware and software synchronization may have been disengaged.

Errors in access to write descriptors to the URAM are detected from the response signal of the AXI-Bus.

(4) Items for Monitoring in Reception

(a) Errors in access to the URAM for writing of data or timestamps

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) is updated.
- DESCR.EI is set to indicate incorrect contents.
- When an access error occurs in a divided frame, the descriptor sequence may be damaged and queues may be unusable.

Errors in access to write data or descriptors to the URAM are detected from the response signal of the AXI-Bus.

(b) Errors in Access for Reading from the Reception FIFO

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) is updated.
- DESCR.EI is set to indicate incorrect contents.
- When an access error occurs in a divided frame, the descriptor sequence may be damaged and queues may be unusable.
- Errors of this type are detected by the ECC checker for the reception FIFO. For details of this ECC function, see **Section 44.3.10, ECC for Peripheral RAM**.

(c) Damaged Data in the Reception FIFO

- Received frames are all invalidated.
- All frames stored in reception FIFO are discarded. At this time, the number of frames and queue information cannot be captured.

If damaged data in the reception FIFO is an error due to the reception FIFO, this is detected by the AVB-DMAC.

(5) Items for Monitoring in Transmission

(a) Errors in Access for Reading Data from the URAM

- Data that have already been fetched are discarded from the transmission FIFO.
- When an error of this type occurs during processing of an FSINGLE or FEND descriptor:
  As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.
- When an error of this type occurs during processing of an FSTART or FMID descriptor:
  - The current descriptor address (ETNBnCDARq.CDA) is not updated.
  - The transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) is set to 0.

Errors in access to read data from the URAM are detected from the response signal of the AXI-Bus.

(b) Overflow of the Transmission FIFO

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.
- The frame will be discarded from the FIFO.

(c) Errors in Access for Reading of the Transmission FIFO

The AVB-DMAC is incapable of detecting an error in reading of the transmission FIFO, resulting in the transmission of a broken frame.

Errors of this type are detected by the ECC checker for the transmission FIFO. For details of this ECC function, see **Section 44.3.10, ECC for Peripheral RAM**.

(d) Frame size error during transmission

- As in the case where no error occurs, the current descriptor address (ETNBnCDARq.CDA) and the transmit start bit in the transmit configuration control register (ETNBnTCCR.TSRQt) are updated. Fetching resumes after the error frame.

A transmit frame size error is detected when the size setting in one or more (in the case of a divided frame) descriptors for frame transmission is 2044 or more bytes. Such frames are cut out and transmitted.

(e) Damaged Data in the Transmission FIFO

- Fetching is not affected by damaged data.
- Since damaged data from the FIFO is only detected during frame transmission, an unexpected frame data may be transmitted.

If damaged data in the transmission FIFO is an error due to the transmission FIFO, this is detected by the AVB-DMAC.

25.4.3 Descriptors

25.4.3.1 Data Representation in URAM

The AVB-DMAC transfers data for transmission and received data to and from the application software via the URAM.

The memory in the URAM for use by the AVB-DMAC is configured with control structures referred to as descriptors and associated areas to which the frame data are allocated. Dividing the memory into a control area and data area allows the flexible allocation of frame data to the URAM. This enables sharing of the areas to which frame data are allocated and the use of non-contiguous areas. Frame data can be copied without using the CPU. Arbitration that ensures hardware and software access to the memory area is also available without using the AVB-DMAC.

Figure 25.15 shows an example of the memory maps for descriptors and the descriptor data area in the URAM.

A descriptor consists of its type (DESCR.DT), which controls the descriptor functions, a descriptor pointer (DESCR.DPTR) indicating the start address for storage of the frame data in the descriptor area, and the data size field (DESCR.DS), indicating the amount of frame data. Respective descriptors can generate interrupts after completion of processing. Enabling and disabling of the interrupt is controlled by the descriptor interrupt enable bits (DESCR.DIE).

The descriptor may also hold information related to content. This information does not affect general descriptor functions. It provides information other than the frame data proper, such as on the state of reception.

For details, see **Section 25.4.4.2, Setting Up Reception Descriptors**, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.

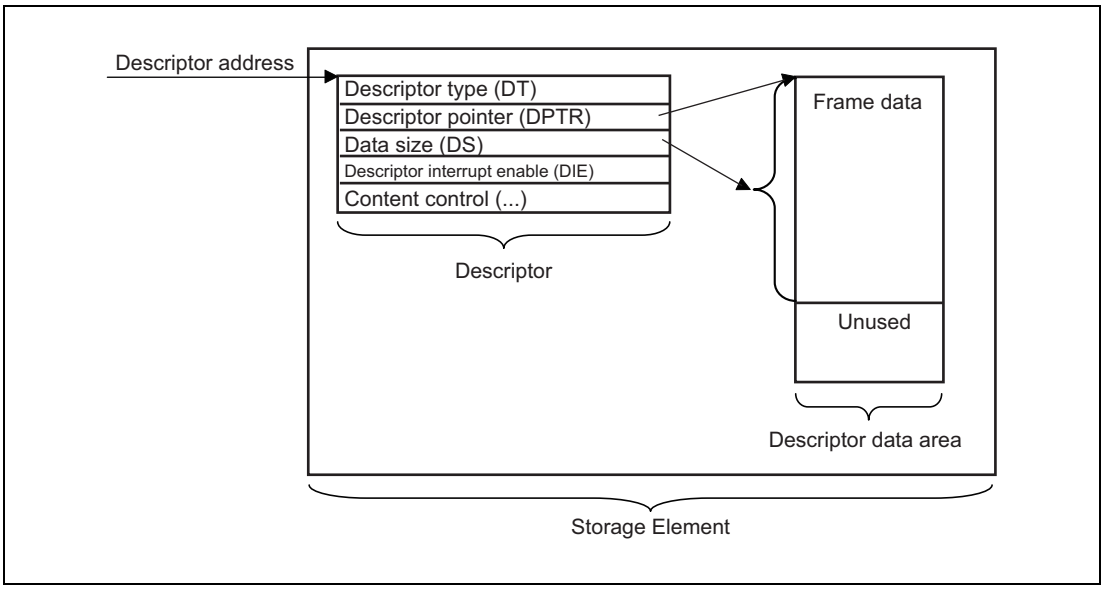


Figure 25.15 Example of URAM Memory Map

The descriptor must be aligned with a 32-bit boundary in the URAM.

Descriptors are generally configured of 64 bits, but are configured of 160 bits when reception and storage of gPTP timestamps is enabled.

The frame data must also be aligned with a 32-bit boundary in the URAM.

The amount of data in the frame is defined by the data size bits (DESCR.DS). In reception, these bits indicate the upper limit on the size of frames to be received. If the data size is not aligned with a 32-bit boundary, the residual bytes in the data area are set as unused bytes.

CAUTION

AVB-DMAC will store the data area pointed by descriptor for specified data size, even if there is no enough space.

25.4.3.2 Using Descriptor Chains in Queues

Transmission and reception descriptors in the URAM are grouped into queues. Each queue handles frames so that they are transmitted in order of priority and received separately. A queue is capable of controlling one or more frames. Accordingly, multiple descriptors can be assigned to one queue. A combination of multiple descriptors is referred to as a descriptor chain.

For a descriptor chain, the three general descriptor types listed below are defined. For details on these descriptor types, see **Section 25.4.3.6, Descriptor Type**.

- Descriptors that define frame data
- Descriptors that control the descriptor chain itself (e.g. LINK, EOS).
- Descriptors that arbitrate access by hardware or software

Figure 25.16 shows the two basic topologies for descriptor chains. In the simplified examples in the figure, all descriptors allocated to the chain are stored in the array.

- For a linear descriptor chain, the last descriptor in the array is a control descriptor indicating the end of the descriptors (e.g. EEMPTY).
- For a cyclic descriptor chain, the last descriptor in the array is a control descriptor that returns to the first descriptor in the array (e.g. LINK).

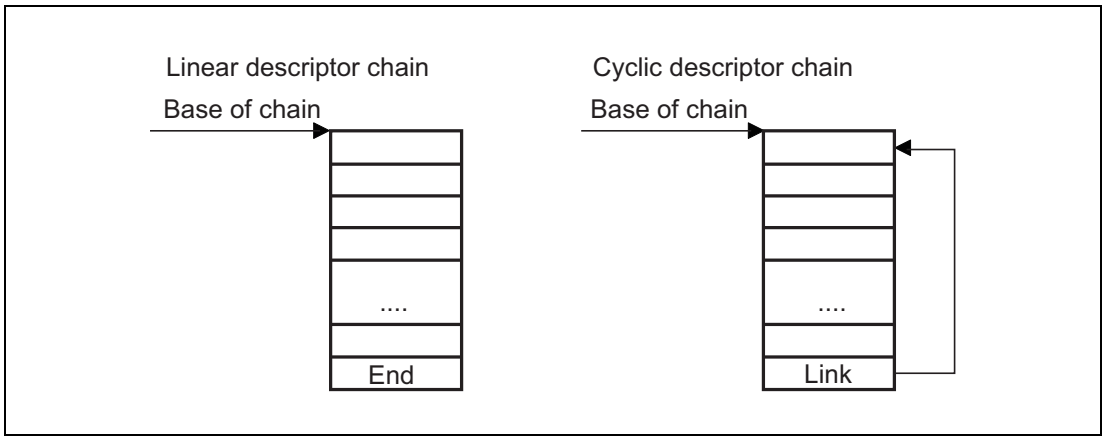


Figure 25.16 Outline of the Basic Descriptor Chains

The relationship between queues and descriptor chains is defined by the base addresses of chains. A queue is connected to one descriptor chain over one round of processing. There is also a method of switching to a different chain while in operation mode.

There are no restrictions on the number of link descriptors and their locations within the chain. The last descriptor of a designed chain determines the topology.

Which chain structure is to be used or which topology is suitable depends on the application. A description of how to design descriptor chains to suit various applications is given in **Section 25.4.4.2**. Procedure for Setting Reception Descriptors, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.

25.4.3.3 Descriptor Base Address Table

Concerning the base address table to be written in the URAM, set the address of the top descriptor of respective queues.

Entries 0 to 3 are used to access transmission queues 0 to 3. Subsequent entries are used to access reception queues. Entry 4 thus corresponds to reception queue 0.

The configuration of entries in the base address table is the same as the configuration of link descriptors. We recommend using the descriptor type (DESCR.DT) LINKFIX. The link descriptor does not need to be updated because it does not change even after descriptor processing, so it does not require updating. The first descriptor of a chain performs hardware and software synchronization. If the application requires hardware and software synchronization for the base addresses, use the descriptor type (DESCR.DT) LINK.

The CPU is only capable of using LINKFIX and LINK as descriptor types (DESCR.DT) of descriptors in the base address table.

Set the location of the base address table in the URAM in the descriptor base address table register (ETNBnDBAT).

Figure 25.17 shows an example of a base address table for controlling four transmission and three reception queues. The boxes to the right of the table represent descriptor chains of a topology.

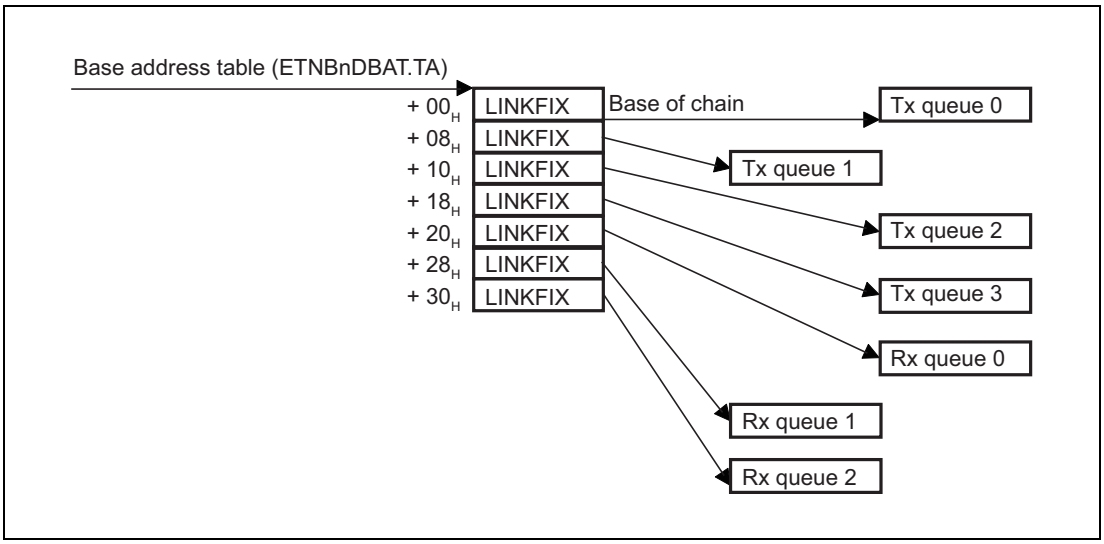


Figure 25.17 Example of a Base Address Table for Reception and Transmission Queues

CAUTION

The size of the descriptors in the base address table is always eight bytes even if the queue itself includes extended descriptors.

25.4.3.4 Descriptor Chain Processing

When a descriptor is currently being processed or the queue for a descriptor is active, the current descriptor is a descriptor to be processed. The current descriptor address for use by a queue q can be checked in the current descriptor address register q (ETNBnCDARq).

The current descriptor is stored in a register or descriptor as described below.

- In the descriptor base address table registers for all q queues (ETNBnDBAT) (ETNBnDBAT.TA + 8 × q) when the operating mode shifts to operation mode.
- In the descriptor base address table register (ETNBnDBAT) (ETNBnDBAT.TA + 8 × q) when a base address load request is issued for a queue q by setting the corresponding bit (ETNBnDLR.LBAq) in the descriptor base address load request register.
- It is set to DESCR.DPTR when a link descriptor (LINK, LINKFIX) is processed.

After a descriptor has been processed, the current descriptor for the same queue is incremented by the size of the descriptors being handled by the queue (8 bytes for normal descriptors and 20 bytes for extended descriptors). The AVB-DMAC updates the descriptor type and informs the CPU that the descriptor has been processed.

25.4.3.5 Descriptor Interrupts

A descriptor is able to issue a descriptor interrupt on completion of its processing. The setting of the descriptor interrupt enable bits (DESCR.DIE) in each descriptor selects disabling or generation of the descriptor interrupt.

The descriptor interrupt is a common resource that is shared between reception and transmission queues. Software control of the descriptor interrupt provides a flexible method of application-specific flag processing.

Figure 25.18 illustrates the way in which the AVB-DMAC generates descriptor interrupts (or sets bits in the descriptor interrupt status register (ETNBnDIS.DPFi)). Processing of a descriptor with the value i in the descriptor interrupt enable bits (DESCR.DIE) leads to the corresponding bit in the descriptor interrupt status register (ETNBnDIS.DPFi) being set.

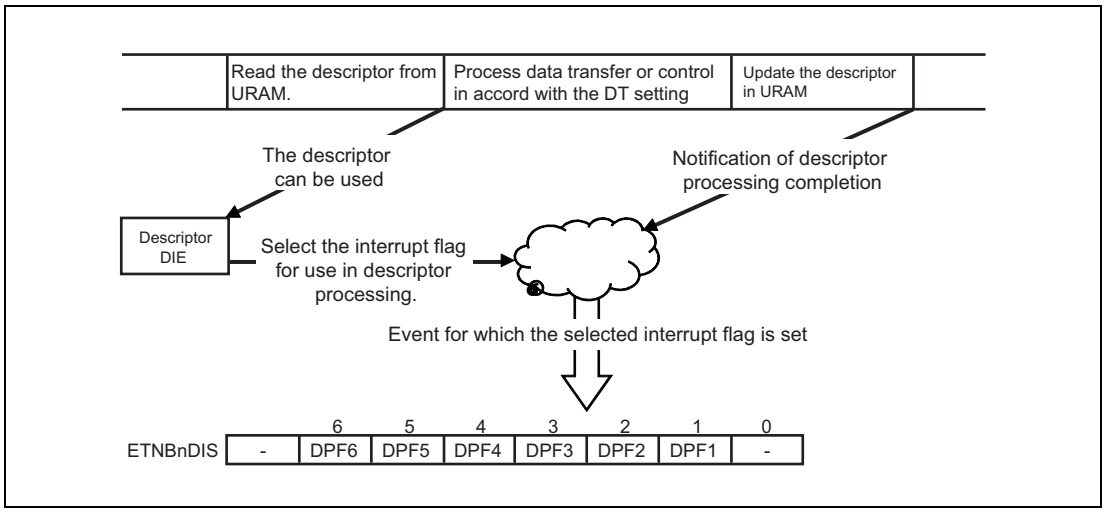


Figure 25.18 Method of Descriptor Interrupt Generation

25.4.3.6 Descriptor Type

The descriptor types (indicated by the DESCR.DT bits) supported by the AVB-DMAC fall into the following three categories.

- Definitions of frame data
- Control of descriptor chains
- Hardware and software arbitration

Table 25.111 is a summary of the descriptor types available for the AVB-DMAC. Entries under "Name" are the names of the descriptor types and the values under "DT" are the corresponding values to be set in the descriptor type field (DESCR.DT). A given descriptor may be handled differently according to whether it is in a transmission or reception queue, so the transmission and reception columns list the scopes of control and processing of the descriptor types.

The abbreviations defined below are used in the transmission and reception columns.

Definition of SW:

- The descriptor is processed by software.
- Software has access to and may modify the descriptor and descriptor data area.
- The descriptor cannot be changed by hardware (AVB-DMAC).

Definition of HW:

- The descriptor is processed by hardware (AVB-DMAC).
- Software must modify neither the descriptor nor the descriptor data area.
- Hardware (AVB-DMAC) processes the descriptor and subsequently changes the descriptor type.

Invalid:

This descriptor type is not used in transfer in the given direction (transmission or reception).

Do not write the value to the descriptor type (DESCR.DT) field.

Hardware does not process the descriptor type. The current descriptor address (ETNBnCDARq.CDA) is not changed even when the queue is of a descriptor type of this setting.

Table 25.111 Summary of Descriptor Types

Name	DT	Description	Reception	Transmission
Frame data				
FSTART	5	Frame Start The descriptor points to valid data for a frame. The frame starts with the given data and continues with that indicated by the next descriptor.	SW	HW
FMID	4	Frame Middle The descriptor points to valid data for a frame. The frame started with a previous descriptor and continues to the data indicated by the next descriptor.	SW	HW
FEND	6	Frame End The descriptor points to valid data for a frame. The frame continues from the previous descriptor and ends with the data indicated by in this descriptor.	SW	HW
FSINGLE	7	Frame Single The descriptor points to valid data for a complete frame.	SW	HW
Chain control				
LINK	8	Link Defines the next descriptor in the chain.	HW	HW
LINKFIX	9	Fixed Link Defines the next descriptor in the chain, but not changed by AVB-DMAC after processing.	SW	SW
EOS	10	End Of Set Control element to split descriptor chain. Chain stops and waits for user interaction.	HW	HW
HW/SW arbitration				
FEMPTY	12	Frame Empty A descriptor related to frame data but not containing valid data for a frame	HW	SW
FEMPTY_IS	13	Frame Empty Incremental Start A descriptor related to frame data but not containing valid data for a frame DESCR.DPTR sets the base address of an “incremental data area” in the URAM.	HW	Invalid
FEMPTY_IC	14	Frame Empty Incremental Continue A descriptor related to frame data but not containing valid data for a frame Data indicated by the pointer are for storage in an incremental data area in the URAM.	HW	Invalid
FEMPTY_ND	15	Frame Empty No Data Storage A descriptor related to frame data but not containing valid data for a frame The descriptor is processed in the same way as FEMPTY but data are not stored in the URAM.	HW	Invalid
LEMPTY	2	Link Empty A link descriptor for processing by the AVB-DMAC	SW	SW
EEMPTY	3	EOS Empty An EOS descriptor for processing by the AVB-DMAC	SW	SW
DT0	0	Reserved	Invalid	Invalid
DT1	1	Reserved	Invalid	Invalid
DT11	11	Reserved	Invalid	Invalid

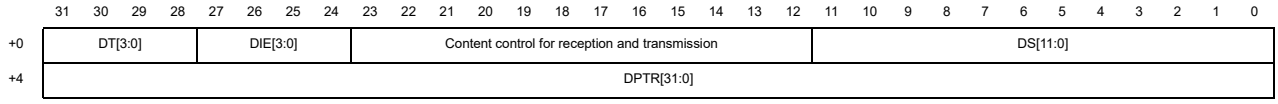
(1) Layout of General Descriptors in the URAM

The AVB-DMAC updates processed descriptors in the URAM. The field to be changed in a descriptor being updated depends upon whether the direction is transmission or reception and the queue mode. Other fields will not be changed. There are no restrictions on the values set in unused descriptor fields (indicated by "—" in the figure).

(2) Frame Data Descriptors

The allocation of bits in the frame data descriptors (FSTART, FMID, FEND, and FSINGLE) is shown below.

- Normal descriptor (usable in both reception and transmission)



- Extended descriptor (usable only in reception)

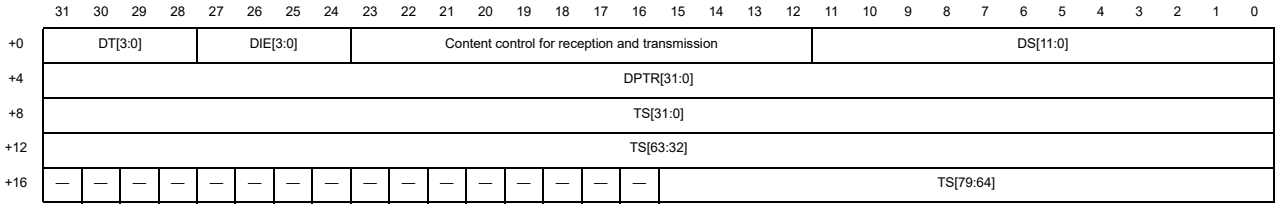


Table 25.112 Contents of Frame Data Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 5: FSTART 4: FMID 6: FEND 7: FSINGLE For details, see **Section 25.4.4.2, Setting Up Reception Descriptors**, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 25.4.4.2, Setting Up Reception Descriptors**, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor
TS[79:0]	Timestamp Timestamp of the received frame (only available in extended descriptors)

CAUTION

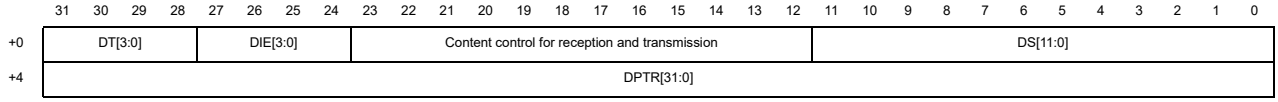
The upper reserved bits of TS[79:0] are set to 0000_H after the time stamp is stored.

(3) Hardware/Software Arbitration Descriptors (Only for Reception)

The allocation of bits in the descriptors for hardware/software arbitration (FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND) is shown below.

The allocation of bits in the arbitration descriptors for use in reception is the same as in frame data descriptors.

- Normal descriptor



- Extended descriptor (usable only in reception)

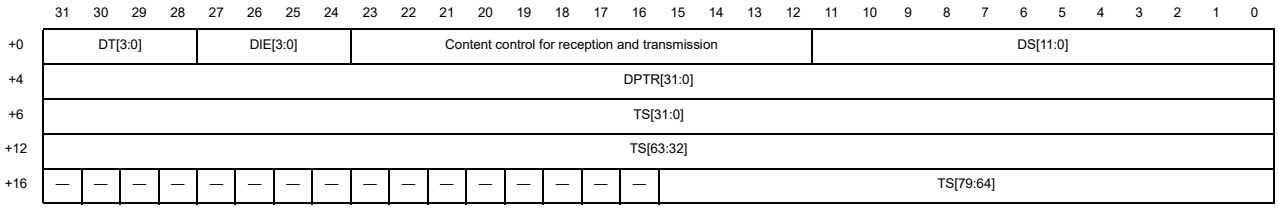


Table 25.113 Contents of Hardware/Software Arbitration Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 12: FEMPTY 13: FEMPTY_IS 14: FEMPTY_IC 15: FEMPTY_ND For details, see **Table 25.111, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 25.4.4.2, Setting Up Reception Descriptors**, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor
TS[79:0]	Timestamp Timestamp of the received frame (only available in extended descriptors)

CAUTION

When the descriptor is an extended descriptor, it has a 12-byte unused area.

In an FEMPTY descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), data size (DS), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IS descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IC descriptor, the descriptor type (DT) and descriptor interrupt enable (DIE) are used.

In an FEMPTY_ND descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and data size (DS) are used.

(4) Link Descriptors

The allocation of bits in the link descriptors (LINK and LINKFIX) is shown below.

- Normal descriptor

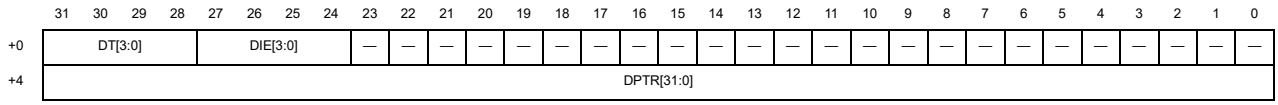


Table 25.114 Contents of Link Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 8: LINK 9: LINKFIX For details, see **Table 25.111, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).
—	Content Control For details, see **Section 25.4.4.2, Setting Up Reception Descriptors**, and **Section 25.4.5.2, Setting Up Transmission Descriptors**.
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor.

CAUTION

When the descriptor is an extended descriptor, it has a 12-byte unused area.

(5) Other Descriptors

The allocation of bits in the other descriptors (EOS, FEMPTY (only for transmission), LEMPTY, and EEMPTY) is shown below.

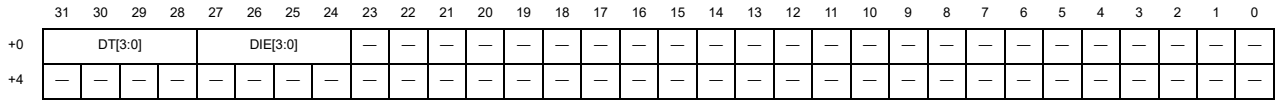


Table 25.115 Contents of Other Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 10: EOS 12: FEMPTY (only for transmission) 2: LEMPTY 3: EEMPTY For details, see **Table 25.111, Summary of Descriptor Types**.
DIE[3:0]	Descriptor Interrupt Enable 0000_B: Descriptor interrupt is disabled. 0001_B to 1111_B: The corresponding descriptor interrupt is generated (ETNBnDIS.DPFi).

CAUTION

When the descriptor is an extended descriptor, it has a 12-byte unused area.

(6) How to Use Frame Data Descriptors

The descriptor data area size field (DESCR.DS) can specify up to 2048 bytes of Ethernet frame data per data area. Settings higher than 2048 (bytes) cannot be made.

In general, Ethernet frames are not of uniform length. The AVB-DMAC is capable of dividing frame data into multiple descriptors in order to minimize the memory capacity for frame data. This function allows processing of frames that are longer than the limit for descriptor data areas. Frames divided based on a frame data structure can also be supported.

To handle both frames divided up into multiple data areas and descriptors for complete single frames, four types (DESCR.DT) FSTART, FEND, FMID, and FSINGLE are defined.

Figure 25.19 shows the mapping of frame data by frame data descriptors. The descriptor data areas are allocated to the URAM. For frames that require division into four or more data areas, additional FMID descriptors can be added as required.

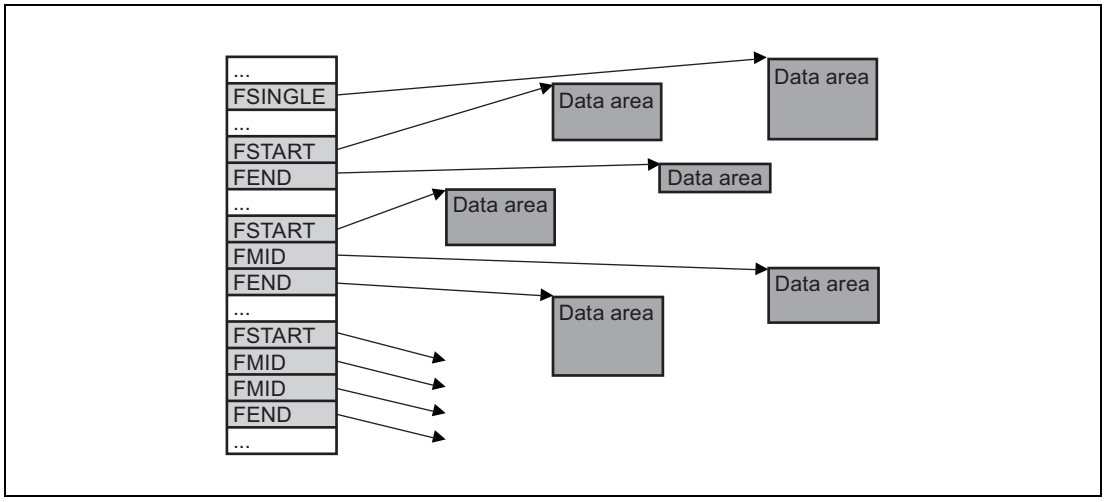


Figure 25.19 Mapping of Frame Data

For reception, set the descriptor data areas to the maximum size (i.e. give DESCR.DS its maximum value). The AVB-DMAC will store received frame data in the given area. If a received frame has more data than the maximum size, the AVB-DMAC will divide the data up.

For transmission, set the frame data size to the actual data size. The AVB-DMAC modifies the descriptor type (DESCR.DT) to FEMPTY after processing the relevant descriptor. The data size (DESCR.DS) and descriptor pointer (DESCR.PTR) fields retain their settings.

A descriptor data area including unused space produces an empty space between frame data. In reception, an “incremental data area” can be used to prevent empty spaces. For incremental data areas, see **Section 25.4.4.3(2), Incremental Data Areas**.

As well as reducing the memory capacity taken up by the descriptor area in the URAM, division into frames can be used to identify different sections of data (e.g. for separating a header and data).

(7) How to Use Chain Control Descriptors

(a) Link Descriptors

The link descriptors can be used to set up cyclic descriptor chains (for details, see **Section 25.4.3.2, Using Descriptor Chains in Queues**)

After a LINK descriptor is processed, its descriptor type (DESCR.DT) is changed to LEMPTY. The descriptor pointer (DESCR.PTR) retains its setting.

After processing of a LINKFIX descriptor, the descriptor type (DESCR.DT) is not updated. Software can change the descriptor type (DESCR.DT), descriptor interrupt enable (DESCR.DIE), and descriptor pointer (DESCR.DPTR). Take care, however, to check the current descriptor address register q (ETNBnCDARq.CDA) before changing the descriptor pointer (DESCR.DPTR).

(b) EOS Descriptor

Use the EOS descriptor to divide a descriptor chain into various segments. The queue can continue even after an EOS descriptor.

In transmission, the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQq) is cleared.

In reception, when the frame currently received is one divided for storage (e.g., received data whose frames are the FMID type or FEND type for storage), the data is not fully stored and the receive queue full interrupt (ETNBnRIS2.QFFr) is generated.

(8) How to Use Hardware and Software Arbitration Descriptors

In hardware processing of descriptors, the empty descriptor types (FEMPTY, LEMPTY, and EEMPTY) are used to distinguish various descriptors. For software, they can be used to initiate checking for empty spaces, etc.

(a) FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND

These descriptor types (DESCR.DT) are used for descriptors that do not contain effective data. Of these, only FEMPTY is used in transmission.

(b) LEMPTY

This descriptor type (DESCR.DT) is assigned to LINK descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an LEMPTY descriptor still points to the linked descriptor.

(c) EEMPTY

This descriptor type (DESCR.DT) is assigned to EOS descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an EEMPTY descriptor is not used.

(9) Synchronization between Descriptor Access by Hardware and Software

In primary HW/SW synchronization, the descriptor type (DESCR.DT) allocated to the URAM is usable. This makes it possible to minimize access by the AVB-DMAC to the SFR via the CPU, leading to higher performance.

Basic concepts of synchronization:

- Descriptor sets are allocated so that it is exclusively used by hardware or software, depending on the direction of transmission (see **Table 25.111, Summary of Descriptor Types**).
- Software must not change a descriptor assigned to hardware processing (the hardware does not change descriptors assigned to software processing).

In the case of software processing, the software must process the information in the descriptor and the corresponding frame data before changing the descriptor type. If a descriptor type assigned to hardware is set in DESCR.DT, the software should not change any part of the descriptor or of the corresponding frame data.

25.4.3.7 Tips for Optimizing Performance in Handling Descriptors

The following items are recommended as ways to ensure the optimal use of data structures in the URAM.

They are not requirements, but using a different approach may increase the load on the system bus within the LSI chip.

- Register descriptors with 64-bit alignment (this does not apply to extended descriptors).
- While in operation mode, use LINKFIX instead of LINK whenever a descriptor need not be changed. Hardware modifies the descriptor type (DESCR.DT) fields of LINK descriptors.
- Make an access to frame data in units of 128-byte blocks maximum.
- Design the descriptor chains in ways that minimize parallelism of processing.
  This helps in dividing the chains into segments allocated to different cache pages, and in arranging the different segments exclusively for access by software or hardware.
- Minimize the number of divided frames. This can reduce the overhead of descriptor handling.

25.4.4 Control in Reception

The point of the AVB-DMAC is to transfer data between the E-MAC and URAM without intervention by the CPU.

Create descriptors that define the amounts of frame data to be stored and the locations. After the E-MAC receives a frame, it stores the received frame data and the conditions of reception as the MAC state. If the descriptor is extended, the timestamp is also stored. For a description of how to set up descriptors for use in reception, see **Section 25.4.4.2, Setting Up Reception Descriptors**.

The AVB-DMAC filters received frames to separate them into various classifications (separation filtering). More specifically, this is done to separate received frames into the various reception queues and to set the priorities of different classes of received frames. For more on separation filtering, see **Section 25.4.4.1(1), Separation Filtering**.

Figure 25.20 shows the reception data bus and the selection of queues for use in reception.

Each frame received from the E-MAC is stored in the reception FIFO; in parallel with this, the frame is analyzed to identify its type and the target queue number. After the E-MAC completes reception, the target queue number is generated and stored in the reception FIFO. Appending of a reception flag depends on the storage of one frame among the reception queues in the URAM, and the unread frame counter (UFC) is also associated with frame storage.

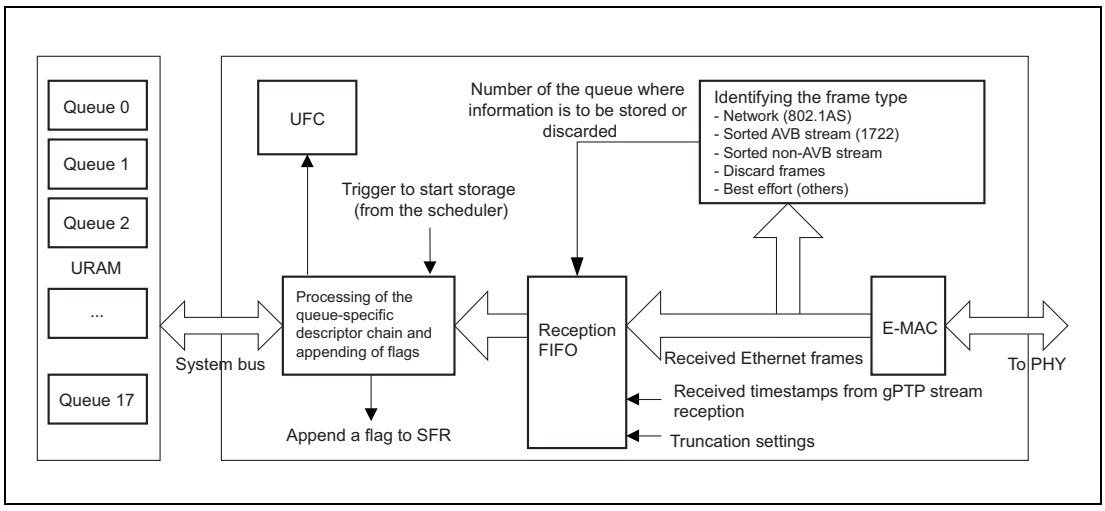


Figure 25.20 Mechanism of General Reception Queue Selection

25.4.4.1 Reception Queues

The AVB-DMAC applies its separation filtering mechanism to select the reception queue for storing a received frame. The AVB-DMAC stores all received frames in the URAM.

There are two conditions for the AVB-DMAC to discard a received frame.

- Detection of an error during reception by the E-MAC
  - Whether error frames are discarded or stored in reception queue 0 (best effort) depends on the setting of the error frame enable bit in the receive configuration register (ETNBnRCR.EFFS). If error frames are to be stored (ETNBnRCR.EFFS = 1), they are always stored in queue 0 (best effort). In this case, characteristics specific to the queue (e.g. truncation) will vary. If the storage of timestamps for reception queue 0 (best effort) is enabled (the timestamp enable bit in the receive configuration register ETNBnRCR.ETS0 = 1), timestamps are stored even for error frames.
  - The separation filter is unable to determine where the frame data should be stored.
- Receive frame failure in the separation filter
  - Whether discarding error frames or storing in reception queue 0 (best effort) depends on the stream filtering select bit (ETNBnRCR.ESF) in the receive configuration register.

The flowchart in **Figure 25.21** shows how the AVB-DMAC selects the reception queue in accord with the frame type, including judgment by the separation filter. Selection of the queue starts when the E-MAC completes frame reception. The result is storage of the frame in the proper queue or the frame being discarded.

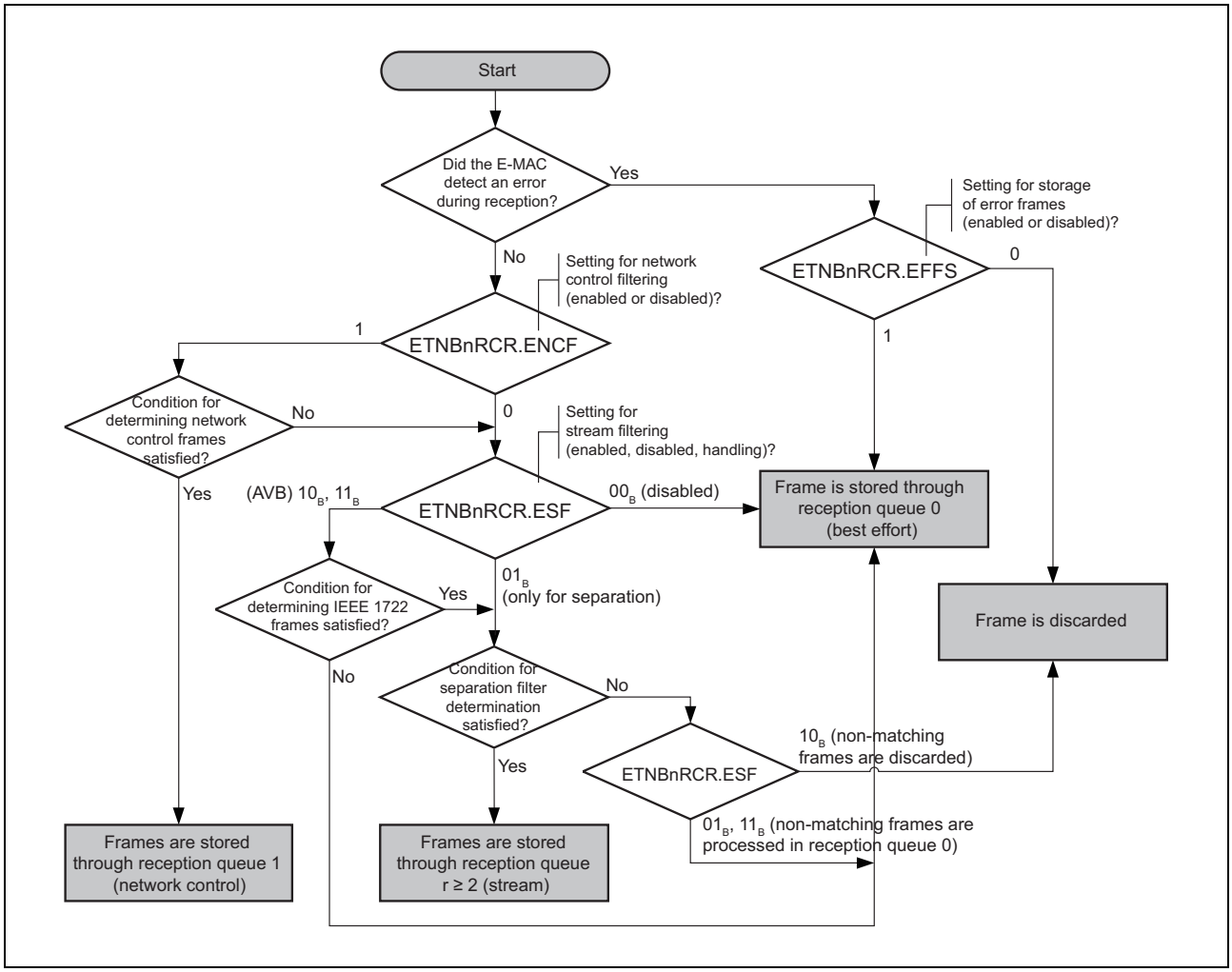


Figure 25.21 Mechanism of Reception Queue Selection

Notes on the meanings of entries in the flowchart

- "Condition for determining network control frames"
  The Ethernet destination address (DA) is 01:80:C2:00:00:0E.
  The Ethernet type (ET) is 88:F7.
- "Condition for determining IEEE 1722 frames"
  The Ethernet destination address (DA) is within the range from 91:E0:F0:00:00:00 to 91:E0:F0:00:FE:FF.
  The VLAN tagged TPID (tag protocol identifier) field (VL) is 81:00.
  The Ethernet type (ET) is 22:F0.
- "Condition for separation filter determination"
  See **Section 25.4.4.1(1), Separation Filtering**.

Figure 25.22 shows the allocation of bits related to the network and stream types in Ethernet frames. The preambles of Ethernet frames are not taken into account.

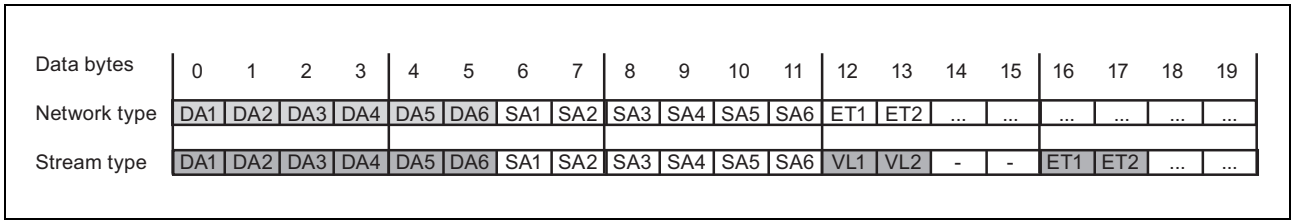


Figure 25.22 Data Bytes of Ethernet Frames Used in Classification

(1) Separation Filtering

Separation filtering involves the checking of up to 64 bits (eight successive bytes) in received Ethernet frames. The setting for the first byte (i.e. the setting of the separation filter offset configuration register (ETNBnSFO.FBP)), selects the part of frames to be used in separation filtering. There is also a common filter mask (separation filter mask configuration register i(ETNBnSFMi.CFM)), which limits the separation filter processing to a bit mask that can be set to the lower number of bytes or can be set appropriately.

Examples

To use one byte in separation, set separation filter mask configuration register 0 (ETNBnSFM0.CFM) to 0000 $00FF_H$ and separation filter mask configuration register 1 (ETNBnSFM1.CFM) to 0000 0000_H.

To use seven bytes in separation, set separation filter mask configuration register 0 (ETNBnSFM0.CFM) to FFFF $FFFF_H$ and separation filter mask configuration register 1 (ETNBnSFM1.CFM) to 00FF $FFFF_H$.

CAUTION

If bits at some positions are set to 0 in the separation mask, in order to match with the pattern, the bits at the corresponding positions of the pattern must also be set to 0. Only those bits in which the separation filter pattern configuration register i (ETNBnSFPi.FPs) setting is equal to the separation filter mask configuration register i (ETNBnSFMi.CFM) are sorted by matching with received data.

Figure 25.23 shows separation filtering. The selected data from a received frame (Rx_Frame[63:0]) are masked by the common filter mask. As a result, the selected frame data can be obtained. This value is compared with all filter patterns. The separation filter circuit in the AVB-DMAC selects the filter pattern that matches the queue having the lowest index s or selects a flag to indicate that there is no matching separation pattern.

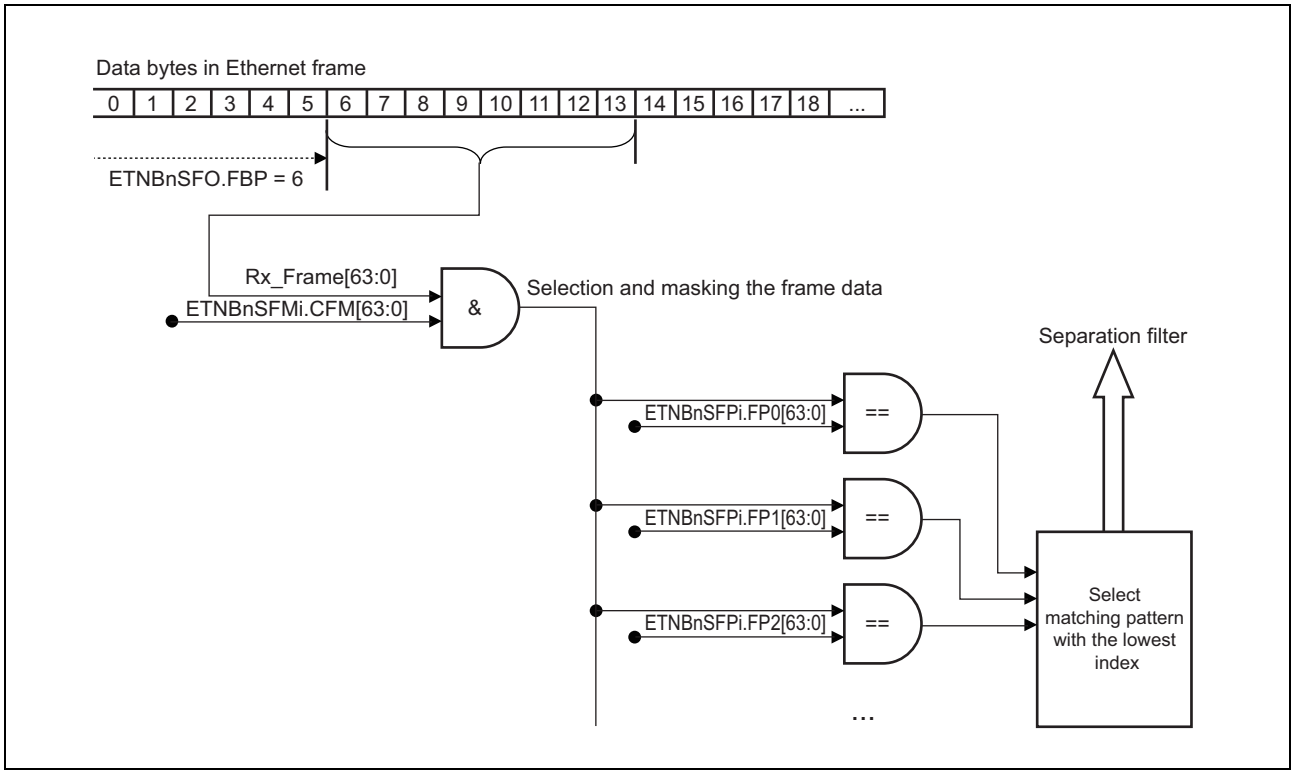


Figure 25.23 Separation Filtering

(2) Stream separation

The AVB-DMAC applies separation filtering to sort frames received in streams. An AVB network has a concept of “Talker” and “Listener”. A Talker is an end station that generates one or more streams. A Listener is an end station that has the role of being a sink for at least one stream. The various A/V streams are identified by 8-byte stream IDs.

The number of end stations within an AVB network and their roles differ with the application.

The stream ID is a general pattern of the AVB network for identifying one stream. **Figure 25.24** shows the bit allocation of bits in IEEE1722 Ethernet frames and stream ID fields.

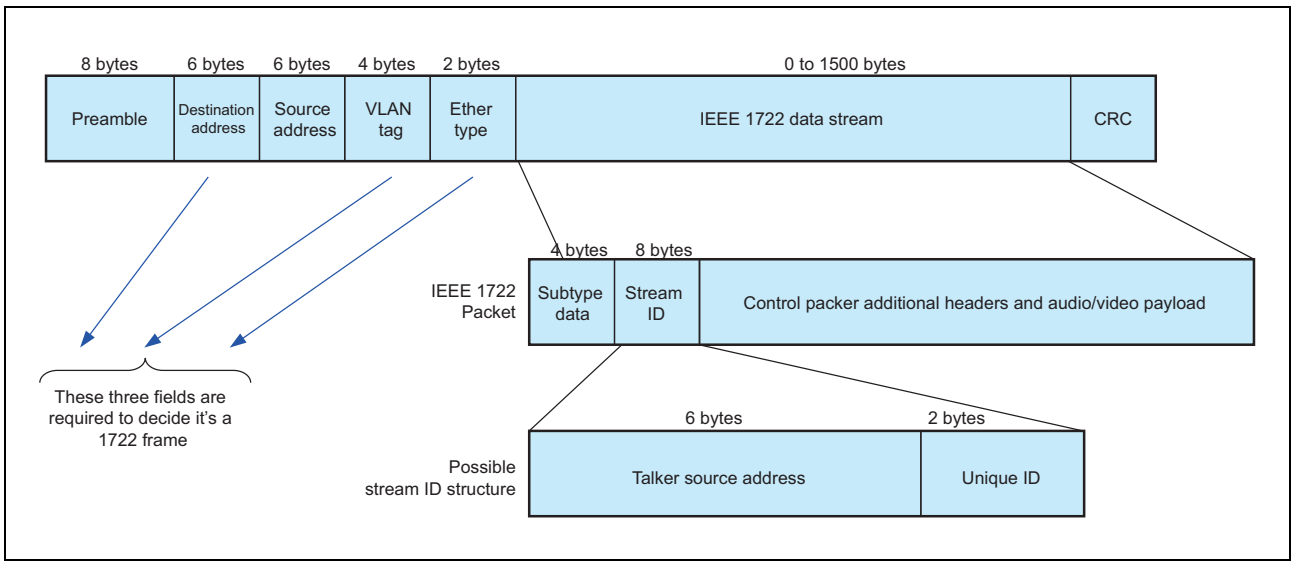


Figure 25.24 IEEE 1722 Frame Layout and Stream ID

The IEEE 1722 standard stipulates that the stream ID field starts from the 23rd byte (not counting the preamble). Accordingly, set the separation filter offset (ETNBnSFO.FBP) to 22 in operations on IEEE 1722 streams. Set the separation filter mask (ETNBnSFMi) and separation filter pattern (ETNBnSFPi) in accord with the specification of the product in which the chip is being used.

Example: In the example of a stream ID shown in **Figure 25.24**, the current application divides the field into the talker source address and the unique stream ID. The unique ID is used to differentiate between multiple streams from the same talker. Based on this, there are two settings for separation filter masking:

- To divide various streams into individual queues, set ETNBnSFM0.CFM to FFFF $FFFF_H$ and ETNBnSFM1.CFM to FFFF $FFFF_H$.
- To divide streams from respective talkers into individual queues, set ETNBnSFM0.CFM to FFFF $FFFF_H$ and ETNBnSFM1.CFM to 0000 $FFFF_H$. This excludes the unique ID from the filter condition.

(3) Reconfiguration of separation filter during operation mode

In configuration mode the initial separation filter configuration is defined. To prevent inconsistent filter results during operation mode, direct parameter change is not possible; instead update mechanism by ETNBnSFVi.LV and ETNBnSFL.LC should be used.

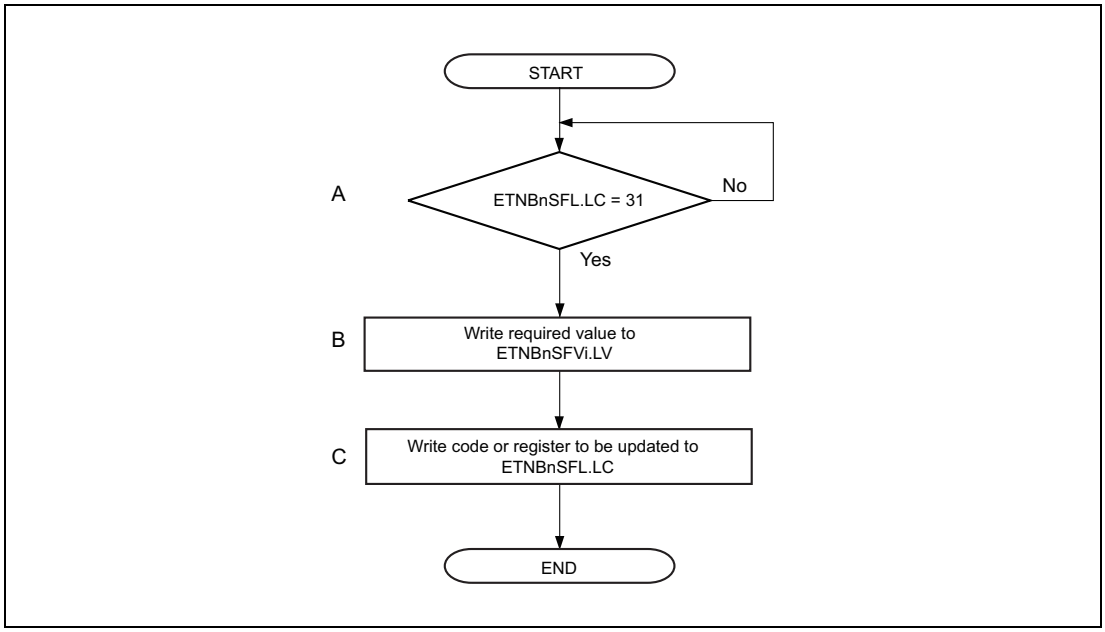


Figure 25.25 SW flow to update separation filter during operation mode

When reconfigure separation filter, AVB-DMAC guarantees consistent classification for one received frame. If reconfiguration requires a sequence of parameter changes, CPU has to order reconfiguration based on application demand.

AVB-DMAC updates filter configurations when no frame reception is handled. In worse case the update is delayed by one frame reception. Exact duration depends on PHY speed and frame size. By reading ETNBnSFL.LC CPU can observe when a previous update has been processed.

Note that frame classification happens during frame reception and storage into internal Rx-FIFO. There may be frames with old classification pending in Rx-FIFO or under storage to URAM when filter reconfiguration is accepted.

25.4.4.2 Setting Up Reception Descriptors

For reception, the descriptor mechanism is essentially as described in **Section 25.4.3, Descriptors**.

This section describes memory operations that are especially required in handling reception queues.

(1) Reception Descriptor Types

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 25.116 shows the descriptor types used in reception.

Table 25.116 Descriptor Types in Reception

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	No data is stored in receive queues. The ETNBnRIS2.QFFr bit indicates that queue r is full and the received frame is not stored. This descriptor is used again for processing of the next reception, if any.	Not changed
Frame Middle (FMID)	Same as FSTART	Not changed
Frame End (FEND)	Same as FSTART	Not changed
Frame Single (FSINGLE)	Same as FSTART	Not changed
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	A stop point defined by software has been reached. A frame of this type within a divided frame (writing of FMID or FEND) stops the frame being stored and the frame is lost. ETNBnRIS2.QFFr indicates that the frame has been lost. If this happens at the start of a frame (writing of FSTART or FSINGLE), storing of frames starts from the next descriptor. In either case, processing shifts to the next descriptor in the chain.	EEMPTY
Frame Empty (FEMPTY)	The descriptor can be used to store received data. Up to DESCR.DS bytes are stored in the descriptor data area. For details, see **Section 25.4.4.3(1), Storing Frame Data in the Descriptor Data Area**.	FSTART, FMID, FEND, or FSINGLE
Frame Empty Incremental Start (FEMPTY_IS)	The descriptor can be used to store received data. All data for the frame are stored in the descriptor's data area. DESCR.DPTR indicates the base address of the incremental data area. For details, see **Section 25.4.4.3(2), Incremental Data Areas**.	FEND or FSINGLE
Frame Empty Incremental Continue (FEMPTY_IC)	The descriptor can be used to store received data. The remaining bytes of frame data are stored in the descriptor's data area. DESCR.DPTR is undefined, but is written back at the start position within the incremental data area after processing. For details, see **Section 25.4.4.3(2), Incremental Data Areas**.	FEND or FSINGLE
Frame Empty No Data storage (FEMPTY_ND)	The descriptor can be used to store received data. Up to DESCR.DS bytes are captured from the reception FIFO but not stored. After processing, DESCR.DS is written back as 0. For details, see **Section 25.4.4.3(c), No Data are Stored**	FSTART, FMID, FEND or FSINGLE
Link Empty (LEMPTY)	Same as FSTART	Not changed
EOS Empty (EEMPTY)	Same as FSTART	Not changed

(2) Configuration of Reception Frame Data Descriptors

Figure 25.26 shows the configuration of descriptors for use with reception queues. The reception-specific fields are the same whether the descriptor is normal or extended. The reception-specific fields (DESCR.MSC, DESCR.PS, DESCR.EI, and DESCR.TR) are described in **Table 25.117**.

For the other fields and the descriptor types, see **Section 25.4.3.6, Descriptor Type**.

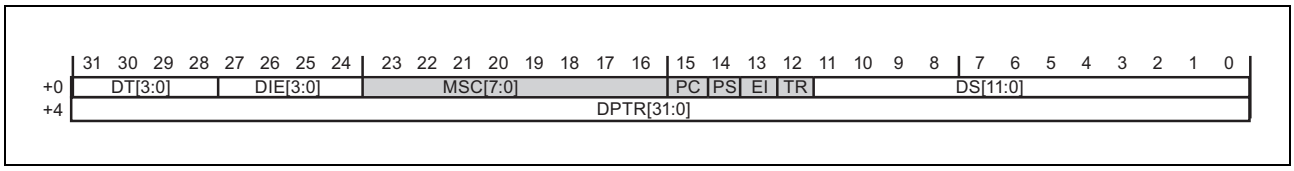


Figure 25.26 Configuration of Descriptor for a Received Frame

Table 25.117 Configuration of a Received Descriptor

Bit Name	Function
MSC	MAC Status Code These bits indicate errors in reception detected by the E-MAC. In the case of a divided frame, these bits are set to the same value within all descriptors for the frame data. Details of the bits are as follows. MSC[7]: Received frame has a multicast address. MSC[6]: Fixed to 0 MSC[5]: Fixed to 0 MSC[4]: Received frame has residual bits. MSC[3]: Received frame is too long*1 MSC[2]: Received frame is too short MSC[1]: Error in frame reception*2 MSC[0]: Received frame has a CRC error.
PC	Payload CRC Status This bit informs about result of payload CRC check. AVB-DMAC does not make any confirmation if a frame contains a payload CRC. CPU need to analyze frame content before evaluate this bit. 0: Payload CRC is not matching 1: Payload CRC is correct
PS	Padding Selection This bit specify whether frame data are to be padded when stored in this descriptor. Insertion of padding data is in accord with the settings in the ETNBnRPC register. 0: Padding is not to be inserted. 1: Padding data may be inserted. This depends on the ETNBnRPC settings.
EI	Error Indication This bit indicates the detection of an error in frame data while a frame was being stored. The bit is set to 1 for a descriptor in which an error has been detected. If the descriptor is for a divided frame, storage of the frame is aborted. 0: No error 1: Error is detected
TR	Truncation Indication This bit will be set if received data is more than 4092 bytes, and received data will be truncate to 4092 bytes. These bits are set to the same value within all frame data descriptors for a divided frame. 0: Data have not been truncated. 1: Data have been truncated.

Note 1. "Received frame is too long" is set when MAC receive the frame that length is more than ETNBnRFLR configuration value.

Note 2. "Error in frame reception" is set when MAC detect AVB_RX_DV=1 and AVB_RX_ER=1 while reception.

CAUTION

The ETNBnRCR.EFFS bit specifies whether or not frames with errors detected by the E-MAC are to be stored in the URAM. When the storing of error frames is disabled, error codes are not written to DESCR.MSC.

25.4.4.3 Reception Processing

After initialization, the AVB-DMAC becomes able to select a proper reception queue and store it in the descriptor data area in the URAM. Received data is stored into the URAM as long as the descriptor data area capacity suffices.

Received frames are classified and stored in the reception FIFO in accord with the algorithm described in **Section 25.4.4.1(1), Separation Filtering**. The frames are sorted, truncated, or discarded by the separation filter on reception by the MAC before stored in the reception FIFO. The following data are stored in the reception FIFO.

- MAC status of received frames
- Length of received frames
- Timestamp of received frames
- Target reception queue
- Received frame data

If the reception FIFO contains even one frame, the scheduler executes storing in the reception queue (see **Section 25.4.2.2, Scheduling Reception and Transmission**).

If there is even one empty data descriptor in a queue for which reception has started, the storage of frame data starts. Received frames for a queue that is already full (there is no empty frame descriptor or the UFC stop level has been reached) are discarded from the reception FIFO. This ensures that one queue being full does not prevent the storage of data in the other queues.

(1) Storing Frame Data in the Descriptor Data Area

Frame data for storage are assumed to be in either of the two patterns described below.

- The data for an entire frame will fit in the descriptor data area.
  - In this case, the descriptor type (DESCR.DT) is FSINGLE.
- Frame data to be stored in the descriptor data area arrive in divided form.
  - In this case, FSTART is written to the descriptor type (DESCR.DT) bits of the first of the frame data to arrive and FMID and FEND are written to the type bits of descriptors for subsequent data.

The descriptor type is updated by the AVB-DMAC in the last step of descriptor processing, so software can always access the descriptor assigned to DESCR.DT.

When normal synchronization mode is used, the CPU can write FEMPTYxxx directly to the descriptor type field after processing the stored element. Do not change the descriptor or any part of the descriptor data area after FEMPTYxxx is written to DESCR.DT.

(a) Storing Frame Data for a Whole Single Frame

For a frame with an FSINGLE descriptor, all data for the frame are held at the position defined by DESCR.DPTR. DESCR.DS indicates the length of the received frame.

If DESCR.DS is bigger than the actual size of a received frame, the FEMPTY or FEMPTY_ND descriptor is stored in place of the FSINGLE descriptor after processing.

Also, the FEMPTY_IS and FEMPTY_IC descriptors, which always hold the full frame data for the reception FIFO, are stored in place of the FSINGLE descriptor after processing.

(b) Storing Frame Data as Divided Frames

Divided frames are handled in the same way as a single frame. A frame stored with divided descriptors must be recombined before use. DESCR.EI and DESCR.TS are only valid in the last descriptor of the sequence for a divided frame.

CAUTION

If the data area size setting in DESCR.DS is not a multiple of four, the number of bytes set in DESCR.DS is fetched from the reception FIFO and the remaining bytes are used as the next storage area.

After a received frame is divided into different descriptors, handle each storage element separately, and allocate the descriptor type, intending software, after processing.

Accordingly, an error frame (FEMPTYxxx instead of FMID or FEND) may exist while a descriptor chain is being processed. In such a case, the CPU must postpone processing of the error frame to the next trigger point.

(c) No Data are Stored

The application specification may lead to some types of received frames being unimportant (for example, when the application only requires stream data from the Ethernet frames). Storing frames in divided form makes separating out the unnecessary parts of Ethernet frames possible.

If part of a divided frame is not required, use the FEMPTY_ND descriptor for that part so that it is not stored in the URAM. Not storing the data negates the need for bandwidth on the data bus, improving the overall performance.

When an FEMPTY_ND descriptor is processed, DESCR.DS is set to 0. This brings the frame data section of the descriptor into agreement with the FEMPTY type. DESCR.DS = 0 is for the unique identification of the descriptor after writing.

(2) Incremental Data Areas

Secure space in the URAM for storing received data. Even when data are placed in the URAM area such that all descriptor data areas of a chain are contiguous, a received frame being shorter than the descriptor data area will lead to an empty space. **Figure 25.27** shows an example of settings and the memory map.

Certain applications require that data areas be contiguous (e.g. when received data are to be processed other than by hardware as A/V codec modules). When the length of received frames differs (e.g. when payloads vary between having one or two A/V packages), the use of a static pointer in the descriptor produces empty spaces in the data area. This may necessitate direct additional processing to remove the empty spaces.

Accordingly, and to reduce the CPU load imposed by copying data, the AVB-DMAC supports an "incremental data area" function.

When incremental data areas are in use, all descriptors use a common data area for storage. One descriptor (FEMPTY_IS) defines the base address of the incremental data area and the next descriptor (FEMPTY_IC) in the descriptor chain to store receive data. **Figure 25.28** shows an example of settings and the memory map.

Use of an incremental data area does not reduce the memory space in the individual descriptor data areas.

The hardware and software synchronization strategy and performance are also not changed.

It is also possible to divide a frame up among various descriptors in a way that reflects its structure (e.g. one descriptor for the Ethernet header and one for the data payload).

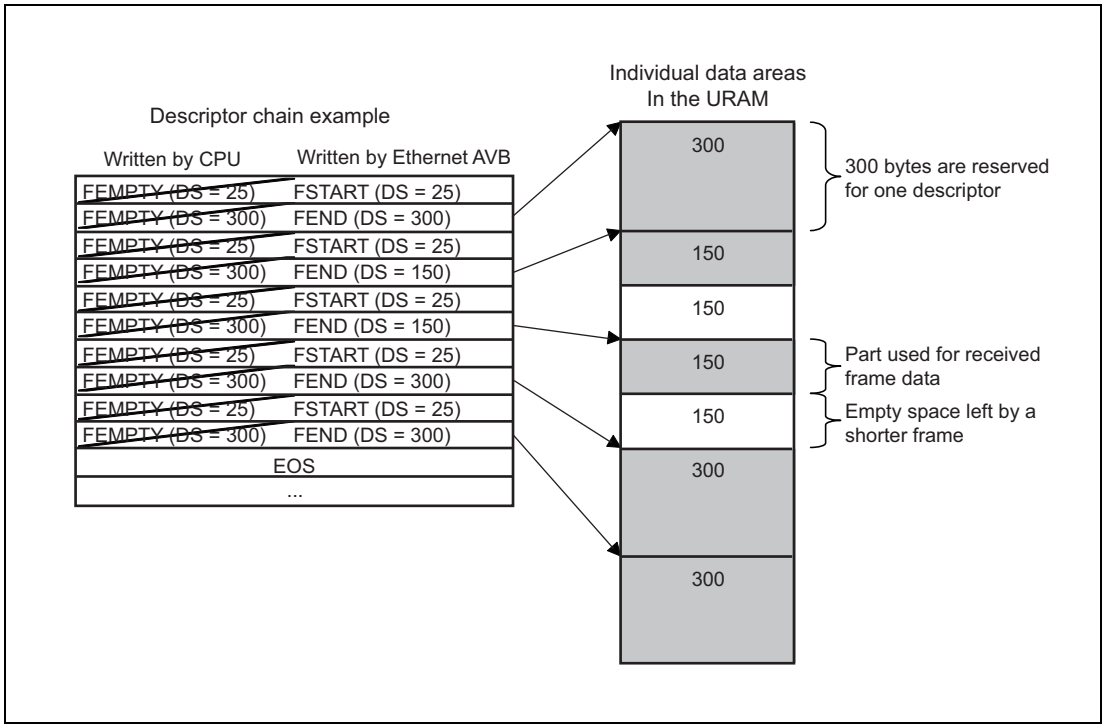


Figure 25.27 A Reception Queue Using Individual Descriptor Data Areas

Figure 25.27 and **Figure 25.28** show how control of the data storage areas by a descriptor chain varies according to whether individual or incremental data areas are in use. The chains are configured for storing received frames consisting of a 25-byte header ((which is treated as one descriptor outside of the range of this example) and a 150- or 300-byte payload (whether one or two 150-byte payload packages are transmitted with one Ethernet frame depends on the data source).

In **Figure 25.27**, the EOS descriptor is added as an example of a re-synchronization point. If the frame source transmits a frame containing more than 325 bytes, the frame will be divided among three descriptors, meaning that synchronization of the header and data sequences is lost. Despite this, however, the frame is not divided across the EOS descriptor, so even if the synchronization is off before EOS, AVB-DMAC handles the next descriptor chain as normal. The EOS is not required because the incremental descriptor always stores all data being processed while an incremental data area is in use.

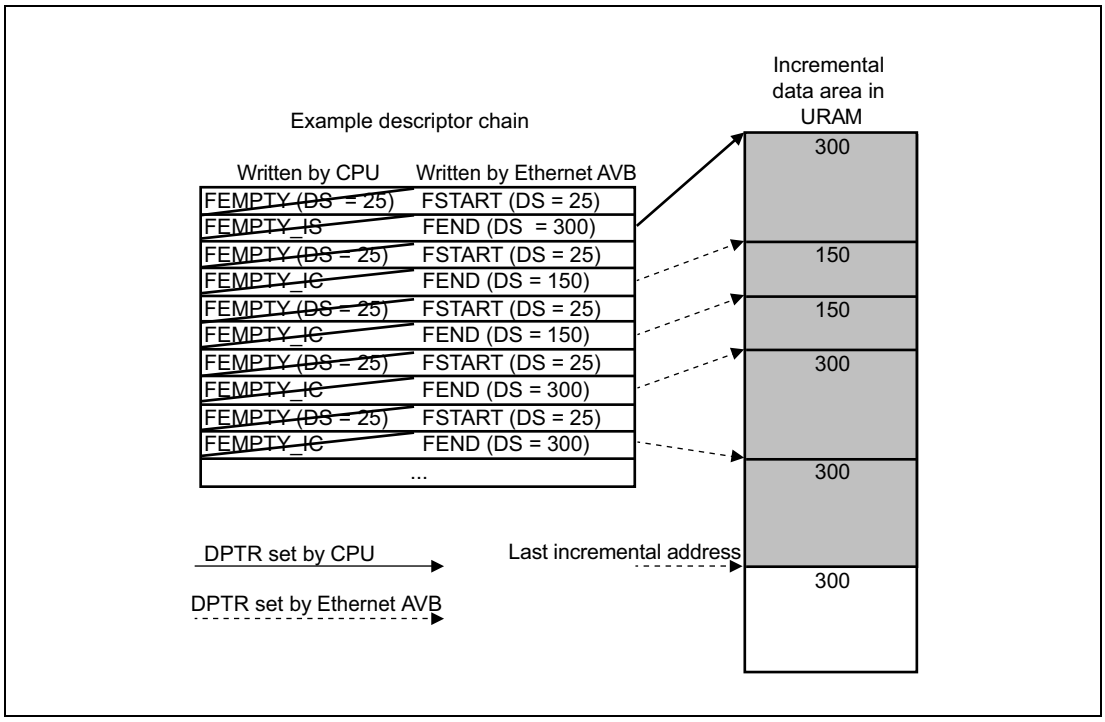


Figure 25.28 Reception Queue Using a Common Incremental Data Area

As **Figure 25.28** shows, when data are stored in an incremental data area, the descriptor pointers in the FEMPTY_IC descriptors (DESCR.DPTR) are updated. Accordingly, the resulting FEND or FSINGLE descriptor is in the same format as after writing to an FEMPTY descriptor.

Software captures received data from an incremental data area that has no empty storage areas between frame data. All empty spaces are allocated to the end of the incremental data areas. If RQCi.PIAr is 0, incremental data areas are limited to multiples of four bytes. When the amount of data for storage in an incremental data area is not a multiple of four bytes, from one to three bytes of empty space will be produced. DESCR.DS can be read to check for such empty spaces.
If RQCi.PIAr is 1, there is no limitation about frame start position inside incremental data areas. No empty space is produced between frames.

It is not possible to directly control the amount of received data to be stored from the incremental descriptor (FEMPTY_IS, FEMPTY_IC) because other descriptors (FEMPTY and FEMPTY_ND) are also applicable with DESCR.DS. All received data in the chain are always stored in an incremental descriptor.

(a) Setting Up an Incremental Data Area

A descriptor chain in the incremental data area having N descriptors (one FEMPTY_IS and N-1 FEMPTY_IC) means that a storage area for the maximum of N times the capacity must be prepared.

As **Figure 25.28** shows, DESCR.DPTR of an FEMPTY_IS descriptor indicates the base address of the incremental data area. The DPTR in FEMPTY_IC descriptor indicates the address of next store data.

(b) Processing an Incremental Data Area Based on Descriptors

Since data processing by the CPU is the same regardless of how the AVB-DMAC stores the data, data stored in an incremental data area do not require any special handling.

(c) Padding

Use padding for received frame data that are not aligned correctly in the specified memory structure. Padding can be set individually for each descriptor. Accordingly, in the reception of divided frames, padding can be restricted to only those frames that require it (e.g. A/V payload data.)

Padding can also be used to optimize system performance in an incremental data area (e.g. to prevent inefficient access by aligning received data with 32-byte boundaries in the incremental data area), as well as to fulfill application-specific requirements for specified memory structures (e.g. formats required by other modules that will be processing the received data).

Padding can only be used in an incremental data area.

The value $0000\ 0000_H$ is always used in padding.

Padding is the addition of the number of words (from one to seven 32-bit words) set in the stored padding counter in the receive padding configuration register (ETNBnRPC.PCNT). This padding is repeatedly inserted in accord with the value in the stored data counter (ETNBnRPC.DCN) (from one to 255 32-bit words). When the stored data counter (ETNBnRPC.DCNT) reaches 0, however, padding is not repeated.

The first word of padding is always inserted at the position specified by DESCR.DPTR. When divided frames are in use, a padding word can be inserted at any byte position, and padding is handled on a 32-bit basis (e.g. an incremental data area where the first descriptor is for a 42-byte header data and the second descriptor holds padded payload data).

The next figure shows a general example of how padding is inserted and an example of setting up padding. A indicates frame data A received from the E-MAC, while B indicates frame data already stored in the descriptor data area (32-bit word units).

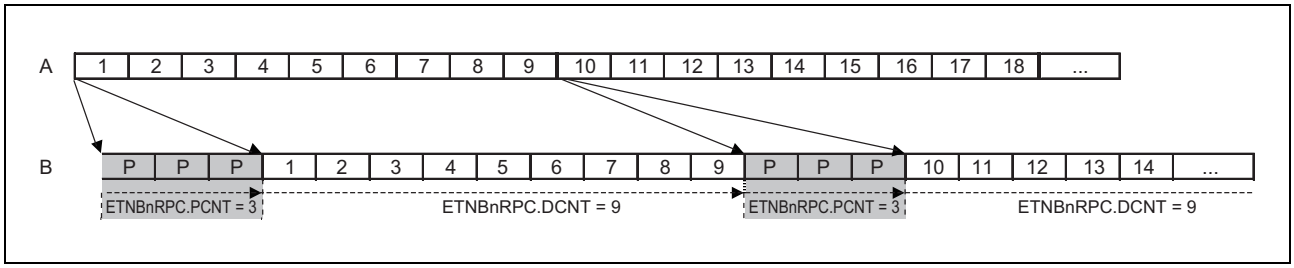


Figure 25.29 Example of a Padding Setting

Both padding and received frame data are counted in the descriptor size (DESCR.DS).

(3) Mode with Write-Back

Constructing a descriptor chain requires software (see **Figure 25.30**).

In the example in the figure, the variable SWdescr (software descriptor pointer) is a structure to identify a descriptor being processed. When Operation mode is entered and when the load base address function (ETNBnDLR.LBAq) is used, SWdescr should be initialized (start condition of flow).

The frame_processing() function processes the stored data. The function can use SWdescr.DT to check whether processing of a frame is completed. How frame data are processed differs with the application, so create functions that handle processing in accord with the specification.

The processing section is common to all modes of reception. The number of frames processed in response to each trigger can be restricted. When multiple frames have to be processed in a batch, waiting for individual trigger boxes must be skipped for these frames.

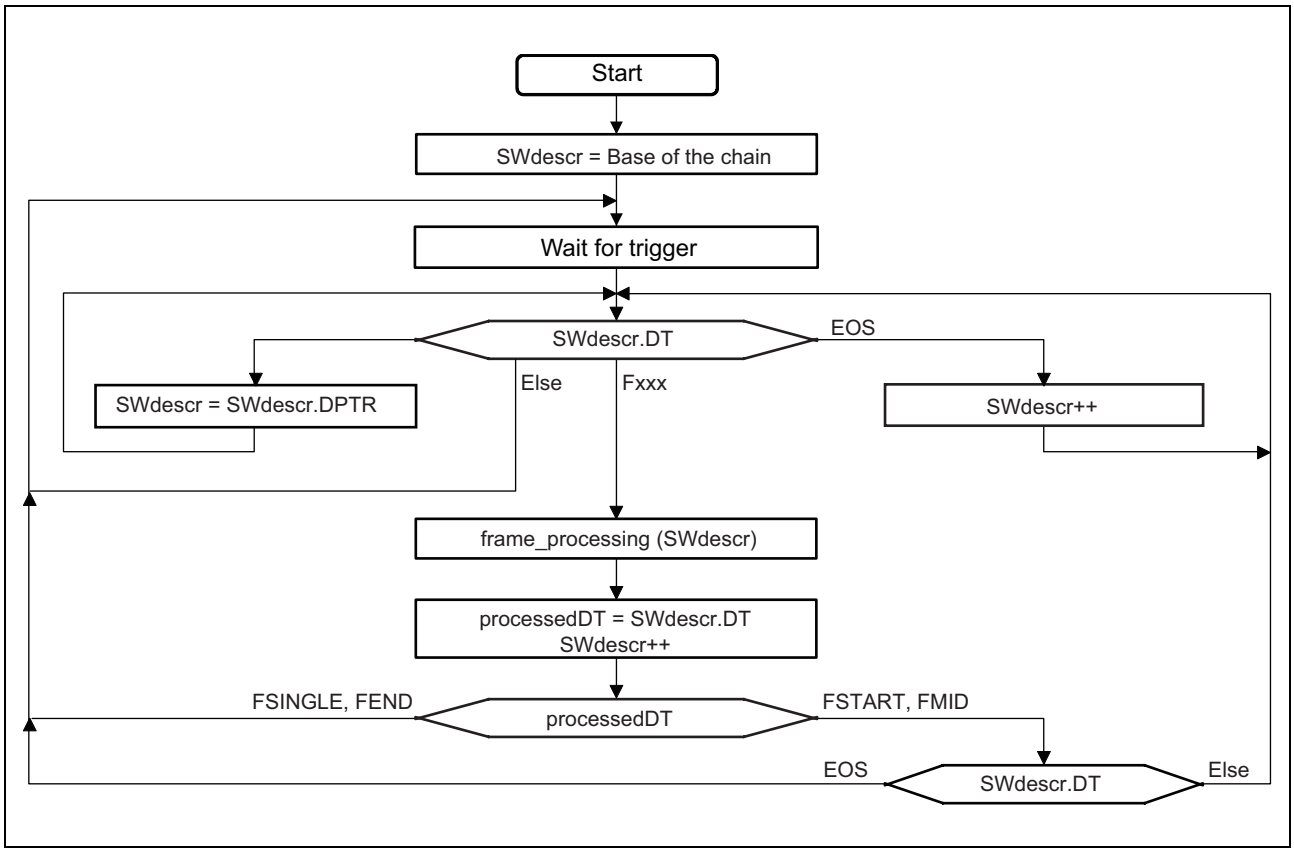


Figure 25.30 Flow of Reception Descriptor Processing (with Write-Back)

(4) Support for Reception Timestamps

Capturing reception timestamps is essential for IEEE 802.1AS time synchronization. Other types of received frames may also require that a reception timestamp be appended; this depends on the application. The AVB-DMAC supports reception timestamps based on the gPTP timer by storing timestamps that have been captured when the frame delimiter (SDF) for a received frame starts in the last frame data descriptor (FEND or FSINGLE). For the gPTP timer, see **Section 25.4.7.1, gPTP Timer**.

When timestamps are to be stored, use extended descriptors for the entire reception queue. Furthermore, timestamps are always stored for reception queue 1 (Network Control). Timestamps for reception queue 0 (best effort) and reception queue r ($r \geq 2$; for stream data) can be selected by the timestamp enable bits in the receive configuration register (ETNBnRCR.ETS0 or ETNBnRCR.ETS2).

25.4.4.4 Unread Frame Counters

Each reception queue has an unread frame counter (ETNBnUFCVi). Use the unread frame counter configuration bits in the receive queue configuration register (ETNBnRQCi.UFCCr) to select from among the four stop levels for each unread frame counter. The 0 setting disables the stop functions. For how to set this up, see **Figure 25.31**.

Operations of the AVB-DMAC (hardware) and CPU (software) drive an unread frame counter (UFC) in the following ways.

- The hardware indicates that it has added a new frame to the descriptor chain for the queue (this increments the counter).
- Software indicates how many frames from the descriptor chain it has processed by writing to the corresponding bits of the unread frame counter decrement register for the queue (this decrements the register by the number written).

The unread frame counter is based on the number of frames stored in the URAM and is only incremented by one even when a received frame is divided into different descriptors. Failure in storing a descriptor chain requires care because this may unread frame counter may fail in synchronization as described in **Section 25.4.4.4(1), Unread Frame (UFC) Synchronization Failure**.

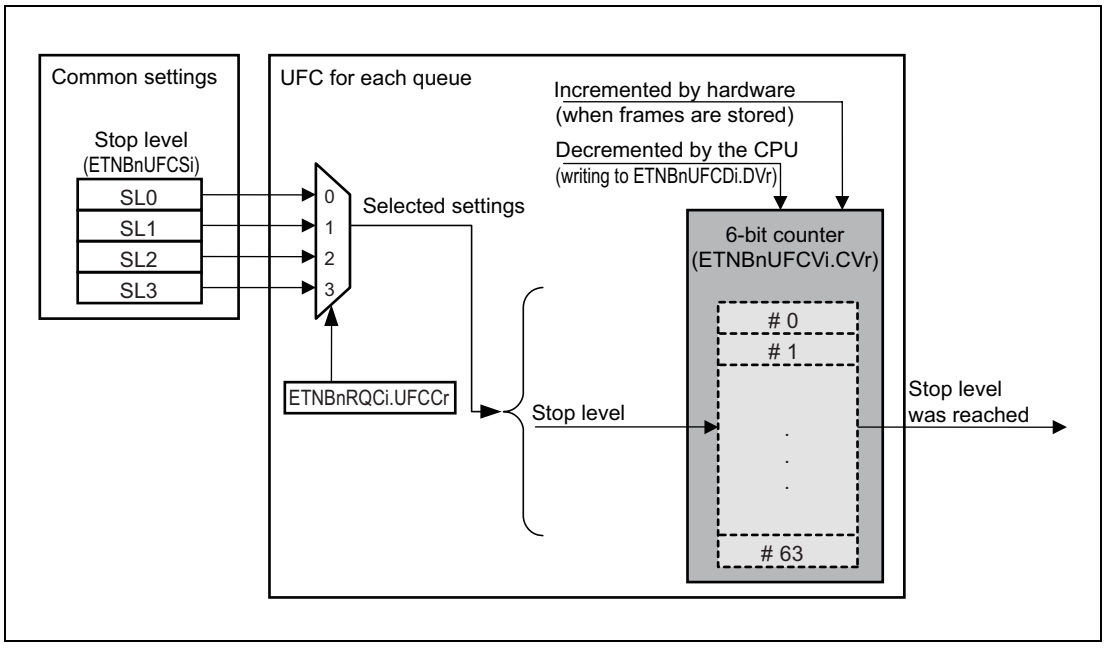


Figure 25.31 Overview of an Unread Frame Counter

Unless synchronization of hardware and software is not established, the current unread frame counter value (ETNBnUFCVi.CVr) indicates the number of unread frames in the queue.

The indicator that the stop level has been reached prevents the storage of further received frames in the descriptor chain. Selecting 0 as the stop level disables this function. Otherwise, further received frames for the queue are discarded once its unread frame counter reaches the stop level. When the unread frame counter stop function activates, the receive queue full interrupt flag in the receive interrupt status register 2 (ETNBnRIS2.QFFr) is set.

Set the unread frame counter stop level configuration register (ETNBnUFCS) for each reception queue that will use the unread frame counter function while the current operating mode is configuration mode.

(1) Unread Frame (UFC) Synchronization Failure

The unread frame counters do not recognize failure to store a frame in the URAM. In other words, the AVB-DMAC increments the counter for a queue each time it captures a frame for that queue from the reception FIFO whether or not it succeeds in storing the frame normally in the descriptor chain.

In general, synchronization of hardware and software fails under the following conditions.

- An unread frame counter reaching its maximum value
  When the value of a counter in an unread frame counter register i (ETNBnUFCVi) (i = 0 to 4) reaches 63, synchronization can fail.
  The CPU can only judge that a failure in synchronization has not occurred when the stop level is set to 63.
- A queue not having enough space for a descriptor or the associated data
  In this case, the corresponding receive queue full interrupt flag (ETNBnRIS2.QFFr) in receive interrupt status register 2 is set.
  If an unread frame counter reaches its stop level write-back mode(ETNBnRQCi.RSM[1:0]=00_B), the receive queue full interrupt flag (ETNBnRIS2.QFFr) in the receive interrupt status register 2 is set. Software must respond to this.
- A problem occurring during access to memory

The unread frame counter may set flags to frames over the number of frames actually usable for a descriptor chain, and consequently, it causes synchronization to fail. To retrieve the correct starting point for operations, use the descriptor base address load request (ETNBnDLR.LBAq) for the given queue.

25.4.4.5 Payload CRC support

AVB-DMAC supports checking a CRC which is located at the end of Ethernet frame data directly before Ethernet Frame Check sequence. Number ETNBnPCRC.CAS defines the first data byte of an Ethernet frame which is part of the payload CRC in range of 0 to 255. Refer to **Figure 25.3** about byte numbering inside an Ethernet frame.
The payload CRC should always be located before Frame Check Sequence. Note that the payload CRC cannot be evaluated if the received frame has padded data after payload CRC to achieve a minimum frame length of 64 bytes (including Frame Check Sequence). The CRC is checked at the point where the received frame data is taken from Rx-FIFO and assembled for writing via system bus to URAM.
The Payload CRC protects complete data path from source node creating the Ethernet frame up to the point the Ethernet frame is written to URAM.
Each frame received is checked for a potential payload CRC. It is up to CPU to analyze frame content to identify if a frame contains CRC (e.g. by frame type or VLAN ID). Only for frames CPU knows that there is a CRC inside, the result of HW CRC check in DESCR.PC should be evaluated.
The CPU should not use padding function for queues where frames containing payload CRC are stored into.

(1) Example of checking CRC used by Autosar

In Autosar End To End Extension, there is a payload format defined containing a payload CRC.
Figure 25.32 illustrates a frame if IEEE 802.3 basic frame format is used to transfer an Autosar E2E frame. In this case ETNBnPCRC.CAS should be set to 13 to check a CRC over all parts of Autosar E2E frame. In case of using additional frame headers (e.g. for 802.1Q VLAN tagged frames) the start position differs.

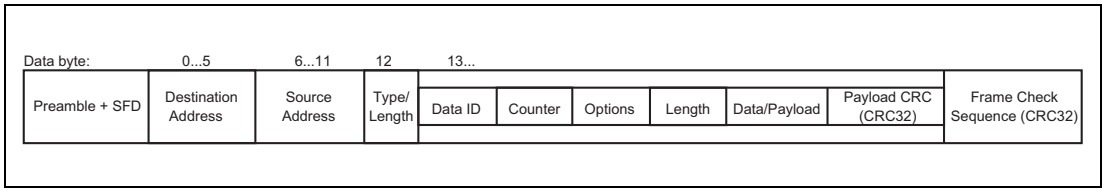


Figure 25.32 Frame format used by Autosar E2E

(2) Algorithm of payload CRC check

AVB-DMAC provides payload CRC check based on 32-bit polynomial F4AC_FB13$_H$ with initial value of FFFF FFFF$_H$ and a check value of 6FB3_2240$_H$.

These reference frames a used for functional confirmation, where the last listed byte is last byte of 60 byte frame.

[52 random data] 00 00 00 00 40 22 b3 6f
[53 random data] F2 01 83 25 1a 72 4f
[52 random data] 0F AA 00 55 f8 2d 66 20
[52 random data] 00 FF 55 11 6e 99 d7 9b
[47 random data] 33 22 55 AA BB CC DD EE FF 3d 34 5a a6
[53 random data] 92 6B 55 78 8a 68 ee
[52 random data] FF FF FF FF ff ff ff ff

(3) Mechanism to check Payload CRC logic

The Payload CRC logic can be checked by using loop back mode during initialization time. By this method CPU has full control over the frame to be checked.
Because payload CRC is stored as normal frame data in URAM, during operation time CPU can confirm DESCR.PC flagging by re-calculating CRC in SW.

25.4.5 Transmission Control

Areas in the URAM for storing transmission descriptors must also be secured (for descriptors, see **Section 25.4.3, Descriptors**).

The AVB-DMAC fetches data from the URAM in accord with the procedure the descriptor describes. Descriptors are retaining information about tags of transmit frames, too. The tag information is used to maintain the relationships between state information and timestamps for the software and the AVB-DMAC. After completion of transmission of frames, information about the statuses and timestamps of transmitted frames are accessible.

25.4.5.1 Transmission Modes

The AVB-DMAC has two modes of transmission.

- AVB transmission mode
  Selection of the transmit configuration register transmission queue priority (ETNBnTGC.TQP[1:0] bits set to 01_B or 11_B).
- Non-AVB transmission mode
  Selection of the transmit configuration register transmission queue priority (ETNBnTGC.TQP[1:0] bits set to 00_B).

(1) AVB Transmission Mode

AVB transmission supports the control of traffic through the output port to implement various traffic classes.

(a) Support for Traffic Classes and Associated Priority

When transmission is in AVB transmission mode, streams of traffic are transmitted in accord with the part of the AVB specification called Forwarding and Queuing for Time Sensitive Streams (FQTSS; for details on this, see the IEEE 802.1Q standard).

In the AVB specification, at least one queue for a reserving stream under the Stream Reservation Protocol (SR stream) and at least one queue for a non-SR stream are present, and high-priority queues are reserved for SRP traffic.

The AVB-DMAC supports four traffic classes: SR Class A, SR Class B, Network Control (NC) traffic (gPTP frames), and best effort (BE) traffic. Allocating a specific queue to Network Control (NC) frames ensures the control of synchronization.

The AVB-DMAC realizes compliance with the AVB standards by handling queues with the following architecture (in terms of traffic classes).

- Four transmission queues (Q3, Q2, Q1, and Q0) are available.
- Q3 and Q2 are for SR streams (one each for Class A and Class B).
- Q1 is for low-bandwidth Network Control (NC) traffic (gPTP frames)
- Q0 is for other types of traffic (MSRPDU*1, MVRPDU*2, best effort (BE), etc.)

Note 1. MSRPDU: Multiple Stream Registration Protocol Data Unit
Note 2. MVRPDU: Multiple VLAN Registration Protocol Data Unit

Fetching from queues proceeds in order of priority of the above traffic types. Three systems of priority are available through the setting of the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP[1:0]). In the default priority scheme, which is called AVB mode 1 (selected by ETNBnTGC.TQP[1:0] = 01_B), operation of the AVB-DMAC is fully in accord with the AVB specification. AVB mode 2 (transmit queue priority bits (ETNBnTGC.TQP[1:0] = 11_B) is an alternative priority scheme and varies from the AVB specification. Using this scheme thus requires more care. The other setting (ETNBnTGC.TQP[1:0]=00_B) is for non-AVB-mode transmission.

Table 25.118 Default and Alternative Priority Orders in AVB Transmission Mode

Priority Schemes (AVB Mode)	Priority Order of Queues
AVB mode 1 (Default)	Q3 (SR Class A) > Q2 (SR Class B) > Q1 (NC) > Q0 (BE)
AVB mode 2 (Alternative)	Q1 (NC) > Q3 (SR Class A) > Q2 (SR Class B) > Q0 (BE)

(b) Transmission Selecting Algorithm and CBS

The algorithm the AVB-DMAC applies to select frames for transmission is in accord with the specifications under section 8.6.8, Transmission selection, of the IEEE 802.1Q standard. For AVB mode, the CBS (credit-based shaping) algorithm is applied to the Class A and Class B SR queues (Q3 and Q2). Use of the CBS enables correct handling of the priorities of transmission from the SR queues. For the CBS algorithm, see **Section 25.4.6, CBS (Credit-Based Shaping)**.

When all the following conditions are determined as True, an SR queue (Q3 or Q2) is selected and transmitted at the specified time.

- The queue contains at least one frame ready for transmission.
- The queue has available credit.
- Unless an SR queue satisfies the above conditions, a higher priority queue is not present (not ready for transmission).

A non-SR queue (Q1 or Q0) is selected if the conditions below both hold.

- The queue contains at least one frame ready for transmission.
- As well as the above condition, a higher priority queue is not present (not ready for transmission).

Figure 25.33 and **Figure 25.34** are flowcharts of selection for transmission in AVB mode 1 (default) and AVB mode 2 (alternative).

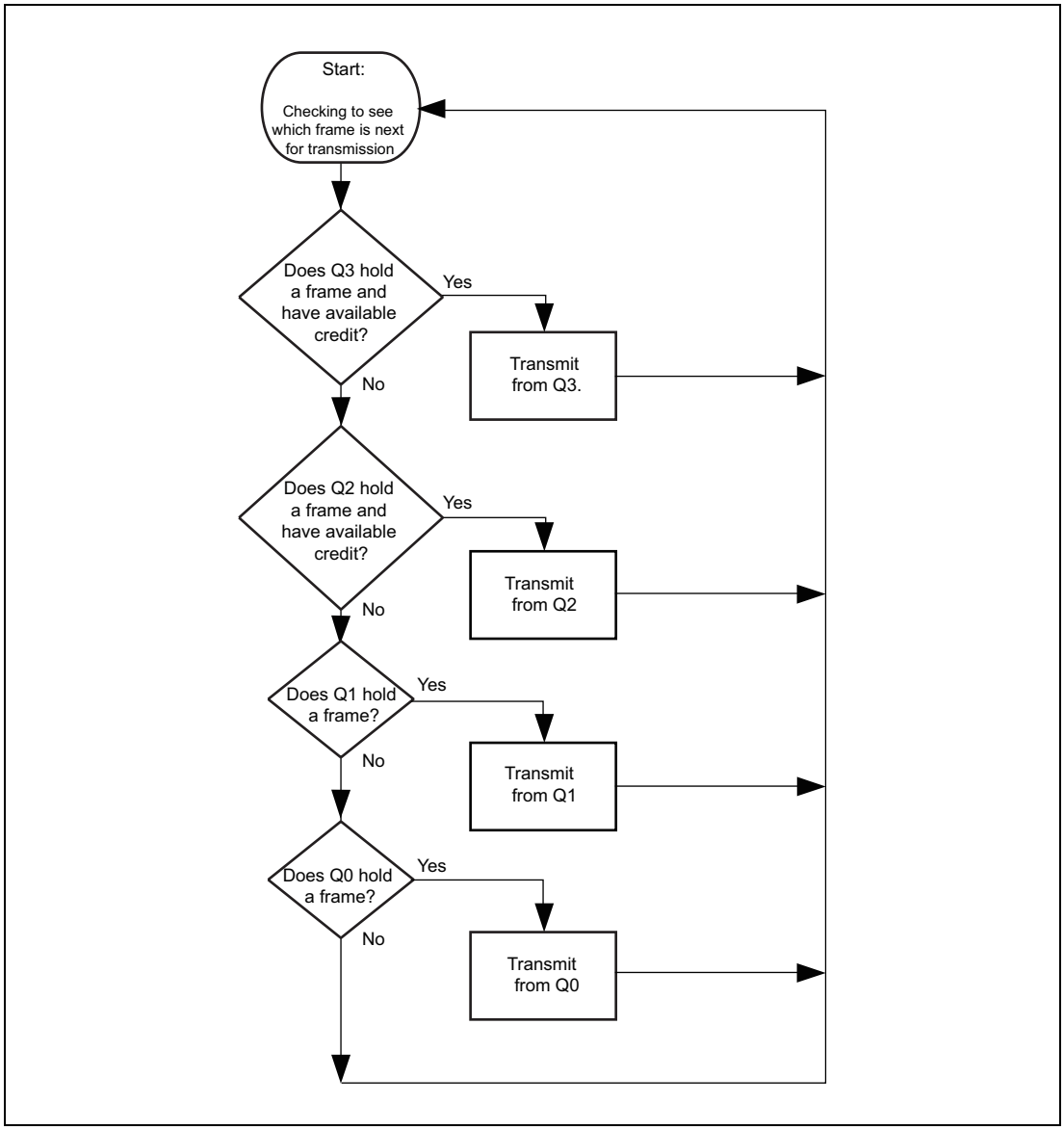


Figure 25.33 Flow of Selection for Transmission in AVB Mode 1 (Default)

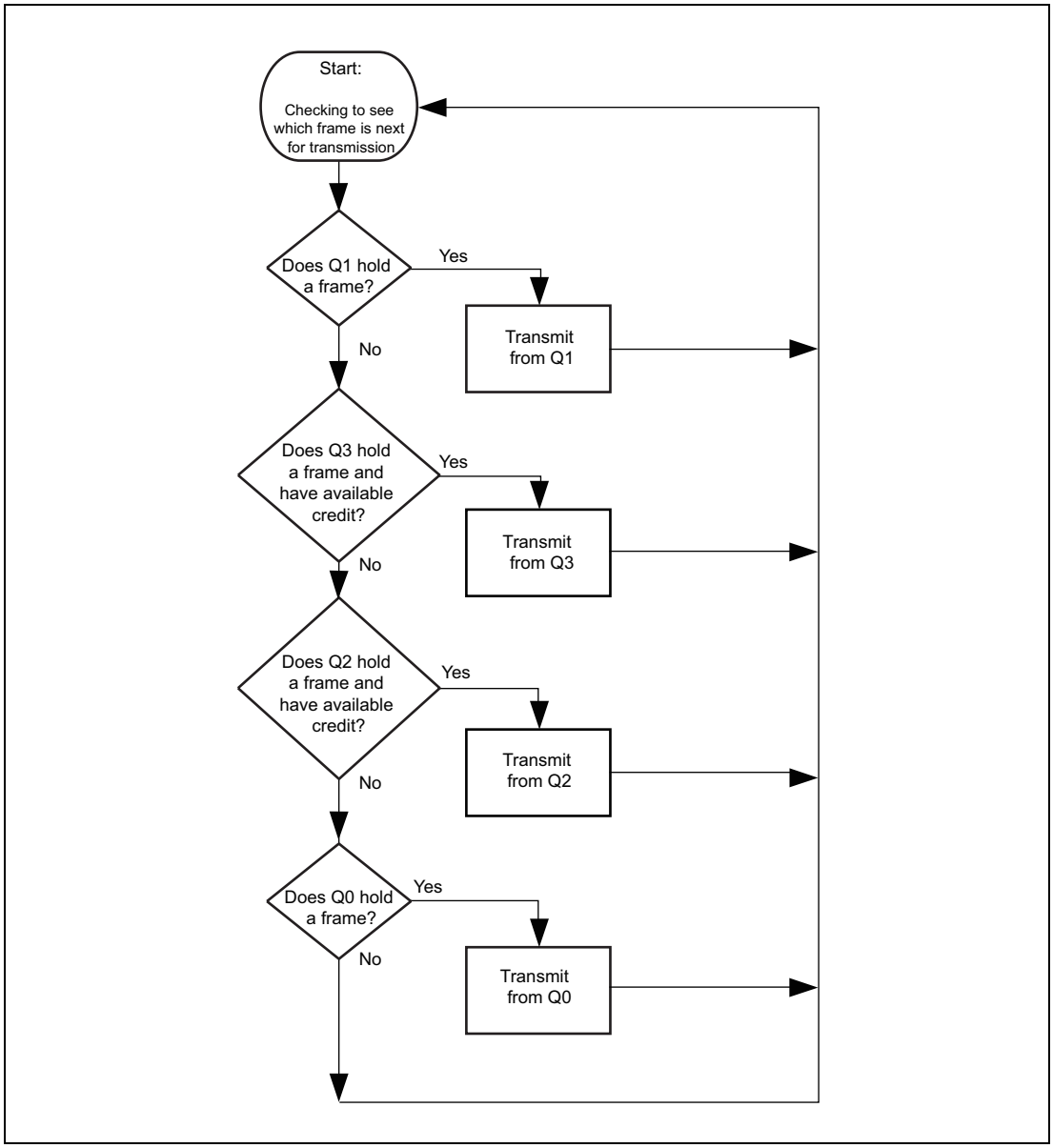


Figure 25.34 Flow of Selection for Transmission in AVB Mode 2 (Alternative)

(2) Non-AVB Transmission Mode

In non-AVB transmission mode, an absolute priority scheme is used. The SR Class is not supported and the CBS algorithm is not used.

In non-AVB transmission mode (when the transmit queue priority bits in the transmit configuration register (ETNBnTGC.TQP[1:0]) are 00_B), data is fetched for transmission in a strict order of priority (Q3 > Q2 > Q1 > Q0).

Figure 25.35 shows the flow of selection in non-AVB transmission mode.

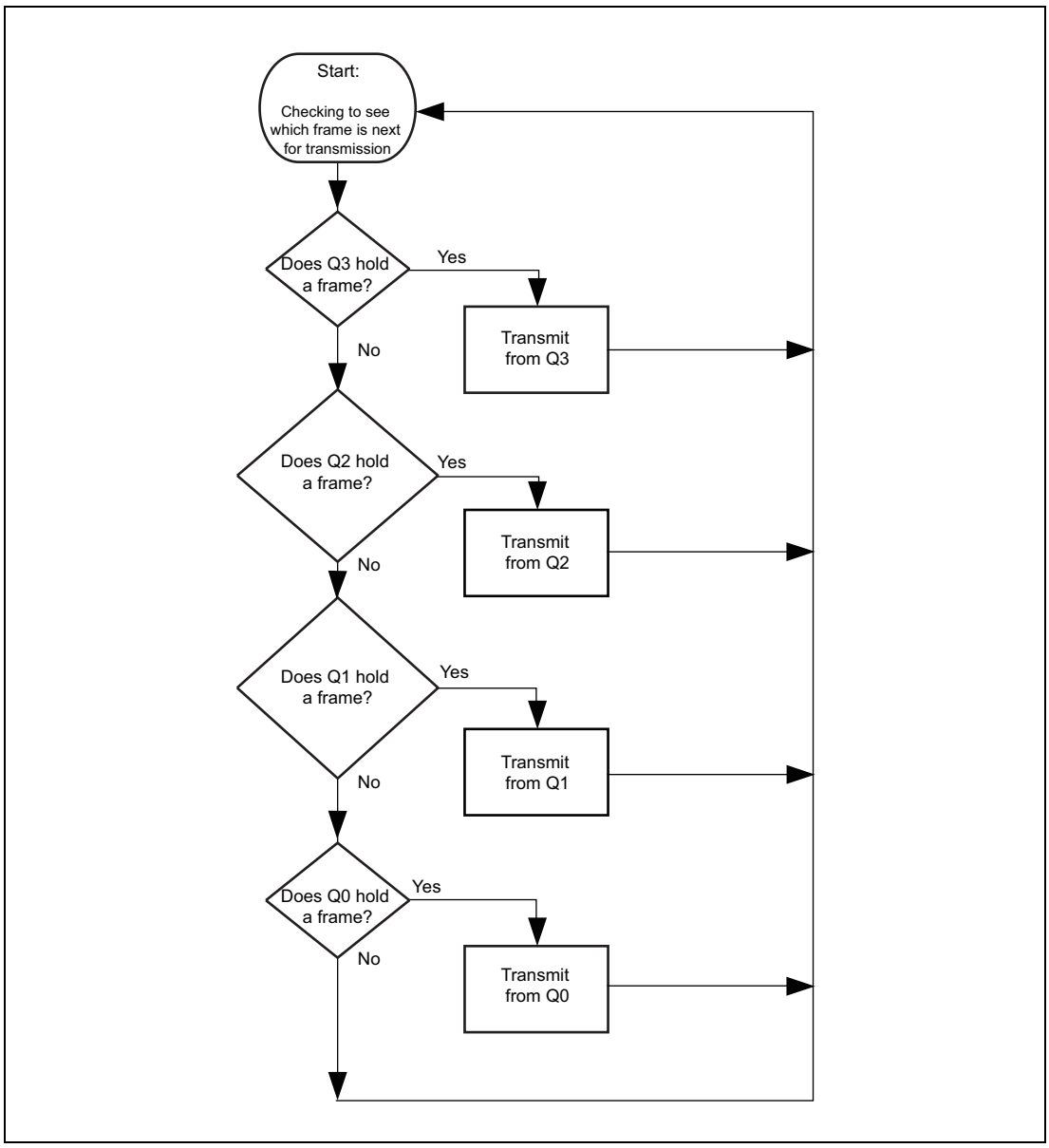


Figure 25.35 Flow of Selection for Transmission in Non-AVB Mode

(3) Setting the Size of the Transmission FIFO

The transmission FIFO is made up of 122 clusters. Each cluster can hold up to 128 bytes.

The size of the part of the transmission FIFO for use by each of the four transmission queues can be set by the corresponding transmit queue configuration t bits in the transmit control register (ETNBnTGC.TBDt). The maximum number of clusters required can be determined from the maximum length of frames for transmission from the queue t.

General Usage Examples:

Q0: Frames containing up to 1500 bytes → 1500/128 = 11.7 → 12 clusters

Q1: Frames containing up to 1024 bytes → 1024/128 = 8.0 → 8 clusters

Q3: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

Q4: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

When the depth of all transmission queues is 2, the following number of clusters is required.

2 × (12 + 8 + 16 + 16) + 16 = 2 × 52 + 16 = 120

25.4.5.2 Setting Up Transmission Descriptors

(1) Transmission Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 25.119 shows the descriptor types used in transmission.

Table 25.119 Descriptor Types in Transmission

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	The AVB-DMAC fetches the first of the data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame Middle (FMID)	The AVB-DMAC fetches the second or subsequent data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame End (FEND)	The AVB-DMAC fetches the last of the data for the divided frame. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Frame Single (FSINGLE)	The AVB-DMAC fetches the frame of data. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	This is a transmission stop point defined by software This leads to clearing of the transmit start request bit (ETNBnTCCR.TSRQt), which stops transmission. When the ETNBnTCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing proceeds to the next descriptor.	EEMPTY
Frame Empty (FEMPTY)	No frame data are ready for transmission This leads to clearing of the transmit start request bit (ETNBnTCCR.TSRQt), which stops transmission. When the ETNBnTCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing starts at this descriptor.	Not changed
Link Empty (LEMPTY)	Same as FEMPTY	Not changed
EOS Empty (EEMPTY)	Same as FEMPTY	Not changed

(2) Configuration of Transmission Frame Data Descriptors

Figure 25.36 shows the configuration of descriptors for use with transmission queues. The transmission-specific fields (DESCR.TSR, and DESCR.TAG) are described in **Table 25.120**.

For the other fields and the descriptor types, see **Section 25.4.3.6, Descriptor Type**.

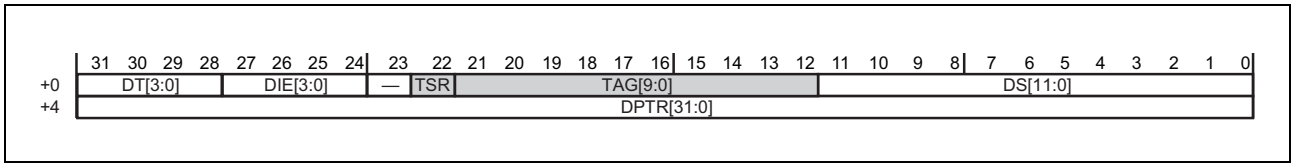


Figure 25.36 Configuration of Descriptor for a Transmitted Frame

Table 25.120 Configuration of a Transmission Descriptor

Bit Name	Function
TSR	Timestamp Store Request This bit specifies whether the transmission timestamp is to be stored within the EthernetAVB module. 0: The timestamp FIFO within the EthernetAVB module does not retain a transmission timestamp. 1: The timestamp FIFO within the EthernetAVB module retains a transmission timestamp. Only control this bit while the current DESCR.DT is FEND or FSINGLE.
TAG	Frame Tag This TAG field is used to associate each frame data with a timestamp. Frame TAG is not required but is recommended. Only control this bit while the current DESCR.DT is FEND or FSINGLE.

For the timestamp FIFO function, see **Section 25.4.5.4, Timestamping in Transmission**.

25.4.5.3 Transmission

(1) Transmitting Frames

Setting the transmit start request bit in the transmit configuration control register (ETNBnTCCR.TSRQt) starts the transfer of frames to the corresponding transmission queue.

The descriptor to the current descriptor address of the queue (ETNBnCDARq.CDA) is read first.

If this descriptor is a descriptor for frame transmission (FSINGLE, etc.), the AVB-DMAC fetches the frame data from the data area indicated by the descriptor, writes FEMPTY back to the descriptor type (DESCR.DT) bits, then proceeds to processing of the next descriptor.

If the descriptor is not for transmission, processing is as dictated by the given descriptor (for these descriptors, see the descriptions in **Section 25.4.3, Descriptors**).

If a base address load request is issued for a descriptor chain while it is being processed (by setting 1 in the LBAq bit for transmission queue that is currently being processed in the descriptor base address load request register, ETNBnDLR), processing proceeds to the new descriptor chain. Changing the chain does not interrupt frame fetching, but note that frames that have not been fetched from the old chain remain where they are.

Figure 25.37 shows descriptor processing during transmission.

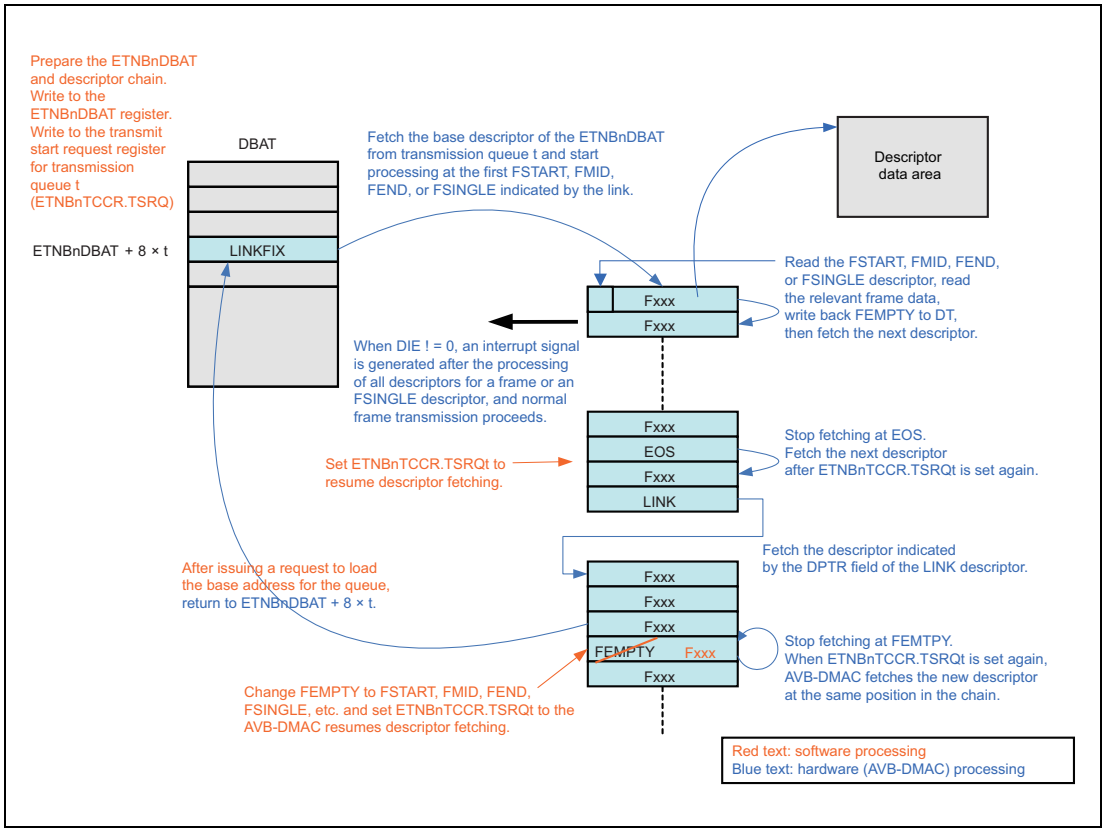


Figure 25.37 Descriptor Processing During Transmission

(2) Examples of Descriptor Usage

(a) Immediate Frame Transmission

Immediate frame transmission is a pattern in which fetching by the AVB-DMAC starts whenever software adds data to a queue. FEMPTY descriptors are used as HW/SW synchronization stop points.

Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 25.38 shows the flow for software implementing this pattern.

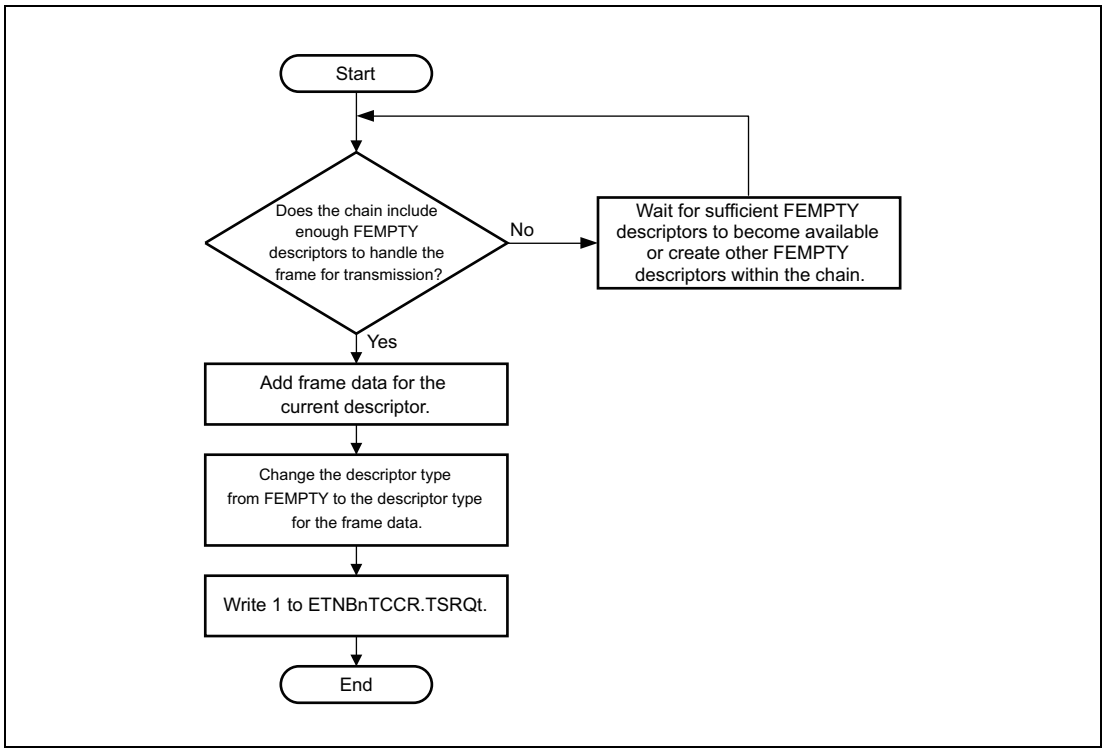


Figure 25.38 Software Flow for Immediate Frame Transmission

Figure 25.39 shows software and AVB-DMAC operations for immediate frame transmission.

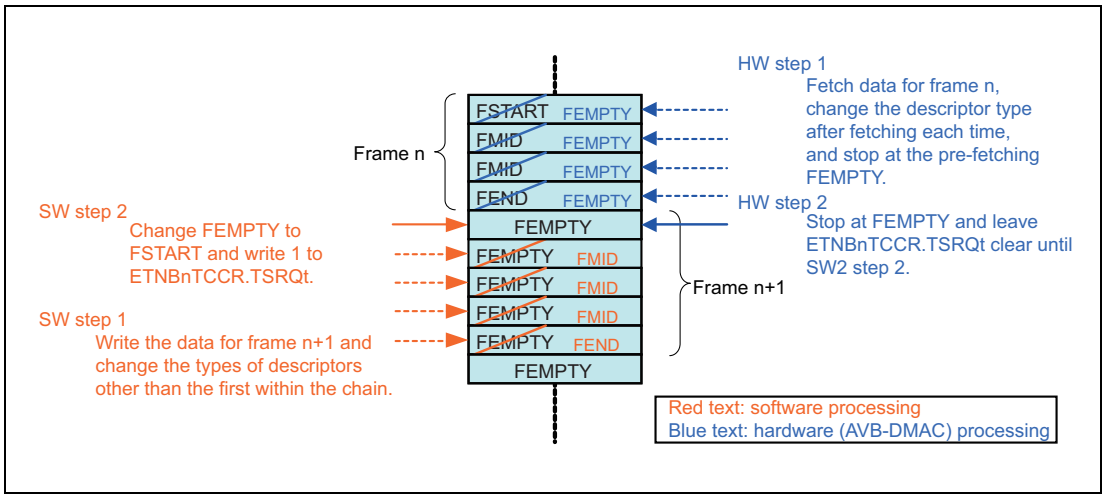


Figure 25.39 Software and AVB-DMAC Operations for Immediate Frame Transmission

(b) Frame Set Transmission with Changing of the Active Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. EOS descriptors are used for the stop points.

Start by creating a descriptor chain that has a FEMPTY descriptor at its stop point.

Figure 25.40 shows the software flow in this pattern.

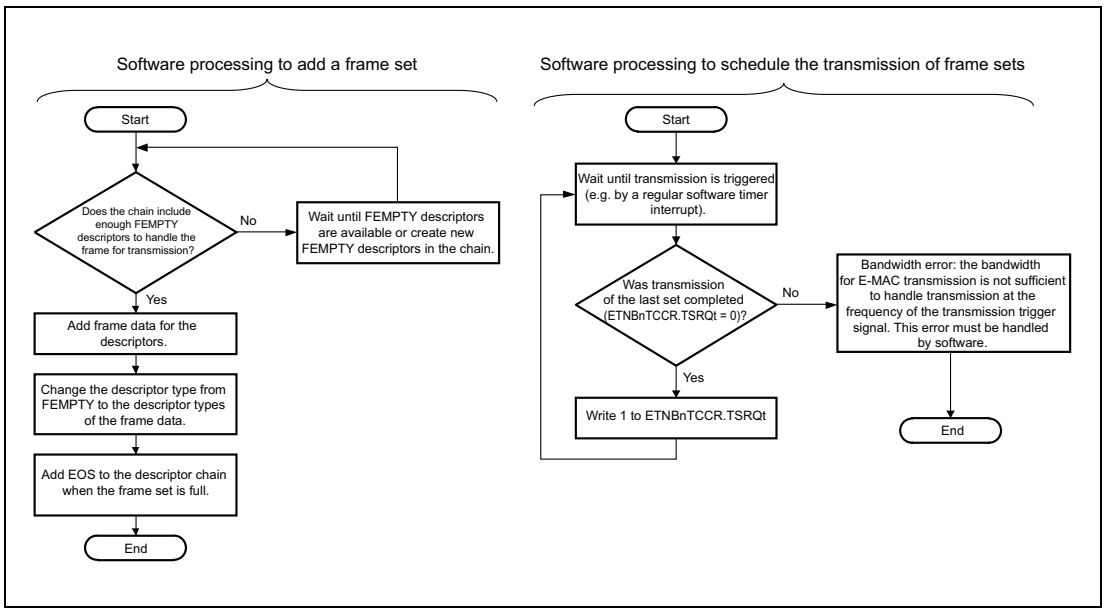


Figure 25.40 Software Flow for Frame Set Transmission with Changing of the Active Descriptor Chain

Figure 25.41 shows software and AVB-DMAC operations for frame set transmission.

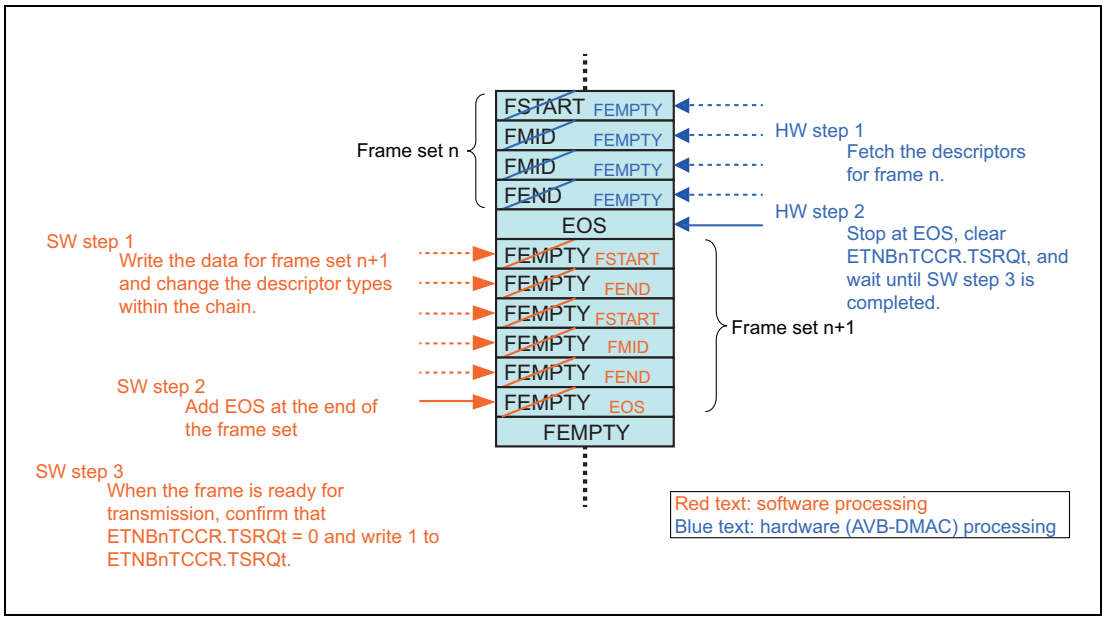


Figure 25.41 SW and AVB-DMAC Operations for Frame Set Transmission with Changing of the Active Descriptor Chain

(c) Frame Set Transmission Using a Shadow Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. Two or more descriptor chains are used. The chains are classified into the active chain and shadow chains. EOS descriptors are used for the stop points.

Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 25.42 shows the flow for software implementing this pattern.

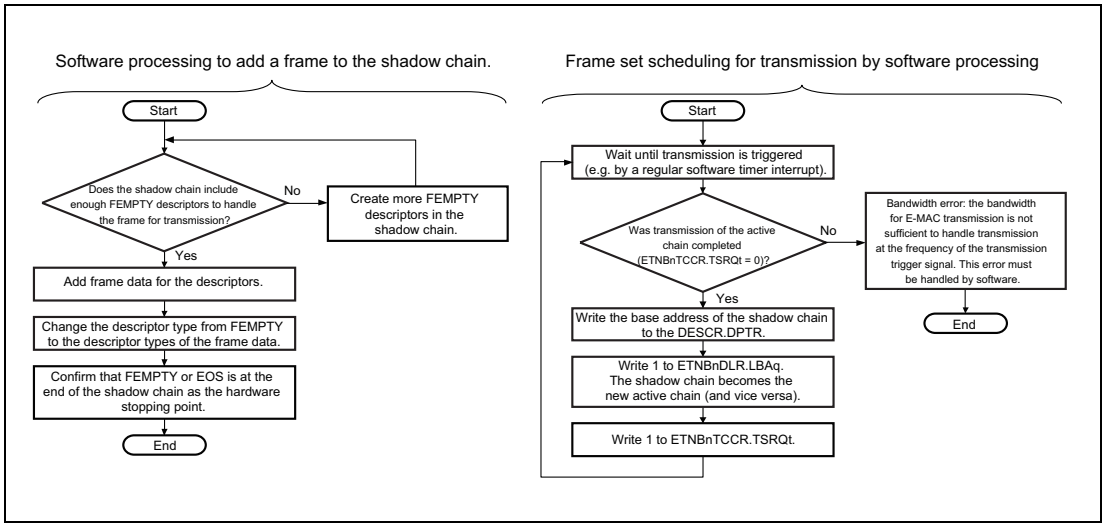


Figure 25.42 Software Flow for Frame Set Transmission Using the Shadow Descriptor Chain

Figure 25.43 shows software and AVB-DMAC operations for frame set transmission.

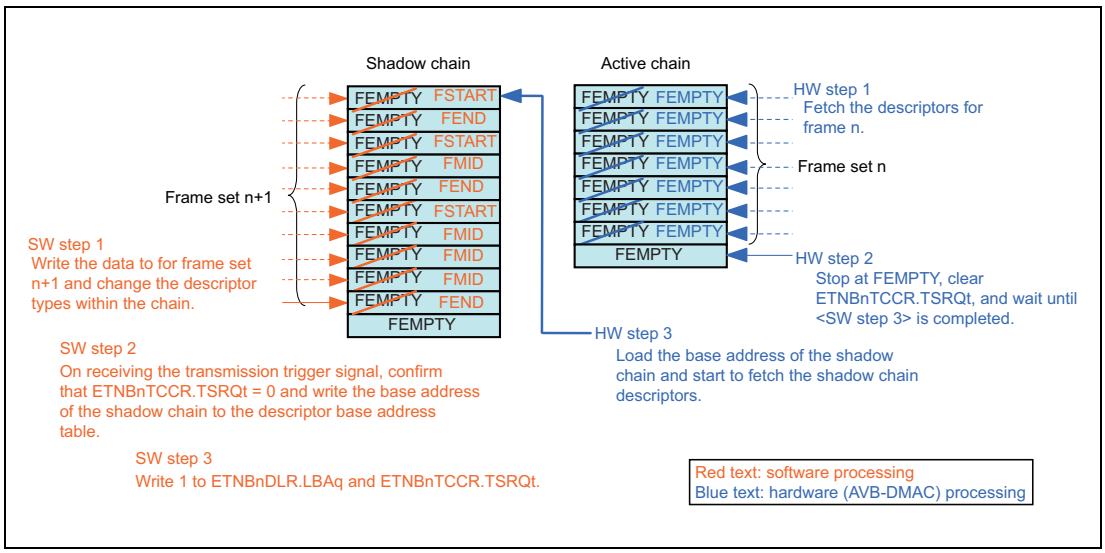


Figure 25.43 SW and AVB-DMAC Operations for Frame Set Transmission Using the Shadow Descriptor Chain

25.4.5.4 Timestamping in Transmission

Transmission timestamps are important in satisfying the requirements for time synchronization of the IEEE 802.1AS standard. This information can also be useful to other applications and in testing. The AVB-DMAC supports the storage of timestamps for transmitted frames. The timestamp values are based on the gPTP timer and are captured at the same time as sending of the Start of Frame Delimiter (SFD) for transmitted frames.

When the timestamp storage request field (DESCR.TSR) is set to 1, selecting storage of a timestamp, the tag number defined in the tag field (DESCR.TAG) of the last descriptor in a set (FEND) or of an FSINGLE descriptor for the frame being transmitted is stored with the timestamp. To make identification and association easy, the values of timestamp are stored together with tag numbers. The timestamp FIFO is accessible at any time.

Figure 25.44 shows the mechanism supporting transmission timestamping.

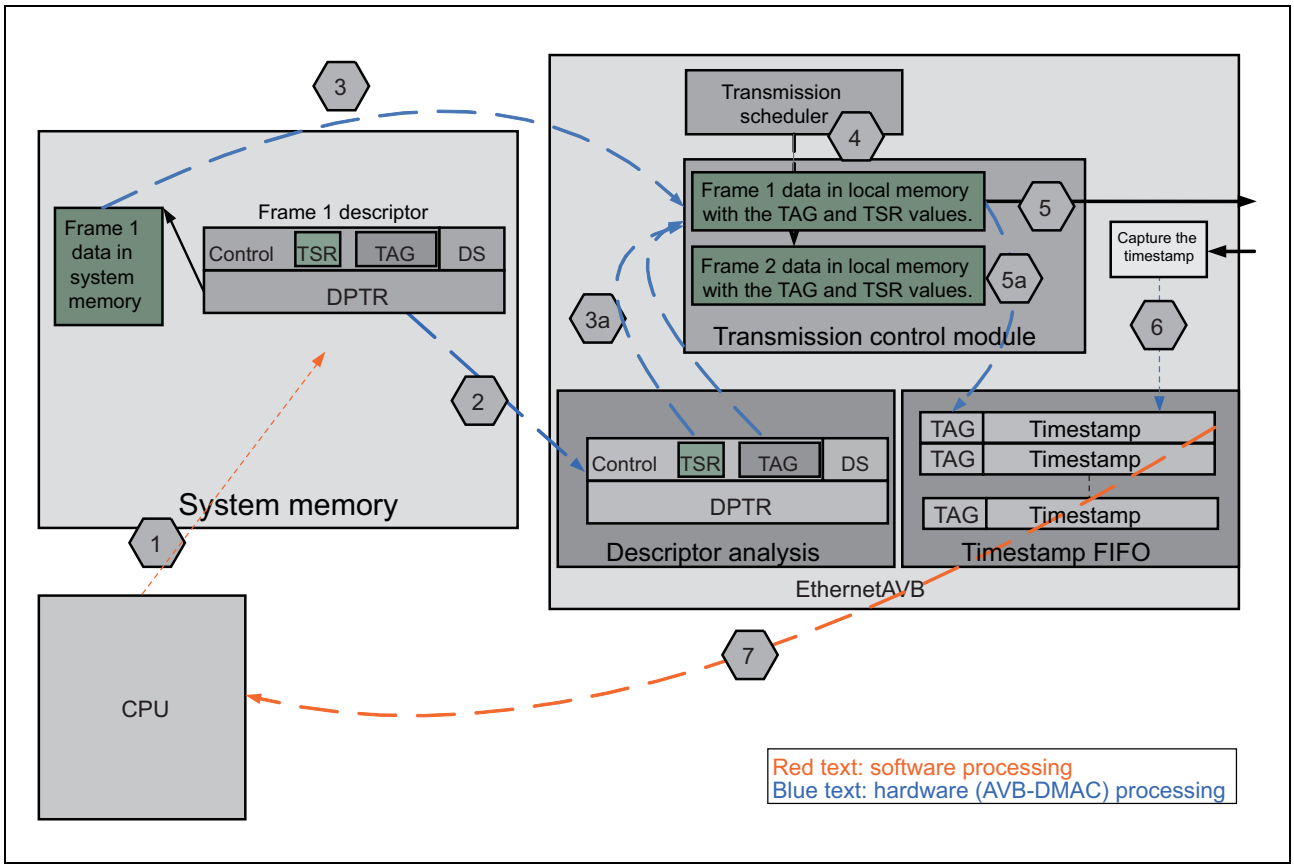


Figure 25.44 Mechanism to Support Transmission Timestamps

The method of using this function is described below:

1. Secure space in the URAM for the frame requiring timestamping.
   Write the tag number of the frame to the frame tag field (DESCR.TAG) and set the timestamp storage request field (DESCR.TSR) to 1.
2. The AVB-DMAC fetches and analyzes the descriptor. The timestamp storage request field (DESCR.TSR) is 1, so it recognizes that transmitting this frame also requires storage of the timestamp.
3. The AVB-DMAC fetches the data for frame 1 and temporarily stores the frame in internal memory for scheduling.

   3a: The frame tag field (DESCR.TAG) and timestamp storage request field (DESCR.TSR) are stored with the fetched data.)
4. Under the control of priority settings according to credit-based shaping (CBS) or another scheme, the transmission scheduler determines the time to transmit frame 1.
5. Transmission of frame 1 starts.

   5a: The information relating to frame 1 is stored in the timestamp FIFO.
6. The gPTP timestamp is captured at the start of sending the frame delimiter (SFD) for transmission and stored with the tag in the timestamp FIFO. On completion of the transmission, an interrupt is generated. For this to happen, the descriptor interrupt control register (ETNBnDIC) must be set beforehand.
7. The entry can now be read from the timestamp FIFO.

Use the timestamp FIFO for the time synchronization of frames with IEEE 802.1AS compliance.

Timestamping can also be used with other frames, but take care not to allow the timestamp FIFO to overflow. When the FIFO is full, further timestamps supplied to it are lost.

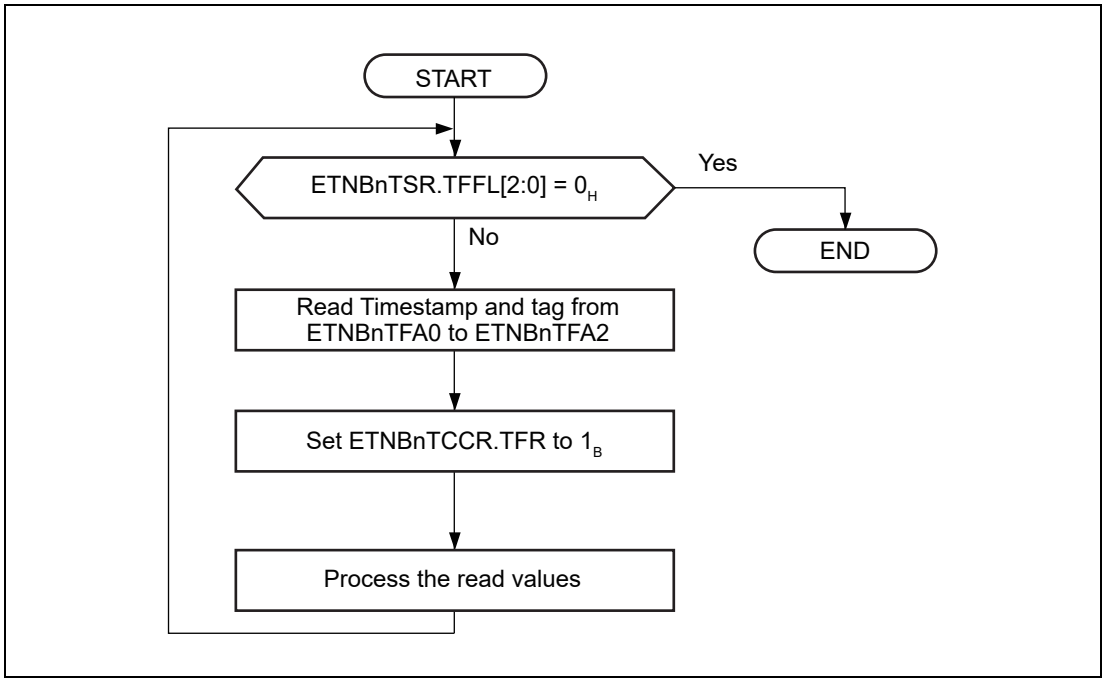


Figure 25.45 Flow of Transmission Timestamping

(1) Ending Transmission

Figure 25.46 shows the procedure for ending transmission.

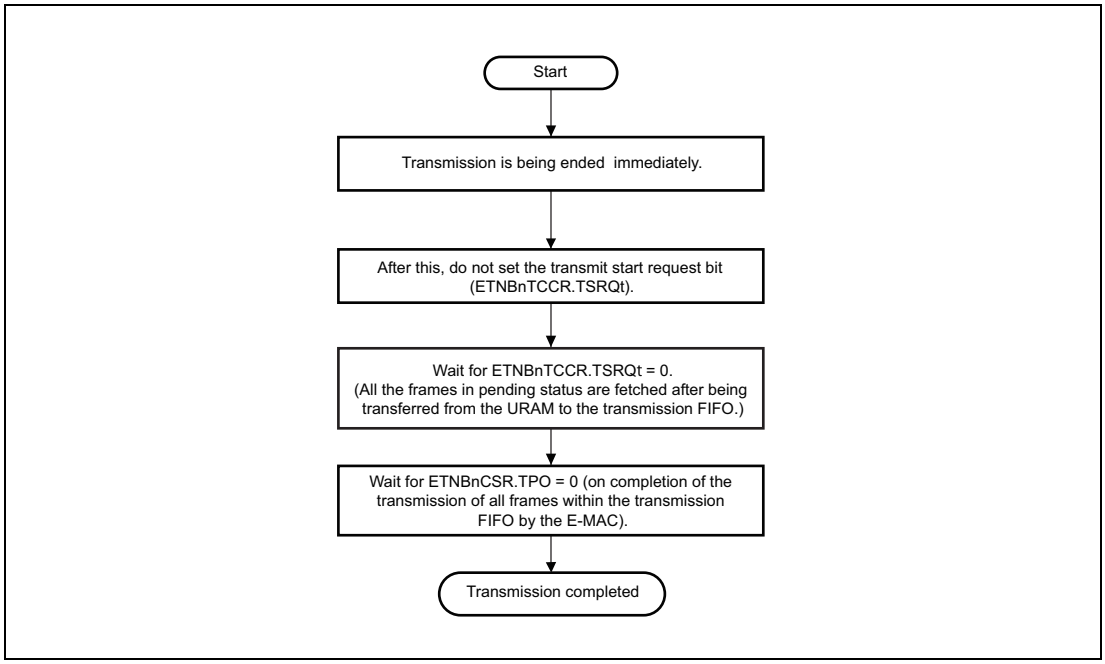


Figure 25.46 Procedures for Ending Transmission

25.4.6 CBS (Credit-Based Shaping)

In AVB transmission mode (i.e. when the transmit queue priority field in the transmit configuration register (ETNBnTGC.TQP) is 01_B or 11_B), transmission queues Q3 and Q2 are respectively assigned to Class A and Class B stream traffic and the CBS (Credit Based Shaping) algorithm is used to select the transmission queues in order to satisfy the Forwarding and Queuing for Time Sensitive Streams (FQTSS) specification (see section 8.6.8 or section 34 in IEEE 802.1Q).

The CBS algorithm is based on the concept of transmission credit for each queue. Credit can be thought of as the degree to which a queue has the “right” to transmit at a given time. Actually, in AVB transmission mode as specified in IEEE 802.1Q, queues that are subject to the CBS algorithm are able to transmit when the following conditions are met.

- At least one frame is stored in the queue.
- The credit for the queue is 0 or a positive value.

The credit for a transmission queue is incremented while one or more frames from the queue are present in the transmission FIFO but transmission of these frames is not proceeding. This state is indicated by the transmission process status bit for queue t in the AVB-DMAC status register (ETNBnCSR.TPOt) being clear (0). The credit is decremented while transmission of a frame from the queue is in progress. This mechanism is used to control transmission so that the transmission of frames from the queues for each of the traffic classes does not exceed the specified maximum bandwidth.

IEEE 802.1Q defines the following parameters for queues under the control of the CBS algorithm.

portTransmitRate: Maximum transmission data rate of an external port. The E-MAC determines this parameter.

bandwidthFraction: Maximum fraction of portTransmitRate that can be used for a queue.

idleSlope: Rate of change of credit for a queue when transmission of frames from the queue is not proceeding so the credit value (in bits per second) is increasing. idleSlope is also equal to the maximum fraction of the total bandwidth (portTransmitRate) that is available to the given queue under a specified condition (frames from the queue can be placed in a continuous stream. See Annex L of IEEE 802.Q.
idleSlope = bandwidthFraction × portTransmitRate

sendSlope: Rate of change of credit for a queue while transmission of a frame from the queue is in progress so the credit value is decreasing. (in bits per second).
The value of sendSlope is defined as follows:
sendSlope = idleSlope - portTransmitRate

Furthermore, the values below are used to define individual traffic classes (or queues for the classes) under control of the algorithm. See Annex L of IEEE 802.Q.

maxFrameSize: Maximum size of frames (in bits) of the corresponding traffic class that can be transmitted from a port

maxInterferenceSize: Maximum burst size (in bits) by which delays for the corresponding traffic class can be allowed

hiCredit: Maximum credit value (positive number). Can be calculated by using the following equation: hiCredit = maxInterferenceSize × (idleSlope / portTransmitRate)

loCredit: Minimum credit value (negative number). Can be calculated by using the following equation: loCredit = maxFrameSize × (sendSlope / portTransmitRate)

Figure 25.47 shows how the CBS algorithm works and the meaning of the above parameters.

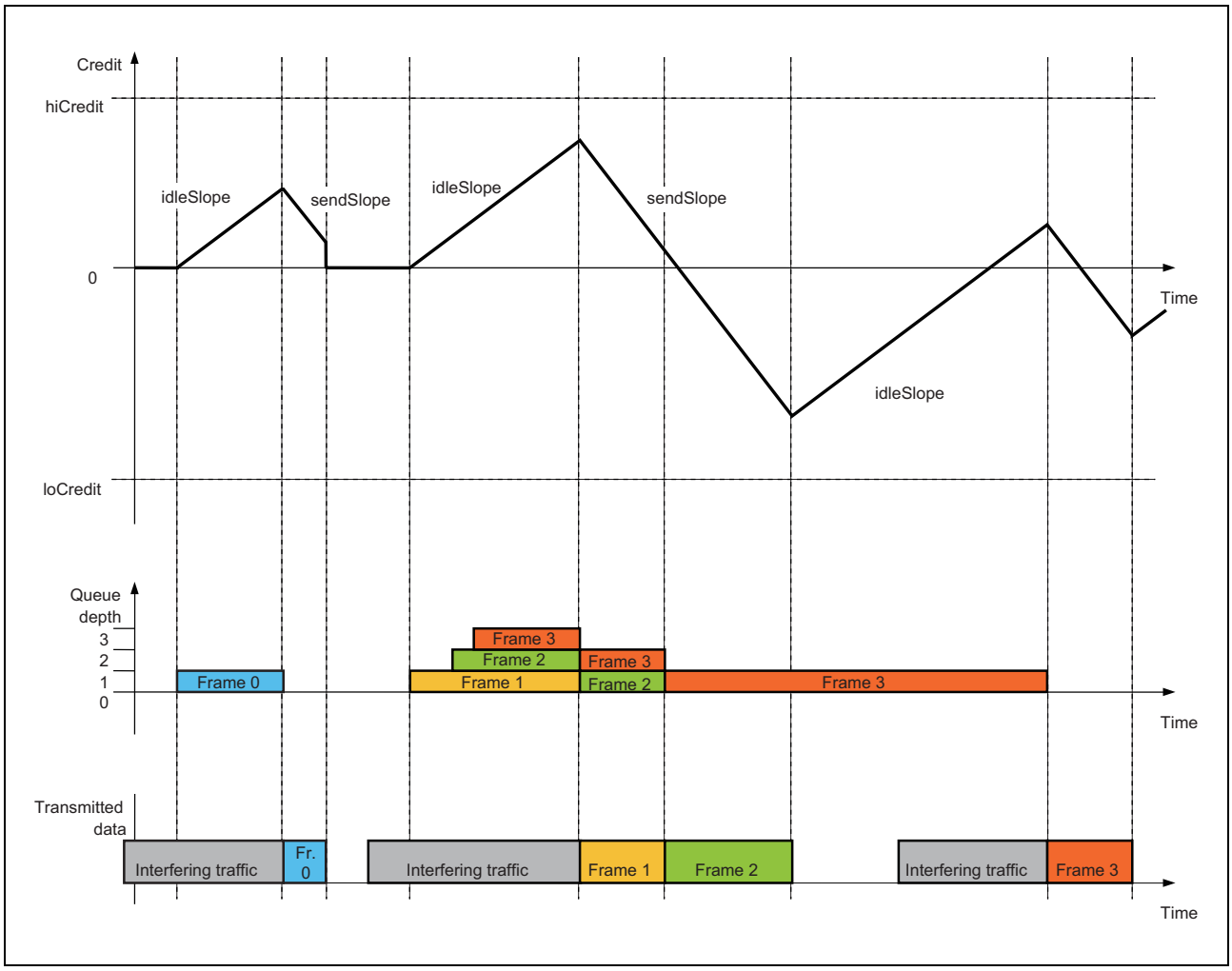


Figure 25.47 CBS (Credit-Based Shaping) Operation

Figure 25.48 shows the implementation of CBS in the AVB-DMAC.

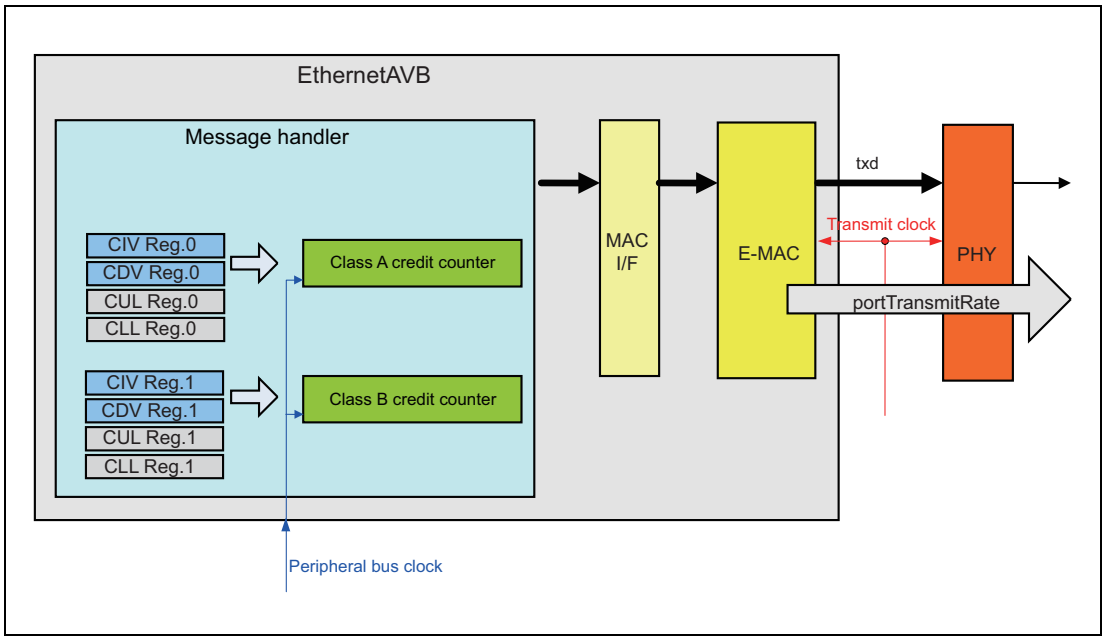


Figure 25.48 CBS (Credit-Based Shaping) Operation in the AVB-DMAC

The above implementation is based on “credit counters” for the respective traffic classes (SR Class A and Class B). The following parameters apply for these classes.

CBS increment value (CIV): Signed positive number
The credit is incremented by this amount every peripheral bus clock cycle while a frame from the queue is pending but transmission has not started (idleSlope).

CBS decrement value (CDV): Signed negative number
The credit is decremented by this amount every peripheral bus clock cycle while transmission of a frame from the queue is proceeding (sendSlope).

The CBS increment value (CIV) and CBS decrement value (CDV) are defined as follows.

CIV = idleSlope × Mfactor

CDV = sendSlope × Mfactor

Mfactor is a multiplier factor to ensure accuracy for CIV and CDV. CIV and CDV are calculated by using the following equations.

CIV = (portTransmitRate/CHI_freq) × bwFraction × Mfactor

CDV = (portTransmitRate/CHI_freq) × (bwFraction - 1) × Mfactor

CHI_freq is the frequency of the peripheral bus clock. The credit counters are driven by the peripheral bus clock, so calculating the slope parameters for CBS requires (1/CHI_freq).

Use software to prepare Mfactor for the CBS parameters. All queues for the same Class must have the same CBS parameters. Mfactor for a specified Class c can be changed during operation, unless transmission is pending for that Class (i.e. the transmit process status bit in the AVB-DMAC status register (ETNBnCSR.TPOt) = 0). At that time, the credit counter values for Class A and Class B are 0.

Note that the credit value will not match a new incrementation or decrementation parameter if Mfactor is changed while the credit counter value is non-zero. Mfactor is not present in the AVB-DMAC registers.

Set the CIV and CDV parameters in the CBS increment value registers c (ETNBnCIVRc) and the CBS decrement value registers c (ETNBnCDVRc). These are treated as dynamic settings since they should be updated when streams are registered or erased in accord with IEEE 802.1Qat.

The AVB-DMAC also has CBS upper limit registers c (ETNBnCULc) (the upper limit registers for Classes A and B) and CBS lower limit registers c (ETNBnCLLc) (the lower limit registers for Classes A and B). Set Mfactor to match the credit value and set the upper limit (hiCredit) and the lower limit (loCredit) for each class as defined above.

CUL = hiCredit × Mfactor = maxInterferenceSize × bwFraction × Mfactor

CLL = loCredit × Mfactor = maxFrameSize × (bwFraction - 1) × Mfactor

Example:

Assume that portTransmitRate = 100 Mbps, CHI_freq = 100 MHz and bwFraction = 3%.
Then idleSlope and sendSlope represented as one bit vs. cycles of the peripheral bus clock are as follows.

idleSlope = (portTransmitRate/CHI_freq) × bwFraction = 100/100 (Mbps/MHz) × 3% = 0.030 of a bit per high-speed peripheral bus clock cycle

sendSlope = idleSlope - (portTransmitRate/CHI_freq) = -0.97 bits per peripheral bus clock cycle

Let Mfactor be 100, then CIV and CDV parameters are determined as follows.

CIV = idleSlope × Mfactor = 3.0

CDV = sendSlope × Mfactor = -97

25.4.6.1 Restrictions on CIV, CDV and Mfactor

The maximum value (the minimum value for negative numbers) up to which the credit counter will not overflow determines the maximum values of CIV and CDV that can be set in the CBS registers. This maximum credit value is equivalent to the worst case of the hiCredit value, and the maximum values for Class A and Class B are calculated as follows.

<Conditions>

- Class A maximum value (hiCredit_max_classA)
  classA bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum frame size.
  hiCredit_max_classA ≅ maxInterferenceSize for Class A = Interference due to one max. sized frame = header + max. size payload + CRC (2000 bytes) + preamble (8 bytes) + IFG (12 bytes) + processing_delay (≅ 80 bytes) ≅ 2100 bytes
- Class B maximum value (hiCredit_max_classB)
  classB bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum size of frames in the Class A transmission queue and other transmission queues.
  hiCredit_max_classB ≅ maxInterferenceSize for Class B = Interference due to two max-size frames = 2× hiCredit_max_classA ≅ 4200 bytes

hiCredit_max_classA = 16800

hiCredit_max_classB = 33600

The maximum values that can be selected with Mfactor for the 32-bit signed counter without overflow are:

Mfactor_max_classA = 2^{31}-1 / hiCredit_max_classA ≅ 127826 and
Mfactor_max_classB = 2^{31}-1 / hiCredit_max_classB ≅ 63913.

A high degree of accuracy can be achieved even with a low bandwidth. In Class B, bandwidthFraction = 0.05% and the bandwidth error < 0.1%.

The maximum value of CIV is calculated from the following equation.

CIV = idleSlope x Mfactor = (portTransmitRate / CHI_freq) × bandwidthFraction × Mfactor

When Mfactor is the maximum value and bandwidthFraction is the maximum value (up to 100%):

CIV_max_classA = (portTransmitRate / CHI_freq) × Mfactor_max_classA and
CIV_max_classB = (portTransmitRate / CHI_freq) × Mfactor_max_classB.

Table 25.121 shows examples of values for portTransmitRate and peripheral clock frequency. Note that the values in the table are the limits of CIV up to which the 32-bit credit counter will not overflow. The CIV parameters are implemented as 16 bits + a sign bit, so a further limit of CIV ≤ 65535 applies to both Class A and Class B.

Table 25.121 Example of Maximum Values for Class A and Class B CIV Parameters

portTransmitRate	clk_chi[MHz]	CIV_max_classA	CIV_max_classB
100 Mbps	25	511304	255652
100 Mbps	100	127826	63913

25.4.6.2 Credit Incrementation during Inter-Frame Gaps (IFGs)

The inter-frame gap (IFG) after a frame is transmitted is not treated as part of frame transmission by the CSB credit counter. During an IFG, the credit is incremented for all SR queues that have pending frames or negative credit. **Figure 25.49** illustrates credit operations during IFGs.

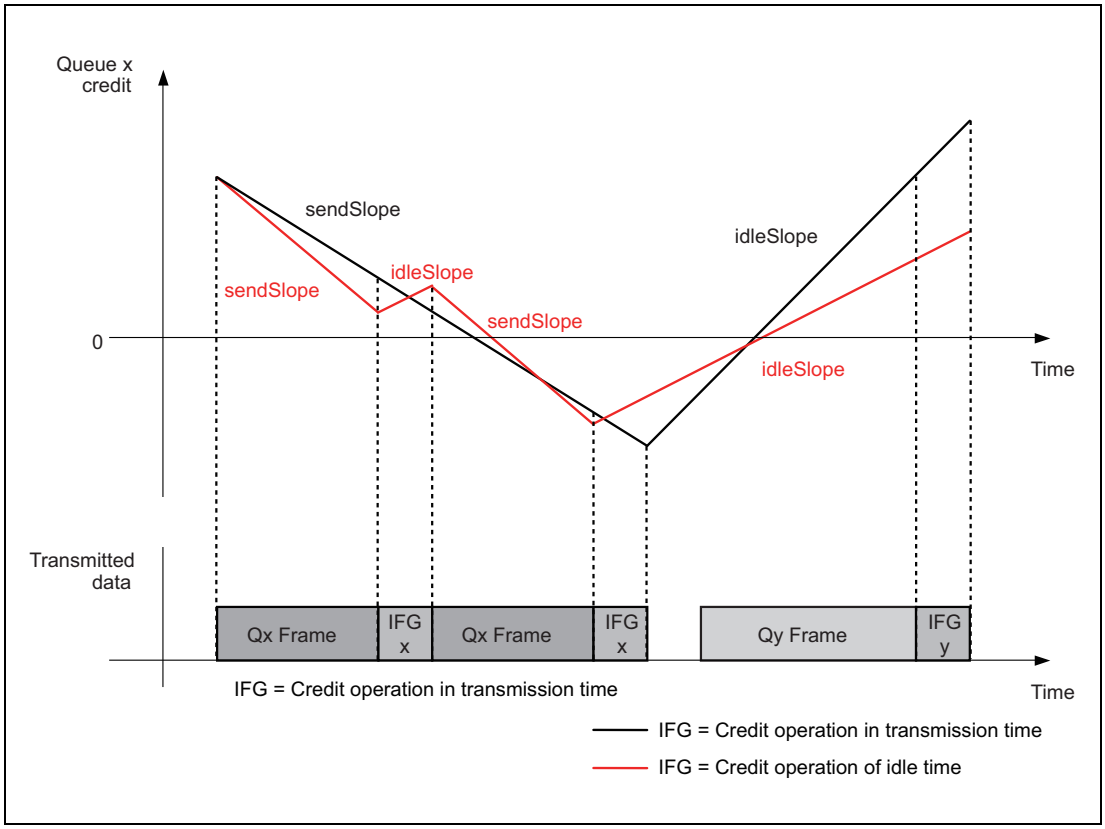


Figure 25.49 Credit Operations during IFGs

Accordingly, the IFG need not be included in calculation of the bandwidth requirements for the specified SR Class when deciding the idleSlope, sendSlope, and CIV and CDV parameters. However, IFG must also be included in the calculation in order to confirm that the total bandwidth allocated to all SR Classes does not exceed 100% of portTransmitRate. This is described in section 35.2.2.8.4 of IEEE 802.1Q.

25.4.6.3 Example

The case of a Class A 48-kHz stereo audio stream among Ethernet frames is described as an example.

After every Class A measurement interval (125 us), 80 octets consisting of two sets of six 32-bit samples plus a 32-octet header are stored as audio data within a frame. The IEEE 802.3 also imposes a 42-octet media-specific framing overhead (an 8-octet preamble, 14-octet IEEE 802.3 header, 4-octet IEEE 802.1Q priority/VID Tag, 4-octet CRC, and 12-octet IFG) are also added. Accordingly, the total frame size is 80 + 42 = 122, and one such frame is transmitted after every class measurement interval.

This represents a total bandwidth of about 7.8 Mbits per second (122 octets × 8 bits per octet × 8000 frames per second) for this class. If the E-MAC is assumed to run at 100 Mbps (portTransmitRate), this is equivalent to the allocation of about 7.8% of the total bandwidth to each Class A queue. If other traffic classes are to share the total transmission bandwidth, checking that this 7.8% allocation does not lead to the total allocation of bandwidth being greater than 100% of portTransmitRate is required.

To obtain the CIV and CDV parameters for a given class, the IFG must not be taken into account in calculation of the frame size must not include the IFG. For this case, therefore, we obtain an 80-bit payload + 30-bit overhead = 110-octet measurement interval for the class ≤ the total bandwidth for the class = 7.04 Mbps = 7.04% of portTransmitRate.

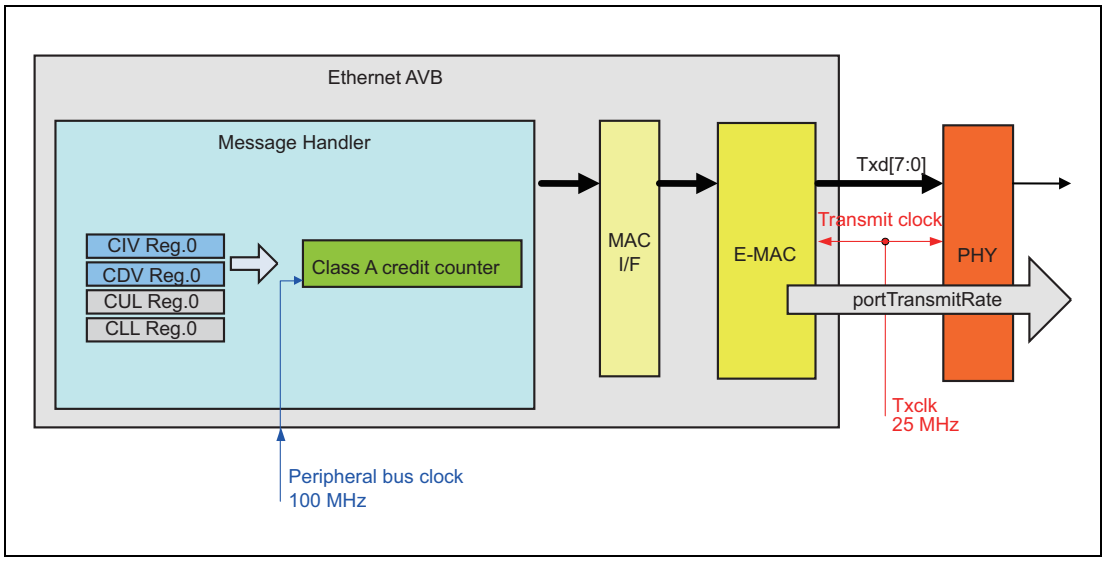


Figure 25.50 Example of CBS Settings

Given that:

- the E-MAC runs at 100 Mbps, so portTransmitRate = 100 Mbps and
- Peripheral bus clock (operating clock for the credit counter) frequency = 100 MHz,

securing a bandwidth of 7.04 Mbits/sec for Class A requires configuring the CBS parameters as follows.

- bandwidthFraction = 7.04%
- idleSlope = (portTransmitRate/CHI_freq) × bandwidthFraction = 0.0704 bits per high-speed peripheral bus clock cycle
- sendSlope = idleSlope – (portTransmitRate/CHI_freq) = –0.9296 bits per high-speed peripheral bus clock cycle

When Mfactor = 100, the parameters are as follows.

- CIV = idleSlope × Mfactor = 7.04 bits per high-speed peripheral bus clock cycle
- CDV = sendSlope × Mfactor = -92.96 bits per high-speed peripheral bus clock cycle

These are the final values for setting in the ETNBnCIVR1 and ETNBnCDVR1 registers.

25.4.7 IEEE802.1: gPTP

25.4.7.1 gPTP Timer

An 84-bit timer is provided to support the gPTP function. **Figure 25.51** shows the definitions of bits for the timer and in related registers. gPTP timer will start after ETNBnCCC.CSEL configure and transit to operation mode.

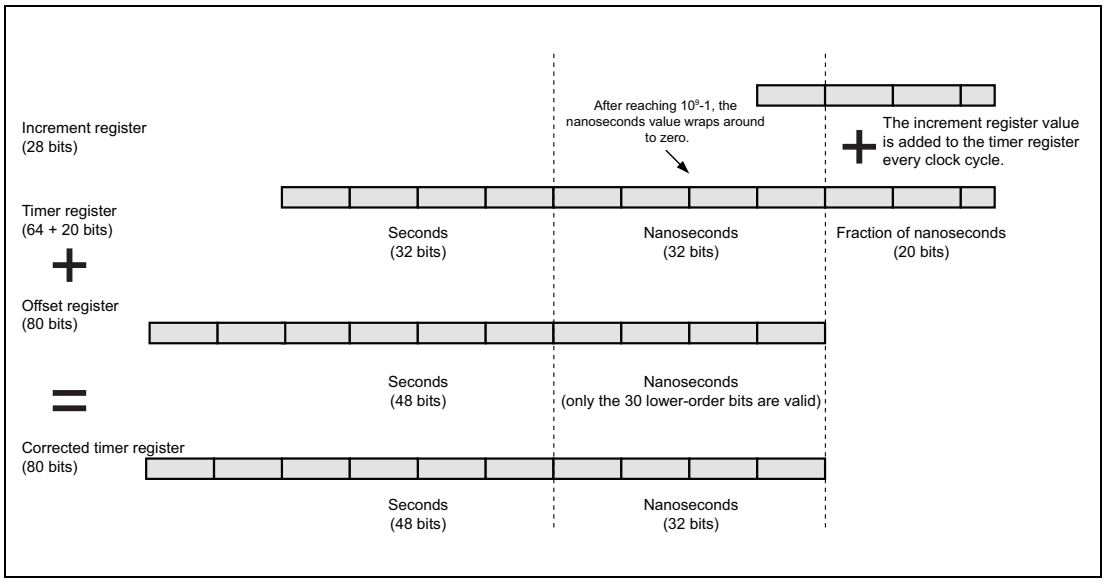


Figure 25.51 Definitions of gPTP Timer Bits and Related Bits

The higher-order 32 bits indicate seconds. For the next 32 bits, counting by one corresponds to the passage of 1 ns. The lower-order 20 bits are a fractional value (less than 1 ns). Software can only read the higher-order 32 bits, indicating seconds, and the subsequent 32-bits, indicating nanoseconds. The lower-order 20 bits, representing less than 1 ns, are not readable. They are only used within the AVB-DMAC to maintain accuracy in time measurement.

The timer can be reset by setting the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 01_B. These bits are set to 00_B on completion of normal resetting of the timer.

After the timer starts, the value in the gPTP timer increment register (ETNBnGTI.TIV) is added to the value of the gPTP timer every clock cycle.

After setting a value in the gPTP timer increment register (ETNBnGTI.TIV), set the timer increment value setting request bit in the gPTP configuration control register (ETNBnGCCR.LTI). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed.

An offset to the gPTP timer is also available. If this is required, set the value in the gPTP timer offset register (ETNBnGTOi.TOV). After setting a value in this register, set the timer offset value setting request bit in the gPTP configuration control register (ETNBnGCCR.LTO). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed. When adding an offset, take care that it does not exceed 80 bits.

The value of the gPTP timer can be read from the gPTP timer capture register (ETNBnGCTi.CTV). Set the timer capture source select bits in the gPTP configuration control register (ETNBnTCCR.TCSS) to select the timer value for capture as the value of the gPTP timer, the corrected value of the gPTP timer (value with the offset added), or the AVTP presentation time. Setting the timer control request bits in

the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 11_B initiates the capture. Once normal capture of the timer is complete, the value of the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) returns to 00_B.

The timer for gPTP operates as a free-running timer but can be synchronized with the Grandmaster clock.

25.4.7.2 Free-Running Operation

The IEEE 802.1 AS standard for timing and synchronization does not prescribe the physical adjustment of local clocks to the Grandmaster clock. To avoid negative effects from the correction procedure, we recommend the use of a free-running timer.

As a free-running timer, the timer counts the local time in seconds or nanoseconds. The gPTP timer increment register (ETNBnGTI.TIV) is set to 1 ns (the setting value = 0010 0000_H) and the gPTP timer offset register (ETNBnGTOi.TOV) is set to 0. The ratio information captured at the time of the gPTP delay measurement and synchronization procedures is used to correct the frequency ratio to that of the Grandmaster clock. The Grandmaster clock can be calculated from the local clock by using the information collected during the gPTP measurement and synchronization procedures.

25.4.7.3 Synchronization with the Grandmaster Clock

In situations requiring physical synchronization of the local clock with the Grandmaster clock, the fractional nanoseconds value (the 20 lower-order bits of the gPTP timer) is used to make the adjustment. Specifically, the increment value is finely adjusted to correct for deviations of the clock frequency from that of the Grandmaster clock.

Use the timer offset value (in the gPTP timer offset registers, ETNBnGTOi.TOV) to correct for offsets from the theoretical value (at start-up, etc.). The sum of the timer value and the offset register is the “corrected timer” value.

Note that only the nanoseconds portion of the gPTP timer offset registers (ETNBnGTOi.TOV[31:0]) is valid.

The following equation gives a method of calculating the increment (ETNBnGTI.TIV) from the frequency of the gPTP clock and its deviation from that of the Grandmaster clock. Variable d is the deviation (d = 10^{-6} for 1 ppm).

$$ETNBnGTI.TIV = round\left(\frac{2^{20}GHz}{f_{GPTP}} \times (1+d)\right)$$

After adjusting for the current deviation of clock frequency, re-set the gPTP timer increment register (ETNBnGTI.TIV).

After calculating the new offset value, re-set the gPTP timer offset register (ETNBnGTOi.TOV).

25.4.7.4 Continuous AVTP capture

By setting ETNBnGCCR.TCR to 10_B continuous AVTP timer value capture is activated. In this mode CPU has always access to current AVTP value by reading ETNBnGCTi.CTV[31:0].

The following flow should be used to enter continuous AVTP capture mode.

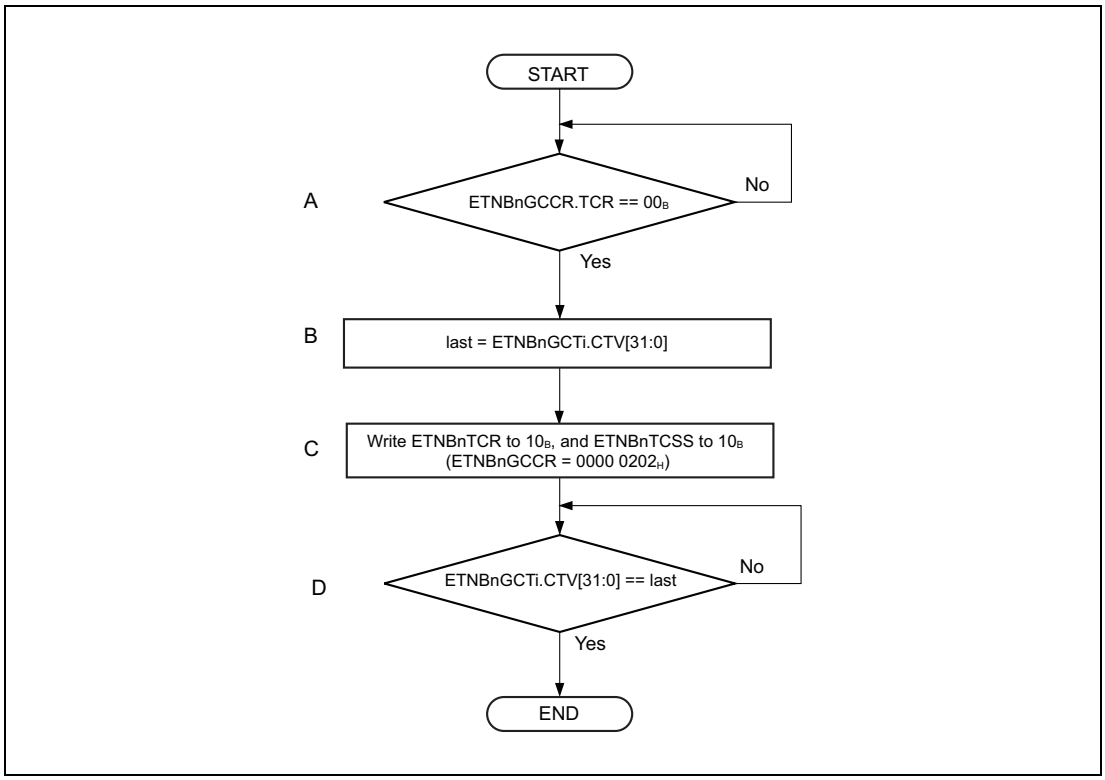


Figure 25.52 SW flow entering continuous AVTP capture mode

The following flow should be used to leave continuous AVTP capture mode.

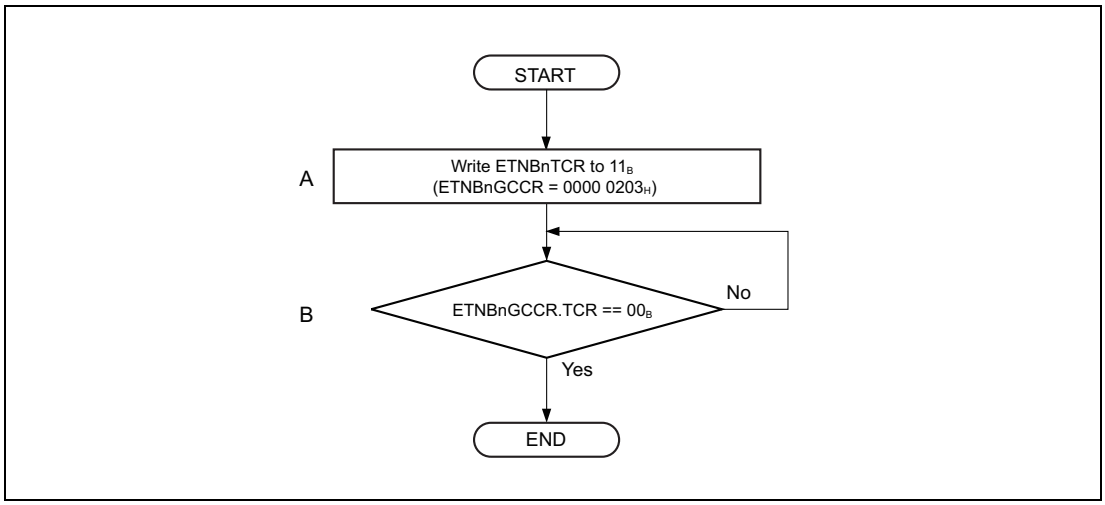


Figure 25.53 SW flow leaving continuous AVTP capture mode

25.4.7.5 Support Provided by the gPTP Timer in Transmission and Reception

The timer value described above is used in the timestamp values captured when start frame delimiters are detected in reception and generated in transmission.

Captured timestamp values for received frames are stored in the corresponding descriptors. Those for transmitted frames are stored with tag information in the timestamp FIFO. The timestamp values are thus correlated with both transmitted and received frames.

Note that the use of corrected timer values can introduce an error due to the offset correction in the gPTP synchronization procedure.

Errors due to the SDF notification and the asynchronous interface between the timer modules must also be taken into account.

25.4.8 Support for IEEE 1722

For IEEE 1722, the following two functions are supported.

- Output and capture of values in the IEEE 1722 AVTP (Audio/Video Transport Protocol) presentation time format
- Comparison of IEEE 1722 AVTP presentation timestamps

The 32-bit AVTP timestamp field of IEEE 1722 frames holds the AVTP presentation time when the AVTP timestamp enable bit in the frame is 1. The AVTP timestamp field is generated from the gPTP timer and is given as seconds (gPTP_seconds) and nanoseconds (gPTP_nanoseconds) according to the following equation.

AVTP timestamp = (gPTP_seconds × 10^9 + gPTP_nanoseconds) modulo 2^{32}

The AVTP presentation time can be read from the gPTP timer capture register (ETNBnGCTi.CTV). Set the timer capture source select bits in the gPTP configuration control register (ETNBnGCCR.TCCS) to select the timer value for capture as the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (ETNBnGCCR.TCR[1:0]) to 11_B initiates the capture. The value is obtained by adding the maximum transit time defined in the gPTP maximum transit time register (ETNBnGMTT.MTTV) to the corrected timer value. The AVTP presentation time wraps around approximately every four seconds.

CAUTION

The AVTP presentation time captured in ETNBnGCTi.CTV is only valid when the corrected timer value is in synchronization with the Grandmaster clock. That is, the timer increment and timer offset values for the corrected timer value must be adjusted during the synchronization procedure so that the corrected gPTP clock is physically adjusted to match the time kept by the Grandmaster clock.

25.4.8.1 AVTP compare

AVB-DMAC provides 8 independent AVTP compare units. Each unit is able to handle single shot comparison as well as periodic comparison. ETNBnGSR.PCMi informs about the current mode of unit i. To 4 units the single shot compare value can provided via AVTP-FIFO to relax timing requirements to application SW.

When the AVTP Presentation Time value has reached or exceeded the configured comparison value a pulse of one clk_chi (peripheral bus clock) clock cycle is applied to the interrupt flag ETNBnGIS.PTMFi is set to 1.
The configured max transit time value is not taken in count when comparing.
SW should ensure that configured comparison value is well in future to take into account SW inaccuracy.
SW should ensure that comparison value is not too far in future (more than 2 seconds) with respect to current AVTP timer value to guarantee comparison functionality.

When writing to ETNBnGCCR, SW can use a 32-bit access if ETNBnGCCR.TCSS is written to 11_B and all other bits not required for comparison value programming are written to 0.

(1) Single shot compare

An AVTP compare unit used in single shot mode generates exactly one match after CPU programs the comparison value. The following flow should be used to program absolute comparison values in single shot mode.

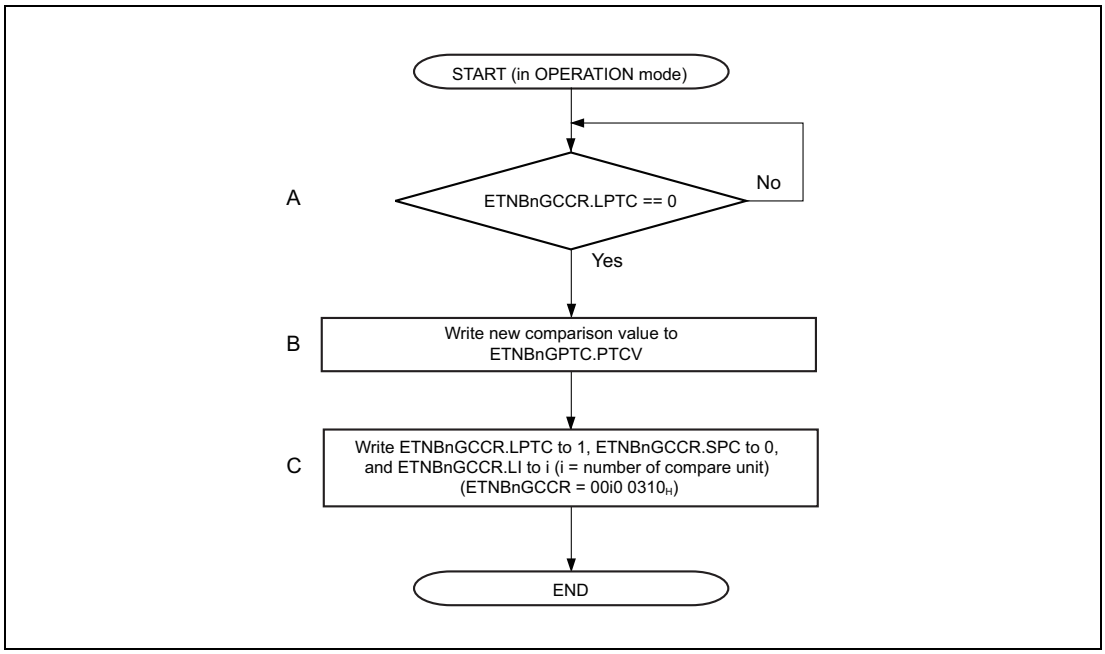


Figure 25.54 SW flow to program a single shot compare value

(2) Periodic compare

An AVTP compare unit used in periodic compare mode generates matches of configured period until CPU disables periodic mode. Optionally, it is possible to define the initial phase before repetition starts.

The following flow should be used to start periodic compare mode or to change period value.

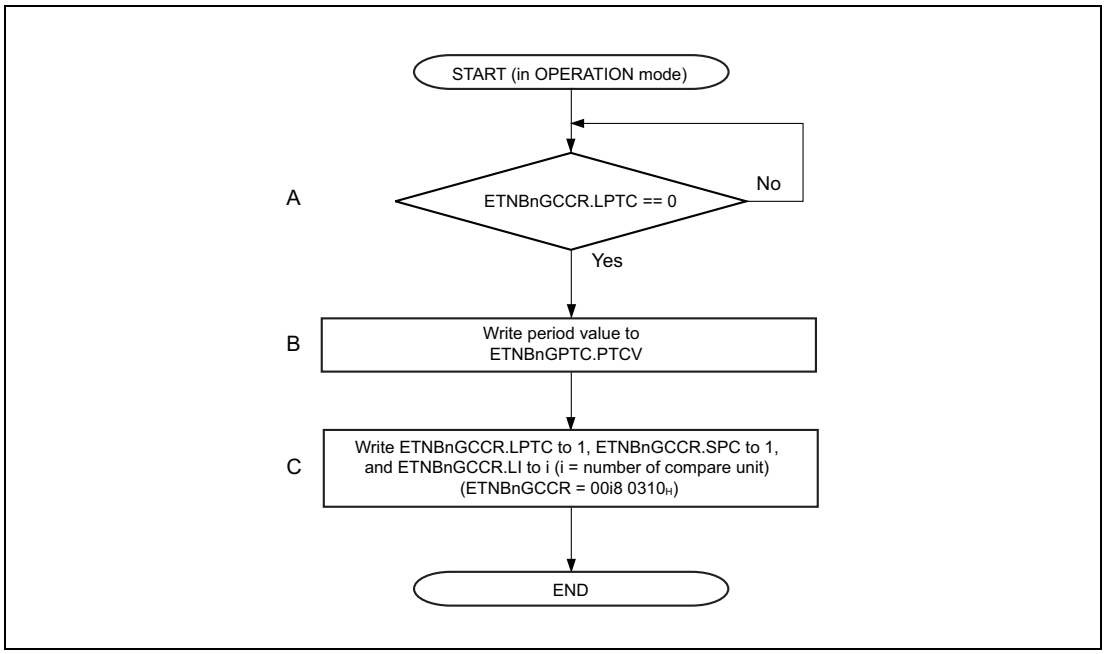


Figure 25.55 SW flow to program period value (without phase relation)

Some application may require phase relation between periodical match events generated by different units. In this case the next flow can be used to enter periodic compare mode with phase relation. A compare unit will only start a period when there is no pending comparison. If the absolute starting point of all units (box B in flow) is well in future the period of all units start at the same time (e.g. 48 kHz on unit 1 and 96 kHz on unit 2).

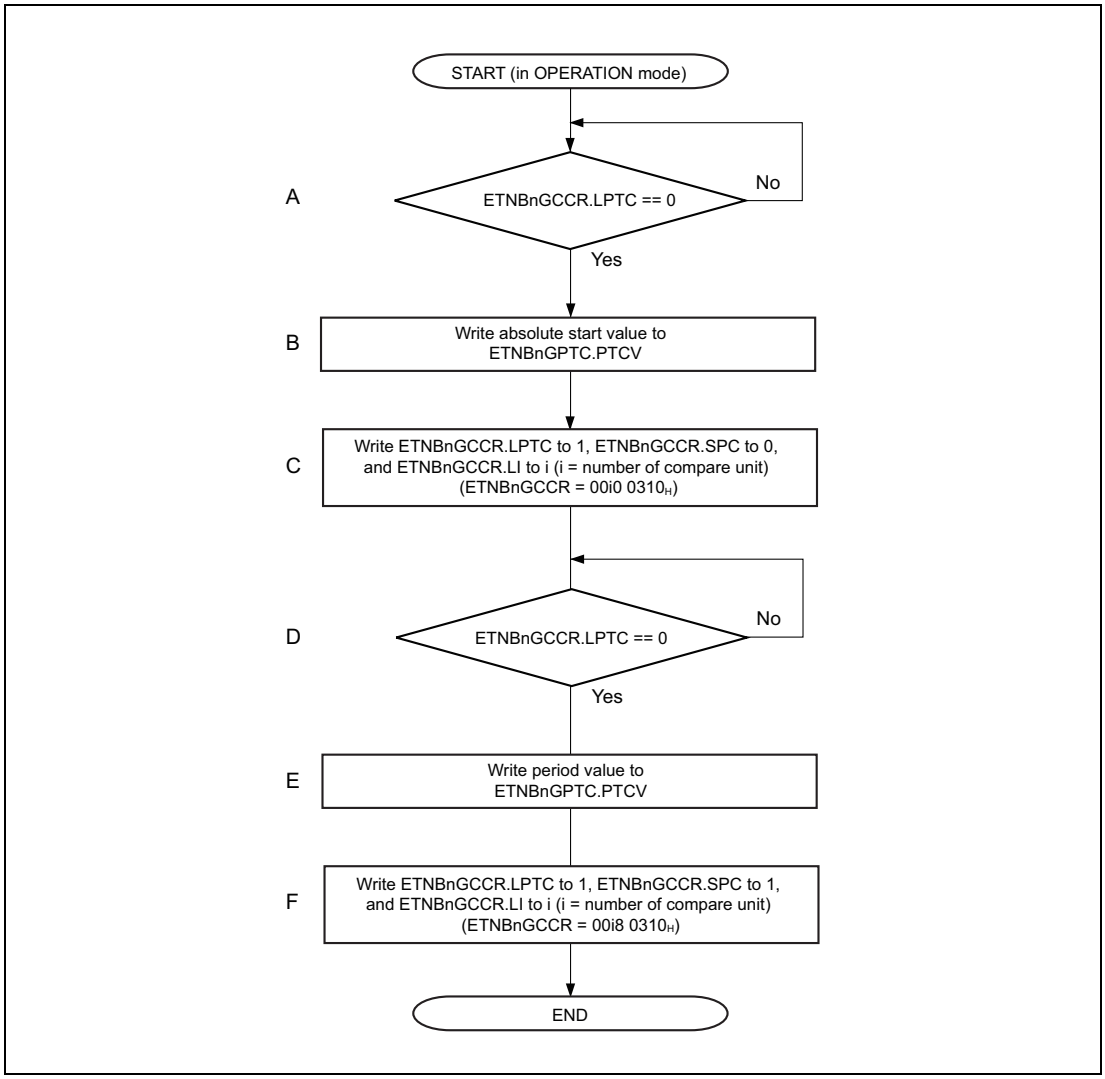


Figure 25.56 SW flow to program period value (with phase relation)

The periodic mode is left by entering single-shot mode. Do not write new comparison value in ETNBnGPTC.PTCV for changing mode as shown in flow below, this value will not be used as new comparison value.

The following flow should be used to leave periodic compare mode.

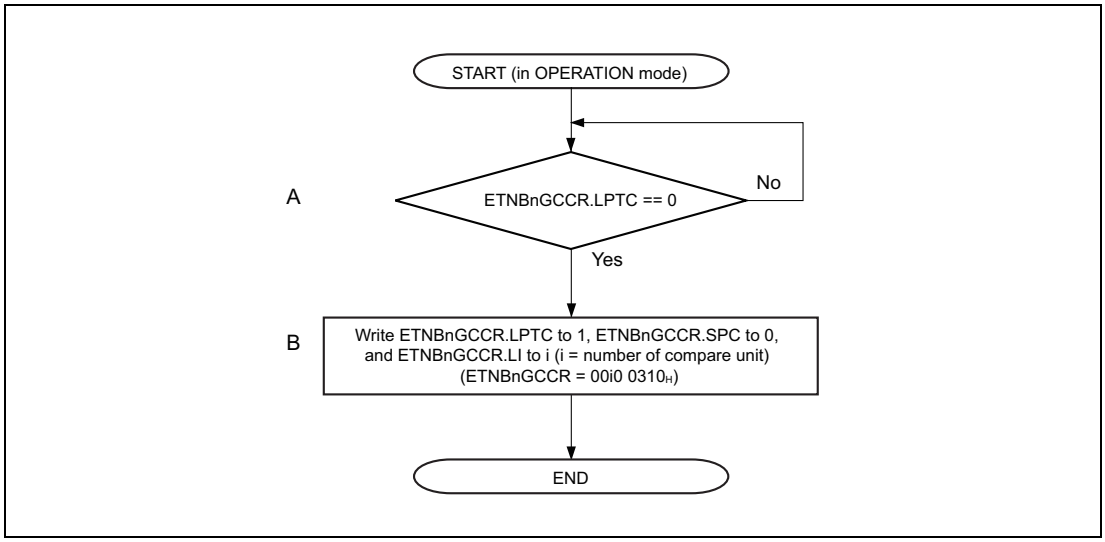


Figure 25.57 SW flow to end period mode

The last periodical comparison happens after Point B in SW flow. AVB-DMAC leaves periodic mode after last periodical comparison happens. This is flagged by ETNBnGSR.PCMi. Comparison unit can only be reused after ETNBnGCR.PCMi is 0.

(3) FIFO based single-shot compare

Some comparison units provides the possibility to use timestamps from a FIFO. To use this possibility CPU writes AVTP timestamp to ETNBnGPTFi instead of ETNBnGPTC. By writing to ETNBnGPTFi the value is added to FIFO, no load is required but CPU needs to check if AVTP-FIFO is able to accept new values.

FIFO usage may reduce CPU load because CPU can program a bunch of timestamps at same time. When there is no comparison ongoing in AVTP compare unit, the next value from AVTP-FIFO is loaded and comparison starts. If the FIFO is empty no value is loaded and no further match event is generated.
When moving from single-shot comparison to FIFO based single-shot comparison, the 1st FIFO value is immediately loaded independent of an ongoing comparison.
By reading ETNBnGSR.AFFLi CPU can observe number of pending timestamps.

The following flow should be used to add timestamps to the AVTP-FIFO of compare unit. If FIFO depth is insufficient, FIFO size should be extended in SW (box E).

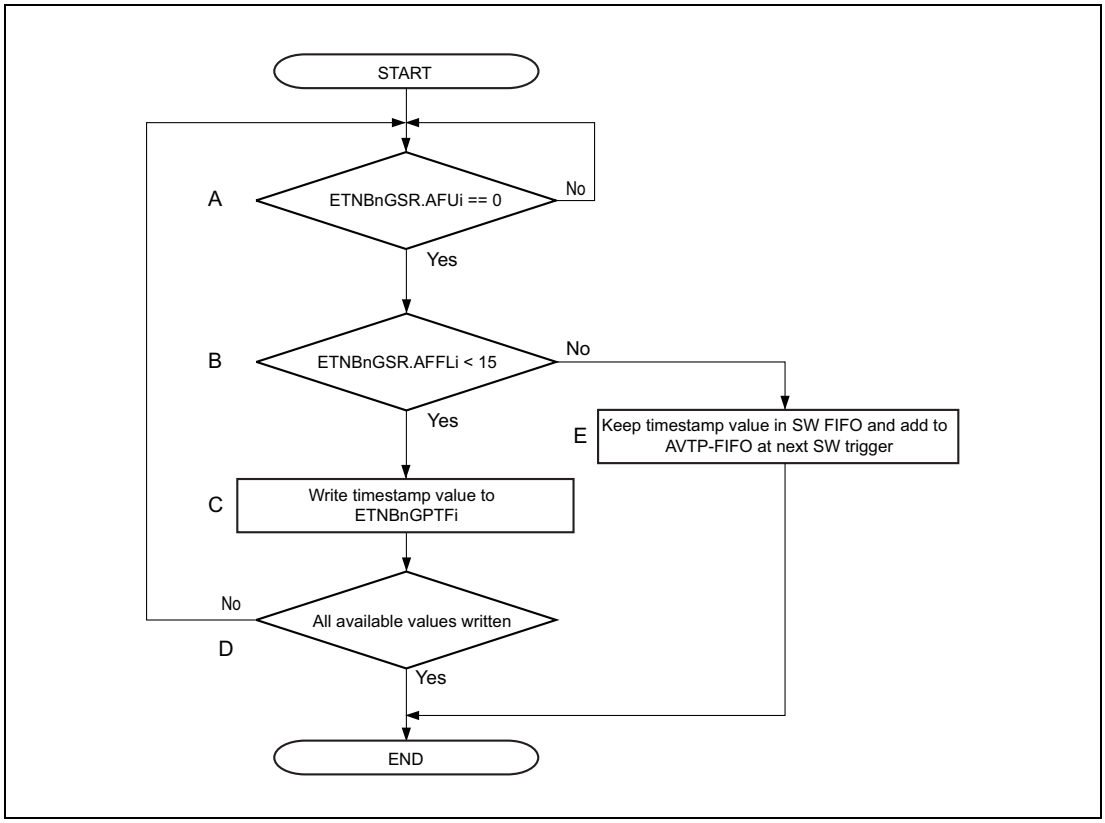


Figure 25.58 SW flow to add timestamp to AVTP-FIFO

It is recommended to combine box A and B into one access to limit accesses to ETNBnGSR register.

25.4.9 gPTP Timer Functionality in Configuration Mode

When ETNBnCCC.GAC is 1, it is possible to use some gPTP functions in configuration mode. Because the frame handling mechanisms are not available in configuration mode, usage of gPTP support is restricted. Only basic timer functions of gPTP support can be used:

- gPTP timer is running based on selected clock source (ETNBnCCC.CSEL).
- Offset and increment value can be configured and adjusted.
- Capture function (timer, corrected timer, AVTP)
- SW compare function

When AVTP FIFO functionality is used in configuration mode, there is no error flagging by ETNBnESR.ET = 1011_B available.

25.4.10 Flow Control

The E-MAC supports flow control for full-duplex operation in compliance with the IEEE 802.3 standards. This flow control is applicable to both reception and transmission. In regard to the transmission of PAUSE frames, flow control operates in the following ways.

(1) PAUSE Frame Transmission

PAUSE frames can also be transmitted in response to software operations. Writing a timer value to the manual PAUSE frame register (ETNBnMPR) starts the transmission of a PAUSE frame. This only causes the transmission of one PAUSE frame.

(2) PAUSE Frame Reception

After reception of a PAUSE frame, transmission of the next frame does not proceed until the time indicated by the Timer value elapses. However, transmission of a frame currently being transmitted continues. PAUSE frames are only received while the operation mode for flow control in reception in the E-MAC mode register (ETNBnECMR.RXF) is set to “1”. The number of received PAUSE frames is counted.

(3) PAUSE Frames with the Timer Value 0

The setting of the 0-time PAUSE frame enable bit (ETNBnECMR.ZPF) enables or disables the reception and transmission of PAUSE frames with the TIME parameter value 0.

- When control of PAUSE frames with the TIME parameter value 0 is enabled
  A PAUSE frame with the TIME parameter value 0 is transmitted when the capacity of the reception FIFO is less than the value of the Receive FIFO Warning Level bits (ETNBnRCR.RFCL[12:0]) while the time indicated by the TIME parameter value has not elapsed.
  Reception of a PAUSE frame with the TIME parameter value 0 leads to release from the transmission standby state.
- When control of PAUSE frames with the TIME parameter value 0 is disabled
  PAUSE frames with the TIME parameter value 0 are not transmitted. Received PAUSE frames with the TIME parameter value 0 are discarded.

25.4.11 Interrupts

The EthernetAVB module has three EI level interrupts from the AVB-DMAC and one EI level interrupt from the E-MAC.

Table 25.122 lists interrupts.

Table 25.122 EthernetAVB Interrupts

Interrupt Source Name
Transmit/receive data management interrupt
Error management interrupt
Other management (FIFO caution level, etc.) interrupt
E-MAC interrupt

The AVB-DMAC related interrupts include descriptor interrupts (15 sources), error interrupts (11 sources), reception interrupts (56 sources), transmission interrupts (8 sources), and gPTP interrupts (8 source). From the CPU's perspective, each appears as three of the above four interrupt sources.

The states of an AVB-DMAC-related interrupt sources can be checked in the following registers.

- Descriptor interrupt status register (ETNBnDIS)
- Error interrupt status register (ETNBnEIS)
- Receive interrupt status register (ETNBnRIS0 to 2)
- Transmit interrupt status register (ETNBnTIS)
- gPTP interrupt status register (ETNBnGIS)

The interrupts are controlled by the corresponding interrupt enable bits. However, the status flags operate independently of the settings of the enable bits.

The states of grouped interrupts can only be checked by reading the interrupt summary status register (ETNBnISS) and the queue full error interrupt status bit in the error interrupt status register (ETNBnEIS.QFS). This reduces the load on the CPU.

25.4.11.1 Transmit/Receive Data Management Interrupt

The management interrupt for transmission and reception is conveyed when the following interrupt sources are generated:

- Receive frame interrupt in the receive interrupt status register 0 (ETNBnRIS0.FRFr)
- Receive warning interrupt in the receive interrupt status register 1 (ETNBnRIS1.RWFr)
- Frame transmitted interrupt in the transmit interrupt status register (ETNBnTIS.FTFt)
- Descriptor interrupt in the descriptor interrupt status register (ETNBnDIS.DPFi)

The general error interrupt state can be checked by reading the descriptor interrupt summary bits in the interrupt summary status register (ETNBnISS.DPMi), the frame received interrupt summary bit (ETNBnISS.FRM), the receive warning interrupt summary bit (ETNBnISS.RWM) and the frame transmitted interrupt summary bit (ETNBnISS.FTM).

25.4.11.2 Error Management Interrupt

The error management interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

- Tx-Buffer full interrupt when delaying fetching of transmit frame (ETNBnEIS.TBFF)
- Interrupt for MAC Status FIFO overwrite (ETNBnEIS.MFFF)
- Timestamp FIFO full error interrupt in the error interrupt status register (ETNBnEIS.TFFF)
- CBS limitation interrupts in the error interrupt status register (ETNBnEIS.CULF1, ETNBnEIS.CULF0, ETNBnEIS.CLLF1, ETNBnEIS.CLLF0)
- Separation Error in the error interrupt status register (ETNBnEIS.SEF)
- Queue Error in the error interrupt status register (ETNBnEIS.QEF)
- MAC Transmission Error in the error interrupt status register (ETNBnEIS.MTEF)
- MAC Reception Error in the error interrupt status register (ETNBnEIS.MREF)
- Receive FIFO full interrupt in the receive interrupt status register 2 (ETNBnRIS2.RFFF)
- Receive queue full interrupt in the receive interrupt status register 2 (ETNBnRIS2.QFFr)

The general error interrupt state can be checked by reading the error interrupt summary bit in the interrupt summary status register (ETNBnISS.EM).

25.4.11.3 Other Management (FIFO Warning, etc.) Interrupts

The other management (FIFO warning, etc.) interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

(1) Reception related interrupt

Receive FIFO warning interrupt in the receive interrupt status register 1 (ETNBnRIS1.RFWF)

(2) Transmission related interrupts

MAC status FIFO warning interrupt in the transmit interrupt status register (ETNBnTIS.MFWF)

MAC status FIFO update interrupt in the transmit interrupt status register (ETNBnTIS.MFUF)

Timestamp FIFO warning interrupt in the transmit interrupt status register (ETNBnTIS.TFWF)

Timestamp FIFO update interrupt in the transmit interrupt status register (ETNBnTIS.TFUF)

(3) gPTP related interrupts

AVTP presentation target match interrupt in the gPTP interrupt status register (ETNBnGIS.PTMFi)

The general error interrupt state can be checked by reading the receive FIFO warning error interrupt status bit in the interrupt summary status register (ETNBnISS.RFWM), the MAC status FIFO warning interrupt status bit (ETNBnISS.MFWM), the MAC status FIFO updated interrupt status bit (ETNBnISS.MFUM), the timestamp FIFO warning interrupt status bit (ETNBnISS.TFWM), the timestamp FIFO update interrupt status bit (ETNBnISS.TFUM), and the combined gPTP interrupt status bit (ETNBnISS.CGIM).

25.4.11.4 E-MAC Interrupt

The E-MAC interrupt is conveyed when the E-MAC interrupt source is generated.

The general error interrupt state can be checked by reading the E-MAC interrupt summary bit in the interrupt summary status register (ETNBnISS.MM).

25.4.12 Flows of Operations

25.4.12.1 Flow of E-MAC Initialization

Figure 25.59 shows the flow of E-MAC initialization (for AVB mode and full-duplex operation).

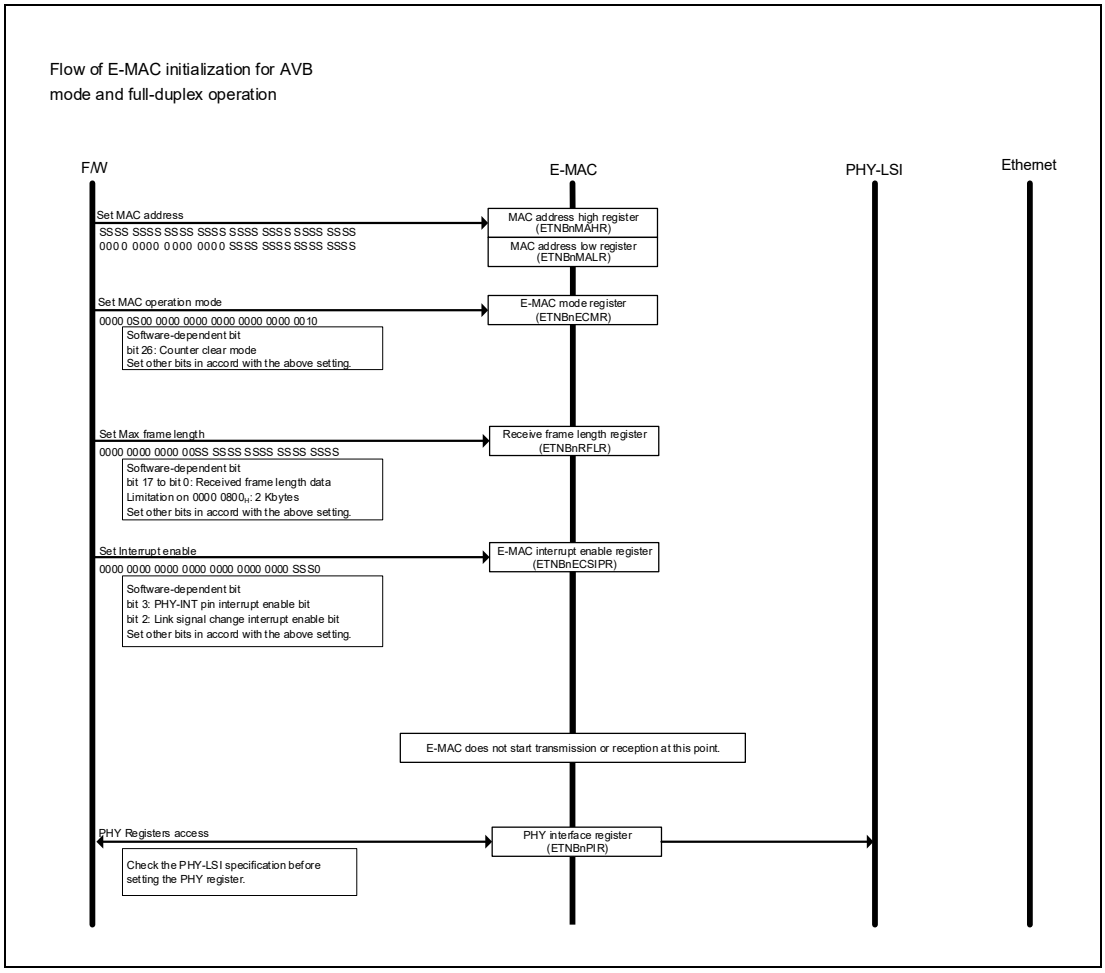


Figure 25.59 Flow of E-MAC Initialization (for AVB Mode and Full-Duplex Operation)

25.4.12.2 Flow of AVB-DMAC Initialization

Figure 25.60 shows the flow of AVB-DMAC initialization (for AVB mode and full-duplex operation).

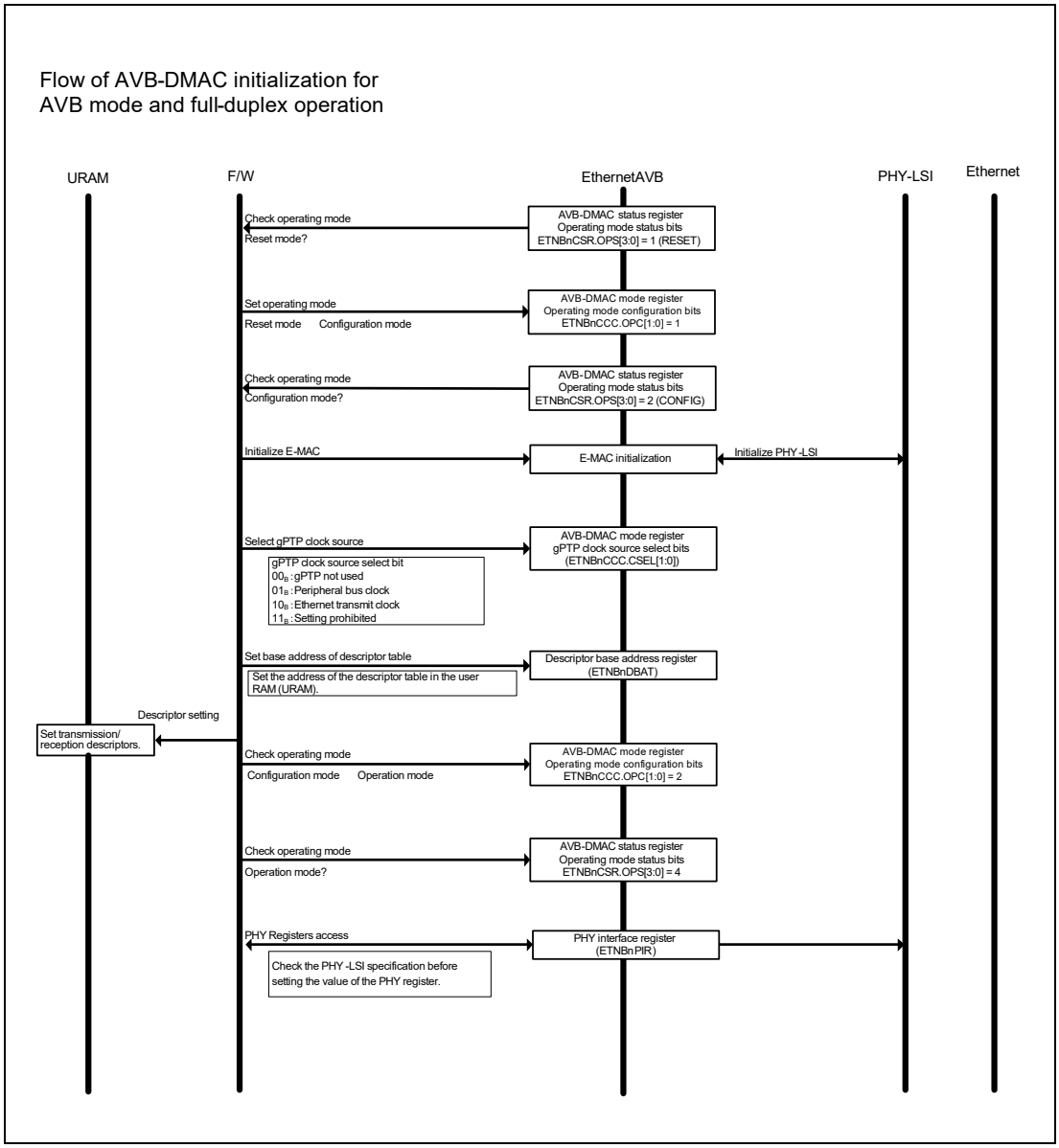


Figure 25.60 Flow of AVB-DMAC Initialization (for AVB Mode and Full-Duplex Operation)

25.4.12.3 Flow for the AVB-DMAC in Reception

Figure 25.61 shows the flow for the AVB-DMAC in reception (in AVB mode and full-duplex operation).

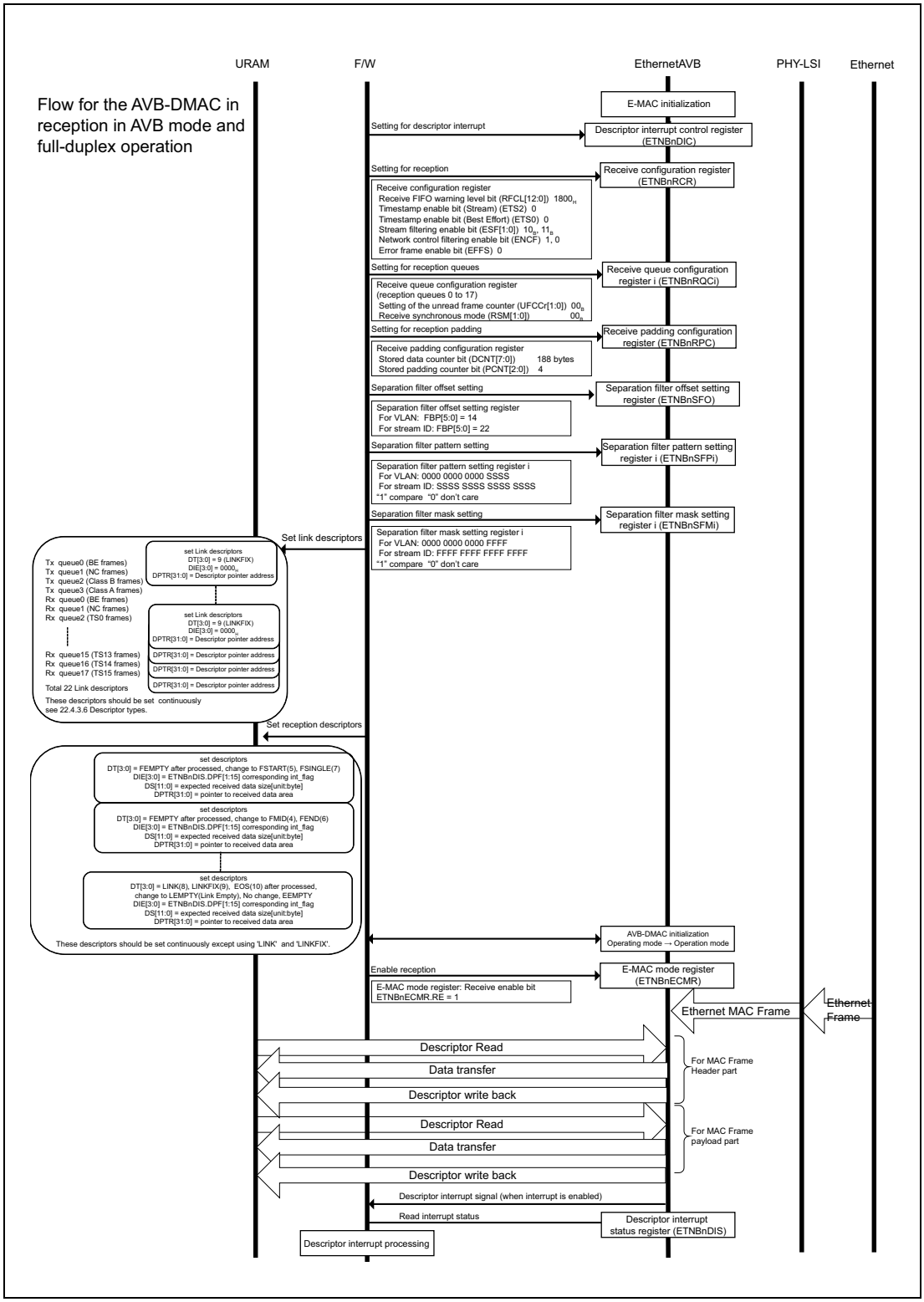


Figure 25.61 Flow for the AVB-DMAC in Reception (in AVB Mode and Full-Duplex Operation)

25.4.12.4 Flow for the AVB-DMAC in Transmission

Figure 25.62 shows the flow for the AVB-DMAC in transmission (in AVB mode and full-duplex operation).

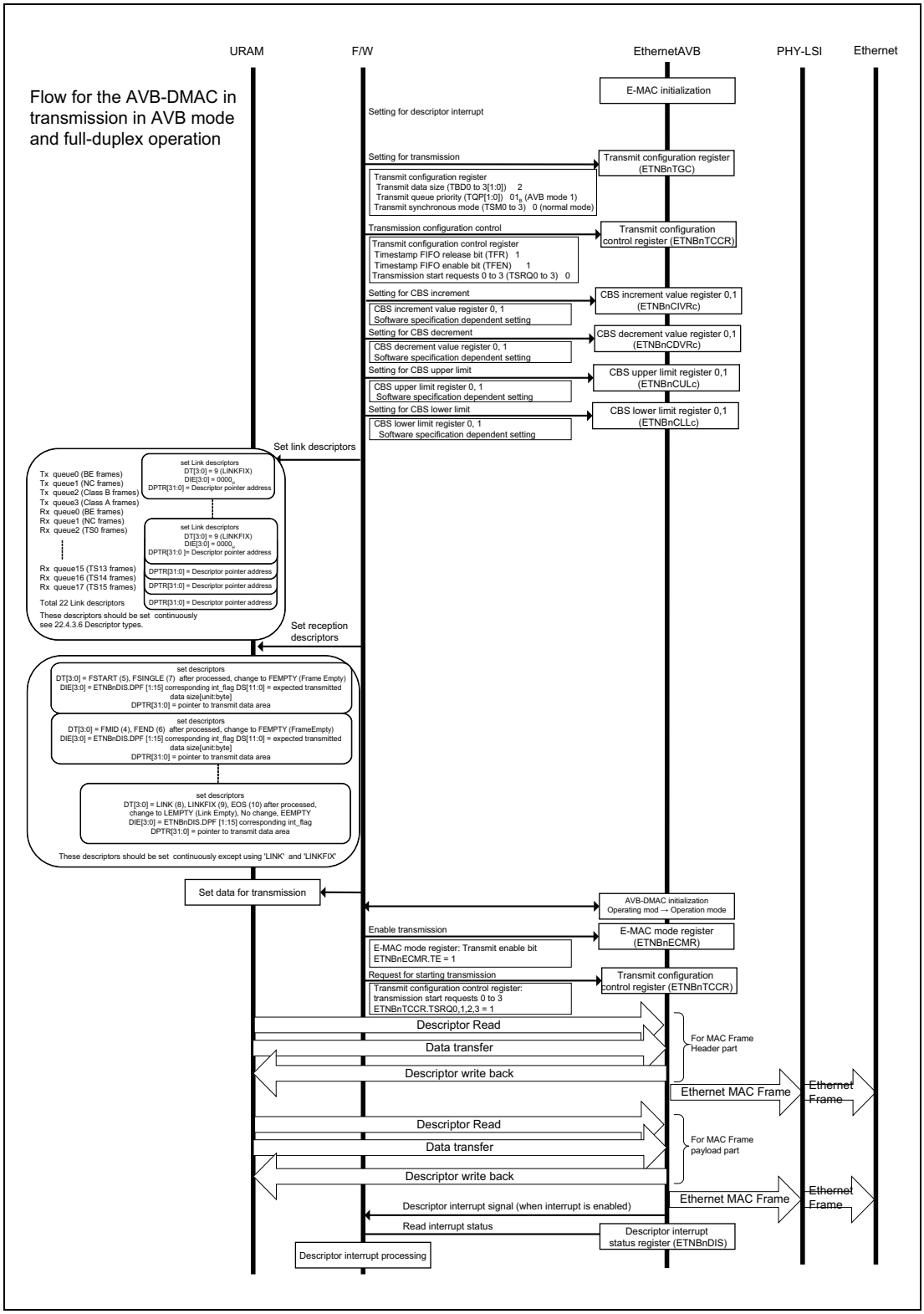


Figure 25.62 Flow for the AVB-DMAC in Transmission (in AVB Mode and Full-Duplex Operation)

25.4.12.5 Flow for Stopping AVB-DMAC Operation in Reception

Figure 25.63 shows the flow for stopping AVB-DMAC operation in reception (normal, common to all modes).

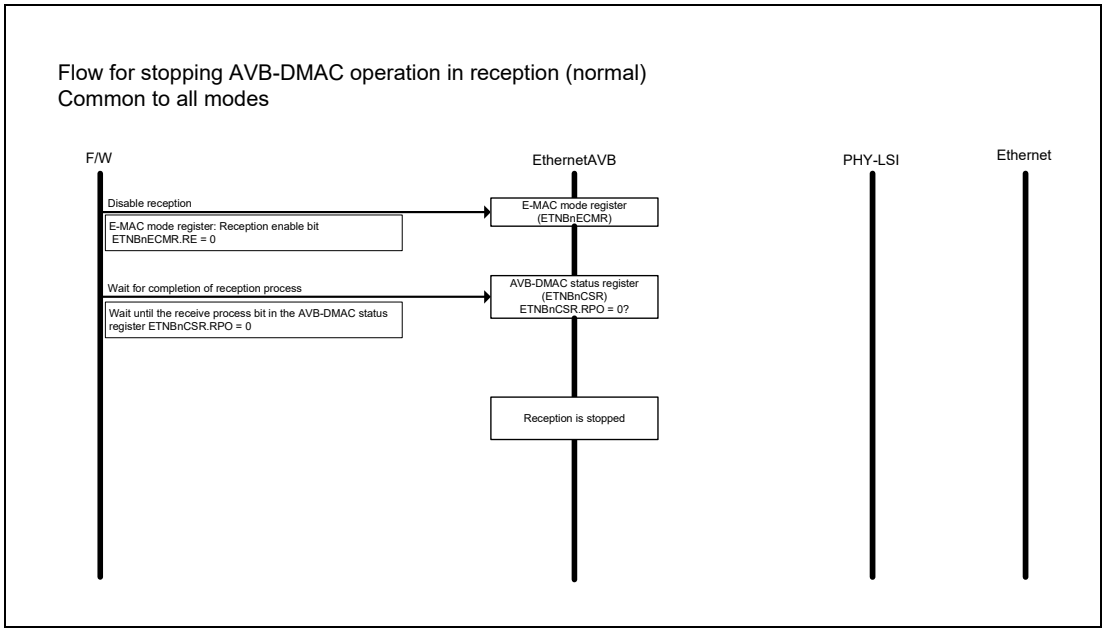


Figure 25.63 Flow for Stopping AVB-DMAC Operation in Reception (Normal, Common to All Modes)

25.4.12.6 Flow for Stopping AVB-DMAC Operation in Transmission

Figure 25.64 shows the flow for stopping AVB-DMAC operation in transmission (normal, common to all modes).

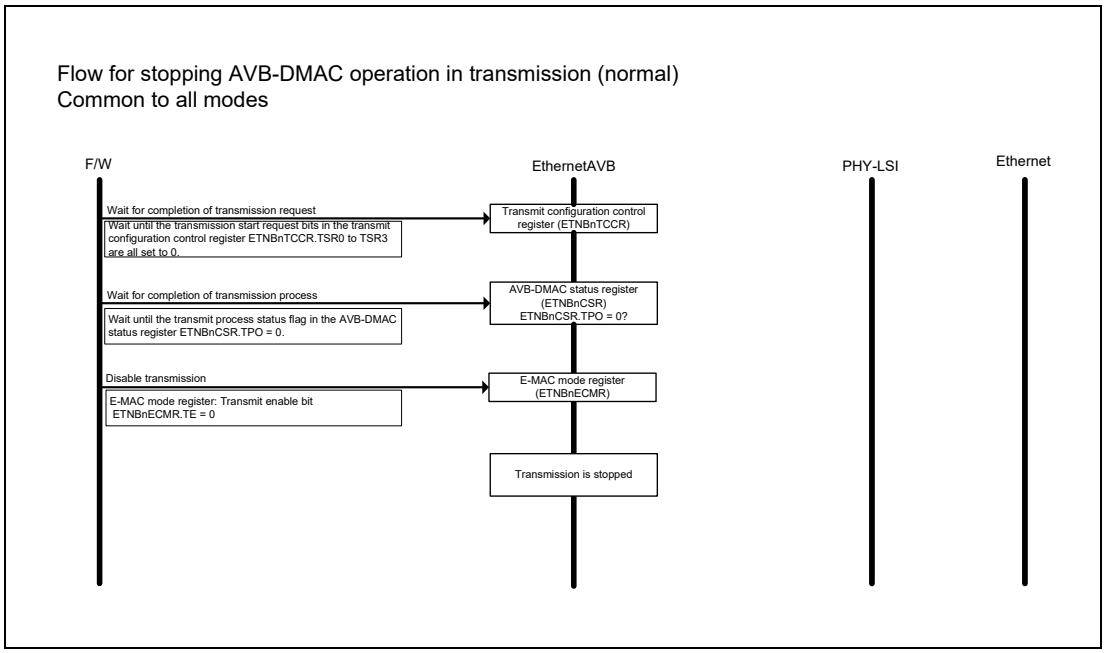


Figure 25.64 Flow for Stopping AVB-DMAC Operation in Transmission (Normal, Common to All Modes)

25.4.12.7 Flow for Stopping and Resetting the AVB-DMAC

Figure 25.65 shows the flow for stopping and resetting the AVB-DMAC (normal, common to all modes).

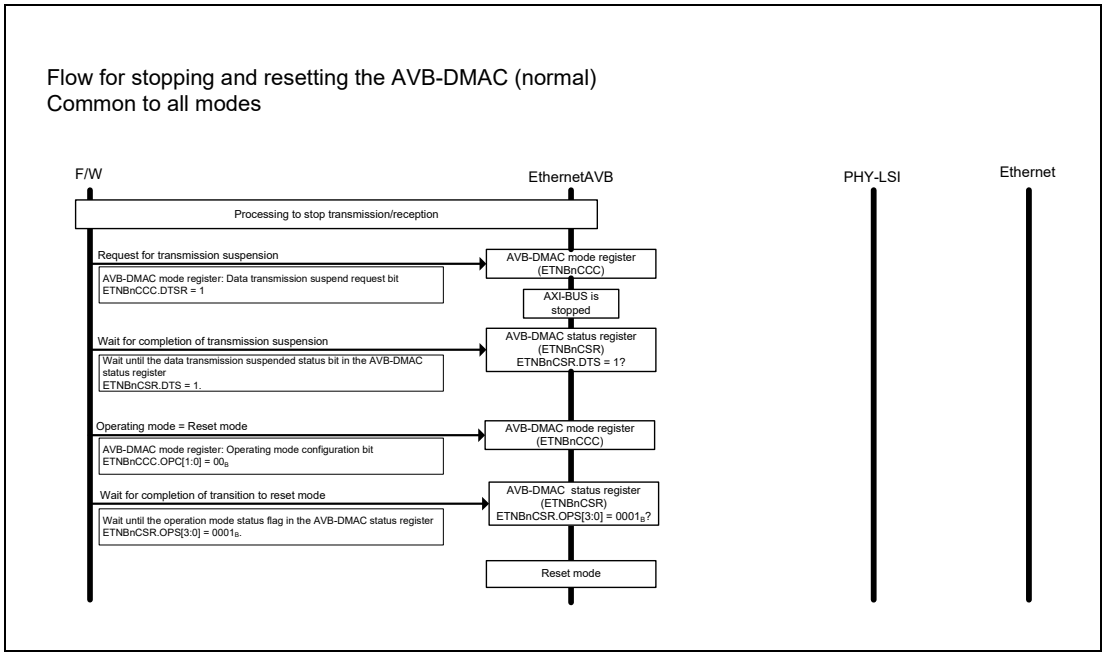


Figure 25.65 Flow for Stopping and Resetting the AVB-DMAC (Normal, Common to All Modes)

25.4.12.8 Flow for Emergency Stopping the AVB-DMAC

Figure 25.66 shows the flow for emergency stopping the AVB-DMAC (normal, common to all modes).

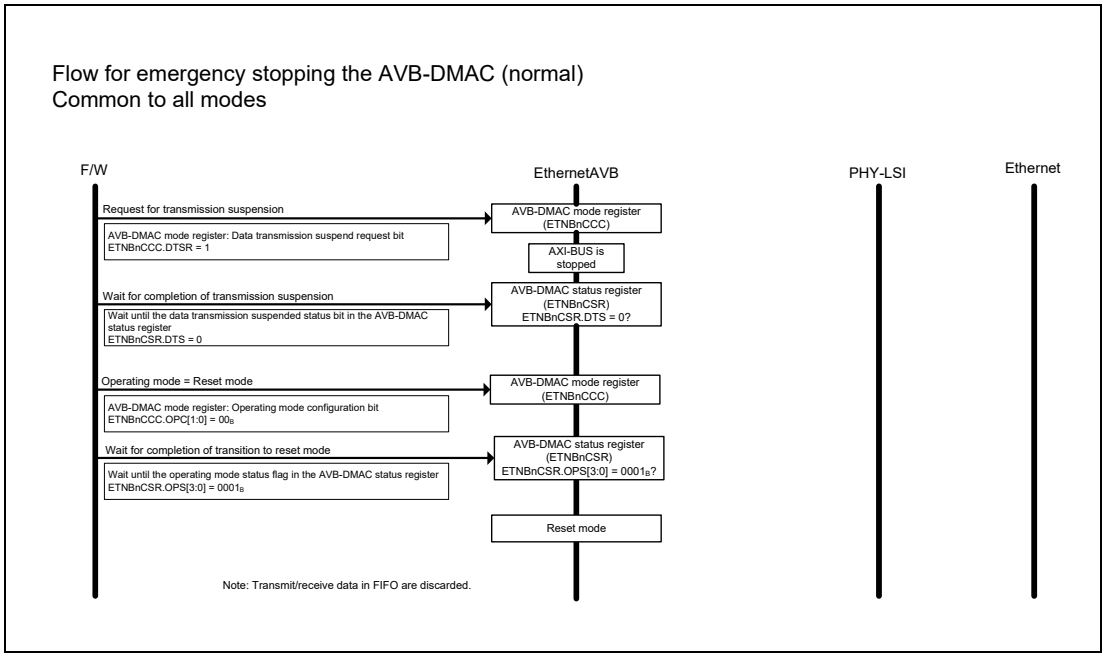


Figure 25.66 Flow for Emergency Stopping the AVB-DMAC (Normal, Common to All Modes)

25.4.12.9 Flow of gPTP Initialization

Figure 25.67 shows the flow of gPTP initialization (normal, common to all modes).

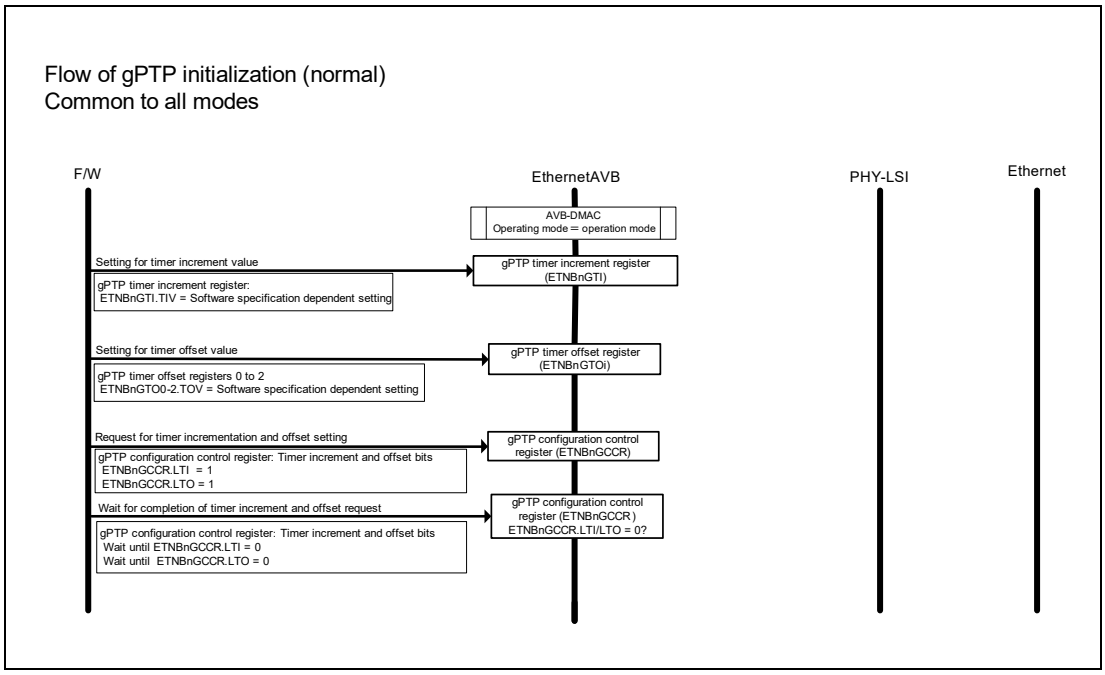


Figure 25.67 Flow of gPTP Initialization (Normal, Common to All Modes)

25.4.12.10Flow of gPTP Timestamping in Transmission

Figure 25.68 shows the flow of gPTP timestamping in transmission (normal, common to all modes).

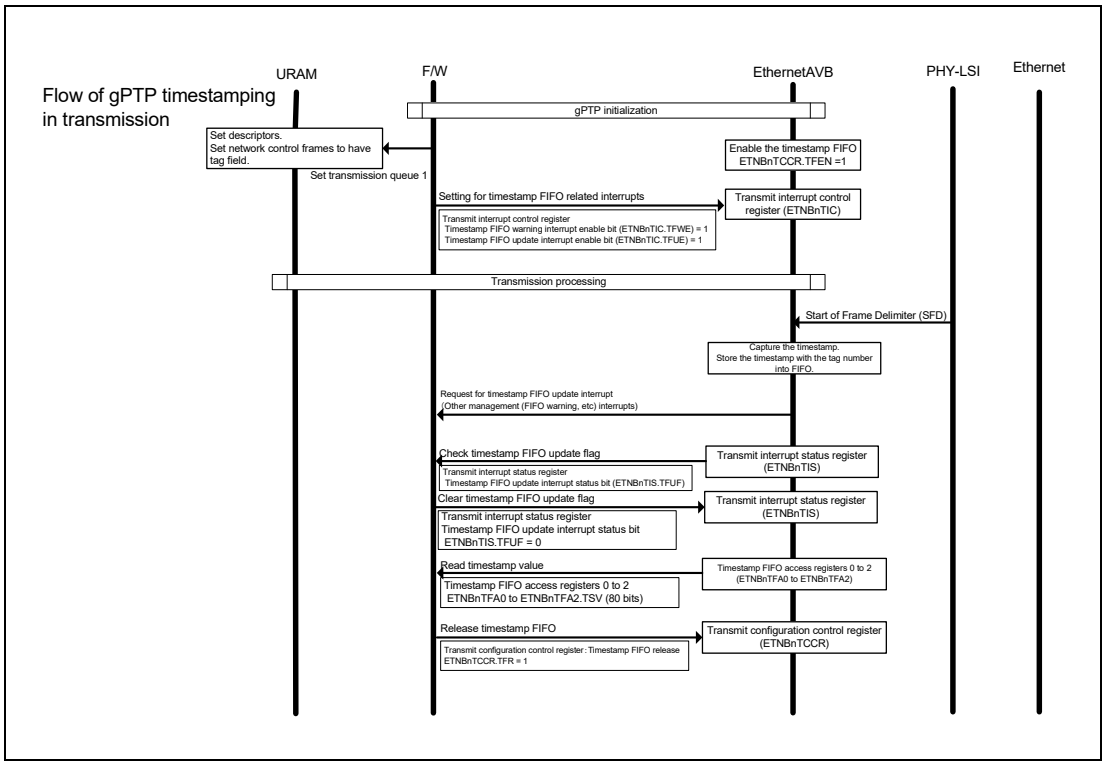


Figure 25.68 Flow of gPTP Timestamping in Transmission (Normal, Common to All Modes)

25.4.12.11 Flow of gPTP Timestamping and Synchronization in Reception

Figure 25.69 shows the flow of gPTP timestamping and synchronization in reception (normal, common to all modes).

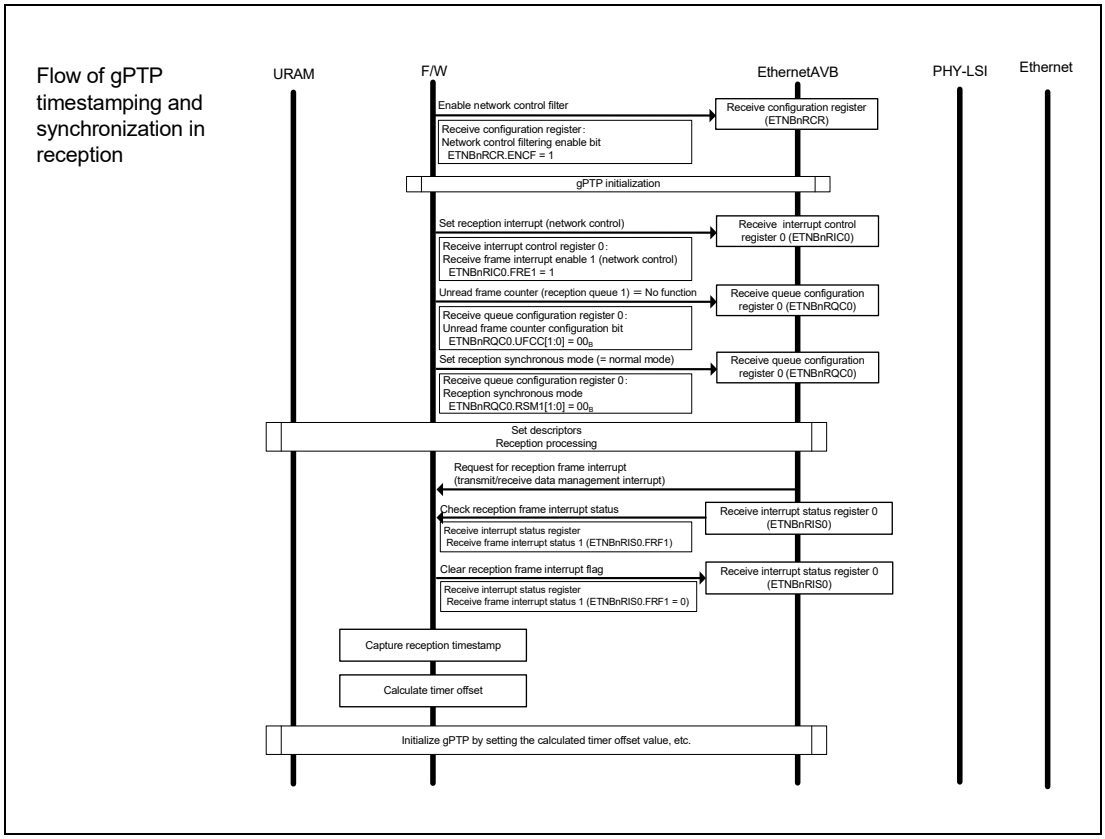


Figure 25.69 Flow of gPTP Timestamping and Synchronization in Reception (Normal, Common to All Modes)

25.4.12.12Flow of Capturing gPTP Presentation Times

Figure 25.70 shows the flow of capturing gPTP presentation times (common to all modes).

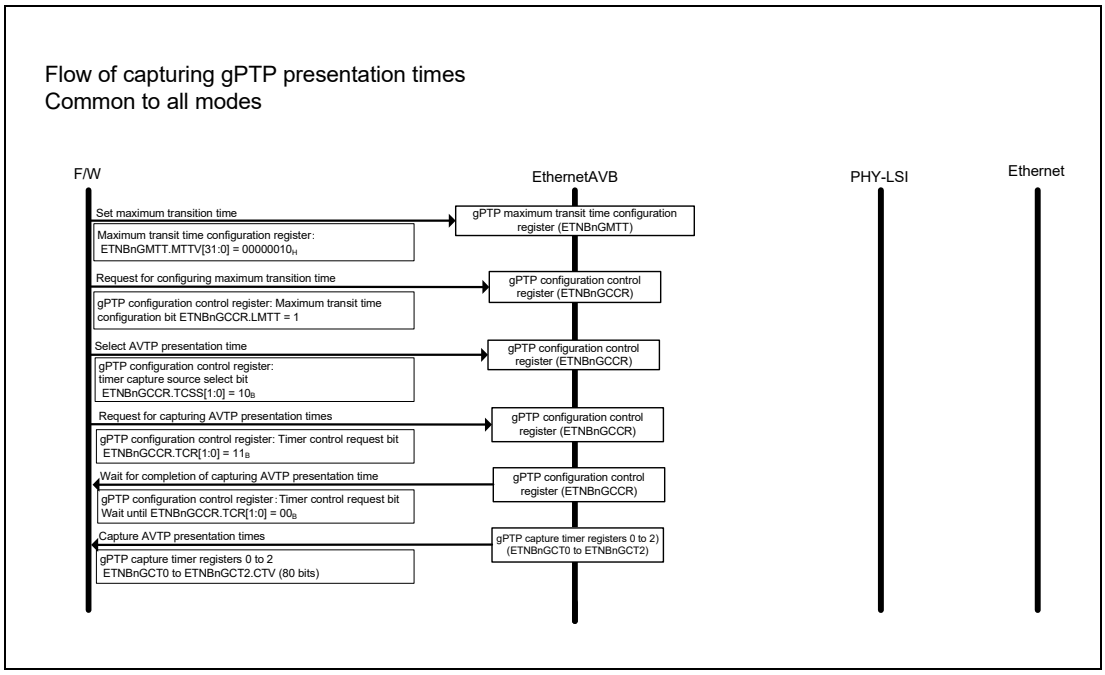


Figure 25.70 Flow of Capturing gPTP Presentation Times (Common to All Modes)

25.4.12.13Flow of AVTP Presentation Time Comparison

Figure 25.71 shows the flow of AVTP presentation time comparison (common to all modes).

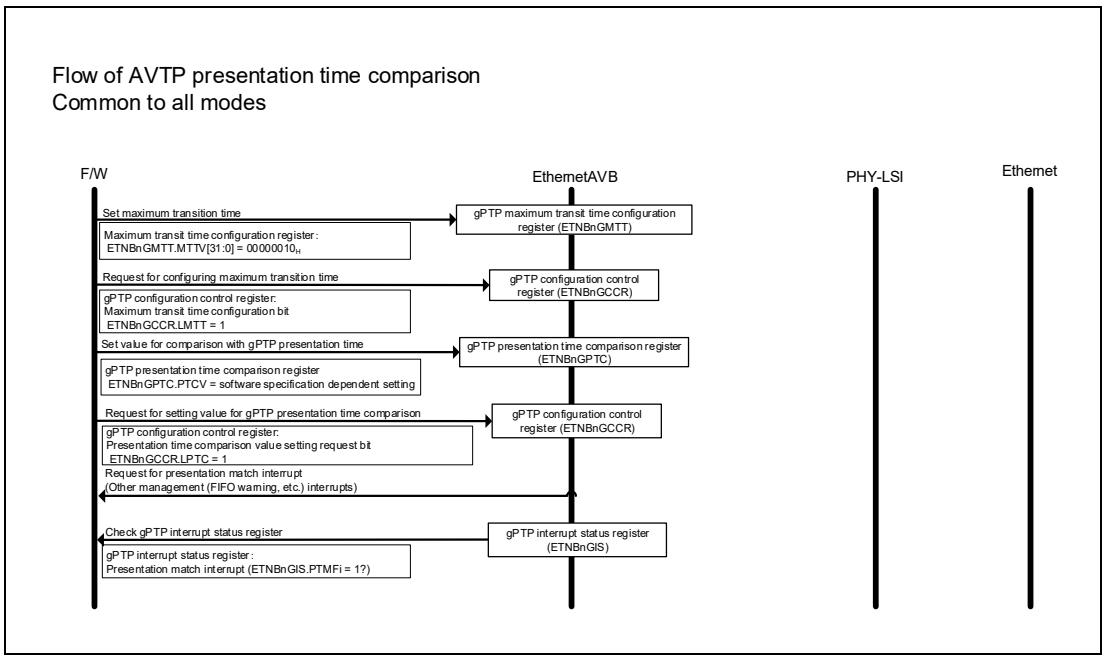


Figure 25.71 Flow of AVTP Presentation Time Comparison (Common to All Modes)

25.4.12.14Flow of Loopback Mode Operation

Figure 25.72 shows the flow of loopback mode operation.

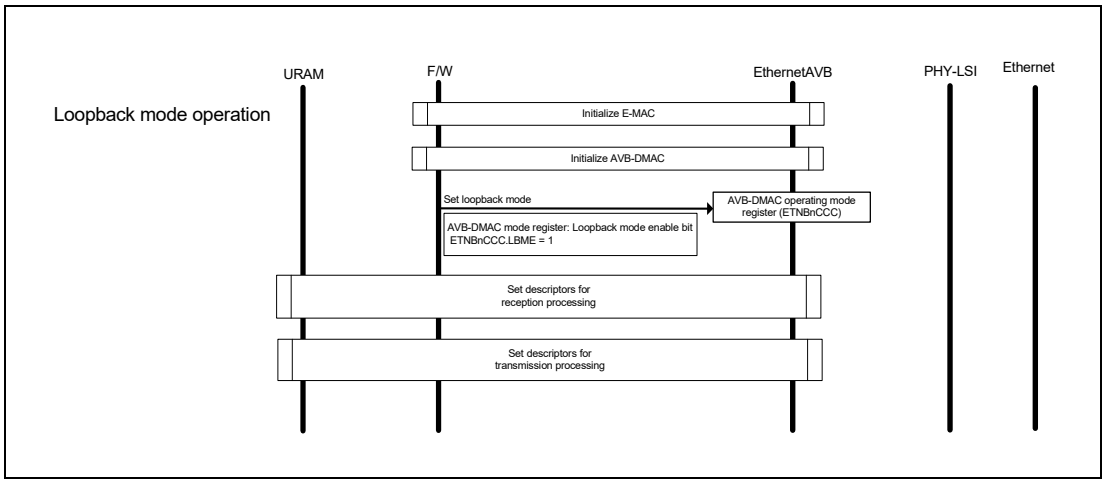


Figure 25.72 Flow of Loopback Mode Operation

25.4.13 Connection to PHY-LSI

25.4.13.1 Accessing MII Registers

MII registers in the PHY-LSI are accessed via ETNBnPIR in this LSI. ETNBnPIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

25.4.13.2 MII Management Frame Format

Figure 25.73 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in **25.4.13.3 MII Register Access Procedure**.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s
ST: Write of 01_B indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 00001_B if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
REGAD: Write of 0001_B if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10_B written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(d) Read: Bus already released in TA: control unnecessary

Figure 25.73 MII Management Frame Format

25.4.13.3 MII Register Access Procedure

The program accesses MII registers via ETNBnPIR. Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. **Figure 25.74** to **Figure 25.77** show the MII register access timing. The timing will differ depending on the PHY-LSI type.

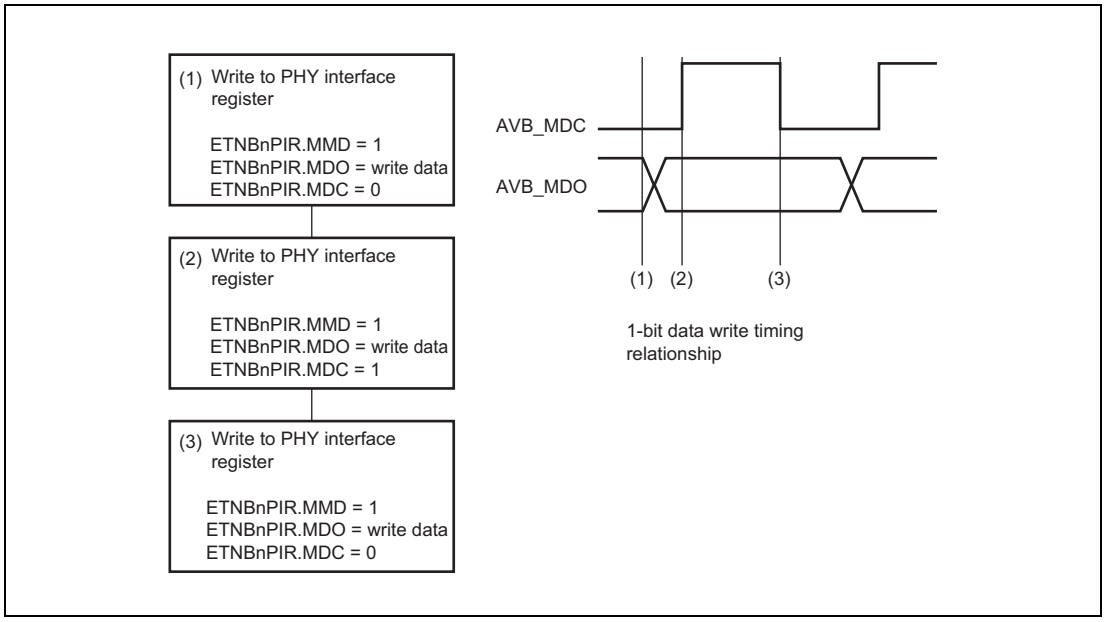


Figure 25.74 1-Bit Data Write Flowchart

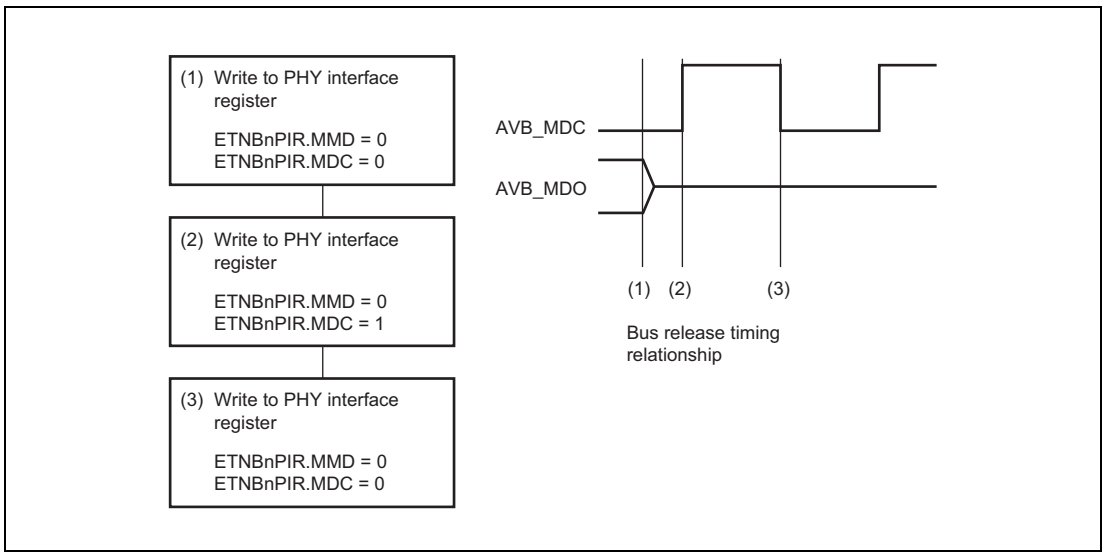


Figure 25.75 Bus Release Flowchart (TA in Read in Figure 25.73)

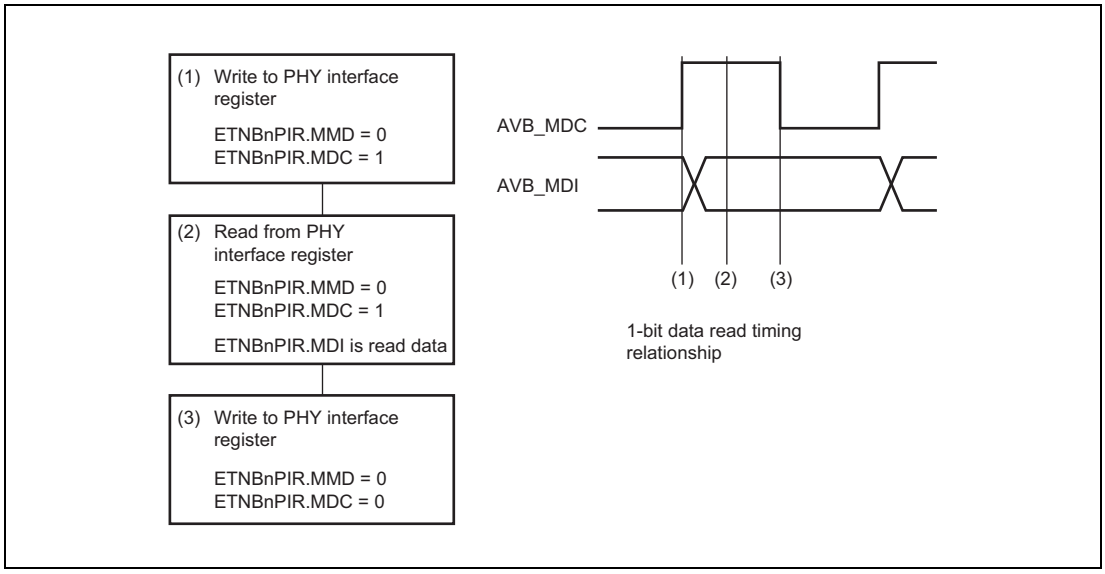


Figure 25.76 1-Bit Data Read Flowchart

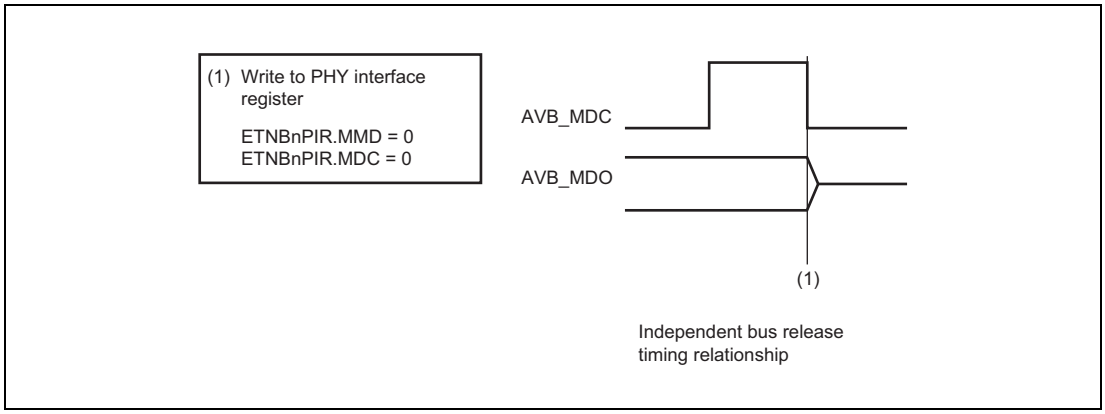


Figure 25.77 Independent Bus Release Flowchart (IDLE in Write in Figure 25.73)

25.4.14 SGMII interface

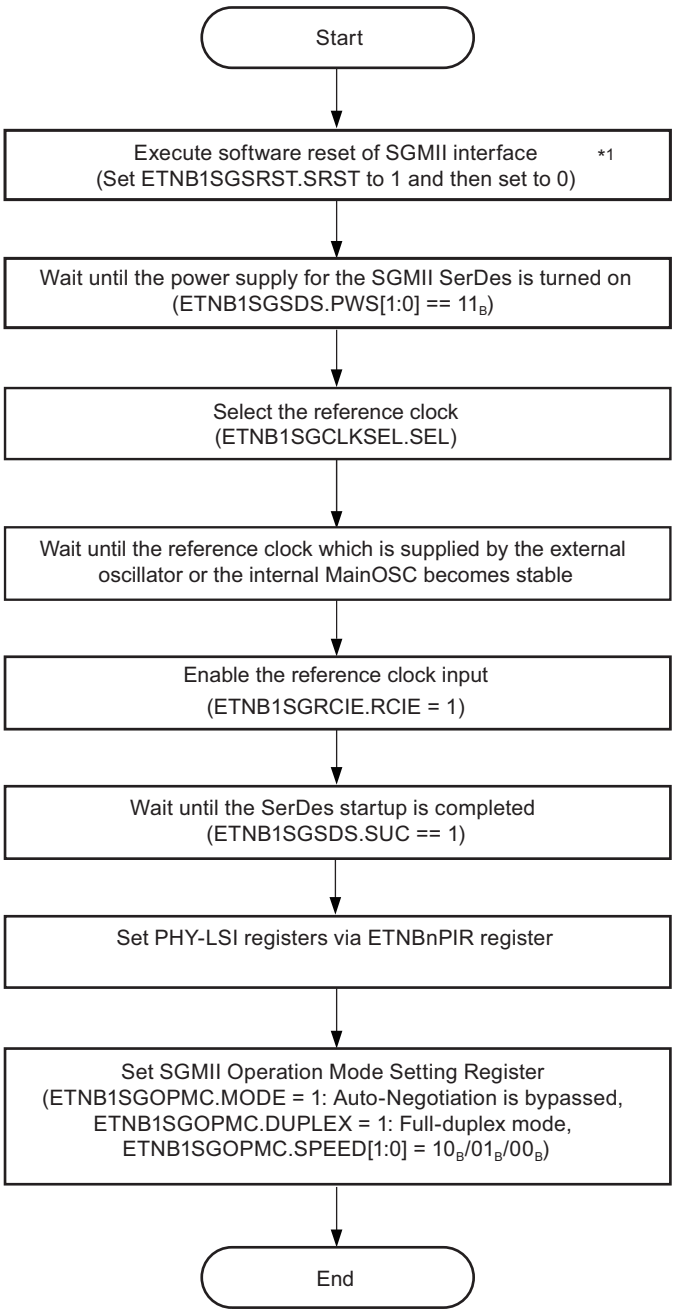
25.4.14.1 SGMII SerDes Power Control

The Power supply for the SGMII SerDes is controlled as follows.

- After power-on reset, the power supply for the SGMII SerDes is off.
- When the ETNB is in Module standby mode, the power supply for the SGMII SerDes is kept off. When the Module standby of the ETNB is cancelled, the power supply for the SGMII SerDes is turned on. When the ETNB transits to Module standby mode again, the power supply for the SGMII SerDes is turned off again.
- When the reset (which sources are shown in **Section 25.1.5, Reset Sources**) for the ETNB is activated, the power supply for the SGMII SerDes is turned off. At this time. the MSR_ETNB register (Module Standby Register for ETNB) is also reset, so the ETNB transits to Module standby mode, and the power supply for SGMII SerDes is kept off until the Module standby mode is cancelled.
- When the chip transits to STOP mode (refer to **Section 15, Standby Controller (STBC)**), the power supply for the SGMII SerDes is turned off. When the STOP mode is cancelled, the power supply for the SGMII SerDes is turned on.

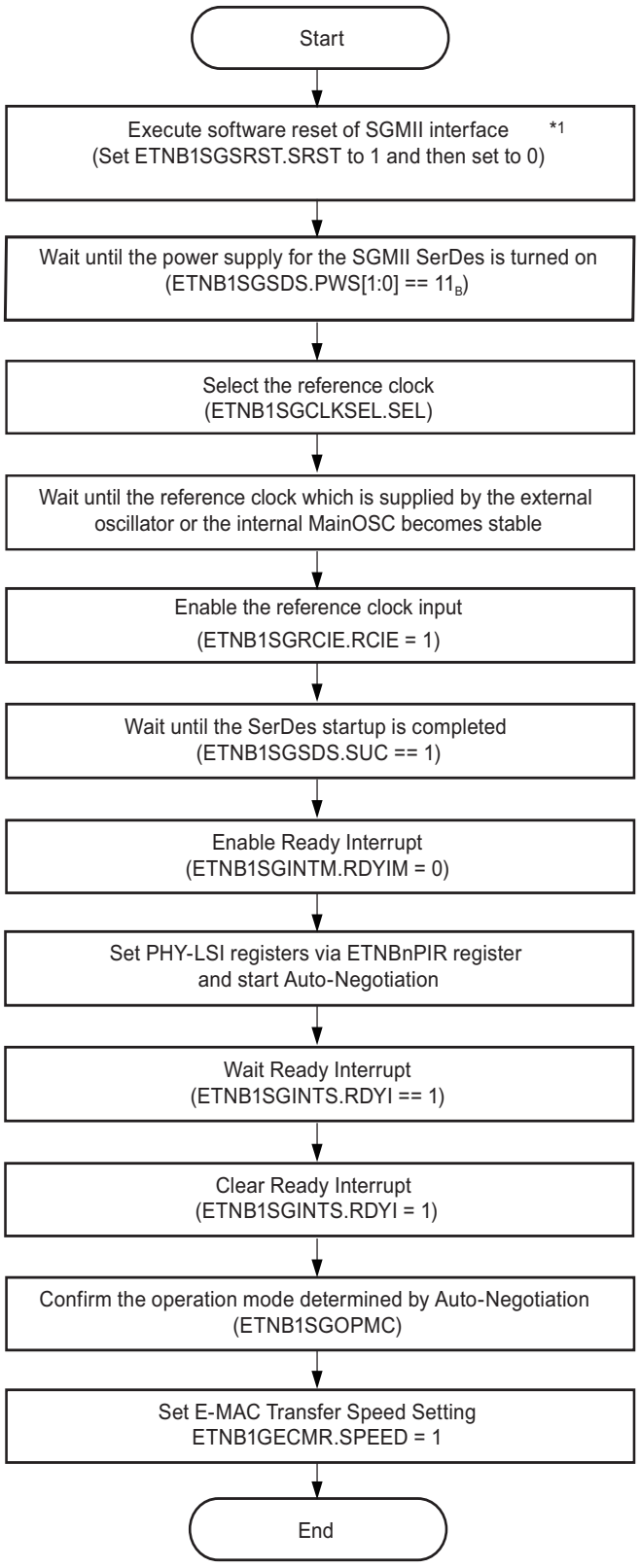
25.4.14.2 SGMII System Startup

After cancelling Module standby mode of the ETNB or STOP mode, before operating AVB-DMAC and E-MAC, the CPU must follow the sequence of steps shown in **Figure 25.78** or **Figure 25.79**. The **Figure 25.78** shows the flow in the case when Auto-Negotiation with PHY-LSI is bypassed (ETNB1SGOPMC.MODE = 1). The **Figure 25.79** shows the flow in the case when Auto-Negotiation with PHY-LSI is not bypassed (ETNB1SGOPMC.MODE = 0).



Note 1. Need to input Configuration pattern or IDLE Ordered Set pattern more than 4µs.
The note doesn't apply for RH850/U2A-EVA.

Figure 25.78 SGMII System Startup (Auto-Negotiation with PHY-LSI is bypassed)



Note 1. Need to input Configuration pattern or IDLE Ordered Set pattern more than 4µs.
The note doesn't apply for RH850/U2A-EVA.

Figure 25.79 SGMII System Startup (Auto-Negotiation with PHY-LSI is not bypassed)

25.4.15 Usage Notes

25.4.15.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to checksum calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. **Figure 25.80** shows schematics indicating which parts of the Ethernet frames are calculated. Calculation involves 16-bit addition only; it does not involve bit inversion. Note that when the checksum data is valid, the CRC data (4 bytes) is not transferred as a receive frame, and the checksum data (sum data) is added automatically. **Figure 25.81** shows schematics of Ethernet frames to which the checksum data has been added.

CAUTION

Also for the frames with VLANtag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

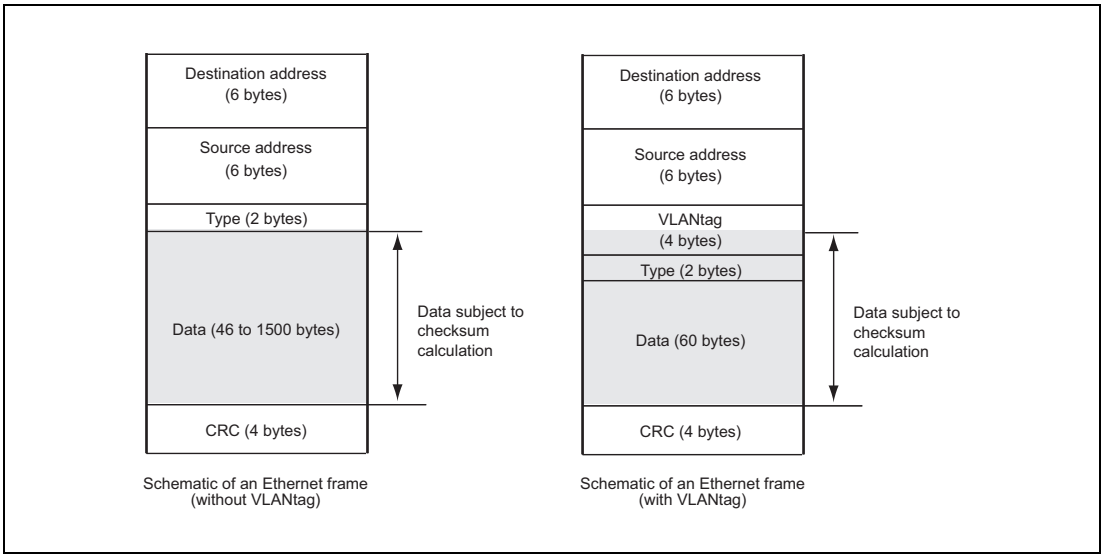


Figure 25.80 Data Subject to Checksum Calculation

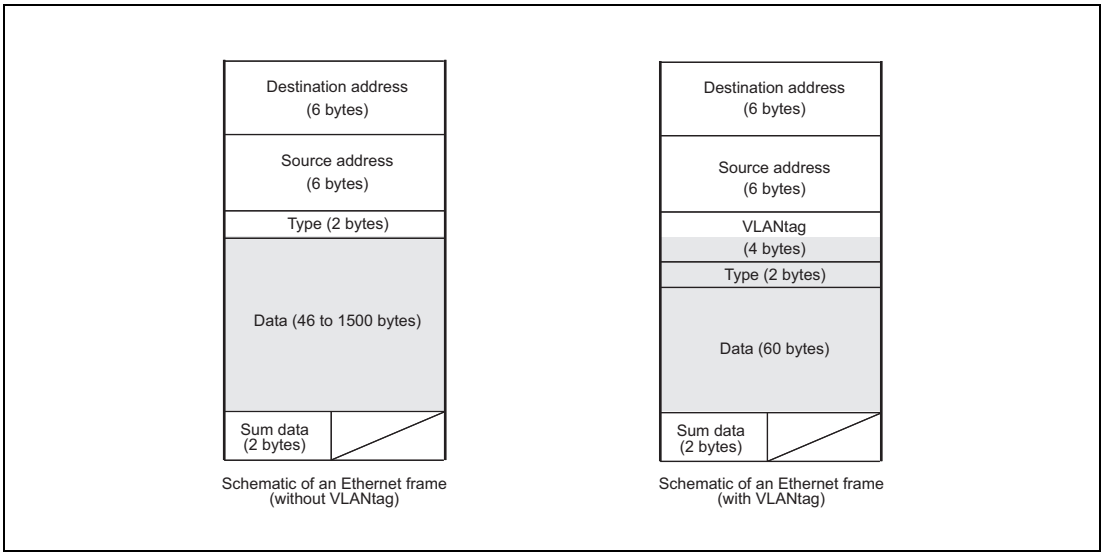


Figure 25.81 Data after Checksum Data Addition

25.4.15.2 Rx-FIFO read error may not be flagged when using FEMPTY_ND descriptor

After reading the last byte targeted to the FEMPTY_ND descriptor, there can be up to 3 bytes read from Rx-FIFO targeted to the next descriptor of the same frame. Rx-FIFO read errors of not stored data due to FEMPTY_ND are neither flagged in DESCR.EI nor by ETNBnEIS.QEF as per target specification.

When external ECC logic flags detect Rx-FIFO read error related to last byte targeted to FEMPTY_ND descriptor, data in the next descriptor may be corrupted without notification.

The issue is limited to implementations where Rx-FIFO error information is provided to Ethernet AVB and also to applications using FEMPTY_ND descriptor.

The issue only occurs when the first byte saved in the descriptor following FEMPTY_ND is not a multiple of 4 bytes inside the received frame.

Use a FEMPTY descriptor to store unwanted data if consistent error flagging is required.

25.4.15.3 When trying to release non-existing timestamp FIFO entry, new FIFO update flag may be lost

When SW releases an entry of timestamp FIFO by writing ETNBnTCCR.TFR while this FIFO is empty (ETNBnTSR.TFFL is set to 0), flagging of next FIFO update may be inconsistent. The next timestamp is correctly stored in FIFO and the fill level is incremented to 1 but ETNBnTIS.TFUF is not set to 1.

The issue is limited to applications releasing FIFO entries without checking if there are entries available.

Do not release not existing FIFO entries .

25.4.15.4 gPTP compare may fail for range of compare values

When the comparison value (ETNBnGPTC.PTCV) is in the range of [x-1 to x+1] (x is the configured increment value in ETNBnGTI.TIV), it may happen that a comparison match is not detected when Timer wraps around.

The issue is limited to applications using the AVTP comparison function.

Do not configure comparison values inside the critical range.

25.4.15.5 UFC stop level triggers ETNBnRIS2.QFFr even no received frame is lost

When a received frame is dropped due to non-availability of empty descriptor, ETNBnRIS2.QFFr is correctly set to 1.

Additionally, when ETNBnUFCVi.CVr reached the configured stop level (ETNBnUFCS.SLj), the queue full flag (ETNBnRIS2.QFFr) is set to 1 even before any further received frame is dropped.

The issue is limited to applications using unread frame counter with stop level function.

Such application gets information about lost received frames which in fact might not have lost.

25.4.15.6 ETNBnRIS0.FRFr may be lost when data processing stops close to or below the configured warning level.

When SW decrements UFC counter value (CV) to WL-1, resulting CV can still be WL due to recent new storage completion, ETNBnRIS0.FRFr is set to 0.

Any further reception will set ETNBnRIS0.FRFr as normal.

The issue is limited to applications using a single frame interrupt triggered by ETNBnRIS0.FRFr and not be able to process each interrupt in time.

The missing interrupt due to this effect is automatically recovered by the next frame reception as usual.

25.4.15.7 Receive Frame Interrupt and Descriptor Interrupt may be issued before Completion of Writing Data

When receive frame interrupt is issued, the software should check that the descriptor type is updated correctly before processing the frame data.

The descriptor type is shown in the descriptor field DESCR.DT (the address of the current descriptor is shown by the register ETNBnCDARq). If DESCR.DT is not yet updated by the DMA hardware (e.g. from FEMPTY to FEND), then the write of data to the memory is not completed. In that case the software should repeat checking the descriptor type until an update has happened.

Alternatively the software can use the Unread Frame Counter (UFC) value to compare with its processed descriptors to confirm there is no missing descriptor from the last received interrupt.

Section 26 Single Edge Nibble Transmission (RSENT)

This section contains a generic description of the Renesas Single Edge Nibble Transmission (RSENT). The first part in this section describes the RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.
The remainder of the section describes the functions and registers of the RSENT.

26.1 Features of RSENT for RH850/U2A-EVA

26.1.1 Number of Units and Channels

This microcontroller has the following number of RSENT units.

Each RSENT unit has a single channel interface. “Number of channels” therefore has the same meaning as “number of units” in this section.

Table 26.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	6 (n = 0 to 5)
Name	RSENTn									

Table 26.2 Index

Index	Description
n	Throughout this section, the individual RSENT units are identified by the index “n” (n = 0 to 7); for example, RSENTnTSPC indicates the RSENTn time stamp register.

26.1.2 Register Base Addresses

RSENTn register addresses are represented by an offset from the base address <RSENTn_base>.

The following table shows the base address <RSENTn_base> of each RSENTn channels.

Table 26.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RSENT0_base>	FFD3 3600$_H$	P-Bus Group 3
<RSENT1_base>	FFD3 3700$_H$	P-Bus Group 3
<RSENT2_base>	FFD3 3800$_H$	P-Bus Group 3
<RSENT3_base>	FFD3 3900$_H$	P-Bus Group 3
<RSENT4_base>	FFD3 3A00$_H$	P-Bus Group 3
<RSENT5_base>	FFD3 3B00$_H$	P-Bus Group 3
<RSENT6_base>	FFD3 3C00$_H$	P-Bus Group 3
<RSENT7_base>	FFD3 3D00$_H$	P-Bus Group 3

26.1.3 Clock Supply

The RSENTn clock supply is shown in the following table.

Table 26.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
RSENTn	RSENT Communication Clock	CLK_HSB	Communication clock
	Register Access Clock	CLK_HSB	Bus clock

Note: Communication clock and register access clock must be in the range of 33.3 MHz to 80 MHz.

26.1.4 Interrupt, DMA/DTS Requests and Error Notifications

RSENTn interrupt and DMA/DTS requests are listed in the following table.

Table 26.5 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger number	DTS Trigger number
RSENT0				
INTRSENT0SI	RSENT status interrupt	512	—	—
INTRSENT0RI	RSENT receive interrupt	513	group0-21	group3-42
RSENT1				
INTRSENT1SI	RSENT status interrupt	514	—	—
INTRSENT1RI	RSENT receive interrupt	515	group0-22	group3-43
RSENT2				
INTRSENT2SI	RSENT status interrupt	516	—	—
INTRSENT2RI	RSENT receive interrupt	517	group0-23	group3-44
RSENT3				
INTRSENT3SI	RSENT status interrupt	518	—	—
INTRSENT3RI	RSENT receive interrupt	519	group0-24	group3-45
RSENT4				
INTRSENT4SI	RSENT status interrupt	520	—	—
INTRSENT4RI	RSENT receive interrupt	521	group0-25	group3-46
RSENT5				
INTRSENT5SI	RSENT status interrupt	522	—	—
INTRSENT5RI	RSENT receive interrupt	523	group0-26	group3-47
RSENT6				
INTRSENT6SI	RSENT status interrupt	524	—	—
INTRSENT6RI	RSENT receive interrupt	525	group0-27	group3-48
RSENT7				
INTRSENT7SI	RSENT status interrupt	526	—	—
INTRSENT7RI	RSENT receive interrupt	527	group0-28	group3-49

This module has no error notifications.

26.1.5 Reset Sources

RSENTn reset sources are listed in the following table. RSENTn is initialized by these sources.

Table 26.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RSENTn	All registers	√	√	√	√	√	√	—

Note: For more details of reset explanation, refer to **Section 9, Reset Controller**.

26.1.6 External Input/Output Signals

External input/output signals of RSENT are listed below.

Table 26.7 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
RSENT0			
rsent_rx	I	RSENT Data Input	RSENT0RX
rsent_spc	O	RSENT SPC Extension Output	RSENT0SPCO
RSENT1			
rsent_rx	I	RSENT Data Input	RSENT1RX
rsent_spc	O	RSENT SPC Extension Output	RSENT1SPCO
RSENT2			
rsent_rx	I	RSENT Data Input	RSENT2RX
rsent_spc	O	RSENT SPC Extension Output	RSENT2SPCO
RSENT3			
rsent_rx	I	RSENT Data Input	RSENT3RX
rsent_spc	O	RSENT SPC Extension Output	RSENT3SPCO
RSENT4			
rsent_rx	I	RSENT Data Input	RSENT4RX
rsent_spc	O	RSENT SPC Extension Output	RSENT4SPCO
RSENT5			
rsent_rx	I	RSENT Data Input	RSENT5RX
rsent_spc	O	RSENT SPC Extension Output	RSENT5SPCO
RSENT6			
rsent_rx	I	RSENT Data Input	RSENT6RX
rsent_spc	O	RSENT SPC Extension Output	RSENT6SPCO
RSENT7			
rsent_rx	I	RSENT Data Input	RSENT7RX
rsent_spc	O	RSENT SPC Extension Output	RSENT7SPCO

26.1.7 Combinations of Pins and Ports

Combinations of RSENT pins and ports are listed in the following table.

Table 26.8 Combinations of Pins and Ports

Function	Pin Name	Port Name
RSENT0	RSENT0RX	P3_4/P6_14*1/P10_7/P20_2
	RSENT0SPCO	P2_4/P6_14*1
RSENT1	RSENT1RX	P3_5/P10_8/P17_4*1/P20_3
	RSENT1SPCO	P17_4*1/P20_9/P20_13
RSENT2	RSENT2RX	P4_13*1/P10_13*1/P20_12*1
	RSENT2SPCO	P4_13*1/P10_13*1/P20_4/P20_12*1
RSENT3	RSENT3RX	P4_12*1/P10_14*1/P20_13
	RSENT3SPCO	P4_12*1/P10_14*1/P20_0
RSENT4	RSENT4RX	P3_2*1/P6_13*1/P17_0
	RSENT4SPCO	P3_2*1/P4_6/P6_13*1/P20_1
RSENT5	RSENT5RX	P3_3*1/P17_1
	RSENT5SPCO	P3_3*1/P4_7
RSENT6	RSENT6RX	P21_1*1/P24_4*1
	RSENT6SPCO	P21_1*1/P24_4*1/P24_6
RSENT7	RSENT7RX	P21_0*1/P24_5*1
	RSENT7SPCO	P21_0*1/P24_5*1/P24_7

Note 1. These ports are share pins. For more details, refer to **Section 26.4.4.1, Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions**.

26.2 Overview

26.2.1 Functional Overview

The RSENT interface supports the following standard specification (SAE J2716 version APR2016) functions:

- Triple speed expansion Tick Time: Clock cycle (1 µs to 90 µs)
- Variable data transmission rate
  - Up to 74.9 kbps (based on 8 nibble data at 3 µs clock rate)
  - Up to 224.7 kbps (based on 8 nibble data at 1 µs clock rate)
- Unidirectional communication: Between the sensor and MCU
- Bidirectional communication: Between the sensor and MCU (supported in SPC mode)
- Single edge data transmission: Coded by the temporal distance of two serially-detected falling edges on a data line
- Transmission frame with up to 8 data nibbles + status and communication nibbles.
- Data transmission protected with CRC is available.
  - CRC data can be read with the RSENTnSRXD.SCRC bits and RSENTnFRXD.FCRC bits.
- Calibration phrase in each data frame (RSENTnCPL.CPLV bits)
- 1-wire interface (sent_rx and sent_spc share a single terminal.)
- Multiple sensors can connect to the RSENT channel that has the standard expansion function. Received data from sensors is detected by software or sDMA.
- The timestamp function: Master or slave can be selected for the RSENT module (RSENTnTSPC.TMS bit).
  Each RSENT macro consists of one RSENT channel.
  Each timestamp counter of macros can run independently, or in order to synchronize the timestamp across multiple channels, one instance can be set as master and reset the configured consecutive timestamp counters.
  Depending on the total number of instances, more than one master-slave (s) pairs can be configured.
- RSENT has two interrupt request (status interrupt and receive interrupt)
- RSENT has one sDMA/DTS request (receive interrupt)
- The RSENT circuit consists of the following functions:
  - Data receive part
  - Clock recovery
  - Register group

26.2.2 Block Diagram

The following figure shows a block diagram of the RSENT module.

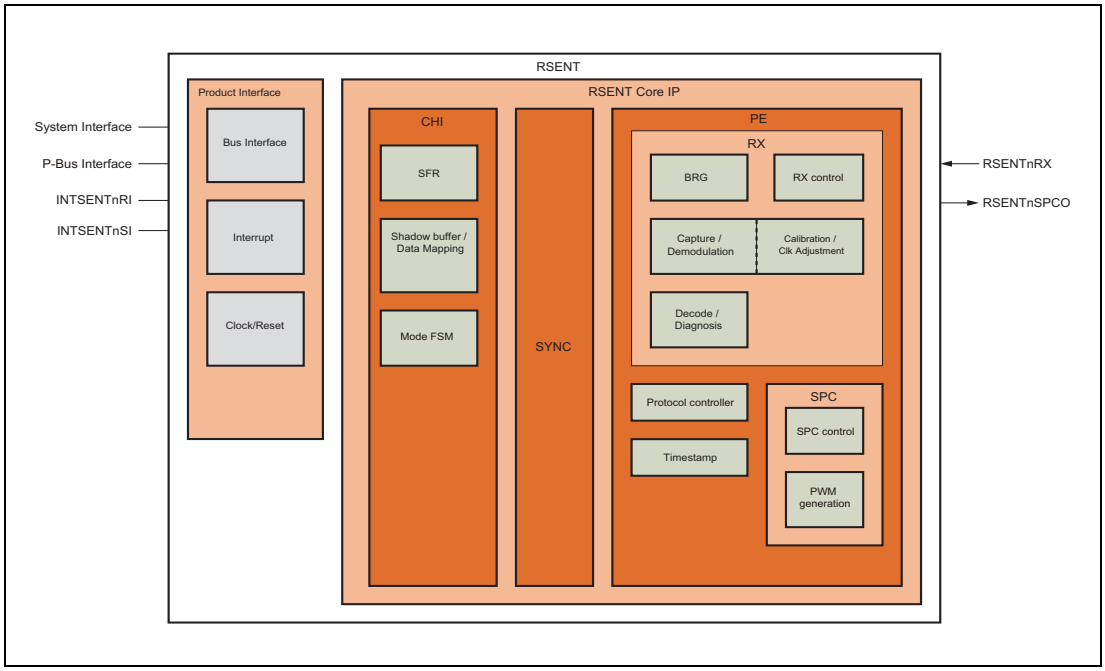


Figure 26.1 Block Diagram of RSENT

26.3 Registers

26.3.1 List of Registers

RSENTn registers are listed in the following table.

For details about <RSENTn_base>, see **Section 26.1.2, Register Base Addresses**.

Table 26.9 List of Registers

Module Name	Register Name	Abbreviation	Address	Access Protection	
				PBG	Other
RSENTn	RSENT Timestamp Register	RSENTnTSPC	<RSENTn_base> + 0000_H	1*1	—
	RSENT Timestamp Counter	RSENTnTSC	<RSENTn_base> + 0004_H	1*1	—
	RSENT Communication Configuration Register	RSENTnCC	<RSENTn_base> + 0010_H	1*1	—
	RSENT Baud Rate Prescaler Register	RSENTnBRP	<RSENTn_base> + 0014_H	1*1	—
	RSENT Interrupt/DMA Enable Register	RSENTnIDE	<RSENTn_base> + 0018_H	1*1	—
	RSENT Mode Control Register	RSENTnMDC	<RSENTn_base> + $001C_H$	1*1	—
	RSENT SPC Transmission Register	RSENTnSPCT	<RSENTn_base> + 0020_H	1*1	—
	RSENT Mode Status Register	RSENTnMST	<RSENTn_base> + 0024_H	1*1	—
	RSENT Communication Status Register	RSENTnCS	<RSENTn_base> + 0028_H	1*1	—
	RSENT Communication Status Clear Register	RSENTnCSC	<RSENTn_base> + $002C_H$	1*1	—
	RSENT Slow Channel Receive Timestamp Register	RSENTnSRTS	<RSENTn_base> + 0030_H	1*1	—
	RSENT Slow Channel Receive Data Register	RSENTnSRXD	<RSENTn_base> + 0034_H	1*1	—
	RSENT Calibration Pulse Length Register	RSENTnCPL	<RSENTn_base> + 0038_H	1*1	—
	RSENT Message Length Register	RSENTnML	<RSENTn_base> + $003C_H$	1*1	—
	RSENT Fast Channel Receive Timestamp Register	RSENTnFRTS	<RSENTn_base> + 0040_H	1*1	—
	RSENT Fast Channel Receive Data Register	RSENTnFRXD	<RSENTn_base> + 0044_H	1*1	—
	RSENT Calibration Pulse Length Mirror Register	RSENTnCPLM	<RSENTn_base> + 0050_H	1*1	—
	RSENT Message Length Mirror Register	RSENTnMLM	<RSENTn_base> + 0054_H	1*1	—
	RSENT Fast Channel Receive Timestamp Mirror Register	RSENTnFRTSM	<RSENTn_base> + 0058_H	1*1	—
	RSENT Expanded Fast Channel Receive Data Register 0	RSENTnEFRD0	<RSENTn_base> + $005C_H$	1*1	—
	RSENT Expanded Fast Channel Receive Data Register 1	RSENTnEFRD1	<RSENTn_base> + 0060_H	1*1	—
	RSENT Timestamp Mode Selection Register	RSENTTSSEL	$FFD33E00_H$	PBG 31 #13	—

Note 1. n = 0: PBG31 #5
n = 1: PBG31 #6
n = 2: PBG31 #7
n = 3: PBG31 #8
n = 4: PBG31 #9
n = 5: PBG31 #10
n = 6: PBG31 #11
n = 7: PBG31 #12

26.3.2 RSENTnTSPC — RSENT Timestamp Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0000_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TTM[6:0]							—	TTPV[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.10 RSENTnTSPC Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TMS	Timestamp Mode Selection 0: Master mode 1: Slave mode
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	TTM[6:0]	Timestamp Tick Multiplier 0000000_B: 1 0000001_B: 2 0000010_B: 3 : 1111111_B: 128
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	TTPV[6:0]	Timestamp Tick Prescaler Value 0000000_B: 1 0000001_B: 2 0000010_B: 3 : 1111111_B: 128

RSENTnTSPC.TMS (Timestamp Mode Selection)

This bit defines the timestamp counter synchronization mode.

For information about the timestamp clock settings, see **Section 26.4.2.1, Timestamp**.

When this bit is set to 0, the timestamp counter operates in master mode.

When writing $0000\ 0000_H$ to RSENTnTSC, the timestamp counter is cleared. In addition all RSENT timestamp counters operating as slave of RSENT are also cleared. For master-slave interconnection, see **Section 26.4.2.1(2), Timestamp Counter Operation**.

When this bit is set to 1, the timestamp counter operates in slave mode.

The timestamp counter is only cleared when writing $0000\ 0000_H$ to the timestamp counter of RSENT module that operates in master mode.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The RSENT module operating in slave mode should have the same timestamp counter prescaler settings as the RSENT module that operates in master mode.

The CPU should not set this bit to 1 for RSENT module that operates in master mode.

RSENTnTSPC.TTM (Timestamp Tick Multiplier)

These bits define the multiplication value of the 1-µs time tick used for the timestamp counter.

For timestamp clock configuration, see **Section 26.4.2.1, Timestamp**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnTSPC.TTPV (Timestamp Tick Prescaler Value)

These bits define the prescaler value to generate a 1-µs clock tick.

For timestamp clock configuration, see **Section 26.4.2.1, Timestamp**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should configure this value in such a way that, based on the supplied communication clock, a 1-µs clock tick is generated.

26.3.3 RSENTnTSC — RSENT Timestamp Counter Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0004_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.11 RSENTnTSC Register Contents

Bit Position	Bit Name	Function
31 to 0	TS[31:0]	Timestamp counter value

RSENTnTSC.TS (Timestamp)

These bits indicate the current timestamp counter value.

The CPU can only write to these bits values other than $0000\ 0000_H$ if RSENT is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

When the timestamp counter is configured to operate in slave mode (RSENTnTSPC.TMS = 1), writing to this register has no effect when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011_B or 101_B).

The timestamp counter is incremented on every timestamp counter tick (as configured in the RSENTnTSPC.TTPV and RSENTnTSPC.TTM bits) when the RSENT module is in either of the OPERATION IDLE or OPERATION ACTIVE modes (the RSENTnMST.OMS bits are either 011_B or 101_B).

When the timestamp counter is configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU writes $0000\ 0000_H$ to these bits and RSENTnTSC.TS is set to $0000\ 0000_H$.

When the slave mode setting is made for the timestamp counter of channel n (RSENTnTSPC.TMS = 1), writing to the timestamp counter of the channel that is the master for channel n leads to the RSENTnTSC.TS bits being set to $0000\ 0000_H$.

For timestamp mode selection, see **Section 26.4.2.1, Timestamp**.

26.3.4 RSENTnCC — RSENT Communication Configuration Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0010_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SOPC	FCM	SCCD	FCCD	DCF	SMF[1:0]		PPTC	PPC	NDN[2:0]			SPCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.12 RSENTnCC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SOPC	SPC Output Polarity Control 0: SPC pulse active high 1: SPC pulse active low
11	FCM	Frame Check Method 0: Check against next calibration pulse 1: Check against previous calibration pulse
10	SCCD	Slow Channel CRC Check 0: Slow channel CRC check enabled 1: Slow channel CRC check disabled
9	FCCD	Fast Channel CRC Check 0: Fast channel CRC check enabled 1: Fast channel CRC check disabled
8	DCF	Data nibble CRC Format 0: SAE J2716 2010/2016 format 1: pre SAE J2716 2010 format
7, 6	SMF[1:0]	Serial Message Format 00_B: No serial message extraction 01_B: Short serial message format 10_B: Enhanced serial message format 11_B: Setting prohibited
5	PPTC	Pause Pulse Type Configuration 0: Pause pulse for variable message length 1: Pause pulse for fixed message length
4	PPC	Pause Pulse Configuration 0: Pause pulse absent 1: Pause pulse present
3 to 1	NDN[2:0]	Number of Data Nibbles 000_B: 1 data nibble 001_B: 2 data nibbles 010_B: 3 data nibbles 011_B: 4 data nibbles 100_B: 5 data nibbles 101_B: 6 data nibbles 110_B: 7 data nibbles 111_B: 8 data nibbles

Table 26.12 RSENTnCC Register Contents (2/2)

Bit Position	Bit Name	Function
0	SPCE	SPC Mode Enable 0: SPC mode disabled 1: SPC mode enabled

RSENTnCC.SOPC (SPC Output Polarity Control)

When this bit is set to 0, the SPC pulse is sent as an active high signal. The default output value is low level.

When this bit is set to 1, the SPC pulse is sent as an active low signal. The default output value is high level.

For SPC operation, see also **Section 26.4.4, SPC Function**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

NOTE

Any change to this bit from the default value becomes effective on the output value when entering the OPERATION_ACTIVE mode (RSENTnMST.OMS is 101_B). When entering RESET mode (RSENTnMST.OMS is 000_B), the output level is set to the default value (low level).

RSENTnCC.FCM (Frame Check Method)

When this bit is set to 0, the current calibration pulse is compared to the next received calibration pulse.

The buffer update mechanism is operating according to the preferred option as described in SAE J2716 2016.

When this bit is set to 1, the current calibration pulse is compared to the previously received calibration pulse.

The buffer update mechanism is operating according to the second option as described in SAE J2716 2016 which should be only used if extra latency to process the second calibration pulse can not be tolerated.

For buffer update timings, see also **Section 26.4.3.2(3), Fast Channel Message Reception**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SCCD (Slow Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the slow channel is disabled. In this case, messages are stored in the slow channel message reception buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.SCS bit is not set.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.FCCD (Fast Channel CRC Check Disable)

When this bit is set to 1, the CRC check for the fast channel is disabled. In this case, messages are stored in the fast channel message reception buffer with the received CRC.

When this bit is set to 1, the RSENTnCS.FCS bit is not set.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.DCF (Data Nibble CRC Format)

This bit selects between the SAE J2716 2016 data nibble CRC format and the legacy format.

When this bit is set to 0 the recommended CRC implementation according to SAE J2716 2016 *Section 5.4.2.2* is selected.

When this bit is set to 1 the legacy CRC implementation according to SAE J2716 2008 (refer to SAE J2716 2016 *Section 5.4.2.1*) is selected.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (RSENTnMST.OMS = 001_B)

RSENTnCC.SMF (Serial Message Format)

These bits define the serial message format expected to be received for automatic extraction.

When these bits are set to 00_B, no serial message is extracted and the status and communications nibbles are stored in the RSENTnSRXD register.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU shall set these bits to 00_B when RSENTnCC.SPCE is set to 1 and more than one sensor is connected to the RSENT module.

RSENTnCC.PPTC (Pause Pulse Type Configuration)

This bit defines the pause pulse type.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

The CPU should not set this bit to 1 when the RSENTnCC.PPC bit is set to 0.

RSENTnCC.PPC (Pause Pulse Configuration)

This bit defines the presence or absence of the pause pulse.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.NDN (Number of Data Nibbles)

These bits define the number of data nibbles included in an RSENT message.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnCC.SPCE (SPC Mode Enable)

This bit enables the SPC mode.

For details about SPC mode operation, see also **Section 26.4.4, SPC Function**.

The CPU can only write to this bit if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

26.3.5 RSENTnBRP — RSENT Baud Rate Prescaler Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0014_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TTF[3:0]				—	TTI[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SCDV[6:0]							—	—	—	SCMV[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 26.13 RSENTnBRP Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	TTF[3:0]	Time Tick Decimal Fraction 0000_B: 0.0 µs 0001_B: 0.1 µs 0010_B: 0.2 µs : 1000_B: 0.8 µs 1001_B: 0.9 µs Other than above: Setting prohibited
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	TTI[6:0]	Time Tick Integer 0000000_B: 1 µs 0000001_B: 2 µs 0000010_B: 3 µs : 1011000_B: 89 µs 1011001_B: 90 µs Other than above: Setting prohibited
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	SCDV[6:0]	Sample Clock Division Value 0000000_B: 1 0000001_B: 2 0000010_B: 3 : 1111110_B: 127 1111111_B: 128
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 26.13 RSENTnBRP Register Contents (2/2)

Bit Position	Bit Name	Function
4 to 0	SCMV[4:0]	Sample Clock Multiplication Value 00000_B: 1 00001_B: 2 00010_B: 3 : 11110_B: 31 11111_B: 32

RSENTnBRP.TTF (Time Tick Decimal Fraction)

These bits define the decimal part of the tick length in 0.1-µs granularity.

For tick length configuration, see **Section 26.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.TTI (Time Tick Integer)

These bits define the integer part of the tick length.

For tick length configuration, see **Section 26.4.2.2(2), RX and SPC Tick Settings**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCDV (Sample Clock Division Value)

These bits define the division value for the sample clock generation logic.

For RSENTnBRP settings, see **Section 26.4.2.2(1), RX BRP Setting**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

RSENTnBRP.SCMV (Sample Clock Multiplication Value)

These bits define the multiplication value for the sample clock generation logic.

For RSENTnBRP settings, see **Section 26.4.2.2(1), RX BRP Setting**.

The CPU can only write to these bits if the RSENT module is in the CONFIGURATION mode (the RSENTnMST.OMS bits are 001_B).

26.3.6 RSENTnIDE — RSENT Interrupt/DMA Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0018_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEIE	SMIE	SCIE	NRIE	CVIE	CLIE	FNIE	FEIE	FMIE	FCIE	FRIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.14 RSENTnIDE Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	SEIE	Slow Channel Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
9	SMIE	Slow Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
8	SCIE	Slow Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
7	NRIE	No Response Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
6	CVIE	Calibration Pulse Length Variation Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
5	CLIE	Calibration Pulse Length Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
4	FNIE	Fast Channel Nibble Count Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
3	FEIE	Fast Channel Nibble Encoding Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
2	FMIE	Fast Channel Message Lost Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
1	FCIE	Fast Channel CRC Error Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled
0	FRIE	Fast Channel Receive Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled

RSENTnIDE.SEIE (Slow Channel Encoding Error Interrupt Enable)

This bit enables the generation of the slow channel encoding error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SMIE (Slow Channel Message Lost Interrupt Enable)

This bit enables the generation of the slow channel message lost interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.SCIE (Slow Channel CRC Error Interrupt Enable)

This bit enables the generation of the slow channel CRC error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.NRIE (No Response Error Interrupt Enable)

This bit enables the generation of the no response error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

The CPU should not set this bit when the SPC mode is disabled (RSENTnCC.SPCE set to 0).

RSENTnIDE.CVIE (Calibration Pulse Length Variation Error Interrupt Enable)

This bit enables the generation of the calibration pulse length variation error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.CLIE (Calibration Pulse Length Error Interrupt Enable)

This bit enables the generation of the calibration pulse length error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FNIE (Fast Channel Nibble Count Error Interrupt Enable)

This bit enables the generation of the fast channel nibble count error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FEIE (Fast Channel Nibble Encoding Error Interrupt Enable)

This bit enables the generation of the fast channel nibble encoding error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FMIE (Fast Channel Message Lost Interrupt Enable)

This bit enables the generation of the fast channel message lost interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FCIE (Fast Channel CRC Error Interrupt Enable)

This bit enables the generation of the fast channel CRC error interrupt.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

RSENTnIDE.FRIE (Fast Channel Receive Interrupt Enable)

This bit enables the generation of the fast channel receive interrupt.

The fast channel receive interrupt can be also used to notify a DMA request.

The CPU can not write to this bit if the RSENT module is in the RESET mode (the RSENTnMST.OMS bits are 000_B).

26.3.7 RSENTnMDC — RSENT Mode Control Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + $001C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 26.15 RSENTnMDC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	OMC[2:0]	Operation Mode Control 000_B: RESET 001_B: CONFIGURATION 011_B: OPERATION IDLE 101_B: OPERATION ACTIVE Other than above: Setting prohibited

RSENTnMDC.OMC (Operation Mode Control)

These bits are used to control the operation mode of the RSENT module.

- 000_B: RESET

  In RESET mode, the mode can only be changed to CONFIGURATION mode.

- 001_B: CONFIGURATION

  In CONFIGURATION mode, the mode can only be changed to RESET mode or OPERATION ACTIVE mode.

- 011_B: OPERATION IDLE

  In OPERATION IDLE mode, the mode can be changed to OPERATION ACTIVE mode, CONFIGURATION mode, or RESET mode.

- 101_B: OPERATION ACTIVE

  In OPERATION ACTIVE mode, the mode can be changed to OPERATION IDLE mode, CONFIGURATION mode, or RESET mode. However, it is recommended to process to the OPERATION IDLE mode first.

  For the recommended methods to change between operation modes, see **Section 26.4.3.1, Changing Operation Modes**.

- Other than above: Setting prohibited

  The CPU should not write any other value than listed above into this register.
  The CPU should follow the mode change flows as shown in **Section 26.4.3.1, Changing Operation Modes**.

26.3.8 RSENTnSPCT — RSENT SPC Transmission Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + 0020_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TLL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.16 RSENTnSPCT Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	TLL[6:0]	Length of the Trigger Low Phase in Ticks 0000000_B: 1 tick 0000001_B: 2 ticks 0000010_B: 3 ticks : 1111110_B: 127 ticks 1111111_B: 128 ticks

RSENTnSPCT.TLL (Trigger Low Length)

These bits define the length of the SPC trigger pulse.

When the CPU writes to these bits, an SPC trigger pulse with the configured length is sent starting from the next SPC trigger tick. In case RSENTnCS.NRS is set by the RSENT module following a write to these bits, no SPC trigger pulse is sent.

For details about SPC communication, see **Section 26.4.4, SPC Function**.

The CPU can only write to these bits if the RSENT module is in the OPERATION ACTIVE mode (the RSENTnMST.OMS bits are 101_B) and SPC communication is enabled (RSENTnCC.SPCE is 1_B).

It is important to note that two consecutive write access might not cause a no response error as the previous request might not have started yet.

After writing to this register, the CPU should wait for at least one SPC trigger tick before writing again to this register.

26.3.9 RSENTnMST — RSENT Mode Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0024_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	OMS[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.17 RSENTnMST Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	OMS[2:0]	Operation Mode 000_B: RESET 001_B: CONFIGURATION 011_B: OPERATION IDLE 101_B: OPERATION ACTIVE Other than above: Reserved

RSENTnMST.OMS (Operation Mode Status)

These bits indicate the current operation mode.

These bits are read only.

These bits are updated after a mode change request is made in the RSENTnMDC.OMC register.

- 000_B: RESET mode

  When in RESET mode, all registers are set to their reset values and write access to all registers except the RSENTnMDC register is disabled.
  When in RESET mode, RSENT communication is disabled.

- 001_B: CONFIGURATION mode

  When in CONFIGURATION mode, write access to the timestamp registers (RSENTnTSPC and RSENTnTSC register), configuration registers (RSENTnCC and RSENTnBRP register), RSENTnIDE register, and mode control register (RSENTnMDC.OMC) is enabled.
  When in CONFIGURATION mode, RSENT communication is disabled.
  When entering CONFIGURATION mode, all status registers and receive buffer registers are set to their reset values.

- 011_B: OPERATION IDLE mode

  In OPERATION IDLE mode, no reception or SPC trigger transmission is possible.
  When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

- 101_B: OPERATION ACTIVE mode

  In OPERATION ACTIVE mode, reception and SPC trigger transmission are possible.

- Other than above: Reserved

26.3.10 RSENTnCS — RSENT Communication Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0028_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SES	SMS	SCS	NRS	CVS	CLS	FNS	FES	FMS	FCS	FRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.18 RSENTnCS Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	SES	Slow channel Encoding Error Status 0: Not detected 1: Detected
9	SMS	Slow Channel Message Lost Status 0: Not detected 1: Detected
8	SCS	Slow Channel CRC Error Status 0: Not detected 1: Detected
7	NRS	No Response Error Status 0: Not detected 1: Detected
6	CVS	Calibration Pulse Length Variation Error Status 0: Not detected 1: Detected
5	CLS	Calibration Pulse Length Error Status 0: Not detected 1: Detected
4	FNS	Fast Channel Nibble Count Error Status 0: Not detected 1: Detected
3	FES	Fast Channel Nibble Encoding Error Status 0: Not detected 1: Detected
2	FMS	Fast Channel Message Lost Status 0: Not detected 1: Detected
1	FCS	Fast Channel CRC Error Status 0: Not detected 1: Detected
0	FRS	Fast Channel Receive Status 0: Not detected 1: Detected

RSENTnCS.SES (Slow Channel Encoding Error Status)

This bit represents the slow channel encoding error status.

This bit is read only.

In the short serial message format (RSENTnCC.SMF = 01_B), this bit is set when the sequence on serial start bit (bit #3) is different from "1000 0000 0000 0000_B" (a single 1 and 15 0s).

In the enhanced serial message format (RSENTnCC.SMF = 10_B), this bit is set when following the reception of a start sequence ("0111 1110_B") on the serial data bit 3, bit 13 or bit 18 are not received as "0".

When this bit is set in the short serial message format the received communication and status nibble is used to assemble a serial message.

When this bit is set in the enhanced serial message format the RSENT module checks the presence of a new start sequence at the same time and uses the received communication and status nibble to assemble a serial message.

This bit is cleared when writing 1'b1 to RSENTnCSC.SEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SMS (Slow Channel Message Lost Status)

This bit represents the slow channel message lost status.

This bit is read only.

This bit is set when there is an attempt to update the slow channel message reception buffer, but the previous message has not been read yet.

This bit is cleared when writing 1 to RSENTnCSC.SMC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.SCS (Slow Channel CRC Error Status)

This bit represents the slow channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the slow channel and the slow channel CRC detection is enabled (RSENTnCC.SCCD is set to 0).

This bit is cleared when writing 1 to RSENTnCSC.SCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.NRS (No Response Error Status)

This bit represents the no response error status.

This bit is read only.

This bit is set when

- The CPU writes to the RSENTnSPCT.TLL bits and

- SPC mode enabled (RSENTnCC.SPCE set to 1) and
- No complete response was received from the sensor for the previous SPC trigger.

This bit is set after 4 Register Access Clocks + 5 RSENT Communication Clocks (Maximum time) from the CPU writes to the RSENTnSPCT.TLL.

This bit is cleared when writing 1 to RSENTnCSC.NRC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001 (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CVS (Calibration Pulse Length Variation Error Status)

This bit represents the calibration pulse length variation error status.

This bit is read only.

When RSENTnCC.PPTC is 0, then this bit is set when two successive calibration pulses differ by more than 1.5625%.

When RSENTnCC.PPTC is 1, this bit is never set. In this mode (pause pulse with fixed message length), the CPU needs to check the variation of the ratio of calibration pulse to message length by reading the RSENTnCPL and RSENTnML registers.

This bit is cleared when writing 1 to RSENTnCSC.CVC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.CLS (Calibration Pulse Length Error Status)

This bit represents the calibration pulse length error status.

This bit is read only.

This bit is set when the measured calibration pulse length is less than 42 clock ticks or more than 70 clock ticks (deviation of 25 % from nominal length).

This bit is cleared when writing 1 to RSENTnCSC.CLC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FNS (Fast Channel Nibble Count Error Status)

This bit represents the fast channel nibble count error status.

This bit is read only.

This bit is set when there is an unexpected number of falling edges between two calibration pulses.

This bit is cleared when writing 1 to RSENTnCSC.FNC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FES (Fast Channel Nibble Encoding Error Status)

This bit represents the fast channel nibble encoding error status.

This bit is read only.

This bit is set when on the fast channel a measured nibble period is less than 12 clock ticks or more than 27 clock ticks.

This bit is cleared when writing 1 to RSENTnCSC.FEC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FMS (Fast Channel Message Lost Status)

This bit represents the fast channel message lost status.

This bit is read only.

This bit is set when the fast channel message reception buffer is updated, but the previous messages in the foreground and background buffer have not been read yet.

This bit is cleared when writing 1 to RSENTnCSC.FMC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FCS (Fast Channel CRC Error Status)

This bit represents the fast channel CRC error status.

This bit is read only.

This bit is set when a CRC error is detected on the fast channel and the fast channel CRC detection is enabled (RSENTnCC.FCCD is set to 0).

This bit is cleared when writing 1 to RSENTnCSC.FCC.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

RSENTnCS.FRS (Fast Channel Receive Status)

This bit represents the fast channel receive status.

This bit is read only.

This bit is set when the fast channel message reception buffer was updated.

This bit is cleared when the CPU reads the RSENTnFRXD or RSENTnEFRD1 register.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

If the set condition occurs simultaneously with the clear condition, the bit is set.

26.3.11 RSENTnCSC — RSENT Communication Status Clear Register

Access: This register can be read or written in 32-bit units.

Address: <RSENTn_base> + $002C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEC	SMC	SCC	NRC	CVC	CLC	FNC	FEC	FMC	FCC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 26.19 RSENTnCSC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	SEC	Slow Channel Encoding Error Clear 0: — 1: Clear
9	SMC	Slow Channel Message Lost Clear 0:— 1: Clear
8	SCC	Slow Channel CRC Error Clear 0: — 1: Clear
7	NRC	No Response Error Clear 0: — 1: Clear
6	CVC	Calibration Pulse Length Variation Error Clear 0: — 1: Clear
5	CLC	Calibration Pulse Length Error Clear 0: — 1: Clear
4	FNC	Fast Channel Nibble Count Error Clear 0: — 1: Clear
3	FEC	Fast Channel Nibble Encoding Error Clear 0: — 1: Clear
2	FMC	Fast Channel Message Lost Clear 0: — 1: Clear
1	FCC	Fast Channel CRC Error Clear 0: — 1: Clear
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RSENTnCSC.SEC (Slow Channel Encoding Error Clear)

Writing 1 sets RSENTnCS.SES to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.SMC (Slow Channel Message Lost Clear)

Writing 1 sets RSENTnCS.SMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.SCC (Slow Channel CRC Error Clear)

Writing 1 sets RSENTnCS.SCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.NRC (No Response Error Clear)

Writing 1 sets RSENTnCS.NRS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.CVC (Calibration Pulse Length Variation Error Clear)

Writing 1 sets RSENTnCS.CVS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.CLC (Calibration Pulse Length Error Clear)

Writing 1 sets RSENTnCS.CLS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FNC (Fast Channel Nibble Count Error Clear)

Writing 1 sets RSENTnCS.FNS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FEC (Fast Channel Nibble Encoding Error Clear)

Writing 1 sets RSENTnCS.FES to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FMC (Fast Channel Message Lost Clear)

Writing 1 sets RSENTnCS.FMS to 0.

Writing 0 has no effect.

This bit is always read as 0.

RSENTnCSC.FCC (Fast Channel CRC Error Clear)

Writing 1 sets RSENTnCS.FCS to 0.

Writing 0 has no effect.

This bit is always read as 0.

26.3.12 RSENTnSRTS — RSENT Slow Channel Receive Timestamp Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0030_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.20 RSENTnSRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	STS[31:0]	Slow Channel Receive Timestamp

RSENTnSRTS.STS (Slow Channel Receive Timestamp)

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated with the timestamp counter value of the last frame provided to the slow channel message.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.13 RSENTnSRXD — RSENT Slow Channel Receive Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0034_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SND	—	SCRC[5:0]						—	—	—	SMGC	IDD[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IDD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.21 RSENTnSRXD Register Contents

Bit Position	Bit Name	Function
31	SND	Slow Channel New Data 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	Reserved	When read, the value after reset is returned.
29 to 24	SCRC[5:0]	Slow Channel CRC Data
23 to 21	Reserved	When read, the value after reset is returned.
20	SMGC	Slow Channel Configuration Bit Data
19 to 0	IDD[19:0]	Slow Channel Data / ID Information

RSENTnSRXD.SND (Slow Channel New Data)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel message reception buffer is updated.

This bit is cleared automatically whenever it is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SCRC (Slow Channel CRC)

These bits are representing the slow channel CRC data.

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.SMGC (Slow Channel Configuration Bit)

This bit represents the slow channel configuration bit.

This bit is read only.

This bit is updated when the slow channel message reception buffer is updated.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnSRXD.IDD (ID/ Data)

These bits are representing the slow channel data and ID information.

The alignment within this register depends on the message format. For details, see **Section 26.4.3.2(5), Slow Channel Message Reception**.

These bits are read only.

These bits are updated when the slow channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.14 RSENTnCPL — RSENT Calibration Pulse Length Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0038_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLV [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.22 RSENTnCPL Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16 to 0	CPLV[16:0]	Calibration Pulse Length Value of Received Message

RSENTnCPL.CPLV (Calibration Pulse Length Value)

These bits are used by the CPU to calculate the ratio of two consecutive calibration pulses or the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

In modes other than pause pulse with fixed message length (RSENTnCC.PPTC = 1) or SPC mode (RSENTnCC.SPCE = 1), these bits are invalid and should not be used.

These bits are updated with the measured calibration pulse length in sample clock ticks when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.15 RSENTnML — RSENT Message Length Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + $003C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.23 RSENTnML Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 0	MLV[20:0]	Message Length Value of Received Message

RSENTnML.MLV (Message Length Value)

These bits are used by the CPU to calculate the ratio of the calibration pulse to message length in pause pulse with fixed message length mode for message diagnostics.

These bits are read only.

In modes other than pause pulse with fixed message length, these bits are invalid and should not be used.

These bits are updated with the measured message length in sample clock ticks when the fast channel receive message buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.16 RSENTnFRTS — RSENT Fast Channel Receive Timestamp Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0040_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTS[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.24 RSENTnFRTS Register Contents

Bit Position	Bit Name	Function
31 to 0	FTS[31:0]	Fast Channel Receive Timestamp

RSENTnFRTS.FTS (Fast Channel Receive Timestamp)

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.17 RSENTnFRXD — RSENT Fast Channel Receive Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0044_H

Value after reset: $0000\ 0000_H$ This register is initialized by various types of reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SNDM	FND	FCCN[1:0]		FCRC[3:0]				ND[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.25 RSENTnFRXD Register Contents

Bit Position	Bit Name	Function
31	SNDM	Slow Channel New Data Mirror 0: Slow channel frame data is not updated since last read. 1: Slow channel frame data is updated since last read.
30	FND	Fast Channel New Data 0: Fast channel frame data is not updated since last read. 1: Fast channel frame data is updated since last read.
29, 28	FCCN[1:0]	Fast Channel Status and Communications Nibble[1:0]
27 to 24	FCRC[3:0]	Fast Channel CRC Data
23 to 0	ND[23:0]	Fast Channel Nibble Data

RSENTnFRXD.SNDM (Slow Channel New Data Mirror)

This bit indicates that the slow channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the slow channel message reception buffer is updated.

This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FND (Fast Channel New Data)

This bit indicates that the fast channel message reception buffer is holding data that has not been read.

This bit is read only.

This bit is set when the fast channel message reception buffer is updated.

This bit is cleared automatically whenever it is read RSENTnFRXD or RSENTnEFRD1.

This bit is cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCCN (Fast Channel Communication Nibble)

These bits are representing the fast channel communication nibble bits [1:0].

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.FCRC (Fast Channel CRC)

These bits are representing the fast channel CRC data.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

RSENTnFRXD.ND (Fast Channel Nibble Data)

These bits are representing the fast channel nibble data.

The alignment of the nibble data depends on nibble data count (RSENTnCC.NDN). For details, see **Section 26.4.3.2(3), Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel message reception buffer is updated.

These bits are cleared when the RSENTnMST.OMS bits are changed to 001_B (CONFIGURATION).

26.3.18 RSENTnCPLM — RSENT Calibration Pulse Length Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPLVM [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CPLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.26 RSENTnCPLM Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16 to 0	CPLVM[16:0]	Calibration pulse length value (Mirror) of received message

Calibration Pulse Length Value Mirror (RSENTnCPLM.CPLVM)

These bits are mirror bits of RSENTnCPL.CPLV.
These bits are read only.
In modes other than pause pulse with fixed message length (RSENTnCC.PPTC = 1) or SPC mode (RSENTnCC.SPCE = 1), these bits are invalid and should not be used.
These bits are updated with the measured calibration pulse length in sample clock ticks when the fast channel receive message buffer is updated.
These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

26.3.19 RSENTnMLM — RSENT Message Length Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0054_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	MLVM[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLVM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.27 RSENTnMLM Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 0	MLVM[20:0]	Message length value (Mirror) of received message

Message Length Value Mirror (RSENTnMLM.MLVM)

These bits are mirror bits of RSENTnML.MLV.
These bits are read only.
These bits are updated when the fast channel receive message buffer is updated.

26.3.20 RSENTnFRTSM — RSENT Fast Channel Receive Timestamp Mirror Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0058_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTSM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTSM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.28 RSENTnFRTSM Register Contents

Bit Position	Bit Name	Function
31 to 0	FTSM[31:0]	Fast channel receive Timestamp (Mirror)

Fast Channel Timestamp Mirror (RSENTnFRTSM.FTSM)

These bits are mirror bits of RSENTnFRTS.FTS.
These bits are read only.
These bits are updated when the fast channel receive message buffer is updated.
These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

26.3.21 RSENTnEFRD0 — RSENT Expanded Fast Channel Receive Data Register 0

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + $005C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SNDM	FND	FCCN[1:0]		FCRC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.29 RSENTnEFRD0 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	SNDM	Slow channel New Data (Mirror) 0: Slow channel frame data not updated since last read. 1: Slow channel frame data updated since last read.
6	FND	Fast channel New Data 0: Fast channel frame data not updated since last read. 1: Fast channel frame data updated since last read.
5, 4	FCCN[1:0]	Fast Channel Communication Nibble bits [1:0]
3 to 0	FCRC[3:0]	Fast channel CRC data

Slow Channel New Data Mirror (RSENTnEFRD0.SNDM)

This bit indicates that the slow channel message buffer is holding data that has not been read.
This bit is read only.
This bit is set when the slow channel receive message buffer is updated.
This bit is cleared automatically whenever the slow channel new data bit (RSENTnSRXD.SND) is read.
This bit is cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel New Data (RSENTnEFRD0.FND)

This bit indicates that the fast channel message buffer is holding data that has not been read.
This bit is read only.
This bit is set when the fast channel receive message buffer is updated.
This bit is cleared automatically whenever it is read RSENTnFRXD or RSENTnEFRD1.
This bit is cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel Communication Nibble (RSENTnEFRD0.FCCN)

These bits represent the fast channel communication nibble bits [1:0].
These bits are read only.
These bits are updated when the fast channel receive message buffer is updated.
These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

Fast Channel CRC (RSENTnEFRD0.FCRC)

These bits represent the fast channel CRC data.
These bits are read only.
These bits are updated when the fast channel receive message buffer is updated.
These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

26.3.22 RSENTnEFRD1 — RSENT Expanded Fast Channel Receive Data Register 1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RSENTn_base> + 0060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ND[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ND[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.30 RSENTnEFRD1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ND[31:0]	Nibble Data Fast channel nibble data

Nibble Data (RSENTnEFRD1.ND)

These bits represent the fast channel nibble data.

The alignment of the nibble data depends on nibble count. For details, refer to **Section 26.4.3.2(3), Fast Channel Message Reception**.

These bits are read only.

These bits are updated when the fast channel receive message buffer is updated.

These bits are cleared when RSENTnMST.OMS is changed to 001_B (CONFIGURATION).

26.3.23 RSENTTSSEL — RSENT Timestamp Mode Selection Register

Access: This register can be read or written in 32-bit units.

Address: $FFD33E00_H$

Value after reset: $0000\ 0000_H$. This register is initialized by any reset source except module reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSSEL7[3:0]				MSSEL6[3:0]				MSSEL5[3:0]				MSSEL4[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSSEL3[3:0]				MSSEL2[3:0]				MSSEL1[3:0]				MSSEL0[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.31 RSENTTSSEL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	MSSEL7[3:0]	RSENT7 Master Selection. This selection is valid only when RSENT7 is in slave mode (when the RSENT7TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT7. 0010_B: RSENT1 is the master of RSENT7. 0011_B: RSENT2 is the master of RSENT7. 0100_B: RSENT3 is the master of RSENT7. 0101_B: RSENT4 is the master of RSENT7. 0110_B: RSENT5 is the master of RSENT7. 0111_B: RSENT6 is the master of RSENT7. 1000_B: No timestamp master. Other than above: Setting prohibited
27 to 24	MSSEL6[3:0]	RSENT6 Master Selection. This selection is valid only when RSENT6 is in slave mode (when the RSENT6TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT6. 0010_B: RSENT1 is the master of RSENT6. 0011_B: RSENT2 is the master of RSENT6. 0100_B: RSENT3 is the master of RSENT6. 0101_B: RSENT4 is the master of RSENT6. 0110_B: RSENT5 is the master of RSENT6. 0111_B: No timestamp master. 1000_B: RSENT7 is the master of RSENT6. Other than above: Setting prohibited
23 to 20	MSSEL5[3:0]	RSENT5 Master Selection. This selection is valid only when RSENT5 is in slave mode (when the RSENT5TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT5. 0010_B: RSENT1 is the master of RSENT5. 0011_B: RSENT2 is the master of RSENT5. 0100_B: RSENT3 is the master of RSENT5. 0101_B: RSENT4 is the master of RSENT5. 0110_B: No timestamp master. 0111_B: RSENT6 is the master of RSENT5. 1000_B: RSENT7 is the master of RSENT5. Other than above: Setting prohibited

Table 26.31 RSENTTSSEL Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	MSSEL4[3:0]	RSENT4 Master Selection. This selection is valid only when RSENT4 is in slave mode (when the RSENT4TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT4. 0010_B: RSENT1 is the master of RSENT4. 0011_B: RSENT2 is the master of RSENT4. 0100_B: RSENT3 is the master of RSENT4. 0101_B: No timestamp master. 0110_B: RSENT5 is the master of RSENT4. 0111_B: RSENT6 is the master of RSENT4. 1000_B: RSENT7 is the master of RSENT4. Other than above: Setting prohibited
15 to 12	MSSEL3[3:0]	RSENT3 Master Selection. This selection is valid only when RSENT3 is in slave mode (when the RSENT3TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT3. 0010_B: RSENT1 is the master of RSENT3. 0011_B: RSENT2 is the master of RSENT3. 0100_B: No timestamp master. 0101_B: RSENT4 is the master of RSENT3. 0110_B: RSENT5 is the master of RSENT3. 0111_B: RSENT6 is the master of RSENT3. 1000_B: RSENT7 is the master of RSENT3. Other than above: Setting prohibited
11 to 8	MSSEL2[3:0]	RSENT2 Master Selection. This selection is valid only when RSENT2 is in slave mode (when the RSENT2TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT2. 0010_B: RSENT1 is the master of RSENT2. 0011_B: No timestamp master. 0100_B: RSENT3 is the master of RSENT2. 0101_B: RSENT4 is the master of RSENT2. 0110_B: RSENT5 is the master of RSENT2. 0111_B: RSENT6 is the master of RSENT2. 1000_B: RSENT7 is the master of RSENT2. Other than above: Setting prohibited
7 to 4	MSSEL1[3:0]	RSENT1 Master Selection. This selection is valid only when RSENT1 is in slave mode (when the RSENT1TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: RSENT0 is the master of RSENT1. 0010_B: No timestamp master. 0011_B: RSENT2 is the master of RSENT1. 0100_B: RSENT3 is the master of RSENT1. 0101_B: RSENT4 is the master of RSENT1. 0110_B: RSENT5 is the master of RSENT1. 0111_B: RSENT6 is the master of RSENT1. 1000_B: RSENT7 is the master of RSENT1. Other than above: Setting prohibited
3 to 0	MSSEL0[3:0]	RSENT0 Master Selection. This selection is valid only when RSENT0 is in slave mode (when the RSENT0TSPC.TMS bit = 1_B) 0000_B: No timestamp master. 0001_B: No timestamp master. 0010_B: RSENT1 is the master of RSENT0. 0011_B: RSENT2 is the master of RSENT0. 0100_B: RSENT3 is the master of RSENT0. 0101_B: RSENT4 is the master of RSENT0. 0110_B: RSENT5 is the master of RSENT0. 0111_B: RSENT6 is the master of RSENT0. 1000_B: RSENT7 is the master of RSENT0. Other than above: Setting prohibited

26.4 Operation

26.4.1 Modes of Operation

The RSENT module can be in one of the following modes:

- RESET mode
- CONFIGURATION mode
- OPERATION IDLE mode
- OPERATION ACTIVE mode

CPU should follow the mode change flow as shown in **Section 26.4.3.1, Changing Operation Modes**.

Figure 26.2 shows the possible transitions between the channel modes:

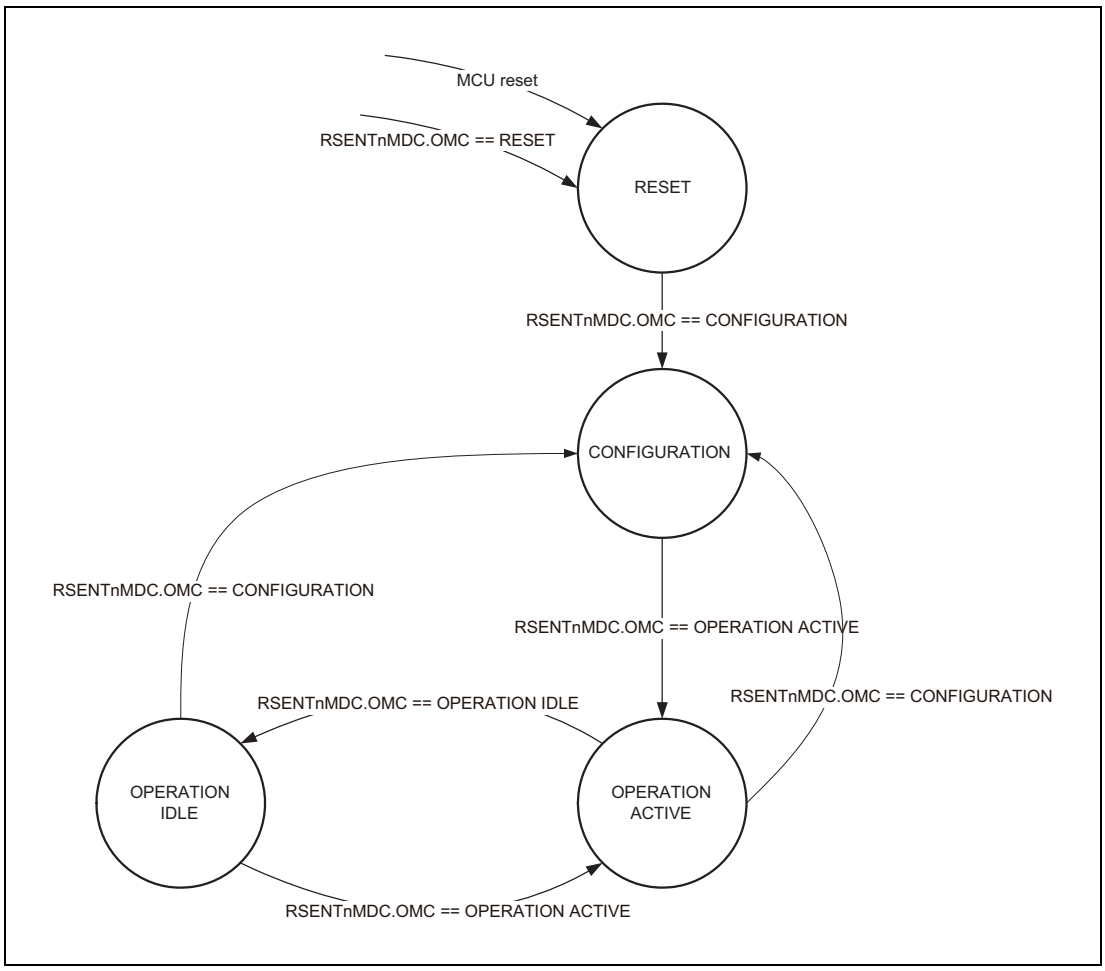


Figure 26.2 Transition between Operation Modes

The current operation mode status can be seen in the RSENTnMST.OMS bits.

26.4.1.1 RESET Mode

This mode is the initial mode that the RSENT module automatically enters after the hardware reset (MCU reset) is cleared. Its purpose is to provide a clean reset of the registers in the RSENT module.

The RESET mode is also entered after the RSENTnMDC.OMC bits have been set to 000_B. In this state, all, configuration, control (except RSENTnMDC.OMC bits and bits in the RSENTTSSEL register), and status registers are set to their reset value. Any on-going transmission or reception process is stopped immediately and the interface pins of the RSENT module are set to their default values.

Read access to all registers is possible in this state. Write access is limited to the RSENTnMDC register and RSENTTSSEL registers.

26.4.1.2 CONFIGURATION Mode

The CONFIGURATION mode is entered after the RSENTnMDC.OMC bits have been set to 001_B.

The interface pins of the RSENT are set to their default values.

Regarding the output polarity setting of RSENTnSPCO pin and the timing that becomes effective, refer to the explanation of “RSENTnCC.SOPC (SPC Output Polarity Control)” in **Section 26.3.4, RSENTnCC — RSENT Communication Configuration Register**.

However, the transition from OPERATION_ACTIVE mode to CONFIGURATION mode is allowed only when the SPC mode is enabled and the communication is not ongoing.

In this state, all status registers (RSENTnCS) and the receive buffer registers (RSENTnSRTS, RSENTnSRXD, RSENTnCPL, RSENTnML, RSENTnFRTS, RSENTnFRXD, RSENTnCPLM, RSENTnMLM, RSENTnFRTSM, RSENTnEFRD0, and RSENTnEFRD1) are set to their default values.

Read access to all registers is possible in this state.

Write access is limited to both timestamp registers (RSENTnTSPC and RSENTnTSC) and configuration registers (RSENTnCC, RSENTnBRP, RSENTnIDE, and RSENTnMDC).

26.4.1.3 OPERATION IDLE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 011_B.

In OPERATION IDLE mode, no reception and transmission are done.

When entering OPERATION IDLE mode, frames in the receive buffer can be analyzed as in OPERATION ACTIVE mode, but no new frames are received.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, and RSENTnCSC.

26.4.1.4 OPERATION ACTIVE Mode

This mode is entered after the RSENTnMDC.OMC bits have been set to 101_B.

In OPERATION ACTIVE mode, transmission and reception can take place.

Frame reception and status flagging starts after a valid calibration pulse (including the falling edge at the beginning) was detected.

Read access to all registers is possible in this state.

Write access is granted only to RSENTnTSC, RSENTnIDE, RSENTnMDC, RSENTnSPCT, and RSENTnCSC.

26.4.1.5 Register Behavior in Operation Modes

Table 26.32 shows the register behavior when the RSENT module transitions to the indicated operation modes. The table also gives an overview about the access restriction in each operation mode.

Table 26.32 Register Behavior in Operation Modes

		MCU Reset	RESET		CONFIGURATION		OPERATION IDLE		OPERATION ACTIVE	
Register Name	**Symbol**	**Change**	**Change**	**R/W**	**Change**	**R/W**	**Change**	**R/W**	**Change**	**R/W**
RSENT Timestamp Register	RSENTnTSPC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
RSENT Timestamp Counter	RSENTnTSC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W *1	Unchanged	R/W *1
RSENT Communication Configuration Register	RSENTnCC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
RSENT Baud Rate Prescaler Register	RSENTnBRP	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R	Unchanged	R
RSENT Interrupt/DMA Enable Register	RSENTnIDE	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
RSENT Mode Control Register	RSENTnMDC	0000 0000$_H$	0000 0000$_H$	R/W	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
RSENT SPC Transmission Register	RSENTnSPCT	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R	Unchanged	R	Unchanged	R/W
RSENT Mode Status Register	RSENTnMST	0000 0000$_H$	0000 0000$_H$	R	0000 0001$_H$	R	0000 0003$_H$	R	0000 0005$_H$	R
RSENT Communication Status Register	RSENTnCS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Communication Status Clear Register	RSENTnCSC	0000 0000$_H$	0000 0000$_H$	R	Unchanged	R/W	Unchanged	R/W	Unchanged	R/W
RSENT Slow Channel Receive Timestamp Register	RSENTnSRTS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Slow Channel Receive Data Register	RSENTnSRXD	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Calibration Pulse Length Register	RSENTnCPL	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Message Length Register	RSENTnML	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Fast Channel Receive Timestamp Register	RSENTnFRTS	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Fast Channel Receive Data Register	RSENTnFRXD	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Calibration Pulse Length Mirror Register	RSENTnCPLM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Message Length Mirror Register	RSENTnMLM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Fast Channel Receive Timestamp Mirror Register	RSENTnFRTSM	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Expanded Fast Channel Receive Data Register 0	RSENTnEFRD0	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R
RSENT Expanded Fast Channel Receive Data Register 1	RSENTnEFRD1	0000 0000$_H$	0000 0000$_H$	R	0000 0000$_H$	R	Unchanged	R	Unchanged	R

Note 1. Means write restriction exists.

26.4.1.6 Mode Status Change Timing

This section shows the change timing of the RSENTnMST.OMS register i.e. the timing properties of the arrows in **Figure 26.2**. **Table 26.33** shows the change timing with respect to Falling Edge Detection, Register Access Clock, RSENT Communication Clock and sample clock.

Table 26.33 Mode Status Change Timing

Transition	Condition	Transition Time
RESET to CONFIGURATION		5 Register Access Clocks + 3 RSENT Communication Clocks
CONFIGURATION to OPERATION ACTIVE		5 Register Access Clocks + 6 RSENT Communication Clocks
OPERATION ACTIVE to OPERATION IDLE		Falling edge of next frame sync nibble + 5 sample clocks + 4 RSENT Communication Clocks + 4 Register Access Clocks
	In case frame reception has not started so far (falling edge of calibration pulse of 1st frame not detected)	5 Register Access Clocks + 1 sample clock + 7 RSENT Communication Clocks
OPERATION IDLE to CONFIGURATION		4 Register Access Clocks + 4 RSENT Communication Clocks
OPERATION ACTIVE to CONFIGURATION		4 Register Access Clocks + 7 RSENT Communication Clocks + 1 sample clock (+ 1 Register Access Clock - safety margin for user)
OPERATION IDLE to OPERATION ACTIVE		5 Register Access Clocks + 6 RSENT Communication Clocks

26.4.2 Clock Configuration

26.4.2.1 Timestamp

(1) Timestamp Clock Configuration

RSENT incorporates the timestamp counter.

The minimum required resolution of the timestamp is 1 µs. Depending on the supplied RSENT communication Clock frequency, the user should configure the RSENTnTSPC.TTPV bits to set the output of prescaler TPV to 1 µs clock tick. The input frequency is divided by the configured timestamp prescaler value RSENTnTSPC.TTPV.

Depending on the configured tick lengths, the resolution can be decreased by configuring the RSENTnTSPC.TTM bits. The already divided input frequency is divided further by the value of the RSENTnTSPC.TTM bits.

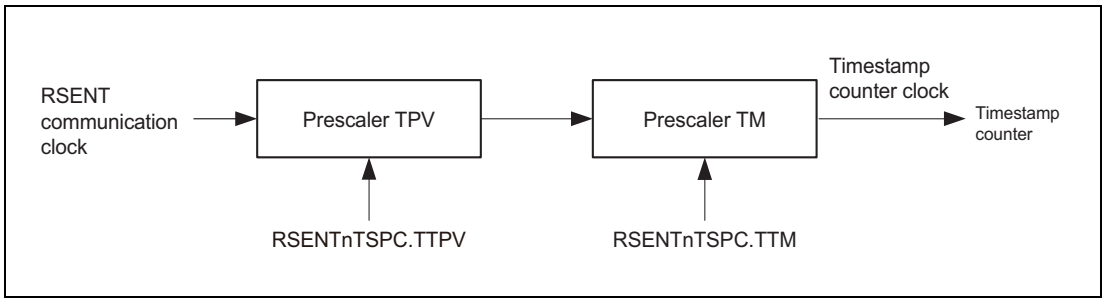


Figure 26.3 Timestamp Counter Clock Generation

(2) Timestamp Counter Operation

The timestamp counter value can be initialized to any value by writing to the RSENTnTSC.TS bits only when the RSENT module is in CONFIGURATION mode.

When timestamp counters are configured to operate in master mode (RSENTnTSPC.TMS = 0), the CPU can reset the timestamp counter by writing $0000\ 0000_H$ to the RSENTnTSC.TS bits when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode.

When timestamp counters are configured to operate in slave mode (RSENTnTSPC.TMS = 1), the timestamp counter is cleared when the CPU writes $0000\ 0000_H$ to the RSENTnTSC.TS bits of the channel set in the master when the RSENT module is in OPERATION IDLE or OPERATION ACTIVE mode. The RSENT module operating in slave mode should have the same timestamp counter prescaler settings as master RSENT module. When timestamp counter synchronization occurs, the internal timestamp counter prescalers are also synchronized.

The current timestamp counter value can be read from the RSENTnTSC.TS bits.

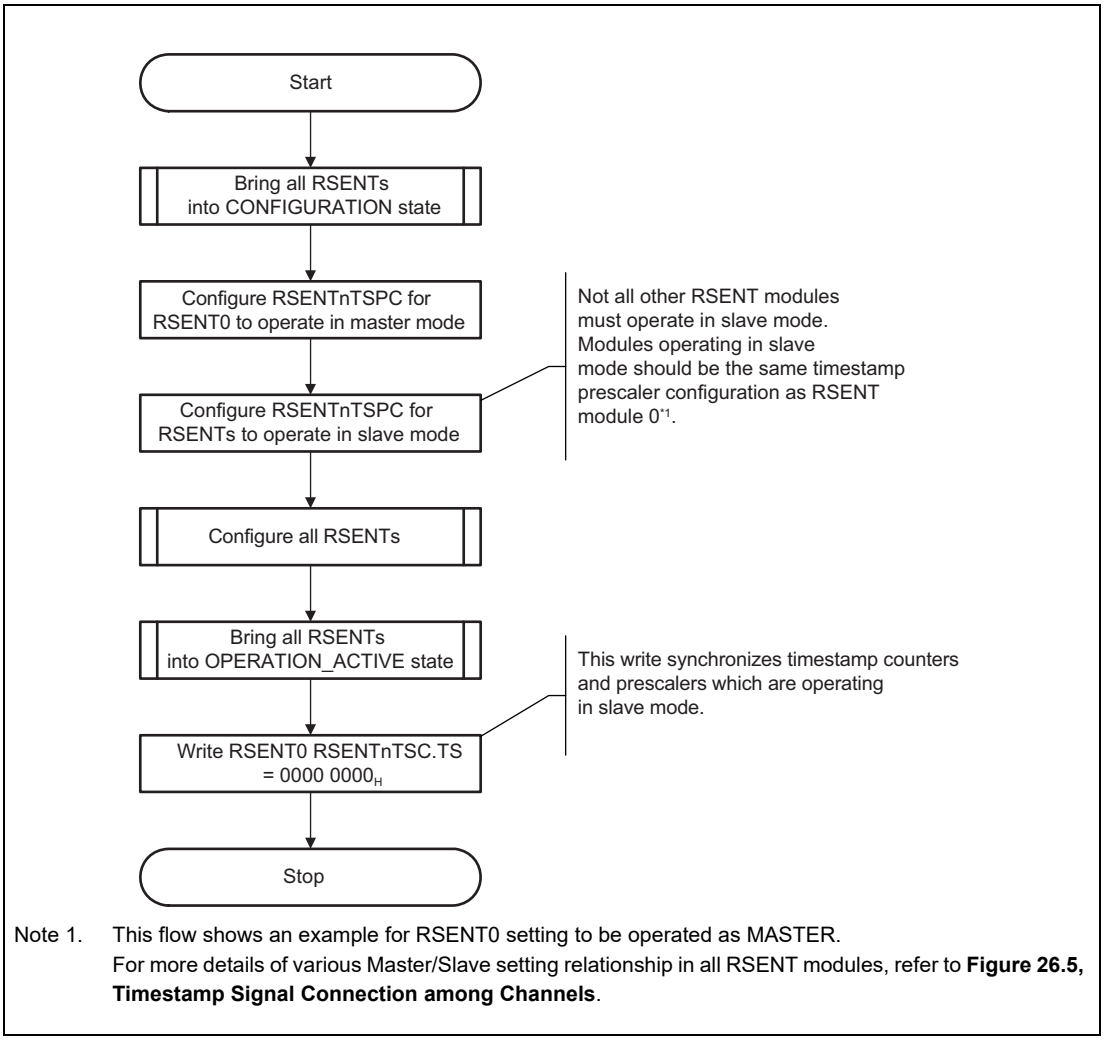
When the RSENT module is in OPERATION ACTIVE mode, each received message is stored with its related timestamp. Timestamp values are taken for fast channel and slow channel data.

The timestamp value is captured when the calibration pulse is detected.

The timestamp value for the fast channel is stored in the RSENTnFRTS.FTS bits or RSENTnFRTSM.FTSM bits register field when the fast channel receive message buffer is updated.

The timestamp value for the slow channel is stored in the RSENTnSRTS.STS bits. The timestamp value for the slow channel is identical to the timestamp value of the last fast channel message contributing to the slow channel message.

In case timestamp counter synchronization is required, the following flow should be used.



Note 1. This flow shows an example for RSENT0 setting to be operated as MASTER.
For more details of various Master/Slave setting relationship in all RSENT modules, refer to **Figure 26.5, Timestamp Signal Connection among Channels**.

Figure 26.4 Timestamp Counter Synchronization

Further synchronization can be done as long as master RSENT module is in either OPERATION ACTIVE or OPERATION IDLE state.

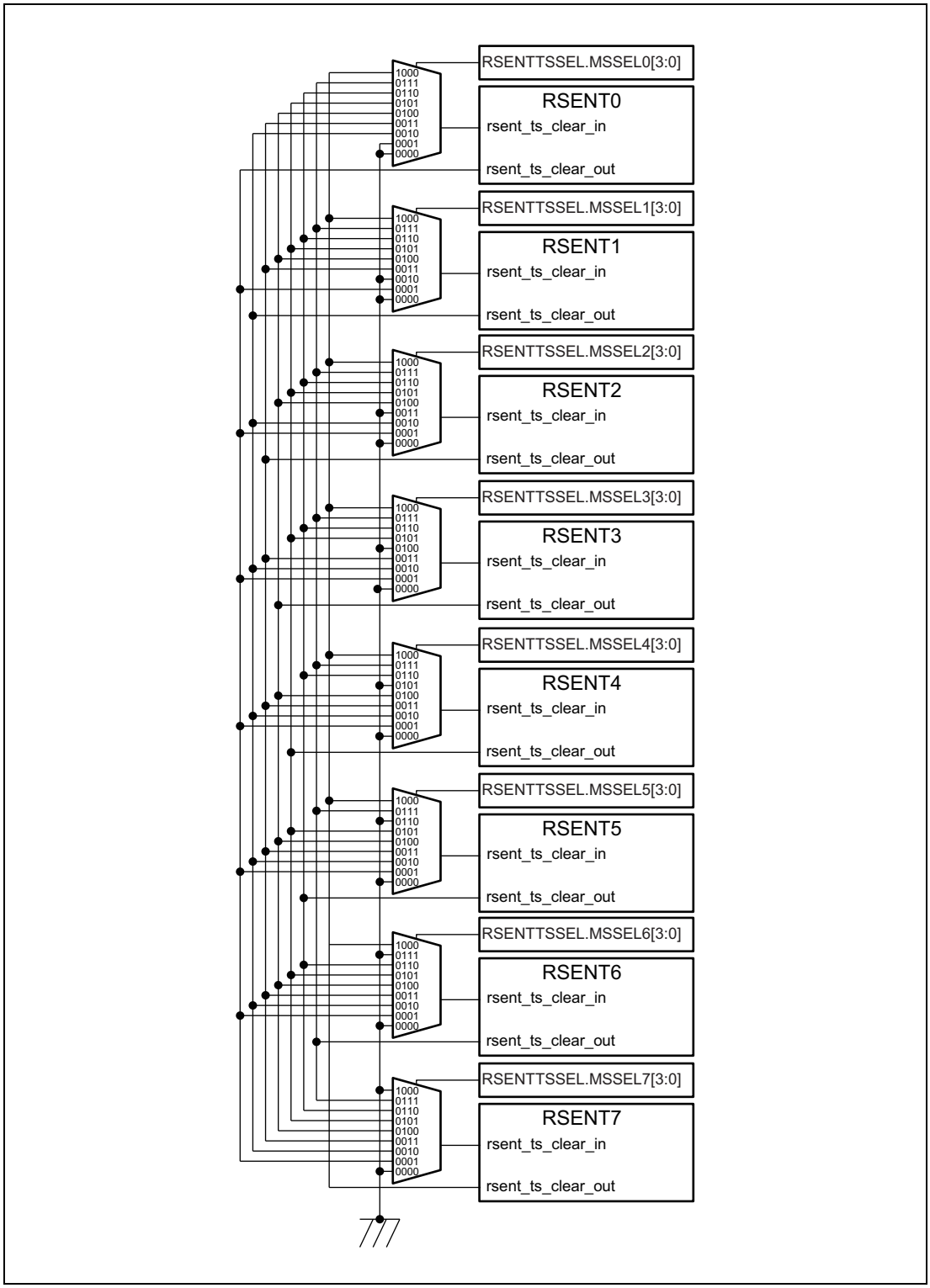


Figure 26.5 Timestamp Signal Connection among Channels

Timestamp clear input (sent_ts_clear_in) is used to clear the timestamp counter value. This signal shall be connected to the sent_ts_clear_out pin of the RSENT units which are dedicated as timestamp masters.

Timestamp clear output (sent_ts_clear_out) is set for one clock cycle when the timestamp counter is configured to operate in the master mode (RSENTnTSPC.TMS = 0) and the CPU clears the timestamp counter.

In master mode, the clearing timing is output to the output signal sent_ts_clear_out. In slave mode, the input signal sent_ts_clear_in from the master can be used to clear the timestamp.

When the timestamp counter is configured to operate in master mode (RSENTnTSPC.TMS = 0), sent_ts_clear_in has no effect on the timestamp counter register. This signal shall be connected to 1'b0 for the RSENT unit which is configured as master.

When the timestamp counter is configured to operate in slave mode (RSENTnTSPC.TMS = 1), the dedicated timestamp masters can be chosen by various setting of RSENTTSSEL register, while multi-masters or master/slave(s) relation is decided by the setting of RSENTnTSPC.TMS of each unit beforehand.

Setting examples:

[Example 1] RSENT2 operate as master and RSENT0, RSENT1, RSENT3, RSENT4, RSENT5, RSENT7 operate as slaves.

- RSENT0TSPC.TMS = 1_B
- RSENT1TSPC.TMS = 1_B
- RSENT2TSPC.TMS = 0_B
- RSENT3TSPC.TMS = 1_B
- RSENT4TSPC.TMS = 1_B
- RSENT5TSPC.TMS = 1_B
- RSENT6TSPC.TMS = Don't care
- RSENT7TSPC.TMS = 1_B
- RSENTTSSEL.MSSEL0[3:0] = 0011_B
- RSENTTSSEL.MSSEL1[3:0] = 0011_B
- RSENTTSSEL.MSSEL2[3:0] = 0011_B
- RSENTTSSEL.MSSEL3[3:0] = 0011_B
- RSENTTSSEL.MSSEL4[3:0] = 0011_B
- RSENTTSSEL.MSSEL5[3:0] = 0011_B
- RSENTTSSEL.MSSEL6[3:0] = 0000_B
- RSENTTSSEL.MSSEL7[3:0] = 0011_B

[Example 2] RSENT0, RSENT1, RSENT2, RSENT3, RSENT4, RSENT5, RSENT6, RSENT7 operate in master mode without slaves (RSENT1 is the timestamp master).

- RSENT0TSPC.TMS = 0_B
- RSENT1TSPC.TMS = 0_B
- RSENT2TSPC.TMS = 0_B
- RSENT3TSPC.TMS = 0_B
- RSENT4TSPC.TMS = 0_B
- RSENT5TSPC.TMS = 0_B
- RSENT6TSPC.TMS = 0_B

- RSENT7TSPC.TMS = 0_B
- RSENTTSSEL.MSSEL0[3:0] = 0001_B
- RSENTTSSEL.MSSEL1[3:0] = 0010_B
- RSENTTSSEL.MSSEL2[3:0] = 0011_B
- RSENTTSSEL.MSSEL3[3:0] = 0100_B
- RSENTTSSEL.MSSEL4[3:0] = 0101_B
- RSENTTSSEL.MSSEL5[3:0] = 0110_B
- RSENTTSSEL.MSSEL6[3:0] = 0111_B
- RSENTTSSEL.MSSEL7[3:0] = 1000_B

26.4.2.2 Communication Clock Configuration

(1) RX BRP Setting

"RSENT communication clock frequency ($f_{COMMUNICATION}$)" is multiplied by the configured lowest term fraction set by RSENTnBRP.SCDV and RSENTnBRP.SCMV to generate the sample clock.

The RSENTnBRP.SCDV and RSENTnBRP.SCMV values should be selected in such a way that a sample clock frequency (f_{SAMPLE}) of 16 MHz is generated according to the following formula.

$$f_{SAMPLE} = 16\ MHz = f_{COMMUNICATION} \times \frac{Sample\ Clock\ Multiplication\ Value\ (RSENTnBRP.SCMV+1)}{Sample\ Clock\ Division\ Value\ (RSENTnBRP.SCDV+1)}$$

Where Sample Clock Multiplication Value = 1 (RSENTnBRP.SCMV = 5'd0),
Sample Clock Division Value = 5 (RSENTnBRP.SCDV = 7'd4),
$f_{COMMUNICATION} = 80\ MHz$
$f_{SAMPLE} = 80 \times 1/5 = 16\ MHz$

(2) RX and SPC Tick Settings

The used tick length in RX and SPC function can be configured with the RSENTnBRP.TTI and RSENTnBRP.TTF bits. Tick lengths from 1.0 µs to 90.0 µs with a resolution of 0.1 µs can be configured.

The RSENTnBRP.TTI holds the integer part of the tick length and the RSENTnBRP.TTF bits hold the fractional part of the tick length. The tick length is then calculated by:

$$T_{TICK} = T_{RSENTnBRP.TTI} + T_{RSENTnBRP.TTF}$$

Where RSENTnBRP.TTI = 0, RSENTnBRP.TTF = 3

$$T_{TICK} = 1 + 0.3 = 1.3\ \mu s$$

26.4.3 RSENT Operation

When starting transfer through RSENTn in this product series, be sure to make the pin settings corresponding to the RSENTn interface, then follow the flow for the individual settings described in this section.

For details of the pin settings, see **Section 2, Pin Functions**.

26.4.3.1 Changing Operation Modes

Once initialization has been completed in CONFIGURATION mode, operation can be enabled by entering OPERATION ACTIVE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION ACTIVE and waiting for the RSENTnMST.OMS to transition to OPERATION ACTIVE.

Once in OPERATION ACTIVE mode the RSENT module begins to receive messages or SPC communication can be started depending on the configuration.

Figure 26.6 shows the communication enabled flow assuming that the RSENT module is in RESET mode:

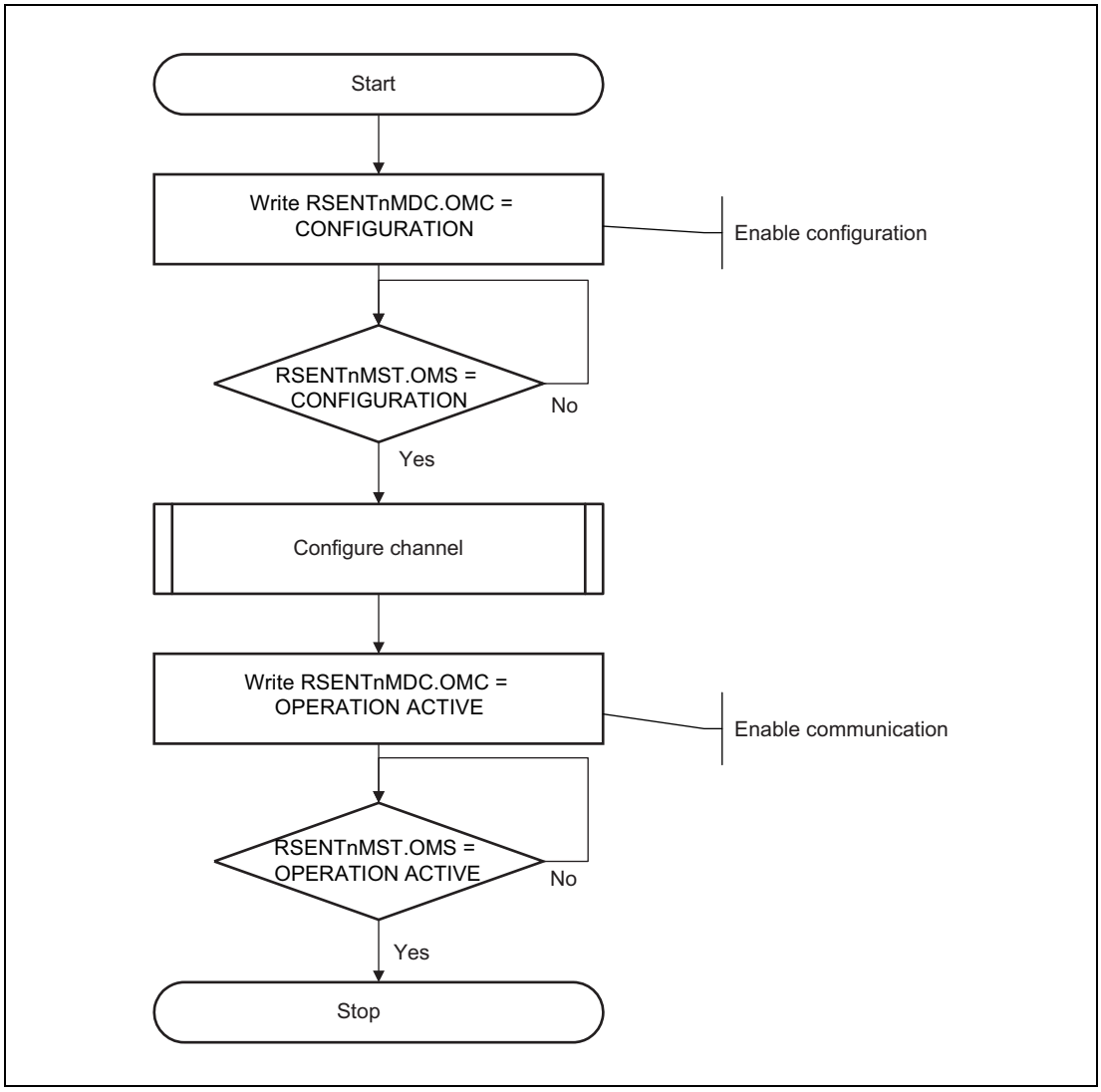


Figure 26.6 Communication Enable Flow

To leave OPERATION ACTIVE mode, communication should be disabled first by entering OPERATION IDLE mode. This is done by setting the RSENTnMDC.OMC bits to OPERATION IDLE and waiting for the RSENTnMST.OMS bits to transition to OPERATION IDLE.

However, when the SPC is enabled (RSENTnCC.SPCE = 1) and the SPC trigger transmission has not been requested after the previous SPC communication has been completed (e.g. successful reception for the previous SPC trigger transmission), the RSENT module can directly enter the CONFIGURATION mode.

The transition between OPERATION ACTIVE and OPERATION IDLE depends on the setting of the RSENTnCC.SPCE bit.

(1) RSENTnCC.SPCE = 0

In case a reception is currently ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge at the end of the next status communication nibble is detected or an error was flagged.

In case no reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place immediately.

(2) RSENTnCC.SPCE = 1

In case a reception is ongoing, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

In case a no response error is flagged, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place at the same time as the error flagging.

The mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the sequence of making a SPC trigger and receiving the response has been completed. This means when a response was already received, the transition takes place immediately. When the response is still pending, the mode change from OPERATION ACTIVE to OPERATION IDLE takes place when the falling edge of the end pulse is received.

CONFIGURATION mode can be entered by writing CONFIGURATION to the RSENTnMDC.OMC bits and waiting for the RSENTnMST.OMS to transition to CONFIGURATION.

Once CONFIGURATION mode is entered, the remaining status and message information stored in the RSENT module is lost since status and message information is cleared in CONFIGURATION mode.

Figure 26.7 shows the communication disable flow assuming that the RSENT module is in OPERATION ACTIVE mode.

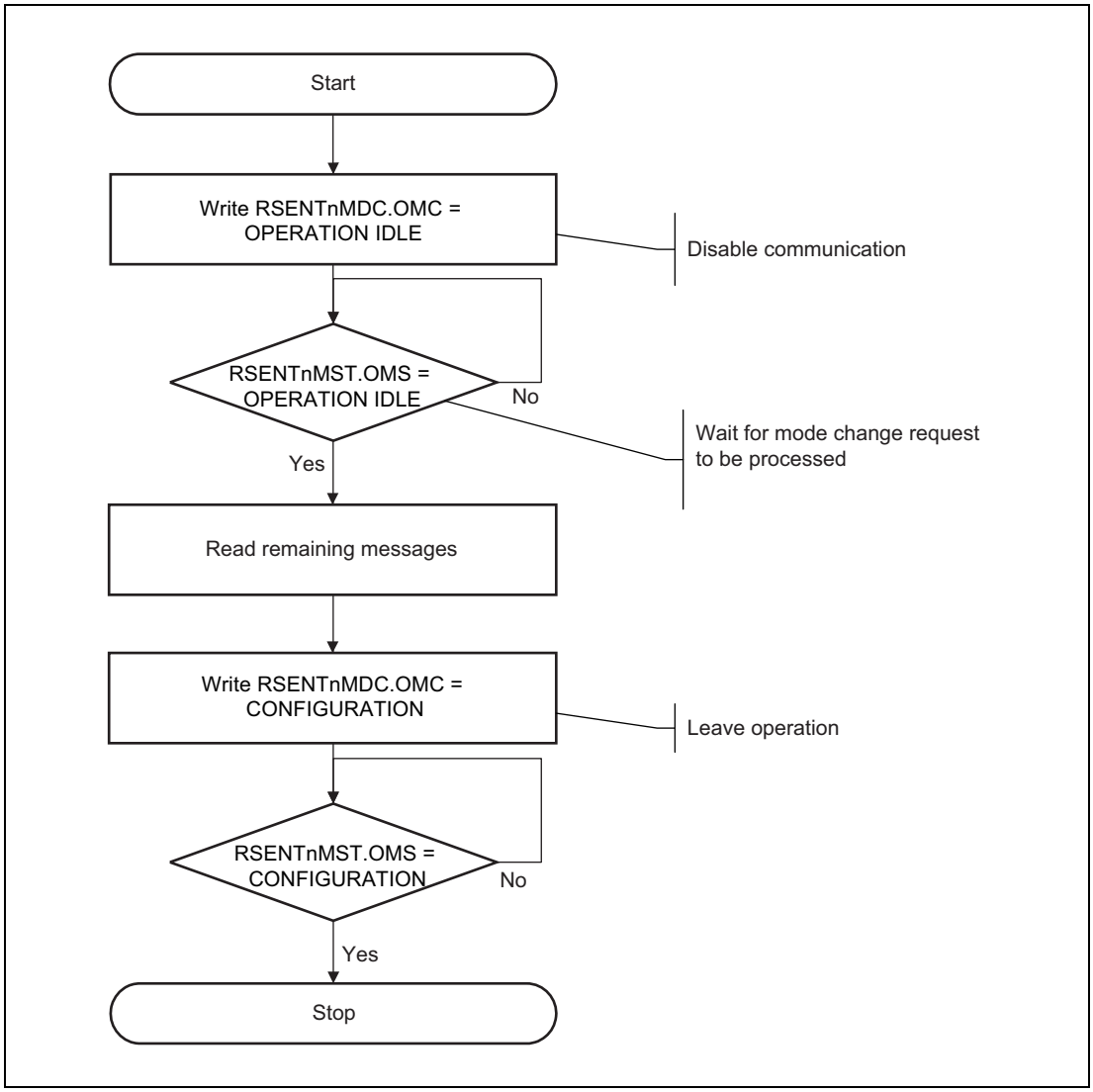


Figure 26.7 Communication Disable Flow

26.4.3.2 Message Reception

RSENT message reception is composed of the calibration pulse reception followed by the data nibble pulse reception.

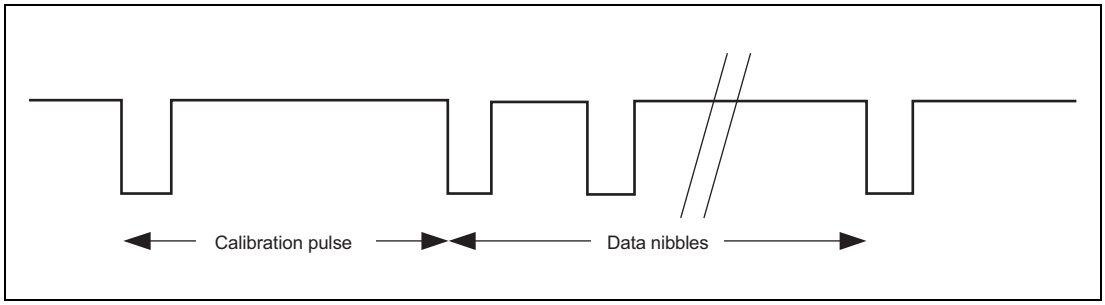


Figure 26.8 RSENT Received Message Structure

(1) Calibration Pulse Reception

Within the calibration pulse reception phase the internally generated clock tick is adjusted to the transmit clock speed.

In addition, the calibration pulse is used to end the previous message and perform message diagnostics. The RSENT module supports automatic calibration pulse length diagnostics in variable message length modes (RSENTnCC.PPTC = 0). In case the calibration pulse ratio check fails, the calibration pulse length variation error flag (RSENTnCS.CVS) is set to 1.

(2) Data Nibble Reception

The receive function of the RSENT module is a straightforward capture and compare function. The RSENT module receives sensor information encoded by the temporal distance of two consecutive falling edges on the data line. The temporal distance (the Number of Clock Ticks) is captured and compared against a set of values to determine the actual nibble value. The data encoding is illustrated in **Table 26.34, Data Nibble Encoding** below.

Table 26.34 Data Nibble Encoding

Nibble Period (Number of Clock Ticks)	Nibble Value (Binary)
12	0000_B
13	0001_B
14	0010_B
15	0011_B
16	0100_B
17	0101_B
18	0110_B
19	0111_B
20	1000_B
21	1001_B
22	1010_B
23	1011_B
24	1100_B
25	1101_B
26	1110_B
27	1111_B

The received data nibbles are composed into an RSENT message which is then stored in the fast channel message reception buffer.

Any other received nibble period during the reception of data nibbles will cause a fast channel nibble encoding error.

(3) Fast Channel Message Reception

Messages received on the fast message channel are stored in a receive buffer.

A fast channel message reception buffer is composed of the calibration pulse length register (RSENTnCPL), the message length register (RSENTnML), the fast channel receive timestamp register (RSENTnFRTS), and the fast channel receive data register (RSENTnFRXD).

All registers are placed on successive addresses that allows transferring the register content into memory using sDMA.

The RSENT module is equipped with a double receive buffer structure that allows the storage of two complete RSENT messages including the related timestamp and message length information. Message decoding and assembling are done in a separate register stage.

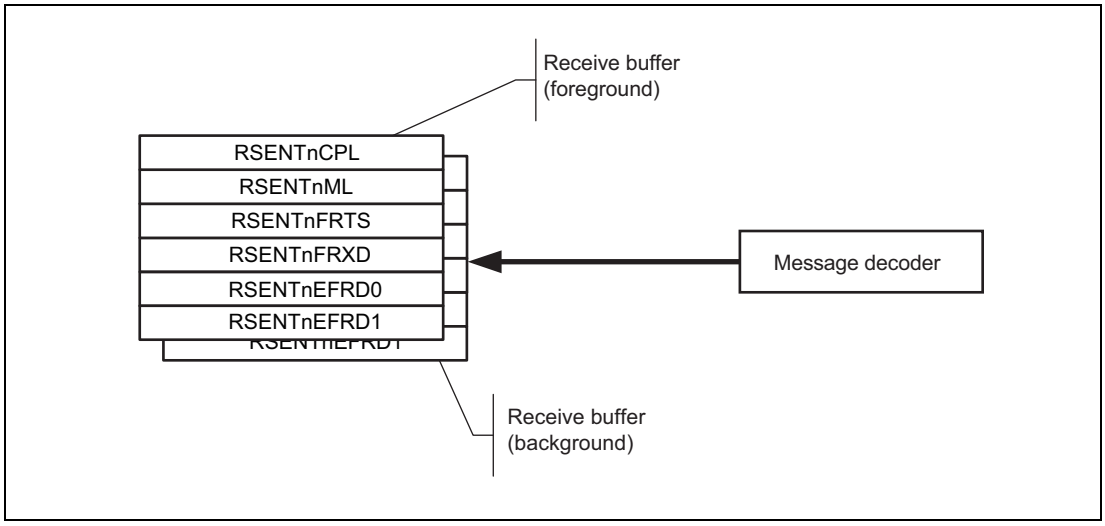


Figure 26.9 Fast Channel Message Reception Buffer

The first received message is placed into the message buffer that can be accessed by the CPU. This buffer (except the RSENTnFRXD.SNDM and RSENTnEFRD0.SNDM bits) is not updated any more until the RSENTnFRXD.FND bit or RSENTnEFRD1 register was read.

When a new message is placed into a receive buffer, the RSENTnFRXD.FND bit and RSENTnEFRD0.FND bits are set. At the same time, the RSENTnCS.FRS bit is set and, if enabled, a receive interrupt request is generated.

When the foreground receive buffer is holding an unprocessed message (the RSENTnFRXD.FND and RSENTnEFRD0.FND bits are 1), any further incoming message is placed in the background buffer. The background buffer is updated with any further incoming messages. In case an unprocessed background message buffer message is overwritten, the RSENTnCS.FMS bit is set to 1.

When the CPU reads the RSENTnFRXD.FND bit or RSENTnEFRD1 register and there is valid data in the background buffer, the data previously located in the background buffer becomes available in the receive buffer and is accessible by the CPU. If enabled, a new interrupt request for fast channel data is generated and RSENTnCS.FRS is set.

When the RSENTnEFRD0.FND and RSENTnFRXD.FND/RSENTnCS.FRS bit is not set, the data in the receive buffer is not defined and the CPU should not access the receive buffer.

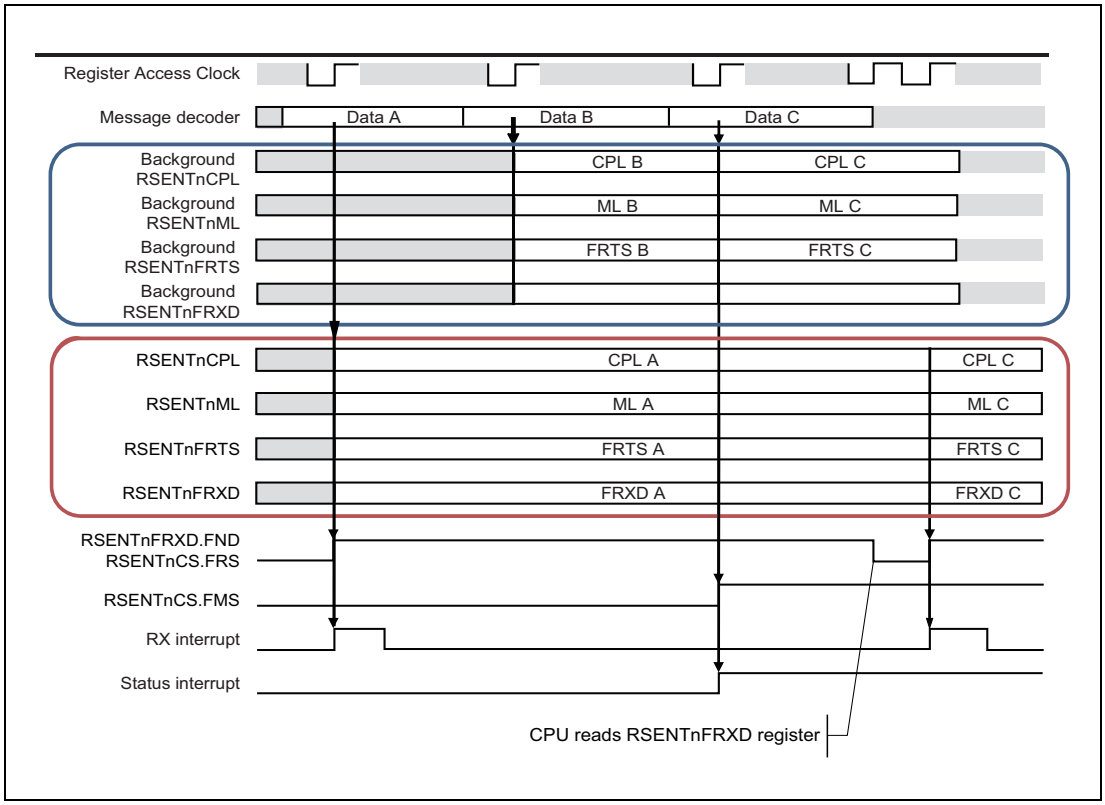


Figure 26.10 Fast Channel Message Reception Buffer Update Timing

The update timing of the receive buffer depends on the applied configuration as depicted in **Figure 26.11** to **Figure 26.14**.

The RSENTnFRTS register is updated with the current timestamp counter register value when the calibration pulse is detected.

The data alignment in the RSENTnFRXD register depends on the nibble data count (RSENTnCC.NDN).

Table 26.35 Data Nibble Alignment in RSENTnFRXD Register

RSENTnFRXD RSENTnCC.NDN	23:20	19:16	15:12	11:8	7:4	3:0
000_B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001_B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010_B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011_B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100_B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101_B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6

Table 26.36 Data Nibble Alignment in RSENTnEFRD1 Register

RSENTnEFRD1 RSENTnCC.NDN	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
000_B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1
001_B	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2
010_B	Undefined	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3
011_B	Undefined	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4
100_B	Undefined	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5
101_B	Undefined	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6
110_B	Undefined	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7
111_B	Nibble 1	Nibble 2	Nibble 3	Nibble 4	Nibble 5	Nibble 6	Nibble 7	Nibble 8

(a) SAE operation with variable message length and preferred check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 0)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the preferred option in the J2716 2016 specification. In this mode, message diagnostics is done after the calibration pulse was received following a message.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

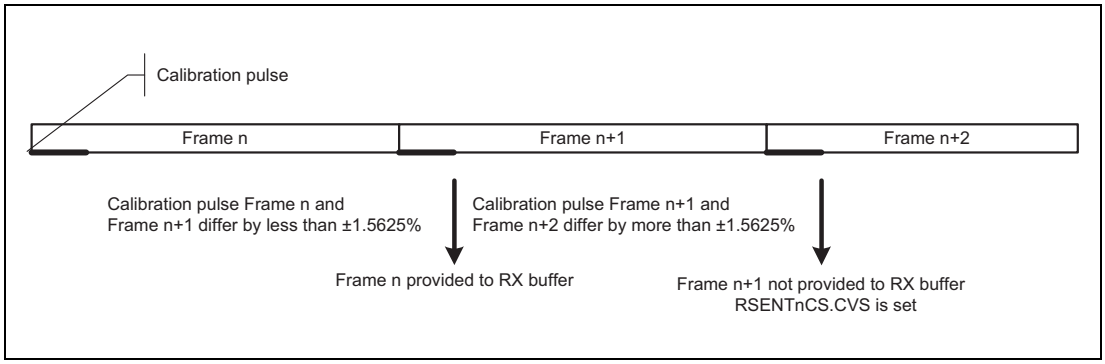


Figure 26.11 Buffer Update in Variable Message Length Mode and Preferred Check Method

(b) SAE operation with variable message length and optional check method (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 0, RSENTnCC.FCM = 1)

In this operation mode, the RSENT module automatically performs the check for successive calibration pulse variation according to the optional frame check method as described in the J2716 2016 specification. In this mode, the calibration pulse of the current frame is compared to the calibration pulse of the last valid preceding frame.

If this check is passed, the message reception buffer is updated.

If this check is not passed, the message reception buffer is not updated and RSENTnCS.CVS is set to 1.

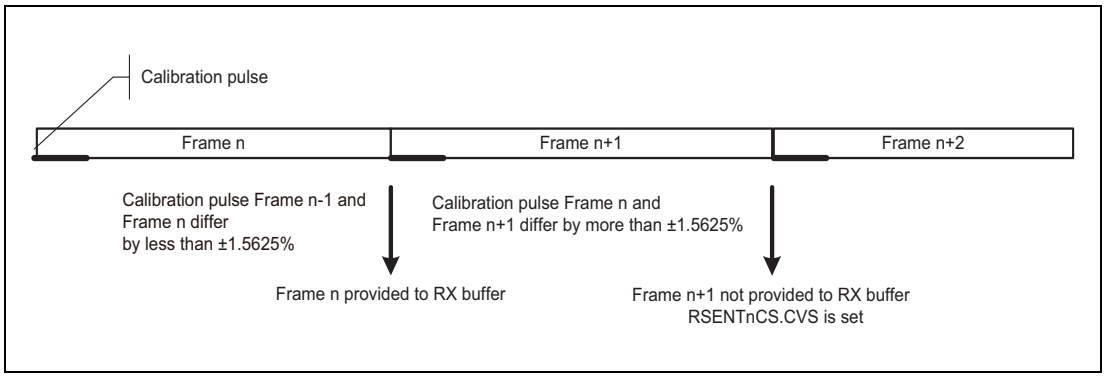


Figure 26.12 Buffer Update in Variable Message Length Mode and Optional Check Method

(c) SAE operation with fixed message length (RSENTnCC.SPCE = 0, RSENTnCC.PPTC = 1)

In this mode, the RSENT module does not perform the check for calibration pulse and message length ratio according to the preferred option in the J2716 2016 specification. In this mode, the RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples.

The message buffer is updated at the beginning of the following calibration pulse irrespective of the values in the RSENTnCPL and RSENTnML registers. The CPU needs to calculate the ratio and either accept or discard the message.

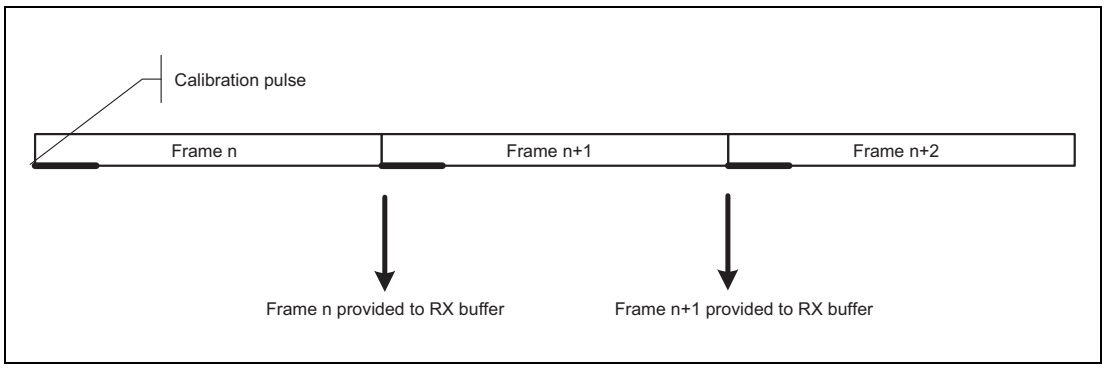


Figure 26.13 Buffer Update in Fixed Message Length Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

(d) SPC operation (RSENTnCC.SPCE = 1)

In this operation mode, sensor data transmission is done following a SPC master trigger pulse. Within SAE SENT communication, the calibration pulse or pause pulse is terminating the previous message. In SPC communication, the sensor is only sending data following a SPC trigger request. An end pulse sent by the sensor is terminating the message. The message buffer is updated at the beginning of the end pulse.

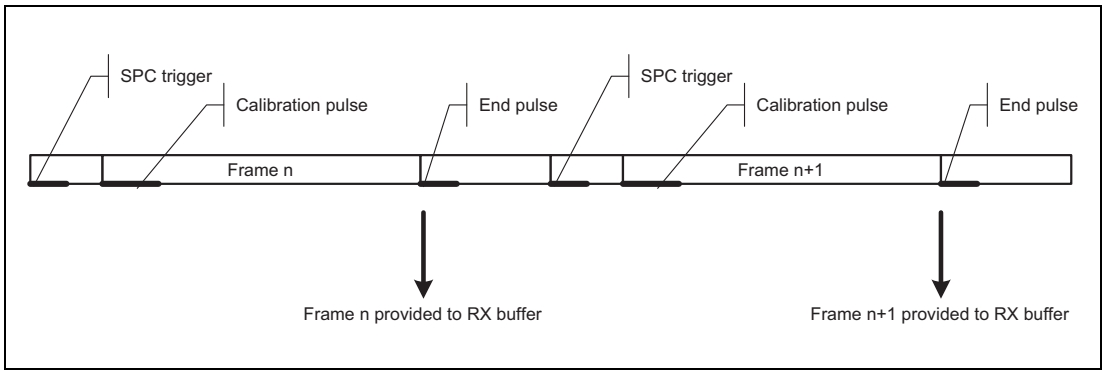


Figure 26.14 Buffer Update in SPC Mode

The RSENTnCS.CVS (calibration pulse width variation error status) bit is never set in this mode.

The RSENT module provides the calibration pulse length in the RSENTnCPL register and the message length information in the RSENTnML register. The numbers provided are based on samples. The CPU needs to calculate the ratio of calibration pulses and/or message length and either accept or discard the message.

In case of variable message length mode, the RSENT module cannot perform this check because the receive timing of the next calibration pulse depends on the next SPC trigger timing.

(4) Fast Channel Reception Flow

In **Figure 26.15**, the recommended reception flow for the fast channel message reception buffer is shown.

When using a polling or event driven method, the CPU should only read the setting of the RSENTnCS.FRS bit to check the availability of new fast channel data.

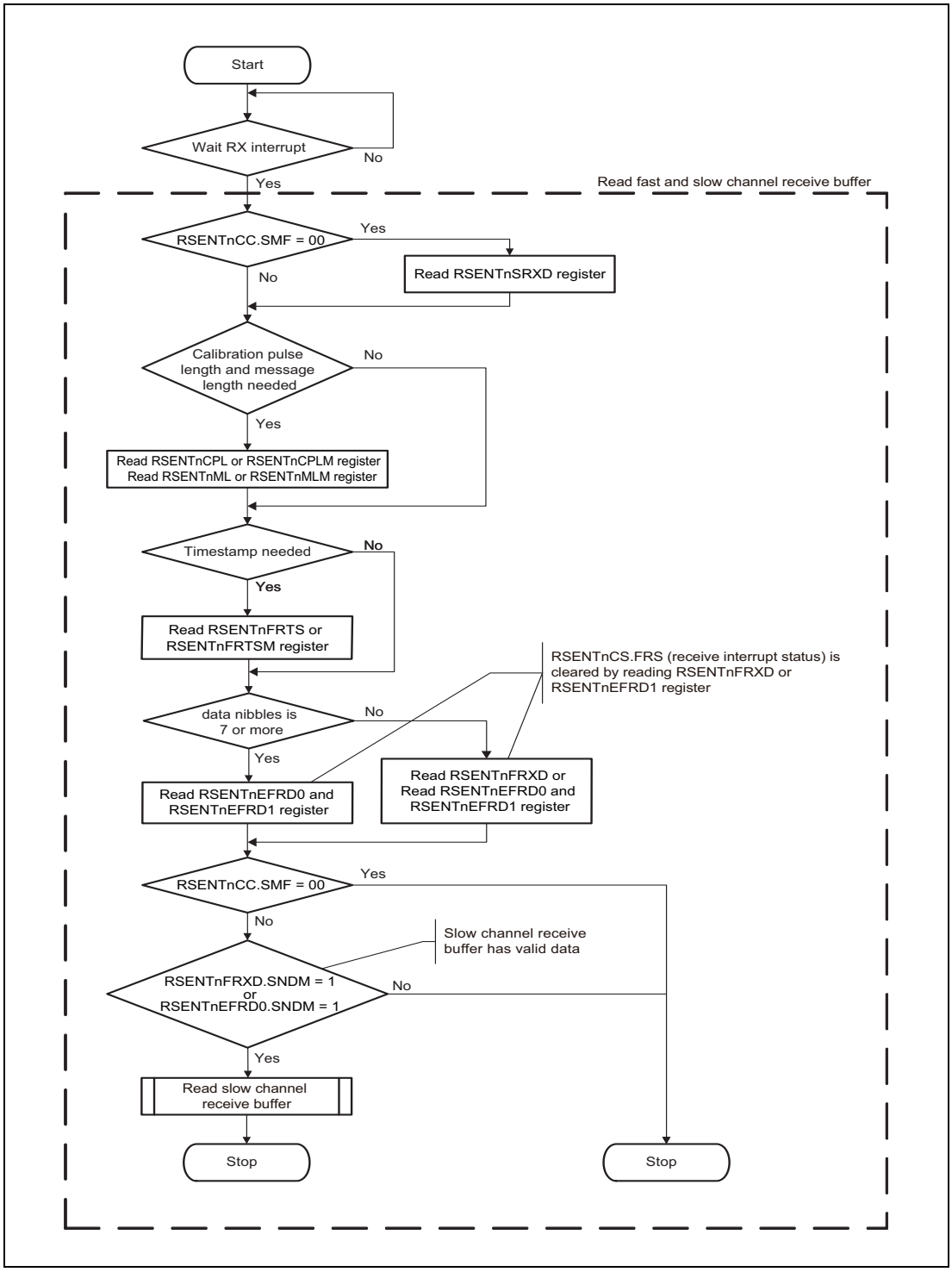


Figure 26.15 Fast Channel Reception Flow

In any case, the CPU should keep the order in reading the receive buffer registers as shown in the flow. The RSENTnFRXD or RSENTnEFRD0 register should be the last register to be accessed.

The handling of the slow channel receive buffer is described in **Section 26.4.3.2(6), Slow Channel Reception Flow**.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

(5) Slow Channel Message Reception

The RSENT module supports extraction of the slow message out of the fast channel messages by using the bits 3 and 2 out of the status and communications nibble. In order to enable the slow channel extraction, the CPU should set the RSENTnCC.SMF bits to the expected serial message format.

When no serial message extraction is selected (RSENTnCC.SMF = 00_B), the RSENTnSRXD register becomes part of the fast channel message reception buffer structure (including background buffer) and RSENTnSRTS register should be ignored. The communications and status nibble is placed in the RSENTnSRXD.IDD bits. Furthermore no slow channel new data and slow channel message lost flags are generated.

In order to receive the slow channel serial message, all fast channel serial messages contributing to a slow channel serial message must be received successfully and the received slow channel serial message must comply with the selected serial message format.

A message lost on the fast channel does not impact the reception on the slow channel.

A slow channel message reception buffer is composed of the slow channel receive timestamp register (RSENTnSRTS) and the slow channel receive data register (RSENTnSRXD).

In opposite to the fast channel message reception buffer, the slow channel message reception buffer does not support a double receive buffer structure; only a single receive buffer structure is available.

Message decoding and assembling is done in a separate register stage.

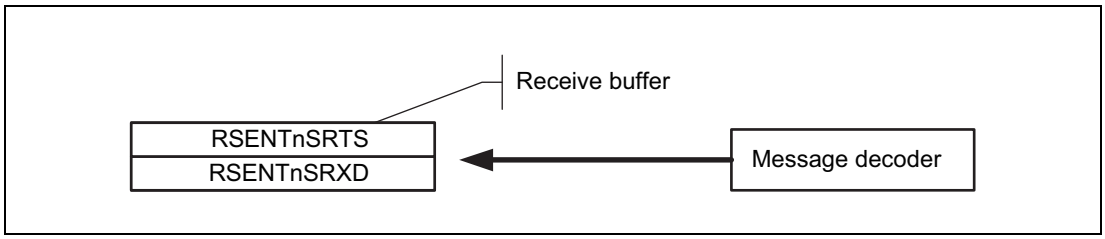


Figure 26.16 Slow Channel Message Reception Buffer

The slow channel message reception buffer is updated at the same time as the fast channel message reception buffer that holds the last status and communication nibble required for the slow channel message. The RSENTnSRXD.SND bit is set to 1 at the same time.

Further updates to the buffer are not carried out until after the RSENTnSRXD.SND bit has been read.

When the receive buffer is holding an unprocessed message (RSENTnSRXD.SND is 1), any further incoming message is lost (the slow channel message reception buffer is not updated) and RSENTnCS.SMS is set to 1.

When the CPU reads the RSENTnSRXD register, RSENTnSRXD.SND is automatically cleared.

The RSENTnSRTS register is updated with the current timestamp counter register value of the last frame contributing to the slow channel message.

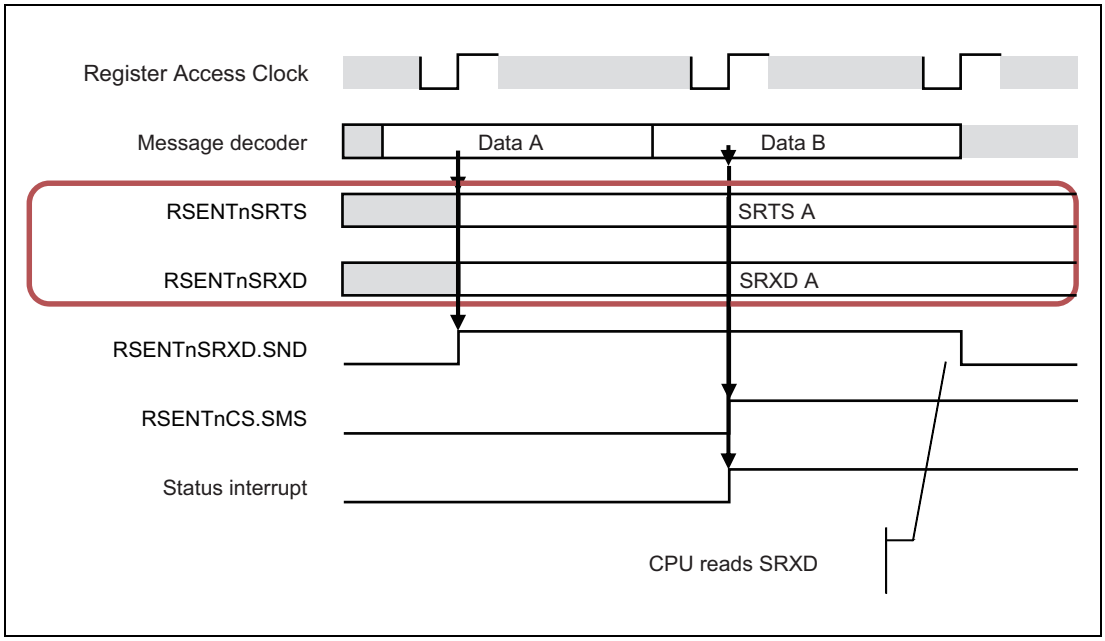


Figure 26.17 Slow Channel Message Reception Buffer Update Timing

The data alignment in the RSENTnSRXD register depends on the slow channel message format (RSENTnCC.SMF) and the received configuration bit.

Table 26.37 Data Alignment in RSENTnSRXD Register

RSENTnCC.SMF	RSENTnSRXD.SMGC	RSENTnSRXD.IDD [19:16]	RSENTnSRXD.IDD [15:12]	RSENTnSRXD.IDD [11:8]	RSENTnSRXD.IDD [7:4]	RSENTnSRXD.IDD [3:0]
00_B	Undefined	Undefined	Undefined	Undefined	Undefined	Status and communication nibble
01_B	Undefined	Undefined	Undefined	Message ID[3:0]	DATA[7:4]	DATA[3:0]
10_B	0	Message ID[7:4]	Message ID[3:0]	DATA[11:8]	DATA[7:4]	DATA[3:0]
10_B	1	Message ID[3:0]	DATA[15:12]	DATA[11:8]	DATA[7:4]	DATA[3:0]

(6) Slow Channel Reception Flow

In **Figure 26.18**, the recommended reception flow for the slow channel message reception buffer is shown. When the slow channel receive data is required, this process should be executed as part of the fast channel reception flow.

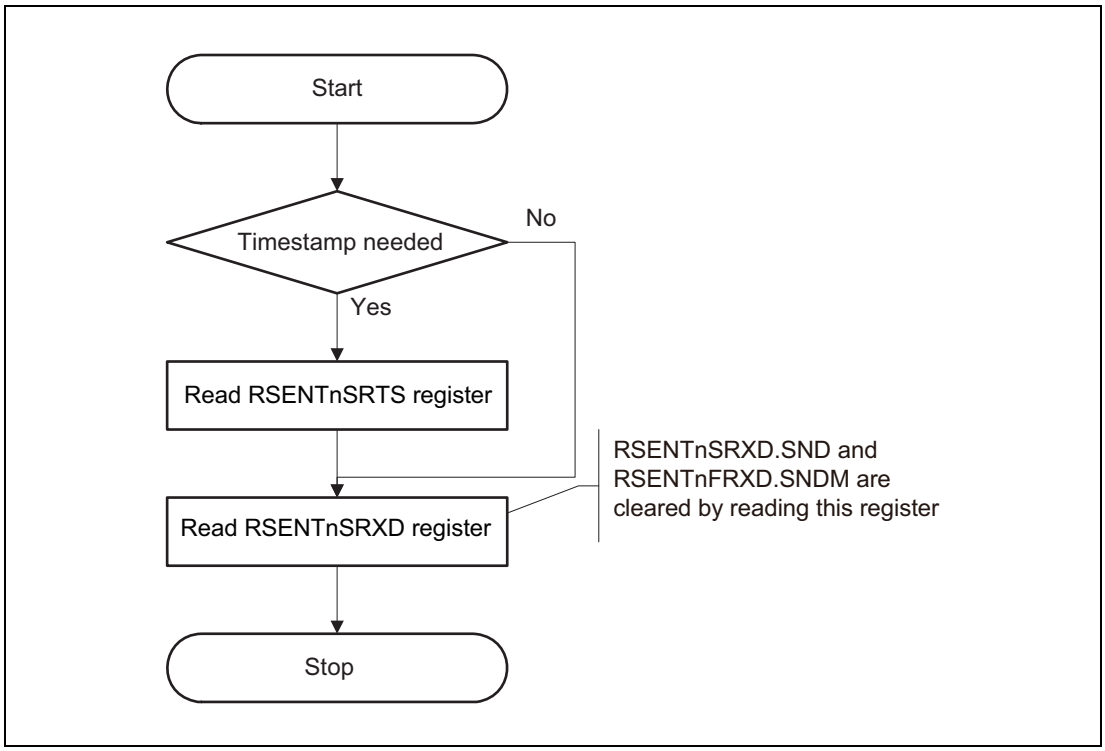


Figure 26.18 Slow Channel Reception Flow

In any case, the CPU should keep the order in reading the slow channel message reception buffer registers as shown in the flow. The RSENTnSRXD.SND bit should be accessed as last.

(7) DMA Flow

In case of DMA usage, the start address for the DMA usage and the number of transfers define which part of the receive buffer will be transferred. The RSENTnFRXD or RSENTnEFRD1 register should be the last register to be accessed using a 32-bit access method.

In case of SAE communication with pause pulse and fixed message length, the flow must be extended by checking of the ratio of the calibration pulse to message length. This variation check must be performed by the CPU. In case the variation check fails, the CPU must discard the received message.

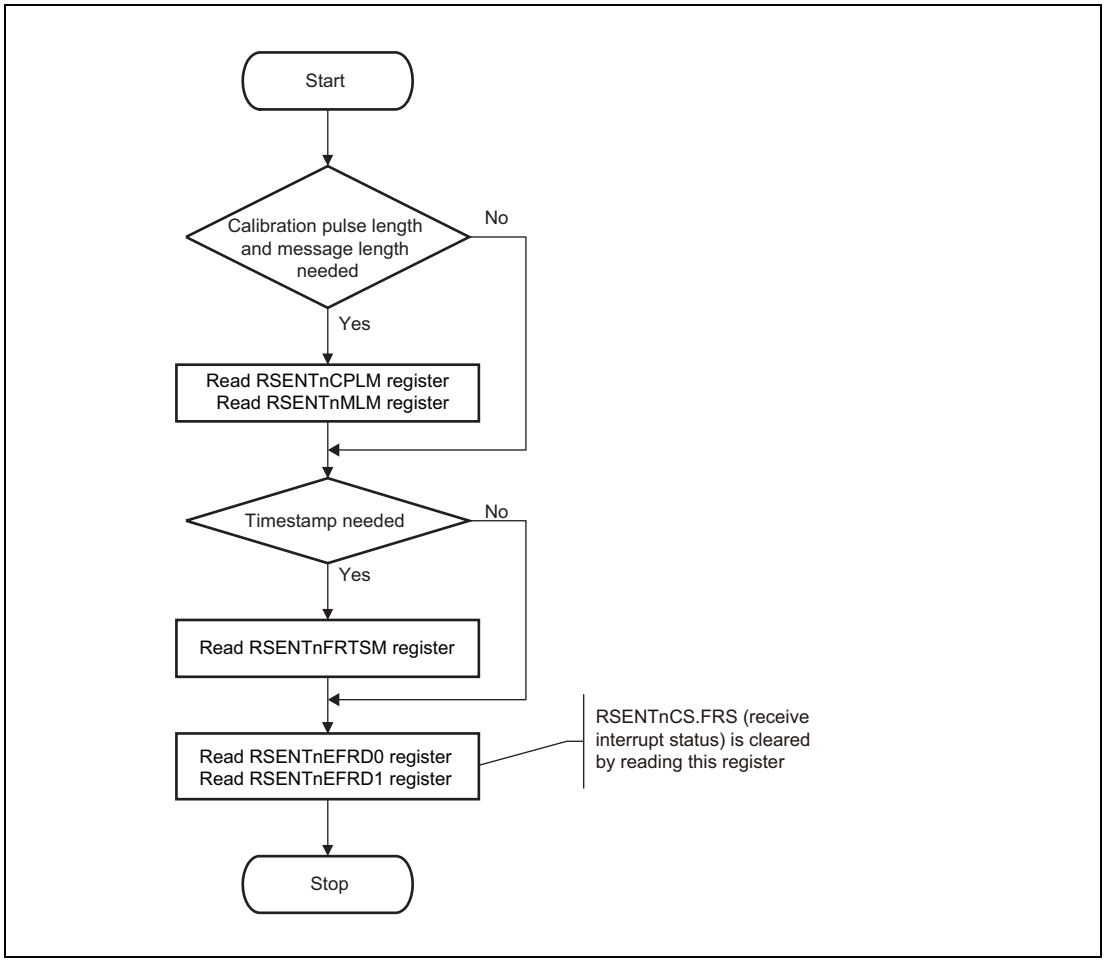


Figure 26.19 DMA Reception Flow

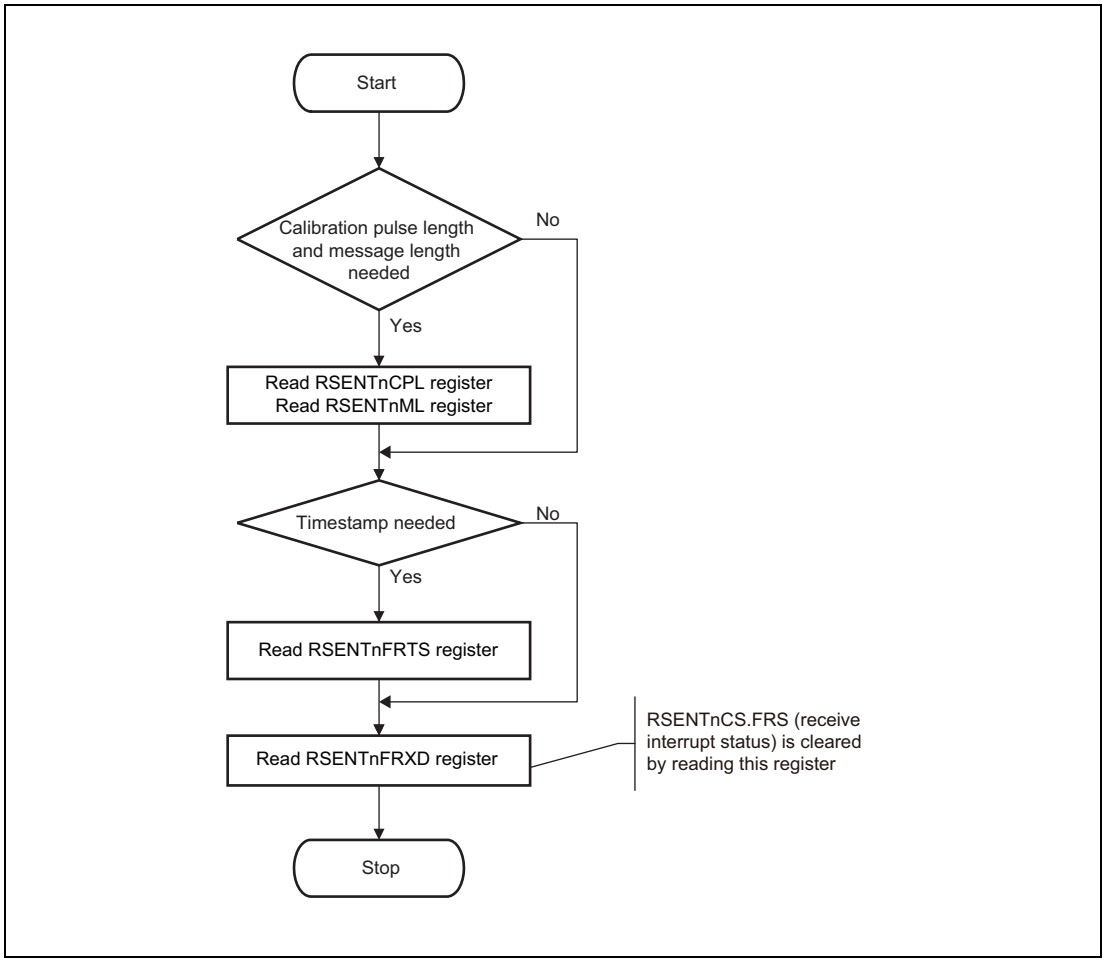


Figure 26.20 DMA Reception Flow (Data Nibble are 6 or Less)

In the software processing, when the transferred data set, the CPU should check the status of the transferred RSENTnFRXD.SNDM bit. If this bit is set to 1, then the user needs to read the slow channel message reception buffer if needed.

(8) Error Flagging

Message lost errors (RSENTnCS.SMS or RSENTnCS.FMS) are flagged when a new message's diagnostics pass before the previous message is read.

The SPC communication error (shown in RSENTnCS.NRS) is flagged when the CPU writes to RSENTnSPTC.TLL before or during response reception.

The update timings for fast channel reception errors (RSENTnCS.CVS, RSENTnCS.CLS, RSENTnCS.FNS, RSENTnCS.FES, and RSENTnCS.FCS) and slow channel reception errors (RSENTnCS.SCS and RSENTnCS.SES) depends on the configuration of RSENTnCC.SPCE, RSENTnCC.FCM, RSENTnCC.PPC, and RSENTnCC.PPTC.

Table 26.38, Error flag set timing when RSENTnCC.SPCE = 0 and **Table 26.39, Error flag set timing when RSENTnCC.SPCE = 1** list the timings with which the error flags corresponding to each setting are updated.

In case a nibble encoding error or calibration pulse length error is detected, message reception is terminated immediately. No further error flagging for this message is done. Message decoding starts again after a calibration pulse without calibration length error (RSENTnCS.CLS) is detected.

Table 26.38 Error flag set timing when RSENTnCC.SPCE = 0

RSENTnCC.SPCE	0							
RSENTnCC.FCM	0				1			
RSENTnCC.PPC	0		1		0		1	
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	EC	×	EC	IM	IM	×	IM	IM
RSENTnCS.FES	EC	×	EC	IM	IM	×	IM	IM
RSENTnCS.FNS	EC	×	EC	—	—	×	—	—
RSENTnCS.SCS	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.SES	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.CLS	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.CVS	EC	×	EC	—	EC	×	EC	—

Note: EC: End of calibration pulse
IM: Immediately when detected
—: Not set
×: Invalid configuration

NOTE

When RSENTnCC.PPC and RSENTnCC.PPTC are 1_B, the mode is pause pulse with fixed message length.

When this mode, this diagnostic can be used as the receiver does not need to wait for the next calibration pulse to diagnose the current receive frame, so RSENTnCS.FNS and RSENTnCS.CVS are not set.

Table 26.39 Error flag set timing when RSENTnCC.SPCE = 1

RSENTnCC.SPCE	1							
RSENTnCC.FCM	0				1			
RSENTnCC.PPC	0		1		0		1	
RSENTnCC.PPTC	0	1	0	1	0	1	0	1
RSENTnCS.FCS	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.FES	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.FNS	—	×	—	—	—	×	—	—
RSENTnCS.SCS	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.SES	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.CLS	IM	×	IM	IM	IM	×	IM	IM
RSENTnCS.CVS	—	×	—	—	—	×	—	—

Note: IM: Immediately when detected
—: Not set
×: Invalid configuration

NOTE

In case the sensor stops communication, no buffer update or status update for last message takes place. The SW should take care of this by timeout checks.
SPC mode is a communication of only one frame that is start the SPC trigger to the starting point.
The Error flags are immediately set when they are detected. So RSENTnCS.FNE and RSENTnCS.CVS are not set.

When a transition to OPERATION IDLE is configured in RSENTnMDC.OMC and an error for the calibration or the fast channel reception is detected in the message in which the mode transition was requested, the error is not flagged and the message is aborted.

In case of a fast channel encoding error or a calibration pulse length error the OPERATION IDLE mode is entered immediately.

In case of a fast channel nibble count error, fast channel CRC error, or fast channel calibration pulse variation error the OPERATION IDLE state is entered at the end of the next status and communications nibble.

RSENTnCS.FNS is only set after a valid calibration pulse was detected and all following nibbles have a valid length (≥ 12 ticks and ≤ 27 ticks) or no nibble was received between two valid calibration pulses.

RSENTnCS.FES is only set if the nibble with an encoding error occurred in the communication and status nibble, CRC nibble or in one of the expected data nibbles.

If SPC is enabled (RSENTnCC.SPCE = 1), RSENTnCS.CLS is set if a calibration pulse was expected but the pulse length does not meet the calibration pulse range. If SPC is disabled (RSENTnCC.SPCE = 0), RSENTnCS.CLS is set only after a valid calibration pulse has been received and a calibration pulse was expected but the pulse length does not meet the calibration pulse range.

During re-synchronization additional error flags might be set which is not affecting the reception of the following frame.

26.4.4 SPC Function

The RSENT module supports an extension of the SAE J2716 specification known as SPC, whereby the RSENT module can pull down the RX line to initiate RSENT message transmission.

The user can configure the polarity of the sent_spc port.

The text below describes the behavior of the sent_spc port with the default settings of RSENTnCC.SOPC. When the value of RSENTnCC.SOPC after a reset is changed, the sent_spc port operates with inverted polarity.

The user can enable or disable the SPC extension by setting the RSENTnCC.SPCE bit. When the RSENTnCC.SPCE bit is set to 0, SPC is disabled. The sent_spc port is driven low by the RSENT module, allowing normal RSENT reception to take place. When the RSENTnCC.SPCE bit is set to 1, SPC is enabled and the sent_spc port can be driven high by the RSENT module to request a frame transmission by the sensor.

The transmission function of the RSENT module is a straightforward PWM function. The purpose of this function is to communicate in direction to the sensor by the output sent_spc. With the sent_spc output, the RSENT module can pull down the signal line by an external transistor.

The signal line will be held low for a configured length of tick time specified in the RSENTnSPCT.TLL bits.

The Tick time is configured with the RSENTnBRP.TTI bits and the RSENTnBRP.TTF bits which are equal to the transmission tick time. For details, see **Section 26.4.2.2(2), RX and SPC Tick Settings**.

Figure 26.21 and **Figure 26.22** show an example of the SPC circuit connection.

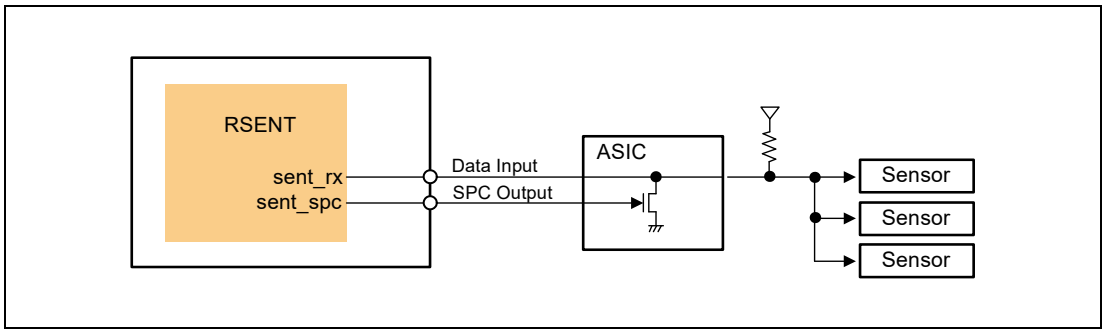


Figure 26.21 Example of the Circuit for two-wire interface

In a single sensor system, this function can be used to trigger data transmission from the sensor. Further data can be sent to the sensor by varying the trigger pulse length. In a multi sensor system, this function can be used to address a dedicated sensor and request a data transmission.

Once RSENT SPC initialization is complete, transmission can be triggered by writing the trigger pulse width to the RSENTnSPCT.TLL register. When a transmission is triggered, the trigger pulse with the configured length is sent. Then a frame reception is expected. After frame reception was done, a new trigger pulse can be sent.

Writing to RSENTnSPCT.TLL requests a SPC trigger transmission. After writing to RSENTnSPCT.TLL, the CPU should read RSENTnCS.NRS to check whether the previous request was completed or not.

In case RSENTnCS.NRS is set, no SPC trigger is sent and any potentially ongoing reception at this time is aborted. The CPU should clear RSENTnCS.NRS by writing 1'b1 to RSENTnCSC.NRC. The CPU can write again to RSENTnSPCT.TLL to request a SPC trigger transmission.

In case RSENTnCS.NRS is not set, the CPU should set a reception timeout counter in software. If a reception occurs before the timeout counter elapses, the user should process the received slow and fast channel data as shown in the fast channel reception flow (**Figure 26.15, Fast Channel Reception Flow**) and slow channel reception flow (**Figure 26.18, Slow Channel Reception Flow**).

When the timeout counter elapses without any successful reception, the addressed sensor seems not to send any valid response. The CPU should analyze the RSENTnCS register to analyze the reason for no successful reception. A new request can be made considering that when RSENTnCS.NRS gets set no SPC trigger is sent.

According to the SPC protocol, a frame is received when the falling edge of the end pulse has been detected. It is the user responsibility to not make a SPC trigger request during the end pulse as this might not be recognized by the sensor.

Purpose of the timeout function is to define a timeout window for response reception in software.

Figure 26.22, Transmission Flow shows a transmission flow with a timeout function implemented in software. The timeout function is optional and can be omitted if not needed.

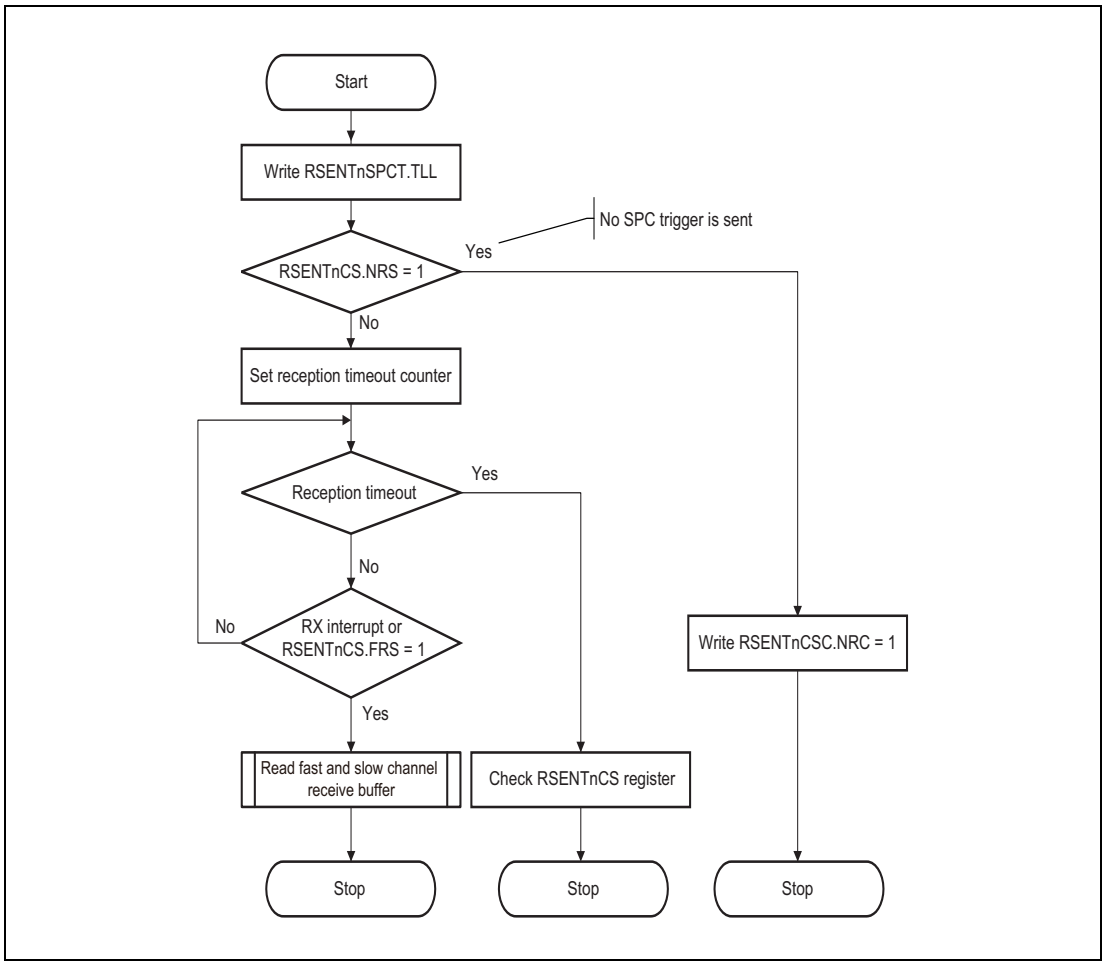


Figure 26.22 Transmission Flow

26.4.4.1 Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

The RSENTnRX input pin and RSENTnSPCO output pin functions are assigned to the same multiplexed pin. Two functions can be used on the pin, when it is set for an N-ch open-drain configuration.

Figure 26.23 is a block diagram of multiplexing of the RSENTnRX and RSENTnSPCO pin functions.

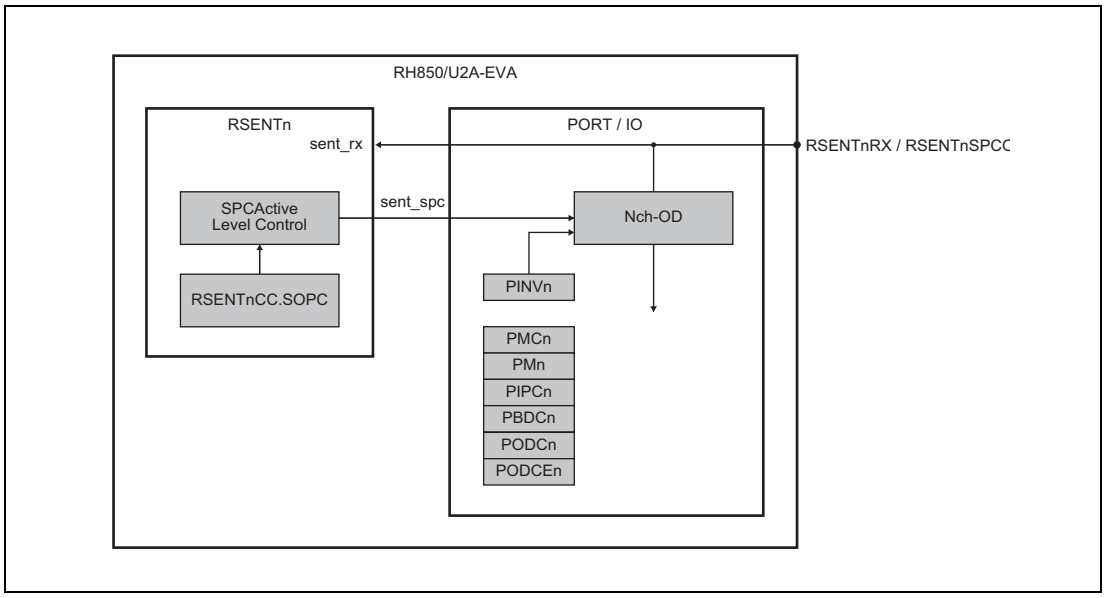


Figure 26.23 Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

Connect the sent_spc signal, which is an RSENT output signal, to an N-ch open drain I/O buffer. The RSENT is able to control the polarity of sent_spc output and the polarity of the pin output by setting the RSENTnCC.SOPC bit and the PINVn register of the I/O buffer, respectively. These registers must be set in an appropriate combination.

Table 26.40 lists the register settings when the RSENTnRX and RSENTnSPCO pin functions share the same pin.

Table 26.40 Setting for Multiplexing of the RSENTnRX and RSENTnSPCO Pin Functions

Register Name		Setting	Description
PMCn	Port mode control register	1	Alternative mode
PMn	Port mode register	0	Output mode (output enabled)
PIPCn	Port IP control register	0	I/O mode is controlled by PMn.PMn_m (software or hardware I/O control)
PBDCn	Port bi-direction control register	1	Bi-direction mode is enabled
PODCn	Port open-drain control register	1	N-ch open drain (PODCn_m = 1 or PODCEn_m = 0)
PODCEn	Port open-drain expansion register	0	N-ch open drain (PODCn_m = 1 or PODCEn_m = 0)
PINVn	Port output level inversion register	0	The pin output level is not inverted (active low when RSENTnCC.SOPC = 1)
		1	The pin output level is inverted (active high when RSENTnCC.SOPC = 0)
PFCn	Port function control register	—	Specify an alternative function of the pins. For details, see **Table 2.8, Alternative mode selection table**.
PFCEn	Port function control expansion register		
PFCAEn	Port function control addition expansion register		

26.4.5 Interrupts and Checks

The RSENT module provides two interrupt lines.

The successful fast channel receive interrupt notifies the CPU that the fast channel message reception buffer was updated and is holding a set of valid received data. Also, the reception status bit is set (RSENTnCS.FRS).

The status interrupt notifies the CPU that at least one of the error flags or message lost flags in the RSENTnCS register is set.

Whether a status flag in the RSENTnCS register is contributing to the generation of an interrupt event or not can be set individually.

The execution of the CRC checks can be disabled for the slow channel and fast channel individually. In case a check is disabled, the CRC of the received message is not checked and the related error flag is never set.

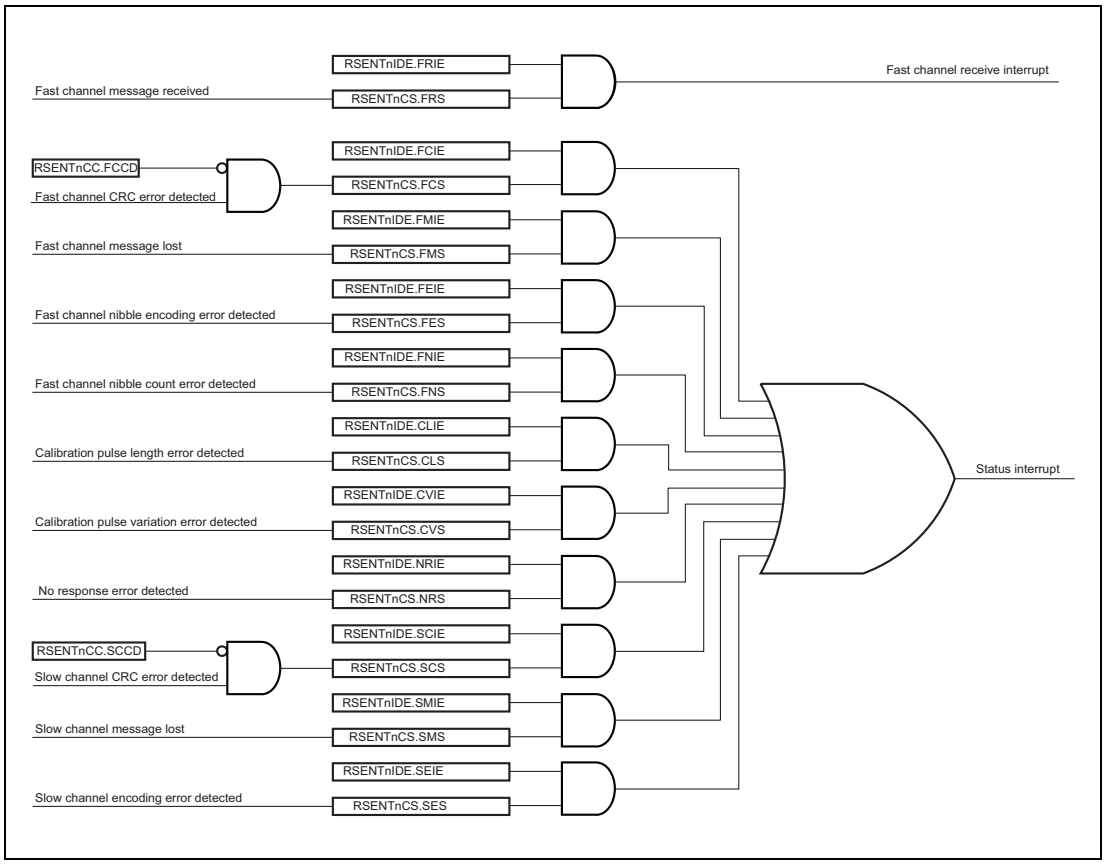


Figure 26.24 Interrupt Structure

Table 26.41, Status Flag Influence to Receive Buffer Behavior gives an overview about the relationship between set status flags and the buffer update.

Table 26.41 Status Flag Influence to Receive Buffer Behavior

RSENTnCS	Fast Channel Message Reception Buffer	Slow Channel Message Reception Buffer
FRS	Updated	Updated if all status and communications nibbles of slow channel messages are received and RSENTnCS.SES=0 and RSENTnCS.SCS=0
FCS	Not updated	Receive process aborted. Search for new start condition
FMS	Background buffer overwritten	Not impacted
FES	Not updated	Receive process aborted. Search for new start condition
FNS	Not updated	Receive process aborted. Search for new start condition
CLS	Not updated	Receive process aborted. Search for new start condition
CVS	Not updated	Receive process aborted. Search for new start condition
NRS	Not updated	Receive process aborted. Search for new start condition
SCS	Not impacted	Not updated
SMS	Not impacted	Message lost
SES	Not impacted	Receive process aborted (Refer to **Section 26.3.10, RSENTnCS — RSENT Communication Status Register** - RSENTnCS.SES (Slow Channel Encoding Error Status))

Section 27 Peripheral Sensor Interface 5 (PSI5)

The Peripheral Sensor Interface 5 (PSI5) is a function of interface the PSI5 (PSI5 v2.0) standard for sensor. PSI5 is implemented 4 channels.

27.1 Features PSI5 for RH850/U2A-EVA

27.1.1 Number of Channels

This microcontroller has the following number of PSI5 channels.

Table 27.1 Number of Channels

Product	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of channels	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	3 (n = 0, 2, 3)
Name	PSI5n									

Table 27.2 Index

Index	Description
n	Throughout this section, the individual channels of the PSI5 are identified by the index “n” (n = 0 to 3).

27.1.2 Register Base Addresses

PSI5n base addresses are listed in the following table. PSI5n register addresses are given as offsets from the base address in general.

Table 27.3 Register Base Addresses

Base address Name	Base Address	Bus Group
<PSI50_base>	FFD5 2000_H	P-Bus Group 3
<PSI51_base>	FFD5 2400_H	P-Bus Group 3
<PSI52_base>	FFD5 2800_H	P-Bus Group 3
<PSI53_base>	FFD5 $2C00_H$	P-Bus Group 3
<PSI5_SELB_base>	FFD5 3000_H	P-Bus Group 3

27.1.3 Clock Supply

Clock supply by and to PSI5n is listed in the following table.

Table 27.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
PSI5n	PCLK	CLK_HSB	Bus clock
	psi5_com_clk		Communication clock
PSI5_SELB	PCLK		Bus clock

For details of clock supply, see **Section 13, Clock Controller**.

27.1.4 Interrupt Requests and Error Notifications

The PSI5 can generate the interrupt requests shown in the following table.

Table 27.5 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt	DMA Trigger	DTS Trigger
PSI50				
INTPSI50SI*1	PSI50 status interrupt	600	—	—
INTPSI50RI	PSI50 receive interrupt	601	Group0-146	Group3-16
INTPSI50TI	PSI50 transfer interrupt	602	—	—
PSI51				
INTPSI51SI*1	PSI51 status interrupt	603	—	—
INTPSI51RI	PSI51 receive interrupt	604	Group0-147	Group3-17
INTPSI51TI	PSI51 transfer interrupt	605	—	—
PSI52				
INTPSI52SI*1	PSI52 status interrupt	606	—	—
INTPSI52RI	PSI52 receive interrupt	607	Group0-148	Group3-18
INTPSI52TI	PSI52 transfer interrupt	608	—	—
PSI53				
INTPSI53SI*1	PSI53 status interrupt	609	—	—
INTPSI53RI	PSI53 receive interrupt	610	Group0-149	Group3-19
INTPSI53TI	PSI53 transfer interrupt	611	—	—

Note 1. A set of multiple interrupts. See **Section 27.4.1, Interrupt**.

This module has no error notifications.

27.1.5 Reset Sources

PSI5n reset sources are shown below. PSI5n is initialized by the following reset sources.

Table 27.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PSI5n	All registers	√	√	√	√	√	√	—

27.1.6 External Input/Output Signals

The external input/output signals of the PSI5 are listed in the following table.

Table 27.7 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
PSI50			
psi5_rx_data	I	PSI50 data input	PSI50RX
psi5_tx_data	O	PSI50 data output	PSI50TX
PSI51			
psi5_rx_data	I	PSI51 data input	PSI51RX
psi5_tx_data	O	PSI51 data output	PSI51TX
PSI52			
psi5_rx_data	I	PSI52 data input	PSI52RX
psi5_tx_data	O	PSI52 data output	PSI52TX
PSI53			
psi5_rx_data	I	PSI53 data input	PSI53RX
psi5_tx_data	O	PSI53 data output	PSI53TX

27.2 Overview

27.2.1 Functional Overview

Among the main functions defined under the PSI5 standard (PSI5 v2.0), this IP supports the following:

- Communication Mode
  - PSI5-A: Asynchronous mode
  - PSI5-P: Synchronous parallel bus mode
  - PSI5-U: Synchronous parallel universal bus mode
  - PSI5-D: Synchronous daisy chain bus mode
  - PSI5-V: Variable time triggered synchronous operation mode
- Sensor to ECU Communication
  - Data can be received up to eight slots.
  - Bit rates: Low speed (125 kbps); High speed (189 kbps)
  - Automatic detection of start bits
  - Manchester to binary code conversion
  - Receivable bit length: 10 to 28 bits
  - Serial message frames can be received up to eight slots.
  - Automatic calculation of check bits for both data and serial message frames
  - Stores CRC and parity bits received with the data.
- ECU to Sensor Communication
  - Tooth Gap Mode
  - Pulse Width Mode
  - Automatic detection of start conditions
  - Frame formats 1 to 4 can be used
  - Automatic appending of synchronization bits
  - Automatic appending of CRCs
- PAS compatibility mode
  - 250 kbps
  - MSB First reception
  - Parity check
- Appending of a timestamp to receive data and serial message frames
  - Master and slave function for synchronization of timestamp

Table 27.8 List of Operation Modes

Operation Mode		Normal Mode	PAS Compatibility Mode
Bit rate		125 kbps / 189 kbps	250 kbps
Data word length		10 to 28 bits	8 to 24 bits
Data direction		LSB first	MSB first
Data transmission parameter			
	Value of bit time (TYP.)	8 µs / 5.3 µs	4 µs
	Value of gap time (MIN.)	8.4 µs / 5.6 µs	2 µs
Communication Mode		PSI5-A, PSI5-P, PSI5-U, PSI5-D, PSI5-V	PSI5-A only

27.2.2 Block Configuration

Figure 27.1 is a block diagram of PSI5n.

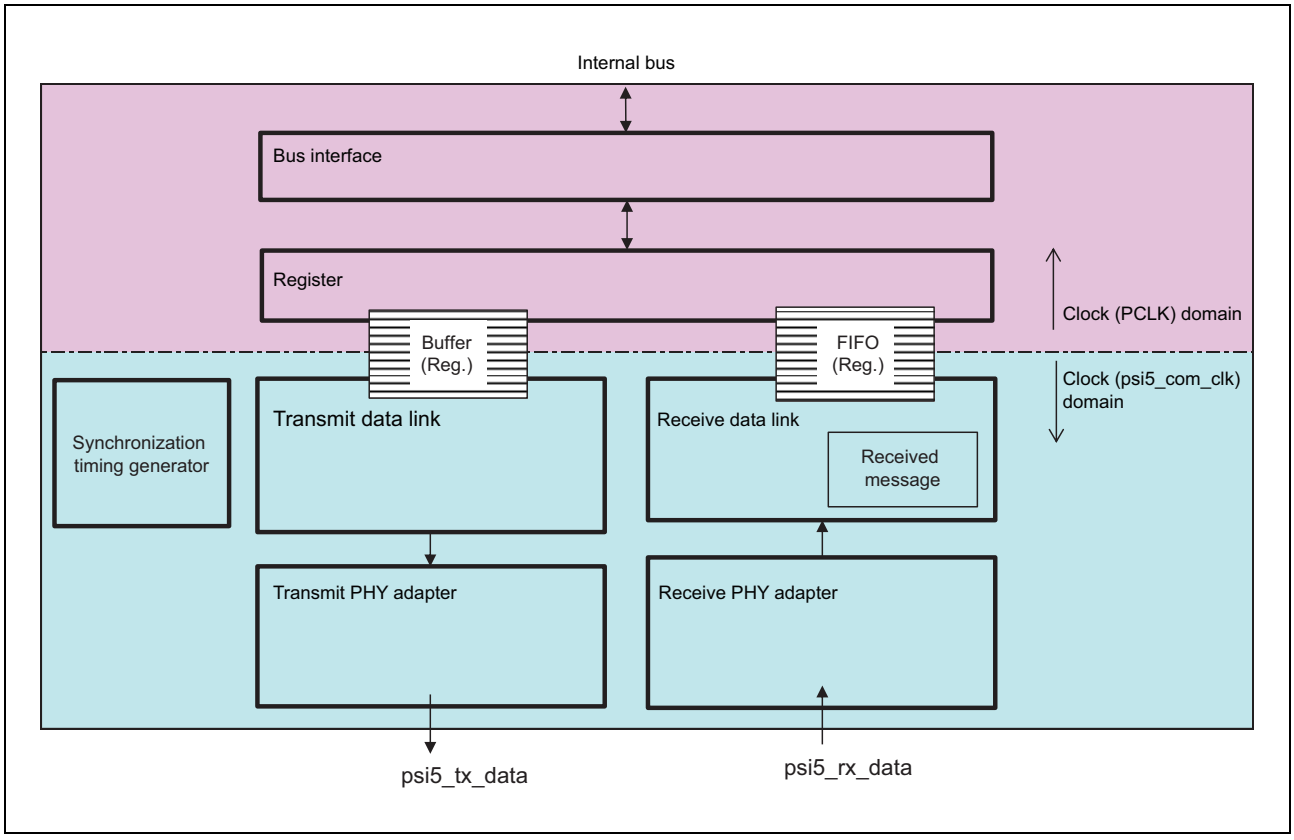


Figure 27.1 Block Diagram of PSI5

27.3 Registers

27.3.1 List of Registers

The PSI5n is controlled and operated by the registers listed in the table below.

For PSI5n base address <PSI5n_base>, see **Section 27.1.2, Register Base Addresses**.

Table 27.9 List of Registers (1/2)

Register Name	Symbol	Address	Access Protection	
			PBG	Other
PSI5 channel control register	PSI5nCHCTRL	<PSI5n_base> + 0000_H	*1	—
PSI5 IP timer control register	PSI5nIPTIMERCTRL	<PSI5n_base> + 0010_H	*1	—
PSI5 IP timer counter	PSI5nIPTIMER	<PSI5n_base> + 0014_H	*1	—
PSI5 operating mode/communication mode register	PSI5nOPMCOMM	<PSI5n_base> + 0020_H	*1	—
PSI5 operating-mode bit rate register	PSI5nOPMBITRATE	<PSI5n_base> + 0024_H	*1	—
PSI5 operating-mode cycle time register	PSI5nOPMCYCT	<PSI5n_base> + 0028_H	*1	—
PSI5 interrupt status register	PSI5nPSI5INT	<PSI5n_base> + 0030_H	*1	—
PSI5 receive data emulation register	PSI5nEMRXDATA	<PSI5n_base> + 0040_H	*1	—
PSI5 receive data status emulation register	PSI5nEMRXDST	<PSI5n_base> + 0044_H	*1	—
PSI5 receive data IP timer emulation register	PSI5nEMRXDTIM	<PSI5n_base> + 0048_H	*1	—
PSI5 receive data FIFO emulation register	PSI5nEMRXDFIFO	<PSI5n_base> + $004C_H$	*1	—
PSI5 receive-message receive message emulation register	PSI5nEMRXMRXMSG	<PSI5n_base> + 0050_H	*1	—
PSI5 receive-message channel receive status emulation register	PSI5nEMRXMRXST	<PSI5n_base> + 0054_H	*1	—
PSI5 receive-message channel receive timestamp emulation register	PSI5nEMRXMRXTIM	<PSI5n_base> + 0058_H	*1	—
PSI5 receive-message channel FIFO emulation register	PSI5nEMRXMFIFO	<PSI5n_base> + $005C_H$	*1	—
PSI5 transmission setting register	PSI5nTXSETTING	<PSI5n_base> + 0080_H	*1	—
PSI5 synchronization control register	PSI5nSYNCCTRL	<PSI5n_base> + 0084_H	*1	—
PSI5 transmission status register	PSI5nTXST	<PSI5n_base> + 0088_H	*1	—
PSI5 transmission status clear register	PSI5nTXSTCLR	<PSI5n_base> + $008C_H$	*1	—
PSI5 transmission status interrupt enable register	PSI5nTXSTINTEN	<PSI5n_base> + 0090_H	*1	—
PSI5 transmit data control register	PSI5nTXDCTRL	<PSI5n_base> + 0094_H	*1	—
PSI5 transmit data register	PSI5nTXDATA	<PSI5n_base> + 0098_H	*1	—
PSI5 receive sampling setting register	PSI5nRXSPLSET	<PSI5n_base> + 0100_H	*1	—
PSI5 receive slot 1 setting register	PSI5nRXS1SET	<PSI5n_base> + 0108_H	*1	—
PSI5 receive slot 2 setting register	PSI5nRXS2SET	<PSI5n_base> + $010C_H$	*1	—
PSI5 receive slot 3 setting register	PSI5nRXS3SET	<PSI5n_base> + 0110_H	*1	—
PSI5 receive slot 4 setting register	PSI5nRXS4SET	<PSI5n_base> + 0114_H	*1	—
PSI5 receive slot 5 setting register	PSI5nRXS5SET	<PSI5n_base> + 0118_H	*1	—
PSI5 receive slot 6 setting register	PSI5nRXS6SET	<PSI5n_base> + $011C_H$	*1	—
PSI5 receive slot 7 setting register	PSI5nRXS7SET	<PSI5n_base> + 0120_H	*1	—
PSI5 receive slot 8 setting register	PSI5nRXS8SET	<PSI5n_base> + 0124_H	*1	—
PSI5 receive data register	PSI5nRXDATA	<PSI5n_base> + 0128_H	*1	—
PSI5 receive data status register	PSI5nRXDST	<PSI5n_base> + $012C_H$	*1	—
PSI5 receive data IP timer register	PSI5nRXDTIM	<PSI5n_base> + 0130_H	*1	—
PSI5 receive data FIFO register	PSI5nRXDFIFO	<PSI5n_base> + 0134_H	*1	—

Table 27.9 List of Registers (2/2)

Register Name	Symbol	Address	Access Protection	
			PBG	Other
PSI5 receive module status register	PSI5nRXMODST	<PSI5n_base> + 0138_H	*1	—
PSI5 receive module status clear register	PSI5nRXMODSTCLR	<PSI5n_base> + $013C_H$	*1	—
PSI5 receive module status interrupt enable register	PSI5nRXMODSTINTEN	<PSI5n_base> + 0140_H	*1	—
PSI5 receive message channel setting register	PSI5nRXMSET	<PSI5n_base> + 0180_H	*1	—
PSI5 receive-message receive message register	PSI5nRXMRXMSG	<PSI5n_base> + 0184_H	*1	—
PSI5 receive-message channel receive status register	PSI5nRXMRXST	<PSI5n_base> + 0188_H	*1	—
PSI5 receive-message channel receive timestamp register	PSI5nRXMRXTIM	<PSI5n_base> + $018C_H$	*1	—
PSI5 receive-message channel FIFO register	PSI5nRXMFIFO	<PSI5n_base> + 0190_H	*1	—
PSI5 receive-message channel module status register	PSI5nRXMMST	<PSI5n_base> + 0194_H	*1	—
PSI5 receive-message channel module status clear register	PSI5nRXMMSTCLR	<PSI5n_base> + 0198_H	*1	—
PSI5 receive-message channel module status interrupt enable register	PSI5nRXMMSTINTEN	<PSI5n_base> + $019C_H$	*1	—
PSI5 timestamp function mode selection register	PSI5TSSEL	<PSI5_SELB_base> + 0000_H	PBG30#6	—

Note 1. n = 0: PBG30#2
n = 1: PBG30#3
n = 2: PBG30#4
n = 3: PBG30#5

27.3.2 PSI5nCHCTRL — PSI5 Channel Control Register

This register controls the channel operation.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0000_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.10 PSI5nCHCTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CHEN	Channel Enable 0: Channel disabled 1: Channel enabled In synchronous mode, the PSI5 starts to send a synchronization pulse after enabling the channel. In asynchronous mode, the PSI5 sets psi5_tx_data according to PSI5nTXSETTING.DEFTXVAL. When the channel is enabled, writing to the following registers is disabled: PSI5nOPMCOMM, PSI5nOPMBITRATE, PSI5nOPMCYCT, PSI5nTXSETTING, PSI5nRXSPLSET, PSI5nRXSmSET, and PSI5nRXMSET After writing to this bit, check that the setting is reflected by reading the bit by software.

CAUTION

The following registers must be set only when channel operation is enabled (PSI5nCHCTRL.CHEN = 1).

- PSI5nRXMODSTINTEN.RXDEXISTINTEN = 1
- PSI5nTXSTINTEN.TXDEMPTYINTEN = 1
- PSI5nTXDATA.TXDATA

27.3.3 PSI5nIPTIMERCTRL — PSI5 IP Timer Control Register

This register controls the IP timer.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0010_H

Value after reset: $0001\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTSLV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSCLR	—	—	—	—	—	—	—	IPTIMEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.11 PSI5nIPTIMERCTRL Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	MSTSLV	Specifies timer master/slave mode. 0: Slave mode is selected 1: Master mode is selected • In slave mode: The IP timer counts up according to the count-up timing signal (psi5_ts_tick_in) input through the master channel. The value of the PSI5nIPTIME register is cleared when the TSCLR bit is set or in response to the input of the clearing timing signal (psi5_ts_clr_in) through the master channel. • In master mode: IPTIMER counts up in response to the clock generated by the internal baud rate generator. PSI5 outputs the count-up timing signal (psi5_ts_tick_out). Setting the TSCLR bit to 1 clears the values in PSI5nIPTIMER (the clearing timing signal (psi5_ts_clr_out) is output).
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TSCLR	IP Timer Clear Trigger 0: The IP timer (PSI5nIPTIME register) is not cleared. 1: The IP timer (PSI5nIPTIME register) is cleared. The TSCLR bit setting is valid when the MSTSLV bit is set to 1. Read value is always 0.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IPTIMEN	IP Timer Enable 0: IP timer is disabled. 1: IP timer is enabled. In slave mode, set the IPTIMEN bit after enabling the IP timer of the master channel.

27.3.4 PSI5nIPTIMER — PSI5 IP Timer Counter

This is a counter for the IP timer.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0014_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COUNTVAL[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COUNTVAL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.12 PSI5nIPTIMER Register Contents

Bit Position	Bit Name	Function
31 to 0	COUNTVAL	In master mode, this timer counts up in response to the internal baud rate clock. In slave mode, this timer counts up by a tick pulse of the psi5_ts_tick_in signal. This value is used by the PSI5nRXDTIM register. This counter wrap around to 0 after reaching $FFFFFFFF_H$. When writing to this register, it needs several cycles until the write value is reflected because the write value should be transmitted to the synchronizer.

27.3.5 PSI5nOPMCOMM — PSI5 Operating Mode/Communication Mode Register

This register sets the communication mode.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0020_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	COMMODE		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 27.13 PSI5nOPMCOMM Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	COMMODE	Specify communication mode. 000_B: Asynchronous mode 001_B: Synchronous Parallel Bus Mode 010_B: Synchronous Universal Bus Mode 011_B: Synchronous Daisy Chain Bus Mode 100_B: Variable Time Triggered Synchronous Operation Mode Other than above: Setting prohibited

27.3.6 PSI5nOPMBITRATE — PSI5 Operating-Mode Bit Rate Register

This register sets a bit rate.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0024_H

Value after reset: $0000\ 029F_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BITRATECNT															
Value after reset	0	0	0	0	0	0	1	0	1	0	0	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.14 PSI5nOPMBITRATE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	BITRATECNT	Specify the 1-bit clock count considering communication clock tolerance. • For transfer at 80 MHz (12.5 ns) and 125 kbps (clock tolerance: 5%: 8.4 µs) psi5_com_clk is used as the counter clock. Counting cycle = 8.4 µs/12.5 ns = 672 = $2A0_H$ Setting value: $2A0\text{-}001_H = 29F_H$ This value is used to secure the bit length while no edge is detected in bit judgment.

27.3.7 PSI5nOPMCYCT — PSI5 Operating-Mode Cycle Time Register

This register sets a count value.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0028_H

Value after reset: $0000\ 9C3F_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	TTTTCNT			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTTTCNT															
Value after reset	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.15 PSI5nOPMCYCT Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	TTTTCNT	Specify the count value. Specify the count value for the SYNC signal's period. Use psi5_com_clk as the count clock. The default value is 500 μs when counting at 80 MHz (T = 12.5 ns). Count cycle: 500 μs / 12.5 ns = 40000 = $9C40_H$ Setting value: $9C40_H$-001_H = $9C3F_H$

27.3.8 PSI5nPSI5INT — PSI5 Interrupt Status Register

This register indicates the status of an interrupt that occurred in the PSI5.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0030_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	INT_SYNCED	INT_SYNCST	INT_TXDEMPTY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	INT_RXDSCNFERR	INT_RXDERR	INT_RXDFOVF	INT_RXDEXIST	INT_RXMERR	INT_RXMFOVF	INT_RXMEXIST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.16 PSI5nPSI5INT Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18	INT_SYNCED	INT_SYNCED interrupt generation 0: Was not generated 1: Generated
17	INT_SYNCST	INT_SYNCST interrupt generation 0: Was not generated 1: Generated
16	INT_TXDEMPTY	INT_TXDEMPTY interrupt generation 0: Was not generated 1: Generated
15 to 7	Reserved	When read, the value after reset is returned.
6	INT_RXDSCNFERR	INT_RXDSCNFERR interrupt generation 0: Was not generated 1: Generated
5	INT_RXDERR	INT_RXDERR interrupt generation 0: Was not generated 1: Generated
4	INT_RXDFOVF	INT_RXDFOVF interrupt generation 0: Was not generated 1: Generated
3	INT_RXDEXIST	INT_RXDEXIST interrupt generation 0: Was not generated 1: Generated
2	INT_RXMERR	INT_RXMERR interrupt generation 0: Was not generated 1: Generated
1	INT_RXMFOVF	INT_RXMFOVF interrupt generation 0: Was not generated 1: Generated
0	INT_RXMEXIST	INT_RXMEXIST interrupt generation 0: Was not generated 1: Generated

NOTE

For details about interrupts in the PSI5, see **Section 27.4, Operation**.

27.3.9 PSI5nEMRXDATA — PSI5 Receive Data Emulation Register

This is a mirror register of the receive data register (PSI5nRXDATA).

For details on PSI5nRXDATA, see **Section 27.3.26, PSI5nRXDATA — PSI5 Receive Data Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0040_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.17 PSI5nEMRXDATA Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDATA	Mirror data of PSI5nRXDATA. Data is not updated even if it is read.

27.3.10 PSI5nEMRXDST — PSI5 Receive Data Status Emulation Register

This is a mirror register of the receive data status register (PSI5nRXDST).

For details on PSI5nRXDST, see **Section 27.3.27, PSI5nRXDST — PSI5 Receive Data Status Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0044_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RXCHKD			RXSLOTNUM				—	—	—	RXSTATUS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.18 PSI5nEMRXDST Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10 to 8	RXCHKD	These bits act as mirrors of the PSI5nRXDST.RXCHKD bits. Data is not updated even if it is read.
7 to 4	RXSLOTNUM	These bits act as mirrors of the PSI5nRXDST.RXSLOTNUM bits. Data is not updated even if it is read.
3 to 1	Reserved	When read, the value after reset is returned.
0	RXSTATUS	This bit acts as a mirror of the PSI5nRXDST.RXSTATUS bit. Data is not updated even if it is read.

27.3.11 PSI5nEMRXDTIM — PSI5 Receive Data IP Timer Emulation Register

This is a mirror register of the receive data IP timer register (PSI5nRXDTIM).

For details on PSI5nRXDTIM, see **Section 27.3.28, PSI5nRXDTIM — PSI5 Receive Data IP Timer Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0048_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.19 PSI5nEMRXDTIM Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDTIM	Mirror data of the PSI5nRXDTIM register. Data is not updated even if it is read.

27.3.12 PSI5nEMRXDFIFO — PSI5 Receive Data FIFO Emulation Register

This is a mirror register of the receive data FIFO register (PSI5nRXDFIFO).

For details on PSI5nRXDFIFO, see **Section 27.3.29, PSI5nRXDFIFO — PSI5 Receive Data FIFO Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + $004C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.20 PSI5nEMRXDFIFO Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDST	Mirror data of the PSI5nRXDFIFO register. Data is not updated even if it is read.

27.3.13 PSI5nEMRXMRXMSG — PSI5 Receive-Message Receive Message Emulation Register

This is a mirror register of the receive-message receive message register (PSI5nRXMRXMSG).

For details on PSI5nRXMRXMSG, see **Section 27.3.34, PSI5nRXMRXMSG — PSI5 Receive-Message Receive Message Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0050_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CONFIGBIT	—	—	—	—	—	—	—	SERIALID							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATAFIELD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.21 PSI5nEMRXMRXMSG Register Contents

Bit Position	Bit Name	Function
31	CONFIGBIT	This bit acts as a mirror of the PSI5nRXMRXMSG.CONFIGBIT bit. Data is not updated even if it is read.
30 to 24	Reserved	When read, the value after reset is returned.
23 to 16	SERIALID	These bits act as mirrors of the PSI5nRXMRXMSG.SERIALID bits. Data is not updated even if it is read.
15 to 0	DATAFIELD	These bits act as mirrors of the PSI5nRXMRXMSG.DATAFIELD bits. Data is not updated even if it is read.

27.3.14 PSI5nEMRXMRXST — PSI5 Receive-Message Channel Receive Status Emulation Register

This is a mirror register of the receive-message channel receive status register (PSI5nRXMRXST).

For details on PSI5nRXMRXST, see **Section 27.3.35, PSI5nRXMRXST — PSI5 Receive-message Channel Receive Status Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0054_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RXSYNC		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXCRC						SLOTNUM				—	—	—	RXSTATUS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.22 PSI5nEMRXMRXST Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 16	RXSYNC	These bits act as mirrors of the PSI5nRXMRXST.RXSYNC bits. Data is not updated even if it is read.
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RXCRC	These bits act as mirrors of the PSI5nRXMRXST.RXCRC bits. Data is not updated even if it is read.
7 to 4	SLOTNUM	These bits act as mirrors of the PSI5nRXMRXST.SLOTNUM bits. Data is not updated even if it is read.
3 to 1	Reserved	When read, the value after reset is returned.
0	RXSTATUS	This bit acts as a mirror of the PSI5nRXMRXST.RXSTATUS bit. Data is not updated even if it is read.

27.3.15 PSI5nEMRXMRXTIM — PSI5 Receive-Message Channel Receive Timestamp Emulation Register

This is a mirror register of the receive-message channel receive timestamp register (PSI5nRXMRXTIM).

For details on PSI5nRXMRXTIM, see **Section 27.3.36, PSI5nRXMRXTIM — PSI5 Receive-message Channel Receive Timestamp Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0058_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXMTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXMTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.23 PSI5nEMRXMRXTIM Register Contents

Bit Position	Bit Name	Function
31 to 0	RXMTIM	Mirror data of the PSI5nRXMRXTIM register. Data is not updated even if it is read.

27.3.16 PSI5nEMRXMFIFO — PSI5 Receive-Message Channel FIFO Emulation Register

This is a mirror register of the receive-message channel FIFO register (PSI5nRXMFIFO).

For details on PSI5nRXMFIFO, see **Section 27.3.37, PSI5nRXMFIFO — PSI5 Receive-message Channel FIFO Register**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + $005C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXMFIFO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXMFIFO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.24 PSI5nEMRXMFIFO Register Contents

Bit Position	Bit Name	Function
31 to 0	RXMFIFO	Mirror data of the PSI5nRXMFIFO register. Data is not updated even if it is read.

27.3.17 PSI5nTXSETTING — PSI5 Transmission Setting Register

This register specifies transmission settings.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0080_H

Value after reset: $1067\ 47F7_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LONGCNT															
Value after reset	0	0	0	1	0	0	0	0	0	1	1	0	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PHY MODE	DEF TXVAL	SHORTCNT													
Value after reset	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.25 PSI5nTXSETTING Register Contents

Bit Position	Bit Name	Function
31 to 16	LONGCNT	Set the allowable upper limit of “long” synchronization pulse width when synchronous signal is output in the Pulse Width method. The upper limit when psi5_com_clk is at 80 MHz is 52.5 µs. Counting cycle: 52.5 µs / 12.5 ns = 4200 = 1068_H Setting value: 1068_H- 0001_H = 1067_H
15	PHYMODE	Specifies synchronous pulse PHY layer mode. 0: Tooth Gap method is selected. 1: Pulse Width method is selected. When the Pulse Width method is selected, the pulse width should comply with the value within the pulse width specified by the SHORTCNT bits in the Tooth Gap method.
14	DEFTXVAL	Specifies the initial value of synchronization pulse when data is not transmitted. 0: (0) is transmitted. 1: (1) is transmitted.
13 to 0	SHORTCNT	Set the allowable upper limit of “short” synchronization pulse width when synchronous signal is output in the Pulse Width or Tooth Gap method. The upper limit when psi5_com_clk is at 80 MHz is 25.5 µs. Counting cycle: 25.5 µs / 12.5 ns = 2040 = $07F8_H$ Setting value: $07F8_H$- 0001_H = $07F7_H$

CAUTION

- The PSI5nTXSETTING.SHORTCNT bit must be set to 1 or higher.
- The setting value below should be used.
  PSI5nOPMCYCT.TTTTCNT > PSI5nTXSETTING.LONGCNT > PSI5nTXSETTING.SHORTCNT

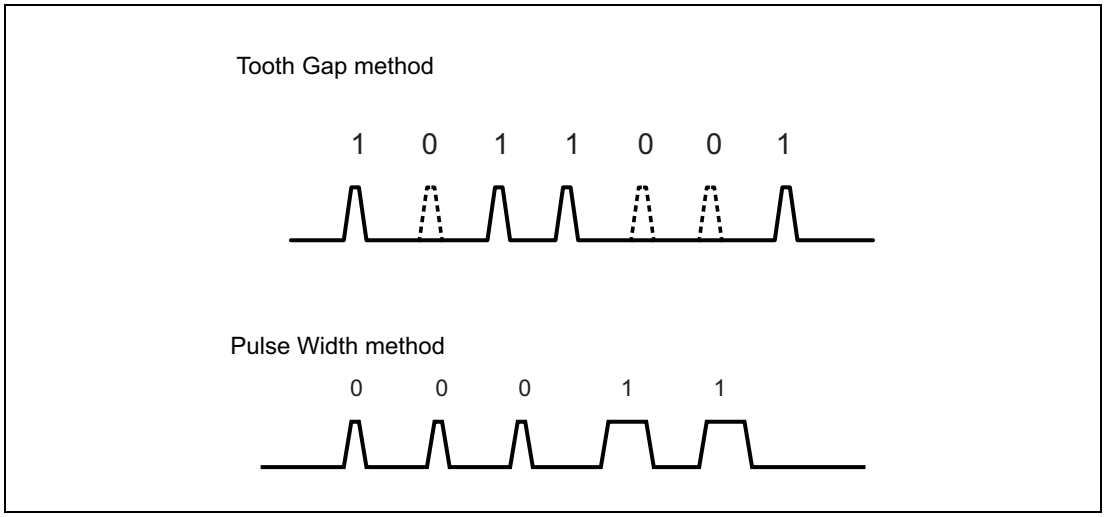


Figure 27.2 Pulse Width Method and Tooth Gap Method

27.3.18 PSI5nSYNCCTRL — PSI5 Synchronization Control Register

This register controls the start trigger in variable time triggered synchronous operation mode.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0084_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VALTIMSYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 27.26 PSI5nSYNCCTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	VALTIMSYNC	Synchronization signal send trigger in variable time triggered synchronous operation mode 0: Not effective. 1: The synchronization pulse is issued. This bit is always 0 when it is read. Do not write 1 to the VALTIMSYNC bit in any modes other than variable time triggered synchronous operation mode.

27.3.19 PSI5nTXST — PSI5 Transmission Status Register

This register indicates the transmission status.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0088_H

Value after reset: $0000\ 0001_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SYNCED	SYNCST	—	—	—	—	—	—	—	TXDEMPTY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.27 PSI5nTXST Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9	SYNCED	Indicates synchronization signal transmit complete. 0: Synchronization signal transmit is not started or is incomplete. 1: Synchronization signal transmit is complete. When the SYNCED bit is set, the INT_SYNCED interrupt is asserted. The SYNCED bit is set even when 0 is sent in the Tooth Gap method.
8	SYNCST	Indicates start of synchronization signal transmit 0: Synchronization signal is not sent. 1: Synchronization signal is started to send. When the SYNCST bit is set, the INT_SYNCST interrupt is asserted. The SYNCST bit is set even when 0 is sent in the Tooth Gap method.
7 to 1	Reserved	When read, the value after reset is returned.
0	TXDEMPTY	Indicates Tx data buffer empty status. 0: Transmit data buffer is not empty 1: Transmit data buffer is empty When the TXDEMPTY bit is set, the INT_TXDEMPTY interrupt is asserted.

27.3.20 PSI5nTXSTCLR — PSI5 Transmission Status Clear Register

This register clears the setting in the PSI5nTXST register.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + $008C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SYNCEDCLR	SYNCSTCLR	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 27.28 PSI5nTXSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	SYNCEDCLR	Clears the value of the PSI5nTXST.SYNCED bit. 0: No operation is done. The read value is always 0. 1: Clears the value of the PSI5nTXST.SYNCED bit
8	SYNCSTCLR	Clears the value of the PSI5nTXST.SYNCST bit. 0: No operation is done. The read value is always 0. 1: Clears the value of the PSI5nTXST.SYNCST bit.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

27.3.21 PSI5nTXSTINTEN — PSI5 Transmission Status Interrupt Enable Register

This register controls transmission status interrupts.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0090_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SYNCEDINTEN	SYNCSTINTEN	—	—	—	—	—	—	—	TXDEMPTYINTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 27.29 TXSINTEN Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	SYNCED INTEN	Controls the INT_SYNCED interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled
8	SYNCSTINTEN	Controls the INT_SYNCST interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TXDEMPTYINTEN	Controls the INT_TXDEMPTY interrupt. 0: Interrupt disabled (Masked) 1: Interrupt enabled

27.3.22 PSI5nTXDCTRL — PSI5 Transmit Data Control Register

This register controls transmit data.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0094_H

Value after reset: $0000\ 0001_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FRMFORMAT		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 27.30 PSI5nTXDCTRL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	FRMFORMAT	Specify the Tx frame format. 001_B: Frame 1 (Short) 010_B: Frame 2 (Long: 4-bit data or 8-bit data) 011_B: Frame3 (XLong) 100_B: Frame4 (XXLong) Other than above: Setting prohibited

27.3.23 PSI5nTXDATA — PSI5 Transmit Data Register

This register stores transmit data.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0098_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXDATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXDATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.31 PSI5nTXDATA Register Contents

Bit Position	Bit Name	Function
31 to 0	TXDATA	Tx Data Write Tx data without the Start field, Synchronization bit, and CRC field (Tx data is written in LSB first). Data arrangement differs depending on the transmit frame format specified in the PSI5nTXDCTRL.FRMFORMAT bit. (See **Table 27.32**.) The data is sent after these bits are written. Read value is always 0.

Table 27.32 PSI5nTXDATA.TXDATA Data Arrangement in Transmit Frame Format

TXDATA Bit Position	Transmit Frame Format (PSI5nTXDCTRL.FRMFORMAT)									
	Frame1 “Short”		Frame2 “Long” (4-Bit Data Nibbles)		Frame2 “Long” (8-Bit Data Word)		Frame3 “XLong”		Frame4 “XXLong”	
31 to 24	All 0		All 0		All 0		All 0		All 0	
23									Data	D19
22										D18
21							Data	D7		D17
20								D6		D16
19								D5		D15
18								D4		D14
17								D3		D13
16								D2		D12
15			Data	D3	Data	D7		D1		D11
14				D2		D6		D0		D10
13				D1		D5	RAdr	X7		D9
12				D0		D4		X6		D8
11			RAdr	X5		D3		X5		D7
10				X4		D2		X4		D6
9				X3		D1		X3		D5
8				X2		D0		X2		D4
7				X1	RAdr	X1		X1		D3
6				X0		X0		X0		D2
5	FC	F2	FC	F2	FC	F2	FC	F2		D1
4		F1		F1		F1		F1		D0
3		F0		F0		F0		F0	SAdr	C
2	SAdr	A2	SAdr	A2	SAdr	A2	SAdr	A2		A2
1		A1		A1		A1		A1		A1
0		A0		A0		A0		A0		A0

27.3.24 PSI5nRXSPLSET — PSI5 Receive Sampling Setting Register

This register controls the sampling timing for received data.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0100_H

Value after reset: $0000\ 0027_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMPLPROD							
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.33 PSI5nRXSPLSET Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	SMPLPROD	Specify the value of the sampling interval counter for received data. This counter is counted up based on the psi5_com_clk signal. Specify a 16 times sampling for 1-bit length. psi5_com_clk: 80 MHz (T = 12.5 ns), Bit rate: L (125 kHz, T = 8 µs), Count cycle: 500 ns / 12.5 ns = 40 = 28_H Setting value: 28_H- 01_H = 27_H. Example settings when psi5_com_clk = 80 MHz

Baud Rate	Setting Value
125 kHz	27_H
189 kHz	19_H
250 kHz	13_H

27.3.25 PSI5nRXSmSET — PSI5 Receive Slot m Setting Register (m = 1 to 8)

These registers set the receive slots.

Access: This register can be read or written in 32-bit units.

Address: PSI5nRXS1SET: <PSI5n_base> + 0108_H, PSI5nRXS2SET: <PSI5n_base> + $010C_H$,
PSI5nRXS3SET: <PSI5n_base> + 0110_H, PSI5nRXS4SET: <PSI5n_base> + 0114_H,
PSI5nRXS5SET: <PSI5n_base> + 0118_H, PSI5nRXS6SET: <PSI5n_base> + $011C_H$,
PSI5nRXS7SET: <PSI5n_base> + 0120_H, PSI5nRXS8SET: <PSI5n_base> + 0124_H

Value after reset: $00A0\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SLTEN	—	—	—	—	PAS CMP	ERR DET	LENGTH					OFFSETCNT			
Value after reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFFSETCNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.34 PSI5nRXSmSET Register Contents

Bit Position	Bit Name	Function
31	SLTEN	Slot Enable 0: Slot m is disabled. 1: Slot m is enabled.
30 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	PASCMP	Specifies PAS compatibility mode. 0: PSI5 mode is selected. (LSB first) 1: PAS compatibility mode is selected. (MSB first)
25	ERRDET	Specifies the error detection scheme. 0: 1-bit parity 1: 3-bit CRC
24 to 20	LENGTH	Specify the data length
19 to 0	OFFSETCNT	Specify the count value of the offset timer. When the count value reaches the specified value, the PSI5 starts to receive the data in Slot m. The value counted by psi5_com_clk is used. • Rx module starts to find start bits (S1, S2) after expiration of the offset counter. A margin should be included taking clock uncertainty, sampling rate, and inter-slot period into consideration. For example, Start time: 44 µs psi5_com_clk: 80 MHz (T = 12.5 ns) Offset value: 44 us / 12.5 ns = 3520 = $0DC0_H$ OFFSETCNT: $0DC0_H$- 0001_H = $0DBF_H$

CAUTION

In asynchronous mode (PSI5nOPMCOMM.COMMODE = 000_B), only PSI5nRXS1SET is available.

The ERRDET must always be set to 0 in PAS compatibility mode (PASCMP = 1).

27.3.26 PSI5nRXDATA — PSI5 Receive Data Register

This register is the receive data register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0128_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.35 PSI5nRXDATA Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDATA	Received Data These bits hold received data that does not include a start bit and CRC/parity bits (received data is stored in LSB first). These bits are effective only when PSI5nRXMODST.RXDEXIST = 1.

27.3.27 PSI5nRXDST — PSI5 Receive Data Status Register

This register indicates the status of received data.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + $012C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RXCHKD			RXSLOTNUM				—	—	—	RXSTATUS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.36 PSI5nRXDST Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10 to 8	RXCHKD	Raw Data of CRC or Parity In parity mode, setting is as below: RXCHKD[2:1]: 00_B RXCHKD[0]: Parity value These bits are effective only when PSI5nRXMODST.RXDEXIST bit = 1
7 to 4	RXSLOTNUM	Indicate the slot number of received PSI5nRXDATA.RXDATA. The slot number of the first received data is 1 (the slot number is from 1 to 8). These bits are effective only when PSI5nRXMODST.RXDEXIST = 1.
3 to 1	Reserved	When read, the value after reset is returned.
0	RXSTATUS	Indicates the status of Rx data. 0: No error. 1: CRC, parity, or syntax error occurred. Syntax error includes the following three types: – An illegal start bit – Data shorter than PSI5nRXSmSET.LENGTH (A long data error cannot be detected.) – Manchester code error

27.3.28 PSI5nRXDTIM — PSI5 Receive Data IP Timer Register

This register indicates the IPTIMER value in received data.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0130_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.37 PSI5nRXDTIM Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDTIM	Specify the PSI5nIPTIMER value when the last data (CRC or Parity) is received. These bit fields are effective only when the PSI5nRXMODST.RXDEXIST bit is set to 1.

27.3.29 PSI5nRXDFIFO — PSI5 Receive Data FIFO Register

This register is the receive data FIFO register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0134_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXDST															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.38 PSI5nRXDFIFO Register Contents

Bit Position	Bit Name	Function
31 to 0	RXDST	These bits are used for DMA. These bits can be read after data is received (when PSI5nRXMODST.RXDEXIST= 1 and the INT_RXDEXIST interrupt is asserted). This register acts as FIFO whose content includes PSI5nRXDATA, PSI5nRXDST, and PSI5nRXDTIM. Read the receive data three times each. Data is read in the following order: PSI5nRXDATA, PSI5nRXDST, and then PSI5nRXDTIM. Exclusive access is required between PSI5nRXDFIFO and “PSI5nRXDATA, PSI5nRXDST, or PSI5nRXDTIM”.

27.3.30 PSI5nRXMODST — PSI5 Receive Module Status Register

This register is the status register for the receive module.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0138_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RXDSCNFERR	—	—	—	—	—	—	—	RXDERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXDFOVF	—	—	—	—	—	—	—	RXDEXIST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.39 PSI5nRXMODST Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24	RXDSCNFERR	Indicates collision errors of synchronization pulse and reception of sensor data. 0: No error has occurred. 1: An error has occurred. When this bit is set, the INT_RXDSCNFERR interrupt is asserted.
23 to 17	Reserved	When read, the value after reset is returned.
16	RXDERR	Indicates the status of receive data. 0: No error. 1: CRC, parity, or syntax error occurred. This bit is set by PSI5nRXDST.RXSTATUS bit. When this bit is set, the INT_RXDERR interrupt is asserted.
15 to 9	Reserved	When read, the value after reset is returned.
8	RXDFOVF	Receive Data FIFO Overflow Flag 0: No overflow. 1: Overflow occurred. When this bit is set, the INT_RXDFOVF interrupt is asserted.
7 to 1	Reserved	When read, the value after reset is returned.
0	RXDEXIST	Indicates whether or not Rx data exists. 0: No Rx data exists. 1: Rx data exists. When the PSI5nRXDATA, PSI5nRXDST, and PSI5nRXDTIM registers are read, this bit is cleared. When this bit is set, the INT_RXDEXIST interrupt is asserted.

27.3.31 PSI5nRXMODSTCLR — PSI5 Receive Module Status Clear Register

This register is the status clear register for the receive module.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + $013C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RXDSCNFERRCLR	—	—	—	—	—	—	—	RXDERRCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXDFOVFCLR	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 27.40 PSI5nRXMODSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	RXDSCNFERRCLR	Clears the PSI5nRXMODST.RXDSCNFERR bit. 0: No operation is done. 1: The RXDSCNFERR bit is cleared. Read value is always 0.
23 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RXDERRCLR	Clears the PSI5nRXMODST.RXDERR bit. 0: No operation is done. 1: The RXDERR bit is cleared. Read value is always 0.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	RXDFOVFCLR	Clears the PSI5nRXMODST.RXDFOVF bit. 0: No operation is done. 1: The RXDFOVF bit is cleared. Read value is always 0.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

27.3.32 PSI5nRXMODSTINTEN — PSI5 Receive Module Status Interrupt Enable Register

This register is the status interrupt enable register for the receive module.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0140_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RXDSCNFERRINTEN	—	—	—	—	—	—	—	RXDERRINTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXDFOVFINTEN	—	—	—	—	—	—	—	RXDEXISTINTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.41 PSI5nRXMODSTINTEN Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	RXDSCNFERRINTEN	INT_RXDSCNFERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled
23 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RXDERRINTEN	INT_RXDERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	RXDFOVFINTEN	INT_RXDFOVF Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RXDEXISTINTEN	INT_RXDEXIST Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled

27.3.33 PSI5nRXMSET — PSI5 Receive Message Channel Setting Register

This register sets the channels for the receive module.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0180_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RXM8EN	RXM7EN	RXM6EN	RXM5EN	RXM4EN	RXM3EN	RXM2EN	RXM1EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.42 PSI5nRXMSET Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	RXM8EN	Controls the messaging channel for slot 8. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
6	RXM7EN	Controls the messaging channel for slot 7. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
5	RXM6EN	Controls the messaging channel for slot 6. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
4	RXM5EN	Controls the messaging channel for slot 5. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
3	RXM4EN	Controls the messaging channel for slot 4. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
2	RXM3EN	Controls the messaging channel for slot 3. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
1	RXM2EN	Controls the messaging channel for slot 2. 0: Messaging channel is disabled. 1: Messaging channel is enabled.
0	RXM1EN	Controls the messaging channel for slot 1. 0: Messaging channel is disabled. 1: Messaging channel is enabled.

27.3.34 PSI5nRXMRXMSG — PSI5 Receive-Message Receive Message Register

This register is the receive message register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0184_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CONFIGBIT	—	—	—	—	—	—	—	SERIALID							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATAFIELD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.43 PSI5nRXMRXMSG Register Contents

Bit Position	Bit Name	Function
31	CONFIGBIT	Configuration Bit in the Received Serial Data Frame See **Figure 27.3**.
30 to 24	Reserved	When read, the value after reset is returned.
23 to 16	SERIALID	Serial ID in the Received Serial Data Frame See **Figure 27.3**.
15 to 0	DATAFIELD	Data Field in the Received Serial Data Frame See **Figure 27.3**.

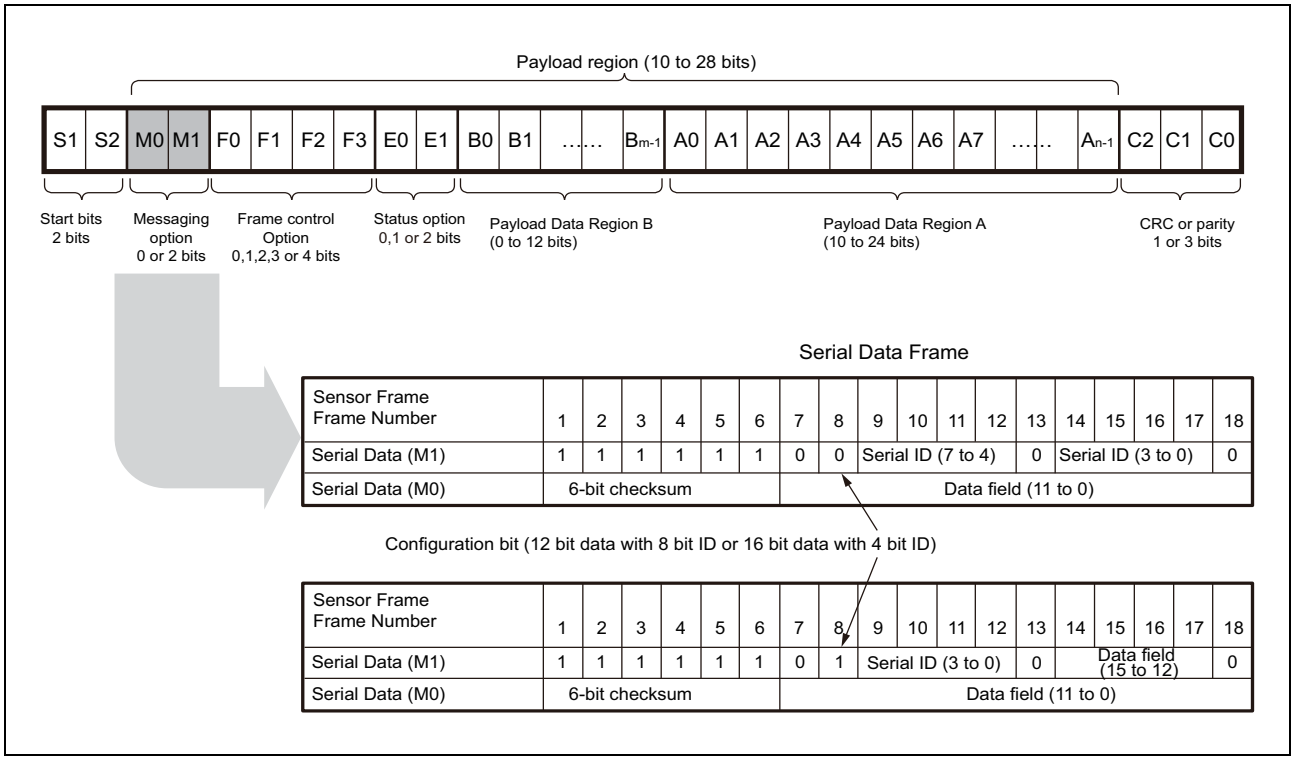


Figure 27.3 Serial Data Frame Configured by the Two Messaging Bits of the Sensor Data Frame (Messaging Channel)

27.3.35 PSI5nRXMRXST — PSI5 Receive-message Channel Receive Status Register

This register is the receive status register for the receive-message channel.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0188_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RXSYNC		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXCRC						SLOTNUM				—	—	—	RXSTATUS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.44 PSI5nRXMRXST Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 16	RXSYNC	Received Synchronization Bit Raw Data (the frame numbers 7, 13, and 18 of the sensor frame) For the sensor frame, see **Figure 27.3**.
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RXCRC	Received CRC Raw Data
7 to 4	SLOTNUM	Received Serial Data Frame's Slot Number
3 to 1	Reserved	When read, the value after reset is returned.
0	RXSTATUS	Indicates the error state of Rx data. 0: No error. 1: CRC or Syntax error occurred. A syntax error also occurs when the receive synchronization bit has received a value other than 0 while RXSYNC = 000_B.

27.3.36 PSI5nRXMRXTIM — PSI5 Receive-message Channel Receive Timestamp Register

This register indicates the value of the receive-message channel receive timestamp.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + $018C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXMTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXMTIM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.45 PSI5nRXMRXTIM Register Contents

Bit Position	Bit Name	Function
31 to 0	RXMTIM	Indicate the PSI5nIPTIMER value when the last data (frame number 18 of the sensor frame) is received. For the sensor frame, see **Figure 27.3**. These bits are effective only when PSI5nRXMMST.RXMEXIST = 1.

27.3.37 PSI5nRXMFIFO — PSI5 Receive-message Channel FIFO Register

This register indicates the receive-message channel FIFO information.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0190_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RXMFIFO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXMFIFO															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.46 PSI5nRXMFIFO Register Contents

Bit Position	Bit Name	Function
31 to 0	RXMFIFO	These bits can be read after data is received (when PSI5nRXMMST.RXMEXIST= 1: INT_RXMEXIST interrupt is asserted). This register acts as FIFO whose content includes PSI5nRXMRXMSG, PSI5nRXMRXST, and PSI5nRXMRXTIM. Read the received data three times each. Data is read in the following order: PSI5nRXMRXMSG, PSI5nRXMRXST, and then PSI5nRXMRXTIM. Exclusive access is required between PSI5nRXMFIFO and “PSI5nRXMRXMSG, PSI5nRXMRXST, or PSI5nRXMRXTIM”.

27.3.38 PSI5nRXMMST — PSI5 Receive-message Channel Module Status Register

This register is the status register for the receive-message channel module.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PSI5n_base> + 0194_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXM ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXM FOVF	—	—	—	—	—	—	—	RXM EXIST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.47 PSI5nRXMMST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	RXMERR	Rx Message Error 0: No error. 1: Error occurred. When this bit is set, the INT_RXMERR interrupt is asserted.
15 to 9	Reserved	When read, the value after reset is returned.
8	RXMFOVF	Rx Message FIFO Overflow Flag 0: No overflow. 1: Overflow occurred. When this bit is set, the INT_RXMFOVF interrupt is asserted.
7 to 1	Reserved	When read, the value after reset is returned.
0	RXMEXIST	Indicates whether Rx Serial Data Frame exists. 0: No Rx serial data exists. 1: Rx serial data exists. When the PSI5nRXMRXMSG, PSI5nRXMRXST, and PSI5nRXMRXTIM registers are read, this bit is cleared. When this bit is set, the INT_RXMEXIST interrupt is asserted.

27.3.39 PSI5nRXMMSTCLR — PSI5 Receive-message Channel Module Status Clear Register

This register is used to clear the receive-message channel module status register.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + 0198_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXMERRCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXMFOVFCLR	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 27.48 PSI5nRXMMSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RXMERRCLR	Clears the status of the PSI5nRXMMST.RXMERR bit. 0: No operation is done. 1: The RXMERR bit is cleared. Read value is always 0.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	RXMFOVFCLR	Clears the status of the PSI5nRXMMST.RXMFOVF bit. 0: No operation is done. 1: The RXMFOVF bit is cleared. Read value is always 0.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

27.3.40 PSI5nRXMMSTINTEN — PSI5 Receive-message Channel Module Status Interrupt Enable Register

This register controls interrupts from the receive-message channel module.

Access: This register can be read or written in 32-bit units.

Address: <PSI5n_base> + $019C_H$

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXM ERR INTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RXM FOVF INTEN	—	—	—	—	—	—	—	RXM EXIST INTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 27.49 PSI5nRXMMSTINTEN Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RXMERR INTEN	INT_RXMERR Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	RXMFOVF INTEN	INT_RXMFOVF Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RXMEXIST INTEN	INT_RXMEXIST Interrupt Enable 0: Interrupt disabled (Masked) 1: Interrupt enabled

27.3.41 PSI5TSSEL — PSI5 Timestamp Function Mode Selection Register

This register specifies the timestamp setting.

This register can be controlled when the PSI5 is stopped (PSI5nCHCTRL.CHEN = 0).

Access: This register can be read or written in 32-bit units.

Address: <PSI5_SELB_base> + 0000_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset source except module reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PSI5MSSEL3 [2:0]			—	PSI5MSSEL2 [2:0]			—	PSI5MSSEL1 [2:0]			—	PSI5MSSEL0 [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 27.50 PSI5TSSEL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 12	PSI5MSSEL3 [2:0]	PSI53 Timestamp Master Selection. This selection is valid only when PSI53 is in slave mode (PSI53IPTIMERCTRL.MSTSLV bit = 0_B) 000_B: No timestamp master. 001_B: PSI50 is the timestamp master of PSI53 (Supplies the count clock from PSI50 to PSI53). 010_B: PSI51 is the timestamp master of PSI53 (Supplies the count clock from PSI51 to PSI53). 011_B: PSI52 is the timestamp master of PSI53 (Supplies the count clock from PSI52 to PSI53). 100_B: No timestamp master. Other than above: Setting prohibited.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	PSI5MSSEL2 [2:0]	PSI52 Timestamp Master Selection. This selection is valid only when PSI52 is in slave mode (PSI52IPTIMERCTRL.MSTSLV bit = 0_B) 000_B: No timestamp master. 001_B: PSI50 is the timestamp master of PSI52 (Supplies the count clock from PSI50 to PSI52). 010_B: PSI51 is the timestamp master of PSI52 (Supplies the count clock from PSI51 to PSI52). 011_B: No timestamp master. 100_B: PSI53 is the timestamp master of PSI52 (Supplies the count clock from PSI53 to PSI52). Other than above: Setting prohibited.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 27.50 PSI5TSSEL Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	PSI5MSSEL1 [2:0]	PSI51 Timestamp Master Selection. This selection is valid only when PSI51 is in slave mode (PSI51IPTIMERCTRL.MSTSLV bit = 0_B) 000_B: No timestamp master. 001_B: PSI50 is the timestamp master of PSI51 (Supplies the count clock from PSI50 to PSI51). 010_B: No timestamp master. 011_B: PSI52 is the timestamp master of PSI51 (Supplies the count clock from PSI52 to PSI51). 100_B: PSI53 is the timestamp master of PSI51 (Supplies the count clock from PSI53 to PSI51). Other than above: Setting prohibited.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PSI5MSSEL0 [2:0]	PSI50 Timestamp Master Selection. This selection is valid only when PSI50 is in slave mode (PSI50IPTIMERCTRL.MSTSLV bit = 0_B) 000_B: No timestamp master. 001_B: No timestamp master. 010_B: PSI51 is the timestamp master of PSI50 (Supplies the count clock from PSI51 to PSI50). 011_B: PSI52 is the timestamp master of PSI50 (Supplies the count clock from PSI52 to PSI50). 100_B: PSI53 is the timestamp master of PSI50 (Supplies the count clock from PSI53 to PSI50). Other than above: Setting prohibited.

27.4 Operation

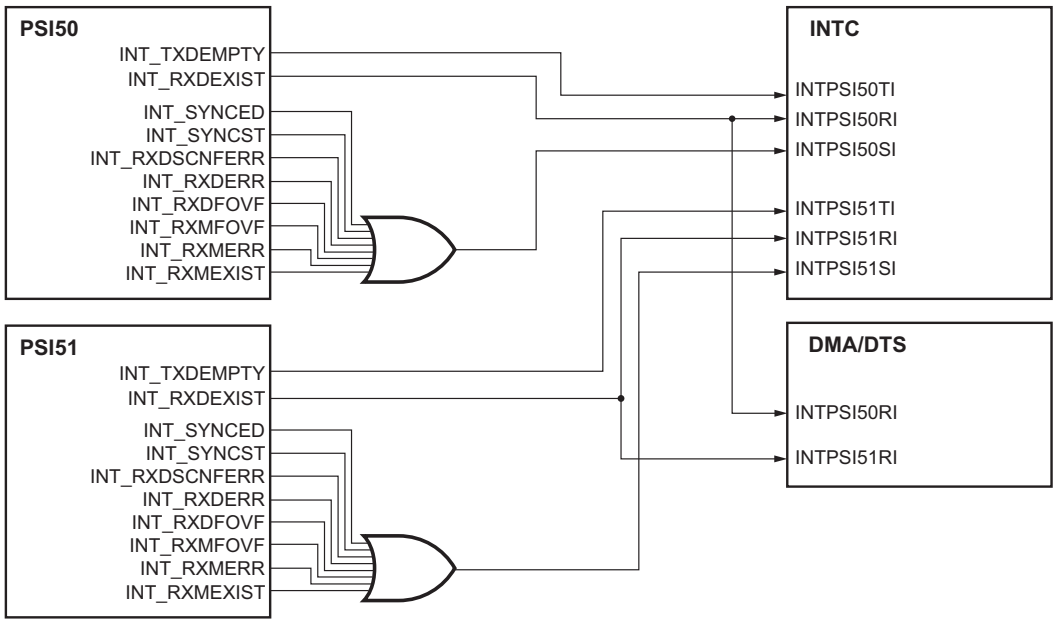
27.4.1 Interrupt

The PSI5 has 10 interrupt signals.

Table 27.51 PSI5 Interrupt Signals and Corresponding Registers

Interrupt Signal	Function	Source Register Bit	Interrupt Enable Register Bit
INT_SYNCED	Synchronization end interrupt	PSI5nTXST. SYNCED	PSI5nTXSTINTEN. SYNCEDINTEN
INT_SYNCST	Synchronization start interrupt	PSI5nTXST. SYNCST	PSI5nTXSTINTEN. SYNCSTINTEN
INT_TXDEMPTY	Transmit data empty interrupt	PSI5nTXST. TXDEMPTY	PSI5nTXSTINTEN. TXDEMPTYINTEN
INT_RXDSCNFERR	Synchronization pulse and receive data conflict error interrupt	PSI5nRXMODST. RXDSCNFERR	PSI5nRXMODSTINTEN. RXDSCNFINTEN
INT_RXDERR	Receive data error interrupt	PSI5nRXMODST. RXDERR	PSI5nRXMODSTINTEN. RXDERRINTEN
INT_RXDFOVF	Receive data FIFO overflow interrupt	PSI5nRXMODST. RXDFOVF	PSI5nRXMODSTINTEN. RXDFOVFINTEN
INT_RXDEXIST	Receive data exist interrupt	PSI5nRXMODST. RXDEXIST	PSI5nRXMODSTINTEN. RXDEXISTINTEN
INT_RXMFOVF	Receive message FIFO overflow interrupt	PSI5nRXMMST. RXMFOVF	PSI5nRXMMSTINTEN. RXMFOVFINTEN
INT_RXMERR	Receive message error interrupt	PSI5nRXMMST. RXMERR	PSI5nRXMMSTINTEN. RXMERRINTEN
INT_RXMEXIST	Receive message exist interrupt	PSI5nRXMMST. RXMEXIST	PSI5nRXMMSTINTEN. RXMEXISTINTEN

The following figure shows the relationship between the PSI5 interrupt signals, and RH850/U2A-EVA's interrupt controller INTC and DMA.



Note 1. Description of PSI52 and PSI53 are omitted (same configuration as PSI50 and PSI51).

Figure 27.4 PSI5 Interrupt Signals

27.4.2 Setting Operation Mode

Before the PSI5n starts operating, initial settings including the sensor connection status and operation mode must be configured. The following describes necessary initial settings.

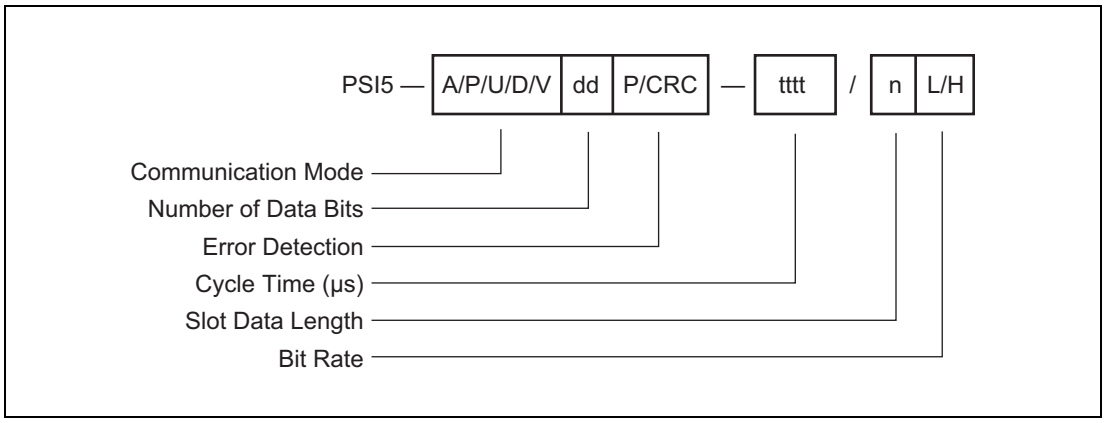


Figure 27.5 Setting PSI5 Operation Mode

Table 27.52 Setting Operation Mode

Item		Description
Communication Modes	A	Asynchronous Mode
	P	Synchronous Parallel Bus Mode
	U	Synchronous Universal Bus Mode
	D	Synchronous Daisy Chain Bus Mode
	V	Variable Time Triggered Synchronous Operation Mode
Number of data bits	dd	Data length of each slot specified in the PSI5nRXSmSET.LENGTH bit (10 to 28 bits)
Error Detection	P	One Parity Bit
	CRC	Three Bits CRC
Cycle time	tttt	Cycle Time (Minimum cycle time allowed in variable time triggered synchronous operation mode) (µs)
Slot count/cycle (n)	n	Select the slot to be used in the PSI5nRXMSET.
Bit Rate	L	125 kbps
	H	189 kbps

27.4.2.1 Setting for Transmit/Receive Mode

Set PSI5nOPMCOMM, PSI5nOPMBITRATE, and PSI5nOPMCYCT to the value corresponding to the specification of the PS15 sensor and its connection status.

27.4.2.2 Setting for Data Reception

The RH850/U2A-EVA receives data on psi5_com_clk.

A 16-times sampling is used for bit detection.

The bit length depends on the setting of PSI5nRXSPLSET.SMPLPROD[7:0].

- PSI5nRXSPLSET.SMPLPROD[7:0]: Set the sampling interval.

Set the value counted by psi5_com_clk. The RH850/U2A-EVA uses a 16-times sampling.

27.4.2.3 Initial Setting for Asynchronous Mode

In asynchronous mode, only PSI5nRXS1SET is available. Writing 1 to SLTEN in PSI5nRXS2SET to PSI5nRXS8SET is prohibited (so disable them).

In asynchronous mode, set the SLTEN, PASCMP, ERRDET, and LENGTH fields in PSI5nRXS1SET. The OFFSETCNT field is not available.

27.4.2.4 Initial Setting for Synchronous Mode

In synchronous mode, specify all fields of PSI5nRXS1SET to PSI5nRXS8SET appropriately.

The recommended setting values for PSI5nRXS1SET.OFFSETCNT and PSI5nRXS2SET to PSI5nRXS8SET.OFFSETCNT can be obtained as the results of the formula in **Figure 27.6** and **Figure 27.7**. In these figures, tcom_clk indicates the time for one psi5_com_clk cycle. Other variables are the variables specified by the PSI5 standard.

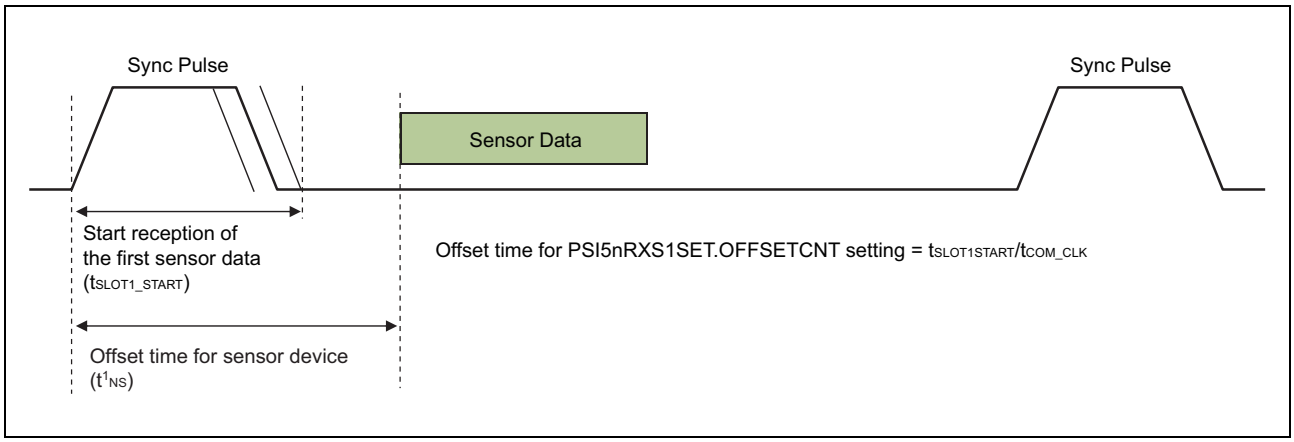


Figure 27.6 Recommended Setting of PSI5nRXS1SET.OFFSETCNT

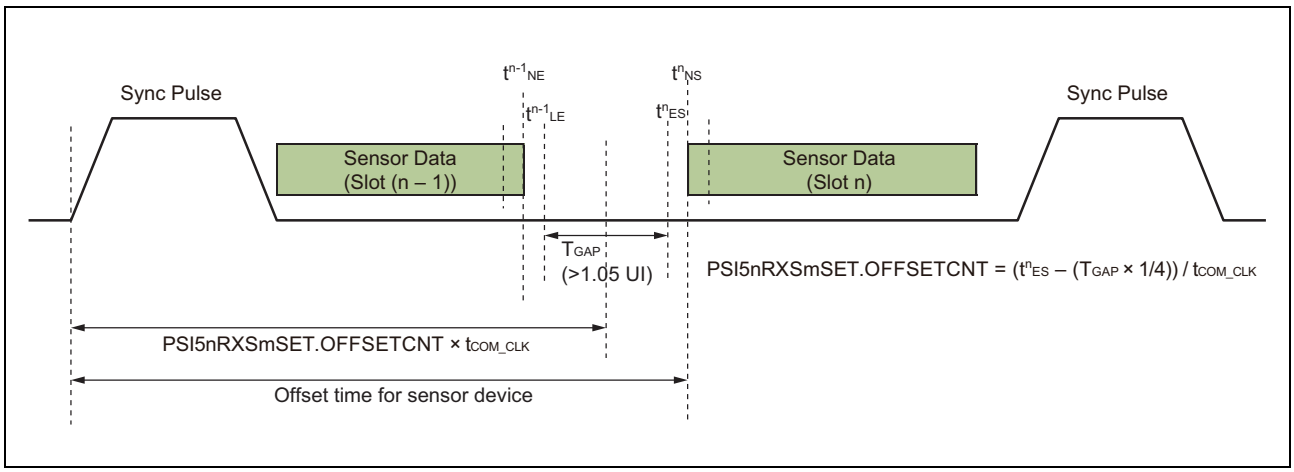


Figure 27.7 Recommended Setting of PSI5nRXSmSET.OFFSETCNT (m = 2 to 8)

27.4.2.5 Setting for Serial Message Reception

To receive a serial message, write 1 to PSI5nRXMSET. RXMmEN (m = 1 to 8) in the corresponding slot.

27.4.2.6 Initial Setting in Asynchronous Mode

In asynchronous mode, registers for transmission are disables except PSI5nTXSETTING.DEFTXVAL fields. Use the value after reset.

The PSI5 outputs the value set in PSI5nTXSETTING. DEFTXVAL after the start of operation (although the standard does not define output data in asynchronous mode). The value after reset is 1.

27.4.2.7 Initial Setting in Synchronous Mode

Set the value in PSI5nTXSETTING.

PSI5nTXSETTING.DEFTXVAL sets a synchronization pulse value for the period during which data is not transmitted. In Tooth GAP mode, set PSI5nTXSETTING.DEFTXVAL to 1. In pulse width mode, the value is not specified in the standard. Set the value specified by a system designer.

In variable time triggered synchronous operation mode, the user must specify the timing for synchronization pulse output.

When issuing a synchronization pulse, write 1 to the PSI5nSYNCCTRL.VALTIMSYNC bits.

Set the data format to be transmitted to the sensor in PSI5nTXDCTRL.FRMFORMAT.

27.4.2.8 Setting for Timestamping

A timestamp is appended to the received data so that the order of the received data and messages can be identified.

Timestamping is available in two modes: master mode for generation of the timing for counting up within the PSI5, and slave mode for counting with an input signal from the master. In master mode, the count-up timing is output to the output signal psi5_ts_tick_out. In slave mode, the input signal psi5_ts_tick_in is used for counting up.

To use a timestamp, set PSI5nIPTIMERCTRL.IPTIMEN to 1. Specify master or slave in PSI5nIPTIMERCTRL.MSTSLV.

In master mode, the count-up timing is complete when the internal baud rate counter is complete. The value to complete the baud rate counter is the setting of PSI5nOPMBITRATE.BITRATECNT.

The timestamp can be cleared by writing 1 to PSI5nIPTIMERCTRL.TSCLR irrespective of master or slave. In master mode, the clearing timing is output to the output signal psi5_ts_clr_out. In slave mode, the input signal psi5_ts_clr_in from the master can be used to clear the timestamp.

(1) Setting of Master/Slave Mode in Timestamp Mode

For PSI5, the dedicated timestamp slaves for a master can be chosen by various setting of PSI5TSSEL register, while multi-masters or master/slave(s) relation is decided by the setting of PSI5nIPTIMERCTRL.MSTSLV of each PSI5 unit beforehand.

Setting examples:

PSI52 operates as master and PSI50, PSI53 operate as slaves

PSI50IPTIMERCTRL.MSTSLV = 0_B

PSI51IPTIMERCTRL.MSTSLV = Don’t care

PSI52IPTIMERCTRL.MSTSLV = 1_B

PSI53IPTIMERCTRL.MSTSLV = 0_B

PSI5TSSEL.PSI5MSSEL0[2:0] = 011_B

PSI5TSSEL.PSI5MSSEL1[2:0] = 000_B (Initial value)

PSI5TSSEL.PSI5MSSEL2[2:0] = 011_B

PSI5TSSEL.PSI5MSSEL3[2:0] = 011_B

PSI50, PSI51, PSI52 and PSI53 operate in master mode without slaves

PSI50IPTIMERCTRL.MSTSLV = 1_B

PSI51IPTIMERCTRL.MSTSLV = 1_B

PSI52IPTIMERCTRL.MSTSLV = 1_B

PSI53IPTIMERCTRL.MSTSLV = 1_B

PSI5TSSEL.PSI5MSSEL0[2:0] = 001_B

PSI5TSSEL.PSI5MSSEL1[2:0] = 010_B

PSI5TSSEL.PSI5MSSEL2[2:0] = 011_B

PSI5TSSEL.PSI5MSSEL3[2:0] = 100_B

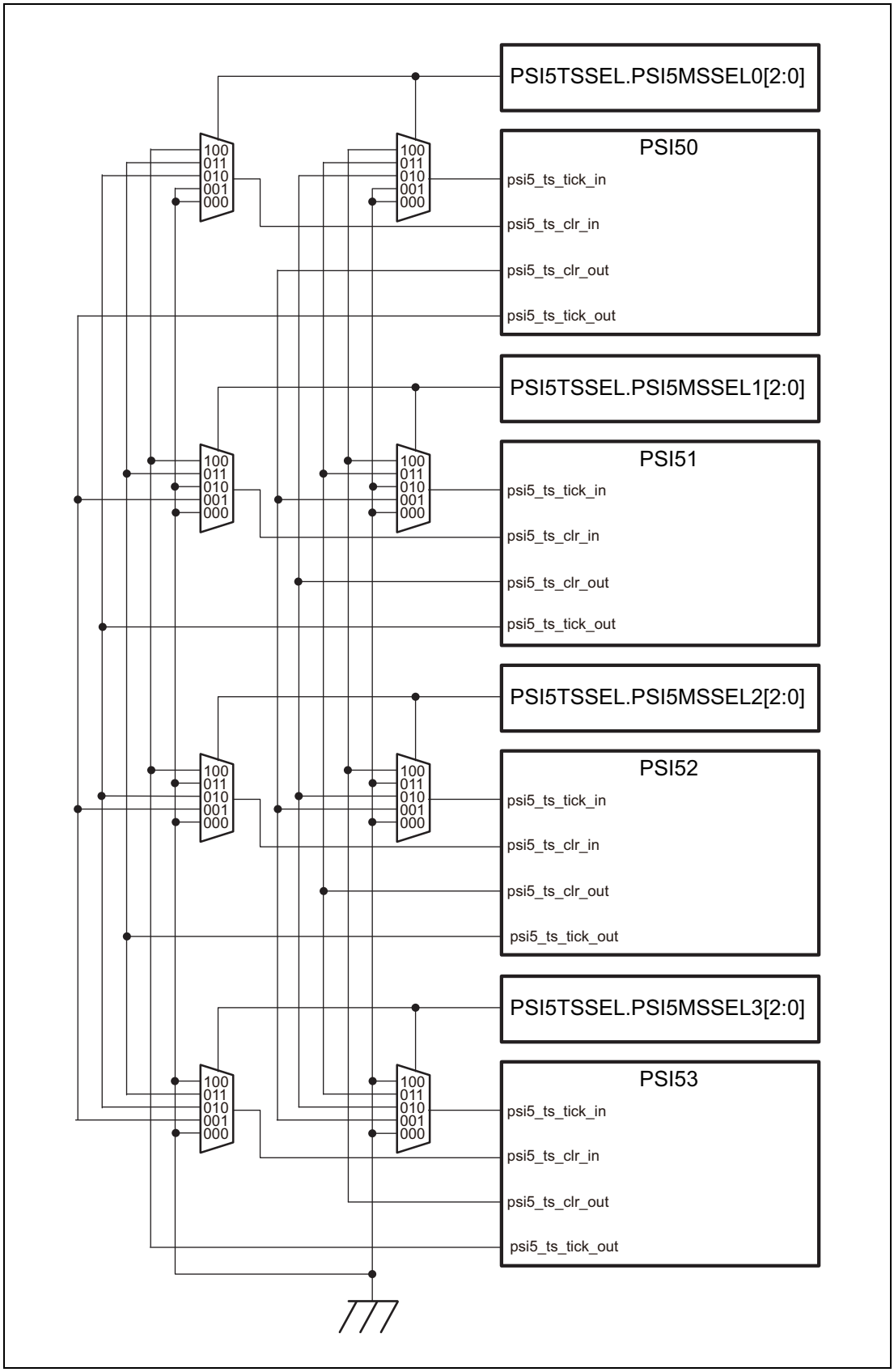


Figure 27.8 Relationship Between PSI50, PSI51, PSI52 and PSI53

27.4.3 Operation Flow

27.4.3.1 Starting Operation

Figure 27.10 shows the flow from initialization to the start of operation. When resetting the sensor or releasing it from reset, issue instructions to the PHY as these operations are performed by the PHY.

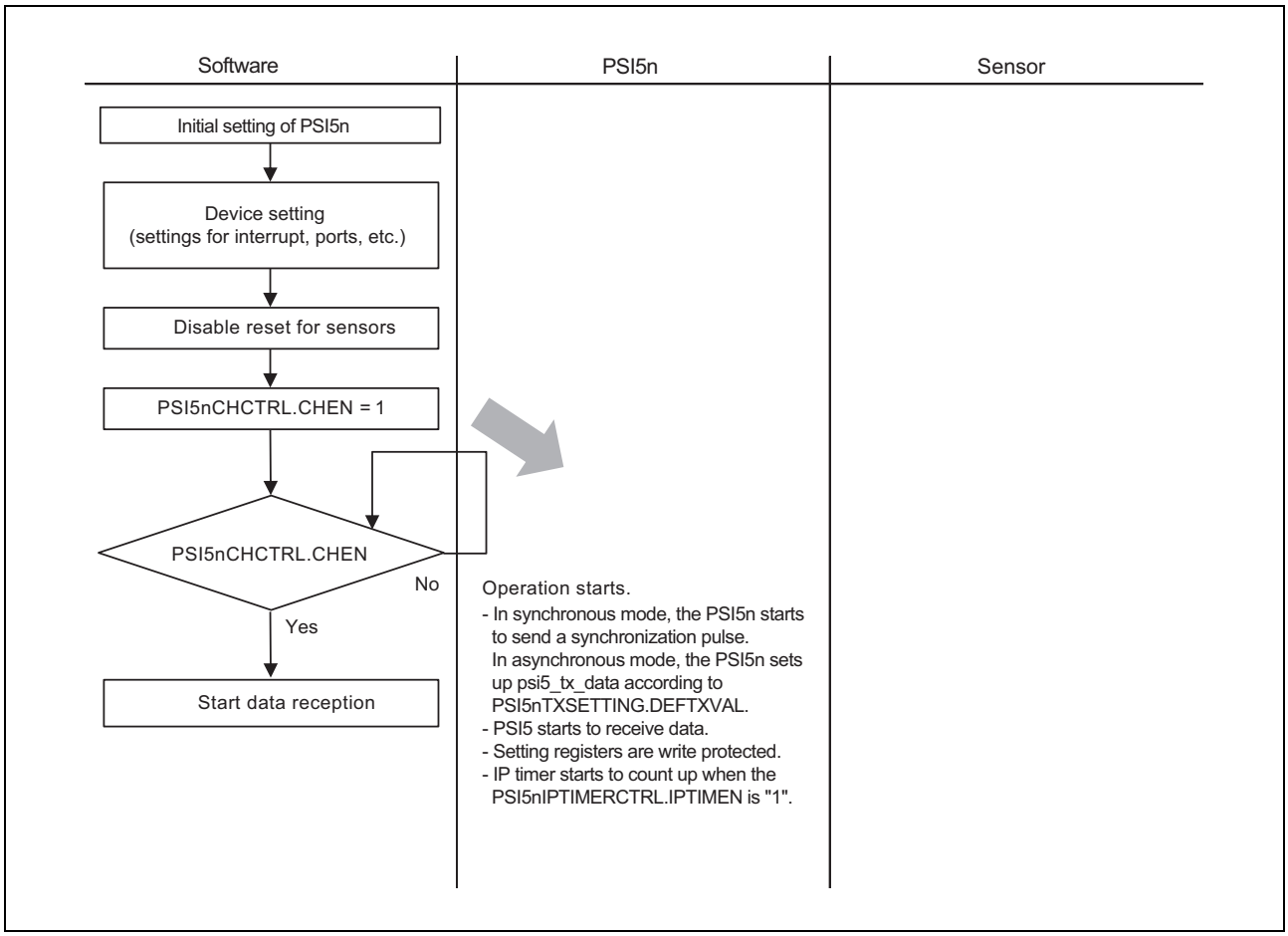


Figure 27.10 Flow of Starting Operation

27.4.3.2 Flow of Data Reception

Figure 27.11 shows the flow of data reception. The interrupt enable (PSI5nRXMODSTINTEN.RXDEXISTINTEN = 1) in the flowchart is not required when polling is used for detection of reception, or when it is always enabled (this applies to enabling of other interrupts in the subsequent sections).

The received data without the start bit and CRC/parity bits can be read out from PSI5nRXDATA.RXDATA.

Regarding the FIFO buffer shown in **Figure 27.11**, when data are received, that state is retained, and even if the register is empty, the received data are set from the FIFO buffer to the register. In cases of overflow because of delays in the reading out of data, new data are written to the buffer.

When reading data from the FIFO buffer, check that the value of PSI5nRXMDST.RXDEXIST is 0. Since PSI5nRXMDST.RXDEXIST being 1 indicates that data which have not been read out remain in the FIFO buffer, repeat the processing for reading.

The PSI5nRXDST. RXSTATUS is set to 1 in response to a start-bit error, Manchester code error, CRC error, or parity error during data reception. Specifically, in cases where multiple errors are detected during the reception of a single frame, the FIFO buffer for received data is updated every time an error occurs.

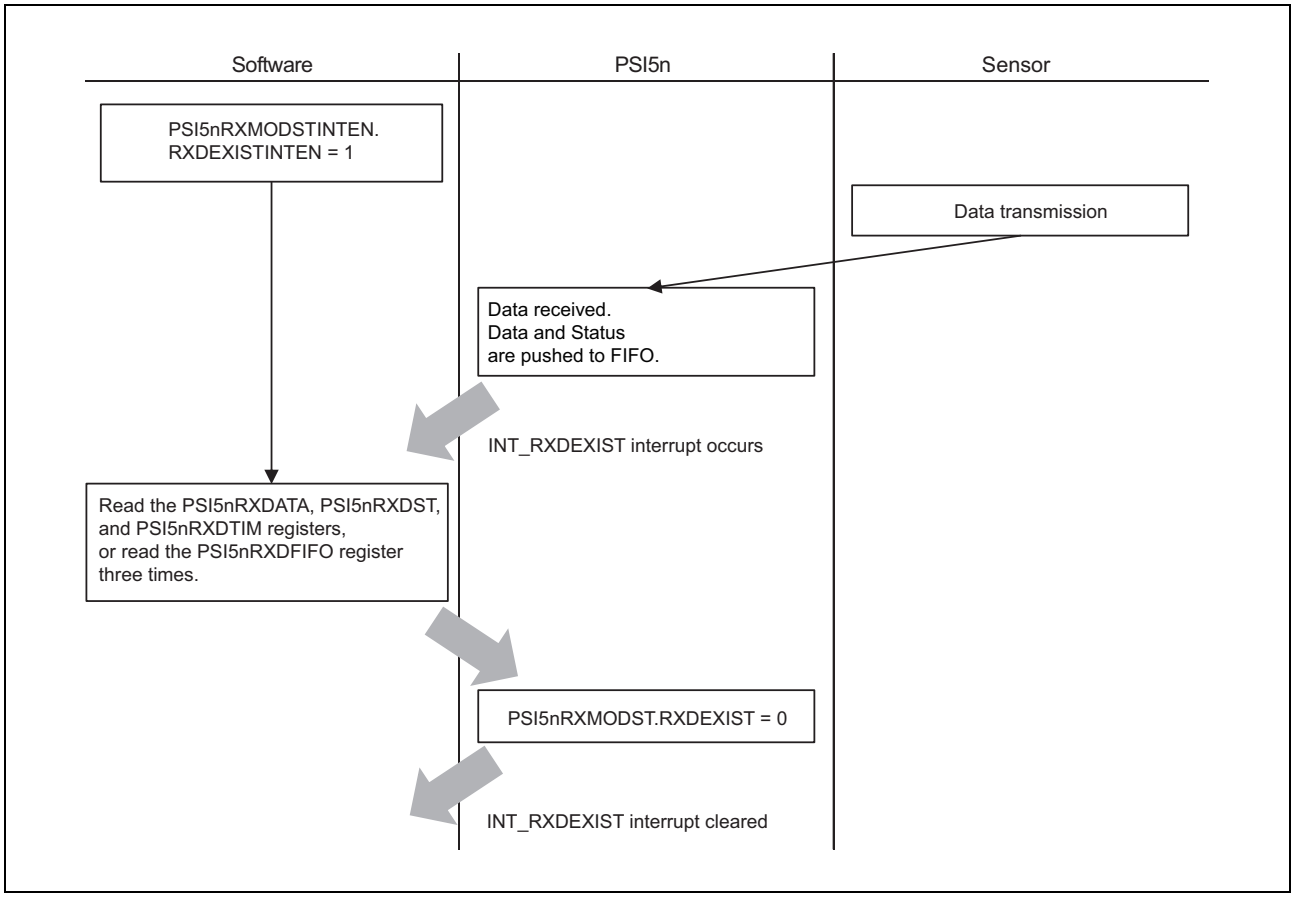


Figure 27.11 Data Reception Flow

27.4.3.3 Flow of Data Transmission

Figure 27.12 shows the flow of data transmission.

Transmit data arrangement to be written to the PSI5nTXDATA.TXDATA differs depending on the transmit frame format specified in the PSI5nTXDCTRL.FRMFORMAT bit (See **Table 27.32**). The data must not include a start bit, synchronization bit, or CRC (these are automatically appended).

Once PSI5nTXDCTRL.FRMFORMAT has been set, it does not need to be set again after that.

The PSI5 module does not have a FIFO buffer for transmission, so processing for transmission is sequential. When you intend to write to the transmission buffer, read the state of the buffer empty flag to check if it currently indicates the buffer empty state (PSI5nTXST.TXDEMPTY = 1). If the transmission buffer holds data to be transmitted at the time of writing (PSI5nTXST.TXDEMPTY = 0), transmitted data become undefined when data are written to the transmission buffer.

PSI5nTXST.TXDEMPTY becomes 1 to indicate the completion of transmission when reading of the last bit of the data from the transmission buffer starts.

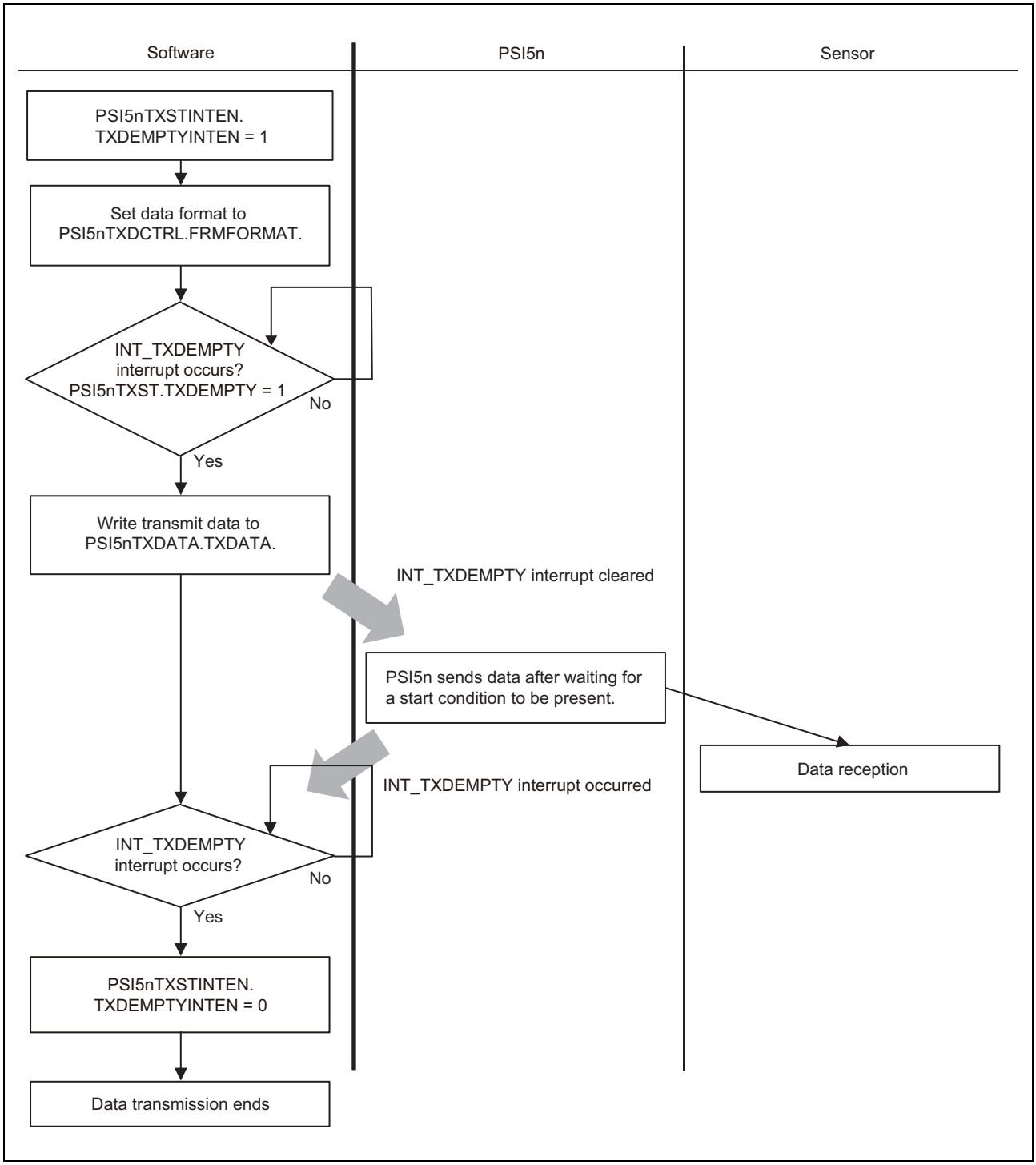


Figure 27.12 Data Transmission Flow

27.4.3.4 Serial Message Reception Flow

Figure 27.13 shows the flow of serial message reception.

The received serial message can be read out from PSI5nRXMRXMSG.

In **Figure 27.13**, the FIFO buffer holds the serial message and its state and the received data are set from the FIFO buffer to the register if the register is empty. In cases of overflow because of delays in the reading out of data, new data are written to the buffer.

When reading data from the FIFO buffer, check that the value of PSI5nRXMMST.RXMEXIST is 0. Since PSI5nRXMMST.RXMEXIST being 1 indicates that data which have not been read out remain in the FIFO buffer, repeat the processing for reading.

If either a CRC error or a receive data error occurs after the detection of the start bit in the message (after reception of M1 (Frame No.) = 111111_B), PSI5nRXMRXST.RXSTATUS is set to 1. In this case, handling of the received message is system-dependent. When an receive data error occurs, the message up to that data is stored in order from the least-significant bit of PSI5nRXMRXMSG.

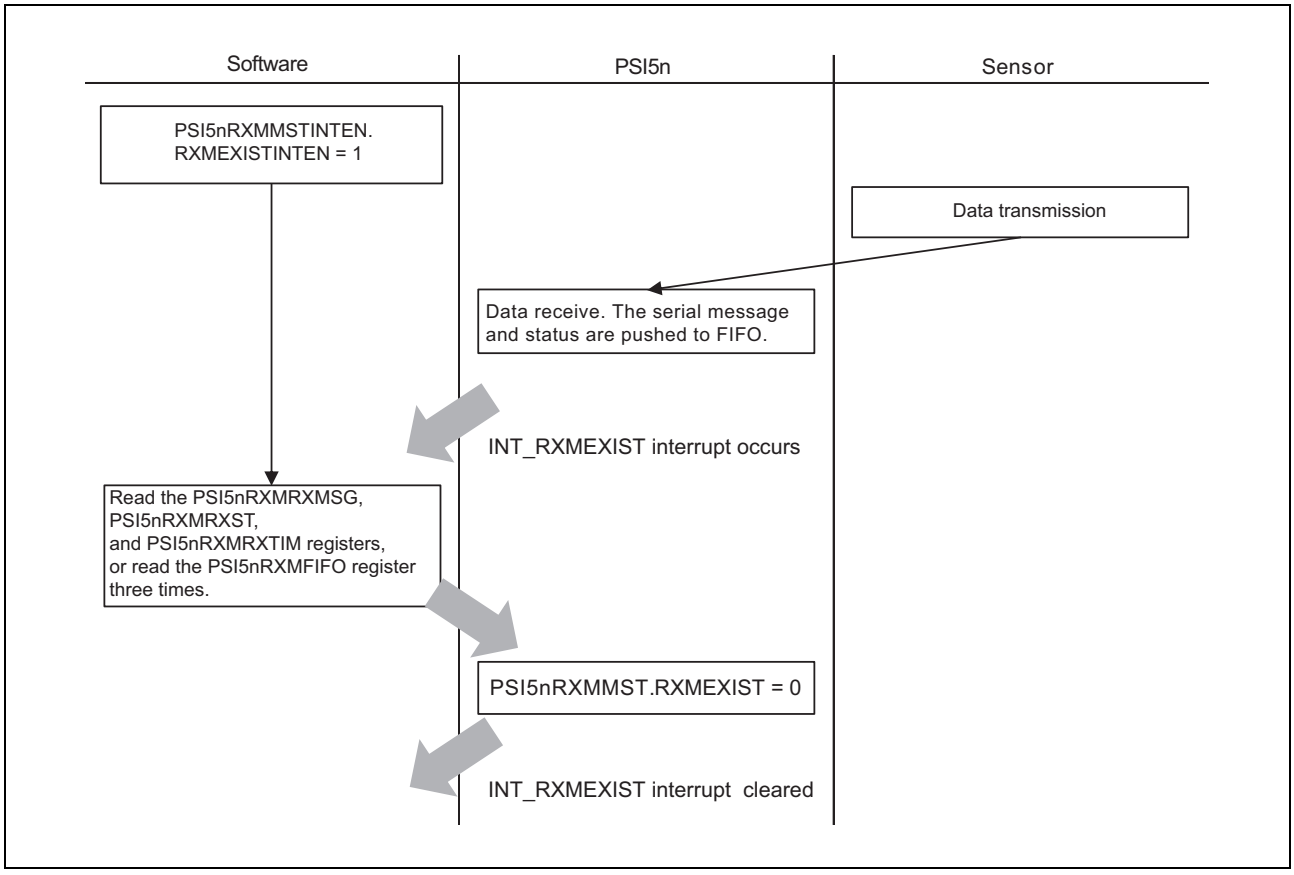


Figure 27.13 Serial Message Reception Flow

27.4.3.5 Parity and CRC Errors in Received Data

Error flags (PSI5nRXMODST.RXDERR and PSI5nRXMMST.RXMERR) are set in response to the detection of parity or CRC errors in received data. Reception operations proceed whether or not errors are detected.

Each of the error flags has a corresponding clearing bit (PSI5nRXMODSTCLR.RXDERRCLR and PSI5nRXMMSTCLR.RXMERRCLR).

27.4.3.5.1 Operation after Error Detection in Asynchronous Mode and PAS Compatibility Mode

When external noise and so on lead to errors in the Manchester code during the reception of data while the PSI5 is in the asynchronous mode or PAS compatibility mode, reception of the data being received is assumed to be complete, the parity or CRC error flag is set, and the data are stored in the reception FIFO buffer. Even when a Manchester code error occurs, the next data reception starts immediately. Due to this specification, even if a Manchester code error occurs for one bit because of external noise during data reception, the following data bit is assumed to be the start bit, and reception starts. In this situation, multiple Manchester code errors may occur and multiple reception complete interrupts can be generated. In such cases, the maximum number of errors corresponds to the number of bits in the unit of data.

Accordingly, this raises the possibility of producing an overflow. If an overflow occurs even though the data retrieval assumed during system designing is performed, the transfer may be malfunctioning due to factors such as noise.

In synchronous mode, data reception is halted at slot n + 1 in response to an error in the Manchester code while reception in slot n is in progress. For this reason, even if further errors occur or data is resent while in slot n, they are ignored.

27.4.4 PAS Compatibility Mode

The RH850/U2A-EVA has a PAS compatibility mode. The PAS compatibility mode supported by the RH850/U2A-EVA is as shown in **Table 27.53**.

Table 27.53 Supported PAS Compatibility Mode

Operation Mode		PAS Compatibility Mode
Bit rate		250 kbps
Data word length		8 to 24 bits
Data direction		MSB first
Data transmission parameter		
	Value of bit time (TYP.)	4 µs
	Value of gap time (MIN.)	2 µs
Communication Mode		PSI5-A only

In PAS compatibility mode, set 1 in PSI5nRXS1SET.PASCMP. At that time, set 0 (parity) in PSI5nRXS1SET.ERRDET.

27.4.5 Baud Rate

The communication clock is generated, by dividing 1 cycle of psi5_com_clk by 1 to 2^{16}.

1 bit cycle waveform = (value set in PSI5nOPMBITRATE.BITRATECNT) / psi5_com_clk

Baud rate = 1/1 bit cycle waveform

The following shows an example baud rate setting.

Table 27.54 Example Baud Rate Setting

Baud Rate [kbps]	psi5_com_clk		Value Set in PSI5nOPMBITRATE.BITRATECNT	Bit Time [µs]
	Frequency [MHz]	Cycle [ns]		
125	80	12.5	$29F_H$	8.4
189	80	12.5	$1BC_H$	5.57
250	80	12.5	$14F_H$	4.2

Section 28 Peripheral Sensor Interface 5 S (PSI5S)

This section contains a generic description of peripheral sensor interface 5 S (PSI5S).

The first part of this section describes the specific properties of this product, such as the number of channels and register base addresses.

The remainder of this section describes the functions and registers of PSI5S.

28.1 Features of PSI5S for RH850/U2A-EVA

28.1.1 Units and Channels

This product contains a PSI5S with the number of channels shown below.

Table 28.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (j = 0 to 1)	2 (j = 0 to 1)	2 (j = 0 to 1)	2 (j = 0 to 1)	2 (j = 0 to 1)	2 (j = 0 to 1)	2 (j = 0 to 1)	1 (j = 1)	1 (j = 1)	1 (j = 1)
Name	PSI5Sj									

Table 28.2 Index

Index	Description
j	The individual units are identified by the index “j” (j = 0 to 1)
n	The individual channels are identified by the index “n” (n = 0 to 7)
m	The individual packet frames in a channel are identified by the index “m” (m = 0 to 6)
r	The individual bit setting in register description by the index “r” (r = 0 to 31)

28.1.2 Register Base Addresses

The base address <PSI5Sj_base> of PSI5S is shown listed in **Table 28.3**.

Table 28.3 Register Base Addresses <PSI5Sj_base>

Base Address Name	Base Address	Bus Group
<PSI5S0_base>	FFD5 0000_H	P-Bus Group 3
<PSI5S1_base>	FFD5 1000_H	P-Bus Group 3

28.1.3 Clock Supply

The clocks supply to PSI5S are listed in the following table.

Table 28.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
PSI5S	PCLK	CLK_HSB	Bus clock
	psis_clk	CLK_HSB	UART communication clock
	psis_mult_clk	CLK_UHSB	UART communication multiply clock

For details of clock supply, see **Section 13, Clock Controller**.

28.1.4 Interrupt Requests and Error Notifications

The PSI5S interrupt requests are listed in the following table.

Table 28.5 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number
PSI5S0			
INTPSI5S0INTCH0	int_psis_ch0	Communication interrupt for CH0	584
INTPSI5S0INTCH1	int_psis_ch1	Communication interrupt for CH1	585
INTPSI5S0INTCH2	int_psis_ch2	Communication interrupt for CH2	586
INTPSI5S0INTCH3	int_psis_ch3	Communication interrupt for CH3	587
INTPSI5S0INTCH4	int_psis_ch4	Communication interrupt for CH4	588
INTPSI5S0INTCH5	int_psis_ch5	Communication interrupt for CH5	589
INTPSI5S0INTCH6	int_psis_ch6	Communication interrupt for CH6	590
INTPSI5S0INTCH7	int_psis_ch7	Communication interrupt for CH7	591
PSI5S1			
INTPSI5S1INTCH0	int_psis_ch0	Communication interrupt for CH0	592
INTPSI5S1INTCH1	int_psis_ch1	Communication interrupt for CH1	593
INTPSI5S1INTCH2	int_psis_ch2	Communication interrupt for CH2	594
INTPSI5S1INTCH3	int_psis_ch3	Communication interrupt for CH3	595
INTPSI5S1INTCH4	int_psis_ch4	Communication interrupt for CH4	596
INTPSI5S1INTCH5	int_psis_ch5	Communication interrupt for CH5	597
INTPSI5S1INTCH6	int_psis_ch6	Communication interrupt for CH6	598
INTPSI5S1INTCH7	int_psis_ch7	Communication interrupt for CH7	599

This module has no error notifications.

The PSI5S DMA requests are listed in the following table.

Table 28.6 DMA/DTS Requests

Interrupt Symbol Name	Unit Signal	Description	sDMA Trigger Number	DTS Trigger Number
PSI5S0				
INTPSI5S0RXDMACH0	dma_psis_ch0_rx	PSI5S0 DMA request (CH0 RX)	Group1-61	Group1-15
INTPSI5S0RXDMACH1	dma_psis_ch1_rx	PSI5S0 DMA request (CH1 RX)	Group1-62	Group1-16
INTPSI5S0RXDMACH2	dma_psis_ch2_rx	PSI5S0 DMA request (CH2 RX)	Group1-63	Group1-17
INTPSI5S0RXDMACH3	dma_psis_ch3_rx	PSI5S0 DMA request (CH3 RX)	Group1-64	Group1-18
INTPSI5S0RXDMACH4	dma_psis_ch4_rx	PSI5S0 DMA request (CH4 RX)	Group1-65	Group1-19
INTPSI5S0RXDMACH5	dma_psis_ch5_rx	PSI5S0 DMA request (CH5 RX)	Group1-66	Group1-20
INTPSI5S0RXDMACH6	dma_psis_ch6_rx	PSI5S0 DMA request (CH6 RX)	Group1-67	Group1-21
INTPSI5S0RXDMACH7	dma_psis_ch7_rx	PSI5S0 DMA request (CH7 RX) / UART RX	Group1-68	Group1-22
INTPSI5S0TXDMACH1	dma_psis_ch1_tx	PSI5S0 DMA request (CH1 TX)	Group1-69	Group1-23
INTPSI5S0TXDMACH2	dma_psis_ch2_tx	PSI5S0 DMA request (CH2 TX)	Group1-70	Group1-24
INTPSI5S0TXDMACH3	dma_psis_ch3_tx	PSI5S0 DMA request (CH3 TX)	Group1-71	Group1-25
INTPSI5S0TXDMACH4	dma_psis_ch4_tx	PSI5S0 DMA request (CH4 TX)	Group1-72	Group1-26
INTPSI5S0TXDMACH5	dma_psis_ch5_tx	PSI5S0 DMA request (CH5 TX)	Group1-73	Group1-27
INTPSI5S0TXDMACH6	dma_psis_ch6_tx	PSI5S0 DMA request (CH6 TX)	Group1-74	Group1-28
INTPSI5S0TXDMACH7	dma_psis_ch7_tx	PSI5S0 DMA request (CH7 TX) / UART TX	Group1-75	Group1-29
PSI5S1				
INTPSI5S1RXDMACH0	dma_psis_ch0_rx	PSI5S1 DMA request (CH0 RX)	Group1-76	Group1-30
INTPSI5S1RXDMACH1	dma_psis_ch1_rx	PSI5S1 DMA request (CH1 RX)	Group1-77	Group1-31
INTPSI5S1RXDMACH2	dma_psis_ch2_rx	PSI5S1 DMA request (CH2 RX)	Group1-78	Group1-32
INTPSI5S1RXDMACH3	dma_psis_ch3_rx	PSI5S1 DMA request (CH3 RX)	Group1-79	Group1-33
INTPSI5S1RXDMACH4	dma_psis_ch4_rx	PSI5S1 DMA request (CH4 RX)	Group1-80	Group1-34
INTPSI5S1RXDMACH5	dma_psis_ch5_rx	PSI5S1 DMA request (CH5 RX)	Group1-81	Group1-35
INTPSI5S1RXDMACH6	dma_psis_ch6_rx	PSI5S1 DMA request (CH6 RX)	Group1-82	Group1-36
INTPSI5S1RXDMACH7	dma_psis_ch7_rx	PSI5S1 DMA request (CH7 RX) / UART RX	Group1-83	Group1-37
INTPSI5S1TXDMACH1	dma_psis_ch1_tx	PSI5S1 DMA request (CH1 TX)	Group1-84	Group1-38
INTPSI5S1TXDMACH2	dma_psis_ch2_tx	PSI5S1 DMA request (CH2 TX)	Group1-85	Group1-39
INTPSI5S1TXDMACH3	dma_psis_ch3_tx	PSI5S1 DMA request (CH3 TX)	Group1-86	Group1-40
INTPSI5S1TXDMACH4	dma_psis_ch4_tx	PSI5S1 DMA request (CH4 TX)	Group1-87	Group1-41
INTPSI5S1TXDMACH5	dma_psis_ch5_tx	PSI5S1 DMA request (CH5 TX)	Group1-88	Group1-42
INTPSI5S1TXDMACH6	dma_psis_ch6_tx	PSI5S1 DMA request (CH6 TX)	Group1-89	Group1-43
INTPSI5S1TXDMACH7	dma_psis_ch7_tx	PSI5S1 DMA request (CH7 TX) / UART TX	Group1-90	Group1-44

28.1.5 Reset Sources

Table 28.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PSI5S	All registers	√	√	√	√	√	√	—

For details of reset source, see **Section 9, Reset Controller**.

28.1.6 External Input/Output Signals

PSI5S has the input/output pins listed in **Table 28.8**.

Table 28.8 Pin Configuration

Unit Signal Name	Description	Alternative Port Pin Signal
PSI5S0		
psis_rx_data	UART Rx data	PSI5S0RX
psis_tx_data	UART Tx data	PSI5S0TX
psis_tx_sclk	UART clock	PSI5S0CLK
PSI5S1		
psis_rx_data	UART Rx data	PSI5S1RX
psis_tx_data	UART Tx data	PSI5S1TX
psis_tx_sclk	UART clock	PSI5S1CLK

28.2 Outline of Functions

28.2.1 Functional Overview

PSI5S performs UART communication with an external transceiver (PSI5 transceiver).

PSI5S implements a peripheral sensor interface compliant with PSI5 Version 2.2 via the external transceiver.

This document describes the general specifications of PSI5S.

The following table lists the features of PSI5S.

Table 28.9 Features

Item	Specification
UART communication	Sampling clock output*1: 6.67 MHz to 26.67 MHz Baud rate*2: 1.333 Mbps to 5.333 Mbps Frame format (total of 10 or 11 bits) • Start bit: 1 bit • Data: 8 bits • Parity: None, even parity, 0 parity, or odd parity (can be specified separately for reception and transmission) • Stop bit: 1 bit
Sensor-to-ECU communication	• Possible reception of eight channels of frame data • Possible reception of 8-bit to 28-bit payload • Automatic calculation of CRC and parity values from payload • Possible storage of the CRC and parity bits attached to the data in received frames • Monitoring of the number of packets in received frames • Timestamp function for received messages. • Monitoring of the received frames by WDT.
ECU-to-sensor communication	• Automatic calculation of the CRC value to be added to frame data • Output format selectable from frame 1 to frame 4
Other interface	Interrupt output • Eight channels (Ch0 to Ch7) DMA request output • Eight channels (Ch0 to Ch7) for reception and seven channels (Ch1 to Ch7) for transmission

Note 1. For details about the clock specifications, see **Section 28.5.1, Common Setting Procedure**.

Note 2. For details about the baud rate of UART communication, see **Section 28.5.1, Common Setting Procedure**.

28.2.1.1 Terms

Table 28.10 Abbreviations

HW	Hardware
SW	Software
PSI5	Peripheral Sensor Interface 5
UART	Universal Asynchronous Receiver Transmitter
Mbps	Megabits per second
SFR	Special Function Register
ECU	Electronic Control Unit
Ch	Channel
Frm	Frame
Sync	Synchronous
Async	Asynchronous
Rx	Receive
Tx	Transmit
WDT	Watch Dog Timer
MB	MailBox
INT	Interrupt
DDSR	Downstream Data Shift Register
GTM	Generic Timer Module

A “packet” means a UART frame.

A “packet frame” means a group of three to six packets (UART frames) that are received from a transceiver.

A “PSI5 frame” means data that is reconstructed from a packet frame after removing start, stop, and parity bits from the packet frame.

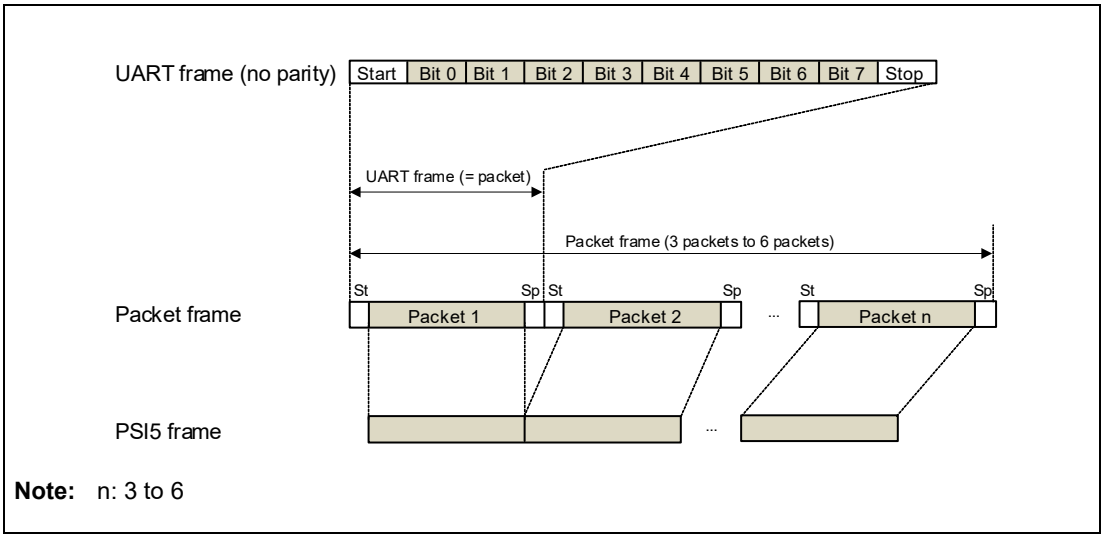


Figure 28.1 Definitions of Packet and Frames (without Parity)

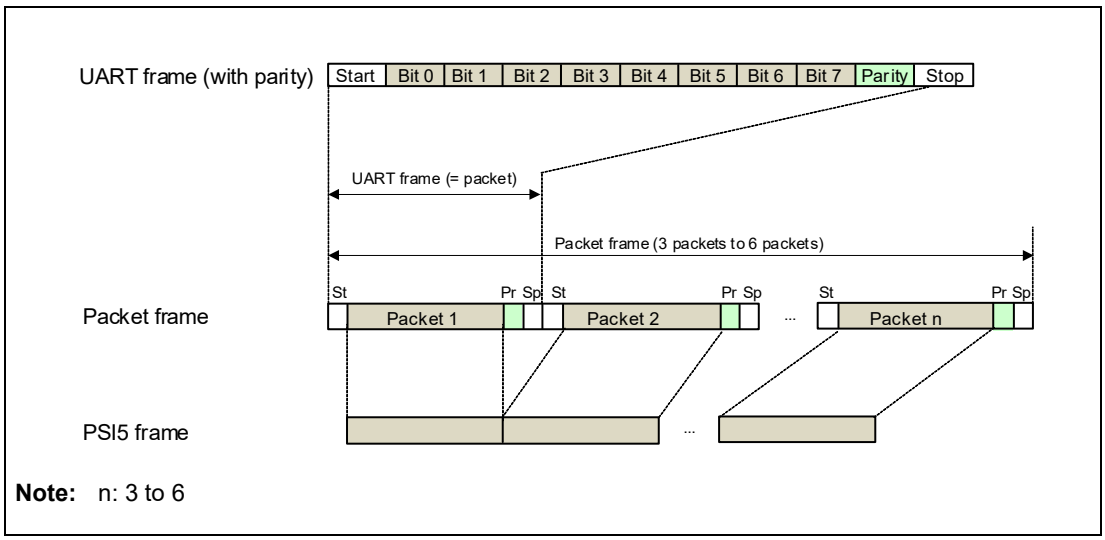


Figure 28.2 Definitions of Packet and Frames (with Parity)

The PSI5 frame format is as follows.

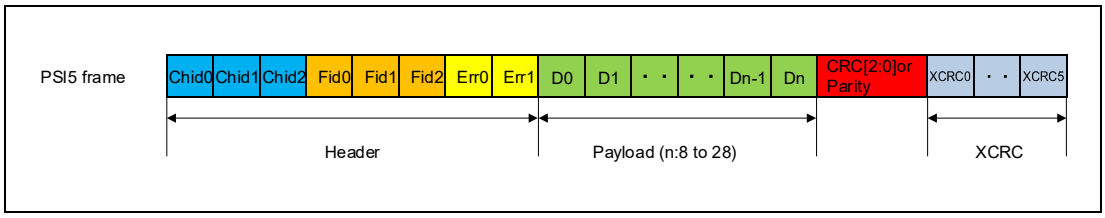


Figure 28.3 PSI5 Frame Format

28.2.2 Block Diagram

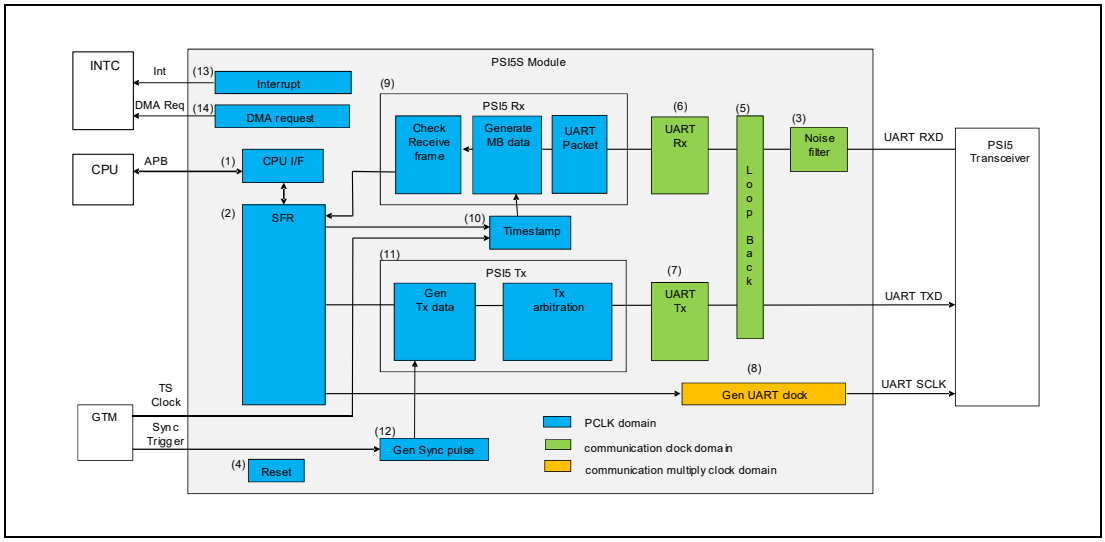


Figure 28.4 PSI5S Block Diagram

The following table lists the functions of PSI5S.

Table 28.11 PSI5S Function List

No	Name	Function
(1)	CPU interface	P-Bus interface (write/read access)
(2)	SFR	Special function registers (detect some kinds of errors)
(3)	Noise filter	Removes signals of less than 37.5 ns.
(4)	Reset	Generates an asynchronous reset via input reset signal or software setting.
(5)	Loopback	UART loopback setting
(6)	UART Rx	UART receiver
(7)	UART Tx	UART transmitter
(8)	Gen UART clock	Generates the UART communication clock.
(9)	PSI5 Rx	Restores PSI5 frame data from UART reception data. Checks for errors in reception data. PSI5 frame data is watched by WDT.
(10)	Timestamp	Generates a timestamp via the GTM or according to internal timing. When a header or synchronization pulse is received, the timestamp value is captured.
(11)	PSI5 Tx	Generates ECU-to-sensor data. When a sync pulse is input to the transceiver as command data, UART transmission is performed.
(12)	Gen Sync pulse	Generates a sync pulse when the input sync trigger is detected by the GTM.
(13)	Interrupt	Generates an interrupt signal when an interrupt factor occurs.
(14)	DMA request	Generates a DMA request signal when a sDMAC/DTS request factor occurs.

28.3 Registers

28.3.1 List of Registers

The PSI5S registers are listed in the following table.

For details on <PSI5Sj_base >, see **Section 28.1.2, Register Base Addresses**.

Table 28.12 List of Registers (1/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S/UART Operation Enable register	PSI5SjPUOEB	<PSI5Sj_base> + 0000_H	*1	—
	PSI5S/UART Operation Mode register	PSI5SjPUOMD	<PSI5Sj_base> + 0004_H	*1	—
	PSI5S/UART Operation Status register	PSI5SjPUOS	<PSI5Sj_base> + 0008_H	*1	—
	PSI5S/UART Noise Filter Set register	PSI5SjPUNFST	<PSI5Sj_base> + $000C_H$	*1	—
	PSI5S/UART Software Reset register	PSI5SjPUSWR	<PSI5Sj_base> + 0010_H	*1	—
	PSI5S Receive Mailbox Data Clear register	PSI5SjPRMBC	<PSI5Sj_base> + 0014_H	*1	—
	PSI5S/UART Communication Loop Back register	PSI5SjPUCLB	<PSI5Sj_base> + 0020_H	*1	—
	PSI5S/UART Rx/Tx Parity Set register	PSI5SjPUPTS	<PSI5Sj_base> + 0024_H	*1	—
	PSI5S/UART Baud Rate Clock Enable register	PSI5SjPUBCE	<PSI5Sj_base> + 0028_H	*1	—
	PSI5S/UART Baud Rate Parameter register	PSI5SjPUBPR	<PSI5Sj_base> + $002C_H$	*1	—
	PSI5S Timestamp Prescaler register	PSI5SjPTPS	<PSI5Sj_base> + 0030_H	*1	—
	PSI5S Timestamp Counter A Select register	PSI5SjPTCAS	<PSI5Sj_base> + 0034_H	*1	—
	PSI5S Timestamp Counter B Select register	PSI5SjPTCBS	<PSI5Sj_base> + 0038_H	*1	—
	PSI5S Timestamp Counter A Enable register	PSI5SjPTCAE	<PSI5Sj_base> + 0040_H	*1	—
	PSI5S Timestamp Counter A Clear register	PSI5SjPTCAC	<PSI5Sj_base> + 0044_H	*1	—
	PSI5S Timestamp Counter B Enable register	PSI5SjPTCBE	<PSI5Sj_base> + 0048_H	*1	—
	PSI5S Timestamp Counter B Clear register	PSI5SjPTCBC	<PSI5Sj_base> + $004C_H$	*1	—
	PSI5S All Timestamp Counter Enable register	PSI5SjPATCE	<PSI5Sj_base> + 0050_H	*1	—
	PSI5S All Timestamp Counter Clear register	PSI5SjPATCC	<PSI5Sj_base> + 0054_H	*1	—
	UART Communication Rx Interrupt Enable	PSI5SjUCRIE	<PSI5Sj_base> + 0058_H	*1	—
	UART Communication Tx Interrupt Enable	PSI5SjUCTIE	<PSI5Sj_base> + $005C_H$	*1	—
	UART Communication DMA Request Enable register	PSI5SjUCDRE	<PSI5Sj_base> + 0060_H	*1	—
	UART Communication Rx Data register	PSI5SjUCRD	<PSI5Sj_base> + 0070_H	*1	—
	UART Communication Rx Status register	PSI5SjUCRS	<PSI5Sj_base> + 0074_H	*1	—
	UART Communication Rx Status Clear register	PSI5SjUCRSC	<PSI5Sj_base> + 0078_H	*1	—
	PSI5S Tx Frame Start register	PSI5SjPTFST	<PSI5Sj_base> + 0080_H	*1	—
	PSI5S Tx Frame Number register	PSI5SjPTFNM	<PSI5Sj_base> + 0084_H	*1	—
	PSI5S Tx Frame Data1 register	PSI5SjPTFD1	<PSI5Sj_base> + 0088_H	*1	—
	PSI5S Tx Frame Data2 register	PSI5SjPTFD2	<PSI5Sj_base> + $008C_H$	*1	—
	PSI5S Tx Frame Status register	PSI5SjPTFS	<PSI5Sj_base> + 0090_H	*1	—
	PSI5S Tx FIFO Status register	PSI5SjPTFIS	<PSI5Sj_base> + 0094_H	*1	—
	UART Communication Tx Data register	PSI5SjUCTD	<PSI5Sj_base> + $00A0_H$	*1	—
	UART Communication Tx Monitoring register	PSI5SjUCTM	<PSI5Sj_base> + $00A4_H$	*1	—
	UART Communication Tx Status register	PSI5SjUCTS	<PSI5Sj_base> + $00A8_H$	*1	—
	UART Communication Tx Status Clear register	PSI5SjUCTSC	<PSI5Sj_base> + $00AC_H$	*1	—

Table 28.12 List of Registers (2/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Receive Config1 ch0 register	PSI5SjPRCF10	<PSI5Sj_base> + 0100_H	*1	—
	PSI5S Receive Config2 ch0 register	PSI5SjPRCF20	<PSI5Sj_base> + 0104_H	*1	—
	PSI5S WDT Enable ch0 register	PSI5SjPWDE0	<PSI5Sj_base> + 0108_H	*1	—
	PSI5S WDT Prescaler ch0 register	PSI5SjPWDP0	<PSI5Sj_base> + $010C_H$	*1	—
	PSI5S WDT Expiration Value ch0 register	PSI5SjPWDEV0	<PSI5Sj_base> + 0110_H	*1	—
	PSI5S CPU Interrupt Enable ch0 register	PSI5SjPCIE0	<PSI5Sj_base> + 0118_H	*1	—
	PSI5S DMA Transfer Request Enable ch0 register	PSI5SjPDRE0	<PSI5Sj_base> + $011C_H$	*1	—
	PSI5S Receive Error Status ch0 register	PSI5SjPRES0	<PSI5Sj_base> + 0130_H	*1	—
	PSI5S Receive Error Status Clear ch0 register	PSI5SjPRESC0	<PSI5Sj_base> + 0134_H	*1	—
	PSI5S Timestamp Capture Data ch0 register	PSI5SjPTCDT0	<PSI5Sj_base> + 0138_H	*1	—
	PSI5S Timestamp Capture Data Clear ch0 register	PSI5SjPTCDC0	<PSI5Sj_base> + $013C_H$	*1	—
	PSI5S CPU Interrupt Status ch0 register	PSI5SjPCIS0	<PSI5Sj_base> + 0150_H	*1	—
	PSI5S CPU Interrupt Status Clear ch0 register	PSI5SjPCISC0	<PSI5Sj_base> + 0154_H	*1	—
	PSI5S Receive Config1 ch1 register	PSI5SjPRCF11	<PSI5Sj_base> + 0180_H	*1	—
	PSI5S Receive Config2 ch1 register	PSI5SjPRCF21	<PSI5Sj_base> + 0184_H	*1	—
	PSI5S WDT Enable ch1 register	PSI5SjPWDE1	<PSI5Sj_base> + 0188_H	*1	—
	PSI5S WDT Prescaler ch1 register	PSI5SjPWDP1	<PSI5Sj_base> + $018C_H$	*1	—
	PSI5S WDT Expiration Value ch1 register	PSI5SjPWDEV1	<PSI5Sj_base> + 0190_H	*1	—
	PSI5S Tx Command Data ch1 register	PSI5SjPTCD1	<PSI5Sj_base> + 0194_H	*1	—
	PSI5S CPU Interrupt Enable ch1 register	PSI5SjPCIE1	<PSI5Sj_base> + 0198_H	*1	—
	PSI5S DMA Transfer Request Enable ch1 register	PSI5SjPDRE1	<PSI5Sj_base> + $019C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch1 register	PSI5SjPSTP1	<PSI5Sj_base> + $01A4_H$	*1	—
	PSI5S Sync Trigger Expiration Value ch1 register	PSI5SjPSTEV1	<PSI5Sj_base> + $01A8_H$	*1	—
	PSI5S Sync Trigger Select ch1 register	PSI5SjPSTS1	<PSI5Sj_base> + $01AC_H$	*1	—
	PSI5S Receive Error Status ch1 register	PSI5SjPRES1	<PSI5Sj_base> + $01B0_H$	*1	—
	PSI5S Receive Error Status Clear ch1 register	PSI5SjPRESC1	<PSI5Sj_base> + $01B4_H$	*1	—
	PSI5S Timestamp Capture Data ch1 register	PSI5SjPTCDT1	<PSI5Sj_base> + $01B8_H$	*1	—
	PSI5S Timestamp Capture Data Clear ch1 register	PSI5SjPTCDC1	<PSI5Sj_base> + $01BC_H$	*1	—
	PSI5S DDSR Type ch1 register	PSI5SjPDDTP1	<PSI5Sj_base> + $01C0_H$	*1	—
	PSI5S DDSR Data ch1 register	PSI5SjPDDD1	<PSI5Sj_base> + $01C4_H$	*1	—
	PSI5S DDSR Status ch1 register	PSI5SjPDDS1	<PSI5Sj_base> + $01C8_H$	*1	—
	PSI5S DDSR Stop ch1 register	PSI5SjPDDSP1	<PSI5Sj_base> + $01CC_H$	*1	—
	PSI5S CPU Interrupt Status ch1 register	PSI5SjPCIS1	<PSI5Sj_base> + $01D0_H$	*1	—
	PSI5S CPU Interrupt Status Clear ch1 register	PSI5SjPCISC1	<PSI5Sj_base> + $01D4_H$	*1	—
	PSI5S Receive Config1 ch2 register	PSI5SjPRCF12	<PSI5Sj_base> + 0200_H	*1	—
	PSI5S Receive Config2 ch2 register	PSI5SjPRCF22	<PSI5Sj_base> + 0204_H	*1	—
	PSI5S WDT Enable ch2 register	PSI5SjPWDE2	<PSI5Sj_base> + 0208_H	*1	—
	PSI5S WDT Prescaler ch2 register	PSI5SjPWDP2	<PSI5Sj_base> + $020C_H$	*1	—
	PSI5S WDT Expiration Value ch2 register	PSI5SjPWDEV2	<PSI5Sj_base> + 0210_H	*1	—
	PSI5S Tx Command Data ch2 register	PSI5SjPTCD2	<PSI5Sj_base> + 0214_H	*1	—
	PSI5S CPU Interrupt Enable ch2 register	PSI5SjPCIE2	<PSI5Sj_base> + 0218_H	*1	—
	PSI5S DMA Transfer Request Enable ch2 register	PSI5SjPDRE2	<PSI5Sj_base> + $021C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch2 register	PSI5SjPSTP2	<PSI5Sj_base> + 0224_H	*1	—

Table 28.12 List of Registers (3/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Sync Trigger Expiration Value ch2 register	PSI5SjPSTEV2	<PSI5Sj_base> + 0228_H	*1	—
	PSI5S Sync Trigger Select ch2 register	PSI5SjPSTS2	<PSI5Sj_base> + $022C_H$	*1	—
	PSI5S Receive Error Status ch2 register	PSI5SjPRES2	<PSI5Sj_base> + 0230_H	*1	—
	PSI5S Receive Error Status Clear ch2 register	PSI5SjPRESC2	<PSI5Sj_base> + 0234_H	*1	—
	PSI5S Timestamp Capture Data ch2 register	PSI5SjPTCDT2	<PSI5Sj_base> + 0238_H	*1	—
	PSI5S Timestamp Capture Data Clear ch2 register	PSI5SjPTCDC2	<PSI5Sj_base> + $023C_H$	*1	—
	PSI5S DDSR Type ch2 register	PSI5SjPDDTP2	<PSI5Sj_base> + 0240_H	*1	—
	PSI5S DDSR Data ch2 register	PSI5SjPDDD2	<PSI5Sj_base> + 0244_H	*1	—
	PSI5S DDSR Status ch2 register	PSI5SjPDDS2	<PSI5Sj_base> + 0248_H	*1	—
	PSI5S DDSR Stop ch2 register	PSI5SjPDDSP2	<PSI5Sj_base> + $024C_H$	*1	—
	PSI5S CPU Interrupt Status ch2 register	PSI5SjPCIS2	<PSI5Sj_base> + 0250_H	*1	—
	PSI5S CPU Interrupt Status Clear ch2 register	PSI5SjPCISC2	<PSI5Sj_base> + 0254_H	*1	—
	PSI5S Receive Config1 ch3 register	PSI5SjPRCF13	<PSI5Sj_base> + 0280_H	*1	—
	PSI5S Receive Config2 ch3 register	PSI5SjPRCF23	<PSI5Sj_base> + 0284_H	*1	—
	PSI5S WDT Enable ch3 register	PSI5SjPWDE3	<PSI5Sj_base> + 0288_H	*1	—
	PSI5S WDT Prescaler ch3 register	PSI5SjPWDP3	<PSI5Sj_base> + $028C_H$	*1	—
	PSI5S WDT Expiration Value ch3 register	PSI5SjPWDEV3	<PSI5Sj_base> + 0290_H	*1	—
	PSI5S Tx Command Data ch3 register	PSI5SjPTCD3	<PSI5Sj_base> + 0294_H	*1	—
	PSI5S CPU Interrupt Enable ch3 register	PSI5SjPCIE3	<PSI5Sj_base> + 0298_H	*1	—
	PSI5S DMA Transfer Request Enable ch3 register	PSI5SjPDRE3	<PSI5Sj_base> + $029C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch3 register	PSI5SjPSTP3	<PSI5Sj_base> + $02A4_H$	*1	—
	PSI5S Sync Trigger Expiration Value ch3 register	PSI5SjPSTEV3	<PSI5Sj_base> + $02A8_H$	*1	—
	PSI5S Sync Trigger Select ch3 register	PSI5SjPSTS3	<PSI5Sj_base> + $02AC_H$	*1	—
	PSI5S Receive Error Status ch3 register	PSI5SjPRES3	<PSI5Sj_base> + $02B0_H$	*1	—
	PSI5S Receive Error Status Clear ch3 register	PSI5SjPRESC3	<PSI5Sj_base> + $02B4_H$	*1	—
	PSI5S Timestamp Capture Data ch3 register	PSI5SjPTCDT3	<PSI5Sj_base> + $02B8_H$	*1	—
	PSI5S Timestamp Capture Data Clear ch3 register	PSI5SjPTCDC3	<PSI5Sj_base> + $02BC_H$	*1	—
	PSI5S DDSR Type ch3 register	PSI5SjPDDTP3	<PSI5Sj_base> + $02C0_H$	*1	—
	PSI5S DDSR Data ch3 register	PSI5SjPDDD3	<PSI5Sj_base> + $02C4_H$	*1	—
	PSI5S DDSR Status ch3 register	PSI5SjPDDS3	<PSI5Sj_base> + $02C8_H$	*1	—
	PSI5S DDSR Stop ch3 register	PSI5SjPDDSP3	<PSI5Sj_base> + $02CC_H$	*1	—
	PSI5S CPU Interrupt Status ch3 register	PSI5SjPCIS3	<PSI5Sj_base> + $02D0_H$	*1	—
	PSI5S CPU Interrupt Status Clear ch3 register	PSI5SjPCISC3	<PSI5Sj_base> + $02D4_H$	*1	—
	PSI5S Receive Config1 ch4 register	PSI5SjPRCF14	<PSI5Sj_base> + 0300_H	*1	—
	PSI5S Receive Config2 ch4 register	PSI5SjPRCF24	<PSI5Sj_base> + 0304_H	*1	—
	PSI5S WDT Enable ch4 register	PSI5SjPWDE4	<PSI5Sj_base> + 0308_H	*1	—
	PSI5S WDT Prescaler ch4 register	PSI5SjPWDP4	<PSI5Sj_base> + $030C_H$	*1	—
	PSI5S WDT Expiration Value ch4 register	PSI5SjPWDEV4	<PSI5Sj_base> + 0310_H	*1	—
	PSI5S Tx Command Data ch4 register	PSI5SjPTCD4	<PSI5Sj_base> + 0314_H	*1	—
	PSI5S CPU Interrupt Enable ch4 register	PSI5SjPCIE4	<PSI5Sj_base> + 0318_H	*1	—
	PSI5S DMA Transfer Request Enable ch4 register	PSI5SjPDRE4	<PSI5Sj_base> + $031C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch4 register	PSI5SjPSTP4	<PSI5Sj_base> + 0324_H	*1	—
	PSI5S Sync Trigger Expiration Value ch4 register	PSI5SjPSTEV4	<PSI5Sj_base> + 0328_H	*1	—

Table 28.12 List of Registers (4/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Sync Trigger Select ch4 register	PSI5SjPSTS4	<PSI5Sj_base> + $032C_H$	*1	—
	PSI5S Receive Error Status ch4 register	PSI5SjPRES4	<PSI5Sj_base> + 0330_H	*1	—
	PSI5S Receive Error Status Clear ch4 register	PSI5SjPRESC4	<PSI5Sj_base> + 0334_H	*1	—
	PSI5S Timestamp Capture Data ch4 register	PSI5SjPTCDT4	<PSI5Sj_base> + 0338_H	*1	—
	PSI5S Timestamp Capture Data Clear ch4 register	PSI5SjPTCDC4	<PSI5Sj_base> + $033C_H$	*1	—
	PSI5S DDSR Type ch4 register	PSI5SjPDDTP4	<PSI5Sj_base> + 0340_H	*1	—
	PSI5S DDSR Data ch4 register	PSI5SjPDDD4	<PSI5Sj_base> + 0344_H	*1	—
	PSI5S DDSR Status ch4 register	PSI5SjPDDS4	<PSI5Sj_base> + 0348_H	*1	—
	PSI5S DDSR Stop ch4 register	PSI5SjPDDSP4	<PSI5Sj_base> + $034C_H$	*1	—
	PSI5S CPU Interrupt Status ch4 register	PSI5SjPCIS4	<PSI5Sj_base> + 0350_H	*1	—
	PSI5S CPU Interrupt Status Clear ch4 register	PSI5SjPCISC4	<PSI5Sj_base> + 0354_H	*1	—
	PSI5S Receive Config1 ch5 register	PSI5SjPRCF15	<PSI5Sj_base> + 0380_H	*1	—
	PSI5S Receive Config2 ch5 register	PSI5SjPRCF25	<PSI5Sj_base> + 0384_H	*1	—
	PSI5S WDT Enable ch5 register	PSI5SjPWDE5	<PSI5Sj_base> + 0388_H	*1	—
	PSI5S WDT Prescaler ch5 register	PSI5SjPWDP5	<PSI5Sj_base> + $038C_H$	*1	—
	PSI5S WDT Expiration Value ch5 register	PSI5SjPWDEV5	<PSI5Sj_base> + 0390_H	*1	—
	PSI5S Tx Command Data ch5 register	PSI5SjPTCD5	<PSI5Sj_base> + 0394_H	*1	—
	PSI5S CPU Interrupt Enable ch5 register	PSI5SjPCIE5	<PSI5Sj_base> + 0398_H	*1	—
	PSI5S DMA Transfer Request Enable ch5 register	PSI5SjPDRE5	<PSI5Sj_base> + $039C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch5 register	PSI5SjPSTP5	<PSI5Sj_base> + $03A4_H$	*1	—
	PSI5S Sync Trigger Expiration Value ch5 register	PSI5SjPSTEV5	<PSI5Sj_base> + $03A8_H$	*1	—
	PSI5S Sync Trigger Select ch5 register	PSI5SjPSTS5	<PSI5Sj_base> + $03AC_H$	*1	—
	PSI5S Receive Error Status ch5 register	PSI5SjPRES5	<PSI5Sj_base> + $03B0_H$	*1	—
	PSI5S Receive Error Status Clear ch5 register	PSI5SjPRESC5	<PSI5Sj_base> + $03B4_H$	*1	—
	PSI5S Timestamp Capture Data ch5 register	PSI5SjPTCDT5	<PSI5Sj_base> + $03B8_H$	*1	—
	PSI5S Timestamp Capture Data Clear ch5 register	PSI5SjPTCDC5	<PSI5Sj_base> + $03BC_H$	*1	—
	PSI5S DDSR Type ch5 register	PSI5SjPDDTP5	<PSI5Sj_base> + $03C0_H$	*1	—
	PSI5S DDSR Data ch5 register	PSI5SjPDDD5	<PSI5Sj_base> + $03C4_H$	*1	—
	PSI5S DDSR Status ch5 register	PSI5SjPDDS5	<PSI5Sj_base> + $03C8_H$	*1	—
	PSI5S DDSR Stop ch5 register	PSI5SjPDDSP5	<PSI5Sj_base> + $03CC_H$	*1	—
	PSI5S CPU Interrupt Status ch5 register	PSI5SjPCIS5	<PSI5Sj_base> + $03D0_H$	*1	—
	PSI5S CPU Interrupt Status Clear ch5 register	PSI5SjPCISC5	<PSI5Sj_base> + $03D4_H$	*1	—
	PSI5S Receive Config1 ch6 register	PSI5SjPRCF16	<PSI5Sj_base> + 0400_H	*1	—
	PSI5S Receive Config2 ch6 register	PSI5SjPRCF26	<PSI5Sj_base> + 0404_H	*1	—
	PSI5S WDT Enable ch6 register	PSI5SjPWDE6	<PSI5Sj_base> + 0408_H	*1	—
	PSI5S WDT Prescaler ch6 register	PSI5SjPWDP6	<PSI5Sj_base> + $040C_H$	*1	—
	PSI5S WDT Expiration Value ch6 register	PSI5SjPWDEV6	<PSI5Sj_base> + 0410_H	*1	—
	PSI5S Tx Command Data ch6 register	PSI5SjPTCD6	<PSI5Sj_base> + 0414_H	*1	—
	PSI5S CPU Interrupt Enable ch6 register	PSI5SjPCIE6	<PSI5Sj_base> + 0418_H	*1	—
	PSI5S DMA Transfer Request Enable ch6 register	PSI5SjPDRE6	<PSI5Sj_base> + $041C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch6 register	PSI5SjPSTP6	<PSI5Sj_base> + 0424_H	*1	—
	PSI5S Sync Trigger Expiration Value ch6 register	PSI5SjPSTEV6	<PSI5Sj_base> + 0428_H	*1	—
	PSI5S Sync Trigger Select ch6 register	PSI5SjPSTS6	<PSI5Sj_base> + $042C_H$	*1	—

Table 28.12 List of Registers (5/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Receive Error Status ch6 register	PSI5SjPRES6	<PSI5Sj_base> + 0430_H	*1	—
	PSI5S Receive Error Status Clear ch6 register	PSI5SjPRESC6	<PSI5Sj_base> + 0434_H	*1	—
	PSI5S Timestamp Capture Data ch6 register	PSI5SjPTCDT6	<PSI5Sj_base> + 0438_H	*1	—
	PSI5S Timestamp Capture Data Clear ch6 register	PSI5SjPTCDC6	<PSI5Sj_base> + $043C_H$	*1	—
	PSI5S DDSR Type ch6 register	PSI5SjPDDTP6	<PSI5Sj_base> + 0440_H	*1	—
	PSI5S DDSR Data ch6 register	PSI5SjPDDD6	<PSI5Sj_base> + 0444_H	*1	—
	PSI5S DDSR Status ch6 register	PSI5SjPDDS6	<PSI5Sj_base> + 0448_H	*1	—
	PSI5S DDSR Stop ch6 register	PSI5SjPDDSP6	<PSI5Sj_base> + $044C_H$	*1	—
	PSI5S CPU Interrupt Status ch6 register	PSI5SjPCIS6	<PSI5Sj_base> + 0450_H	*1	—
	PSI5S CPU Interrupt Status Clear ch6 register	PSI5SjPCISC6	<PSI5Sj_base> + 0454_H	*1	—
	PSI5S Receive Config1 ch7 register	PSI5SjPRCF17	<PSI5Sj_base> + 0480_H	*1	—
	PSI5S Receive Config2 ch7 register	PSI5SjPRCF27	<PSI5Sj_base> + 0484_H	*1	—
	PSI5S WDT Enable ch7 register	PSI5SjPWDE7	<PSI5Sj_base> + 0488_H	*1	—
	PSI5S WDT Prescaler ch7 register	PSI5SjPWDP7	<PSI5Sj_base> + $048C_H$	*1	—
	PSI5S WDT Expiration Value ch7 register	PSI5SjPWDEV7	<PSI5Sj_base> + 0490_H	*1	—
	PSI5S Tx Command Data ch7 register	PSI5SjPTCD7	<PSI5Sj_base> + 0494_H	*1	—
	PSI5S CPU Interrupt Enable ch7 register	PSI5SjPCIE7	<PSI5Sj_base> + 0498_H	*1	—
	PSI5S DMA Transfer Request Enable ch7 register	PSI5SjPDRE7	<PSI5Sj_base> + $049C_H$	*1	—
	PSI5S Sync Trigger Prescaler ch7 register	PSI5SjPSTP7	<PSI5Sj_base> + $04A4_H$	*1	—
	PSI5S Sync Trigger Expiration Value ch7 register	PSI5SjPSTEV7	<PSI5Sj_base> + $04A8_H$	*1	—
	PSI5S Sync Trigger Select ch7 register	PSI5SjPSTS7	<PSI5Sj_base> + $04AC_H$	*1	—
	PSI5S Receive Error Status ch7 register	PSI5SjPRES7	<PSI5Sj_base> + $04B0_H$	*1	—
	PSI5S Receive Error Status Clear ch7 register	PSI5SjPRESC7	<PSI5Sj_base> + $04B4_H$	*1	—
	PSI5S Timestamp Capture Data ch7 register	PSI5SjPTCDT7	<PSI5Sj_base> + $04B8_H$	*1	—
	PSI5S Timestamp Capture Data Clear ch7 register	PSI5SjPTCDC7	<PSI5Sj_base> + $04BC_H$	*1	—
	PSI5S DDSR Type ch7 register	PSI5SjPDDTP7	<PSI5Sj_base> + $04C0_H$	*1	—
	PSI5S DDSR Data ch7 register	PSI5SjPDDD7	<PSI5Sj_base> + $04C4_H$	*1	—
	PSI5S DDSR Status ch7 register	PSI5SjPDDS7	<PSI5Sj_base> + $04C8_H$	*1	—
	PSI5S DDSR Stop ch7 register	PSI5SjPDDSP7	<PSI5Sj_base> + $04CC_H$	*1	—
	PSI5S CPU Interrupt Status ch7 register	PSI5SjPCIS7	<PSI5Sj_base> + $04D0_H$	*1	—
	PSI5S CPU Interrupt Status Clear ch7 register	PSI5SjPCISC7	<PSI5Sj_base> + $04D4_H$	*1	—
	PSI5S Receive MailBox ch0 Frm1 Status register	PSI5SjPMB01S	<PSI5Sj_base> + 0500_H	*1	—
	PSI5S Receive MailBox ch0 Frm1 Data register	PSI5SjPMB01D	<PSI5Sj_base> + 0504_H	*1	—
	PSI5S Receive MailBox ch0 Frm1 Timestamp register	PSI5SjPMB01T	<PSI5Sj_base> + 0508_H	*1	—
	PSI5S Receive MailBox ch0 Frm2 Status register	PSI5SjPMB02S	<PSI5Sj_base> + $050C_H$	*1	—
	PSI5S Receive MailBox ch0 Frm2 Data register	PSI5SjPMB02D	<PSI5Sj_base> + 0510_H	*1	—
	PSI5S Receive MailBox ch0 Frm2 Timestamp register	PSI5SjPMB02T	<PSI5Sj_base> + 0514_H	*1	—
	PSI5S Receive MailBox ch1 frm1 Status register	PSI5SjPMB11S	<PSI5Sj_base> + 0548_H	*1	—
	PSI5S Receive MailBox ch1 frm1 Data register	PSI5SjPMB11D	<PSI5Sj_base> + $054C_H$	*1	—
	PSI5S Receive MailBox ch1 frm1 Timestamp register	PSI5SjPMB11T	<PSI5Sj_base> + 0550_H	*1	—
	PSI5S Receive MailBox ch1 frm2 Status register	PSI5SjPMB12S	<PSI5Sj_base> + 0554_H	*1	—
	PSI5S receive MailBox ch1 frm2 Data register	PSI5SjPMB12D	<PSI5Sj_base> + 0558_H	*1	—

Table 28.12 List of Registers (6/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Receive MailBox ch1 frm2 Timestamp register	PSI5SjPMB12T	<PSI5Sj_base> + $055C_H$	*1	—
	PSI5S Receive MailBox ch1 frm3 Status register	PSI5SjPMB13S	<PSI5Sj_base> + 0560_H	*1	—
	PSI5S Receive MailBox ch1 frm3 Data register	PSI5SjPMB13D	<PSI5Sj_base> + 0564_H	*1	—
	PSI5S Receive MailBox ch1 frm3 Timestamp register	PSI5SjPMB13T	<PSI5Sj_base> + 0568_H	*1	—
	PSI5S Receive MailBox ch1 frm4 Status register	PSI5SjPMB14S	<PSI5Sj_base> + $056C_H$	*1	—
	PSI5S Receive MailBox ch1 frm4 Data register	PSI5SjPMB14D	<PSI5Sj_base> + 0570_H	*1	—
	PSI5S Receive MailBox ch1 frm4 Timestamp register	PSI5SjPMB14T	<PSI5Sj_base> + 0574_H	*1	—
	PSI5S Receive MailBox ch1 frm5 Status register	PSI5SjPMB15S	<PSI5Sj_base> + 0578_H	*1	—
	PSI5S Receive MailBox ch1 frm5 Data register	PSI5SjPMB15D	<PSI5Sj_base> + $057C_H$	*1	—
	PSI5S Receive MailBox ch1 frm5 Timestamp register	PSI5SjPMB15T	<PSI5Sj_base> + 0580_H	*1	—
	PSI5S Receive MailBox ch1 frm6 Status register	PSI5SjPMB16S	<PSI5Sj_base> + 0584_H	*1	—
	PSI5S Receive MailBox ch1 frm6 Data register	PSI5SjPMB16D	<PSI5Sj_base> + 0588_H	*1	—
	PSI5S Receive MailBox ch1 frm6 Timestamp register	PSI5SjPMB16T	<PSI5Sj_base> + $058C_H$	*1	—
	PSI5S Receive MailBox ch2 frm1 Status register	PSI5SjPMB21S	<PSI5Sj_base> + 0590_H	*1	—
	PSI5S Receive MailBox ch2 frm1 Data register	PSI5SjPMB21D	<PSI5Sj_base> + 0594_H	*1	—
	PSI5S Receive MailBox ch2 frm1 Timestamp register	PSI5SjPMB21T	<PSI5Sj_base> + 0598_H	*1	—
	PSI5S Receive MailBox ch2 frm2 Status register	PSI5SjPMB22S	<PSI5Sj_base> + $059C_H$	*1	—
	PSI5S Receive MailBox ch2 frm2 Data register	PSI5SjPMB22D	<PSI5Sj_base> + $05A0_H$	*1	—
	PSI5S Receive MailBox ch2 frm2 Timestamp register	PSI5SjPMB22T	<PSI5Sj_base> + $05A4_H$	*1	—
	PSI5S Receive MailBox ch2 frm3 Status register	PSI5SjPMB23S	<PSI5Sj_base> + $05A8_H$	*1	—
	PSI5S Receive MailBox ch2 frm3 Data register	PSI5SjPMB23D	<PSI5Sj_base> + $05AC_H$	*1	—
	PSI5S receive MailBox ch2 frm3 Timestamp register	PSI5SjPMB23T	<PSI5Sj_base> + $05B0_H$	*1	—
	PSI5S receive MailBox ch2 frm4 Status register	PSI5SjPMB24S	<PSI5Sj_base> + $05B4_H$	*1	—
	PSI5S Receive MailBox ch2 frm4 Data register	PSI5SjPMB24D	<PSI5Sj_base> + $05B8_H$	*1	—
	PSI5S Receive MailBox ch2 frm4 Timestamp register	PSI5SjPMB24T	<PSI5Sj_base> + $05BC_H$	*1	—
	PSI5S Receive MailBox ch2 frm5 Status register	PSI5SjPMB25S	<PSI5Sj_base> + $05C0_H$	*1	—
	PSI5S Receive MailBox ch2 frm5 Data register	PSI5SjPMB25D	<PSI5Sj_base> + $05C4_H$	*1	—
	PSI5S Receive MailBox ch2 frm5 Timestamp register	PSI5SjPMB25T	<PSI5Sj_base> + $05C8_H$	*1	—
	PSI5S Receive MailBox ch2 frm6 Status register	PSI5SjPMB26S	<PSI5Sj_base> + $05CC_H$	*1	—
	PSI5S Receive MailBox ch2 frm6 Data register	PSI5SjPMB26D	<PSI5Sj_base> + $05D0_H$	*1	—
	PSI5S Receive MailBox ch2 frm6 Timestamp register	PSI5SjPMB26T	<PSI5Sj_base> + $05D4_H$	*1	—
	PSI5S Receive MailBox ch3 frm1 Status register	PSI5SjPMB31S	<PSI5Sj_base> + $05D8_H$	*1	—
	PSI5S Receive MailBox ch3 frm1 Data register	PSI5SjPMB31D	<PSI5Sj_base> + $05DC_H$	*1	—
	PSI5S Receive MailBox ch3 frm1 Timestamp register	PSI5SjPMB31T	<PSI5Sj_base> + $05E0_H$	*1	—
	PSI5S Receive MailBox ch3 frm2 Status register	PSI5SjPMB32S	<PSI5Sj_base> + $05E4_H$	*1	—
	PSI5S Receive MailBox ch3 frm2 Data register	PSI5SjPMB32D	<PSI5Sj_base> + $05E8_H$	*1	—
	PSI5S Receive MailBox ch3 frm2 Timestamp register	PSI5SjPMB32T	<PSI5Sj_base> + $05EC_H$	*1	—
	PSI5S Receive MailBox ch3 frm3 Status register	PSI5SjPMB33S	<PSI5Sj_base> + $05F0_H$	*1	—
	PSI5S Receive MailBox ch3 frm3 Data register	PSI5SjPMB33D	<PSI5Sj_base> + $05F4_H$	*1	—
	PSI5S Receive MailBox ch3 frm3 Timestamp register	PSI5SjPMB33T	<PSI5Sj_base> + $05F8_H$	*1	—
	PSI5S Receive MailBox ch3 frm4 Status register	PSI5SjPMB34S	<PSI5Sj_base> + $05FC_H$	*1	—
	PSI5S Receive MailBox ch3 frm4 Data register	PSI5SjPMB34D	<PSI5Sj_base> + 0600_H	*1	—
	PSI5S Receive MailBox ch3 frm4 Timestamp register	PSI5SjPMB34T	<PSI5Sj_base> + 0604_H	*1	—

Table 28.12 List of Registers (7/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Receive MailBox ch3 frm5 Status register	PSI5SjPMB35S	<PSI5Sj_base> + 0608_H	*1	—
	PSI5S Receive MailBox ch3 frm5 Data register	PSI5SjPMB35D	<PSI5Sj_base> + $060C_H$	*1	—
	PSI5S Receive MailBox ch3 frm5 Timestamp register	PSI5SjPMB35T	<PSI5Sj_base> + 0610_H	*1	—
	PSI5S Receive MailBox ch3 frm6 Status register	PSI5SjPMB36S	<PSI5Sj_base> + 0614_H	*1	—
	PSI5S Receive MailBox ch3 frm6 Data register	PSI5SjPMB36D	<PSI5Sj_base> + 0618_H	*1	—
	PSI5S Receive MailBox ch3 frm6 Timestamp register	PSI5SjPMB36T	<PSI5Sj_base> + $061C_H$	*1	—
	PSI5S Receive MailBox ch4 frm1 Status register	PSI5SjPMB41S	<PSI5Sj_base> + 0620_H	*1	—
	PSI5S Receive MailBox ch4 frm1 Data register	PSI5SjPMB41D	<PSI5Sj_base> + 0624_H	*1	—
	PSI5S Receive MailBox ch4 frm1 Timestamp register	PSI5SjPMB41T	<PSI5Sj_base> + 0628_H	*1	—
	PSI5S Receive MailBox ch4 frm2 Status register	PSI5SjPMB42S	<PSI5Sj_base> + $062C_H$	*1	—
	PSI5S Receive MailBox ch4 frm2 Data register	PSI5SjPMB42D	<PSI5Sj_base> + 0630_H	*1	—
	PSI5S Receive MailBox ch4 frm2 Timestamp register	PSI5SjPMB42T	<PSI5Sj_base> + 0634_H	*1	—
	PSI5S Receive MailBox ch4 frm3 Status register	PSI5SjPMB43S	<PSI5Sj_base> + 0638_H	*1	—
	PSI5S Receive MailBox ch4 frm3 Data register	PSI5SjPMB43D	<PSI5Sj_base> + $063C_H$	*1	—
	PSI5S Receive MailBox ch4 frm3 Timestamp register	PSI5SjPMB43T	<PSI5Sj_base> + 0640_H	*1	—
	PSI5S Receive MailBox ch4 frm4 Status register	PSI5SjPMB44S	<PSI5Sj_base> + 0644_H	*1	—
	PSI5S Receive MailBox ch4 frm4 Data register	PSI5SjPMB44D	<PSI5Sj_base> + 0648_H	*1	—
	PSI5S Receive MailBox ch4 frm4 Timestamp register	PSI5SjPMB44T	<PSI5Sj_base> + $064C_H$	*1	—
	PSI5S Receive MailBox ch4 frm5 Status register	PSI5SjPMB45S	<PSI5Sj_base> + 0650_H	*1	—
	PSI5S Receive MailBox ch4 frm5 Data register	PSI5SjPMB45D	<PSI5Sj_base> + 0654_H	*1	—
	PSI5S Receive MailBox ch4 frm5 Timestamp register	PSI5SjPMB45T	<PSI5Sj_base> + 0658_H	*1	—
	PSI5S Receive MailBox ch4 frm6 Status register	PSI5SjPMB46S	<PSI5Sj_base> + $065C_H$	*1	—
	PSI5S Receive MailBox ch4 frm6 Data register	PSI5SjPMB46D	<PSI5Sj_base> + 0660_H	*1	—
	PSI5S Receive MailBox ch4 frm6 Timestamp register	PSI5SjPMB46T	<PSI5Sj_base> + 0664_H	*1	—
	PSI5S Receive MailBox ch5 frm1 Status register	PSI5SjPMB51S	<PSI5Sj_base> + 0668_H	*1	—
	PSI5S Receive MailBox ch5 frm1 Data register	PSI5SjPMB51D	<PSI5Sj_base> + $066C_H$	*1	—
	PSI5S Receive MailBox ch5 frm1 Timestamp register	PSI5SjPMB51T	<PSI5Sj_base> + 0670_H	*1	—
	PSI5S Receive MailBox ch5 frm2 Status register	PSI5SjPMB52S	<PSI5Sj_base> + 0674_H	*1	—
	PSI5S Receive MailBox ch5 frm2 Data register	PSI5SjPMB52D	<PSI5Sj_base> + 0678_H	*1	—
	PSI5S Receive MailBox ch5 frm2 Timestamp register	PSI5SjPMB52T	<PSI5Sj_base> + $067C_H$	*1	—
	PSI5S Receive MailBox ch5 frm3 Status register	PSI5SjPMB53S	<PSI5Sj_base> + 0680_H	*1	—
	PSI5S Receive MailBox ch5 frm3 Data register	PSI5SjPMB53D	<PSI5Sj_base> + 0684_H	*1	—
	PSI5S Receive MailBox ch5 frm3 Timestamp register	PSI5SjPMB53T	<PSI5Sj_base> + 0688_H	*1	—
	PSI5S Receive MailBox ch5 frm4 Status register	PSI5SjPMB54S	<PSI5Sj_base> + $068C_H$	*1	—
	PSI5S Receive MailBox ch5 frm4 Data register	PSI5SjPMB54D	<PSI5Sj_base> + 0690_H	*1	—
	PSI5S Receive MailBox ch5 frm4 Timestamp register	PSI5SjPMB54T	<PSI5Sj_base> + 0694_H	*1	—
	PSI5S Receive MailBox ch5 frm5 Status register	PSI5SjPMB55S	<PSI5Sj_base> + 0698_H	*1	—
	PSI5S Receive MailBox ch5 frm5 Data register	PSI5SjPMB55D	<PSI5Sj_base> + $069C_H$	*1	—
	PSI5S Receive MailBox ch5 frm5 Timestamp register	PSI5SjPMB55T	<PSI5Sj_base> + $06A0_H$	*1	—
	PSI5S Receive MailBox ch5 frm6 Status register	PSI5SjPMB56S	<PSI5Sj_base> + $06A4_H$	*1	—
	PSI5S Receive MailBox ch5 frm6 Data register	PSI5SjPMB56D	<PSI5Sj_base> + $06A8_H$	*1	—
	PSI5S Receive MailBox ch5 frm6 Timestamp register	PSI5SjPMB56T	<PSI5Sj_base> + $06AC_H$	*1	—
	PSI5S Receive MailBox ch6 frm1 Status register	PSI5SjPMB61S	<PSI5Sj_base> + $06B0_H$	*1	—

Table 28.12 List of Registers (8/8)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
PSI5S	PSI5S Receive MailBox ch6 frm1 Data register	PSI5SjPMB61D	<PSI5Sj_base> + $06B4_H$	*1	—
	PSI5S Receive MailBox ch6 frm1 Timestamp register	PSI5SjPMB61T	<PSI5Sj_base> + $06B8_H$	*1	—
	PSI5S Receive MailBox ch6 frm2 Status register	PSI5SjPMB62S	<PSI5Sj_base> + $06BC_H$	*1	—
	PSI5S Receive MailBox ch6 frm2 Data register	PSI5SjPMB62D	<PSI5Sj_base> + $06C0_H$	*1	—
	PSI5S Receive MailBox ch6 frm2 Timestamp register	PSI5SjPMB62T	<PSI5Sj_base> + $06C4_H$	*1	—
	PSI5S Receive MailBox ch6 frm3 Status register	PSI5SjPMB63S	<PSI5Sj_base> + $06C8_H$	*1	—
	PSI5S Receive MailBox ch6 frm3 Data register	PSI5SjPMB63D	<PSI5Sj_base> + $06CC_H$	*1	—
	PSI5S Receive MailBox ch6 frm3 Timestamp register	PSI5SjPMB63T	<PSI5Sj_base> + $06D0_H$	*1	—
	PSI5S Receive MailBox ch6 frm4 Status register	PSI5SjPMB64S	<PSI5Sj_base> + $06D4_H$	*1	—
	PSI5S Receive MailBox ch6 frm4 Data register	PSI5SjPMB64D	<PSI5Sj_base> + $06D8_H$	*1	—
	PSI5S Receive MailBox ch6 frm4 Timestamp register	PSI5SjPMB64T	<PSI5Sj_base> + $06DC_H$	*1	—
	PSI5S Receive MailBox ch6 frm5 Status register	PSI5SjPMB65S	<PSI5Sj_base> + $06E0_H$	*1	—
	PSI5S Receive MailBox ch6 frm5 Data register	PSI5SjPMB65D	<PSI5Sj_base> + $06E4_H$	*1	—
	PSI5S Receive MailBox ch6 frm5 Timestamp register	PSI5SjPMB65T	<PSI5Sj_base> + $06E8_H$	*1	—
	PSI5S Receive MailBox ch6 frm6 Status register	PSI5SjPMB66S	<PSI5Sj_base> + $06EC_H$	*1	—
	PSI5S Receive MailBox ch6 frm6 Data register	PSI5SjPMB66D	<PSI5Sj_base> + $06F0_H$	*1	—
	PSI5S Receive MailBox ch6 frm6 Timestamp register	PSI5SjPMB66T	<PSI5Sj_base> + $06F4_H$	*1	—
	PSI5S Receive MailBox ch7 frm1 Status register	PSI5SjPMB71S	<PSI5Sj_base> + $06F8_H$	*1	—
	PSI5S Receive MailBox ch7 frm1 Data register	PSI5SjPMB71D	<PSI5Sj_base> + $06FC_H$	*1	—
	PSI5S Receive MailBox ch7 frm1 Timestamp register	PSI5SjPMB71T	<PSI5Sj_base> + 0700_H	*1	—
	PSI5S Receive MailBox ch7 frm2 Status register	PSI5SjPMB72S	<PSI5Sj_base> + 0704_H	*1	—
	PSI5S Receive MailBox ch7 frm2 Data register	PSI5SjPMB72D	<PSI5Sj_base> + 0708_H	*1	—
	PSI5S Receive MailBox ch7 frm2 Timestamp register	PSI5SjPMB72T	<PSI5Sj_base> + $070C_H$	*1	—
	PSI5S Receive MailBox ch7 frm3 Status register	PSI5SjPMB73S	<PSI5Sj_base> + 0710_H	*1	—
	PSI5S Receive MailBox ch7 frm3 Data register	PSI5SjPMB73D	<PSI5Sj_base> + 0714_H	*1	—
	PSI5S Receive MailBox ch7 frm3 Timestamp register	PSI5SjPMB73T	<PSI5Sj_base> + 0718_H	*1	—
	PSI5S Receive MailBox ch7 frm4 Status register	PSI5SjPMB74S	<PSI5Sj_base> + $071C_H$	*1	—
	PSI5S Receive MailBox ch7 frm4 Data register	PSI5SjPMB74D	<PSI5Sj_base> + 0720_H	*1	—
	PSI5S Receive MailBox ch7 frm4 Timestamp register	PSI5SjPMB74T	<PSI5Sj_base> + 0724_H	*1	—
	PSI5S Receive MailBox ch7 frm5 Status register	PSI5SjPMB75S	<PSI5Sj_base> + 0728_H	*1	—
	PSI5S Receive MailBox ch7 frm5 Data register	PSI5SjPMB75D	<PSI5Sj_base> + $072C_H$	*1	—
	PSI5S Receive MailBox ch7 frm5 Timestamp register	PSI5SjPMB75T	<PSI5Sj_base> + 0730_H	*1	—
	PSI5S Receive MailBox ch7 frm6 Status register	PSI5SjPMB76S	<PSI5Sj_base> + 0734_H	*1	—
	PSI5S Receive MailBox ch7 frm6 Data register	PSI5SjPMB76D	<PSI5Sj_base> + 0738_H	*1	—
	PSI5S Receive MailBox ch7 frm6 Timestamp register	PSI5SjPMB76T	<PSI5Sj_base> + $073C_H$	*1	—

Note 1. j=0: PBG30#7
j=1: PBG30#8

28.3.2 Common Register/Config

28.3.2.1 PSI5SjPUOEB — PSI5S/UART Operation Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.13 PSI5SjPUOEB Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OPEN	Operation enable 0: Disable 1: Enable The value 1 can be written to this bit when PSI5SjPUOS.ACSTS is 0 (= configuration mode). The value 0 can be written to this bit at any time. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.2 PSI5SjPUOMD — PSI5S/UART Operation Mode Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.14 PSI5SjPUOMD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OPMD	Operation mode select 0: UART mode 1: PSI5S mode This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared when by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.3 PSI5SjPUOS — PSI5S/UART Operation Status Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTS	ACSTS	SWSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.15 PSI5SjPUOS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	MSTS	Mode status 0: UART mode (PSI5SjPUOEB.OPEN = 1, PSI5SjPUOMD.OPMD = 0) 1: PSI5S mode (PSI5SjPUOEB.OPEN = 1, PSI5SjPUOMD.OPMD = 1) This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	ACSTS	Active status 0: Not Active (Configuration mode: PSI5SjPUOEB.OPEN = 0) 1: Active (UART mode or PSI5S mode: PSI5SjPUOEB.OPEN = 1) This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	SWSTS	Software (SW) reset status 0: SW reset not asserted 1: SW reset asserted Writing 1 to PSI5SjPUSWR.SWRST sets this bit to 1. This bit is read as 1 during SW reset execution. After SW reset execution, the bit is cleared to 0.

28.3.2.4 PSI5SjPUNFST — PSI5S/UART Noise Filter Set Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $000C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NFSET
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.16 PSI5SjPUNFST Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	NFSET	Noise filter setting 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.5 PSI5SjPUSWR — PSI5/UART Software Reset Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.17 PSI5SjPUSWR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SWRST	Software (SW) reset 0: Ignored 1: Start software reset for PSI5S This bit is always read as 0.

28.3.2.6 PSI5SjPRMBC — PSI5S Receive MailBox Data Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MBCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.18 PSI5SjPRMBC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MBCLR	MailBox clear All MailBox data clear 0: Ignored 1: All MailBox data cleared to 0 This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode).

28.3.2.7 PSI5SjPUCLB — PSI5S/UART Communication Loop Back Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0020_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TMKV[6:0]							LBEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.19 PSI5SjPUCLB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 1	TMKV	Test Mode Key Values Test mode key values for loopback test function These bits are always read as 0.
0	LBEN	3rd write value of loopback test sequence enable 0: Disable 1: Enable Test function set sequence is described in **Section 28.5.11.1, Loopback Test Setting Procedure** This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.8 PSI5SjPUPTS — PSI5S/UART Rx/Tx Parity Set Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UTPRTY[1:0]		—	—	—	—	—	—	URPRTY[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 28.20 PSI5SjPUPTS Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 8	UTPRTY[1:0]	UART Tx Parity 0: Parity disable 1: Even parity 2: 0 parity (Parity is always 0) 3: Odd parity When the CPU sets parity to disable (= 00B), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Otherwise, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1 to 0	URPRTY[1:0]	UART Rx parity 0: Parity disable 1: Even parity 2: Parity don’t care 3: Odd parity When the CPU sets parity to disable (= 00B), a packet (UART frame) is composed of 1 start bit, 8 data bits and 1 stop bit (10 bits in total). Otherwise, a packet (UART frame) is composed of 1 start bit, 8 data bits, 1 parity bit and 1 stop bit (11 bits in total). These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.9 PSI5SjPUBCE — PSI5S/UART Baud Rate Clock Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0028_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCKEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.21 PSI5SjPUBCE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SCKEN	UART clock (Psis_tx_sclk) output enable 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.10 PSI5SjPUBPR — PSI5S/UART Baud Rate Parameter Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: <PSI5Sj_base> + $002C_H$

Value after reset: $0004\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RXOSMP[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKDIV[7:0]								—	SCKPRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.22 PSI5SjPUBPR Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	RXOSMP[3:0]	RX Over sample number 0 to 3: Setting prohibited 4: 5 samples … *x*: *x*+1 samples … 15: 16 samples These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 8	SCKDIV[7:0]	Clock divide value of UART SCLK (psis_tx_sclk) 0 = 1/1 1 = 1/2 … *x* = 1/(*x*+1) … 255 = 1/256 Writing to these bits are prohibited when PSI5SjPUBCE.SCKEN is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	SCKPRS[6:0]	Prescaler of UART SCLK (psis_tx_sclk) 0 = 1/1 1 = 1/2 … *x* = 1/(*x*+1) … 127 = 1/128 Writing to these bits are prohibited when PSI5SjPUBCE.SCKEN is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.11 PSI5SjPTPS — PSI5S Timestamp Prescaler Register

Access: This register can be read or written in 32-bit units.

Address: <PSI5Sj_base> + 0030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	TSPRSU[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TSPRSL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.23 PSI5SjPTPS Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	TSPRSU[9:0]	Timestamp prescaler (upper) These bits define the load data of the timestamp prescaler counter (upper 10 bits). These bits configure a max 1 ms enable pulse from a 1 µs enable pulse The timestamp prescaler counter (upper) loads this setting and starts counting down until it reaches 1. Loading and counting down count are continuously repeated. When the timestamp prescaler counter (upper) becomes 1, the timestamp tick is output. If the CPU sets 0 to these bits, this pulse is not output. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	TSPRSL[6:0]	Timestamp prescaler (lower) These bits define the load data of the timestamp prescaler counter (lower 7 bits). The timestamp prescaler counter (lower) loads this setting and starts counting down until it reaches 1. Loading and counting down count are continuously repeated. These bits configure a 1 µs clock from PCLK. When the timestamp prescaler counter (lower) becomes 1, a 1 µs enable pulse is output. When the CPU sets 0 to these bits, the 1 µs enable pulse is not output. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.12 PSI5SjPTCAS — PSI5S Timestamp Counter A Select Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCACLS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSCAEBS	—	—	—	—	—	—	—	TSCACKS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 28.24 PSI5SjPTCAS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TSCACLS	Timestamp counter A clear select 0: Signal generated by PSI5S is selected 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TSCAEBS	Timestamp counter A enable select 0: Signal generated by PSI5S is selected (= PSI5SjPTCAE.TSCAEB = 1 or PSI5SjPATCE.ATSCEB = 1). 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCACKS	Timestamp counter A clock select 0: Signal generated by PSI5S is selected 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.13 PSI5SjPTCBS — PSI5S Timestamp Counter B Select Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBCLS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSCBEBS	—	—	—	—	—	—	—	TSCBCKS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 28.25 PSI5SjPTCBS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TSCBCLS	Timestamp counter B clear select 0: Signal generated by PSI5S is selected 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TSCBEBS	Timestamp counter B enable select 0: Signal generated by PSI5S is selected 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCBCKS	Timestamp counter B clock select 0: Signal generated by PSI5S is selected 1: GTM output is selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.14 PSI5SjPTCAE — PSI5S Timestamp Counter A Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCAEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.26 PSI5SjPTCAE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCAEB	Timestamp counter A enable 0: Disable 1: Enable When PSI5SjPTCAS.TSCAEBS is 0, this bit is selected and is used for timestamp counter A. When PSI5SjPTCAS.TSCAEBS is 1, this bit is disabled. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.15 PSI5SjPTCAC — PSI5S Timestamp Counter A Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCACLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.27 PSI5SjPTCAC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCACLR	Timestamp counter A clear 0: Ignored 1: Clear timestamp counter A When PSI5SjPTCAS.TSCACLS is 0, this bit is enabled and uses timestamp counter A clear. When PSI5SjPTCAS.TSCACLS is 1, this bit is disabled. This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.2.16 PSI5SjPTCBE — PSI5S Timestamp Counter B Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0048_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.28 PSI5SjPTCBE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCBEB	Timestamp counter B enable 0: Disable 1: Enable When PSI5SjPTCBS.TSCBEBS is 0, this bit is enabled and is used for timestamp counter B. When PSI5SjPTCBS.TSCBEBS is 1, this bit is disabled. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.17 PSI5SjPTCBC — PSI5S Timestamp Counter B Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $004C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCBCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.29 PSI5SjPTCBC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCBCLR	Timestamp counter B clear 0: Ignored 1: Clears timestamp counter B When PSI5SjPTCBS.TSCBCLS is 0, this bit is enabled and uses timestamp counter B clear. When PSI5SjPTCBS.TSCBCLS is 1, this bit is not used. This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.2.18 PSI5SjPATCE — PSI5S All Timestamp Counter Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSCEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.30 PSI5SjPATCE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ATSCEB	All timestamp counter enable 0: Ignored Timestamp counter A is enabled according to the setting of PSI5SjPTCAE.TSCAEB Timestamp counter B is enabled according to the setting of PSI5SjPTCBE.TSCBEB 1: All timestamp counters are enabled regardless of the settings of PSI5SjPTCAE.TSCAEB / PSI5SjPTCBE.TSCBEB. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.19 PSI5SjPATCC — PSI5S All Timestamp Counter Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0054_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.31 PSI5SjPATCC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ATSCCLR	All Timestamp Counters Clear 0: Ignored 1: Clears all Timestamp counter When PSI5SjPTCAS.TSCACLS is 0, this bit is enabled clearing timestamp counter A. When PSI5SjPTCAS.TSCACLS is 1, this bit is disabled. When PSI5SjPTCBS.TSCBCLS is 0, this bit is enabled clearing timestamp counter B. When PSI5SjPTCBS.TSCBCLS is 1, this bit is disabled. This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.2.20 PSI5SjUCRIE — UART Communication Rx Interrupt Enable

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0058_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	IERFIN	IEROE	IERFE	IERPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 28.32 PSI5SjUCRIE Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	IERFIN	Interrupt enable of UART Rx finish flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTRFIN. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
2	IEROE	Interrupt enable of UART Rx overrun error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTROE. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	IERFE	Interrupt enable of UART Rx framing error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTRFE. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	IERPE	Interrupt enable of UART Rx parity error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTRPE. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.21 PSI5SjUCTIE — UART Communication Tx Interrupt Enable

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $005C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IETFIN	IETOWE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 28.33 PSI5SjUCTIE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IETFIN	Interrupt enable of UART Tx finish flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTTFIN. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	IETOWE	Interrupt enable of UART Tx overwrite error flag 0: Disable 1: Enable This bit controls interrupt by PSI5SjUCRS.UTTFOWE. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.2.22 PSI5SjUCDRE — UART Communication DMA Request Enable Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQEUTFN	DRQEURFN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 28.34 PSI5SjUCDRE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DRQEUTFN	DMA request enable at UART Tx finish 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	DRQEURFN	DMA request enable at UART Rx finish 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). This bit is cleared when writing 1 to PSI5SjPUSWR.SWRST.

28.3.3 Common Register/Rx

28.3.3.1 PSI5SjUCRD — UART Communication Rx Data Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UTRDT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.35 PSI5SjUCRD Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	UTRDT[7:0]	UART read data When the stop bit of the Rx UART frame is detected in UART mode, PSI5SjUCRD.UTRDT is stored. If the next UART frame finishes before the CPU has read this address, PSI5SjUCRS.UTROE is set to 1 and this register is overwritten with new data. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.3.2 PSI5SjUCRS — UART Communication Rx Status Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0074_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTRFIN	UTROE	UTRFE	UTRPE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.36 PSI5SjUCRS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	UTRFIN	UART Rx finish flag 0: A frame is not received successfully 1: A frame is received successfully When the stop bit of the UART frame is detected in UART mode without any error occurring, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch7_rx) occur. This bit is cleared by writing 1 to PSI5SjUCRSC.UTRFINCL. When PSI5SjUCRS.UTRFIN is set and 1 is written to PSI5SjUCRSC.UTRFINCL in the same clock cycle, PSI5SjUCRS.UTRFIN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
2	UTROE	UART Rx overrun error flag 0: No error 1: An overrun error detected If the stop bit of the next UART frame is detected in UART mode before the CPU has read PSI5SjUCRD.UTRDT, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjUCRSC.UTROECL. When PSI5SjUCRS.UTROE is set and 1 is written to PSI5SjUCRSC.UTROECL in the same clock cycle, PSI5SjUCRS.UTROE remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
1	UTRFE	UART Rx framing error flag 0: No error 1: A framing error detected If the stop bit of the UART frame bit is detected in UART mode when the bit detection value is 0, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. If an Rx overrun error occurs during stop bit detection, this bit remains 0. This bit is cleared by writing 1 to PSI5SjUCRSC.UTRFECL. When PSI5SjUCRS.UTRFE is set and 1 is written to PSI5SjUCRSC.UTRFECL in the same clock cycle, PSI5SjUCRS.UTRFE remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Table 28.36 PSI5SjUCRS Register Contents (2/2)

Bit Position	Bit Name	Function
0	UTRPE	UART Rx parity error 0: No error 1: A parity error detected If the stop bit of the UART frame is detected in UART mode when a parity error has been detected, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. If a framing error or Rx overrun error occurs during stop bit detection, this bit remains 0. This bit is cleared by writing 1 to PSI5SjUCRSC.UTRPECL. When PSI5SjUCRS.UTRPE is set and 1 is written to PSI5SjUCRSC.UTRPECL in the same clock cycle, PSI5SjUCRS.UTRPE remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.3.3 PSI5SjUCRSC — UART Communication Rx Status Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0078_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UTRFINCL	UTROECL	UTRFECL	UTRPECL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 28.37 PSI5SjUCRSC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	UTRFINCL	UART Rx finish flag clear 0: Ignored 1: UART Rx finish flag (PSI5SjUCRS.UTRFIN) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).
2	UTROECL	UART Rx overrun error clear 0: Ignored 1: Overrun error (PSI5SjUCRS.UTROE) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).
1	UTRFECL	UART Rx framing error clear 0: Ignored 1: Framing error (PSI5SjUCRS.UTRFE) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).
0	UTRPECL	UART Rx parity error clear 0: Ignored 1: Parity error (PSI5SjUCRS.UTRPE) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).

28.3.4 Common Register/Tx

28.3.4.1 PSI5SjPTFST — PSI5S Tx Frame Start Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0080_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.38 PSI5SjPTFST Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TXST	Tx start PSI5S command data transmission start 0: Ignored 1: Transmission start When the CPU writes 1 to this bit, 1-8 command data (PSI5SjPTFD1.TDT1 to PSI5SjPTFD2.TDT8) transmission starts. This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.4.2 PSI5SjPTFNM — PSI5S Tx Frame Number Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0084_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXNUM[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 28.39 PSI5SjPTFNM Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	TXNUM[2:0]	Tx command data number 000_B: 1 packet (UART frame) send (PSI5SjPTFD1.TDT1) 001_B: 2 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD1.TDT2) 010_B: 3 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD1.TDT3) 011_B: 4 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD1.TDT4) 100_B: 5 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD2.TDT5) 101_B: 6 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD2.TDT6) 110_B: 7 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD2.TDT7) 111_B: 8 packet (UART frame) send (PSI5SjPTFD1.TDT1 to PSI5SjPTFD2.TDT8) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.4.3 PSI5SjPTFD1 — PSI5S Tx Frame Data1 Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0088_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDT4[7:0]								TDT3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDT2[7:0]								TDT1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.40 PSI5SjPTFD1 Register Contents

Bit Position	Bit Name	Function
31 to 24	TDT4[7:0]	These bits define the transmission data of the 4th packet (UART frame). When PSI5SjPTFNM.TXNUM is less than 3, this setting is ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
23 to 16	TDT3[7:0]	These bits define the transmission data of the 3rd packet (UART frame). When PSI5SjPTFNM.TXNUM is less than 2, this setting is ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 8	TDT2[7:0]	These bits define the transmission data of the 2nd packet (UART frame). When PSI5SjPTFNM.TXNUM is less than 1, this setting is ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 0	TDT1[7:0]	These bits define the transmission data of the 1st packet (UART frame). These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.4.4 PSI5SjPTFD2 — PSI5S Tx Frame Data2 Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $008C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDT8[7:0]								TDT7[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDT6[7:0]								TDT5[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.41 PSI5SjPTFD2 Register Contents

Bit Position	Bit Name	Function
31 to 24	TDT8[7:0]	These bits define the transmission data of the 8th packet. When PSI5SjPTFNM.TXNUM is less than 7, these setting are ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
23 to 16	TDT7[7:0]	These bits define the transmission data of the 7th packet. When PSI5SjPTFNM.TXNUM is less than 6, these setting are ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
15 to 8	TDT6[7:0]	These bits define the transmission data of the 6th packet. When PSI5SjPTFNM.TXNUM is less than 5, these setting are ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 0	TDT5[7:0]	These bits define the transmission data of the 5th packet. When PSI5SjPTFNM.TXNUM is less than 4, these setting are ignored. These bits cannot be written when PSI5SjPTFS.TXSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.4.5 PSI5SjPTFS — PSI5S Tx Frame Status Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0090_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.42 PSI5SjPTFS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	TXSTS	Tx status 0: Transmission not busy 1: Transmission busy When the CPU sets 1 to PSI5SjPTFST.TXST, this bit is set to 1. When all transmission data (PSI5SjPTFD1, PSI5SjPTFD2) has been sent to the Tx shifter, this bit is reset to 0. When this bit is 1, PSI5SjPTFD1 and PSI5SjPTFD2 cannot be written. This bit is read only. The write value is ignored. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.4.6 PSI5SjPTFIS — PSI5S Tx FIFO Status Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0094_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXFFFL	TXFFEP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.43 PSI5SjPTFIS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	TXFFFL	Tx FIFO full 0: Not full 1: Full This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	TXFFEP	Tx FIFO empty 0: Not empty 1: Empty This bit is set to 1 by writing 1 to PSI5SjPUSWR.SWRST. This bit is set to 1 when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.4.7 PSI5SjUCTD — UART Communication Tx Data Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $00A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UTTDT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.44 PSI5SjUCTD Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	UTTDT[7:0]	UART transmission data These bits cannot be written when PSI5SjUCTM.UTTBBF is 1. When the CPU writes PSI5SjUCTD.UTTDT while PSI5SjUCTM.UTTBBF is 1, PSI5SjUCTS.UTTOWE is set to 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.4.8 PSI5SjUCTM — UART Communication Tx Monitoring Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $00A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTF	UTTBBF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.45 PSI5SjUCTM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	UTTF	UART transmission flag 0: Not transmitting 1: Transmitting When stop bit output ends without any subsequent write data, this bit is reset to 0. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	UTTBBF	UART Tx shifter busy flag 0: Permitted to write to PSI5SjUCTD.UTTDT 1: Prohibited to write to PSI5SjUCTD.UTTDT This bit shows status of Tx shifter busy in UART mode. When CPU writes PSI5SjUCTD.UTTDT, this bit is set to 1. When PSI5SjUCTD.UTTDT data is stored in the Tx shifter, this bit is reset to 0. When this bit is 1, PSI5SjUCTD.UTTDT cannot be written. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.4.9 PSI5SjUCTS — UART Communication Tx Status Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $00A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTFIN	UTTOWE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.46 PSI5SjUCTS Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	UTTFIN	UART transmission finish 0: Transmission not finished 1: A UART frame transmission has finished When PSI5SjUCTD.UTTDT data is sent to the Tx shifter, this bit is set to 1 and an interrupt (int_psis_ch1) and DMA request (dma_psis_ch7_tx) occur. This bit is cleared by writing 1 to PSI5SjUCTSC.UTTFINCL. When PSI5SjUCTS.UTTFIN is set and 1 is written to PSI5SjUCTSC.UTTFINCL in the same clock cycle, PSI5SjUCTS.UTTFIN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	UTTOWE	UART transmission overwrite error 0: No error 1: Overwrite error When the CPU writes 1 to PSI5SjUCTD.UTTDT in PSI5SjUCTM.UTTBBF, this bit sets to 1 and an interrupt (int_psis_ch1) occurs. This bit is cleared by writing 1 to PSI5SjUCTSC.UTTOWECL. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.4.10 PSI5SjUCTSC — UART Communication Tx Status Clear Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $00AC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UTTFINCL	UTTOWECL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 28.47 PSI5SjUCTSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	UTTFINCL	UART Tx finish flag clear 0: Ignored 1: UART Tx finish flag (PSI5SjUCTS.UTTFIN) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).
0	UTTOWECL	UART Tx overwrite error clear 0: Ignored 1: Overwrite error (PSI5SjUCTS.UTTOWE) clear This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 0 (= UART mode).

28.3.5 Ch0 Register/Config

28.3.5.1 PSI5SjPRCF10 — PSI5S Receive Config1 ch0 Register

Access: This register can be read or written in 32-bit units.

Address: <PSI5Sj_base> + 0100_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PFRMIDLE[3:0]				F6PKT[2:0]			F5PKT[2:0]			F4PKT[2:1]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PKT[0]	F3PKT[2:0]			F2PKT[2:0]			F1PKT[2:0]			—*1	—*2	TSCS	TSEN	RFCPS	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Table 28.48 PSI5SjPRCF10 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	PFRMIDLE[3:0]	Packet frame idle Minimum packet frame gap (all channels) 0: The next packet frame is detected after 1 gap. 1: The next packet frame is detected after 2 gaps. ... n: The next packet frame is detected after n+1 gaps. ... 15: The next packet frame is detected after 16 gaps. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
23 to 21	F6PKT[2:0]	Frame 6 packet number (Ch0) 000_B: Frame 6 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 6” is a packet frame whose FID is 101_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
20 to 18	F5PKT[2:0]	Frame 5 packet number (Ch0) 000_B: Frame 5 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 5” is a packet frame whose FID is 100_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.48 PSI5SjPRCF10 Register Contents (2/3)

Bit Position	Bit Name	Function
17 to 15	F4PKT[2:0]	Frame 4 packet number (Ch0) 000_B: Frame 4 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 4” is a packet frame whose FID is 011_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
14 to 12	F3PKT[2:0]	Frame 3 packet number (Ch0) 000_B: Frame 3 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 3” is a packet frame whose FID is 010_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
11 to 9	F2PKT[2:0]	Frame 2 packet number (Ch0) 000_B: Frame 2 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 2” is a packet frame whose FID is 001_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared when writing 1 to PSI5SjPUSWR.SWRST.
8 to 6	F1PKT[2:0]	Frame 1 packet number (Ch0) 000_B: Frame 1 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited “Frame 1” is a packet frame whose FID is 000_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared when writing 1 to PSI5SjPUSWR.SWRST.
5 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.*1, *2
3	TSCS	Timestamp counter select (Ch0) 0: Select timestamp counter B 1: Select timestamp counter A This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
2	TSEN	Timestamp capture enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	RFCPS	Rx frame checksum CRC/parity select (Ch0) 0: Parity select 1: CRC select This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.48 PSI5SjPRCF10 Register Contents (3/3)

Bit Position	Bit Name	Function
0	CHEN	Channel enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note 1. bit[5]: Channel 0 does not have SYSEL, because channel 0 does not have synchronous mode.

Note 2. bit[4]: Channel 0 does not have TSCTS (Timestamp capture trigger select), because channel 0 is always the header receive timing is selected.

28.3.5.2 PSI5SjPRCF20 — PSI5S Receive Config2 ch0 Register

Access: This register can be read or written in 32-bit units.

Address: <PSI5Sj_base> + 0104_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	F6PAYLD[4:0]					F5PAYLD[4:0]					F4PAYLD[4:1]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PAYLD[0]	F3PAYLD[4:0]					F2PAYLD[4:0]					F1PAYLD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.49 PSI5SjPRCF20 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 25	F6PAYLD[4:0]	The payload bit length of packet frame 6 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
24 to 20	F5PAYLD[4:0]	The payload bit length of packet frame 5 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
19 to 15	F4PAYLD[4:0]	The payload bit length of packet frame 4 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.49 PSI5SjPRCF20 Register Contents (2/2)

Bit Position	Bit Name	Function
14 to 10	F3PAYLD[4:0]	The payload bit length packet frame 3 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
9 to 5	F2PAYLD[4:0]	The payload bit length of packet frame 2 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
4 to 0	F1PAYLD[4:0]	The payload bit length of packet frame 1 (Ch0) 8: Sets 8 … *x*: Sets *x* … 28: Sets 28 Other than above: Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.5.3 PSI5SjPWDE0 — PSI5S WDT Enable ch0 Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0108_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.50 PSI5SjPWDE0 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	WDTEB	Watchdog timer of Rx frame enable (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.5.4 PSI5SjPWDP0 — PSI5S WDT Prescaler ch0 Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: <PSI5Sj_base> + $010C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.51 PSI5SjPWDP0 Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	WDTPRS[11:0]	Watchdog timer prescaler (Ch0) 0: Stop watchdog timer 1 to 4095: Enabled at 1 clock/*x* clock (*x*:1 to 4095) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.5.5 PSI5SjPWDEV0 — PSI5S WDT Expiration Value ch0 Register

Access: This register can be read or written in 32-bit units.

Address: <PSI5Sj_base> + 0110_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WDTEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.52 PSI5SjPWDEV0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	WDTEX[23:0]	Watchdog timer expiration value (Ch0) These bits define the expiration value of the watchdog timer for Rx packet frame monitor in Ch0. When the watchdog counter counts down to 0 from this setting value, the watchdog timer is judged to have expired. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.5.6 PSI5SjPCIE0 — PSI5S CPU Interrupt Enable Ch0 Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: <PSI5Sj_base> + 0118_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IEBCTFN	—	—	IEBRFN	IEBRFEX	IEBRFLK	IEBROV	IEBRWDT	—	IEBUTFR	IEBUTPT	IEBTRST	IEBPT	IEBCRC	IEBXCRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.53 PSI5SjPCIE0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	IEBCTFN	Interrupt enable of command Tx finish (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
13 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	IEBRFN	Interrupt enable of Rx packet frame finish flag (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
10	IEBRFEX	Interrupt enable of Rx frame excess error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
9	IEBRFLK	Interrupt enable of Rx frame lack error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
8	IEBROV	Interrupt enable of Rx overrun error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.53 PSI5SjPCIE0 Register Contents (2/2)

Bit Position	Bit Name	Function
7	IEBRWDT	Interrupt enable Rx WDT error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	IEBUTFR	Interrupt enable of Rx UART framing error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
4	IEBUTPT	Interrupt enable of Rx UART parity error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
3	IEBTRST	Interrupt enable of Rx transceiver status error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
2	IEBPT	Interrupt enable of Rx payload data parity error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	IEBCRC	Interrupt enable of Rx payload data CRC error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	IEBXCRC	Interrupt enable of packet frame XCRC error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.5.7 PSI5SjPDRE0 — PSI5S DMA transfer Request Enable ch0 Register

Access: This register can be read or written in 32-bit units.

Address: <PSI5Sj_base> + $011C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQEWDT	DRQERFN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 28.54 PSI5SjPDRE0 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DRQEWDT	Enable of DMA request by the WDT error (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	DRQERFN	Enable of DMA request by the channel data Rx finish (Ch0) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.6 Ch0 Register/Rx

28.3.6.1 PSI5SjPRES0 — PSI5S Receive Error Status ch0 Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0130_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—*1	—*1	—*1	—*1	RERRF2	RERRF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.55 PSI5SjPRES0 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	RERRF2	Rx error at packet frame 2 (Ch0) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “Rx WDT error”, “UART framing error”, “UART parity error”, “transceiver status error”, “payload data parity error”, “payload data CRC error” or “packet frame XCRC error” in packet frame2 of Ch0 has occurred. This bit is cleared when writing 1 to PSI5SjPRESC0.RERRCLF2. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	RERRF1	Rx error at packet frame 1 (Ch0) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame1 of Ch0 has occurred. This bit is cleared when writing 1 to PSI5SjPRESC0.RERRCLF1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Note 1. Ch0 only has frame 1 and frame2. (Other channels have frame 1 to frame 6.)

28.3.6.2 PSI5SjPRESC0 — PSI5S Receive Error Status Clear ch0 Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0134_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—*1	—*1	—*1	—*1	RERRCLF2	RERRCLF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 28.56 PSI5SjPRESC0 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RERRCLF2	Rx error clear for packet frame 2 (Ch0) 0: Ignored 1: Clears PSI5SjPRES0.RERRF2 This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
0	RERRCLF1	Rx error clear for packet frame 1(Ch0) 0: Ignored 1: Clears PSI5SjPRES0.RERRF1 This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Note 1. Ch0 only has packet frame 1 and packet frame2. (Other channels have frame1 to frame 6.)

28.3.6.3 PSI5SjPTCDT0 — PSI5S Timestamp Capture Data ch0 Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + 0138_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSCD[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSCD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.57 PSI5SjPTCDT0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	TSCD[23:0]	Timestamp capture data (Ch0) These bits are cleared by writing 1 to PSI5SjPTCDC0.TSCCLR. These bits are cleared by writing 0 to PSI5SjPRCF10.TSEN. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.6.4 PSI5SjPTCDC0 — PSI5S Timestamp Capture Data Clear ch0 Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PSI5Sj_base> + $013C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.58 PSI5SjPTCDC0 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCCLR	Timestamp capture clear(Ch0) 0: Ignored 1: Clears timestamp capture This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.7 Ch0 Register/Interrupt

28.3.7.1 PSI5SjPCIS0 — PSI5S CPU Interrupt Status ch0 Register

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <PSI5Sj_base> + 0150_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ISTCTFN	—	—	ISTRFN	ISTRFEX	ISTRFLK	ISTROV	ISTRWDT	—	ISTUTFR	ISTUTPT	ISTTRST	ISTPT	ISTCRC	ISTXCRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.59 PSI5SjPCIS0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	ISTCTFN	Interrupt status of command Tx finish 0: PSI5S command transmission has not finished 1: PSI5S command transmission has finished When the last command sent to the Tx shifter, a PSI5S frame is stored in a MB and the frame has no errors, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCCTFN. When PSI5SjPCIS0.ISTCTFN is set and 1 is written to PSI5SjPCISC0.ISTCCTFN in the same clock cycle, PSI5SjPCIS0.ISTCTFN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
13 to 12	Reserved	When read, the value after reset is returned.
11	ISTRFN	Interrupt status of Rx finish (Ch0) 0: PSI5S frame is not received successfully 1: PSI5S frame is received successfully When a PSI5S frame is stored in a MB and the PSI5S frame has no errors, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCRFN. When PSI5SjPCIS0.ISTRFN is set and 1 is written to PSI5SjPCISC0.ISTCRFN in the same clock cycle, PSI5SjPCIS0.ISTRFN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Table 28.59 PSI5SjPCIS0 Register Contents (2/3)

Bit Position	Bit Name	Function
10	ISTRFEX	Interrupt status Rx frame excess error (Ch0) 0: No error 1: Error detected When a packet over PSI5SjPRCF10.FmPKT(m = 1 to 6) is received, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCRFEX. When PSI5SjPCIS0.ISTRFEX is set and 1 is written to PSI5SjPCISC0.ISCTRFEX in the same clock cycle, PSI5SjPCIS0.ISTRFEX remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
9	ISTRFLK	Interrupt status of Rx frame lack error (Ch0) 0: No error 1: Error detected When a packet frame gap is detected while the packet is under PSI5SjPRCF10.FmPKT (m=1 to 6), this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCRFLK. When PSI5SjPCIS0.ISTRFLK is set and 1 is written to PSI5SjPCISC0.ISTCRFLK in the same clock cycle, PSI5SjPCIS0.ISTRFLK remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
8	ISTROV	Interrupt status of Rx overrun error (Ch0) 0: No error 1: Error detected When an attempt is made to store the next PSI5S frame in a MB before the CPU has read that MB or set PSI5SjPRMBC.MBCLR, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCROV. When PSI5SjPCIS0.ISTROV is set and 1 is written to PSI5SjPCISC0.ISTCROV in the same clock cycle, PSI5SjPCIS0.ISTROV remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
7	ISTRWDT	Interrupt status of Rx WDT error (Ch0) 0: No error 1: Error detected When a WDT error occurs in PSI5S mode, this bit is set to 1 and an interrupt (int_psis_ch0) and DMA request (dma_psis_ch0_rx) occur. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCRWDT. When PSI5SjPCIS0.ISTRWDT is set and 1 is written to PSI5SjPCISC0.ISTCRWDT in the same clock cycle, PSI5SjPCIS0.ISTRWDT remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
6	Reserved	When read, the value after reset is returned.
5	ISTUTFR	Interrupt status of Rx UART framing error (Ch0) 0: No error 1: Error detected If the stop bit of the UART frame is detected in PSI5S mode when the bit detection value is 0, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCUTFR. When PSI5SjPCIS0.ISTUTFR is set and 1 is written to PSI5SjPCISC0.ISTCUTFR in the same clock cycle, PSI5SjPCIS0.ISTUTFR remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Table 28.59 PSI5SjPCIS0 Register Contents (3/3)

Bit Position	Bit Name	Function
4	ISTUTPT	Interrupt status of Rx UART parity error (Ch0) 0: No error 1: Error detected If the stop bit of the UART frame is detected in PSI5S mode when a UART parity error has occurred, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCUTPT. When PSI5SjPCIS0.ISTUTPT is set and 1 is written to PSI5SjPCISC0.ISTCUTPT in the same clock cycle, PSI5SjPCIS0.ISTUTPT remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
3	ISTTRST	Interrupt status of Rx transceiver status error (Ch0) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a transceiver status error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCTRST. When PSI5SjPCIS0.ISTTRST is set and 1 is written to PSI5SjPCISC0.ISTCTRST in the same clock cycle, PSI5SjPCIS0.ISTTRST remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
2	ISTPT	Interrupt status of payload data parity error (Ch0) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a payload data parity error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCPT. When PSI5SjPCIS0.ISTPT is set and 1 is written to PSI5SjPCISC0.ISTCPT in the same clock cycle, PSI5SjPCIS0.ISTPT remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
1	ISTCRC	Interrupt status of payload data CRC error (Ch0) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a payload data CRC error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCCRC. When PSI5SjPCIS0.ISTCRC is set and 1 is written to PSI5SjPCISC0.ISTCCRC in the same clock cycle, PSI5SjPCIS0.ISTCRC remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	ISTXCRC	Interrupt status of packet frame XCRC error (Ch0) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and an XCRC error occurs, this bit is set to 1 and an interrupt (int_psis_ch0) occurs. This bit is cleared by writing 1 to PSI5SjPCISC0.ISTCXCRC. When PSI5SjPCIS0.ISTXCRC is set and 1 is written to PSI5SjPCISC0.ISTCXCRC in the same clock cycle, PSI5SjPCIS0.ISTXCRC remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

28.3.7.2 PSI5SjPCISC0 — PSI5S CPU Interrupt Status Clear ch0 Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: <PSI5Sj_base> + 0154_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ISTCCTFN	—	—	ISTCRFN	ISTCRFEX	ISTCRFLK	ISTCROV	ISTCRWDT	—	ISTCUTFR	ISTCUTPT	ISTCTRST	ISTCPT	ISTCCRC	ISTCXCRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.60 PSI5SjPCISC0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	ISTCCTFN	Interrupt status of command Tx finish (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTCTFN This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
13 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ISTCRFN	Interrupt status of Rx finish (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTRFN This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
10	ISTCRFEX	Interrupt status of Rx frame excess error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTRFEX This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
9	ISTCRFLK	Interrupt status of Rx frame lack error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTRFLK This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
8	ISTCROV	Interrupt status of Rx overrun error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTROV This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Table 28.60 PSI5SjPCISC0 Register Contents (2/2)

Bit Position	Bit Name	Function
7	ISTCRWDT	Interrupt status of Rx WDT error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTRWDT This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	ISTCUTFR	Interrupt status of Rx UART framing error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTUTFR This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
4	ISTCUTPT	Interrupt status of Rx UART parity error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTUTPT This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
3	ISTCTRST	Interrupt status of Rx transceiver status error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTTRST This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
2	ISTCPT	Interrupt status of payload data parity error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTPT This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
1	ISTCCRC	Interrupt status of payload data CRC error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTCRC This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
0	ISTCXCRC	Interrupt status of packet frame XCRC error (Ch0) 0: Ignored 1: Clears PSI5SjPCIS0.ISTXCRC This bit is always read as 0. This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

28.3.8 Ch *n* Register/Config (*n*: 1 to 7)

28.3.8.1 PSI5SjPRCF1n — PSI5S Receive Config1 chn Register (n: 1 to 7)

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPRCF11: <PSI5Sj_base> + 0180_H
PSI5SjPRCF12: <PSI5Sj_base> + 0200_H
PSI5SjPRCF13: <PSI5Sj_base> + 0280_H
PSI5SjPRCF14: <PSI5Sj_base> + 0300_H
PSI5SjPRCF15: <PSI5Sj_base> + 0380_H
PSI5SjPRCF16: <PSI5Sj_base> + 0400_H
PSI5SjPRCF17: <PSI5Sj_base> + 0480_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	F6PKT[2:0]			F5PKT[2:0]			F4PKT[2:1]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PKT[0]	F3PKT[2:0]			F2PKT[2:0]			F1PKT[2:0]			SYSEL	TSCTS	TSCS	TSEN	RFCPS	CHEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.61 PSI5SjPRCF1n Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 21	F6PKT[2:0]	Frame 6 packet number (Ch *n*) 000_B: Frame 6 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. “Frame 6” is the packet frame whose FID is 101_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
20 to 18	F5PKT[2:0]	Frame 5 packet number (Ch *n*) 000_B: Frame 5 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. “Frame 5” is the packet frame whose FID is 100_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.61 PSI5SjPRCF1n Register Contents (2/3)

Bit Position	Bit Name	Function
17 to 15	F4PKT[2:0]	Frame 4 packet number (Ch *n*) 000_B: Frame 4 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. "Frame 4" is the packet frame whose FID is 011_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
14 to 12	F3PKT[2:0]	Frame 3 packet number (Ch *n*) 000_B: Frame 3 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. "Frame 3" is the packet frame whose FID is 010_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
11 to 9	F2PKT[2:0]	Frame 2 packet number (Ch *n*) 000_B: Frame 2 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. "Frame 2" is the packet frame whose FID is 001_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
8 to 6	F1PKT[2:0]	Frame 1 packet number (Ch *n*) 000_B: Frame 1 data is ignored 011_B: Packet number is set to 3 100_B: Packet number is set to 4 101_B: Packet number is set to 5 110_B: Packet number is set to 6 Other than above: Setting prohibited The packet number means the number of UART frames number per packet frame. "Frame 1" is the packet frame whose FID is 000_B. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
5	SYSEL	Asynchronous mode/synchronous mode select 0: Synchronous mode 1: Asynchronous mode This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
4	TSCTS	Timestamp capture trigger select (Ch *n*) 0: Transmission synchronous pulse timing selected 1: Header receive timing selected (CH *n*) This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.61 PSI5SjPRCF1n Register Contents (3/3)

Bit Position	Bit Name	Function
3	TSCS	Timestamp counter select (Ch *n*) 0: Timestamp counter B selected 1: Timestamp counter A selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
2	TSEN	Timestamp capture enable (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	RFCPS	Rx frame checksum CRC/parity select (Ch n) 0: Parity selected 1: CRC selected This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	CHEN	Channel enable (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.2 PSI5SjPRCF2n — PSI5S Receive Config2 chn Register

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPRCF21: <PSI5Sj_base> + 0184_H
PSI5SjPRCF22: <PSI5Sj_base> + 0204_H
PSI5SjPRCF23: <PSI5Sj_base> + 0284_H
PSI5SjPRCF24: <PSI5Sj_base> + 0304_H
PSI5SjPRCF25: <PSI5Sj_base> + 0384_H
PSI5SjPRCF26: <PSI5Sj_base> + 0404_H
PSI5SjPRCF27: <PSI5Sj_base> + 0484_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	F6PAYLD[4:0]					F5PAYLD[4:0]					F4PAYLD[4:1]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F4PAYLD[0]	F3PAYLD[4:0]					F2PAYLD[4:0]					F1PAYLD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.62 PSI5SjPRCF2n Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 25	F6PAYLD[4:0]	The payload bit length of packet frame 6 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope (*n*:1 to 7). These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
24 to 20	F5PAYLD[4:0]	The payload bit length of packet frame 5 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine about the CRC/parity calculation scope (*n*:1 to 7). These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.62 PSI5SjPRCF2n Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 15	F4PAYLD[4:0]	The payload bit length of packet frame 4 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope.(*n*:1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
14 to 10	F3PAYLD[4:0]	The payload bit length of packet frame 3 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope. (*n*:1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
9 to 5	F2PAYLD[4:0]	The payload bit length of packet frame 2 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope.(*n*:1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
4 to 0	F1PAYLD[4:0]	The payload bit length of packet frame 1 (Ch *n*) (Refer to **Figure 28.3** about payload) 0 to 7 : Setting prohibited 8 : Sets 8 … x : Sets x … 28 : Sets 28 29 to 31 : Setting prohibited If the initial value (0) is set to these bits, it is handled as 8. These bits are used to determine the CRC/parity calculation scope.(*n*:1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.3 PSI5SjPWDEn — PSI5S WDT Enable chn Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPWDE1: <PSI5Sj_base> + 0188_H
PSI5SjPWDE2: <PSI5Sj_base> + 0208_H
PSI5SjPWDE3: <PSI5Sj_base> + 0288_H
PSI5SjPWDE4: <PSI5Sj_base> + 0308_H
PSI5SjPWDE5: <PSI5Sj_base> + 0388_H
PSI5SjPWDE6: <PSI5Sj_base> + 0408_H
PSI5SjPWDE7: <PSI5Sj_base> + 0488_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTEB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.63 PSI5SjPWDEn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	WDTEB	Watchdog Timer of Rx frame Enable (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) and when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.4 PSI5SjPWDPn — PSI5S WDT Prescaler chn Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: PSI5SjPWDP1: <PSI5Sj_base> + $018C_H$
PSI5SjPWDP2: <PSI5Sj_base> + $020C_H$
PSI5SjPWDP3: <PSI5Sj_base> + $028C_H$
PSI5SjPWDP4: <PSI5Sj_base> + $030C_H$
PSI5SjPWDP5: <PSI5Sj_base> + $038C_H$
PSI5SjPWDP6: <PSI5Sj_base> + $040C_H$
PSI5SjPWDP7: <PSI5Sj_base> + $048C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.64 PSI5SjPWDPn Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	WDTPRS[11:0]	Watchdog timer prescaler (Ch *n*) 0: Stop WDT timer 1 to 4095: Enabled at 1 clock/*x* clock (*x*:1 to 4095) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.5 PSI5SjPWDEVn — PSI5S WDT Expiration Value chn Register

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPWDEV1: <PSI5Sj_base> + 0190_H
PSI5SjPWDEV2: <PSI5Sj_base> + 0210_H
PSI5SjPWDEV3: <PSI5Sj_base> + 0290_H
PSI5SjPWDEV4: <PSI5Sj_base> + 0310_H
PSI5SjPWDEV5: <PSI5Sj_base> + 0390_H
PSI5SjPWDEV6: <PSI5Sj_base> + 0410_H
PSI5SjPWDEV7: <PSI5Sj_base> + 0490_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WDTEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.65 PSI5SjPWDEVn Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	WDTEX[23:0]	Watchdog timer expiration value (Ch *n*) When the watchdog counter counts down to 0 from this setting value, the watchdog timer is judged to have expired.(*n*: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.6 PSI5SjPTCDn — PSI5S Tx Command Data chn Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: PSI5SjPTCD1: <PSI5Sj_base> + 0194_H
PSI5SjPTCD2: <PSI5Sj_base> + 0214_H
PSI5SjPTCD3: <PSI5Sj_base> + 0294_H
PSI5SjPTCD4: <PSI5Sj_base> + 0314_H
PSI5SjPTCD5: <PSI5Sj_base> + 0394_H
PSI5SjPTCD6: <PSI5Sj_base> + 0414_H
PSI5SjPTCD7: <PSI5Sj_base> + 0494_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATRSCMD[4:0]					ACHID[2:0]			TRSCMD[4:0]					CHID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.66 PSI5SjPTCDn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 11	ATRSCMD[4:0]	Alternate transport command (Ch *n*) When ECU-to-sensor data is 1, select this command.(*n*: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
10 to 8	ACHID[2:0]	Alternate transport ChID (Ch *n*) When ECU-to-sensor data is 1, select this ChID.(*n*: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
7 to 3	TRSCMD[4:0]	Transport command (Ch *n*) When ECU-to-sensor data is 0, select this command.(*n*: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
2 to 0	CHID[2:0]	Transport ChID (Ch *n*) When ECU-to-sensor data is 0, select this ChID.(*n*: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.7 PSI5SjPCIEn — PSI5S CPU Interrupt Enable chn Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: PSI5SjPCIE1: <PSI5Sj_base> + 0198_H
PSI5SjPCIE2: <PSI5Sj_base> + 0218_H
PSI5SjPCIE3: <PSI5Sj_base> + 0298_H
PSI5SjPCIE4: <PSI5Sj_base> + 0318_H
PSI5SjPCIE5: <PSI5Sj_base> + 0398_H
PSI5SjPCIE6: <PSI5Sj_base> + 0418_H
PSI5SjPCIE7: <PSI5Sj_base> + 0498_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	IEBDDSFN	IEBDDSOW	IEBRFN	IEBRFEX	IEBRFLK	IEBROV	IEBRWDT	—	—	—	IEBTRST	IEBPT	IEBCRC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R

Table 28.67 PSI5SjPCIEn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	IEBDDSFN	Interrupt enable of DDSR finish flag (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
12	IEBDDSOW	Interrupt enable of DDSR overwrite (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
11	IEBRFN	Interrupt enable of Rx frame finish flag (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
10	IEBRFEX	Interrupt enable of Rx frame excess error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
9	IEBRFLK	Interrupt enable of Rx frame lack error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.67 PSI5SjPCIEn Register Contents (2/2)

Bit Position	Bit Name	Function
8	IEBROV	Interrupt enable of Rx overrun error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
7	IEBRWDT	Interrupt enable of Rx frame WDT error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	IEBTRST	Interrupt enable of Rx transceiver status error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
2	IEBPT	Interrupt enable of parity error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	IEBCRC	Interrupt enable of CRC error (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note: n: 1 to 7

28.3.8.8 PSI5SjPDREn — PSI5S DMA Transfer Request Enable chn Register

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPDRE1: <PSI5Sj_base> + $019C_H$
PSI5SjPDRE2: <PSI5Sj_base> + $021C_H$
PSI5SjPDRE3: <PSI5Sj_base> + $029C_H$
PSI5SjPDRE4: <PSI5Sj_base> + $031C_H$
PSI5SjPDRE5: <PSI5Sj_base> + $039C_H$
PSI5SjPDRE6: <PSI5Sj_base> + $041C_H$
PSI5SjPDRE7: <PSI5Sj_base> + $049C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQETFN	DRQEWDT	DRQERFN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 28.68 PSI5SjPDREn Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	DRQETFN	DMA request enable at ddsr Tx finish (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
1	DRQEWDT	DMA request enable at WDT (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
0	DRQERFN	DMA request enable at Rx finish (Ch *n*) 0: Disable 1: Enable This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.9 PSI5SjPSTPn — PSI5S Sync Trigger Prescaler chn Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: PSI5SjPSTP1: <PSI5Sj_base> + $01A4_H$
PSI5SjPSTP2: <PSI5Sj_base> + 0224_H
PSI5SjPSTP3: <PSI5Sj_base> + $02A4_H$
PSI5SjPSTP4: <PSI5Sj_base> + 0324_H
PSI5SjPSTP5: <PSI5Sj_base> + $03A4_H$
PSI5SjPSTP6: <PSI5Sj_base> + 0424_H
PSI5SjPSTP7: <PSI5Sj_base> + $04A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	STPRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.69 PSI5SjPSTPn Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	STPRS[11:0]	Synchronous trigger generation counter's prescaler (Ch *n*) 0: Enabled at 1 clock/1 clock 1: Enabled at 1 clock/2 clock … *x*: Enabled at 1 clock/(*x*+1) clock … 4095: Enabled at 1 clock/4096 clock When PSI5SjPSTSn.STSEL is 0, these bits are enabled. When PSI5SjPSTSn.STSEL is 1, these bits are disabled. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.10 PSI5SjPSTEVn — PSI5S Sync Trigger Expiration Value chn Register

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPSTEV1: <PSI5Sj_base> + $01A8_H$
PSI5SjPSTEV2: <PSI5Sj_base> + 0228_H
PSI5SjPSTEV3: <PSI5Sj_base> + $02A8_H$
PSI5SjPSTEV4: <PSI5Sj_base> + 0328_H
PSI5SjPSTEV5: <PSI5Sj_base> + $03A8_H$
PSI5SjPSTEV6: <PSI5Sj_base> + 0428_H
PSI5SjPSTEV7: <PSI5Sj_base> + $04A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEX[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STEX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.70 PSI5SjPSTEVn Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	STEX[23:0]	Synchronous trigger generation counter expiration value (Ch *n*) When PSI5SjPSTSn.STSEL is 0, these bits are enabled. When PSI5SjPSTSn.STSEL is 1, these bits are disabled. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.8.11 PSI5SjPSTSn — PSI5S Sync Trigger Select ch n Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPSTS1: <PSI5Sj_base> + $01AC_H$
PSI5SjPSTS2: <PSI5Sj_base> + $022C_H$
PSI5SjPSTS3: <PSI5Sj_base> + $02AC_H$
PSI5SjPSTS4: <PSI5Sj_base> + $032C_H$
PSI5SjPSTS5: <PSI5Sj_base> + $03AC_H$
PSI5SjPSTS6: <PSI5Sj_base> + $042C_H$
PSI5SjPSTS7: <PSI5Sj_base> + $04AC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.71 PSI5SjPSTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STSEL	Sync trigger select Synchronous trigger select (Ch *n*) 0: Signal generated by PSI5S is selected 1: GTM output is selected, This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.9 Ch n Register/Rx (n: 1 to 7)

28.3.9.1 PSI5SjPRESn — PSI5S Receive Error Status chn Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPRES1: <PSI5Sj_base> + $01B0_H$
PSI5SjPRES2: <PSI5Sj_base> + 0230_H
PSI5SjPRES3: <PSI5Sj_base> + $02B0_H$
PSI5SjPRES4: <PSI5Sj_base> + 0330_H
PSI5SjPRES5: <PSI5Sj_base> + $03B0_H$
PSI5SjPRES6: <PSI5Sj_base> + 0430_H
PSI5SjPRES7: <PSI5Sj_base> + $04B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RERRF6	RERRF5	RERRF4	RERRF3	RERRF2	RERRF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.72 PSI5SjPRESn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5	RERRF6	Rx error at packet frame 6 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame6 of Ch *n* has occurred. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
4	RERRF5	Rx error at packet frame 5 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame5 of Ch *n* has occurred. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
3	RERRF4	Rx error at packet frame 4 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame4 of Ch *n* has occurred. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Table 28.72 PSI5SjPRESn Register Contents (2/2)

Bit Position	Bit Name	Function
2	RERRF3	Rx error at packet frame 3 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame3 of Ch *n* has occurred. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
1	RERRF2	Rx error at packet frame 2 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame2 of Ch *n* has occurred. This bit is cleared by when writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	RERRF1	Rx error at packet frame 1 (Ch *n*) 0: No error 1: An error has occurred This bit is set to 1 when any of “Rx overrun error”, “transceiver status error”, “payload data parity error”, or “payload data CRC error” in packet frame1 of Ch *n* has occurred. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Note: n: 1 to 7

28.3.9.2 PSI5SjPRESCn — PSI5S Receive Error Status Clear chn Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPRESC1: <PSI5Sj_base> + $01B4_H$
PSI5SjPRESC2: <PSI5Sj_base> + 0234_H
PSI5SjPRESC3: <PSI5Sj_base> + $02B4_H$
PSI5SjPRESC4: <PSI5Sj_base> + 0334_H
PSI5SjPRESC5: <PSI5Sj_base> + $03B4_H$
PSI5SjPRESC6: <PSI5Sj_base> + 0434_H
PSI5SjPRESC7: <PSI5Sj_base> + $04B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RERRCLF6	RERRCLF5	RERRCLF4	RERRCLF3	RERRCLF2	RERRCLF1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.73 PSI5SjPRESCn Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	RERRCLF6	Rx error clear packet Frame6 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF6 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
4	RERRCLF5	Rx error clear packet Frame5 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF5 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
3	RERRCLF4	Rx error clear packet Frame 4 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF4 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
2	RERRCLF3	Rx error clear packet Frame 3 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF3 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
1	RERRCLF2	Rx error clear packet Frame 2 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF2 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Table 28.73 PSI5SjPRESCn Register Contents (2/2)

Bit Position	Bit Name	Function
0	RERRCLF1	Rx error clear packet Frame1 (Ch *n*) 0: Ignored 1: Clear PSI5SjPRES*n*.RERRF1 This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Note: n: 1 to 7

28.3.9.3 PSI5SjPTCDTn — PSI5S Timestamp Capture Data Chn Register

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPTCDT1: <PSI5Sj_base> + $01B8_H$
PSI5SjPTCDT2: <PSI5Sj_base> + 0238_H
PSI5SjPTCDT3: <PSI5Sj_base> + $02B8_H$
PSI5SjPTCDT4: <PSI5Sj_base> + 0338_H
PSI5SjPTCDT5: <PSI5Sj_base> + $03B8_H$
PSI5SjPTCDT6: <PSI5Sj_base> + 0438_H
PSI5SjPTCDT7: <PSI5Sj_base> + $04B8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSCD[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSCD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.74 PSI5SjPTCDTn Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	TSCD[23:0]	Timestamp capture data (Ch *n*) These bits are cleared by writing 1 to PSI5SjPTCDCn.TSCCLR. These bits are cleared by writing 0 to PSI5SjPRCF1n.TSEN. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.9.4 PSI5SjPTCDCn — PSI5S Timestamp Capture Data Clear Chn Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPTCDC1: <PSI5Sj_base> + $01BC_H$
PSI5SjPTCDC2: <PSI5Sj_base> + $023C_H$
PSI5SjPTCDC3: <PSI5Sj_base> + $02BC_H$
PSI5SjPTCDC4: <PSI5Sj_base> + $033C_H$
PSI5SjPTCDC5: <PSI5Sj_base> + $03BC_H$
PSI5SjPTCDC6: <PSI5Sj_base> + $043C_H$
PSI5SjPTCDC7: <PSI5Sj_base> + $04BC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.75 PSI5SjPTCDCn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSCCLR	Timestamp capture clear (Ch *n*) 0: Ignored 1: Clear timestamp capture data (PSI5SjPTCDTn.TSCD) This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Note: n: 1 to 7

28.3.10 Ch n Register/Tx (n: 1 to 7)

28.3.10.1 PSI5SjPDDTPn — PSI5S DDSR Type chn Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPDDTP1: <PSI5Sj_base> + $01C0_H$
PSI5SjPDDTP2: <PSI5Sj_base> + 0240_H
PSI5SjPDDTP3: <PSI5Sj_base> + $02C0_H$
PSI5SjPDDTP4: <PSI5Sj_base> + 0340_H
PSI5SjPDDTP5: <PSI5Sj_base> + $03C0_H$
PSI5SjPDDTP6: <PSI5Sj_base> + 0440_H
PSI5SjPDDTP7: <PSI5Sj_base> + $04C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRTYPE[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Note: n: 1 to 7

Table 28.76 PSI5SjPDDTPn Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1 to 0	DDSRTYPE[1:0]	DDSR transmission type (Ch *n*) (n: 1 to 7) 00_B: Frame 1 (Short) 01_B: Frame 2 (Long) 10_B: Frame 3 (XLong) 11_B: Frame 4 (XXLong) Writing to these bits is prohibited when PSI5SjPDDSn.DDSRSTS is 1. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

28.3.10.2 PSI5SjPDDDn — PSI5S DDSR Data chn Register

Access: This register can be read or written in 32-bit units.

Address: PSI5SjPDDD1: <PSI5Sj_base> + $01C4_H$
PSI5SjPDDD2: <PSI5Sj_base> + 0244_H
PSI5SjPDDD3: <PSI5Sj_base> + $02C4_H$
PSI5SjPDDD4: <PSI5Sj_base> + 0344_H
PSI5SjPDDD5: <PSI5Sj_base> + $03C4_H$
PSI5SjPDDD6: <PSI5Sj_base> + 0444_H
PSI5SjPDDD7: <PSI5Sj_base> + $04C4_H$

Value after reset: $00FF\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DDSRDT[19:12]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DDSRDT[11:0]												DDSRADR[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.77 PSI5SjPDDDn Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 4	DDSRDT[19:0]	DDSR transmission data (Ch *n*) These bits cannot be written when PSI5SjPDDSn.DDSRSTS is 1. When PSI5SjPDDTPn.DDSRTYPE is 0, DDSR transmission data uses the 3 LSBs. PSI5SjPDDDn.DDSRDT [19:3] should be set to all 1's. When PSI5SjPDDTPn.DDSRTYPE is 1, DDSR transmission data uses the 13 LSBs. PSI5SjPDDDn.DDSRDT [19:13] should be set to all 1's. When PSI5SjPDDTPn.DDSRTYPE is 2, DDSR transmission data uses the 19 LSBs. PSI5SjPDDDn.DDSRDT [19] should be set to all 1's. When PSI5SjPDDTPn.DDSRTYPE is 3, DDSR transmission data uses 20 bits. These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
3 to 0	DDSRADR[3:0]	DDSR transmission address (Ch *n*) These bits cannot be written when PSI5SjPDDSn.DDSRSTS is 1.(n: 1 to 7) These bits can be written when PSI5SjPUOS.ACSTS is 0 (= configuration mode) or when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode). These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7

28.3.10.3 PSI5SjPDDSn — PSI5S DDSR Status Chn Register

Access: This register can be read-only in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPDDS1: <PSI5Sj_base> + $01C8_H$
PSI5SjPDDS2: <PSI5Sj_base> + 0248_H
PSI5SjPDDS3: <PSI5Sj_base> + $02C8_H$
PSI5SjPDDS4: <PSI5Sj_base> + 0348_H
PSI5SjPDDS5: <PSI5Sj_base> + $03C8_H$
PSI5SjPDDS6: <PSI5Sj_base> + 0448_H
PSI5SjPDDS7: <PSI5Sj_base> + $04C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.78 PSI5SjPDDSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DDSRSTS	DDSR status 0: DDSR transmission is not in progress 1: DDSR transmission is in progress In PSI5S mode, writing to PSI5SjPDDDn sets this bit to 1. When the last data of PSI5SjPDDDn is written to the Tx shifter, this bit becomes 0. PSI5SjPDDDn is cannot written when PSI5SjPDDSn.DDSRSTS is 1. This bit is cleared by writing 1 to PSI5SjPDDSPn.DDSRSTP. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Note: n: 1 to 7

28.3.10.4 PSI5SjPDDSPn — PSI5S DDSR Stop chn Register

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: PSI5SjPDDSP1: <PSI5Sj_base> + $01CC_H$
PSI5SjPDDSP2: <PSI5Sj_base> + $024C_H$
PSI5SjPDDSP3: <PSI5Sj_base> + $02CC_H$
PSI5SjPDDSP4: <PSI5Sj_base> + $034C_H$
PSI5SjPDDSP5: <PSI5Sj_base> + $03CC_H$
PSI5SjPDDSP6: <PSI5Sj_base> + $044C_H$
PSI5SjPDDSP7: <PSI5Sj_base> + $04CC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSRSTP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 28.79 PSI5SjPDDSPn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DDSRSTP	DDSR Tx stop (Ch *n*) 0: Ignored 1: Stop transmission When this bit is written, PSI5SjPDDSn.DDSRSTS is reset to 0. This bit is always read as 0.(n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Note: n: 1 to 7

28.3.11 Ch n Register/Interrupt (n: 1 to 7)

28.3.11.1 PSI5SjPCISn — PSI5S CPU Interrupt Status chn Register

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: PSI5SjPCIS1: <PSI5Sj_base> + $01D0_H$
PSI5SjPCIS2: <PSI5Sj_base> + 0250_H
PSI5SjPCIS3: <PSI5Sj_base> + $02D0_H$
PSI5SjPCIS4: <PSI5Sj_base> + 0350_H
PSI5SjPCIS5: <PSI5Sj_base> + $03D0_H$
PSI5SjPCIS6: <PSI5Sj_base> + 0450_H
PSI5SjPCIS7: <PSI5Sj_base> + $04D0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ISTDDSFN	ISTDDSOW	ISTRFN	ISTRFEX	ISTRFLK	ISTROV	ISTRWDT	—	—	—	ISTTRST	ISTPT	ISTCRC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.80 PSI5SjPCISn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	ISTDDSFN	CPU interrupt status of DDSR finish (Ch *n*) 0: DDSR transmission has not finished 1: DDSR transmission has finished When PSI5SjPDDDn data is written to the Tx shifter, this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_tx) occur. (n = 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCDDSFN. When PSI5SjPCISn.ISTDDSFN is set and 1 is written to PSI5SjPCISCn.ISTCDDSFN in the same clock cycle, PSI5SjPCISn.ISTDDSFN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
12	ISTDDSOW	CPU interrupt status of DDSR overwrite error (Ch *n*) 0: No error 1: Error detected When the CPU writes 1 to PSI5SjPDDD*n* in PSI5SjPDDS*n*.DDSRSTS, this bit is set to 1 and an interrupt (int_psis_ch*n*) occurs.(*n*: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISC*n*.ISTCDDSOW. When PSI5SjPCIS*n*.ISTDDSOW is set and 1 is written to PSI5SjPCISC*n*.ISTCDDSOW in the same clock cycle, PSI5SjPCIS*n*.ISTDDSOW remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).

Table 28.80 PSI5SjPCISn Register Contents (2/3)

Bit Position	Bit Name	Function
11	ISTRFN	Interrupt status of Rx finish (Ch *n*) 0: PSI5S frame is not received successfully 1: PSI5S frame is received successfully When a PSI5S frame is stored in a MB and the PSI5S frame has no errors (no mailbox overrun error), this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_rx) occur. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCRFN. When PSI5SjPCISn.ISTRFN is set and 1 is written to PSI5SjPCISCn.ISTCRFN in the same clock cycle, PSI5SjPCISn.ISTRFN remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
10	ISTRFEX	Interrupt status of Rx frame excess error (Ch0) 0: No error 1: Error detected When a packet over PSI5SjPRCF1n.FmPKT (m = 1 to 6) is received, this bit is set to 1 and an interrupt (int_psis_chn) occurs. This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCRFEX. When PSI5SjPCISn.ISTRFEX is set and 1 is written to PSI5SjPCISCn.ISTCRFEX in the same clock cycle, PSI5SjPCISn.ISTRFEX remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
9	ISTRFLK	Interrupt status of Rx frame lack error (Ch0) 0: No error 1: Error detected When a packet frame gap is detected while the packet is under PSI5SjPRCF1n.FmPKT (m = 1 to 6), this bit is set to 1 and an interrupt (int_psis_chn) occurs. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCRFLK. When PSI5SjPCISn.ISTRFLK is set and 1 is written to PSI5SjPCISCn.ISTCRFLK in the same clock cycle, PSI5SjPCISn.ISTRFLK remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
8	ISTROV	Interrupt status of Rx overrun error (Ch *n*) 0: No error 1: Error detected When an attempt is made to store the next PSI5S frame in a MB before the CPU has read that MB, this bit is set to 1 and an interrupt (int_psis_ch*n*) occurs. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCROV. When PSI5SjPCISn.ISTROV is set and 1 is written to PSI5SjPCISCn.ISTCROV in the same clock cycle, PSI5SjPCISn.ISTROV remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
7	ISTRWDT	Interrupt status of Rx WDT error (Ch *n*) 0: No error 1: Error detected When an WDT error occurs in PSI5S mode, this bit is set to 1 and an interrupt (int_psis_chn) and DMA request (dma_psis_chn_rx) occur. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCRWDT. When PSI5SjPCISn.ISTRWDT is set and 1 is written to PSI5SjPCISCn.ISTCRWDT in the same clock cycle, PSI5SjPCISn.ISTRWDT remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
6 to 4	Reserved	When read, the value after reset is returned.

Table 28.80 PSI5SjPCISn Register Contents (3/3)

Bit Position	Bit Name	Function
3	ISTTRST	Interrupt status of Rx transceiver status error (Ch *n*) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a transceiver status error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCTRST. When PSI5SjPCISn.ISTTRST is set and 1 is written to PSI5SjPCISCn.ISTCTRST in the same clock cycle, PSI5SjPCISn.ISTTRST remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
2	ISTPT	Interrupt status of payload data parity error (Ch *n*) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a payload data parity error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCPT. When PSI5SjPCISn.ISTPT is set and 1 is written to PSI5SjPCISCn.ISTCPT in the same clock cycle, PSI5SjPCISn.ISTPT remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
1	ISTCRC	Interrupt status of payload data CRC error (Ch *n*) 0: No error 1: Error detected When a PSI5S frame is stored in a MB and a payload data CRC error occurs, this bit is set to 1 and an interrupt (int_psis_chn) occurs. (n: 1 to 7) This bit is cleared by writing 1 to PSI5SjPCISCn.ISTCCRC. When PSI5SjPCISn.ISTCRC is set and 1 is written to PSI5SjPCISCn.ISTCCRC in the same clock cycle, PSI5SjPCISn.ISTCRC remains 1. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. This bit is cleared when PSI5SjPUOS.ACSTS is changed to 0 (= configuration mode).
0	Reserved	When read, the value after reset is returned*1.

Note: n: 1 to 7

Note 1. b[5][4][0]: These bits are not reserved in Ch0. If these errors (UART framing error, UART parity error, XCRC Error) occur, the error is stored in channel 0, frame 2".

28.3.11.2 PSI5SjPCISCn — PSI5S CPU Interrupt Status Clear chn Register

Access: This register can be read or written in 32-bit or 16-bit units.

Address: PSI5SjPCISC1: <PSI5Sj_base> + $01D4_H$
PSI5SjPCISC2: <PSI5Sj_base> + 0254_H
PSI5SjPCISC3: <PSI5Sj_base> + $02D4_H$
PSI5SjPCISC4: <PSI5Sj_base> + 0354_H
PSI5SjPCISC5: <PSI5Sj_base> + $03D4_H$
PSI5SjPCISC6: <PSI5Sj_base> + 0454_H
PSI5SjPCISC7: <PSI5Sj_base> + $04D4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ISTCDDSFN	ISTCDDSOW	ISTCRFN	ISTCRFEX	ISTCRFLK	ISTCROV	ISTCRWDT	—	—	—	ISTCTRST	ISTCPT	ISTCCRC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R

Table 28.81 PSI5SjPCISCn Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	ISTCDDSFN	Interrupt status clear DDSR Tx finish (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTDDSFN This bit defines about clearance of CPU interrupt status of DDSR finish (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTDDSFN). When this bit is written to 1, PSI5SjPCIS0.ISTDDSFN is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
12	ISTCDDSOW	Interrupt status clear DDSr overwrite error Clear at CPU interrupt status of DDSR overwrite error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTDDSOW This bit defines about clearance of CPU interrupt status of DDSR overwrite (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTDDSOW). When this bit is written to 1, PSI5SjPCIS0.ISTDDSOW is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
11	ISTCRFN	Interrupt status clear Rx finish Clear at CPU interrupt status of Rx finish (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTRFN This bit defines about clearance of CPU interrupt status of Rx finish (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTRFN). When this bit is written to 1, PSI5SjPCISn.ISTRFN is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Table 28.81 PSI5SjPCISCn Register Contents (2/3)

Bit Position	Bit Name	Function
10	ISTCRFEX	Interrupt status clear Rx frame excess error Clear at CPU interrupt status of Rx frame excess error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTRFEX This bit defines about clearance of CPU interrupt status of Rx frame (=packet) (Ch *n*) excess error in PSI5S mode (PSI5SjPCIS*n*.ISTRFEX). When this bit is written to 1, PSI5SjPCIS*n*.ISTRFEX is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
9	ISTCRFLK	Interrupt status clear Rx frame lack error Clear at CPU interrupt status of Rx frame lack error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTRFLK This bit defines about clearance of CPU interrupt status of Rx frame (packet) (Ch *n*) lack error (Ch*n*) in PSI5S mode (PSI5SjPCIS*n*.ISTRFLK). When this bit is written to 1, PSI5SjPCISn.ISTRFLK is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
8	ISTCROV	Interrupt status clear Rx overrun error Clears at CPU interrupt status of Rx overrun error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTROV This bit defines about clearance of CPU interrupt status of overrun error (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTROV). When this bit is written to 1, PSI5SjPCISn.ISTROV is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
7	ISTCRWDT	Interrupt status clear Rx WDT error Clear at CPU interrupt status of Rx WDT error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTRWDT This bit defines about clearance of CPU interrupt status of WDT error (Ch*n*) in PSI5S mode (PSI5SjPCIS*n*.ISTRWDT). When this bit is written to 1, PSI5SjPCISn.ISTRWDT is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ISTCTRST	Interrupt status clear transceiver status error Clear at CPU interrupt status of Rx transceiver status error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTTRST This bit defines about clearance of CPU interrupt status of transceiver status error (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTTRST). When this bit is written to 1, PSI5SjPCISn.ISTTRST is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).

Table 28.81 PSI5SjPCISCn Register Contents (3/3)

Bit Position	Bit Name	Function
2	ISTCPT	Interrupt status clear parity error Clear at CPU interrupt status of payload data parity error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTPT This bit defines about clearance of CPU interrupt status of payload data parity error (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTPT). When this bit is written to 1, PSI5SjPCISn.ISTPT is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
1	ISTCCRC	Interrupt status clear CRC error Clear at CPU interrupt status of payload data CRC error (Ch*n*) 0: Ignored 1: Clear PSI5SjPCIS*n*.ISTCRC This bit defines about clearance of CPU interrupt status of payload data CRC error (Ch *n*) in PSI5S mode (PSI5SjPCIS*n*.ISTCRC). When this bit is written to 1, PSI5SjPCISn.ISTCRC is cleared to 0. This bit is always read as 0. (n: 1 to 7) This bit can be written when PSI5SjPUOS.ACSTS is 1 and PSI5SjPUOS.MSTS is 1 (= PSI5S mode).
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note: n: 1 to 7

28.3.12 Ch 0 Frm m MB Data (m: 1, 2)

28.3.12.1 PSI5SjPMB0mS — PSI5S Receive MailBox ch0 Frmm Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: PSI5SjPMB01S: <PSI5Sj_base> + 0500_H
PSI5SjPMB02S: <PSI5Sj_base> + $050C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	CHID[2:0]			FID[2:0]			MBORERR	WDTERR	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UTFRERR	UTPTERR	HEADERR	HEADST[1:0]		CRCERR	CRC[2:0]			XCRCERR	XCRC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.82 PSI5SjPMB0mS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch0, Frm *m*) The DCI value is generated by a 4-bit counter, and every time the PSI5 frame data is restored, it is incremented by 1. These bits are read only. The write value is ignored. (m: 1, 2) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 25	Reserved	When read, the value after reset is returned.
24 to 22	CHID[2:0]	Rx channel ID (Ch0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
21 to 19	FID[2:0]	Rx frame ID (Ch0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
18	MBORERR	MailBox overrun error (Ch0, Frm *m*) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
17	WDTERR	Rx frame WDT error (Ch0, Frm *m*) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
16	Reserved	When read, the value after reset is returned.
15	UTFRERR	UART framing error (Ch0, Frm2)*1 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2. *1 This bit is only in 50CH (Frm2)

Table 28.82 PSI5SjPMB0mS Register Contents (2/2)

Bit Position	Bit Name	Function
14	UTPTERR	UART parity error (Ch0, Frm2) *1 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2. *1 This bit is only in $50C_H$ (Frm2)
13	HEADERR	Header error (Ch0, Frm *m*) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
12 to 11	HEADST[1:0]	Header status These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
10	CRCERR	Rx CRC/Parity error (Ch0, Frm *m*) 0: No error 1: Error detected When PSI5SjPRCF10.RFCPS is 1, this bit shows a CRC error. When PSI5SjPRCF10.RFCPS is 0, this bit shows a parity error. This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
9 to 7	CRC[2:0]	Rx frame CRC/Parity (Ch0, Frm *m*)*2 When PSI5SjPRCF10.RFCPS is 1, these bits show the CRC (3 bits). When PSI5SjPRCF10.RFCPS is 0, bit [9] shows the parity, and bits [8:7] are reserved. These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
6	XCRCERR	Rx XCRC error (Ch0, Frm2) *1 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST. When this error occurs, data is stored in channel 0, frame 2. So, this error exists only in channel 0, frame 2. *1 This bit is only in 50CH (Frm2)
5 to 0	XCRC[5:0]	Rx frame XCRC (Ch0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: m: 1, 2
Frm, frame: packet frame

Note 1. Bits [17], [15], [14] and [6] are used only in $50C_H$ (Frm2)

Note 2. When PSI5SjPRCF10.RFCPS is 1, bits [9:7] are the CRC, and when PSI5SjPRCF10.RFCPS is 0, bit [9] is the parity, bits [8:7] are reserved.

28.3.12.2 PSI5SjPMB0mD — PSI5S Receive MailBox ch0 Frmm Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: PSI5SjPMB01D: <PSI5Sj_base> + 0504_H
PSI5SjPMB02D: <PSI5Sj_base> + 0510_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				DATA[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.83 PSI5SjPMB0mD Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 0	DATA[27:0]	Message data (Ch0, Frm *m*) When the number of payloads (PSI5SjPRCF20.FmPAYLD) is less than 28, the module stores the payload from the LSB and the MSB is filled with 0. These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: m: 1, 2
Frm, frame: packet frame

28.3.12.3 PSI5SjPMB0mT — PSI5S Receive MailBox ch0 Frmm Timestamp Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: PSI5SjPMB01T: <PSI5Sj_base> + 0508_H
PSI5SjPMB02T: <PSI5Sj_base> + 0514_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	—	TMST[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMST[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.84 PSI5SjPMB0mT Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (CH0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 24	Reserved	When read, the value after reset is returned.
23 to 0	TMST[23:0]	Timestamp data (CH0, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: m: 1, 2
Frm, frame: packet frame

28.3.13 Ch n Frm m MB Data (n: 1 to 7) (m: 1 to 6)

28.3.13.1 PSI5SjPMBnmS — PSI5S Receive MailBox ch n Frm m Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 28.12**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	CHID[2:0]			FID[2:0]			MBORERR	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	HEADERR	HEADST[1:0]		CRCERR	CRC[2:0]			—	XCRC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.85 PSI5SjPMBnmS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch*n*, Frm *m*) The DCI value is generated by a 4-bit counter, and every time the PSI5 frame data is restored, the count is incremented by 1. These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 25	Reserved	When read, the value after reset is returned.
24 to 22	CHID[2:0]	Rx channel ID (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
21 to 19	FID[2:0]	Rx frame ID (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
18	MBORERR	Mailbox overrun error (Ch*n*, Frm *m*) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
17 to 14	Reserved	When read, the value after reset is returned.
13	HEADERR	Header error (Ch*n*, Frm *m*) 0: No error 1: Error detected This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.
12 to 11	HEADST[1:0]	Header status (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
10	CRCERR	Rx CRC/Parity error (Ch*n*, Frm *m*) 0: No error 1: Error detected When PSI5SjPRCF1*m*.RFCPS is 1, this bit shows a CRC error. And when PSI5SjPRCF1*m*.RFCPS is 0, this bit shows a parity error. This bit is cleared by writing 1 to PSI5SjPRMBC.MBCLR. This bit is cleared by writing 1 to PSI5SjPUSWR.SWRST.

Table 28.85 PSI5SjPMBnmS Register Contents (2/2)

Bit Position	Bit Name	Function
9 to 7	CRC[2:0]	Rx frame CRC/Parity (Ch*n*, Frm *m*) *1 These bits show the Rx frame CRC/Parity. (Ch*n*, Frm *m*) When PSI5SjPRCF1*m*.RFCPS is 1, these bits show the CRC (3 bits). When PSI5SjPRCF1*m*.RFCPS is 0, bit [9] shows the parity, and bits [8:7] are reserved. These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
6	Reserved	When read, the value after reset is returned.
5 to 0	XCRC[5:0]	Rx frame XCRC (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7
m: 1 to 6
Frm, frame: packet frame
Bits [15],[14] and [6]: These bits are not reserved in channel 0. When these errors (UART framing error, UART parity error, XCRC Error) occur, the error is stored in channel 0, frame 2.

Note 1. When PSI5SjPRCF1n.RFCPS is 1 bits [9:7] are the CRC, and when PSI5SjPRCF1n.RFCPS is 0 bit [9] is the parity, and bits [8:7] are reserved.

28.3.13.2 PSI5SjPMBnmD — PSI5S Receive MailBox ch0 Frmm Data Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 28.12**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				DATA[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.86 PSI5SjPMBnmD Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 0	DATA[27:0]	Rx message data (Ch*n*, Frm *m*) When the number of payloads (PSI5SjPRCF2n.FmPAYLD) is less than 28, the PSI5-S stores the payload from the LSB and the MSB is filled with 0. These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7
m: 1 to 6
Frm, frame: packet frame

28.3.13.3 PSI5SjPMBnmT — PSI5S Receive MailBox chn Frmm Timestamp Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 28.12**

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCI[3:0]				—	—	—	—	TMST[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMST[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 28.87 PSI5SjPMBnmT Register Contents

Bit Position	Bit Name	Function
31 to 28	DCI[3:0]	DCI value (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.
27 to 24	Reserved	When read, the value after reset is returned.
23 to 0	TMST[23:0]	Timestamp data (Ch*n*, Frm *m*) These bits are cleared by writing 1 to PSI5SjPRMBC.MBCLR. These bits are cleared by writing 1 to PSI5SjPUSWR.SWRST.

Note: n: 1 to 7
m: 1 to 6
Frm, frame: packet frame

28.4 Operation Modes

PSI5S has three operation modes: Configuration, PSI5S, and UART modes.

The operation mode state machine controls the transition of PSI5S between these operation modes.

Figure 28.5 shows the state transitions controlled by the operation mode state machine.

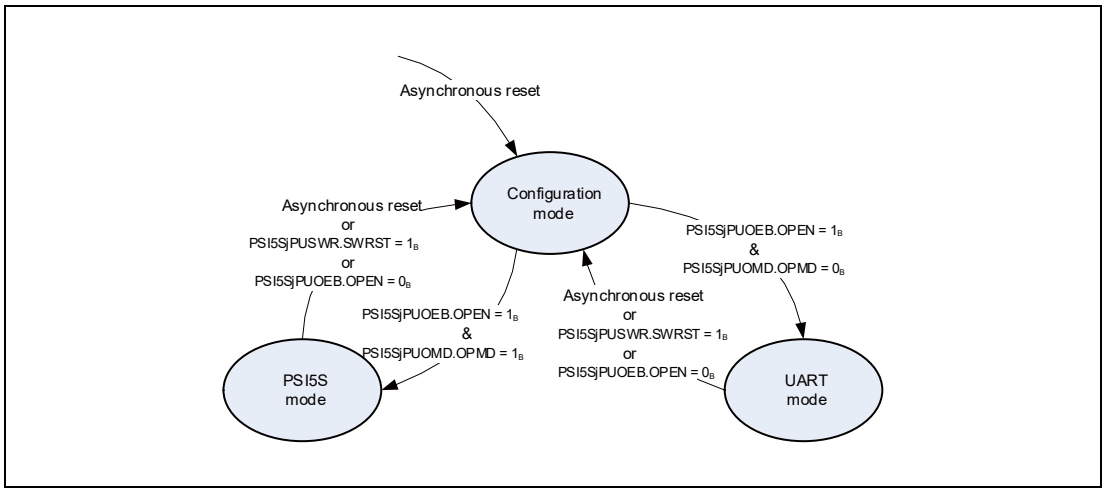


Figure 28.5 State Transitions by Operation Mode State Machine

Table 28.88 below describes the individual states:

Table 28.88 Operation Modes (States)

No	Operation Mode	Explanation
1	Configuration mode	Configuration operation mode • Setting SFR of PSI5S.
2	PSI5S mode	PSI5S operation mode • Communication of a PSI-5 frame is performed.
3	UART mode	UART operation mode • Communication of UART packet frame.

When the PSI5S is not transmitting after 16 PCLK cycles of setting of returning to the configuration mode from PSI5S mode or UART mode, the PSI5S returns to the configuration mode.

When the PSI5S is transmitting after 16 PCLK cycles of setting of returning to the configuration mode, the PSI5S returns to the configuration mode in the following timing:

In UART mode: The PSI5S ends transmission operation of the 2 UART frames at the maximum.

In PSI5S mode: The PSI5S ends transmission operation of the 9 UART frames at the maximum.

NOTE

Even during reception, the PSI5S stops reception operation and returns to the reception standby state.

Interrupt and DMA request when the PSI5S return to the configuration mode is following:

Interrupt: PSI5S does not generate interrupts after 14 or more PCLK cycles of returning to the configuration mode. Note that, because interrupt status flags are cleared when the PSI5S module returns to the configuration mode, interrupt status cannot be read if the PSI5S module returns to the configuration mode after it has issued an interrupt.

DMA request: PSI5S module does not issue DMA requests after 14 or more PCLK cycles of returning to the configuration mode.

28.5 Setting Procedures

Figure 28.6 shows the setting procedure to use PSI5S in PSI5S mode.

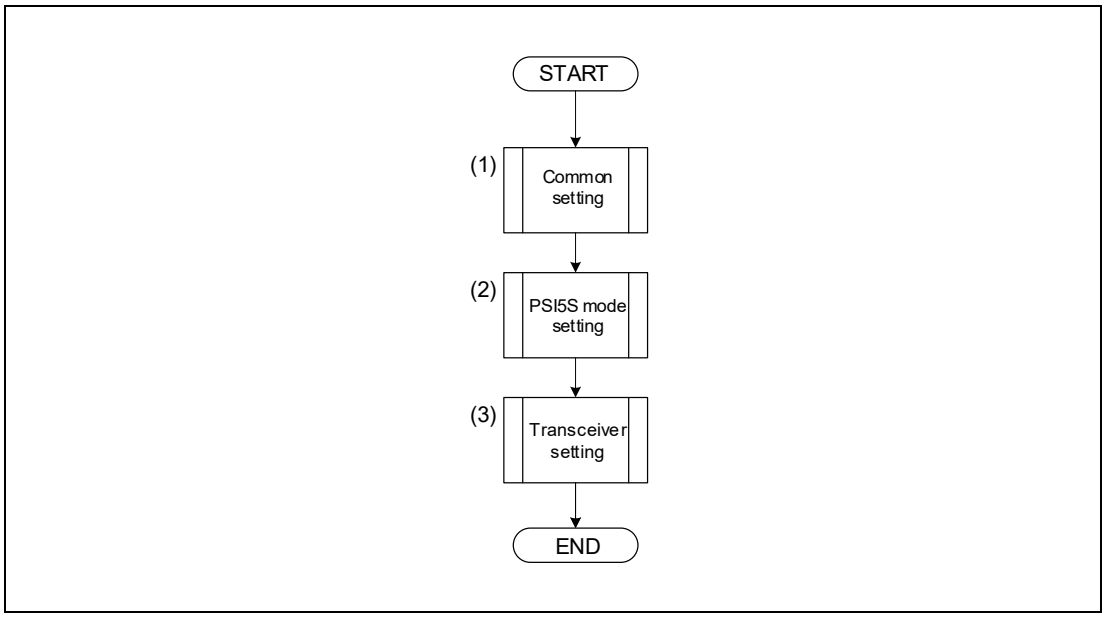


Figure 28.6 PSI5S mode Setting Procedure

(1) Specify the common settings. For details, see **Section 28.5.1, Common Setting Procedure**.

(2) Set up PSI5S mode. For details, see **Section 28.5.2, PSI5S mode Setting Procedure**.

(3) Set up the transceiver. For details, see **Section 28.5.4, Procedure for Transmitting Transceiver Commands**.

Figure 28.7 shows the setting procedure to use the PSI5S in UART mode.

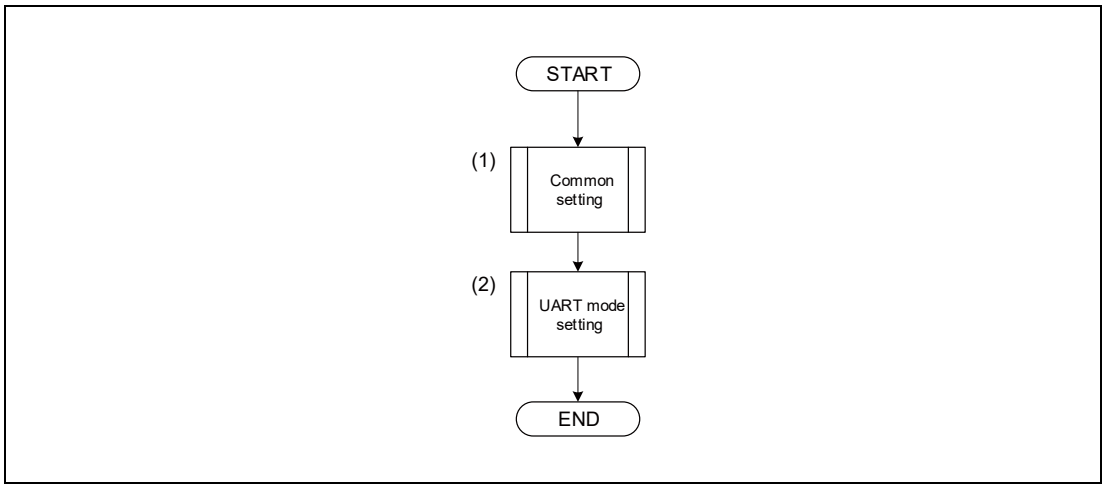


Figure 28.7 UART Mode Setting Procedure

(1) Specify the common settings. For details, see **Section 28.5.1, Common Setting Procedure**.

(2) Set up the UART mode. For details, see **Section 28.5.3, UART Mode Setting Procedure**.

28.5.1 Common Setting Procedure

This section describes the common setting procedure to use the PSI5S in PSI5S or UART mode.

Perform the procedure shown in **Figure 28.8** below when the PSI5S is in configuration mode.

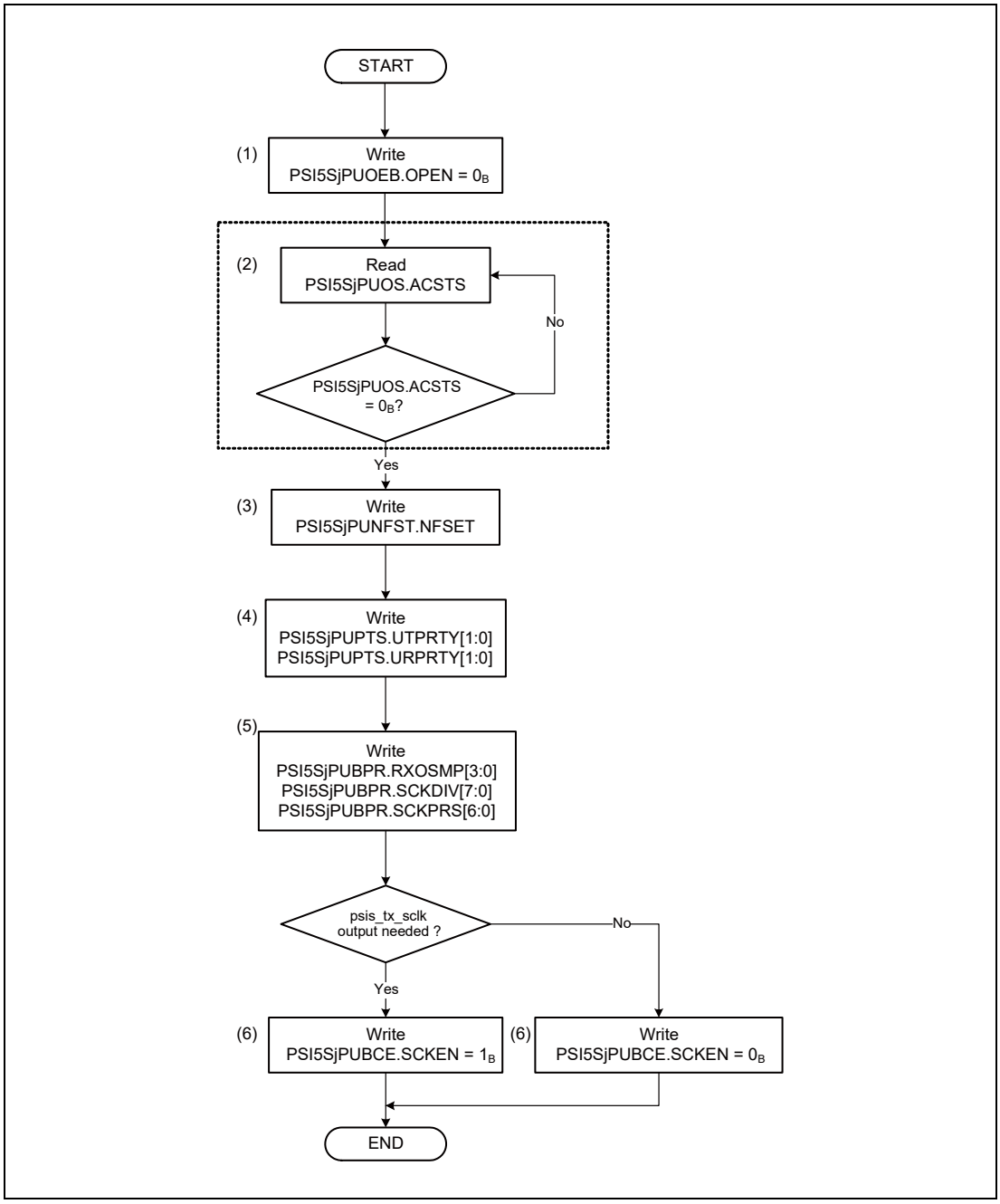


Figure 28.8 Common Setting Procedure

(1) PSI5SjPUOEB: PSI5S/UART Operation Enable register
Change the operation mode to configuration mode.

(2) PSI5SjPUOS: PSI5S /UART Operation Status register
Wait until PSI5S transitions to the configuration mode.

(3) PSI5SjPUNFST: PSI5S/UART Noise Filter Set register
Specify whether to use a noise filter for the signal input to the psis_rx_data pin, which is a UART Rx pin.

(4) PSI5SjPUPTS: PSI5S/UART rx/tx Parity Set register
Specify the parity bit settings for both UART reception and UART transmission.

(5) PSI5SjPUBPR: PSI5S/UART Baud rate Parameter register
Specify the baud rate for UART communication. The baud rate is determined by the oversample number, clock division ratio, and division ratio of the prescaler.
The clock division ratio and the division ratio of the prescaler are also used to set the frequency of the sampling clock (psis_tx_sclk).

Calculate the baud rate of UART communication using the following formula:

Baud rate = [psis_clk frequency] * (1/prescaler division ratio) * (1/clock division ratio) * (1/oversample number)

Example: Prescaler division ratio = 1 (PCKPRS setting = 0)

Clock division ratio = 3 (SCKDIV setting = 2)

Oversample number = 5 (RXOSMP setting = 4)

The calculation result is as follows:

Baud rate = 80MHz * (1/1) * (1/3) * (1/5) = 5.33 Mbps

Calculate the frequency of the sampling clock using the following formula:

psis_tx_sclk = [psis_mult_clk frequency] * (1/2) * (1/prescaler division ratio) * (1/clock division ratio)

Example: Prescaler division ratio = 1 (PCKPRS setting = 0)

Clock division ratio = 3 (SCKDIV setting = 2)

The calculation result is as follows:

psis_tx_sclk = 160MHz * (1/2) * (1/1) * (1/3) = 26.67 MHz

If the output of the psis_tx_sclk signal is required, specify the following setting:

(6) PSI5SjPUBCE: PSI5S/UART Baud rate Clock Enable register
Specify whether to enable the output of psis_tx_sclk clock.

28.5.2 PSI5S mode Setting Procedure

This section describes the setting procedure to use PSI5S in PSI5S mode.

Perform the procedure shown in **Figure 28.9** below after performing the procedure described in **Section 28.5.1, Common Setting Procedure**.

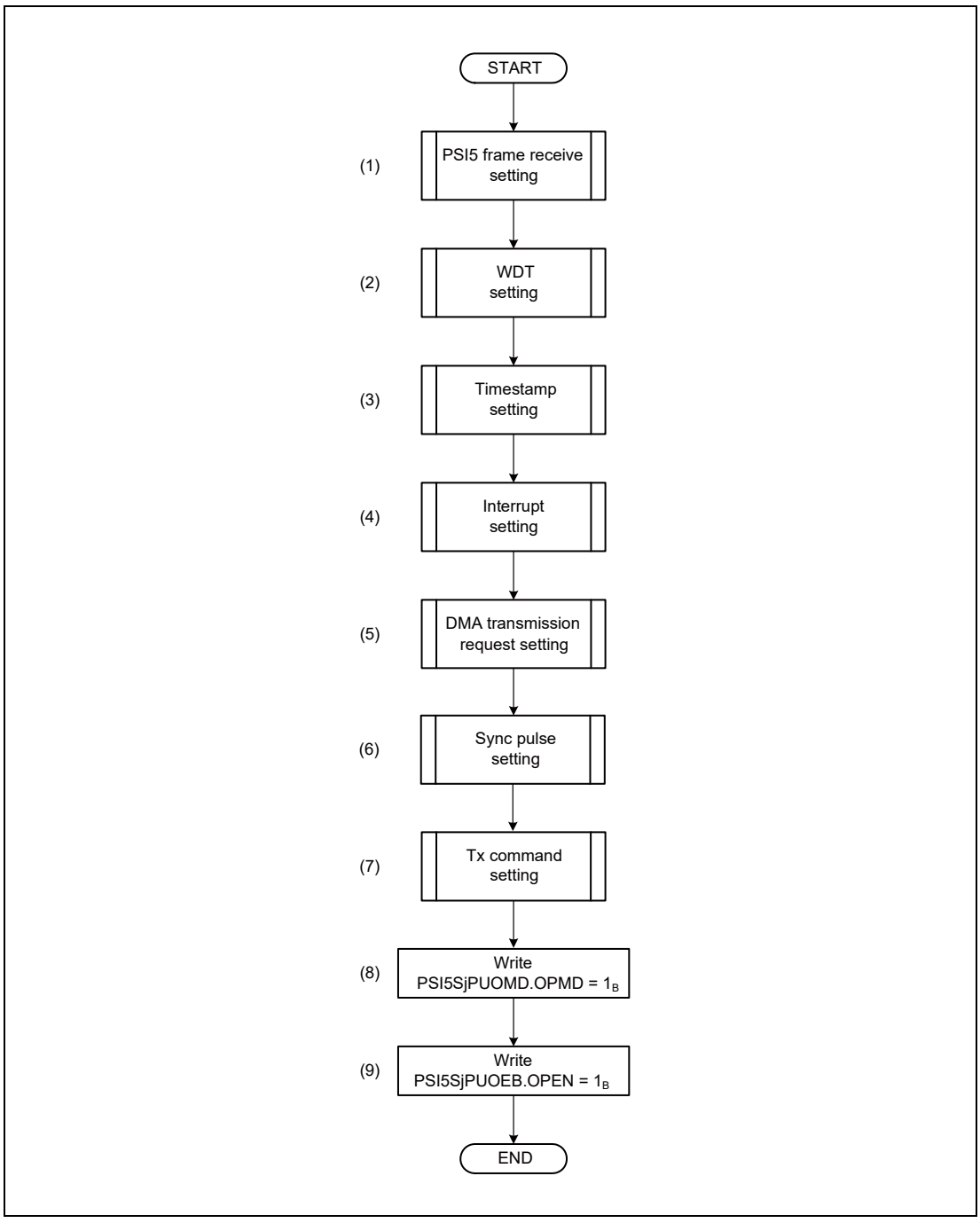


Figure 28.9 PSI5S mode Setting Procedure

(1) Set up PSI5 frame reception. For details, see **Section 28.5.2.1, PSI5 Frame Reception Setting**.

(2) Set up the WDT. For details, see **Section 28.5.2.2, WDT Setting**.

(3) Set up timestamps. For details, see **Section 28.5.2.3, Timestamp Setting**.

(4) Set up interrupts. For details, see **Section 28.5.2.4, Interrupt Setting**.

(5) Set up DMA transfer requests. For details, see **Section 28.5.2.5, DMA Transfer Request Setting**.

(6) Setup synchronization pulse. For details, see **Section 28.5.2.6, Synchronization Pulse Setting Procedure**.

(7) Set up Tx command. For details, see **Section 28.5.2.7, Tx Command Setting Procedure**.

(8) PSI5SjPUOMD: PSI5S/UART Operation Mode register
Set the operation mode to PSI5S mode.

(9) PSI5SjPUOEB: PSI5S/UART Operation Enable register
Start the operation of PSI5S.

28.5.2.1 PSI5 Frame Reception Setting

This section describes how to set up PSI5 frame reception.

Perform the procedure shown in **Figure 28.10** below for channel 0 and the required channels among channels 1 to 7.

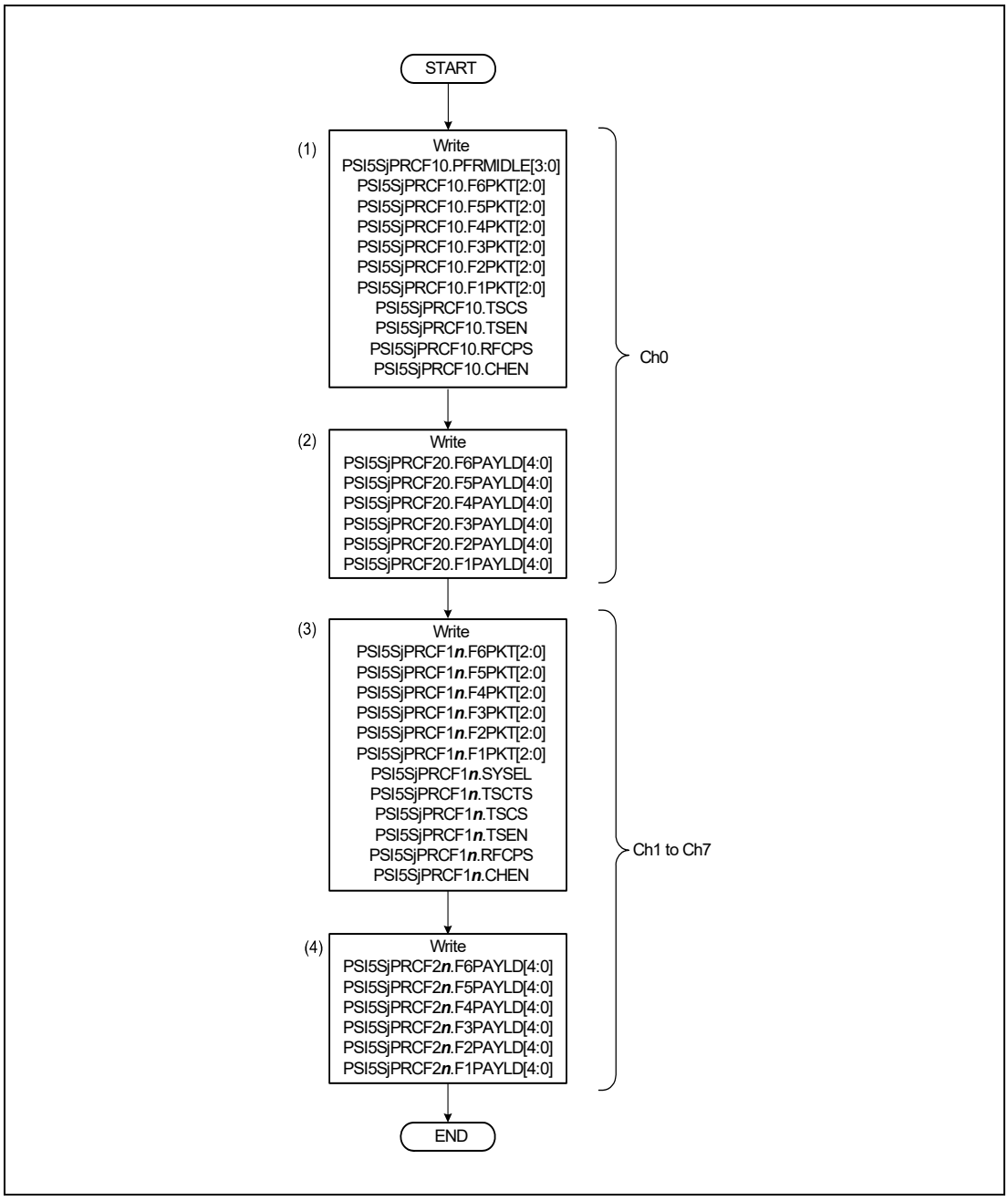


Figure 28.10 PSI5 Frame Reception Setting Procedure

(1) PSI5SjPRCF10: PSI5S Receive Config1 ch0 register
Specify the desired settings in the PSI5 frame reception configuration 1 register for channel 0.

(2) PSI5SjPRCF20: PSI5S Receive Config2 ch0 register
Specify the desired settings in the PSI5 frame reception configuration 2 register for channel 0.

(3) PSI5SjPRCF1*n*: PSI5S Receive Config1 ch***n*** register (*n* = 1 to 7)
Specify the desired settings in the PSI5 frame reception configuration 1 register for channel *n*.

(4) PSI5SjPRCF2*n*: PSI5S Receive Config2 ch***n*** register (*n* = 1 to 7)
Specify the desired settings in the PSI5 frame reception configuration 2 register for channel *n*.

The value of the PSI5SjPRCF1*n*.F*m*PKT[2:0] bits is determined by the relationship between the values of the PSI5SjPRCF1*n*.RFCPS and PSI5SjPRCF2*n*.F*m*PAYLD[4:0] bits as shown in **Table 28.89**. Note that if a value other than those shown in the table is set to the PSI5SjPRCF1*n*.F*m*PKT[2:0] bits, the operation of PSI5S will be unpredictable. (*n* = 1 to 7, *m* = 1 to 6)

Table 28.89 List of PSI5SjPRCF1*n*.F*m*PKT[2:0] Settings

	PSI5SjPRCF1*n*.F*m*PKT[2:0]	
PSI5SjPRCF2*n*.F*m*PAYLD[4:0]	PSI5SjPRCF1*n*. RFCPS = 0 (Checksum of Rx frame is parity)	PSI5SjPRCF1*n*. RFCPS = 1 (Checksum of Rx frame is CRC)
8	3_H	4_H
9	3_H	4_H
10	4_H	4_H
11	4_H	4_H
12	4_H	4_H
13	4_H	4_H
14	4_H	4_H
15	4_H	4_H
16	4_H	5_H
17	4_H	5_H
18	5_H	5_H
19	5_H	5_H
20	5_H	5_H
21	5_H	5_H
22	5_H	5_H
23	5_H	5_H
24	5_H	6_H
25	5_H	6_H
26	6_H	6_H
27	6_H	6_H
28	6_H	6_H

Note: n = 1 to 7, m = 1 to 6

(1) PSI5 Frame Reception Setting (no payload checksum in reception frame)

IF there is no Payload Checksum in a reply frame from a transceiver (excluding the Payload bit number of 10, 18 and 24 bits), the Rx Frame checksum CRC/Parity Select bit (PSI5SjPRCF1***n***.RFCPS) is set as Parity (= 0) and the payload parity error is ignored. PSI5S recognizes the staffing bit as a parity bit.

For the procedure of the frame reception from a transceiver, see **Section 28.5.6.3, Response Frame from Transceiver Reception**.

28.5.2.2 WDT Setting

This section describes how to set up the WDT.

Perform the procedure shown in **Figure 28.11** below for the channels that require the WDT.

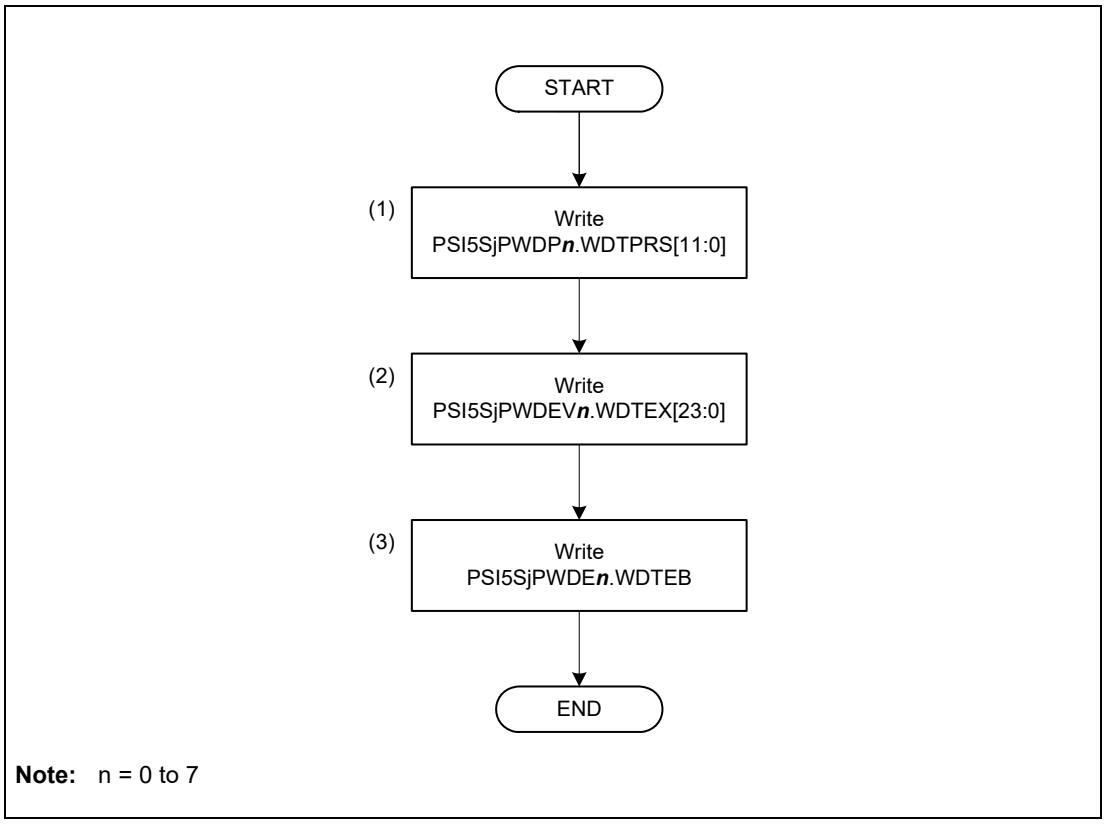


Figure 28.11 WDT Setting Procedure

(1) PSI5SjPWDP*n*: PSI5S WDT Prescaler ch***n*** register
Specify the WDT prescaler settings for channel *n*. (*n*: 0 to 7)

(2) PSI5SjPWDEV*n*: PSI5S WDT Expiration Value ch***n*** register
Specify the expiration value of the WDT for channel *n*. (*n*: 0 to 7)

(3) PSI5SjPWDE*n*: PSI5S WDT Enable ch***n*** register
Enable the WDT for channel *n*. (*n*: 0 to 7)
The WDT can be enabled when PSI5S is in either the PSI5S_active or configuration state.

28.5.2.3 Timestamp Setting

This section describes how to set up timestamp counters A and B.

Perform the procedure shown in **Figure 28.12** below for the timestamps to be used. **Figure 28.12** shows the setting procedure to control timestamp counters A and B separately.

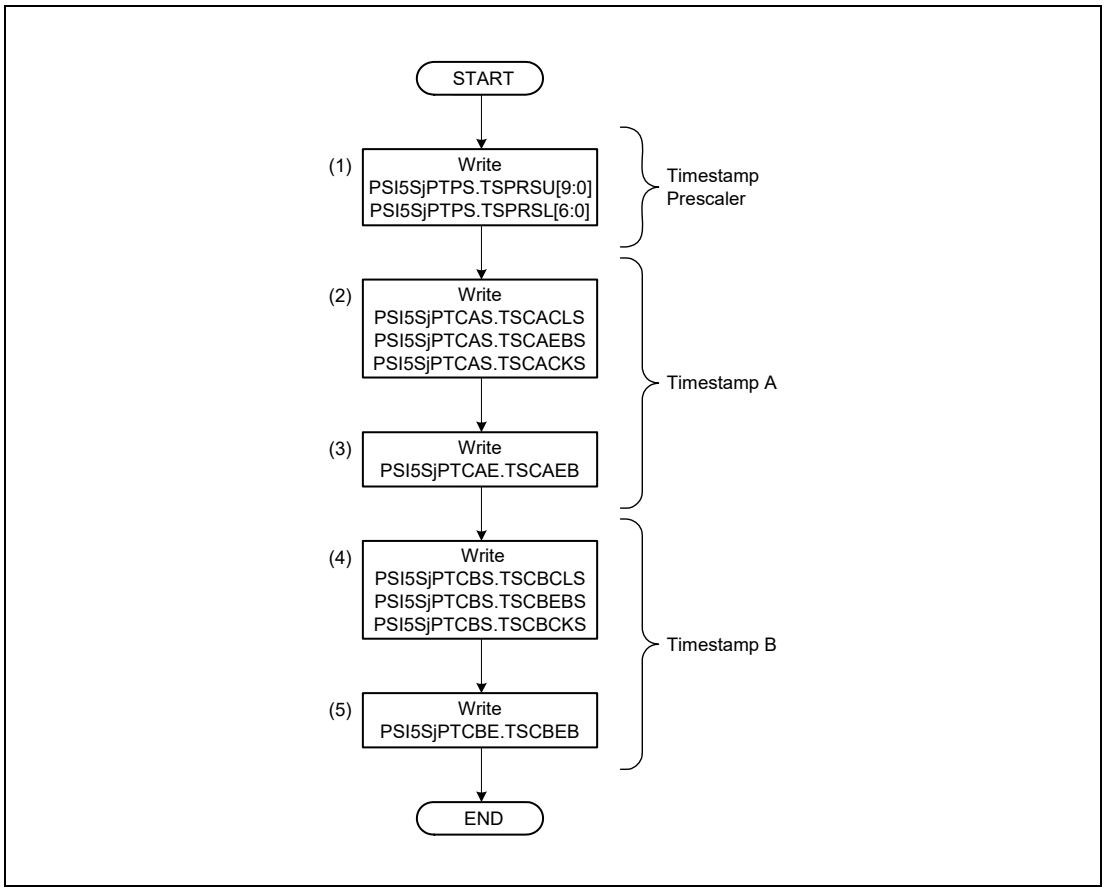


Figure 28.12 Timestamp Setting Procedure (for Separate Control)

(1) PSI5SjPTPS: PSI5S Timestamp Prescaler register
Specify the settings of the timestamp prescaler.

(2) PSI5SjPTCAS: PSI5S Timestamp Counter A Select register
Select the clear signal, enable signal, and clock signal for timestamp counter A.

(3) PSI5SjPTCAE: PSI5S Timestamp Counter A Enable register
Enable the operation of timestamp counter A. By this setting, timestamp counter A starts.

(4) PSI5SjPTCBS: PSI5S Timestamp Counter B Select register
Select the clear signal, enable signal, and clock signal for timestamp counter B.

(5) PSI5SjPTCBE: PSI5S Timestamp Counter B Enable register
Enable the operation of timestamp counter B.

Figure 28.13 below shows a different setting procedure than the above to control timestamp counters A and B together.

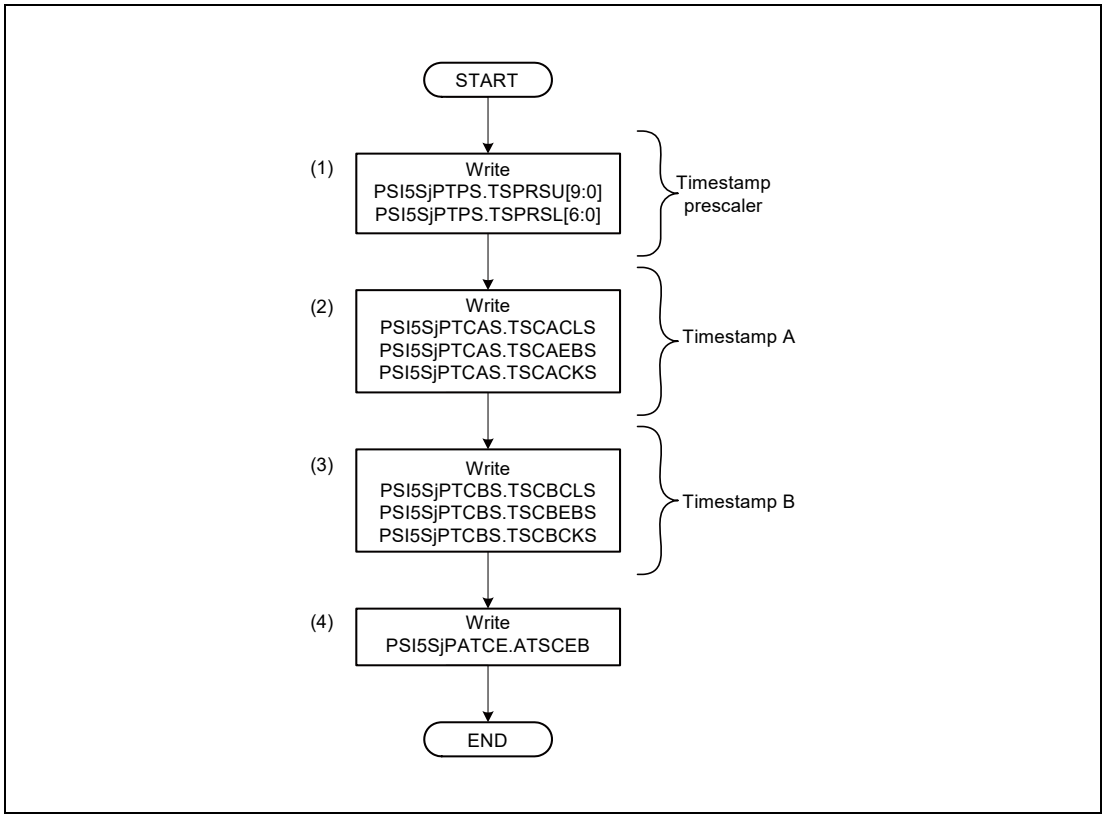


Figure 28.13 Timestamp Setting Procedure (for Batch Control)

(1) PSI5SjPTPS: PSI5S Timestamp Prescaler register
Specify the settings of the timestamp prescaler.

(2) PSI5SjPTCAS: PSI5S Timestamp Counter A Select register
Select the clear signal, enable signal, and clock signal for timestamp counter A.

(3) PSI5SjPTCBS: PSI5S Timestamp Counter B Select register
Select the clear signal, enable signal, and clock signal for timestamp counter B.

(4) PSI5SjPATCE: PSI5S All Timestamp Counter Enable register
Enable the operation of timestamp counters A and B together. By this setting, timestamp counters A and B start.

Note that clear instructions can also be issued for timestamp counters A and B separately or in a batched manner.

28.5.2.4 Interrupt Setting

This section describes how to set up interrupts.

Perform the procedure shown in **Figure 28.14** below for the required channels among channels 0 to 7.

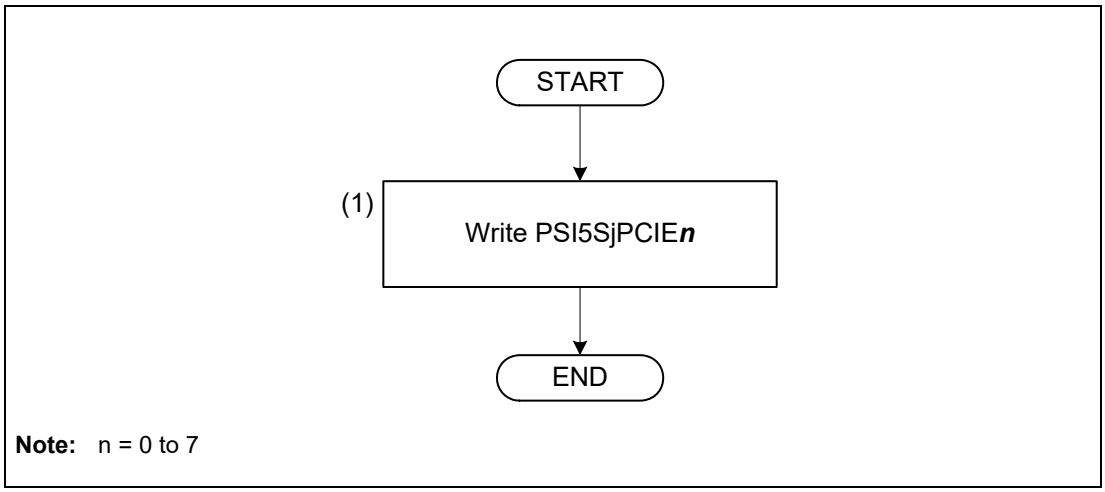


Figure 28.14 Interrupt Setting Procedure

(1) PSI5SjPCIE*n*: PSI5S CPU Interrupt Enable ch*n* register (*n*: 0 to 7)
Specify the settings of the interrupt enable bits according to the interrupt factors.

Table 28.90 below lists the enable bits in the interrupt enable register.

Table 28.90 Interrupt Enable Register Bits

Register Symbol	Bit Symbol	Factor
PSI5SjPCIE0	IEBXCRC	XCRC error
PSI5SjPCIE(0 to 7)	IEBCRC	CRC error
PSI5SjPCIE(0 to 7)	IEBPT	Parity error
PSI5SjPCIE(0 to 7)	IEBTRST	Transceiver status error
PSI5SjPCIE0	IEBUTPT	UART Rx parity error
PSI5SjPCIE0	IEBUTFR	UART Rx framing error
PSI5SjPCIE(0 to 7)	IEBRWDT	WDT error
PSI5SjPCIE(0 to 7)	IEBROV	PSI5 frame reception overrun error
PSI5SjPCIE(0 to 7)	IEBRFLK	PSI5 frame lack error
PSI5SjPCIE(0 to 7)	IEBRFEX	PSI5 frame excess error
PSI5SjPCIE(0 to 7)	IEBRFN	PSI5 frame reception finish
PSI5SjPCIE(1 to 7)	IEBDDSOW	DDSR overwrite error
PSI5SjPCIE(1 to 7)	IEBDDSFN	DDSR transmit finish
PSI5SjPCIE0	IEBCTFN	Command data transmit finish

28.5.2.5 DMA Transfer Request Setting

This section describes how to set up DMA transfer requests.

Perform the procedure shown in **Figure 28.15** below for the required channels among channels 0 to 7.

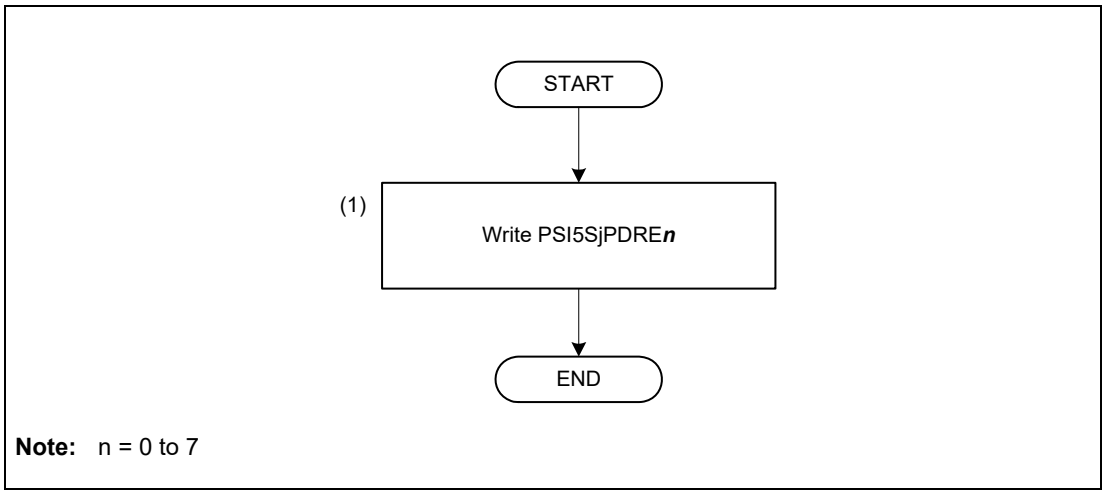


Figure 28.15 DMA Transfer Request Enable Setting Procedure

When you enable the output of DMA transfer requests, check the contents of the following status register:

(1) PSI5SjPDRE*n*: PSI5S DMA transfer Request Enable ch*n* register (*n*: 0 to 7)
Enable the output of DMA transfer requests.

Table 28.91 below lists the enable bits in the transfer request enable register.

Table 28.91 DMA Transfer Request Enable Register Bits

Register Symbol	Bit Symbol	Factor
PSI5SjPDRE(0 to 7)	DRQERFN	PSI5 last frame reception finish
PSI5SjPDRE(0 to 7)	DRQEWDT	WDT error
PSI5SjPDRE(1 to 7)	DRQETFN	DDSR transmit finish

28.5.2.6 Synchronization Pulse Setting Procedure

This section describes how to set up a synchronization pulse.

Perform the procedure shown in **Figure 28.16** below for the channels that require a synchronization pulse.

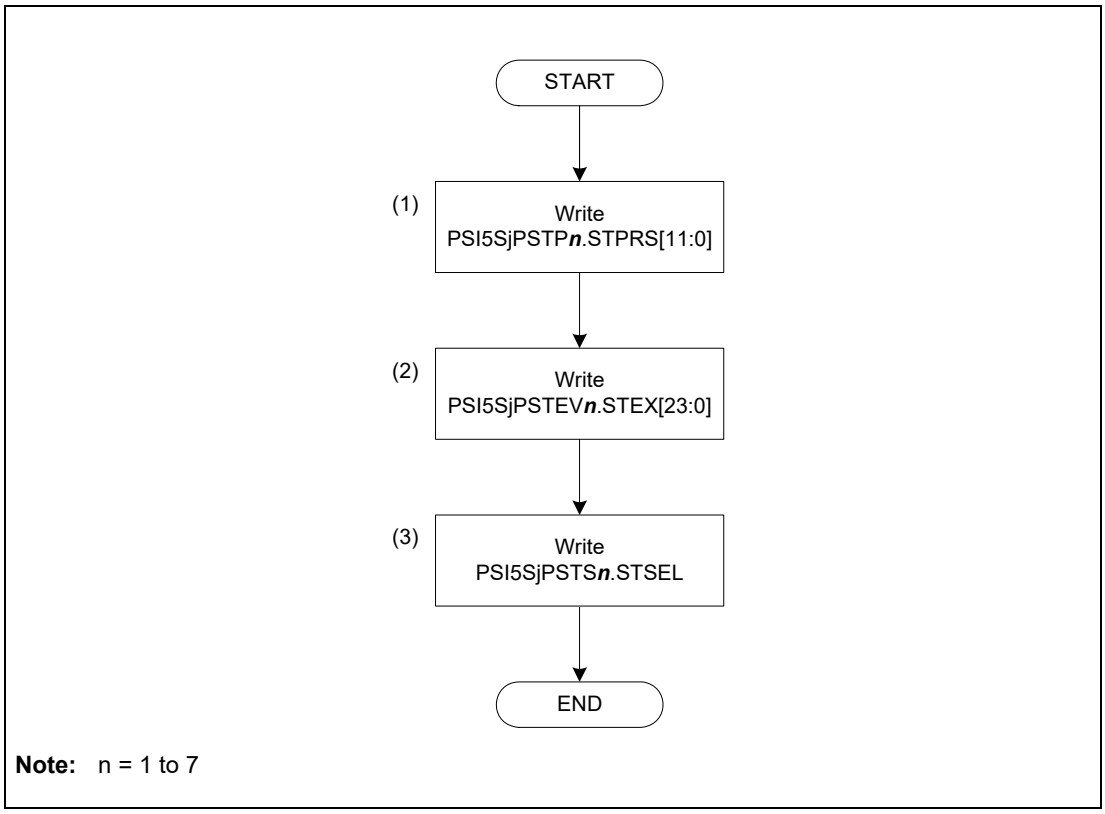


Figure 28.16 Synchronization Pulse Setting Procedure

(1) PSI5SjPSTP***n***: PSI5S Sync Trigger Prescaler ch***n*** register.
Specify settings of the sync trigger prescaler for channel n. (***n***: 1 to 7)

(2) PSI5SjPSTEV***n***: PSI5S Sync Trigger Expiration Value ch***n*** register.
Specify settings of the sync trigger expiration value for channel n. (***n***: 1 to 7)

(3) PSI5SjPSTS***n***: PSI5S Sync Trigger Select ch***n*** register.
Specify settings of the selection of synchronous trigger for channel n. (***n***: 1 to 7)

28.5.2.7 Tx Command Setting Procedure

This section describes how to set up the Tx command data for ECU-to-sensor communication.

Perform the procedure shown in **Figure 28.17** below for the Tx command of each channels that requires a synchronization pulse.

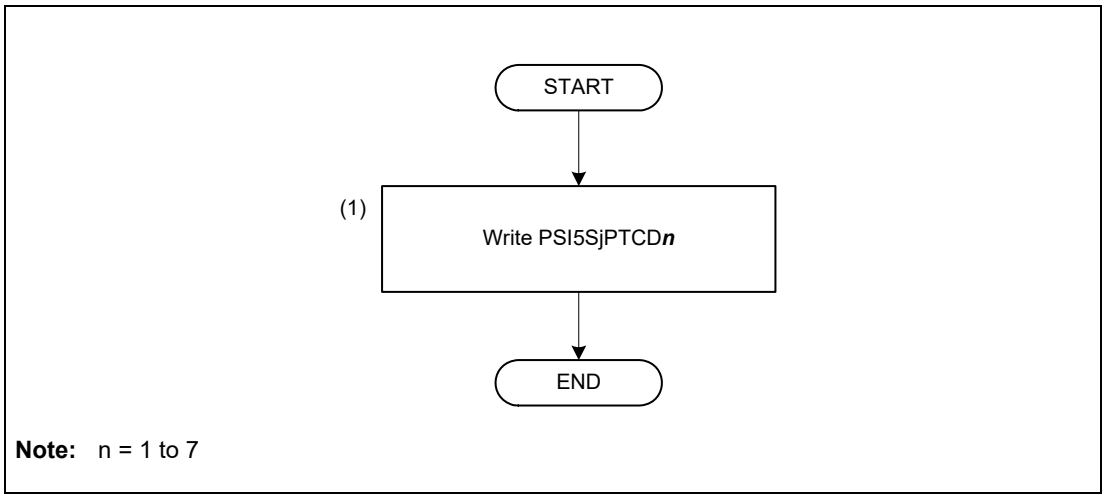


Figure 28.17 Tx Command Setting Procedure

(1) PSI5SjPTCD***n***: PSI5S Tx command Data ch***n*** register.
Specify the settings of the Tx command for channel ***n***. (***n***: 1 to 7)

28.5.3 UART Mode Setting Procedure

PSI5S has a UART mode in which PSI5S can operate as a UART with limited functions.

For details on the UART specifications, see **Table 28.9, Features**.

Perform the procedure shown in **Figure 28.18** below after performing the procedure described in **Section 28.5.1, Common Setting Procedure**.

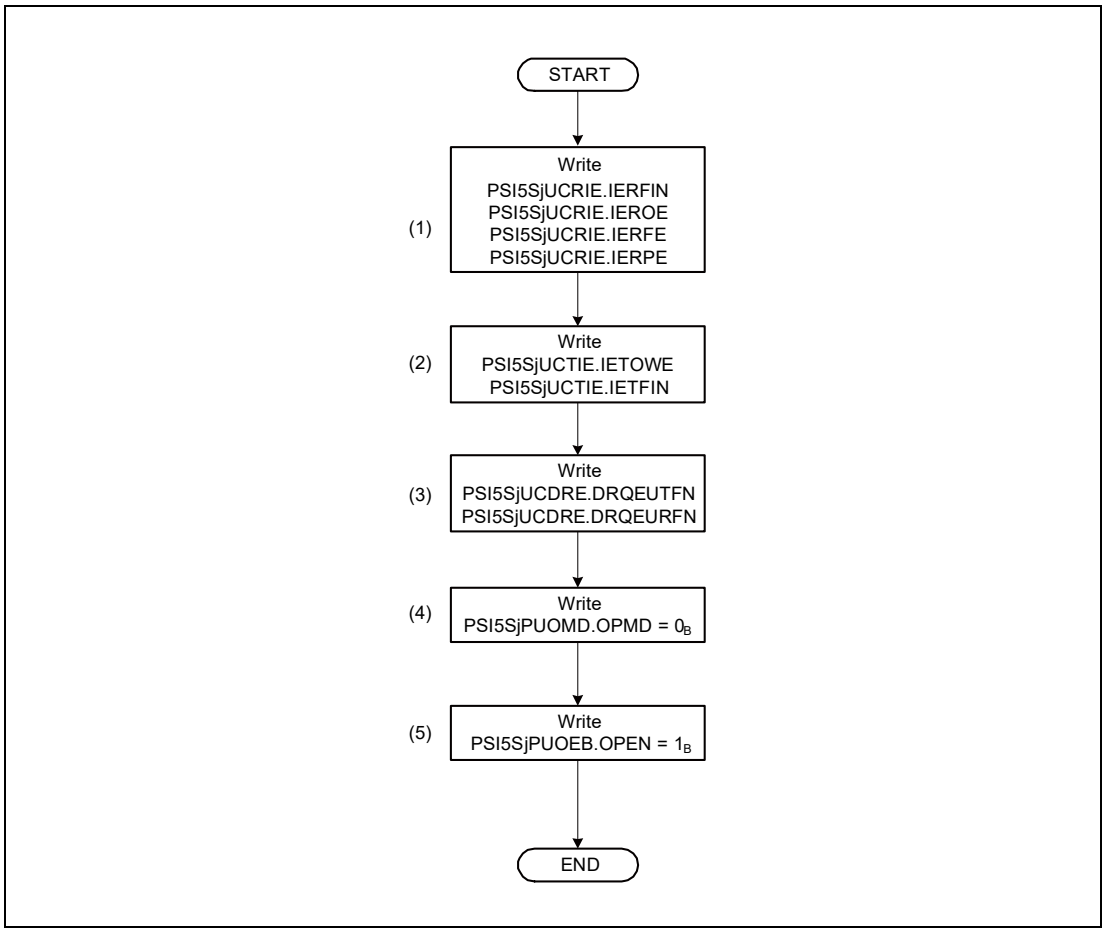


Figure 28.18 UART Mode Setting Procedure

(1) PSI5SjUCRIE: UART Communication Rx Interrupt Enable register
Specify settings of the interrupts at the end of UART frame reception, at the occurrence of a UART overrun error, at the occurrence of a UART framing error and at the occurrence of a UART parity error.

(2) PSI5SjUCTIE: UART Communication Tx Interrupt Enable register
Specify settings of the interrupts at the end of UART frame transmission and at the occurrence of a UART overwrite error.

(3) PSI5SjUCDRE: UART Communication DMA Request Enable register
Specify settings of the DMA transfer requests at the end of UART reception and at the end of UART transmission.

(4) PSI5SjPUOMD: PSI5S/UART Operation Mode register
Set the operation mode to UART mode.

(5) PSI5SjPUOEB: PSI5S/UART Operation Enable register
Start the operation of PSI5S.

28.5.4 Procedure for Transmitting Transceiver Commands

This section describes how to transmit commands for an externally connected PSI5 transceiver.

Perform this setting procedure after performing the procedure described in **Section 28.5.2, PSI5S mode Setting Procedure**.

PSI5 frame reception setting of Ch0 is required, because the response frame from the transceiver is stored in Ch0 MB.

(See **Section 28.5.2.1, PSI5 Frame Reception Setting**)

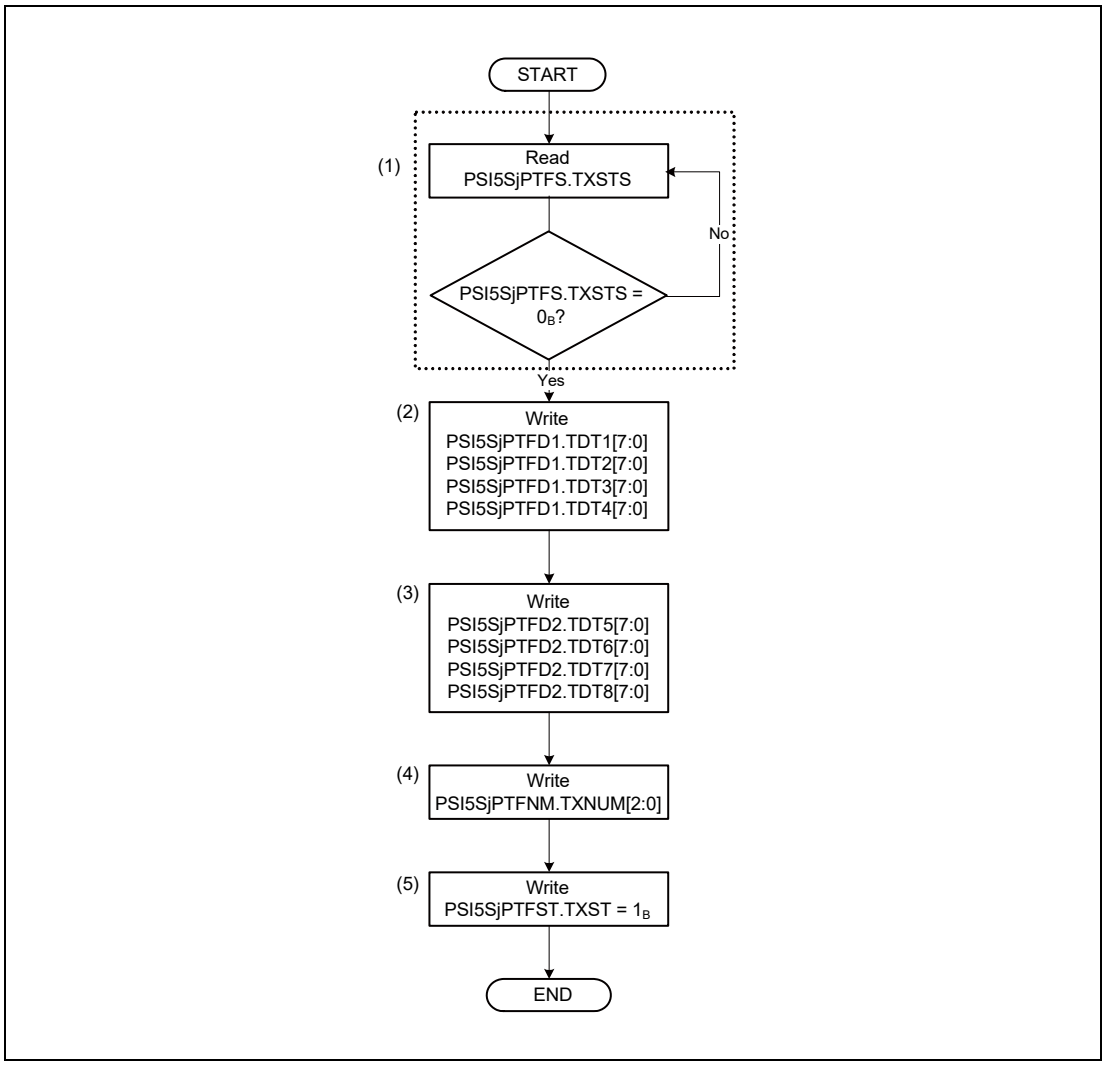


Figure 28.19 Transceiver Setting Procedure

(1) PSI5SjPTFS: PSI5S Tx Frame Status register
Read the TXSTS bit to check that the UART is not busy with transmission (TXSTS = 0). If the UART is busy (TXSTS = 1), wait until it is not busy.

(2) PSI5SjPTFD1: PSI5S Tx Frame Data1 register
Sequentially set the 1st to 4th bytes of the data to be transmitted continuously in this register. If the continuously transmitted data is less than 4 bytes, set only the data to be transmitted.

(3) PSI5SjPTFD2: PSI5S Tx Frame Data2 register
Sequentially set the 5th to 8th bytes of the data to be transmitted continuously in this register. If the continuously transmitted data is less than 9 bytes, set only the data to be transmitted.

(4) PSI5SjPTFNM: PSI5S Tx Frame Number register
Set the number of bytes to be transmitted continuously in this register.

(5) PSI5SjPTFST: PSI5S Tx Frame Start register
Start continuous transmission.

As an example, **Table 28.92** below describes the settings of PSI5SjPTFD1, PSI5SjPTFD2, and PSI5SjPTFNM registers to set up the external transceiver that uses the command format shown below in **Figure 28.20**.

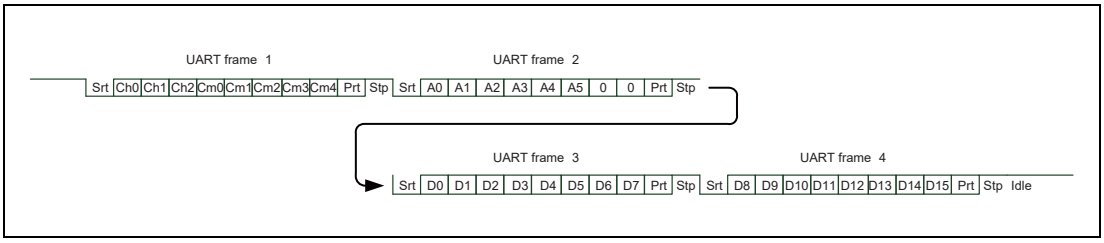


Figure 28.20 Example of Command Format

Table 28.92 List of Register Settings

Register Symbol	Bit Symbol	Value
PSI5SjPTFD1	TDT1[7:0]	{Cm[4:0], Ch[2:0]}
PSI5SjPTFD1	TDT2[7:0]	{0b00, A[5:0]}
PSI5SjPTFD1	TDT3[7:0]	D[7:0]
PSI5SjPTFD1	TDT4[7:0]	D[15:8]
PSI5SjPTFD2	TDT5[7:0]	N/A
PSI5SjPTFD2	TDT6[7:0]	N/A
PSI5SjPTFD2	TDT7[7:0]	N/A
PSI5SjPTFD2	TDT8[7:0]	N/A
PSI5SjPTFNM	TXNUM[2:0]	3_H

28.5.5 DDSR Transmission Procedure

This section describes how to perform DDSR transmission.

Perform the procedure shown in **Figure 28.21** below for channels 1 to 7.

This procedure must be performed when the operation mode is PSI5S mode.

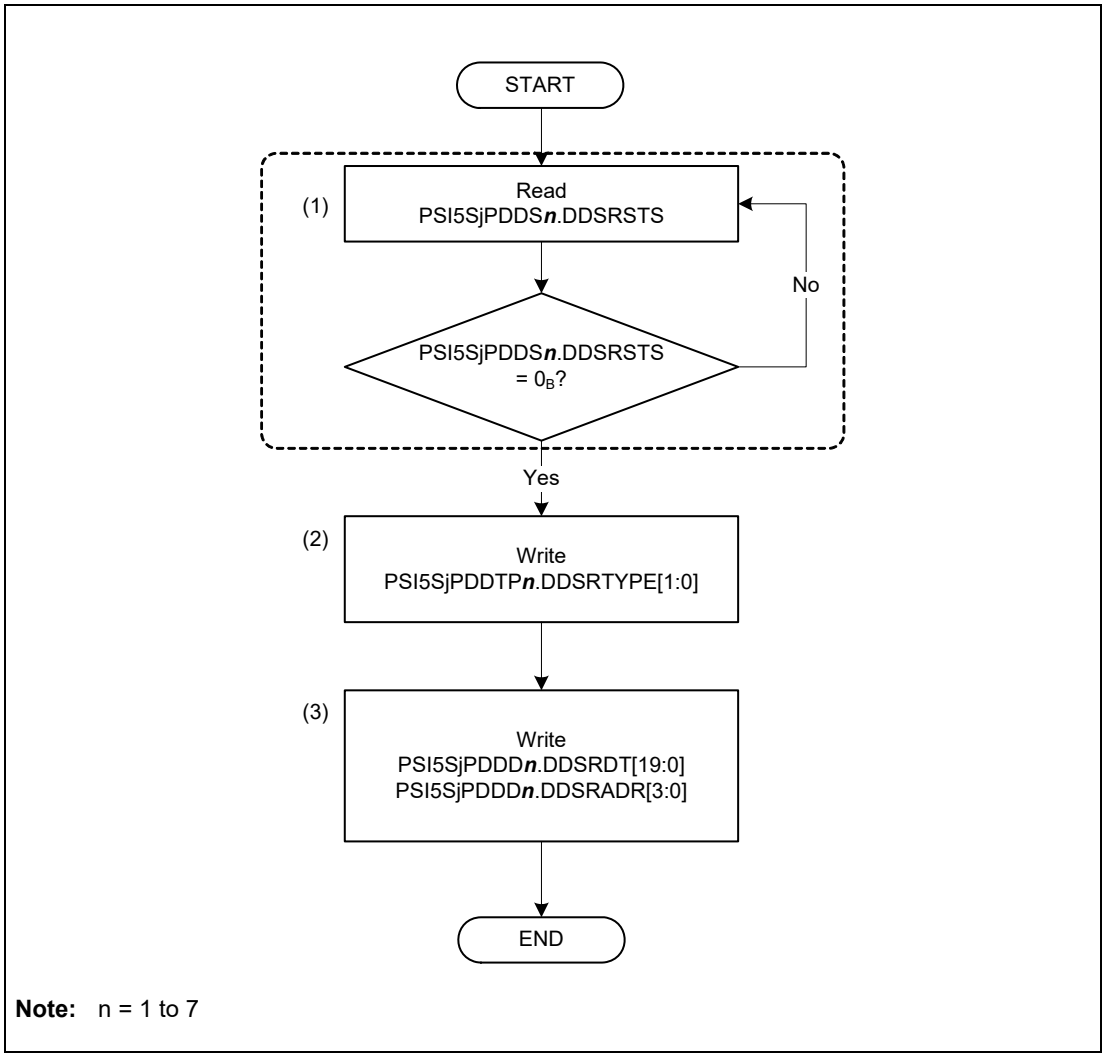


Figure 28.21 DDSR Transmission Procedure

(1) PSI5SjPDDS*n*. DDSRSTS: PSI5S DDSR Status Ch*n* register (*n*: 1 to 7)
Check the DDSR transmission status. When the DDSRSTS bit is 1, wait until the value changes to 0.
Note that if DDSR transmit data is written to the PSI5SjPDDD*n* register when the DDSRSTS bit is 1, an overwrite error occurs. For details, see **Section 28.6.2.9, Abnormal Transmission (Transmission Error)**.

(2) PSI5SjPDDTP*n*: PSI5S DDSR Type Ch*n* register (*n*: 1 to 7)
Specify the type of ECU-to-sensor data.

(3) PSI5SjPDDD*n*: PSI5S DDSR Data Ch*n* register (*n*: 1 to 7)
Set DDSR transmit data in this register.

Figure 28.22 shows the procedure of DDSR overwrite error processing.

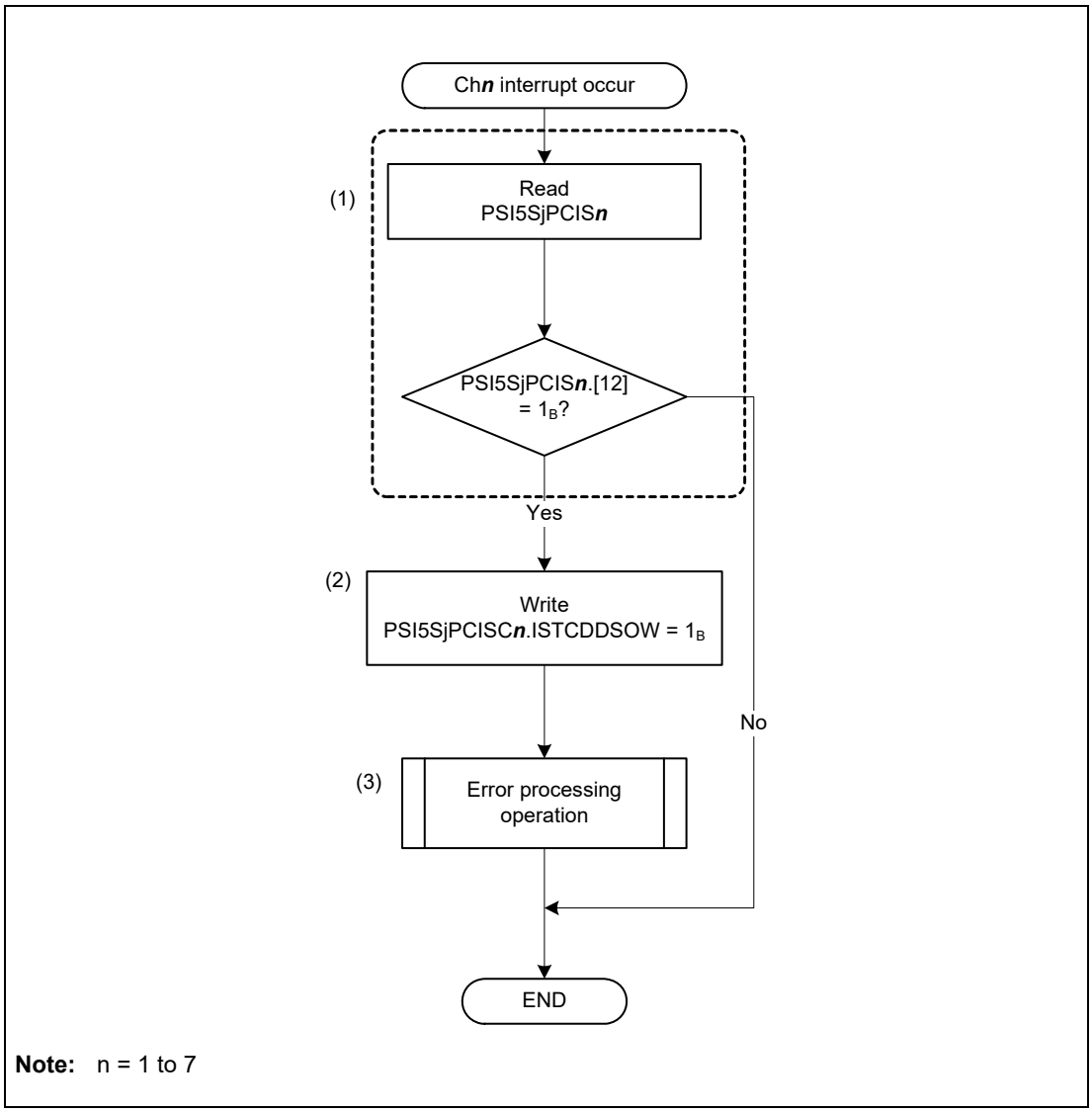


Figure 28.22 DDSR Overwrite Error Processing Procedure

(1) PSI5SjPCIS***n***: PSI5S CPU Interrupt Status ch***n*** register. (***n***: 1 to 7)
Read PSI5SjPCIS***n*** to check whether the DDSR overwrite error is set (PSI5SjPCIS***n***[12] = 1). If the DDSR overwrite error is not set (PSI5SjPCIS***n***[12] = 0), finish the processing.

(2) PSI5SjPCISC***n***: PSI5S CPU Interrupt Status chn Clear register. (***n***: 1 to 7)
Clear the DDSR overwrite error.

(3) Perform the processing to be done when a DDSR overwrite error occurs.

28.5.6 PSI5 Frame Reception Procedures

28.5.6.1 Normal Reception

This section describes the procedure of PSI-5 frame reception.

PSI5S can receive PSI5 frames only in PSI5S mode.

Figure 28.23 shows the operations to normally receive PSI5 frames.

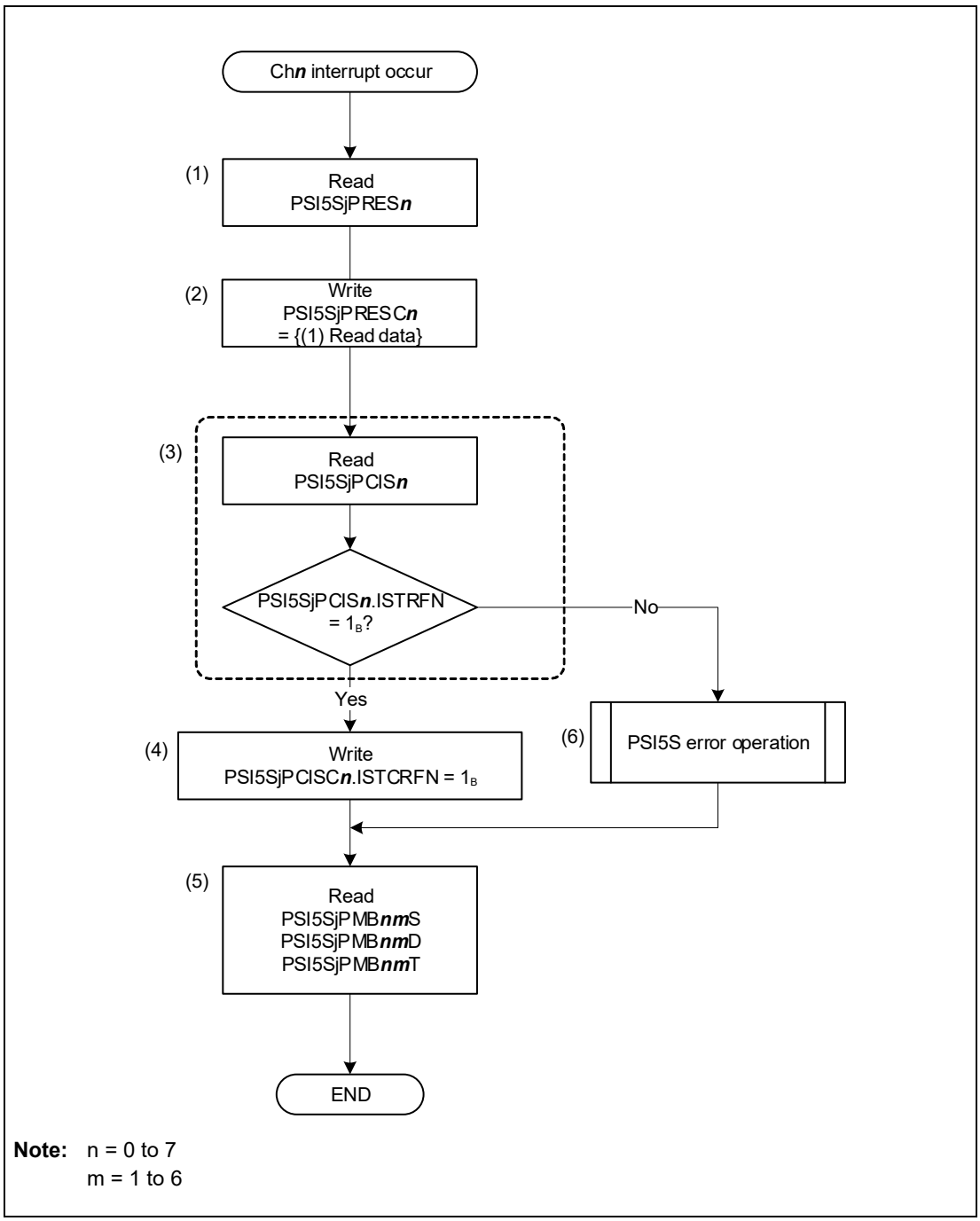


Figure 28.23 Procedure of Normal Reception of PSI5 Frames

(1) PSI5SjPRES*n*: PSI5S Receive Error Status ch*n* register (*n*: 0 to 7)
Check the frame that caused the interrupt. (This step is optional.)

(2) PSI5SjPRESC*n*: PSI5S Receive Error Status Clear ch*n* register (*n*: 0 to 7)
Clear the flag indicating the status of the frame that caused the interrupt. (This operation is optional.)

(3) PSI5SjPCIS*n*: PSI5S CPU Interrupt Status ch*n* register (*n*: 0 to 7)
Read the PSI5S interrupt status register to check that the ISTRFN bit is set.

(4) PSI5SjPCISC*n*: PSI5S CPU Interrupt Status ch*n* register (*n*: 0 to 7)
Clear the flag indicating the end of frame reception.

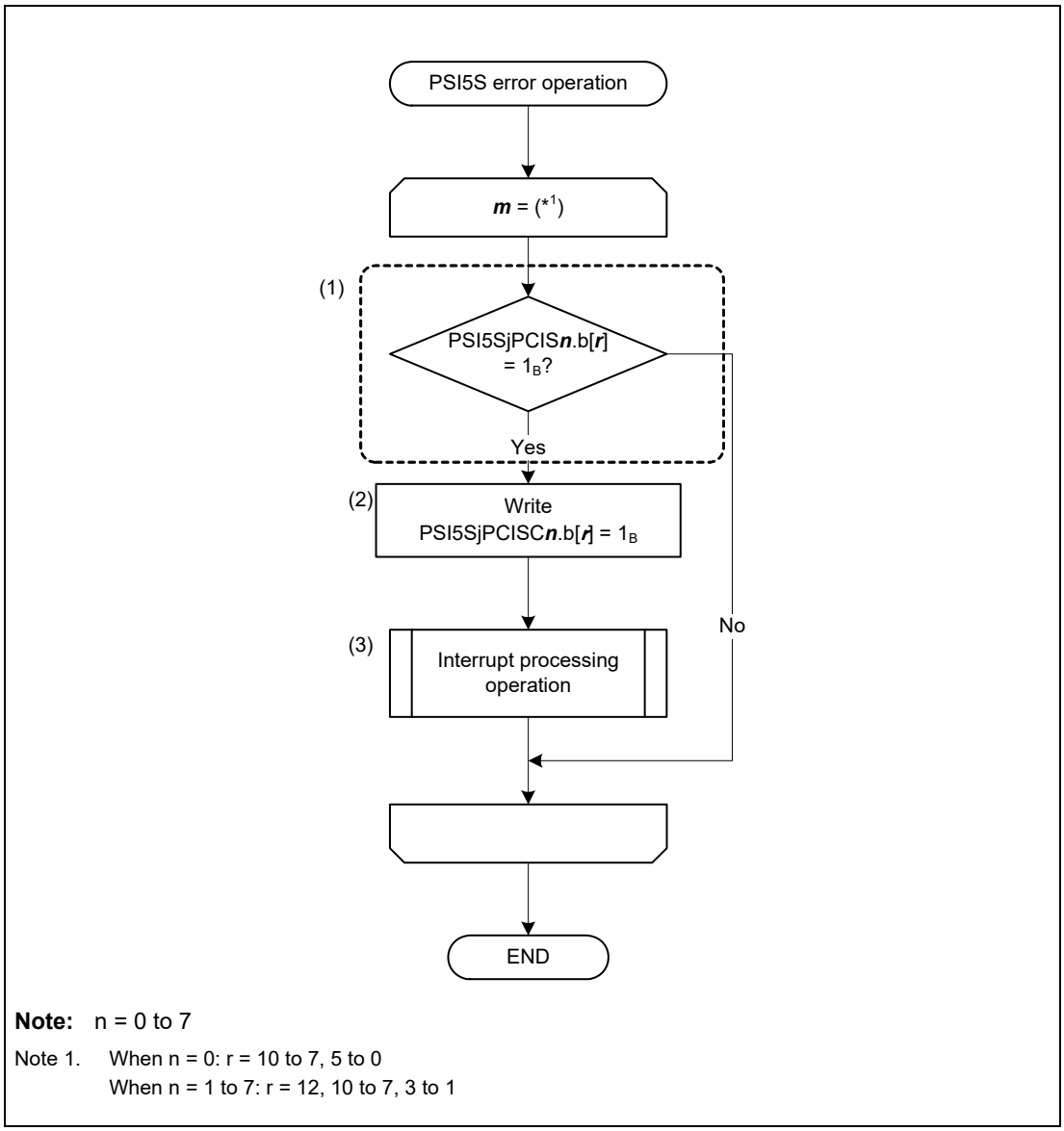
(5) PSI5SjPMB*nm*S: PSI5S receive MailBox ch*n* frm*m* Status register (*n*: 0 to 7, *m*: 1 to 6)
PSI5SjPMB*nm*D: PSI5S receive MailBox ch*n* frm*m* Data register (*n*: 0 to 7, *m*: 1 to 6)
PSI5SjPMB*nm*T: PSI5S receive MailBox ch*n* frm*m* Timestamp register (*n*: 0 to 7, *m*: 1 to 6)
Read the PSI5 frame status, data, and timestamp stored in the mailbox.

(6) If the ISTRFN bit is not set, perform error processing.

28.5.6.2 Abnormal Reception

This section describes how to process an error during PSI5 frame reception.

For the processes from the occurrence of an interrupt to the reading of the interrupt status, see processes (1) and (2) in **Figure 28.24**.



Note: n = 0 to 7

Note 1. When n = 0: r = 10 to 7, 5 to 0
When n = 1 to 7: r = 12, 10 to 7, 3 to 1

Figure 28.24 Procedure of Error Processing in PSI5 Frame Reception

(1) Check the cause of the error indicated in the PSI5S interrupt status register.

(2) PSI5SjPCISC*n*: PSI5S CPU Interrupt Status ch*n* register (*n*: 0 to 7)
If an error-cause flag is set, clear the flag.

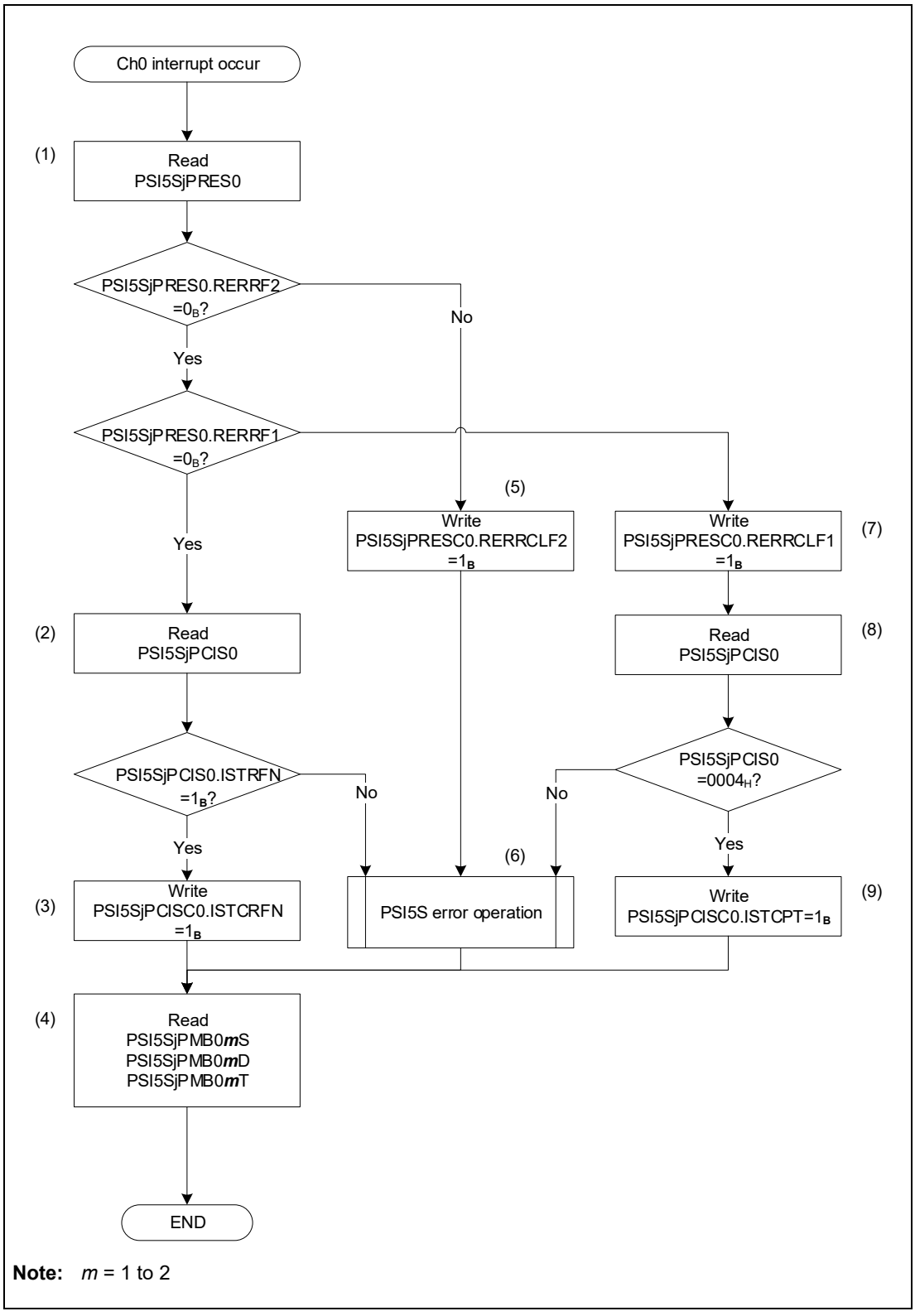
(3) Perform the error processing that corresponds to the indicated cause of the error.

Figure 28.74 shows the occurrence timing of errors other than WDT errors during PSI5 frame reception.

For the occurrence timing of WDT errors, see **Section 28.6.2.8, Abnormal Reception (WDT Error)**.

28.5.6.3 Response Frame from Transceiver Reception

This section describes the procedure of the response frame from the transceiver reception.



Note: *m* = 1 to 2

Figure 28.25 Procedure of Response Frame from Transceiver Reception

(1) PSI5SjPRES0: PSI5S Receive Error Status ch0 register.
Check the error frame that caused the interrupt.

(2) PSI5SjPCIS0: PSI5S CPU Interrupt Status ch0 register.
Read the PSI5S interrupt status register to check that the ISTRFN bit is set.

(3) PSI5SjPCISC0: PSI5S CPU Interrupt Status ch0 register.
Clear the flag indicating the end of frame reception.

(4) PSI5SjPMB0***m***S: PSI5S receive MailBox ch0 frm***m*** Status register (***m***: 1 to 2)
PSI5SjPMB0***m***D: PSI5S receive MailBox ch0 frm***m*** Data register (***m***: 1 to 2)
PSI5SjPMB0***m***T: PSI5S receive MailBox ch0 frm***m*** Timestamp register (***m***: 1 to 2)
Read the PSI5 frame status, data, and timestamp stored in the mailbox.

(5) PSI5SjPRESC0: PSI5S Receive Error Status Clear ch0 register.
Clear the flag indicating the error of frame 2.

(6) When the following cases, perform error processing.
- The flag indicating the end of reception is not set.
- The error frame occurs in frame 2 of ch0.
- The error frame occurs in frame 1 of ch0, and it has been set to other than a payload data parity error.

(7) PSI5SjPRESC0: PSI5S Receive Error Status Clear ch0 register.
Clear the flag indicating the error of frame 1.

(8) PSI5SjPCIS0: PSI5S CPU Interrupt Status ch0 register.
Read the PSI5S CPU Interrupt Status ch0 register to check that none of the flags except the ISTPT bit have been set.

(9) PSI5SjPCISC0: PSI5S CPU Interrupt Status ch0 register.
Clear the flag indicating the payload data parity error.

For the setting of frame reception from a transceiver, see **Section 28.5.2.1(1), PSI5 Frame Reception Setting (no payload checksum in reception frame)**.

28.5.7 Initialization and Operation Stop Procedures

28.5.7.1 Initialization by Software Reset

This section describes how to initialize PSI5S by a software reset.

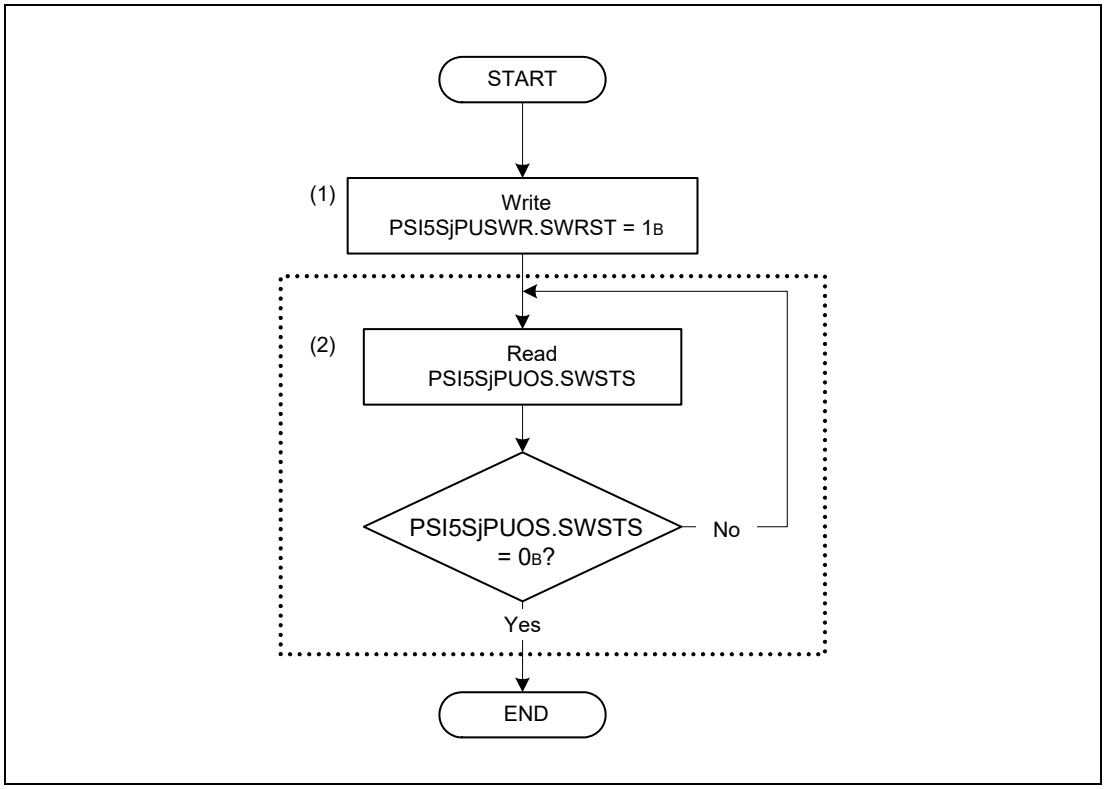


Figure 28.26 Initialization by Software Reset Procedure

(1) PSI5SjPUSWR: PSI5S/UART Software Reset register
Issue a software reset.

(2) PSI5SjPUOS: PSI5S/UART Operation Status register
Wait until the SW reset bit changes to 0.

28.5.7.2 DDSR Transmission Stop

This section describes how to stop DDSR transmission.

The transmission stop operation can be performed for individual channels.

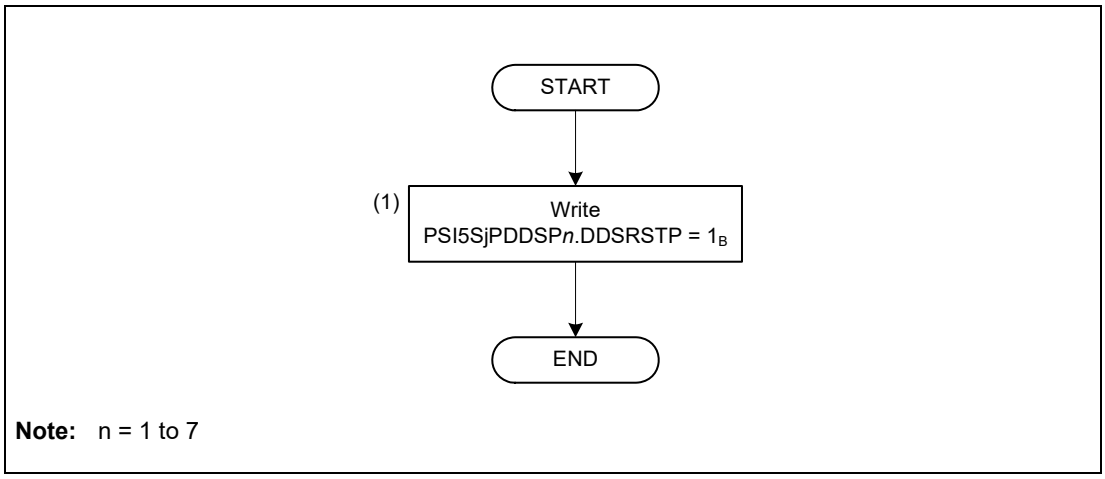


Figure 28.27 DDSR Transmission Stop Operation Procedure

(1) PSI5SjPDDSP*n*: PSI5S DDSR Stop ch*n* register (*n*: 1 to 7)
Stop data transmission from DDSR.

28.5.7.3 MailBox Initialization

This section describes the operation to initialize mailboxes (MBs).

This operation initializes the MBs for all channels together.

This operation must be performed when PSI5S is in configuration mode.

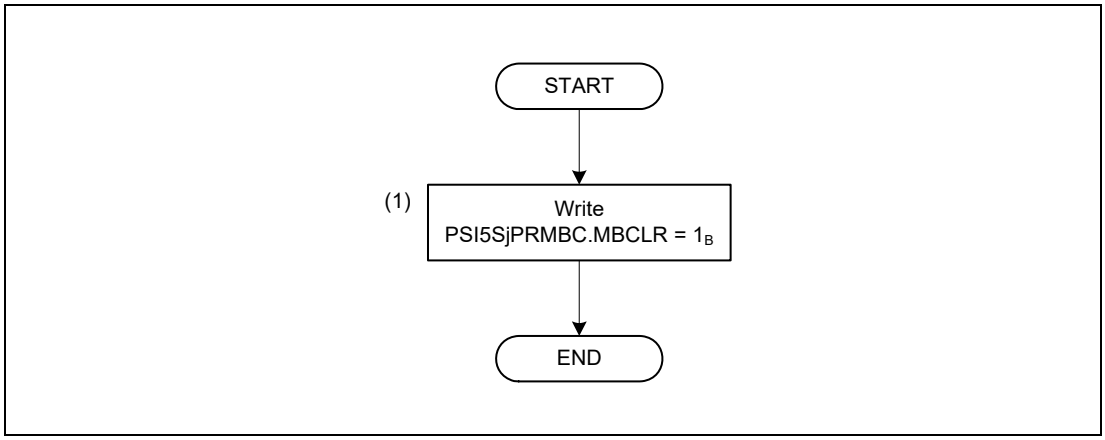


Figure 28.28 MailBox Initialization Procedure

(1) PSI5SjPRMBC: PSI5S Receive MailBox Data Clear register
Write 1 to the MBCLR bit to clear the mailboxes.

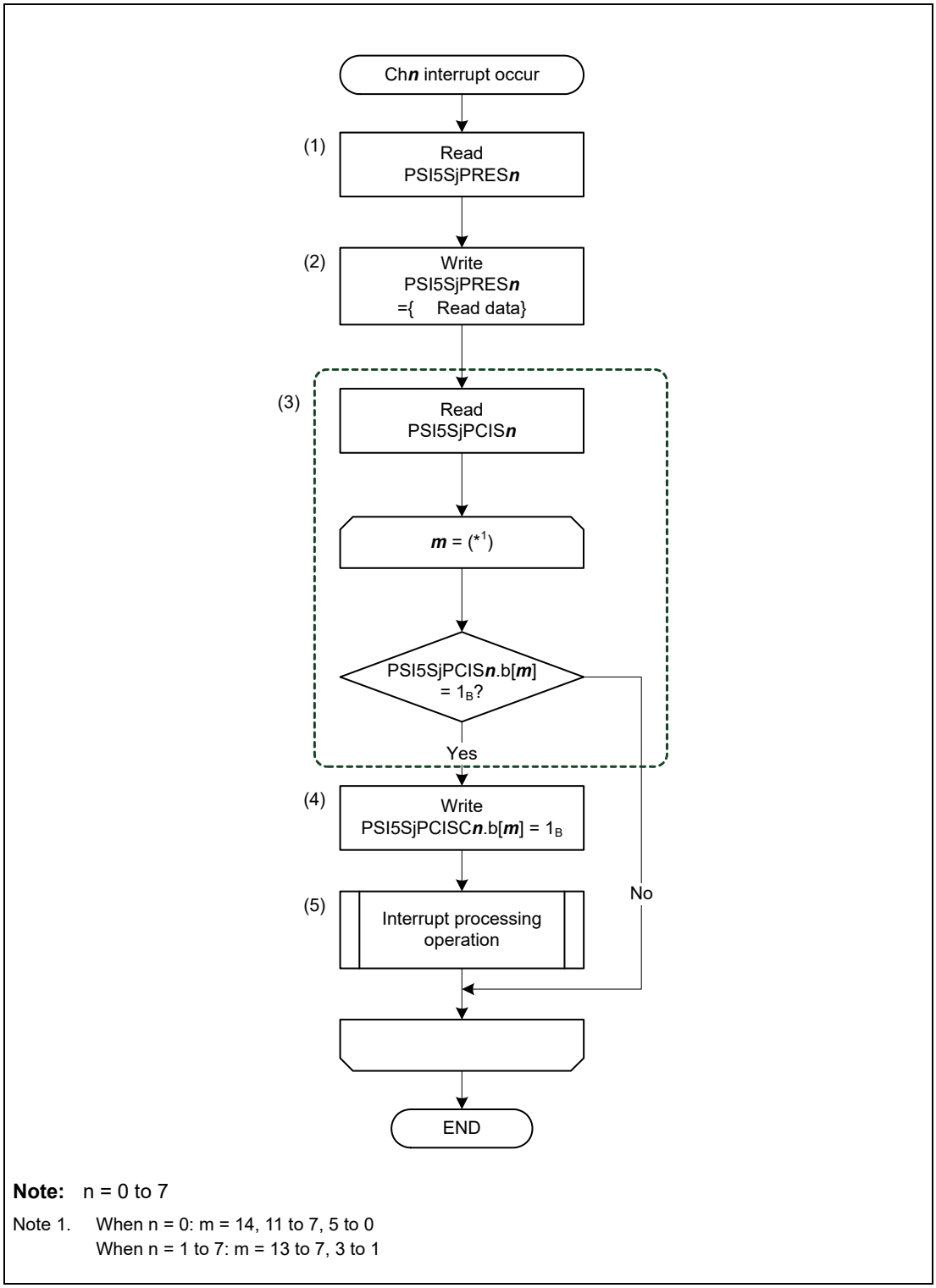
28.5.8 Interrupt Processing Procedures

28.5.8.1 Interrupt Processing in PSI5S mode

This section describes how to process interrupts in PSI5S mode.

Interrupts occur on individual channels.

This interrupt processing operation can be used for channels 0 to 7.



Note: n = 0 to 7

Note 1. When n = 0: m = 14, 11 to 7, 5 to 0
When n = 1 to 7: m = 13 to 7, 3 to 1

Figure 28.29 Interrupt Processing Procedure

(1) PSI5SjPRES*n*: PSI5S Receive Error Status ch*n* register (*n*: 0 to 7)
Check the frame that caused the reception error. (This step is optional.)

(2) PSI5SjPRESC*n*: PSI5S Receive Error Status Clear chn register (*n*: 0 to 7)
Clear the flag indicating the status of the frame that caused the reception error. (This step is optional.)

(3) PSI5SjPCIS*n*: PSI5S CPU Interrupt Status ch*n* register (*n*: 0 to 7)
Read the interrupt status register, and check all interrupt factors.

(4) PSI5SjPCISC*n*: PSI5S CPU Interrupt Status ch*n* register (*n*: 0 to 7)
Clear the bit corresponding to the relevant interrupt factor.

(5) Perform the interrupt process that corresponds to the interrupt factor.

Note that processes (3) and (4) must be performed for each interrupt factor bit.

28.5.8.2 Interrupt Processing in UART Mode

This section describes how to process interrupts in UART mode.

Interrupts can occur during both UART reception and UART transmission.

For the processing of interrupts during UART reception, see **Section 28.5.10.1, UART Reception**.

For the processing of interrupts during UART transmission, see **Section 28.5.10.2, UART Transmission**.

28.5.9 DMA Request Processing Procedures

28.5.9.1 DMA Request (Reception) Processing in PSI5S mode

This section describes how to process DMA requests (reception) in PSI5S mode.

DMA requests (reception) can occur on individual channels.

This request processing operation can be used for channels 0 to 7.

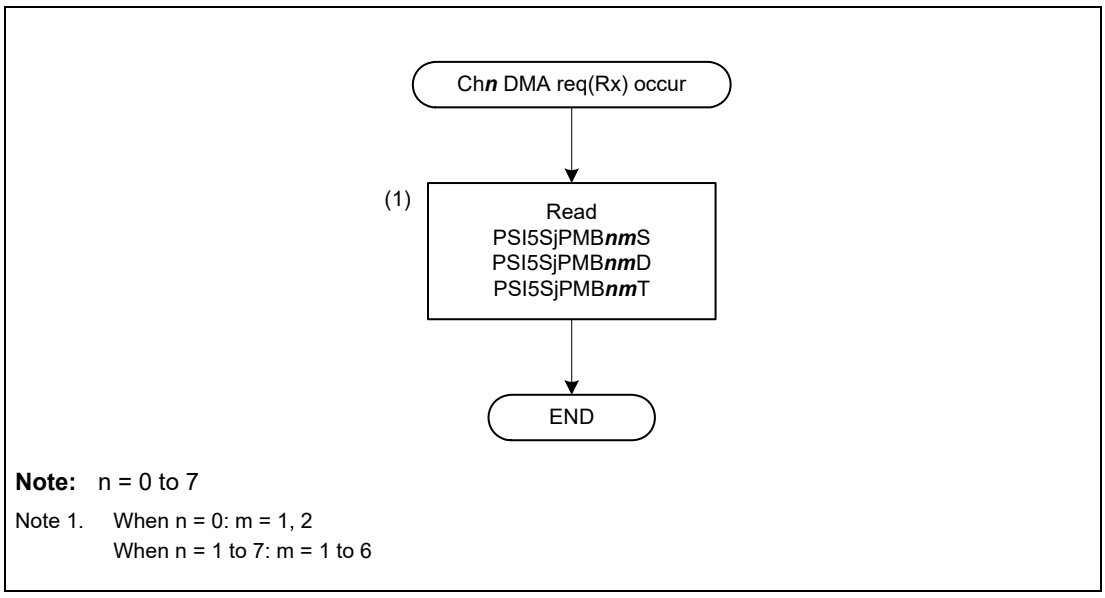


Figure 28.30 Procedure of DMA Request (Reception) Processing in PSI5S mode

(1) PSI5SjPMB*nm*S: PSI5S Receive MailBox ch0 Frm1 Status register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
PSI5SjPMB*nm*D: PSI5S Receive MailBox ch0 Frm1 Data register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
PSI5SjPMB*nm*T: PSI5S Receive MailBox ch0 Frm1 Timestamp register (when $n = 0$, $m = 1$ to 2; when $n = 1$ to 7, $m = 1$ to 6)
Check the mailbox data corresponding to the relevant frame.

NOTE

In asynchronous mode of Ch1 to Ch0, the data storage position is switched from frm1 to frm2, ..., frm6, and frm1, in this order, each time a PSI5 frame is received. Therefore, when received data is to be read by DMA, storage addresses must be managed by software. (Data storage position of Ch0 is always frm1.)

28.5.9.2 DMA Request (Transmission) Processing in PSI5S mode

This section describes how to process DMA requests (transmission) in PSI5S mode.

DMA requests (transmission) can occur on individual channels.

This request processing operation can be used for channels 0 to 7.

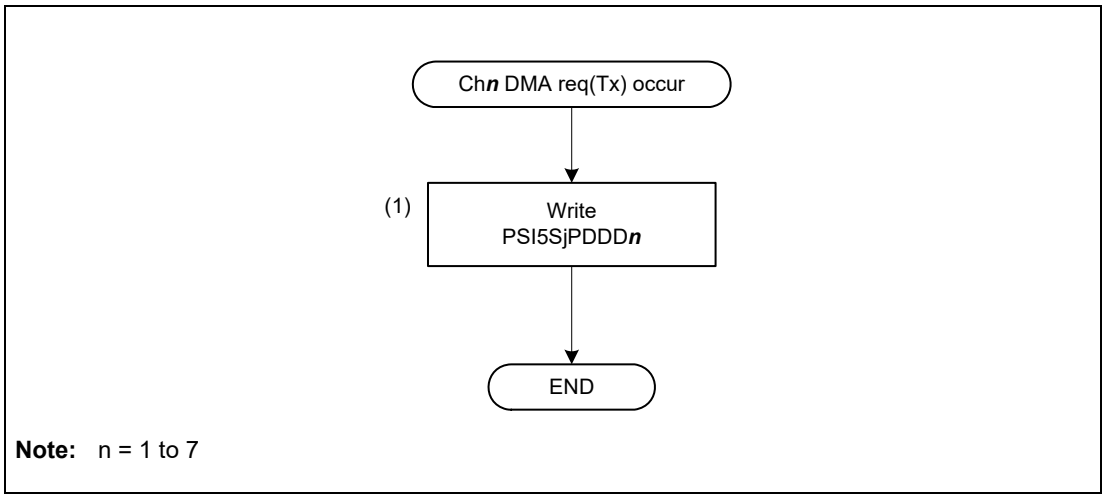


Figure 28.31 Procedure of DMA Request (Transmission) Processing in PSI5S mode

(1) PSI5SjPDDD*n*: PSI5S DDSR Data Ch*n* register (***n***: 1 to 7)
Set transmit data in DDSR.

28.5.9.3 DMA Request (Reception) Processing in UART Mode

This section describes how to process DMA requests (reception) in UART mode.

DMA requests (reception) occur on channel 7.

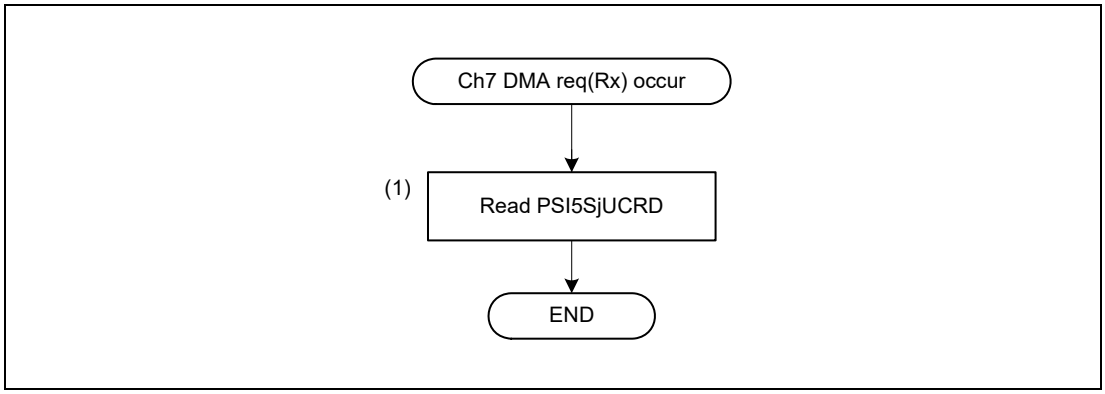


Figure 28.32 Procedure of DMA Request (Reception) Processing in UART Mode

(1) PSI5SjUCRD: UART Communication Rx Data register
Read receive data from this register.

28.5.9.4 DMA Request (Transmission) Processing in UART Mode

This section describes how to process DMA requests (transmission) in UART mode.

DMA requests (transmission) occur on channel 7.

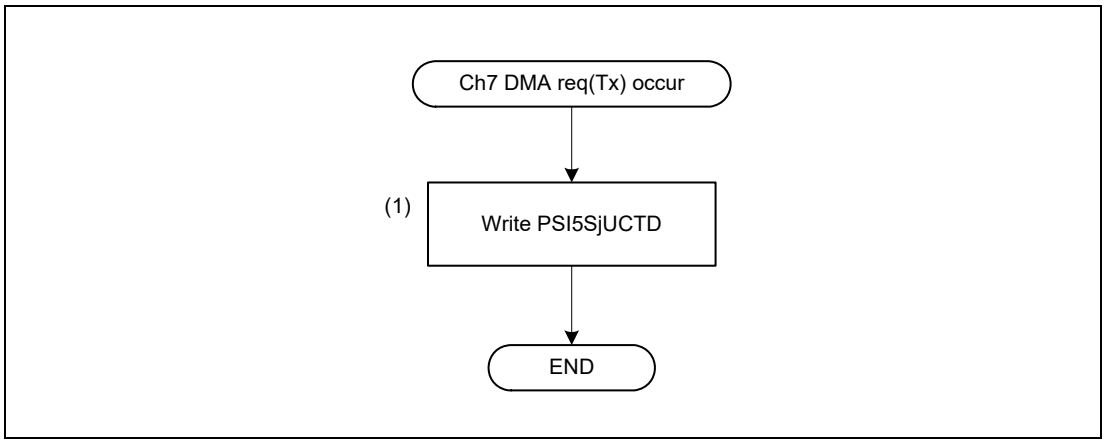


Figure 28.33 Procedure of DMA Request (Transmission) Processing in UART Mode

(1) PSI5SjUCTD: UART Communication Tx Data register
Write transmit data to this register.

28.5.10 UART Mode Communication Procedures

This section describes UART communication procedures.

UART communication can be performed only in UART mode.

28.5.10.1 UART Reception

Figure 28.34 below shows the processing procedure of normal UART reception.

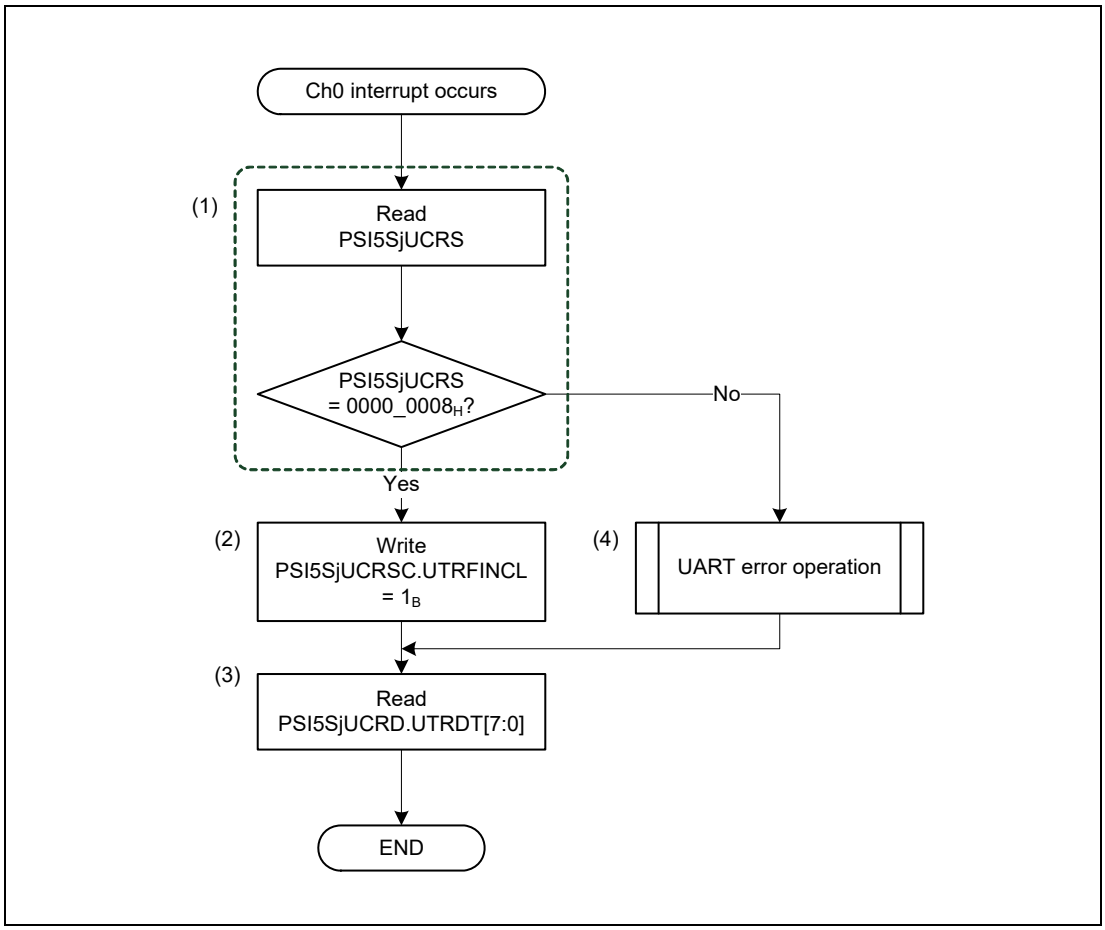


Figure 28.34 UART Reception Procedure

(1) PSI5SjUCRS: UART Communication Rx Status register
Read the UART reception status register, and check that the flag indicating the end of reception is set.

(2) PSI5SjUCRSC: UART Communication Rx Status Clear register
If the flag indicating the end of reception is set, clear it.

(3) PSI5SjUCRD: UART Communication Rx Data register
Read receive data from this register.

Note: If this step is skipped, a reception overrun error occurs at the time of receiving the next data.

(4) If the flag indicating the end of reception is not set, perform error processing.

Figure 28.35 shows the processing to be performed when a reception overrun error, receive framing error, or receive parity error occurs.

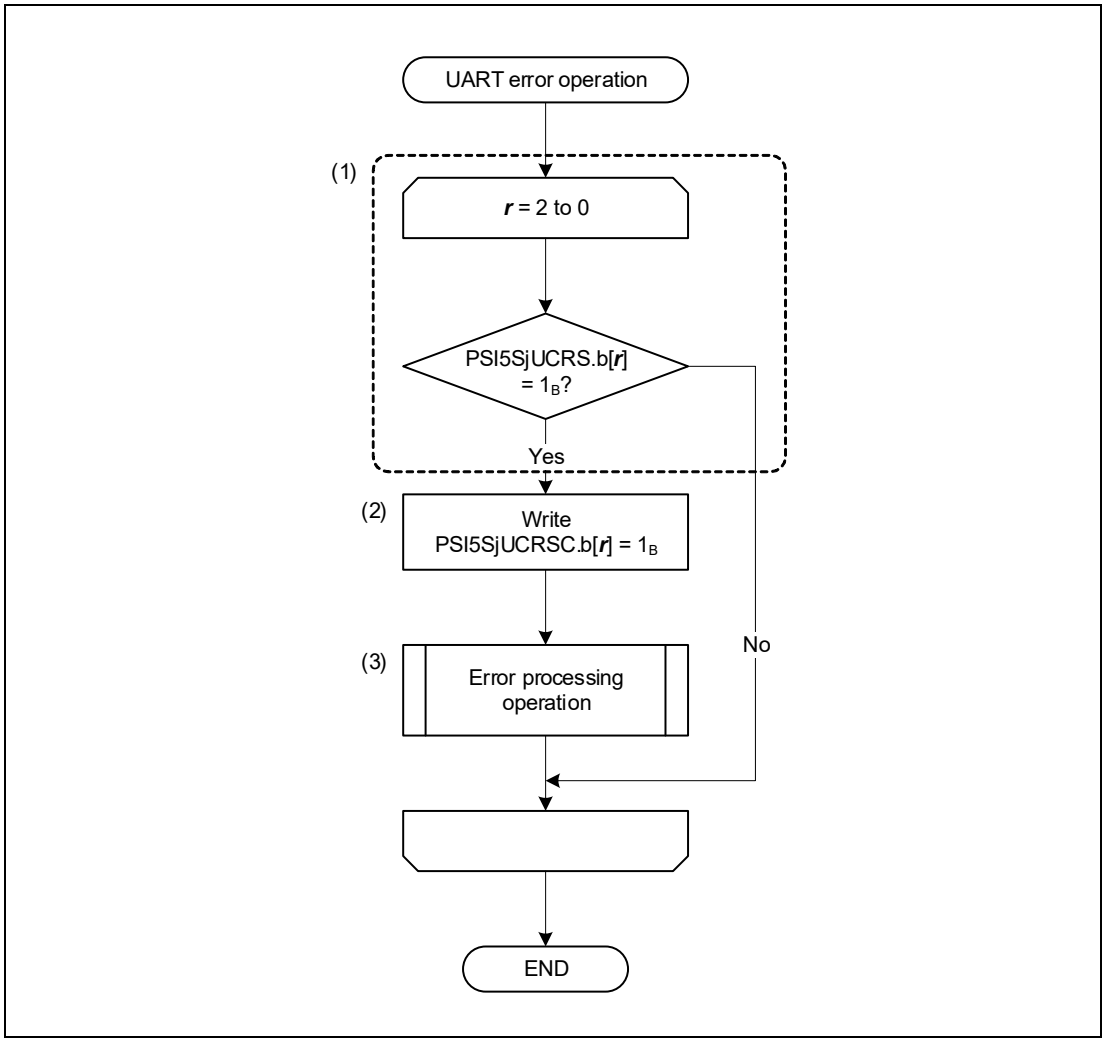


Figure 28.35 UART Reception Error Processing Procedure

(1) Check the states of error factor bits in the UART reception status register.

(2) PSI5SjUCRSC: UART Communication Rx Status Clear register
If an error-cause flag is set, clear the flag.

(3) Perform the processing corresponding to the indicated error cause.

28.5.10.2 UART Transmission

The procedure of UART transmission uses the UART transmission busy flag or the transmission end interrupt.

Figure 28.36 shows the UART transmission procedure using the UART transmission busy flag.

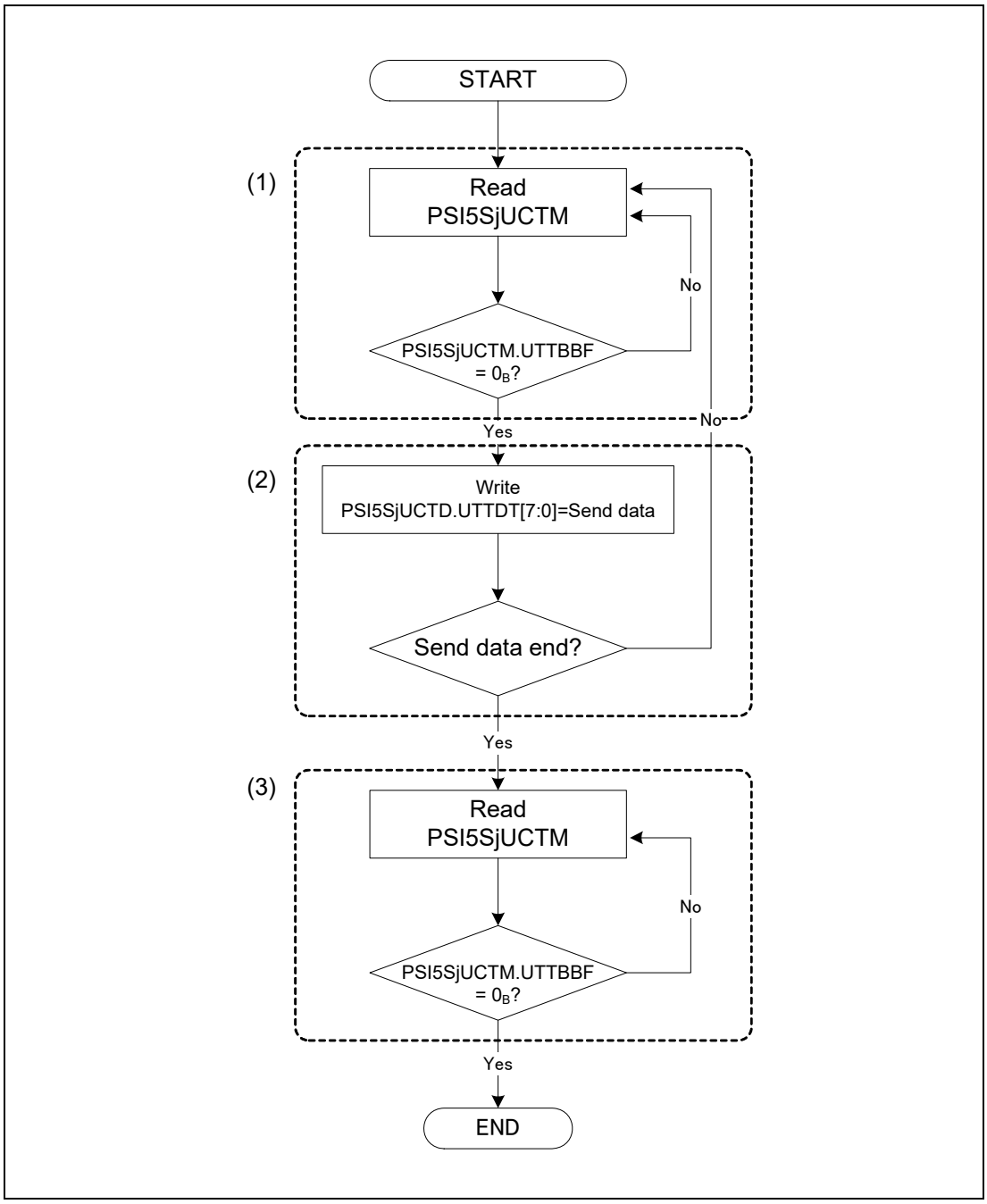


Figure 28.36 UART Transmission Procedure (Using UART Transmission Busy Flag)

(1) PSI5SjUCTM: UART Communication Tx Monitoring register
Read the UART communication Tx monitoring register to check that the UART Tx buffer is not busy.

(2) PSI5SjUCTD: UART Communication Tx Data register
Write transmit data to this register.

(3) PSI5SjUCTM: UART Communication Tx Monitoring register
Read the UART communication Tx monitoring register to check that the UART Tx buffer is not busy.

Figure 28.37 below shows the UART transmission procedure using the transmission end interrupt.

Because no transmission end interrupt of the previous transmission is generated for the first transmission operation, the UART transmission busy flag must be used for the first transmission operation and the transmission end interrupt thereafter.

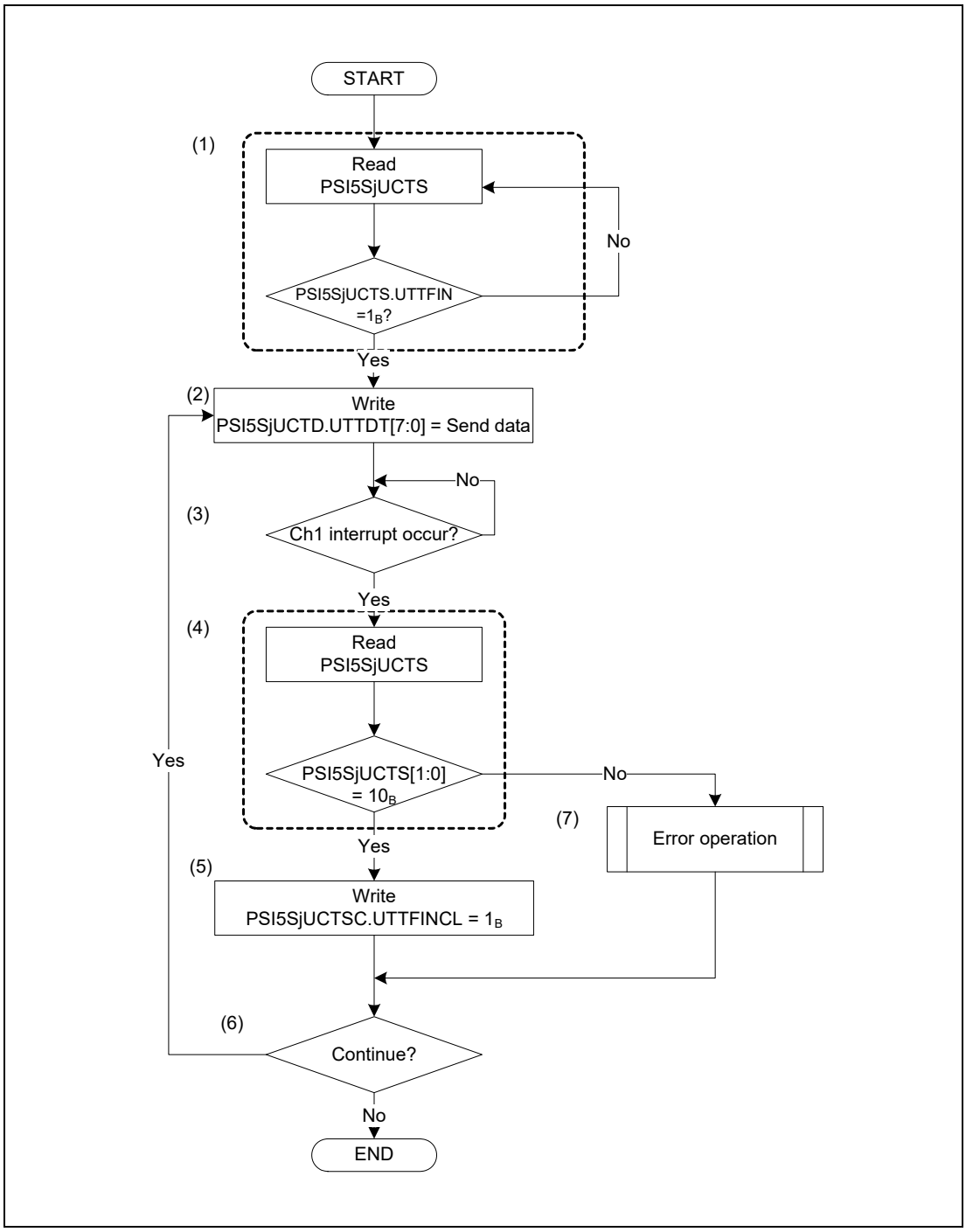


Figure 28.37 UART Transmission Procedure (Using Transmission End Interrupt)

(1) PSI5SjUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the UART is ready for transmission.

(2) PSI5SjUCTD: UART Communication Tx Data register
Write transmit data to this register.

(3) Wait until a channel 1 interrupt occurs. When it occurs, proceed to step (4).

(4) PSI5SjUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the transmission end flag is set and the overwrite error flag is not set.

(5) PSI5SjUCTSC: Uart Communication Tx Status Clear register
Clear the flag indicating the end of transmission.

(6) To continue the processing, proceed to step (2). If not, finish the processing.

(7) Perform the corresponding error processing, see **Figure 28.38**.

Figure 28.38 below shows the procedure of the processing to be performed when a transmission overwrite error occurs.

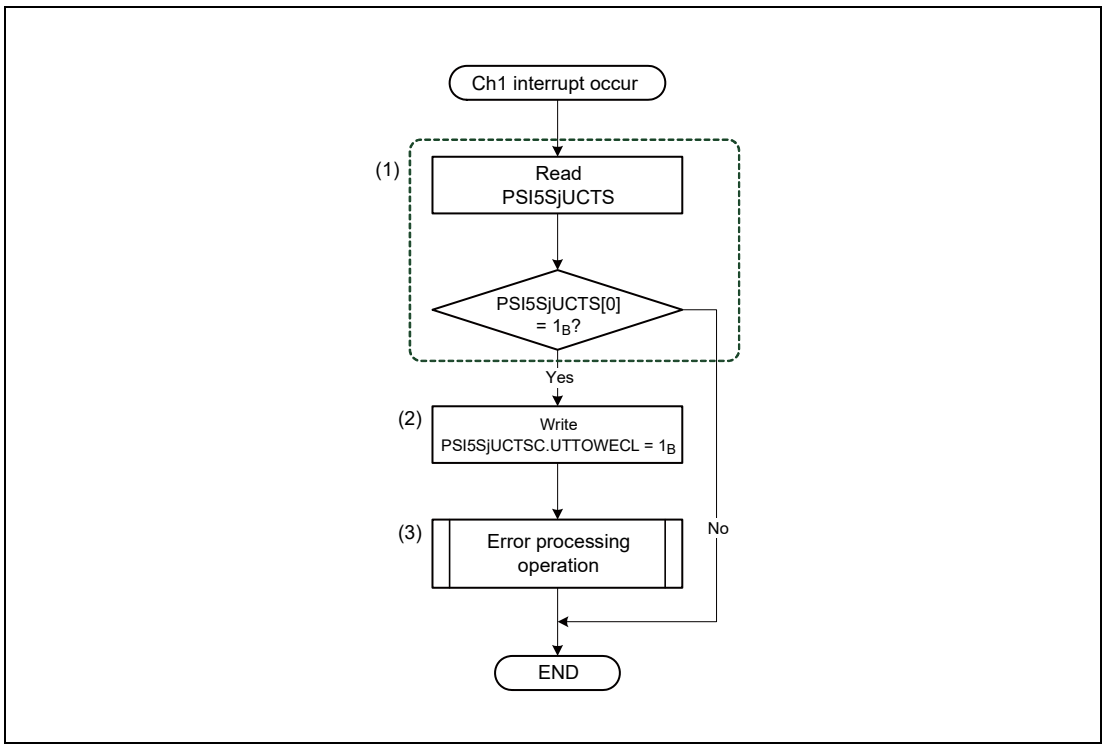


Figure 28.38 UART Transmission Error Processing Procedure

(1) PSI5SjUCTS: UART Communication Tx Status register
Read the UART communication status register to check that the transmission overwrite error flag is set.
If the flag is not set, finish the processing.

(2) PSI5SjUCTSC: UART Communication Tx Status Clear register
Clear the transmission overwrite error flag.

(3) Perform the processing to be done when a transmission overwrite error occurs.

28.5.11 Loopback Test

28.5.11.1 Loopback Test Setting Procedure

This setting procedure must be performed after the common setting procedure (**Section 28.5.1, Common Setting Procedure**).

Figure 28.39 below shows the Loopback test setting procedure.

PSI5S enters the test mode only when the values below are written to the register in the order shown in the figure three times.

If a different value is written midway, the whole procedure becomes invalid. In such a case, perform three new writes from the beginning.

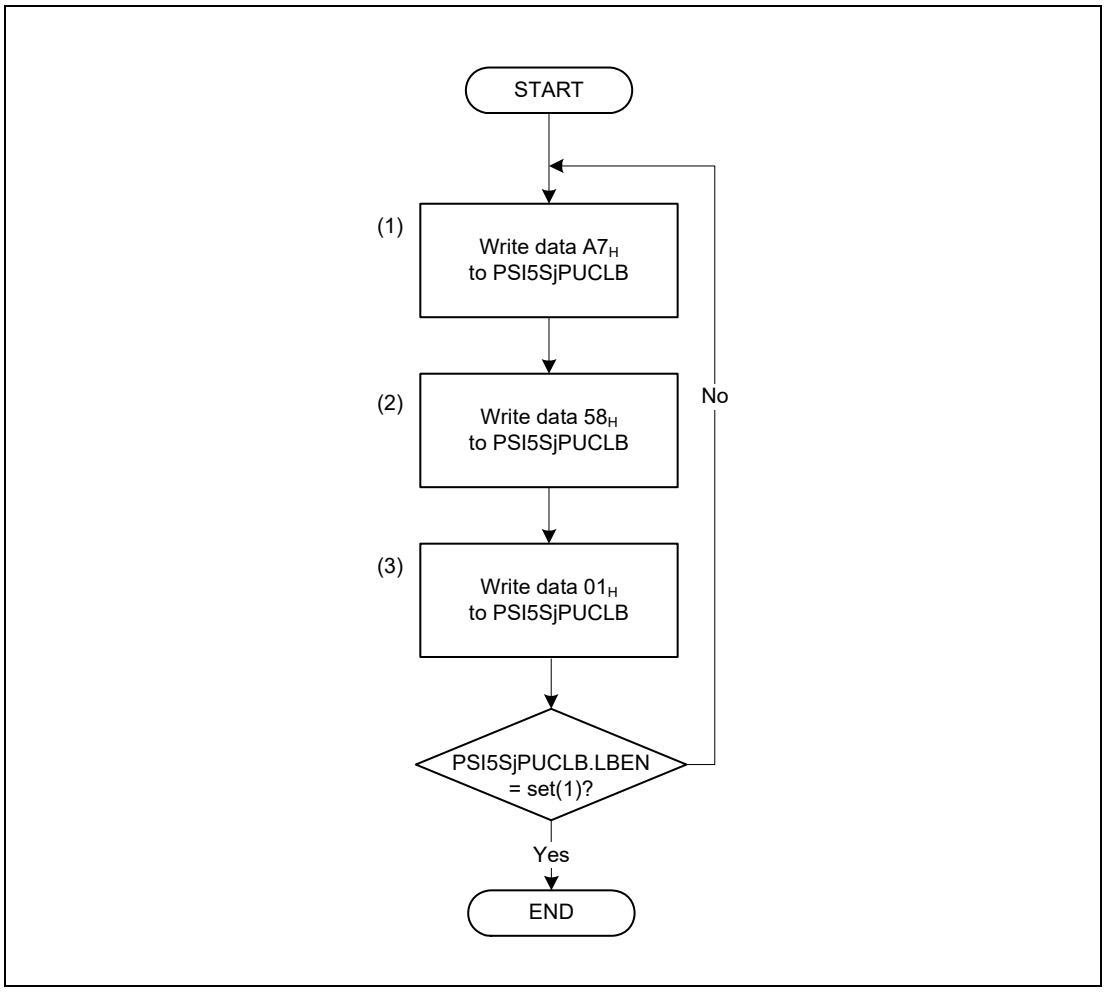


Figure 28.39 Loopback Test Setting Procedure

Then PSI5S mode setting (**Section 28.5.2, PSI5S mode Setting Procedure**) is performed.

The loopback test is performed under the following conditions:

- Communication mode is asynchronous mode.
- Sync commands are not transmitted (sync pulses are not generated).

28.5.11.2 Loopback Test Procedure

This section describes the procedure to perform a loopback test.

As an example, **Figure 28.40** below shows the procedure of the loopback test in which a PSI5 frame (five packets) on pseudo channel *n* (*n*: 1 to 7) are turned back to the reception block and the reception of the frame is checked by using a special function register (SFR).

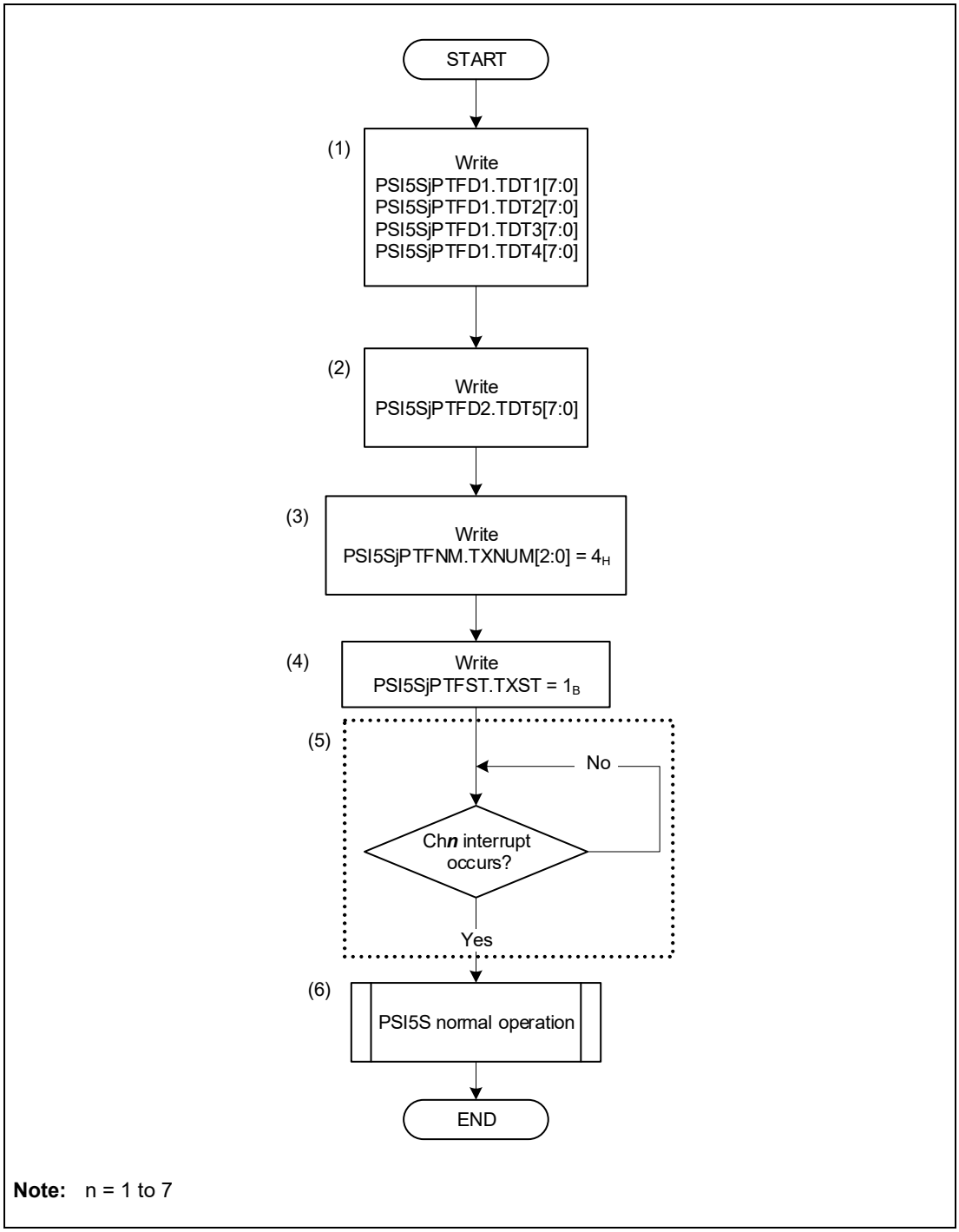


Figure 28.40 Loopback Test Procedure

(1) PSI5SjPTFD1: PSI5S Tx Frame Data1 register
In this register, set the first to fourth bytes of the data (PSI5 frame on pseudo channel *n* [*n*: 1 to 7]) to be transmitted continuously.

(2) PSI5SjPTFD2: PSI5S Tx Frame Data2 register
In this register, set the fifth byte of the data (PSI5 frame on pseudo channel *n* [*n*: 1 to 7]) to be transmitted continuously.

(3) PSI5SjPTFNM: PSI5S Tx Frame Number register
Specify 5-packet transmission (4_H) in this register.

(4) PSI5SjPTFST: PSI5S Tx Frame Start register
Write 1 to this register to start transmission.

(5) Wait until an interrupt outputs on channel *n* (*n*: 1 to 7).

(6) Subsequent processes follow the PSI5 frame reception procedure described in **Section 28.5.6, PSI5 Frame Reception Procedures**.

28.5.12 Configuration Mode Entering Procedure

Figure 28.41 below shows the configuration mode entering procedure.

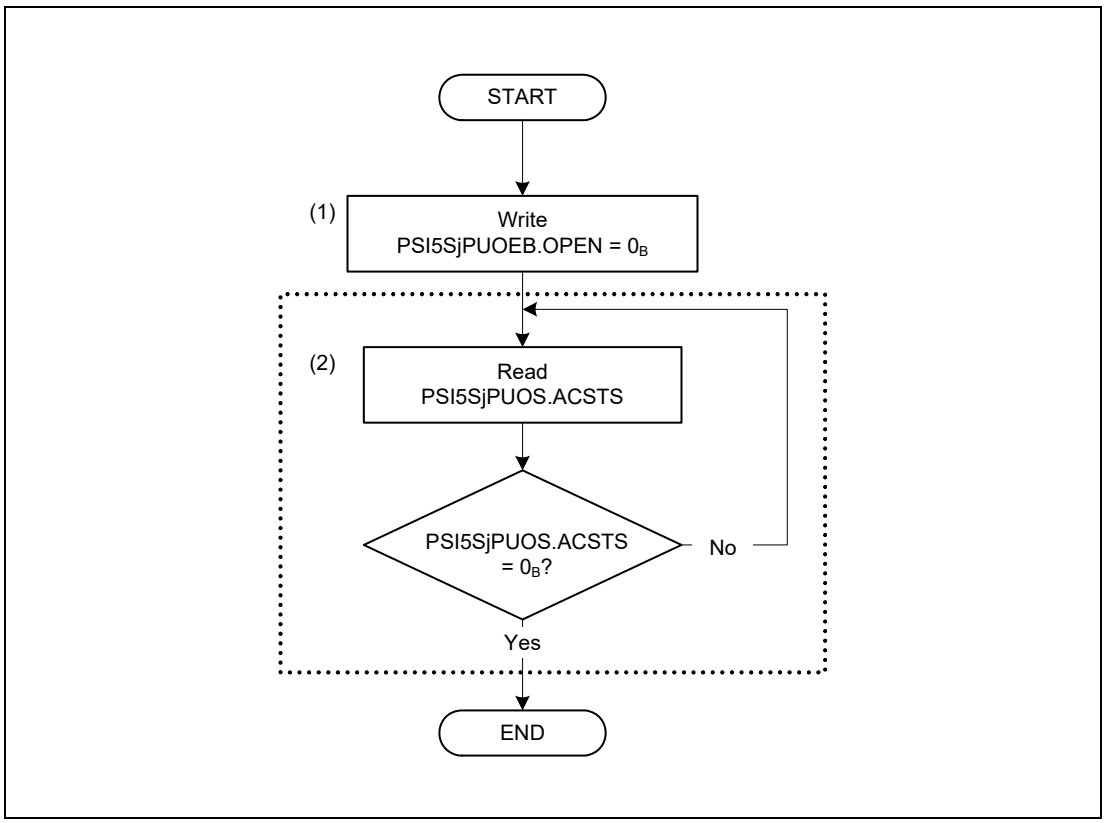


Figure 28.41 Configuration Mode Entering Procedure

(1) PSI5SjPUOEB: PSI5S/UART Operation Enable register
Stop the operation of PSI5S module.

(2) PSI5SjPUOS: PSI5S/UART Operation Status register
Wait until the Active Status bit change to 0.

The timing of return to the configuration mode, see **Section 28.4, Operation Modes**.

28.6 Operations

This chapter describes the functional configuration of PSI5S and PSI5S operations in individual operation modes.

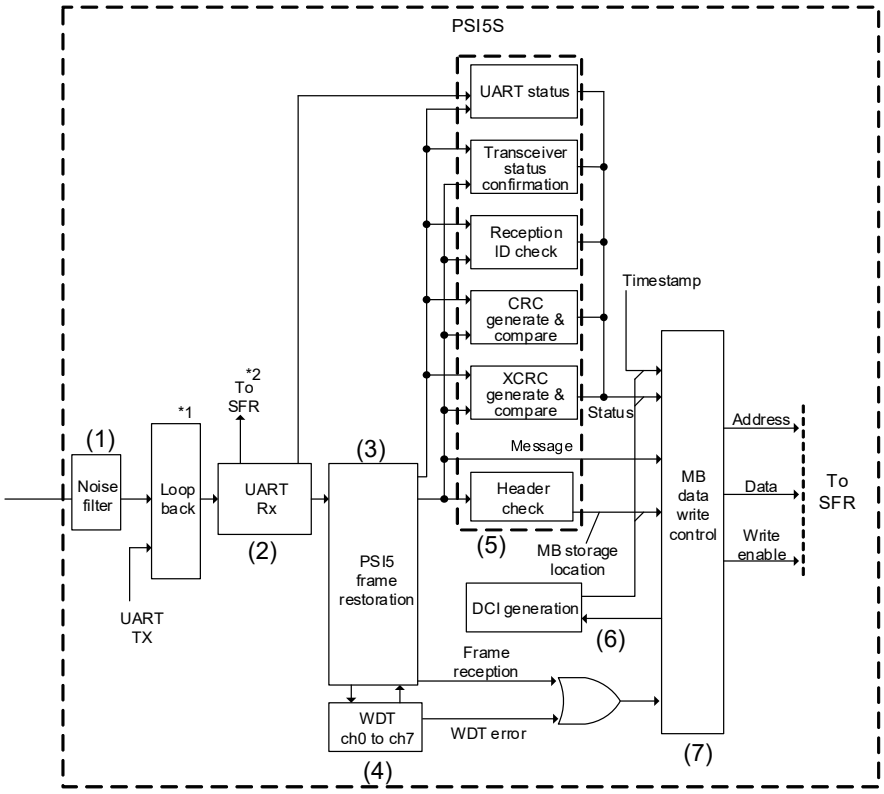
28.6.1 Functional Configuration

The sections that follow describe the functions of PSI5S.

28.6.1.1 PSI5S Reception Function

PSI5S receives sensor-to-ECU data, and stores the status (if a reception error is detected), received data, and timestamp in SFR.

Figure 28.42 below shows the block diagram for PSI5S reception.



(1) Take the majority of received signals to remove the noise.
(2) Perform UART reception, and output 1-byte data and reception error.
(3) Receive UART frames. If the idle time longer than the specified value has elapsed, the next frame is regarded as a header. After the header reception, check the number of frames by ChID and FrmID. If the specified number of frames has been received, it is restored as PSI5 frames.
(4) Monitor whether PSI5 frames of each channel receive a specified number of packets within the time. WDT errors are generated if the time expires without receiving.
(5) Generate the followings from the restored PSI5 frame data.
· After generating XCRC/CRC, the error determination status value, UART status flag, and status value from the header
· Message data
· MB storage location address from ChID and FrmID in the reception header
(6) Generate the 4-bit DCI value and update it after MB data for 3 addresses is written. MB data is configured by combining the DCI value with status/message/timestamp data.
(7) When the WDT error or frame is received, it is stored to the MB storage location in the order of status, message, and timestamp.

Note 1. When loopback is set, UART transmit data is wrapped.
Note 2. When data is received in UART mode, the received data and status are output to SFR.

Figure 28.42 Block Diagram for PSI5S Reception

For the timing of reception, see **Figure 28.74**.

The following figure shows the flow of data reception.

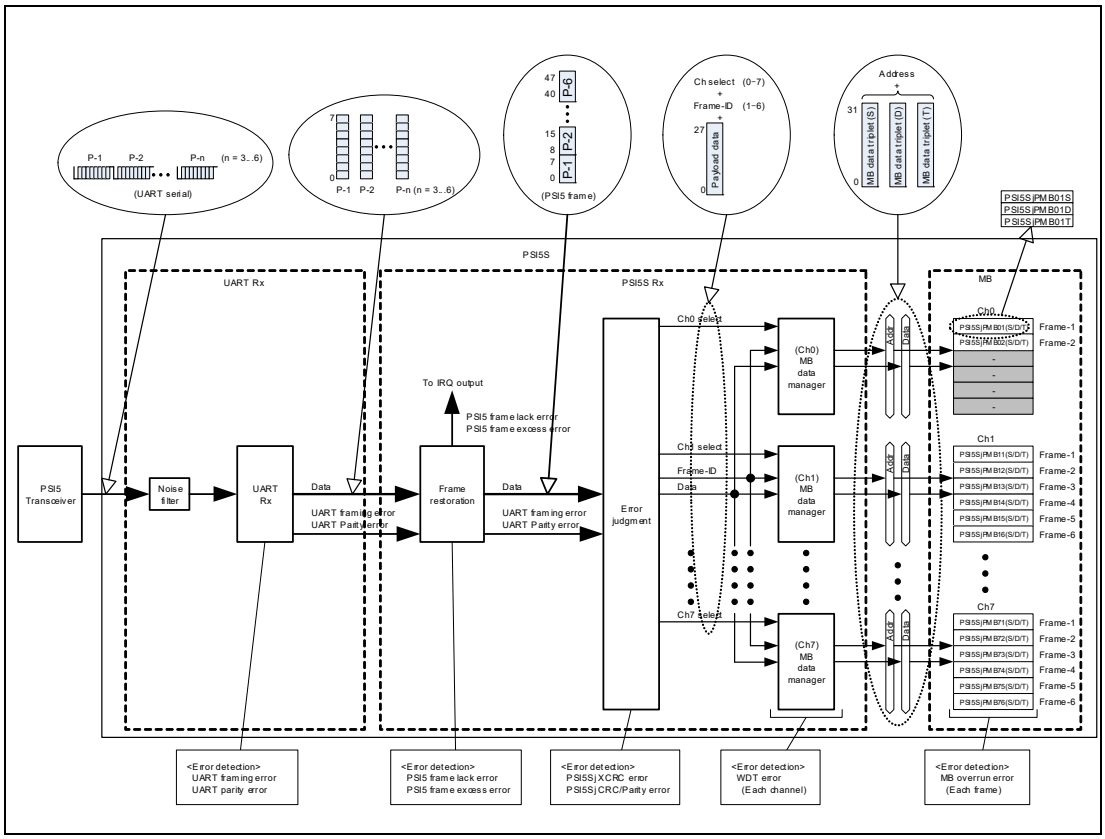


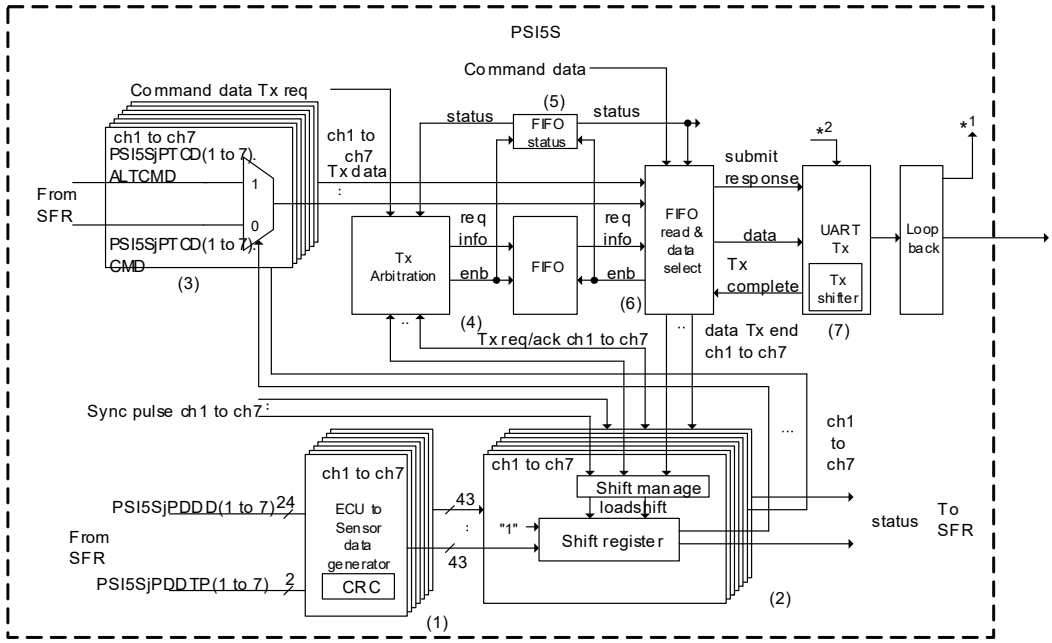
Figure 28.43 PSI5S Reception Data Flow

28.6.1.2 PSI5S Transmission Function

PSI5S generates ECU-to-sensor data, and transmits a Sync command when it is triggered by a synchronization pulse on individual channels.

PSI5S also has a function to transmit the data set by the CPU via SFR.

The following figure shows the block diagram for PSI5S transmission.



(1) After transmission data is updated, calculate the 6-bit or 3-bit CRC value according to the PSI5 type.
After the calculation, generate ECU to sensor data by the start bit, data, and CRC value.
(The length of data differs according to the PSI5 type.)
(2) After ECU to sensor data is generated, load the data to the shift register.
Output data in 1-bit unit by sync pulse with starting from the start bit.
Shift register is shifted by data Tx end, and continue to output after the load of status flag to the shift of last data.
(3) Select PSI5SjPTCD.ALTCMD or CMD by the value output in (2) above. After the selection, generate 1-byte data by combining it with ChID.
(4) To carry out arbitration in response to a request from the CPU or Sync pulse. Request information sequentially to the transmission FIFO.
(5) Generate status (full/empty) by write/read pointer to the FIFO.
(6) If request information of 1 or more is in the FIFO, read it.
Output the transmission start and select data signal to UART transmission unit.
(7) Generate the UART frame data by the transmission start signal, and perform UART transmission.
When the stop bit is transmitted, output the transmission complete signal.

Note 1. When loopback is set, UART transmission data is wrapped and not output externally.

Note 2. When data is updated by CPU write in UART mode, perform UART transmission.

Figure 28.44 Block Diagram for PSI5S Transmission

For the timing of the transmission, see **Figure 28.68**.

Transmit requests between command data (**Section 28.6.2.3, Normal Transmission (Command Data)**) and ECU-to-sensor data (**Section 28.6.2.4, Normal Transmission (ECU-to-Sensor Data)**) are arbitrated, and then UART starts transmission.

Figure 28.45 below shows the block configuration related to the transmit request arbitration function.

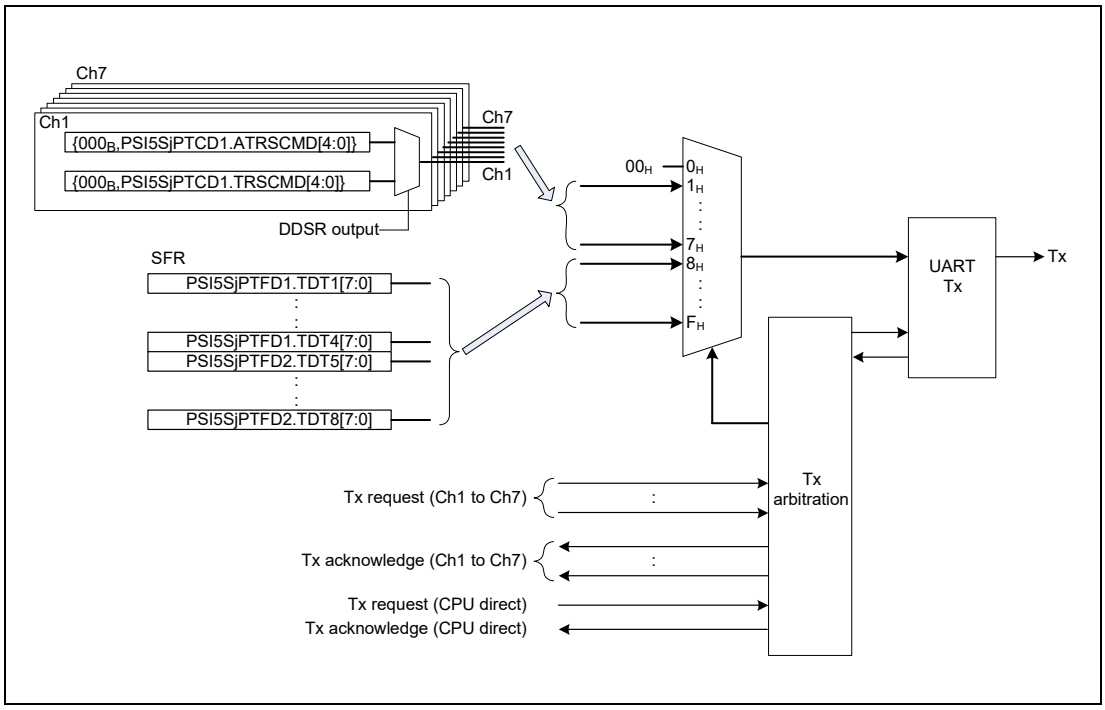


Figure 28.45 Block Diagram for Transmit Request Arbitration

28.6.1.3 Timestamp Generation Function

Based on the timing of clock input from the GTM or the timing of a divided clock (1 to 1023 µs in 1-µs steps) generated from PCLK, a timestamp counter runs to measure time. PSI5S has two timestamp counters for timestamps A and B, and the timestamp counters can be enabled and cleared together or separately.

In synchronous mode of Ch1 to Ch7, the timestamp value is captured at the timing of PSI5 frame header reception or the synchronization pulse. In asynchronous mode of Ch1 to Ch7, the timestamp value is captured in the timing of PSI5 frame header reception. In asynchronous mode of Ch0, the timestamp value is captured in the timing of PSI5 frame header reception. (Ch0 is always asynchronous mode.) The timestamp value of Ch0 is also captured at the timing of a WDT error occurs in any of the channels.

Timestamp capture values are available for channels 0 to 7. When PSI5 frame reception ends, one of the timestamp capture values is selected according to the channel ID in the header, and used as a timestamp value. If the received frame is an invalid frame, the timestamp capture value for channel 0 is selected. Determination of invalid frames follows the specifications listed in **Table 28.94** and **Table 28.96**. If a WDT error occurs, the timestamp capture value of Ch0 is used as the timestamp value.

Note that timestamp capture values are all zeros when the timestamp capture enable bit (PSI5SjPRCF1n.TSEN) is 0 (disable). Also, the timestamp capture values can be cleared by the clear bit (PSI5SjPTCDCn).

Figure 28.46 below shows the block diagram for the timestamp generation function.

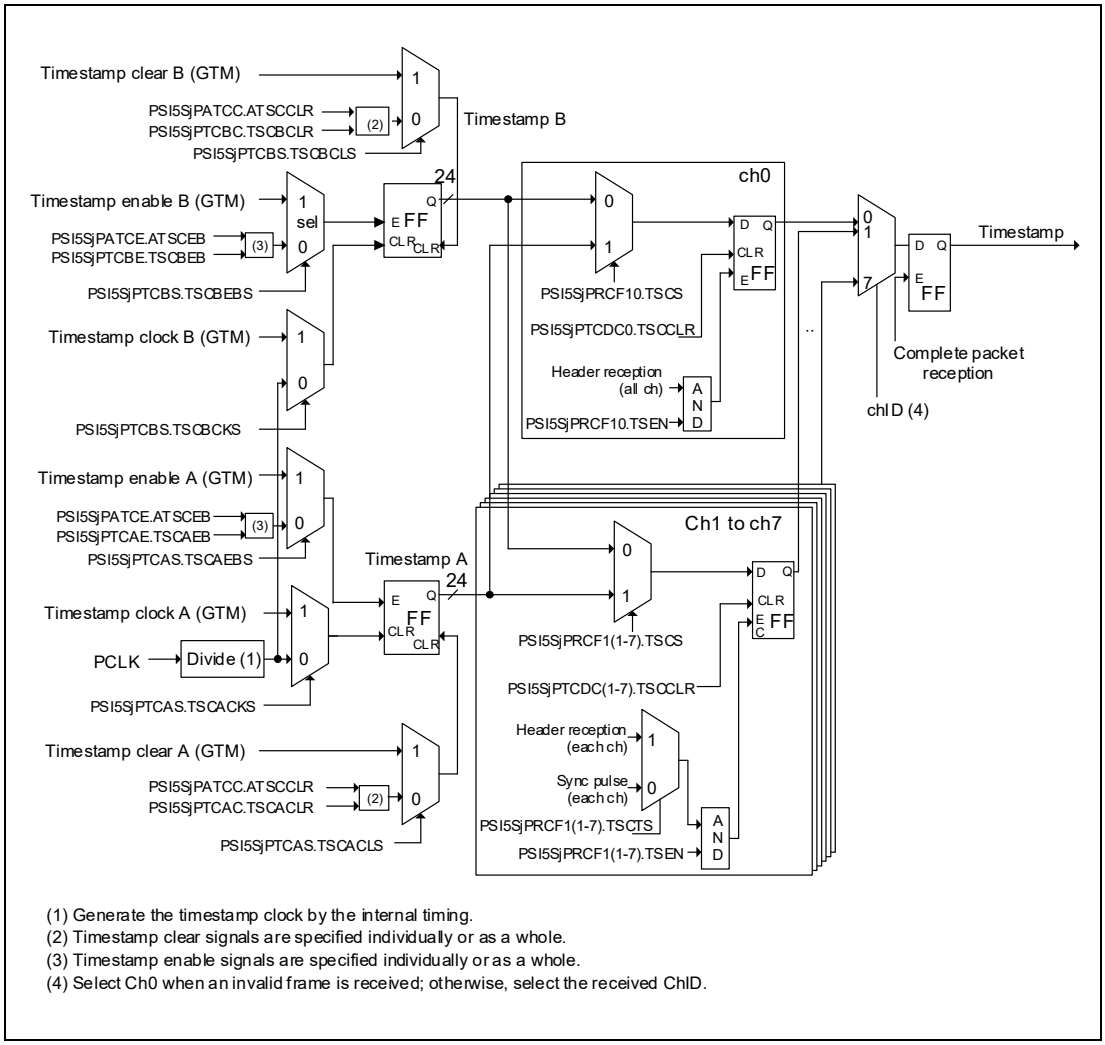


Figure 28.46 Block Diagram for Timestamp Generation

Figure 28.47 below shows an operation in which a timestamp capture value is fetched when the header of a PSI5 frame is received, and a timestamp value is generated when PSI5 frame reception ends.

The same operation can be performed in both synchronous and asynchronous modes.

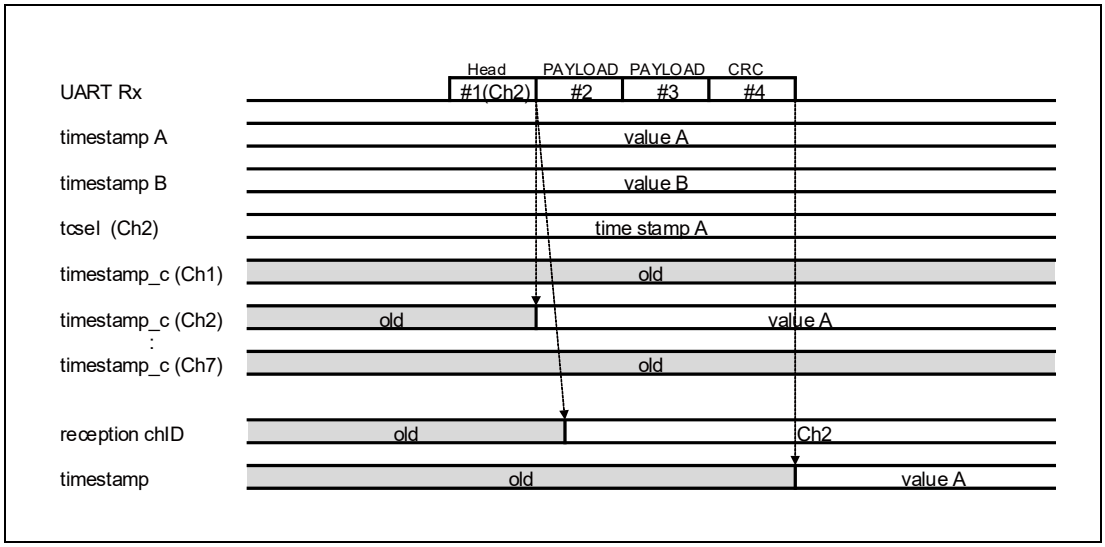
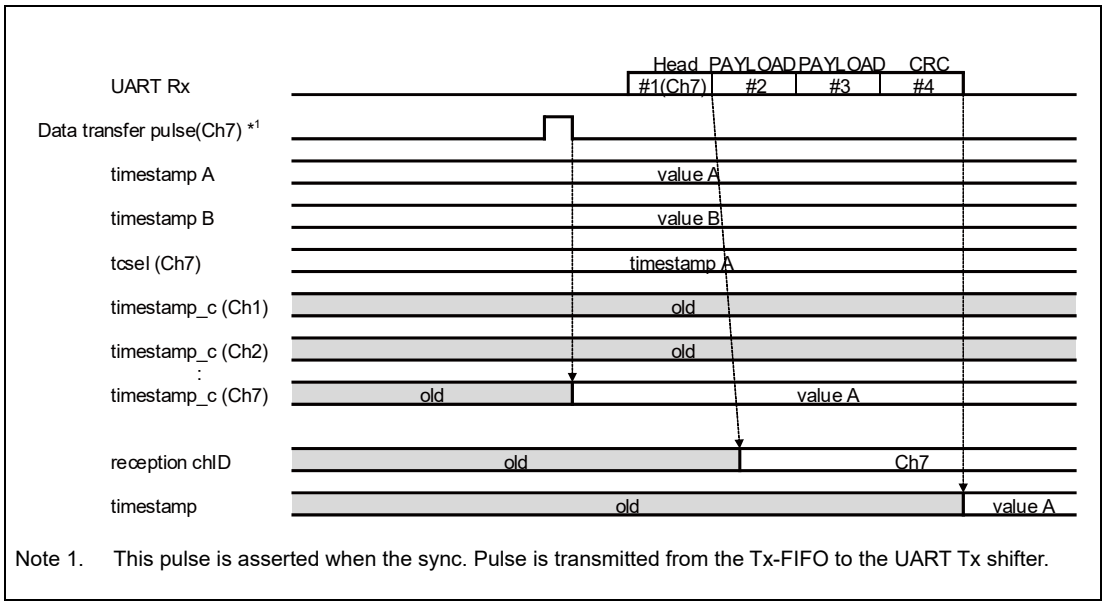


Figure 28.47 Timestamp Generation Operation (Timestamp Capture Triggered by Header Reception)

Figure 28.48 below shows an operation in which a timestamp capture value is fetched when a synchronization pulse outputs, and a timestamp value is generated when PSI5 frame reception ends.

This operation can be performed only in synchronous mode.



Note 1. This pulse is asserted when the sync. Pulse is transmitted from the Tx-FIFO to the UART Tx shifter.

Figure 28.48 Timestamp Generation Operation (Timestamp Capturing Triggered by Synchronization Command Transmission)

Figure 28.49 below shows an operation in which a timestamp capture value is fetched when a WDT error occurs, and a timestamp value is generated immediately.

The same operation can be performed in both synchronous and asynchronous modes.

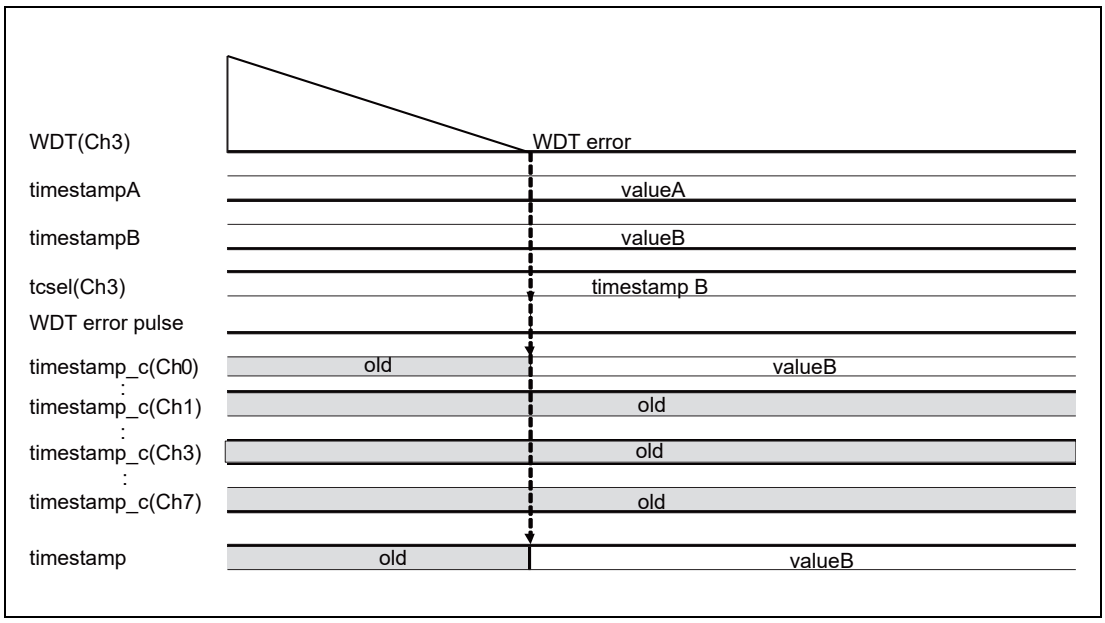


Figure 28.49 Timestamp Generation Operation (Capturing Triggered by WDT Error)

28.6.1.4 Synchronization Pulse Generation Function

The timing of the synchronization pulse for channels 0 to 7 is selected from the rising edge of the synchronization trigger signal for each channel input from the GTM or the timing of a divided clock (1 to 1024 µs in 1-µs step) generated from PCLK.

The divided clock timing generation function operates when the function is selected in SFR and its operation is enabled (PUOEB.OPEN = 0x1). With other settings, counters stop with a counter value of 0.

Interval of the synchronization pulse for each channel should be longer than time of the number of packets to be received. Otherwise, operation of the PSI5S module will be unpredictable.

The following figure shows a block diagram for the synchronization pulse generation circuit:

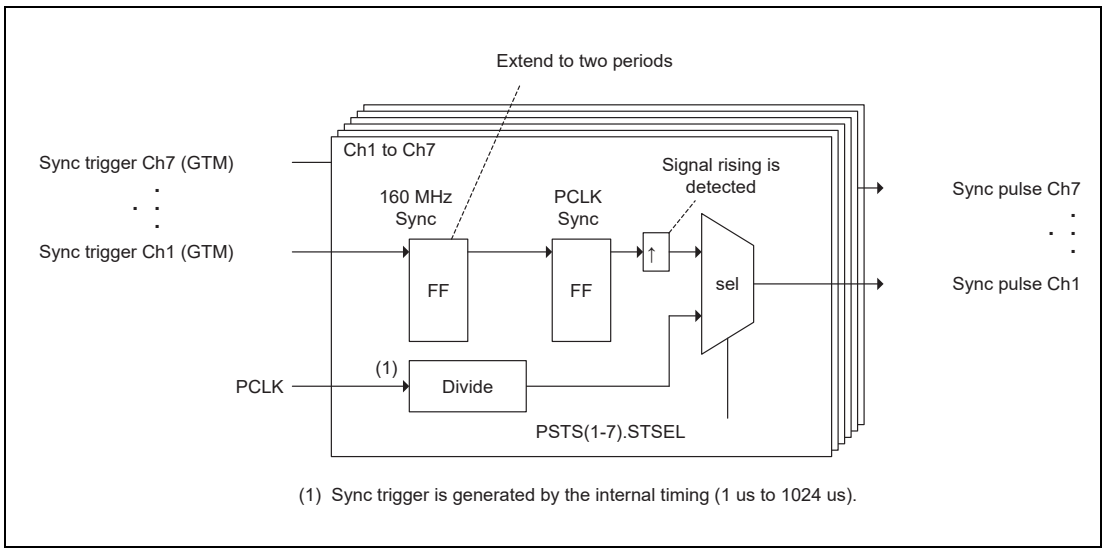


Figure 28.50 Block Diagram for Synchronization Pulse Generation

Figure 28.51 below shows an outline of the operation in which synchronization pulses are generated.

Synchronization pulses are generated by synchronizing the Sync trigger signal from the GTM with PCLK.

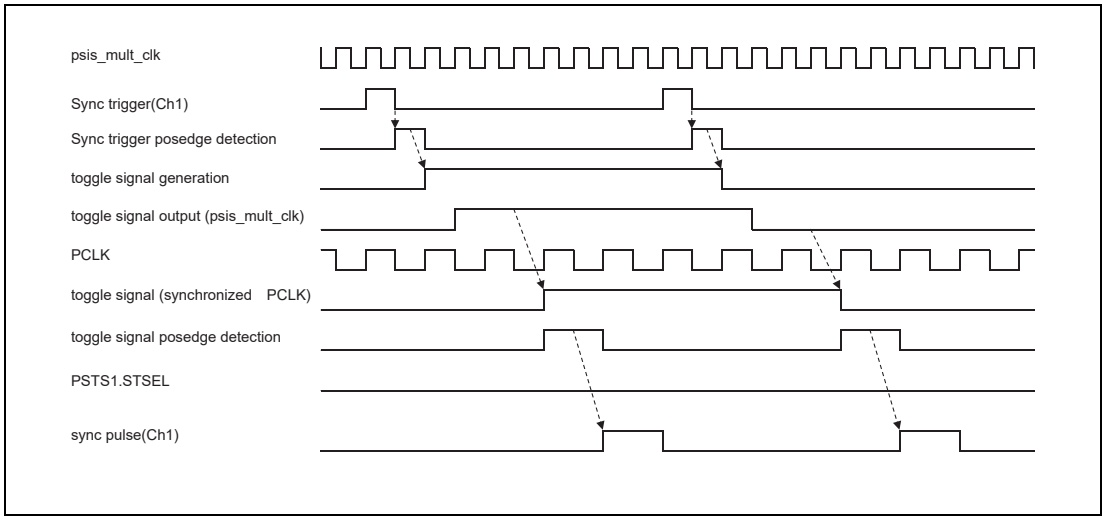


Figure 28.51 Synchronization Pulse Generation Operation (Input from the GTM)

The synchronization pulses (sync pulse signals) are also generated by a frequency divider.

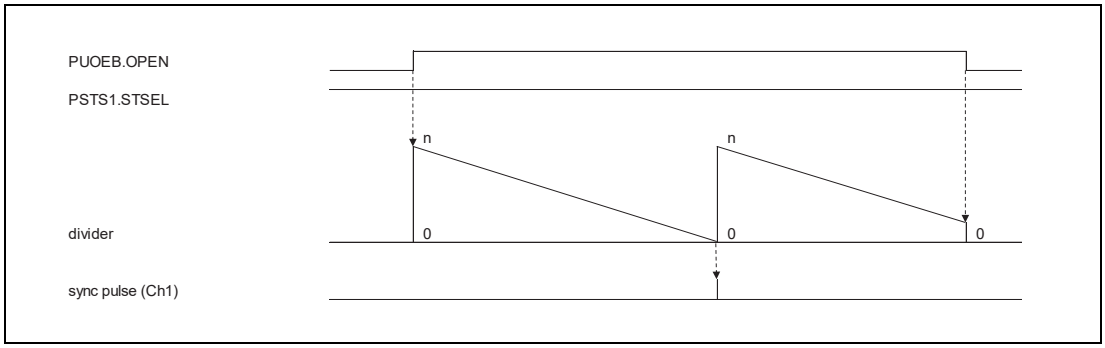


Figure 28.52 Synchronization Pulse Generation Operation (Internal Timing)

28.6.2 Operations in PSI5S mode

PSI5S performs UART reception of four to six frames when the operation mode is set to PSI5S mode (PSI5SjPUOMD.OPMD = 1_H) and the module operation is enabled (PSI5SjPUOEB.OPEN = 1_H).

PSI5S restores PSI5 frame data with corresponding functions shown in **Figure 28.42**.

After restoring the data, PSI5S checks for reception errors, and then stores data (such as the status value generated from the received frame data, payload value of the received frame data, and the timestamp value generated as described in **Section 28.6.1.3, Timestamp Generation Function**) as mailbox data in SFR.

The following figure shows the flow of reception.

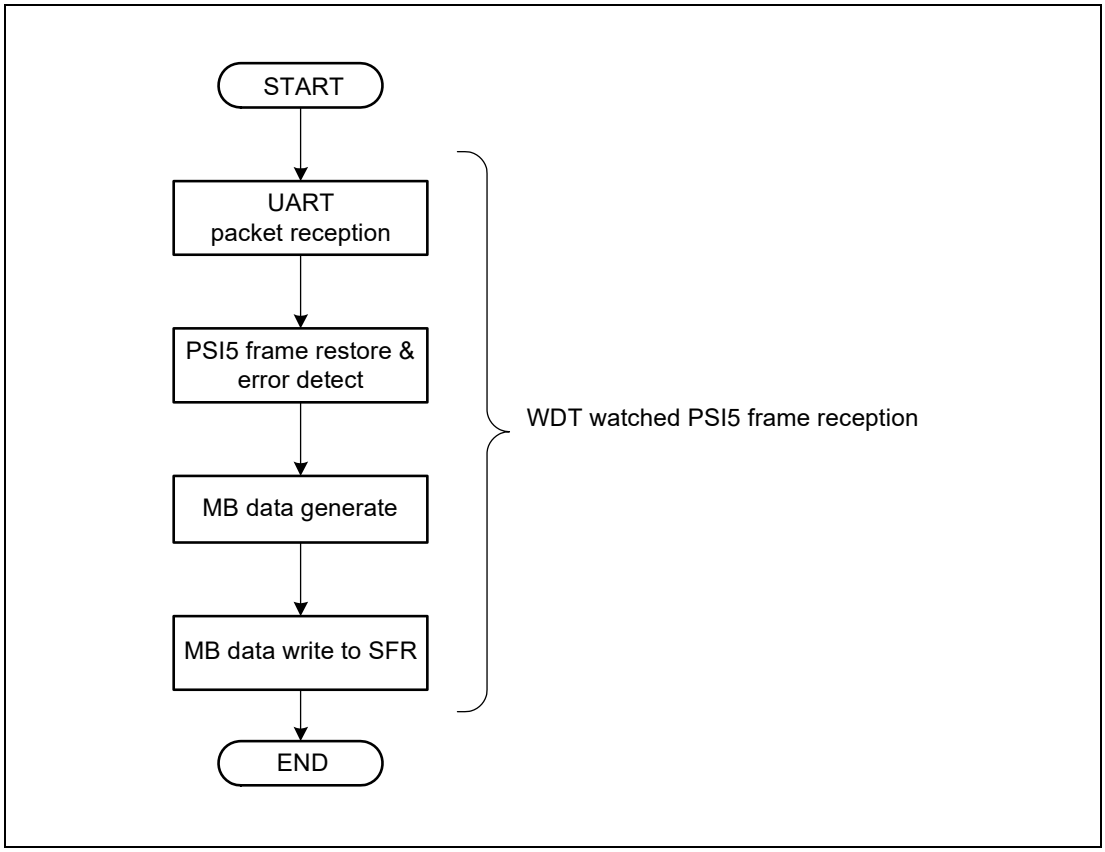


Figure 28.53 Flow of Reception

28.6.2.1 Normal Reception (in Synchronous Mode)

PSI5S performs reception according to the flow shown in **Figure 28.53**.

In synchronous mode, PSI5S starts reception with the synchronization pulse for each channel used as the trigger, and receives a specified number of PSI5 frames.

A WDT installed for each channel is used to monitor reception to determine whether the specified number of PSI5 frames are received within a specified time.

(1) Receiving UART Frames

PSI5S receives UART frames, and restores PSI5 frame data.

For the UART specifications, see **Section 28.2.1, Functional Overview**.

UART receive data is filtered by a noise filter (filtering of three prescaler cycles of data signal based on a majority decision) according to the settings.

Sampling enable generate counter and Sampling counter is reset and UART reception starts when the start bit is detected at a sampling point that starting from the falling edge detection after noise filtering.

NOTE

A sampling point is different in oversample number (odd or even).

Figure 28.54 and **Figure 28.55** below show examples of UART reception.

The following example assumes that the oversample number is set to 5_H.

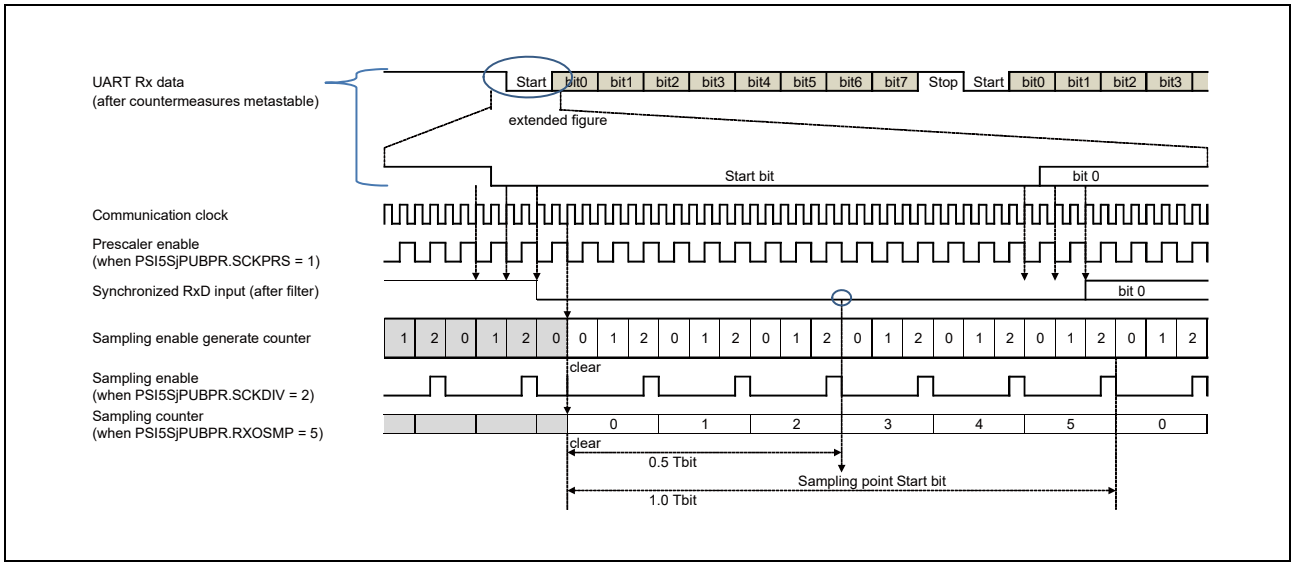


Figure 28.54 UART Frame Reception Start (PSI5SjUBPR.RXOSMP = 5 [ODD])

The following example assumes that the oversample number is set to 4_H.

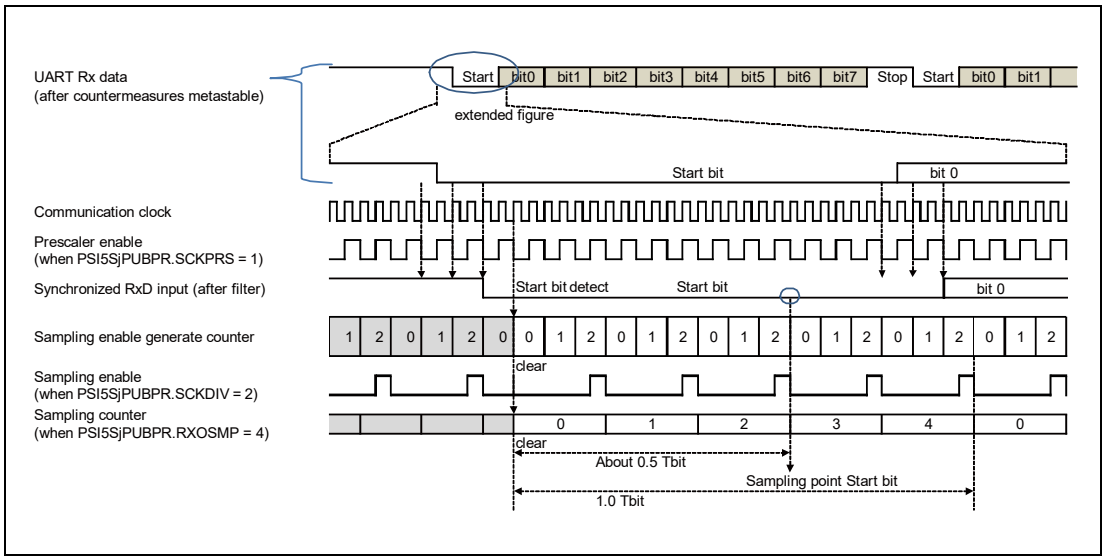


Figure 28.55 UART Frame Reception Start (PSI5SjUBPR.RXOSMP = 4 [EVEN])

Figure 28.56 below shows an example of an operation in which UART frames are received continuously.

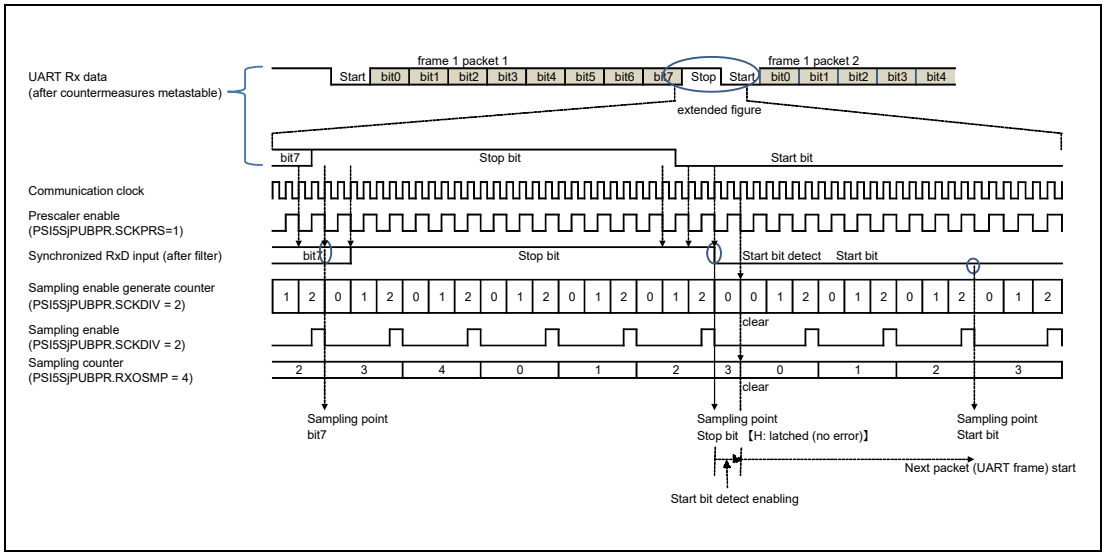


Figure 28.56 UART Frame Reception End (Frame Continue) (PSI5SjPRCF10.PFRMIDLE = 0)

Figure 28.57 below shows an example of an operation in which the last packet frame is received.

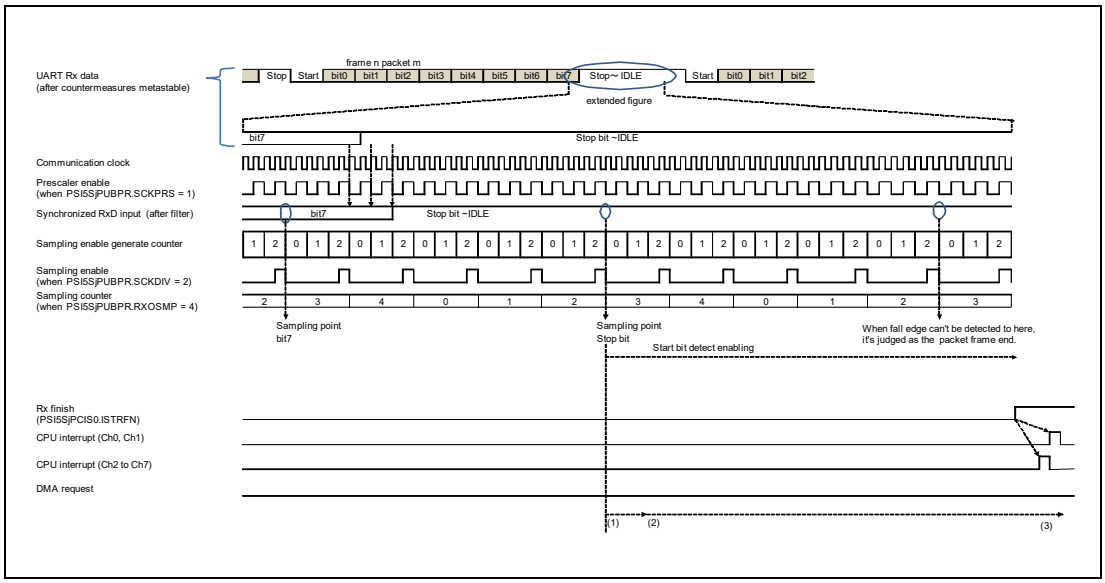


Figure 28.57 UART Frame Reception End (Packet Frame End) (PSI5SjPRCF10.PFRMIDLE = 0)

(1) The stop bit of the last packet is detected.

(2) The asynchronous transfer processing based on PCLK and the communication clock, causes a delay of several clock cycles.

(3) After 50-60 cycles of PCLK (for restoring PSI5 frame data, determining errors and storing mailbox data), the flag indicating the end of reception is set (PSI5SjPCIS0.ISTRFN = 1) and an interrupt is output.

Figure 28.58 below shows an example of an operation in which the end of packet frame is detected.

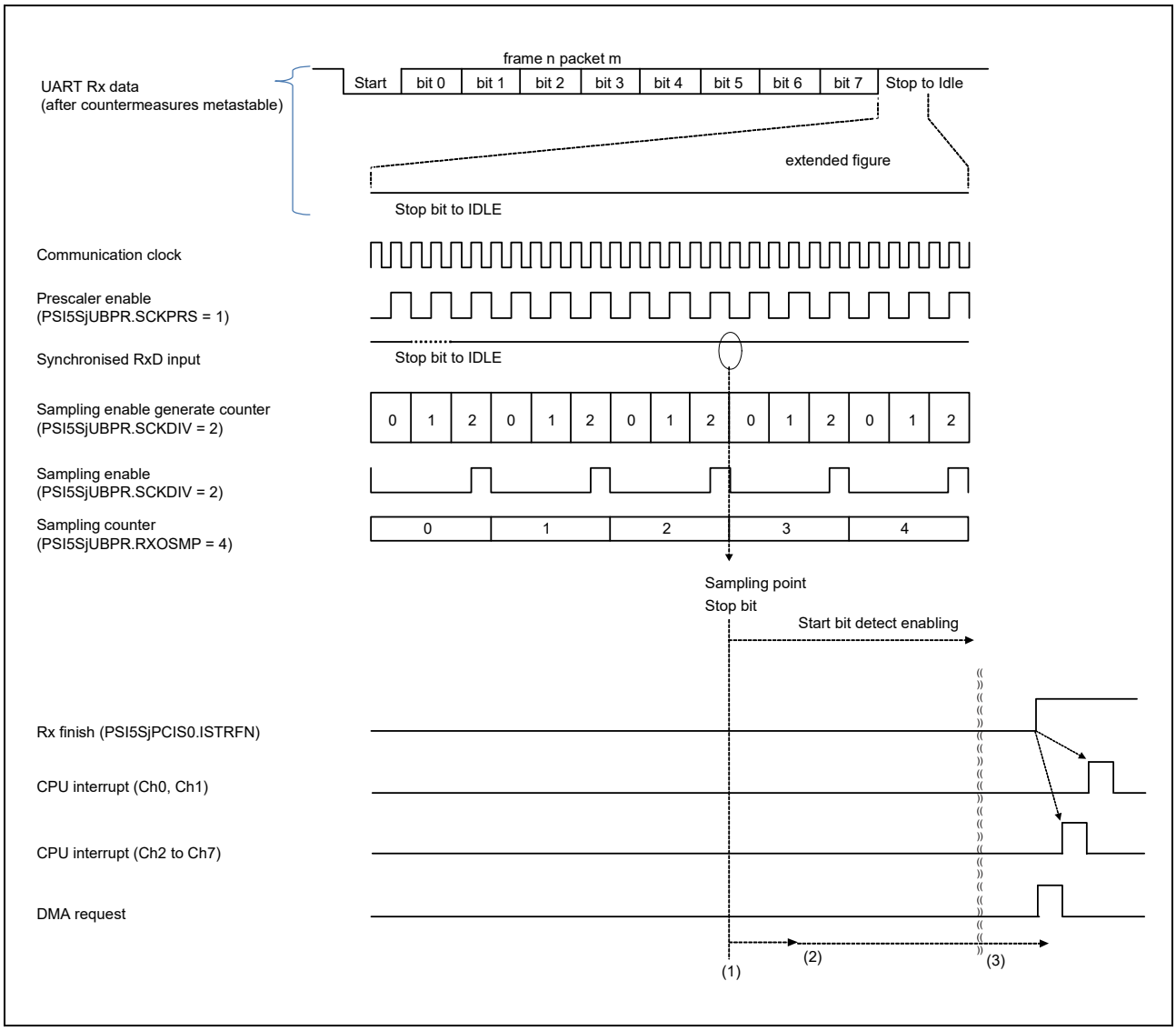


Figure 28.58 UART Frame Reception (Channel End) (PSI5SjPRCF10.PFRMIDLE = 0)

(1) When the stop bit of the last packet is detected, the flag indicating the end of reception is set (PSI5SjPCIS0.ISTRFN = 1).
The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

(2) After 50-60 cycle of PCLK (for Restoring PSI5 frame data, determining errors and storing mailbox data), the flag indicating the end of reception is set (PSI5SjPCIS0.ISTRFN = 1) and an interrupt outputs.
When reception of all packet frames in a channel ends, an interrupt and a DMA request are output.

PSI5 frame data consists of a header, payload, and XCRC bits (**Figure 28.3**).

UART frames are input successively, and the maximum interval to the next packet frame is specified in the PSI5SjPRCF10. PFRMIDLE[3:0] bits.

If the actual interval between frames exceeds the specified value, the UART frame that is received next is treated as a header.

After the header of a frame is received, the number of received packets in the frame (PSI5SjPRCF1(0-7).F(1-6)PKT) is referenced according to the channel ID and frame ID in the header. The number of packets to be received is determined from the number of already received packets. When FrmID is not 0_H in ch0, the number of packets to be received is 6_H (maximum value). When FrmID is an illegal value (6_H or 7_H) in ch1 to ch7, the number of packets to be received is 6_H (maximum value).

If excess or deficiency of the number of packets to be received is detected, reception error occurs. For details, see **Section 28.6.2.7, Abnormal Reception (Reception Error)**.

If too few packets are received, the received packets are not stored.

If a WDT error occurs, the received packets at the time of the WDT error occurrence are stored.

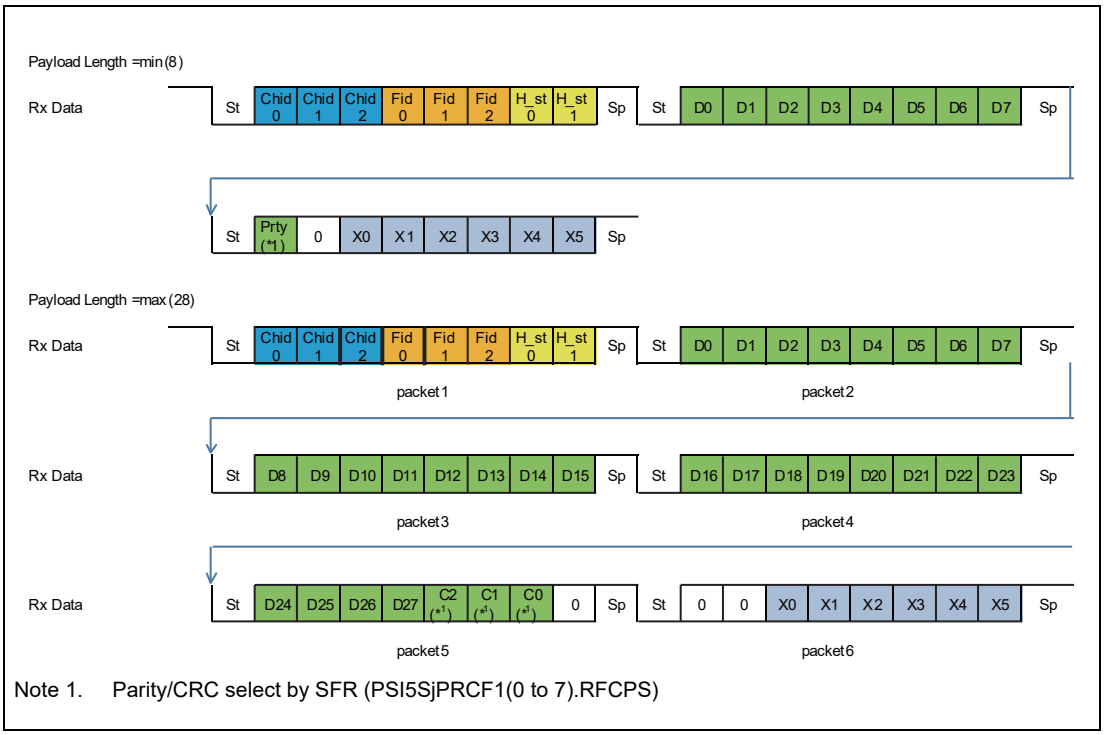
If too many packets are received, the received packets over the specified number are not stored.

Table 28.93 below shows the correspondence between the settings of PSI5SjPRCF1(0-7).F(1-6)PKT and the number of packets to be received.

Table 28.93 PSI5 Frame Reception Packet Number

Value of PSI5SjPRCF1(0 to 7).F(1 to 6)PKT	Specification
0_H	Mask the corresponding frame. (Receive data is not stored in SFR)
1_H, 2_H	Setting prohibited
3_H to 6_H	Packet number is PSI5SjPRCF1(0-7).F(1-6)PKT value (3_H to 6_H)
7_H	Setting prohibited

The following figure shows the data format of a PSI5 frame when the payload bit length is min/max.



Note 1. Parity/CRC select by SFR (PSI5SjPRCF1(0 to 7).RFCPS)

Figure 28.59 PSI5 Frame Data Format (Payload = 8 Bits (min.) to 28 Bits (max.))

(2) Restoring PSI5 Frame Data and Determining Errors

After receiving all the packets to be received, PSI5S finishes packet frame reception and restores the PSI5 frame data.

For how to determine errors when restoring the PSI5 frame data, see **Section 28.6.2.7, Abnormal Reception (Reception Error)**.

(3) Generating MailBox Storage Data

After determining errors in PSI5 frame data, PSI5S generates three types of mailbox storage data (triplet).

The three types of mailbox storage data contain the status, data, and timestamp, respectively, and have the same value of data consistency indicator (DCI) in the upper four bits.

The DCI value installed for each channel is generated by a 4-bit up-counter. The up-counter value is incremented each time PSI5 data is restored. PSI5S is initialized by an asynchronous reset or software reset.

The following figure shows the timing of generating the DCI value.

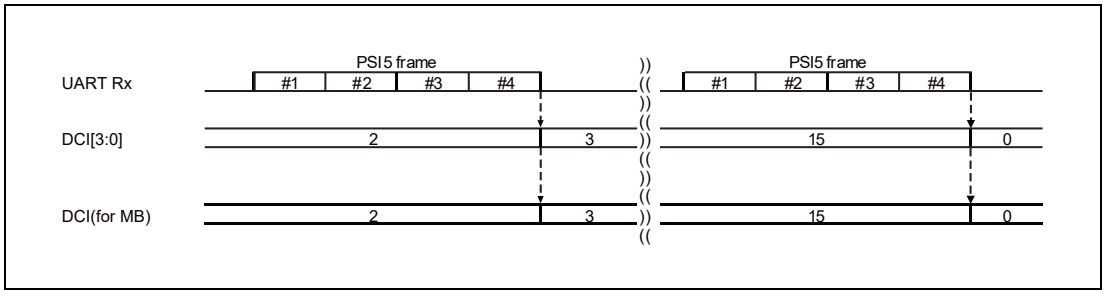


Figure 28.60 DCI Generation Timing

Figure 28.61 below shows the bit format of the mailbox data (triplet).

For details, see **Section 28.3.12, Ch 0 Frm m MB Data (m: 1, 2)** or **Section 28.3.13, Ch n Frm m MB Data (n: 1 to 7) (m: 1 to 6)**.

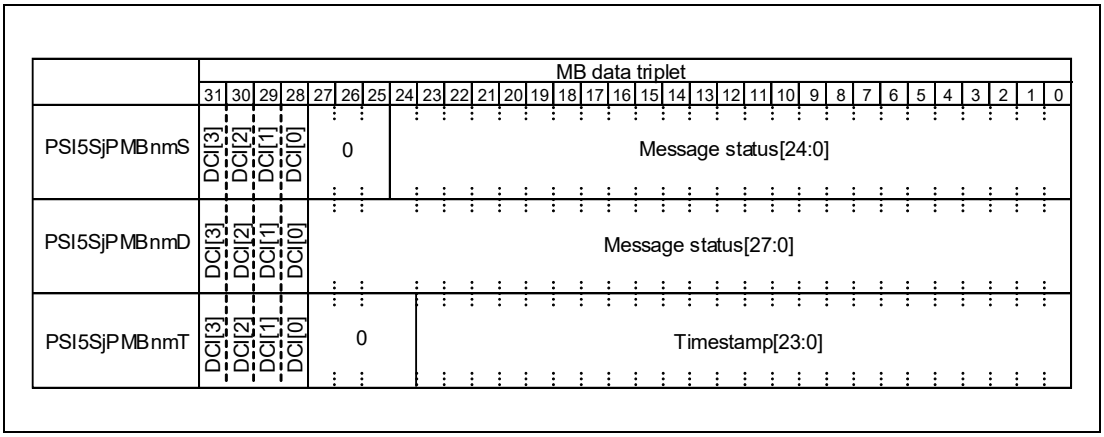


Figure 28.61 MB Data Triplet

(4) Writing MailBox Data to SFR

After generating mailbox storage data, PSI5S obtains the storage destination of the mailbox data in SFR by using the channel ID and frame ID in the header.

The conditions for deciding the storage destination are as follows.

Channel 0 is a virtual channel. If an XCRC error or UART reception error occurs, the received frame is determined to be an invalid frame, and the mailbox data is stored in frames on channel 0. The mailbox data is also stored in frames on channel 0 when the frame ID is invalid or a WDT error occurs.

Table 28.94 PSI5 Frame MB Target Generate (Sync)

XCRC error, UART parity error, UART framing error	FrmID	MB target
No	0≤FrmID≤5	Ch: Receive ChID Frm: Receive FrmID
	6, 7	Ch: 0
Yes	Not related	Frm: 2

The storage data and the storage destination when a WDT error occurs in synchronous mode are as shown in **Table 28.99**.

Table 28.95 below lists the 1st addresses of the frames to write mailbox data that is generated from the received channel ID and frame ID.

Table 28.95 MB write 1st Address Generate (by Frame ID)

Frm ID	Frame	Write 1st address							
		Ch0	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7
0	Frame1	500_H	548_H	590_H	$5D8_H$	620_H	668_H	$6B0_H$	$6F8_H$
1	Frame2	$50C_H$	554_H	$59C_H$	$5E4_H$	$62C_H$	674_H	$6BC_H$	704_H
2	Frame3	$50C_H$	560_H	$5A8_H$	$5F0_H$	638_H	680_H	$6C8_H$	710_H
3	Frame4	$50C_H$	$56C_H$	$5B4_H$	$5FC_H$	644_H	$68C_H$	$6D4_H$	$71C_H$
4	Frame5	$50C_H$	578_H	$5C0_H$	608_H	650_H	698_H	$6E0_H$	728_H
5	Frame6	$50C_H$	584_H	$5CC_H$	614_H	$65C_H$	$6A4_H$	$6EC_H$	734_H
6	Error frame	$50C_H$							
7	Error frame	$50C_H$							

The mailbox data storage addresses must be in the range from 500_H to $73C_H$ shown in the address map (**Table 28.12**).

When mailbox data is stored in frame 3 on channel 1, the 1st address of the storage destination is 560_H.

After deciding the storage destination, PSI5S stores the status in the area beginning with the 1st address, data in the area beginning with the address “1st address + 4,” and timestamp in the area beginning with the address “1st address + 8.”

(5) Monitoring PSI5 Frame Reception with the WDT

With the WDT installed for each channel, PSI5S monitors reception of the specified number of PSI5 frames to be received.

When the synchronization pulse is transmitted to the Tx shifter register after synchronization pulse input from GTM, the WDT counter fetches a specified expiration value and starts counting. When the specified number of frames are received or the WDT counter value becomes 0, the WDT counter stops, and waits for the rising edge of the next synchronization pulse. At the rising edge of the next synchronization pulse, the WDT counter fetches a specified expiration value, and starts counting. If the WDT counter value becomes 0, a WDT error is assumed. For details, see **Section 28.6.2.8, Abnormal Reception (WDT Error)**.

The WDT expiration value (PSI5SjPWDEVn.WDTEX) should be shorter than the interval time of synchronization pulse.

NOTE

Frame reception time < WDT expiration time < {(the interval time of synchronization pulse) - (maximum arbitration delay time)}
(maximum arbitration delay time) = bit time * 154 + 50 * (PCLK period)

The following figure shows the WDT operation (without WDT error).

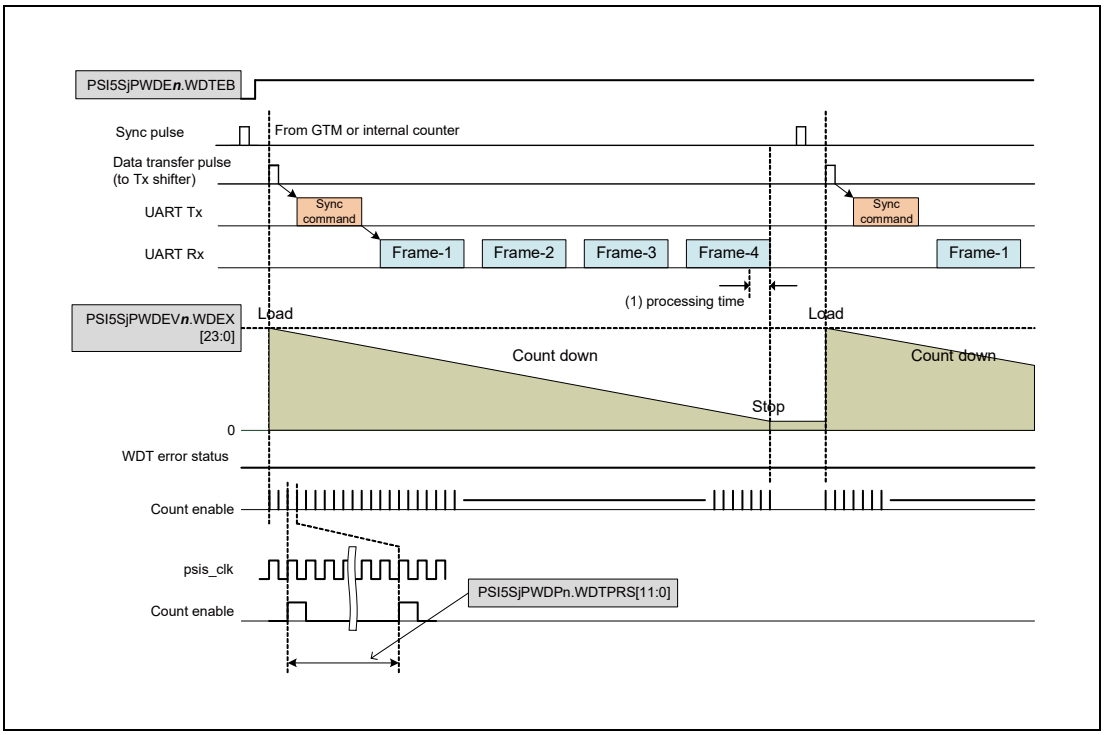


Figure 28.62 WDT Operation Timing (Sync Mode) (without WDT Error)

NOTE

The WDT counter stops in 50 cycles after a stop bit of the last frame is received. A WDT error occurs in a case that the WDT counter expires within the 50 clock cycles even if the stop bit of the last frame is received.

The number of packets to be received on each channel is obtained as follows:

- When the value set in PSI5SjPRCF1(0-7).F(1-6)PKT is not 0_H, the frame is monitored.
- When the value set in PSI5SjPRCF1(0-7).F(1-6)PKT is 0_H, the frame is not monitored.

28.6.2.2 Normal Reception (in Asynchronous Mode)

Unlike synchronous communication, asynchronous communication does not transfer PSI5 frames in synchronization with synchronizing pulses. In asynchronous mode, PSI5S receives PSI5 frames from the transceiver according to the packets transmitted from the sensor.

(1) Receiving UART frames

This operation is the same as that described in **Section 28.6.2.1(1), Receiving UART Frames**.

(2) Restoring PSI5 Frame data and determining Errors

This operation is the same as that described in **Section 28.6.2.1(2), Restoring PSI5 Frame Data and Determining Errors**.

(3) Generating MailBox storage data

This operation is the same as that described in **Section 28.6.2.1(3), Generating MailBox Storage Data**.

(4) Writing MailBox data to SFR

This operation is different from that described in **Section 28.6.2.1(4), Writing MailBox Data to SFR** in the following points.

After generating mailbox storage data, PSI5S obtains the storage destination of the mailbox data by using the channel ID in the header and the frame ID counter*1 installed for each channel.

In asynchronous mode, the frame ID is fixed to 0. Therefore, the frame ID is replaced with the frame ID counter in order to always store the data of the six latest frames.

Note 1. The frame ID counter is a hexadecimal counter that operates only in asynchronous mode and is incremented each time mailbox data is stored in SFR.

The following figure shows an outline of the counter operation.

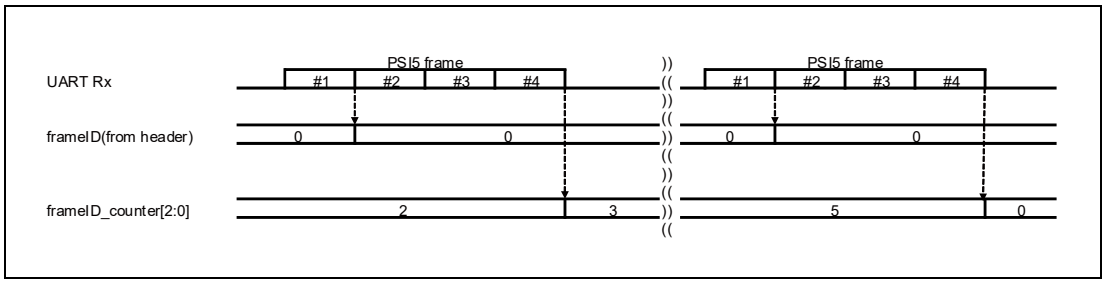


Figure 28.63 Frame ID Counter (Asynchronous Mode Only)

Table 28.96 below shows the conditions for obtaining the storage destination.

Table 28.96 PSI5 Frame MB Target Generate (Async)

XCRC error, UART parity error, UART framing error	FrmID	MB target
No	= 0_H	Ch: Receive ChID Frm: Frame ID counter data (Active Asynchronous mode)
	≠ 0_H	Ch: 0 Frm: 2
Yes	Not related	

The storage data and the storage destination when a WDT error occurs in asynchronous mode are as shown in **Table 28.100**.

(5) Monitoring PSI5 Frame Reception with the WDT

With the WDT installed for each channel, PSI5S checks that PSI5 frames are received frame by frame.

When the WDT enable bit (PSI5SjPWDE(0-7).WDTEB) is set to 1 in PSI5S mode, the WDT counter fetches a specified expiration value and starts counting. Each time a frame is received or the WDT counter value becomes 0, the WDT counter fetches the specified expiration value again and starts counting.

If the WDT counter value becomes 0, occurrence of a WDT error is assumed. For details, see **Section 28.6.2.8, Abnormal Reception (WDT Error)**.

The following figure shows the WDT operation (without a WDT error).

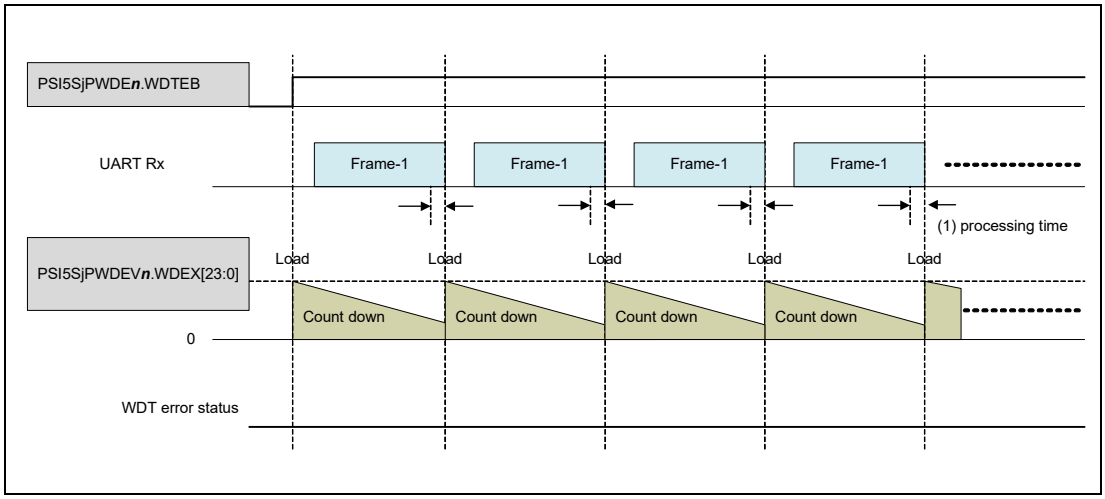


Figure 28.64 WDT Operation Timing (Asynchronous Mode) (without WDT Error)

NOTE

The WDT counter restart in 50 cycles after a stop bit of the frame is received. A WDT error occurs in a case that the WDT counter expires within the 50 clock cycles even if the stop bit of the frame is received.

28.6.2.3 Normal Transmission (Command Data)

After transmitting data (PSI5SjPTFD1 and PSI5SjPTFD2) and the number of transmit packets (PSI5SjPTFNM.TXNUM) is set, PSI5S transmits UART frames according to the specified number of transmit packets when the transmission start bit (PSI5SjPTFST.TXST) is set. No idle time is set between UART frames. To maintain continuous transmission, packet frames are transmitted without idle time between them.

For the UART specifications, see **Section 28.2.1, Functional Overview**.

For 8-byte communication, data is transmitted in order of PSI5SjPTFD1.TDT1[7:0], PSI5SjPTFD1.TDT2[7:0], ..., and PSI5SjPTFD2.TDT8[7:0].

The following figure shows the operation of command data transmission.

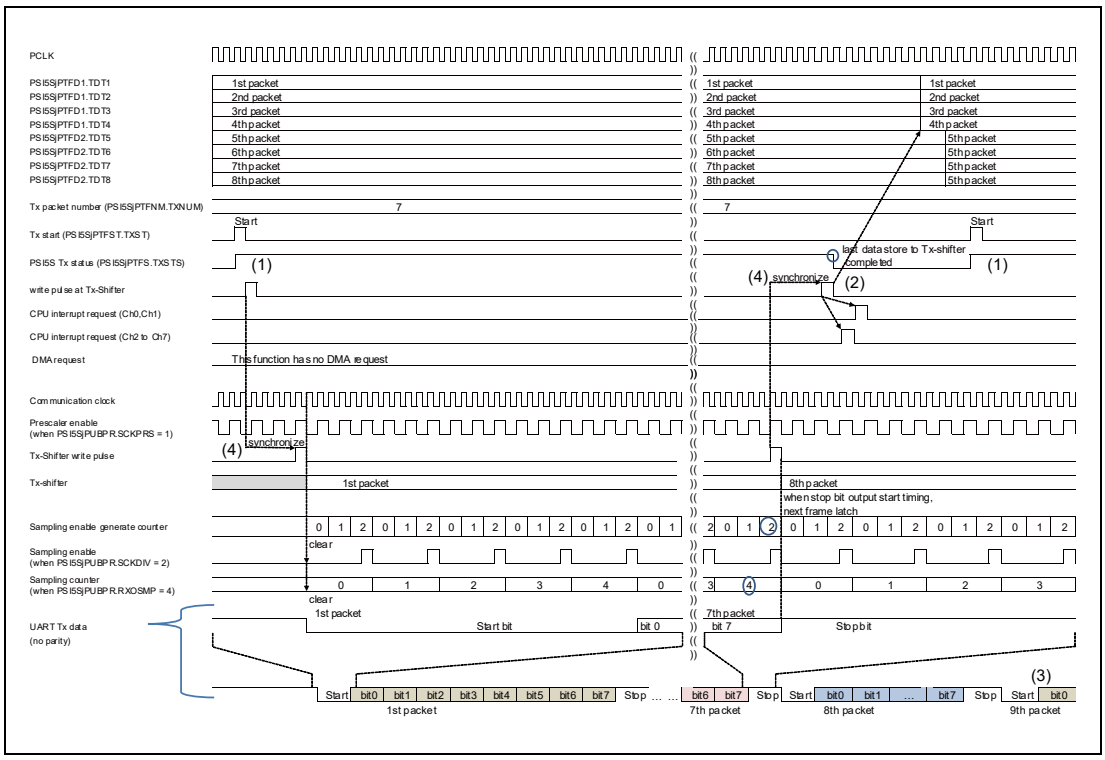


Figure 28.65 Command Data Transmission (8-byte)

(1) The PSI5SjPTFS.TXSTS bit is 1 when TXST is 1.

(2) In the timing of stop bit output, next packet data is set in the transmit data shift register (Tx shifter). The PSI5SjPTFS.TXSTS bit is cleared to 0 to enable writing of the next data by the CPU.

(3) If the CPU writes transmit data continuously, packets are transmitted without idle time.

(4) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

28.6.2.4 Normal Transmission (ECU-to-Sensor Data)

In PSI5S mode, PSI5S transmits a Sync command in synchronization with the synchronization pulse generated by the GTM or an internal counter. Because the initial value of ECU-to-sensor data is set to all 1's, the output value of the Sync command is 1 (command data (PSI5SjPTCD*n* (*n*: 1-7).ATRSCMD[4:0] and PSI5SjPTCD*n* (*n*: 1-7).ACHID[2:0])) unless the transmit data is updated.

After the frame type (frame 1 to frame 4) of ECU-to-sensor data and the ECU-to-sensor data (maximum data size: 20 bits, address: 4 bits) to be transmitted are set, PSI5S generates frame data by adding CRC bits at the end of ECU-to-sensor data. Note that all unused bits for ECU-to-sensor data must be 1.

The generating polynomial for CRC is "g(x) = x6 + x4 +x3 + 1" (initial value: 010101_B) when the number of CRC bits is 6, or "g(x) = x3 + x + 1 (initial value: 111_B)" when the number of CRC bits is 3. The range of CRC calculation is zero extension of the number of CRC bit in the MSB of ECU-to-sensor data (maximum of 24 bits). Data bits are transmitted LSB first.

Because the channels set to asynchronous mode do not use synchronization pulses, ECU-to-sensor transmission is not performed on those channels.

Table 28.97 below shows the frame types available for ECU-to-sensor data.

Table 28.97 ECU-to-Sensor Frame Types

Value of PSI5SjDDSRTYPE[1:0]	Specification	Start Bit	CRC	CRC Calculation Scope
0	Frame 1 (Short) Frame length = 15 ECU to Sensor data = data[2:0], address[2:0]	010_B	3 bits x3+x+1	000_B,data[2:0], address[2:0] (9 bits)
1	Frame 2 (Long) Frame length=29 ECU to Sensor data = data[12:0], address [2:0]	010_B	3 bits x3+x+1	000_B, data[12:0], address [2:0] (19 bits)
2	Frame 3 (XLong) Frame length = 37 ECU to Sensor data = data[18:0], address[2:0]	010_B	3 bits x3+x+1	000_B, data[18:0], address [2:0] (25 bits)
3	Frame 4 (XXLong) Frame length = 43 ECU to Sensor data = data[19:0], address[3:0]	011111110_B	6 bits x6+x4+x3+1	000000_B, data[19:0], address[3:0] (30 bits)

The following figure shows the format of frame data.

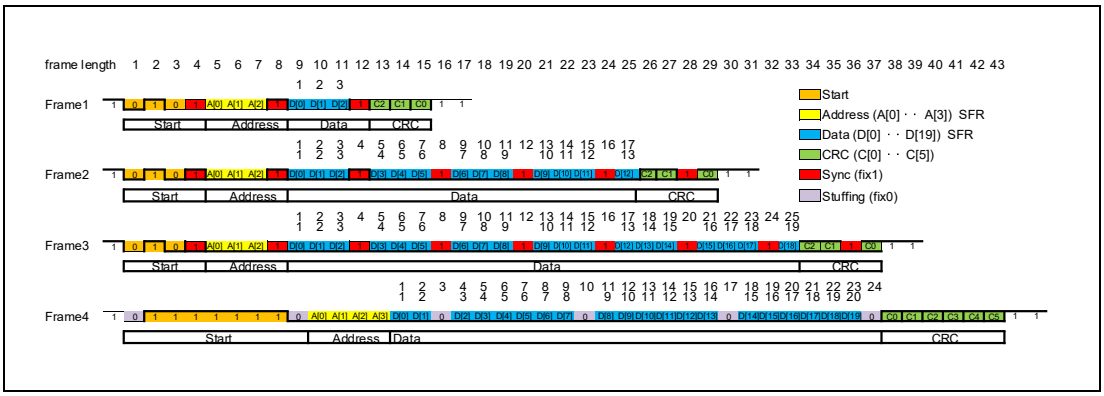


Figure 28.66 ECU-to-Sensor Data Format

After ECU-to-sensor data is generated, the data shifts bit by bit beginning with the start bit at the rising edge of a synchronization pulse for each channel.

Based on the value output by shifting, the command data in SFR is selected (from "PSI5SjPTCD*n* (n: 1 to 7). ATRSCMD[4:0] and one bit of PSI5SjPTCD*n* (n: 1 to 7). ACHID[2:0]" and "PSI5SjPTCD*n* (n: 1 to 7). TRSCMD[4:0] and one bit of PSI5SjPTCD*n*(n: 1 to 7). CHID[2:0]").

When the value output by shifting is 0, the command data in SFR is "PSI5SjPTCD***n*** (n: 1 to 7).TRSCMD[4:0] and one bit of PSI5SjPTCD***n*** (n: 1 to 7).CHID[2:0]".

When the value output by shifting is 1, the command data in SFR is "PSI5SjPTCD***n***(n: 1 to 7).ATRSCMD[4:0] and one bit of PSI5SjPTCD***n***(n: 1 to 7).ACHID[2:0]".

A UART frame is generated from the selected data, and output.

The following figure shows the format of the UART frame to be output.

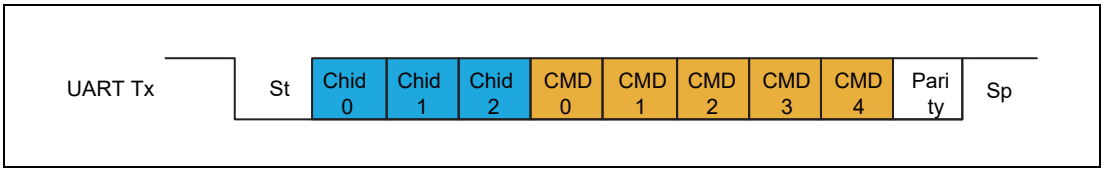
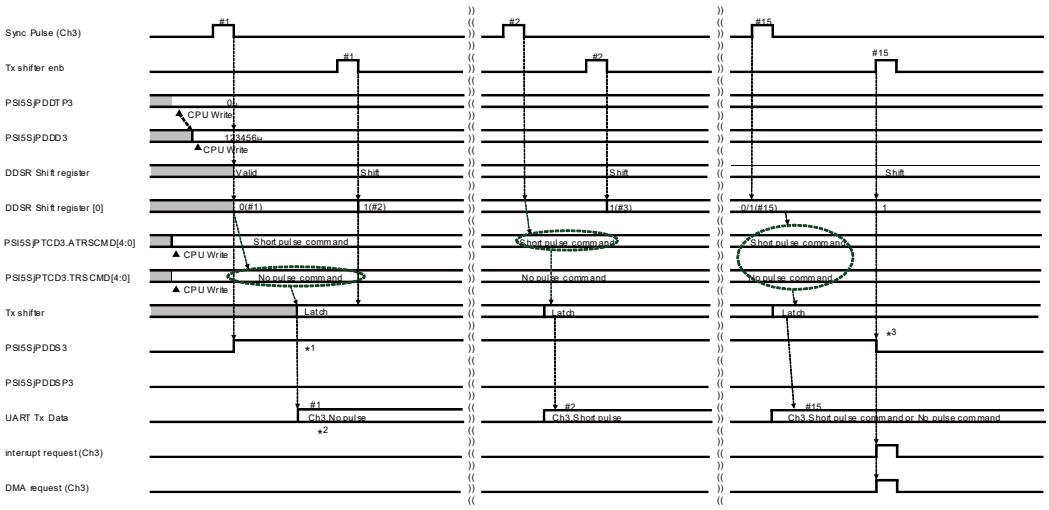


Figure 28.67 ECU-to-Sensor Data Format (UART Frame)

The following figure shows the operation of ECU-to-sensor data transmission.

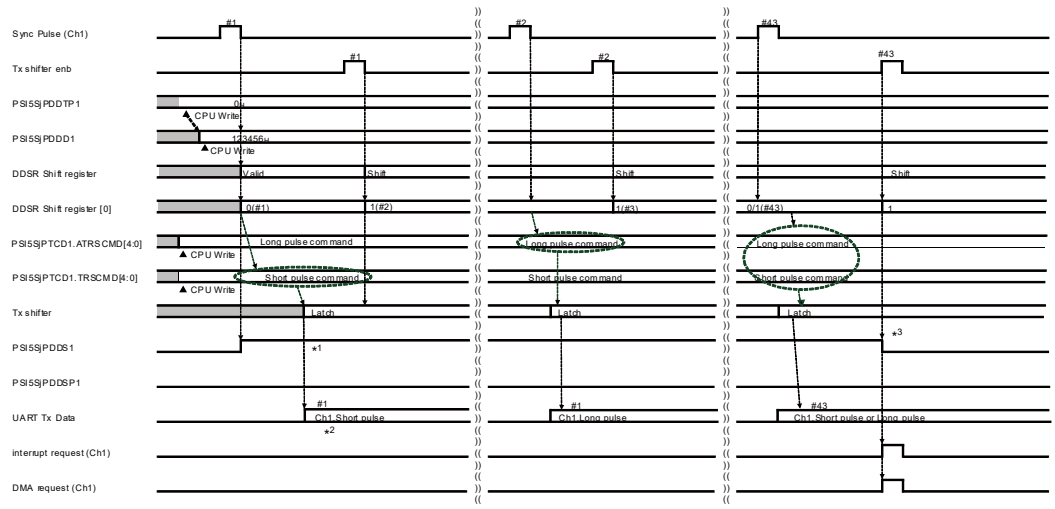


Note 1. When DDSR data is changed, DDSR transmission status is asserted.

Note 2. When the sync pulse (Ch3) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).

Note 3. When the last data is transmitted, the DDSR transmission status is deasserted.

Figure 28.68 ECU-to-Sensor Data Transmission (Frame 1)



Note 1. When DDSR data is changed, the DDSR transmission status is asserted.

Note 2. When the sync pulse (Ch1) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).

Note 3. When the last data is transmitted, the DDSR transmission status is deasserted.

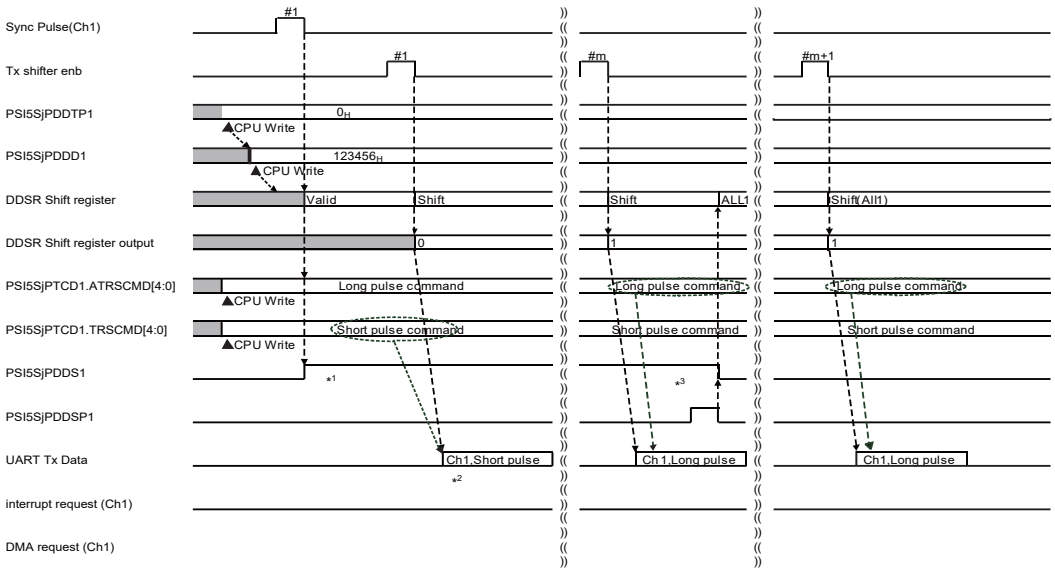
Figure 28.69 ECU-to-Sensor Data Transmit (Frame 4)

28.6.2.5 Normal Transmission (ECU-to-Sensor Data Transmission Stopped)

If transmission is stopped (PSI5SjPDDSP(1-7).DDSRSTP = 1) during ECU-to-sensor data transmission, PSI5S performs the following operations:

- Initializing the DDSR shift register (set all 1)
- De-asserting the DDSR transmission status signal

The following figure shows the operation that is performed when ECU-to-sensor data transmission is stopped.



Note 1. When DDSR data is changed, the DDSR transmission status is asserted.

Note 2. When the sync pulse (Ch1) is asserted, UART Tx data is transmitted.
Command data is selected by the shift register output value (0: TRSCMD, 1: ATRSCMD).

Note 3. When the DDSR stop comes during data transmission, Tx data is initialized (ALL1).

Figure 28.70 ECU-to-Sensor Data Transmission Stop

28.6.2.6 Normal Transmission (Transmit Request Arbitration)

Transmit requests are arbitrated by using the function shown in **Figure 28.45** and output by UART transmission.

The priority of arbitration is: Ch1 (high priority) > Ch2 > Ch3 > Ch4 > Ch5 > Ch6 > Ch7 > Command data (low priority)

The following figure shows an operation in which requests are input after the lower priority requests are input.

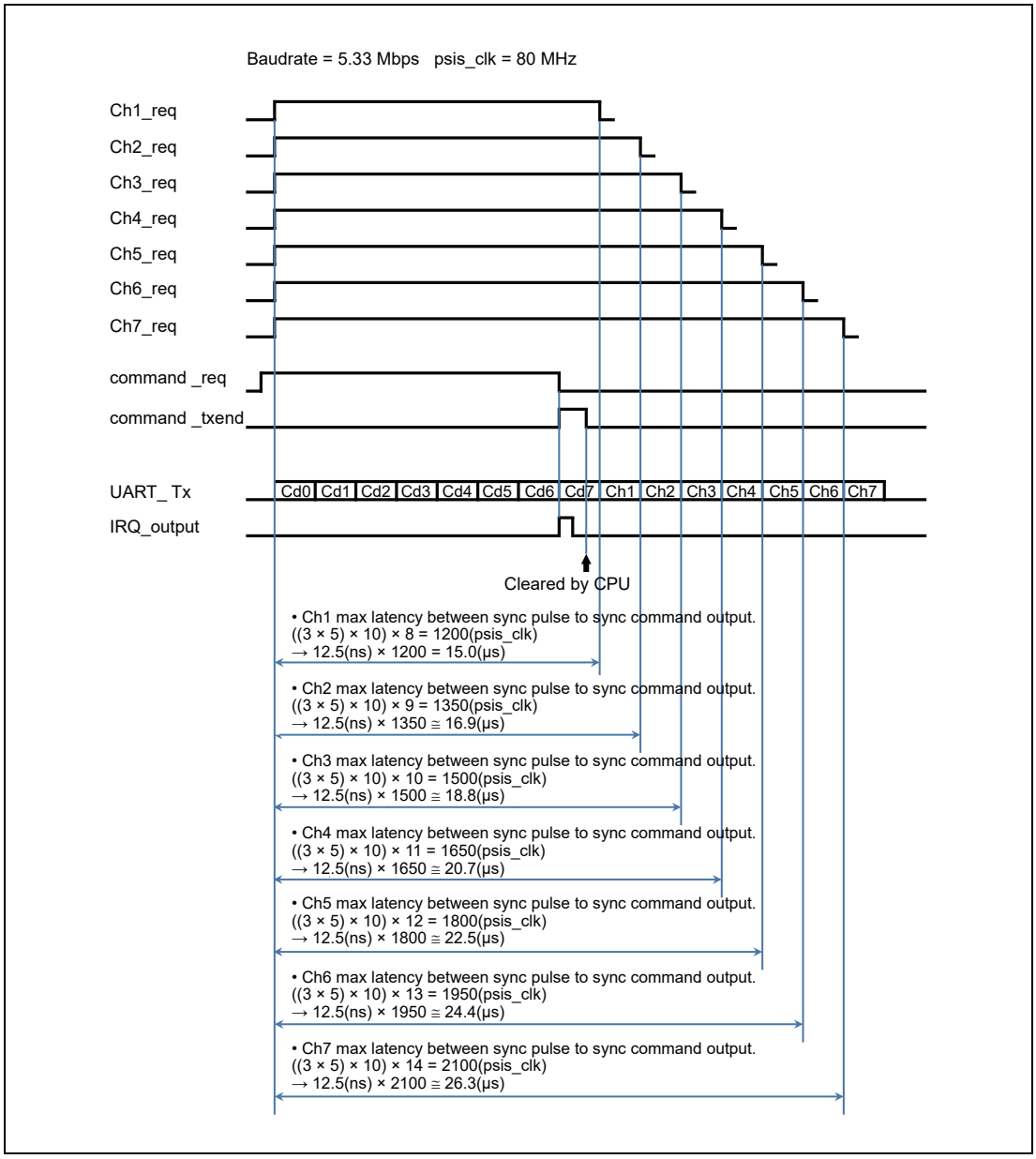


Figure 28.71 Arbitration Tx Request (max latency)

The following figure shows an operation in which all requests are input simultaneously.

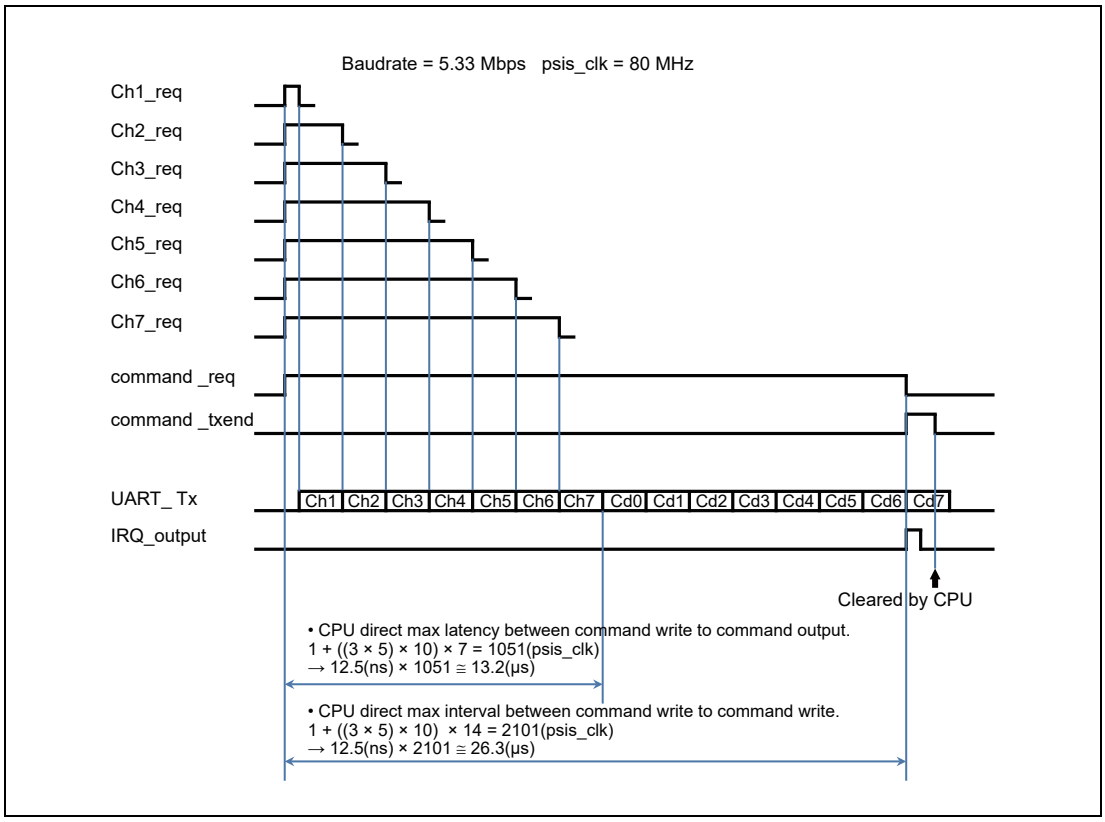


Figure 28.72 Arbitration Tx Request (All Requests Share the Same Timing)

Tx request in FIFO is cleared when the operation enable register (PSI5SjPUOEB.OPEN) is set to 0.

28.6.2.7 Abnormal Reception (Reception Error)

After restoring a PSI5 frame, PSI5S determines whether reception errors have occurred. Then, PSI5S checks for XCRC errors, CRC errors, transceiver status errors, and mailbox overrun.

If mailbox overrun has occurred, the data in the mailbox will be unpredictable.

Other errors are checked for at different timings. The timing for determining other errors is as follows:

UART-related errors (parity and framing errors) are checked for when a UART frame is received.

A WDT error is determined to have occurred when the WDT counter value becomes 0. For details, see **Section 28.6.2.8, Abnormal Reception (WDT Error)**.

An error status is cleared by writing 1 to the corresponding bit in the reception status clear register. The error status is also cleared when the operation mode is switched to configuration mode.

Table 28.98 below lists PSI5 frame reception errors.

Table 28.98 PSI5 Frame Reception Errors

Error	Detect Timing	Detection Condition
XCRC	PSI5 frame received	The agreement of XCRC (6-bit) value which was calculated from received data and the reception XCRC (6-bit) value is confirmed.
CRC	PSI5 frame received & RFCPS=1	The agreement of CRC (3-bit) value which was calculated from received data and the reception CRC (3-bit) value is confirmed.
Parity	PSI5 frame received & RFCPS=0	The agreement of even parity value which was calculated from received data and the reception parity value is confirmed.
Transceiver status	PSI5 frame received & error in header$\neq 0_H$	When the error status value of the received header data is other than 0.
UART parity	UART frame received	The received parity value does not match.
UART framing	UART frame received	The received stop bit value is 0.
WDT	WDT counter value reaches 0	Same as timing of detection
MB overrun	PSI5 frame received	The previous data has not been read.
PSI5 frame lack	IDLE time is passed*1	The number of received packets is less than the number of packets to be received.
PSI5 frame excess	UART frame received*2	The number of received packets exceeds the number of packets to be received.

Note 1. When a PSI5 frame lack error is occurred, the new PSI5 frame data is not stored.
Then if a WDT error occurs, the received packets at the time of the WDT error occurrence is stored.

Note 2. When a PSI5 frame excess error is occurred, the extra data of the new PSI5 frame is not stored.

When the number of received packets exceeds the number of packets to be received, a PSI5 frame excess error, XCRC error and CRC error occur. In this case, a PSI5 frame excess error is detected for the received Ch, and the received data is stored as Ch 0 Frm 2 MB data.

The generating polynomial for XCRC is "$g(x)$ = x6 + x4 +x3 + 1." The initial value of the XCRC bits is 010101_B.

The generating polynomial for CRC is "$g(x)$ = x3 + x + 1." The initial value of the XCRC bits is 111_B.

The following figure shows the ranges of calculation for the XCRC and CRC values for a PSI5S frame that uses CRC and has a payload of 13 bits.

The calculation target of XCRC is PSI5 frame data excluding XCRC, and 0 extension of 6bits to MSB.
The calculation target of CRC is the payload data, and 0 extension of 3bits to MSB.

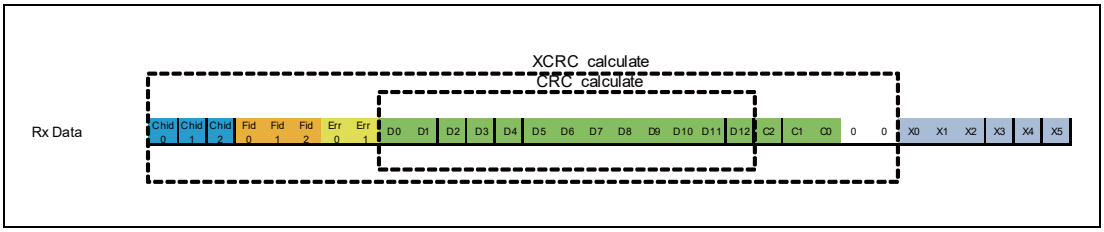
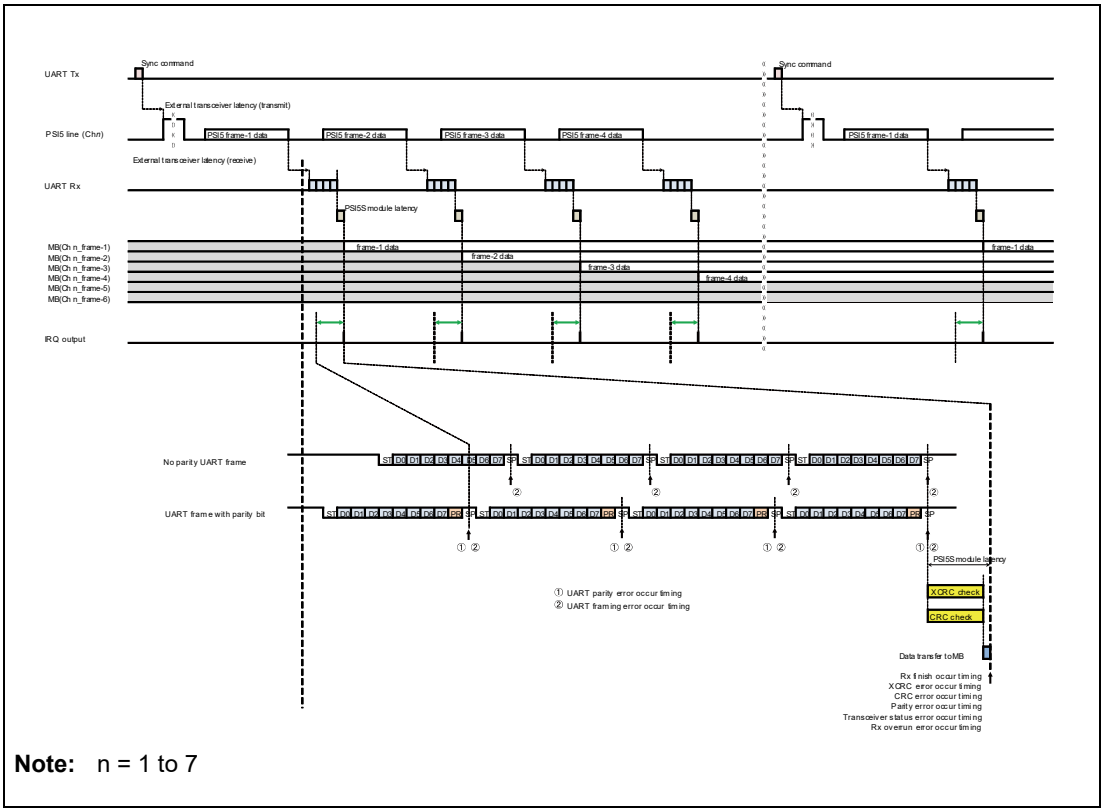


Figure 28.73 Calculate Area of XCRC, CRC

The following figure shows the detection timing of reception errors.



Note: n = 1 to 7

Figure 28.74 Sensor-to-ECU Data Is Received (Reception Error Is Detected)

The following figure shows the operation for error detection when too many or too few UART frames are received.

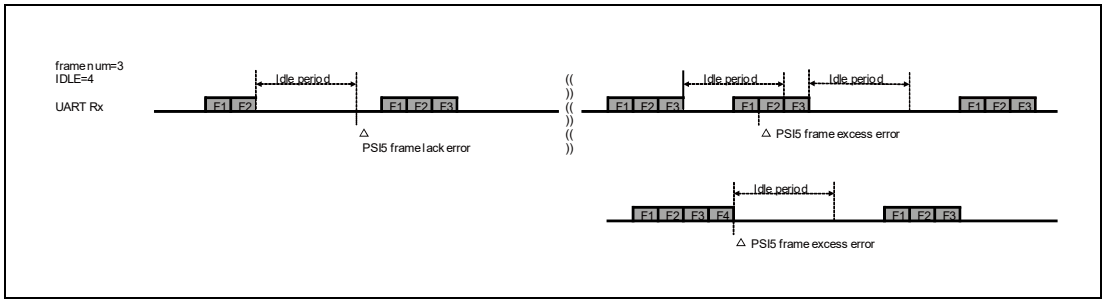


Figure 28.75 UART Frame Error Is Detected (Frame Lack or Frame Excess)

When the PSI5S transitions to the PSI5S mode during a PSI5 frame reception, any error of “UART framing error”, “UART parity error”, “payload data parity error”, “Frame lack error” and “packet frame XCRC error” possibly occurs. Because the PSI5S doesn't detect a bus idle condition when the PSI5S transitions to the PSI5S mode.

28.6.2.8 Abnormal Reception (WDT Error)

The synchronous and asynchronous modes differ from each other in WDT error determination, as follows.

(1) Synchronous Mode

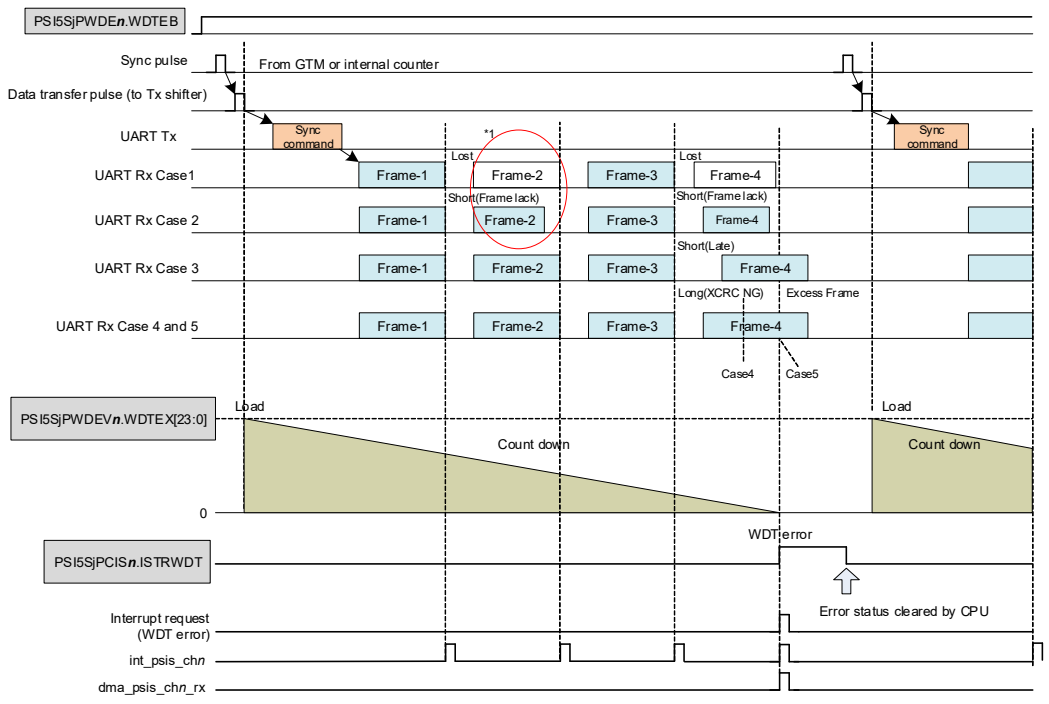
When the synchronization pulse is transmitted to the Tx shifter after synchronization pulse input, the WDT value is reset to the expiration value, and, at the same time, the WDT starts the next operation. If the WDT counter reaches 0 before receiving all the packets to be received, a WDT error is assumed.

Note that the WDT starts operation again after the rising edge of the next synchronization pulse for each channel is detected.

If a WDT error occurs, an interrupt status (PSI5SjPCISn.ISTRWDT) occurs. The interrupt status is cleared by writing 1 to the corresponding bit (PSI5SjPCISCn.ISTCRWDT) in the interrupt status clear register.

The data received until the transmission of the synchronization pulse to the Tx shifter after a WDT error is detected is not stored.

The following figure shows the operation that is performed when a WDT error occurs in synchronous mode.



Note: n = 0 to 7

Note 1. Missing or incomplete packet frames are not stored the data and the error flags.

Figure 28.76 WDT Operation Timing (Sync Mode) — WDT Error

Below shows the MailBox triplet when a WDT error occurs in synchronous mode.

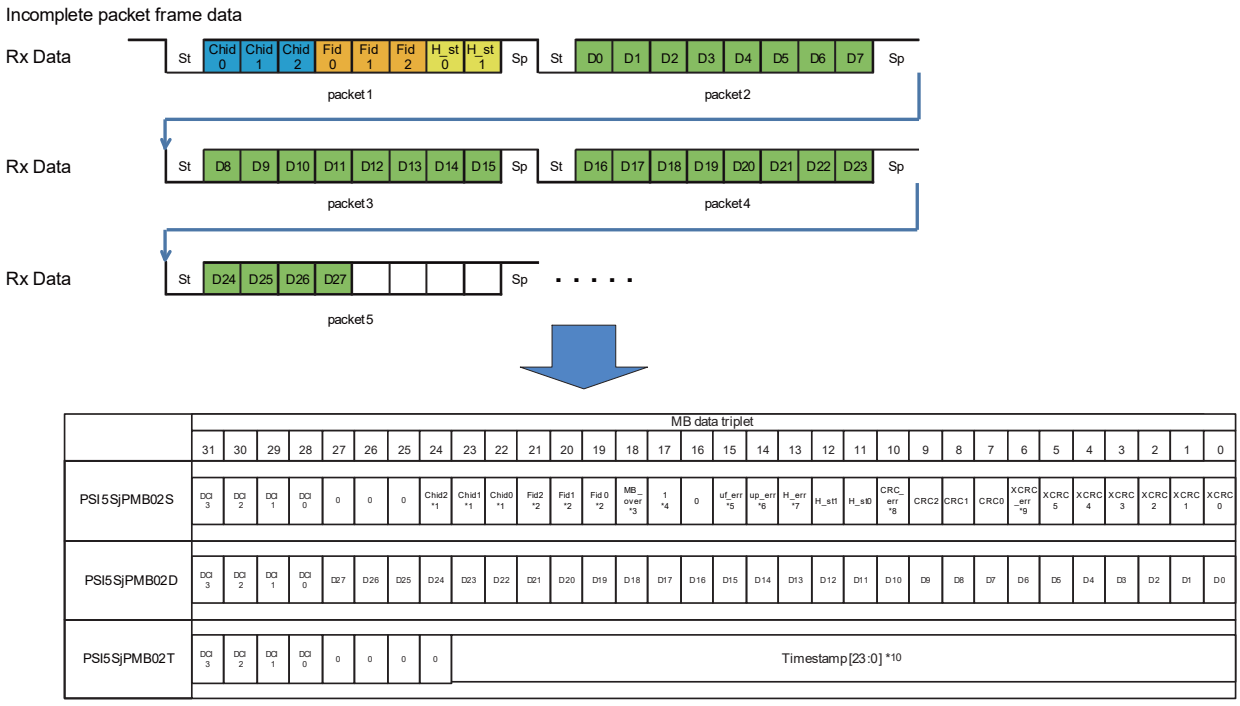
Table 28.99 MB Data Triplet - WDT Error (Synchronous Mode)

Case of UART Rx	Case1	Case2	Case3	Case4	Case5
Store location	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2
DCI	Ch0	Ch0	Ch0	Ch0	Ch0
CHID	Channel where WDT error detected	Channel where WDT error detected	Channel where WDT error detected	CHID in the header of the reception frame	Channel where WDT error detected
FID	Packet Frame Counter of channel where WDT error detected	Packet Frame Counter of channel where WDT error detected	Packet Frame Counter of channel where WDT error detected	FrmID in the header of reception frame	Packet Frame Counter of channel where WDT error detected
MBORERR	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error
WDTERR	1	1	1	0	1
UTFRERR	0	UART framing error	UART framing error	UART framing error	UART framing error
UTPTERR	0	UART parity error	UART parity error	UART parity error	UART parity error
HEADERR	0	Header error	Header error	Header error	Header error
HEADST	0	HEADST in the header of the reception frame	HEADST in the header of the reception frame	HEADST in the header of the reception frame	HEADST in the header of the reception frame
CRCERR	0	0	0	CRC error	CRC error
CRC	000_B	000_B	000_B	CRC of reception frame data	CRC of reception frame data
XCRCERR	0	0	0	1	1
XCRC	00_0000_B	00_0000_B	00_0000_B	XCRC of reception frame data	XCRC of reception frame data
Payload data	000_0000_H	Incomplete frame data *1, *2	Incomplete frame data *1, *2	Payload data of received frame data	Incomplete frame data *1, *2
Time stamp	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At header reception (Ch0)	At WDT error detection (Ch0)

Note 1. Not received (missing or incomplete) packet bits are “0”.

Note 2. Incomplete data is stored in packets 2 to 4 and the lower 4 bits of packet 5.

The following figure shows incomplete data format and store location.



(*1) The channel ID of the WDT error occurrence channel (0-7).
(*2) When the WDT error occurrence channel is set Async mode (PSI5SjPRCF1n.SYSEL = 1), Fid[2:0] is stored.
When the WDT error occurrence channel is set Sync mode (PSI5SjPRCF1n.SYSEL = 0), packet frame counter value is stored.
(*3) When MB overrun error occur, this bit is set.
(*4) WDT error status is set.
(*5) When 6packet frame data has UART framing error, this bit is set.
(*6) When 6packet frame data has UART parity error, this bit is set.
(*7) If H_st[1:0] ≠ 0b00, this bit is set.
(*8) When 6packet frame data has CRC error, this bit is set.
(*9) When 6packet frame data has XCRC error, this bit is set.
(*10) The capture value of timestamp (Ch0) at WDT error occurrence.

Figure 28.77 PSI5 frame data format and MB data triplet at WDT error

The following figure shows the operation of the packet frame counter.

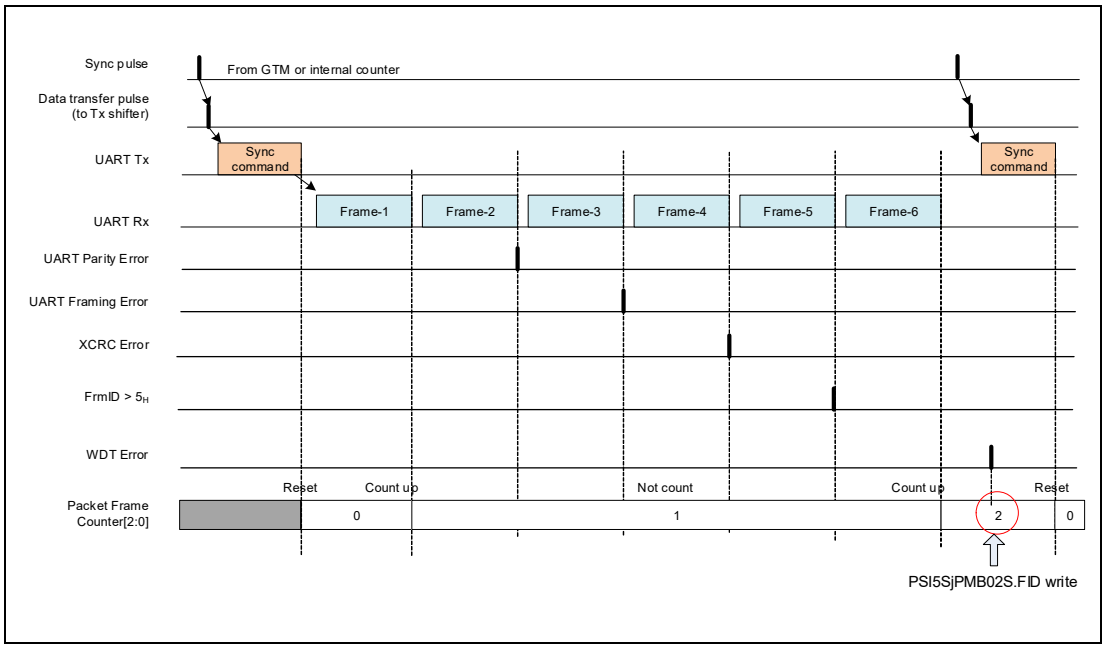
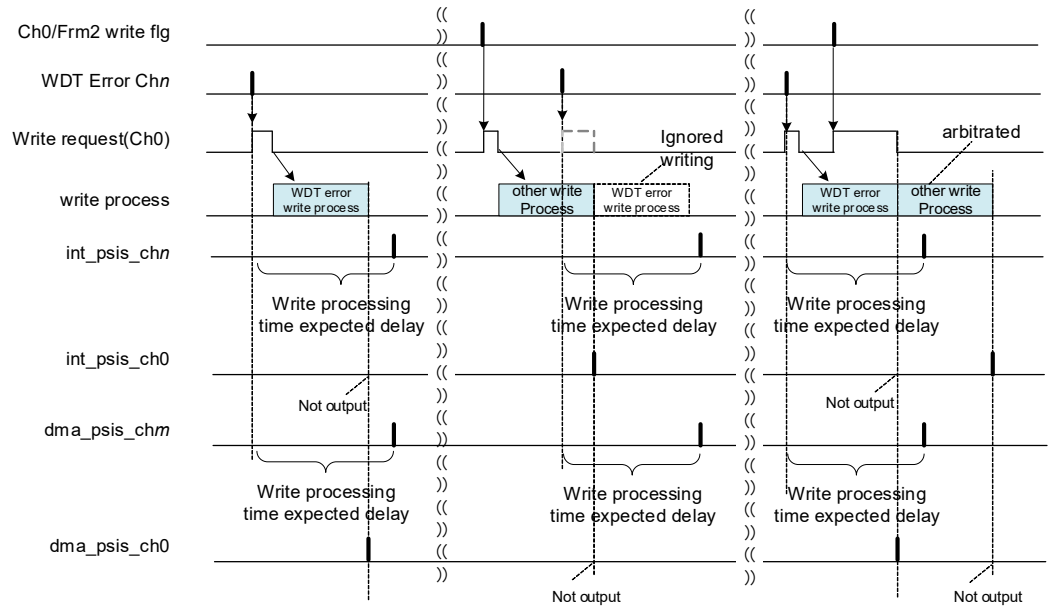


Figure 28.78 Packet Frame Counter

The following figure shows the arbitration of the write request to Ch0 and the operation of the interrupt and DMA request.



Note: n = 0 to 7, m = 1 to 7

Figure 28.79 Write Request Arbitrated and Interrupt/DMA Output (Synchronous Mode)

The interrupt / DMA request by WDT error outputs from detection Ch after writing the MailBox of Ch0.

When a write request by WDT error during the write processing to Ch0/frm2, interrupt/DMA request outputs and the MB data triplet is discarded.

When a write request by another error (not WDT error) during the write processing to Ch0/frm2, the interrupt/DMA request outputs and the MB data triplet write after the write processing to Ch0/frm2.

(2) Asynchronous Mode

When reception of a frame ends, the WDT value is reset to the expiration value, and, at the same time, the WDT starts the next operation. If the WDT counter reaches 0 before reception of the next frame ends, occurrence of a WDT error is assumed.

In such a case, the WDT value is reset to the initial value, and the WDT starts operation again.

The conditions for starting the first operation of the WDT are as follows.

When the WDT enable bit is set to 1, an expiration value is set in the WDT counter. When the operation mode is subsequently switched to PSI5S mode, the WDT starts operation. If the WDT enable bit is 0 (disable), the WDT does not start operation.

If a WDT error occurs, an interrupt status (PSI5SjPCISn.ISTRWDT) occurs. The interrupt status is cleared by writing 1 to the corresponding bit (PSI5SjPCISCn.ISTCRWDT) in the interrupt status clear register.

The following figure shows the operation that is performed when a WDT error occurs in asynchronous mode.

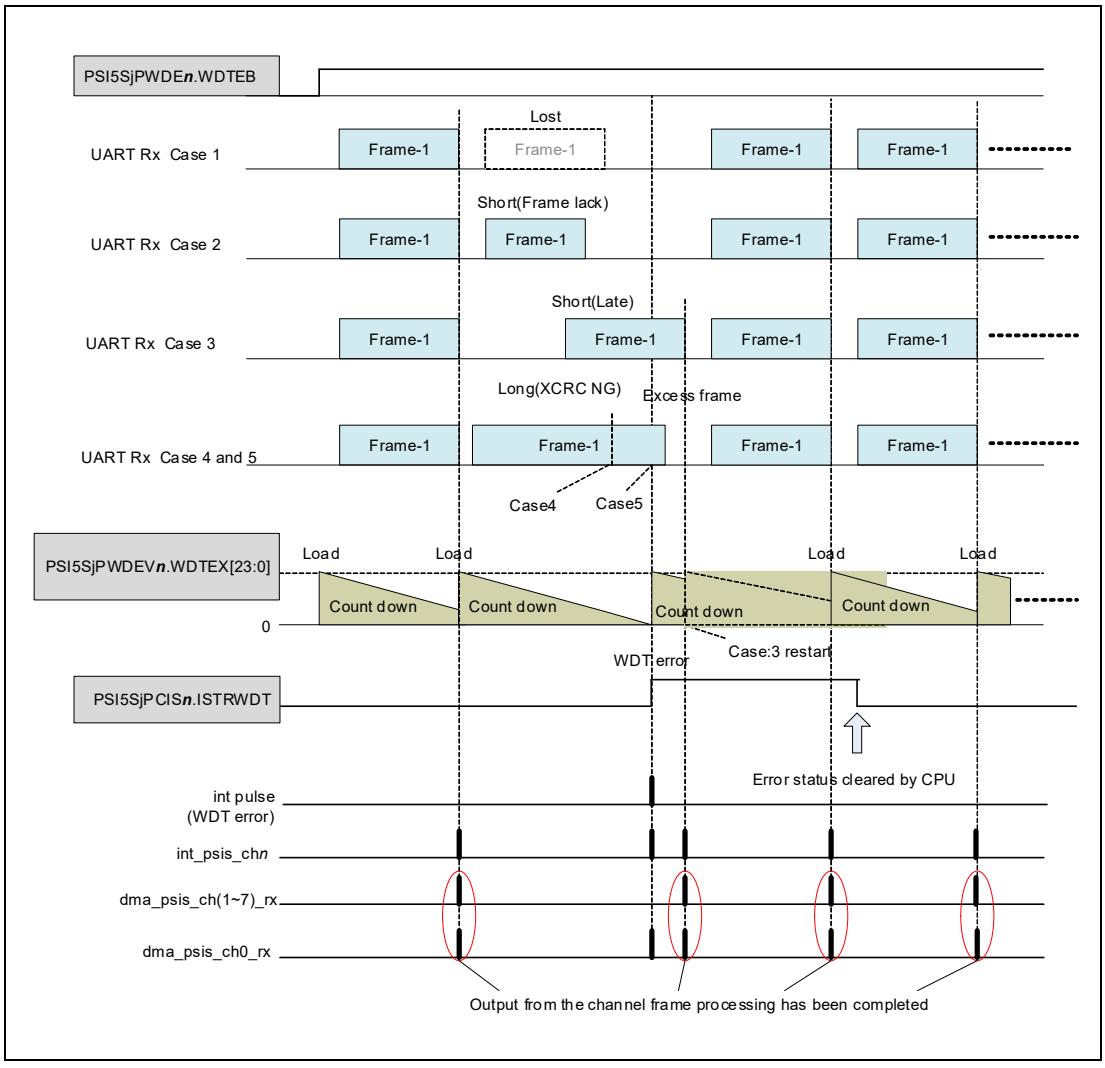


Figure 28.80 WDT Operation Timing (Asynchronous Mode) (WDT Error)

Below shows the MailBox triplet when a WDT error occurs in asynchronous mode.

Table 28.100 MB Data Triplet - WDT Error (Asynchronous Mode)

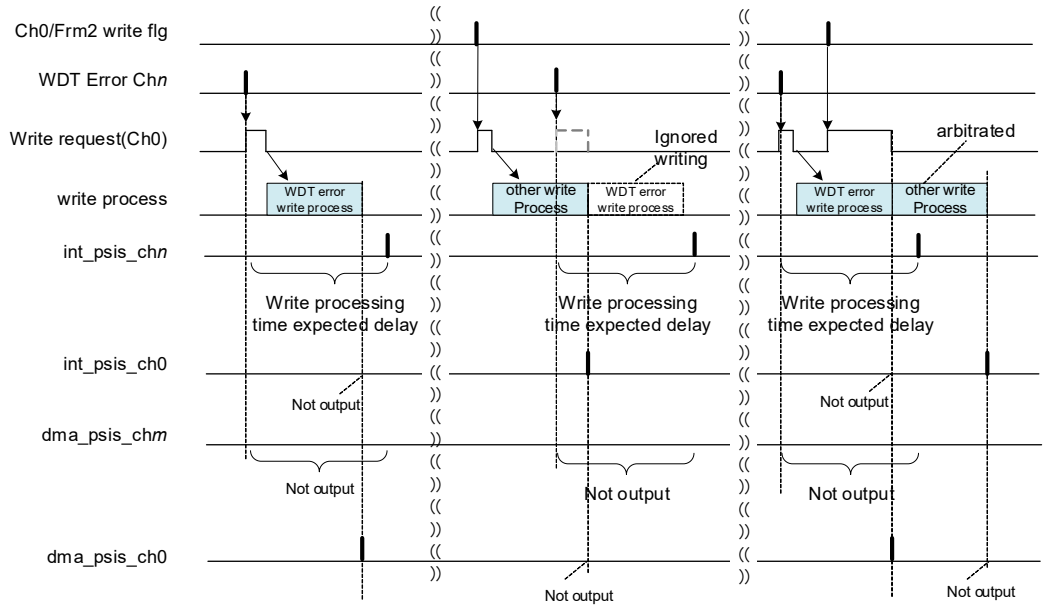
Case of UART Rx	Case1	Case2	Case3	Case4	Case5
Store location	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2	ch0/frm2
DCI	Ch0	Ch0	Ch0	Ch0	Ch0
CHID	Channel where WDT error detected	Channel where WDT error detected	Channel where WDT error detected	CHID in the header of the reception frame	Channel where WDT error detected
FID	000_B	FmID in the header of reception frame	FmID in the header of reception frame	FmID in the header of reception frame	FmID in the header of reception frame
MBORERR	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error	MailBox overrun error
WDTERR	1	1	1	0	1
UTFRERR	0	UART framing error	UART framing error	UART framing error	UART framing error
UTPTERR	0	UART parity error	UART parity error	UART parity error	UART parity error
HEADERR	0	Header error	Header error	Header error	Header error
HEADST	0	HEADST in the header of reception frame	HEADST in the header of reception frame	HEADST in the header of reception frame	HEADST in the header of reception frame
CRCERR	0	0	0	CRC error	CRC error
CRC	000_B	000_B	000_B	CRC of reception frame data	CRC of reception frame data
XCRCERR	0	0	0	1	1
XCRC	00_0000_B	00_0000_B	00_0000_B	XCRC of reception frame data	XCRC of reception frame data
Payload data	000_0000_H	Incomplete frame data *1, *2, *3	Incomplete frame data *1, *2, *3	Payload data of received frame data	Incomplete frame data *1, *2, *3
Time stamp	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At WDT error detection (Ch0)	At header reception (Ch0)	At WDT error detection (Ch0)

Note 1. Not received (missing or incomplete) packet bits are “0”.

Note 2. Incomplete data is stored in packets 2 to 4 and the lower 4 bits of packet 5.

Note 3. The incomplete data format is the same as synchronous mode.

The following figure shows the arbitration of the write request to Ch0 and the operation of the interrupt and DMA request.



Note: n = 0 to 7, m = 1 to 7

Figure 28.81 Write Request Arbitrated and Interrupt/DMA Output (Asynchronous Mode)

The interrupt request by WDT error outputs from detection Ch after writing the MailBox of Ch0.

The DMA request by WDT error outputs from Ch0 after writing the MailBox of Ch0.

When a write request by WDT error during the write processing to Ch0/frm2, interrupt request outputs and the MB data triplet is discarded.

When a write request by another error (not WDT error) during the write processing to Ch0/frm2, the interrupt request outputs and the MB data triplet write after the write processing to Ch0/frm2.

NOTE

When UART Parity error or UART Framing error or XCRC error is occurred, the received data is stored in Frame2 of Ch0. In that case, there is possible to detect the WDT error of reception Ch, because the reception is not stored in expected Ch.

28.6.2.9 Abnormal Transmission (Transmission Error)

If ECU-to-sensor data is set for a channel while ECU-to-sensor data is being transmitted*1 in the same channel, an ECU-to-sensor data overwrite error is determined to have occurred.

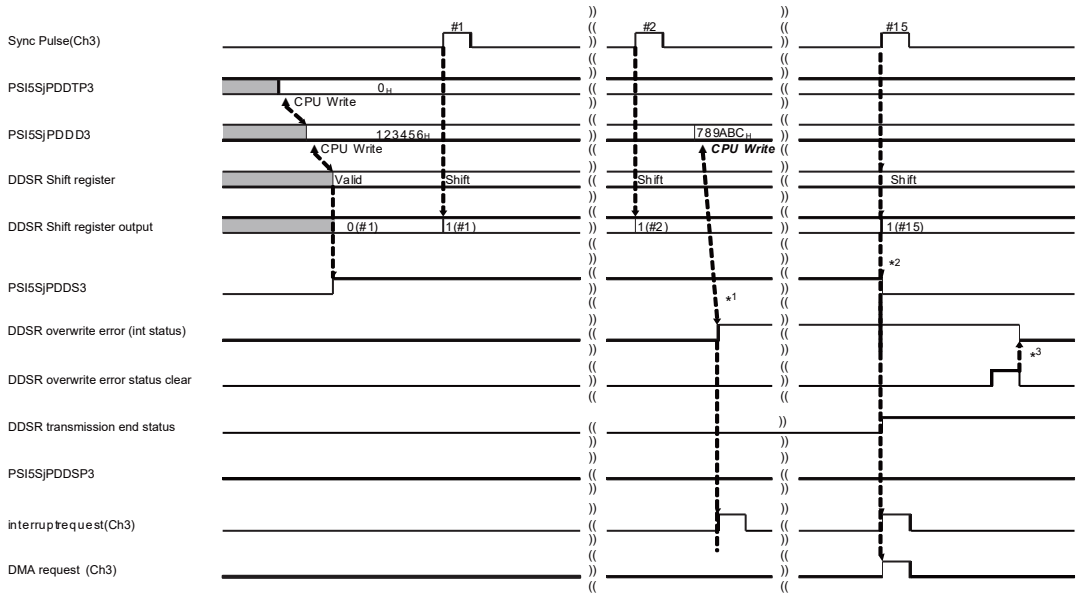
Note 1. Period from setting the value of transmission command to storing the last data in the Tx shifter.

If an overwrite error is determined, the data written afterward is not transmitted.

Data that is already being transmitted is not affected by the overwrite error.

If an overwrite error occurs, an interrupt status signal (PSI5SjPCIS1.ISTDDSOW) is asserted. The interrupt status signal is cleared by writing 1 to the corresponding bit (PSI5SjPCISC1.ISTCDDSOW) in the interrupt status clear register.

The following figure shows the operation that is performed when an ECU-to-sensor data overwrite error occurs.



Note 1. When the DDSR data is changed while DDSR status = 1, the DDSR overwrite error status is asserted.
Note 2. When all data has been outputs, the status is cleared
Note 3. 1 is written to the DDSR overwrite error clear status bit and the DDSR overwrite error status is cleared.

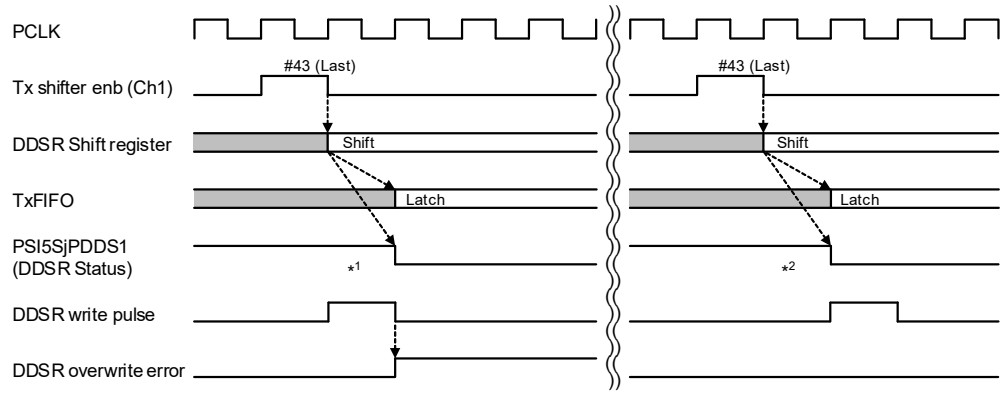
Figure 28.82 ECU-to-Sensor Data Transmission (Overwrite Error)

The following describes the operation that is performed when updating the ECU-to-sensor data conflicts with the start of the last data transmission triggered by a synchronization pulse.

When updating the ECU-to-sensor data occurs after 1 PCLK cycle from the start of the last data transmission triggered by a synchronization pulse, occurrence of an overwrite error is assumed.

When updating the ECU-to-sensor data occurs after 2 PCLK cycles from the start of the last data transmission triggered by a synchronization pulse, occurrence of an overwrite error is not assumed.

In configuration mode, even if updating the ECU-to-sensor data occurs, PSI5S does not start the transmission and PSI5S does not detect an overwrite error.



Note 1. If DDSR data is updated before or during the same PCLK cycle that the DDSR status is cleared, the DDSR overwrite error status is asserted.

Note 2. If DDSR data is updated 1 PCLK cycle or more after the DDSR status is cleared, the DDSR overwrite error status is not asserted.

Figure 28.83 ECU-to-Sensor Data Transmit (Overwrite Error Operation is Conflicted)

28.6.2.10 Interrupt Output

PSI5S can output eight interrupt signals.

Interrupts can be output only when interrupt output is enabled. If an interrupt factor occurs when interrupt output is disabled, no interrupt is output, but the interrupt status is indicated.

Table 28.101 below lists the factors of interrupts that are output in PSI5S mode.

Table 28.101 Interrupt Output Timings

Output Signal	Interrupt Output Timing
int_psis_ch0 to 7	When MB data is received and stored to SFR.
int_psis_ch0 to 7	When WDT operates, the specified time elapses, and the value matches the compare value.
int_psis_ch0 to 7	When a UART frame is received after the start of the idle period.
int_psis_ch0 to 7	When a UART frame is received and the next UART frame is received before the start of the idle period.
int_psis_ch0	When the last command data is stored in the Tx shifter.
int_psis_ch1 to 7	When the last DDSR data is stored in the Tx shifter.
int_psis_ch1 to 7	When DDSR data is written while another DDSR data is being transmitted.

The following describes the interrupt statuses and how to clear them.

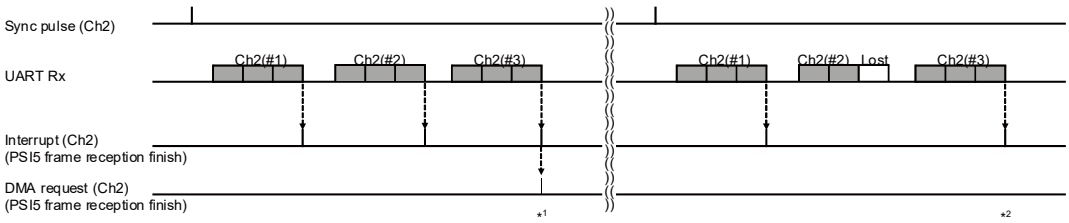
To clear an interrupt status, write 1 to the bit corresponding to the interrupt status in the interrupt status clear register.

Table 28.102 Interrupt Statuses

Equipped Ch	Interrupt Status
Ch0	XCRC error
Ch0 to Ch7	CRC error
Ch0 to Ch7	Parity error
Ch0 to Ch7	Transceiver status error
Ch0	UART Rx parity error
Ch0	UART Rx framing error
Ch0 to Ch7	WDT error
Ch0 to Ch7	PSI5 frame reception overrun error
Ch0 to Ch7	PSI5 frame lack error
Ch0 to Ch7	PSI5 frame excess error
Ch0 to Ch7	PSI5 frame reception finish
Ch1 to Ch7	DDSR overwrite error
Ch1 to Ch7	DDSR transmit finish
Ch0	Command data transmit finish

The following describes the operation in which an interrupt status occurs and is cleared.

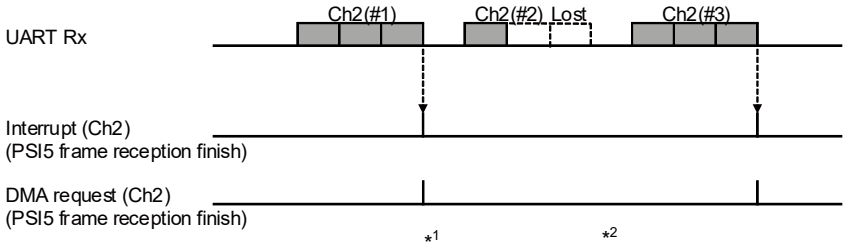
The following figure shows interrupt output and DMA request output operations in synchronous mode.



Note 1. When the last frame is received at each Ch, a DMA request is asserted.
Note 2. When the last frame is received at each Ch but there is a frame lost, a DMA request is not asserted.

Figure 28.84 Interrupt and DMA Request Timing (Synchronous Mode)

The following figure shows interrupt output and DMA request output operations in asynchronous mode.



Note 1. When a frame is received at each Ch, an interrupt and DMA request are asserted.
Note 2. When a frame is received at each Ch but there is a frame lost, an interrupt and DMA request are not asserted.

Figure 28.85 Interrupt and DMA Request Timing (Asynchronous Mode)

If the occurrence of an error status and the clearing of an interrupt status occur at the same time, the occurrence of the error has priority over the occurrence of the interrupt status.

The following figure shows the operation that is performed under conditions for setting and clearing interrupts.

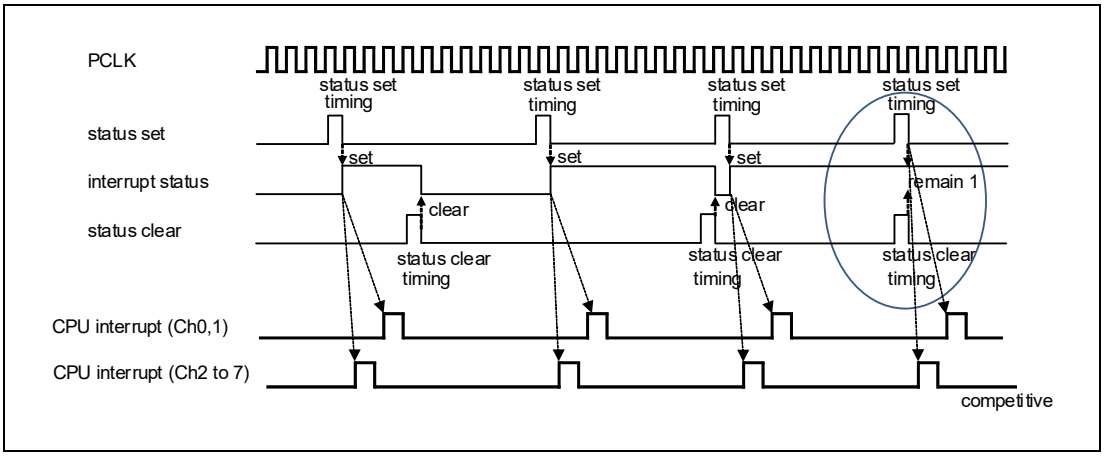


Figure 28.86 Interrupt Operation (Competitive)

If an interrupt setting condition is met while an interrupt status bit is set, an interrupt is output.

The following figure shows the operation described above.

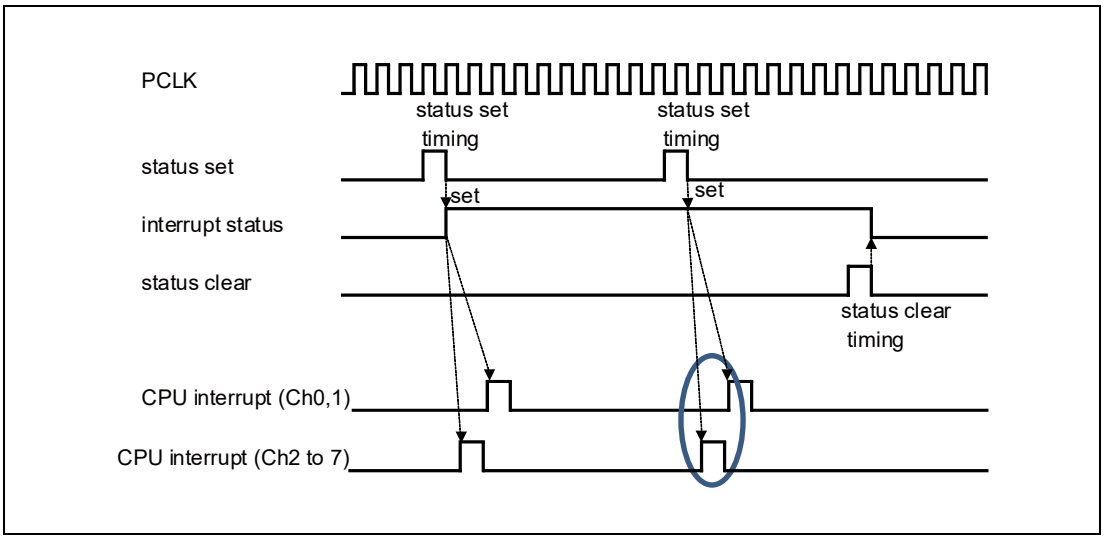


Figure 28.87 Interrupt Operation (Error Setting Occurs When Interrupt Status Is Set)

NOTE

A interrupt may be outputted in a period of maximum 14 PCLK even after PUOS.ACSTS becomes 0.

Then the interrupt should be ignored.

28.6.2.11 DMA Request Output

PSI5S can output 15 DMA requests (eight receive requests and seven transmit requests).

DMA requests are output only when DMA request output is enabled. DMA requests are not output when DMA request output is disabled.

Table 28.103 below lists the factors of DMA requests that are output in PSI5S mode.

Table 28.103 Timings of DMA Transfer Request Output

Operation Mode	Equipped Channel	Rx/Tx	DMA Request Output Factor
PSI5S mode	Ch0 to Ch7	Rx	<Sync> When last the PSI5 frame is stored in SFR. <Async> When a PSI5 frame is stored in SFR.
	Ch0 to Ch7	Rx	When a WDT error occurs.
	Ch1 to Ch7	Tx	When the last DDSR data is stored in the Tx shifter

Figure 28.76, **Figure 28.80**, **Figure 28.84**, and **Figure 28.85** show the DMA request output operations for reception.

Figure 28.68 and **Figure 28.69** show the DMA request output operations for transmission.

28.6.2.12 Transition to Configuration Mode

The operation of PSI5S to return to the configuration mode is described in **Section 28.4, Operation Modes**.

Figure 28.88 below describes the transition to the configuration mode during reception and transmission.

When CPU writes PSI5SjPUOEB.OPEN to 0 during reception, PSI5S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5S is not sending a frame.

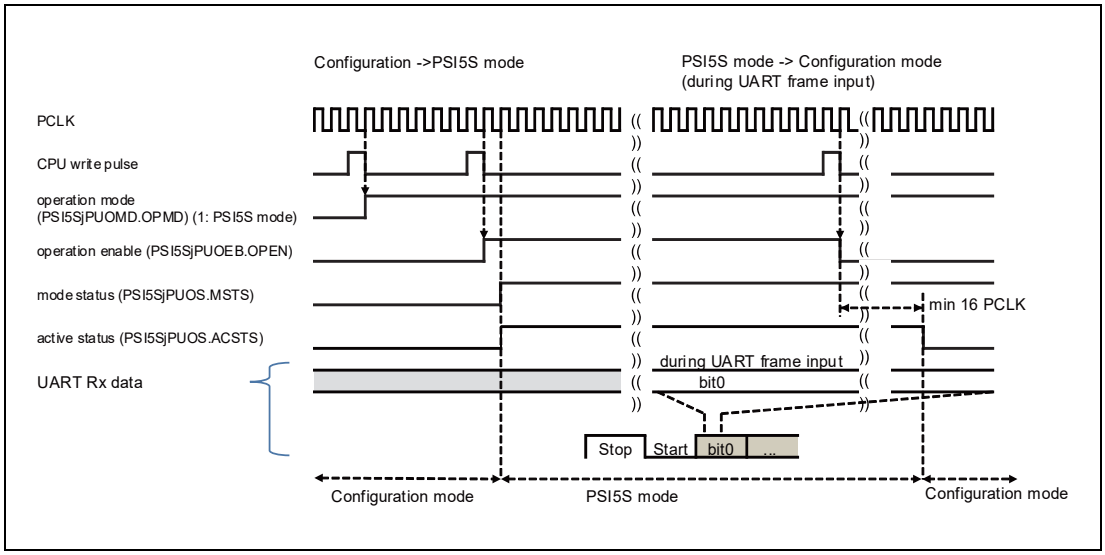


Figure 28.88 Move from PSI5S mode to Configuration Mode (Rx)

When CPU writes PSI5SjPUOEB.OPEN to 0 during transmission, PSI5S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5S isn't sending a frame. If PSI5S sends a frame, it transitions to configuration mode after it completes to send the frame.

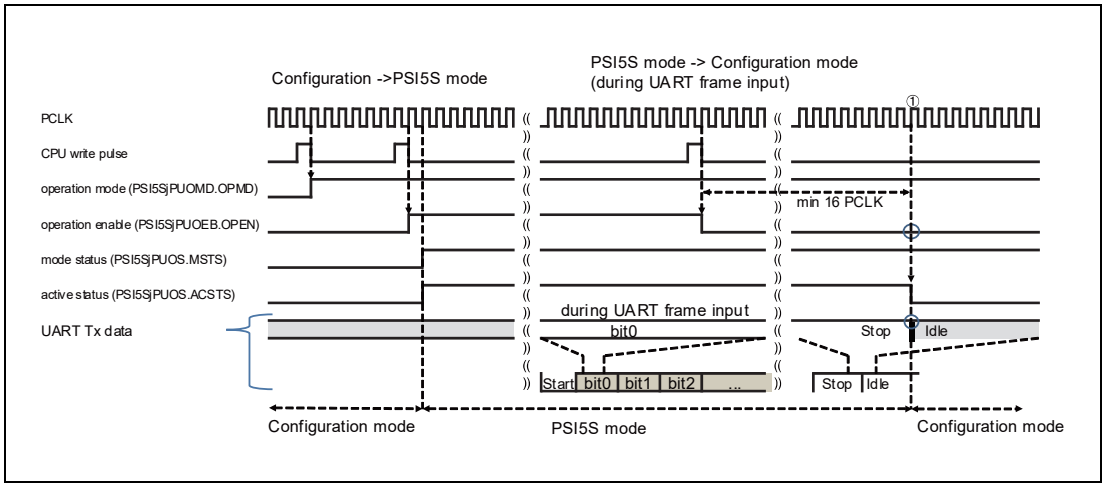


Figure 28.89 Move from PSI5S mode to Configuration Mode (Tx)

NOTE

In a case that the command to transceiver has been accepted in UART transmission module in PSI5S when CPU writes PSI5SjPUOEB.OPEN to 0, PSI5S wait for completion of the UART transmission. The maximum period of the mode transition time is a period of 9 UART frames.

28.6.3 Operations in UART Mode

When the operation mode is UART mode (PSI5SjPUOMD.OPMD = 0) and the operation is enabled (PSI5SjPUOEB.OPEN = 1), PSI5S can perform 1-byte UART communication with external devices.

Baud rate settings (prescaler division ratio, clock division ratio, and oversample number) are described in **Section 28.5.1, Common Setting Procedure**.

For the UART specifications, see **Section 28.2.1, Functional Overview**.

28.6.3.1 Normal Reception

PSI5S receives UART frames. Each time PSI5S receives one UART frame, PSI5S performs serial-to-parallel (S/P) conversion and determines whether there are any reception errors.

If no reception errors are found, PSI5S determines that the reception is normal, and then updates the receive data register (PSI5SjUCRD) and normal reception end status (PSI5SjUCRS.UTRFIN).

Figure 28.90 shows an example of UART reception.

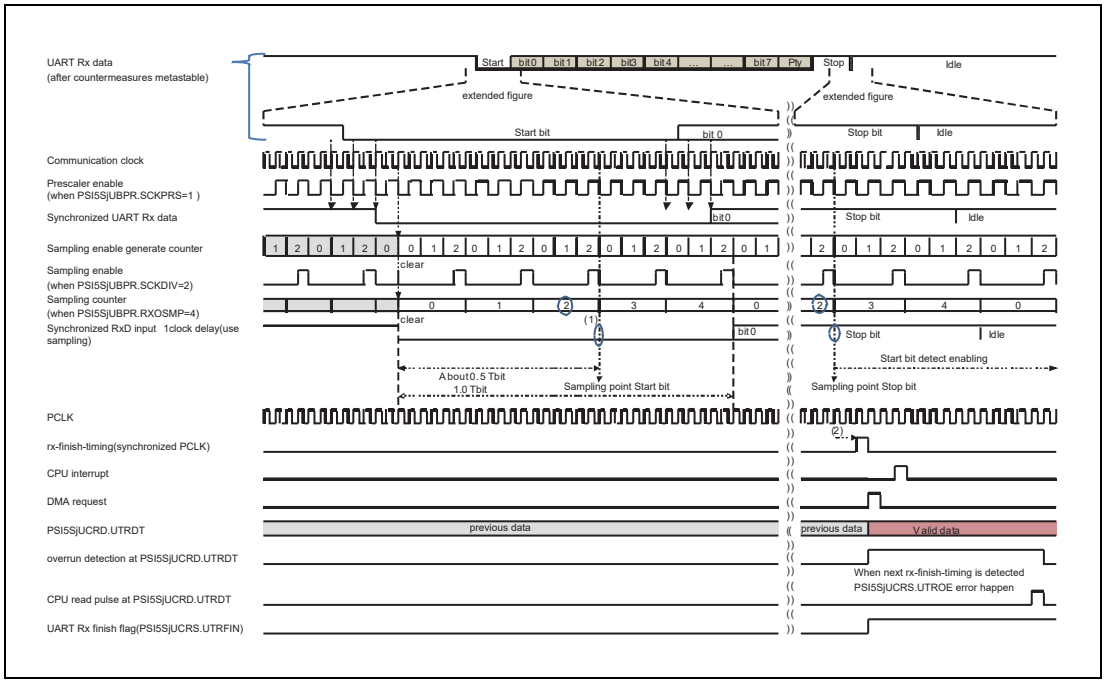


Figure 28.90 UART Rx (No Error)

(1) When the sampling counter value is “PSI5SjPUBPR.RXOSMP/2,” PSI5S samples receive data. (Example: When PSI5SjPUBPR.RXOSMP = 4, PSI5S fetches receive data when the sampling counter value is 2.)

(2) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

28.6.3.2 Normal Transmission

When the transmit data register (PSI5SjUCID) is updated, PSI5S transmits UART frames.

When transmission ends, PSI5S updates the transmission status (PSI5SjUCTS, PSI5SjUCTM).

Figure 28.91 shows an example of UART transmission.

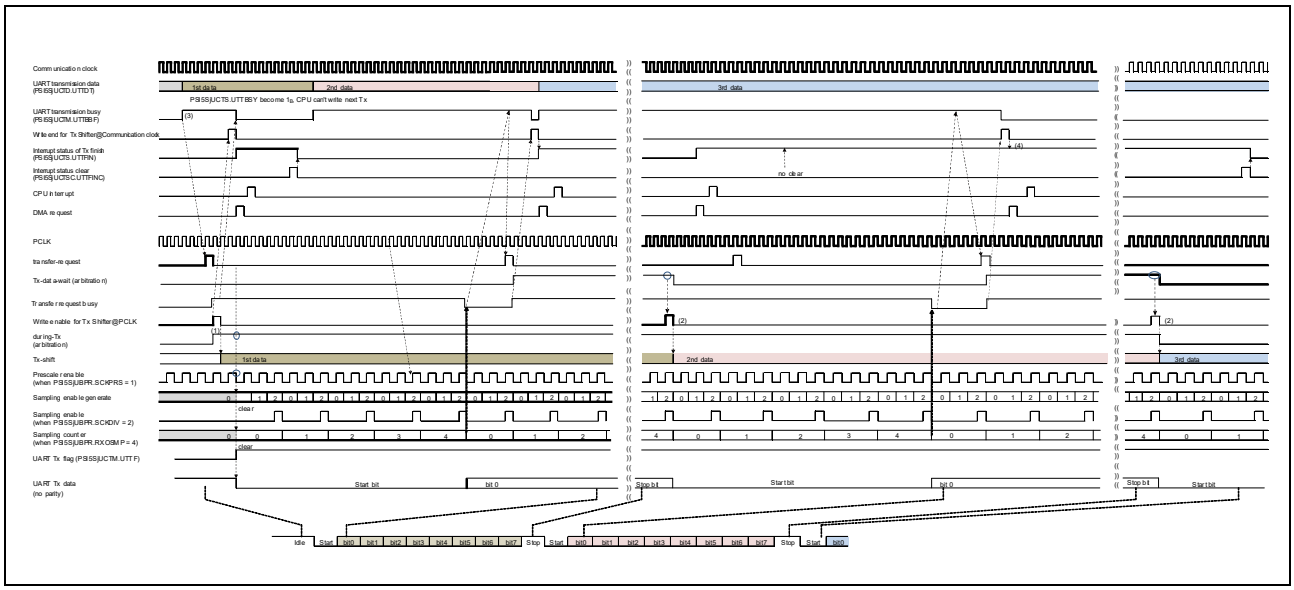


Figure 28.91 UART Tx (No Error)

(1) When the value of transfer-request is 1 and the during-Tx signal is high level, the signal to enable writing to the transmit data shift register (Tx Shifter) is output.

(2) When the Tx-data-wait signal is high at the end of stop bit, the write enable signal to the Tx shifter is generated.

(3) Even if the Tx Shifter register is not being used, PSI5S is busy until asynchronous transfer ends.

(4) Even when the PSI5SjUCTS.UTTFIN signal is high level, interrupts and DMA requests are generated.

Items (1) and (2) above describe the conditions for storing data in the Tx shifter.

28.6.3.3 Abnormal Reception (Reception Error)

After UART reception ends, parity, framing, and overrun errors are detected if they exist.

If an overrun error occurs, the data which is read from the mailbox is unpredictable.

If an error occurs, an interrupt status (see **Table 28.104**) occurs. The interrupt status signal is cleared by writing 1 to the corresponding bit (see **Table 28.104**) in the interrupt status clear register.

Table 28.104 below describes reception errors that can occur during operations in UART mode.

Table 28.104 UART Communication Rx Errors

Error	Status	Status Clear	Detection Condition
Parity error	PSI5SjUCRS.UTRPE	PSI5SjUCRSC.UTRPECL	When UART stop bit is received, the received parity value does not match.
Framing error	PSI5SjUCRS.UTRFE	PSI5SjUCRSC.UTRFECL	When UART stop bit is received, the received stop bit value is 0.
Overrun error	PSI5SjUCRS.UTROE	PSI5SjUCRSC.UTROECL	When UART stop bit is received, a new data is received before the previous data is read.*1

Note 1. When the PSI5S module returns to UART mode after a transition to configuration mode, an overrun error is not detected even if a new data is received in a condition that a previous received data has not been read.

PSI5S controls the Rx-finish timing and various errors when the stop bit is sampled (midway between stop bits).

Figure 28.92 and **Figure 28.93** below show examples of operation in which an Rx overrun, framing, and parity error occur when reception ends.

If both parity and framing errors occur, occurrence of a framing error is assumed.

When no errors occur, the Rx-finish status signal is set to high level.

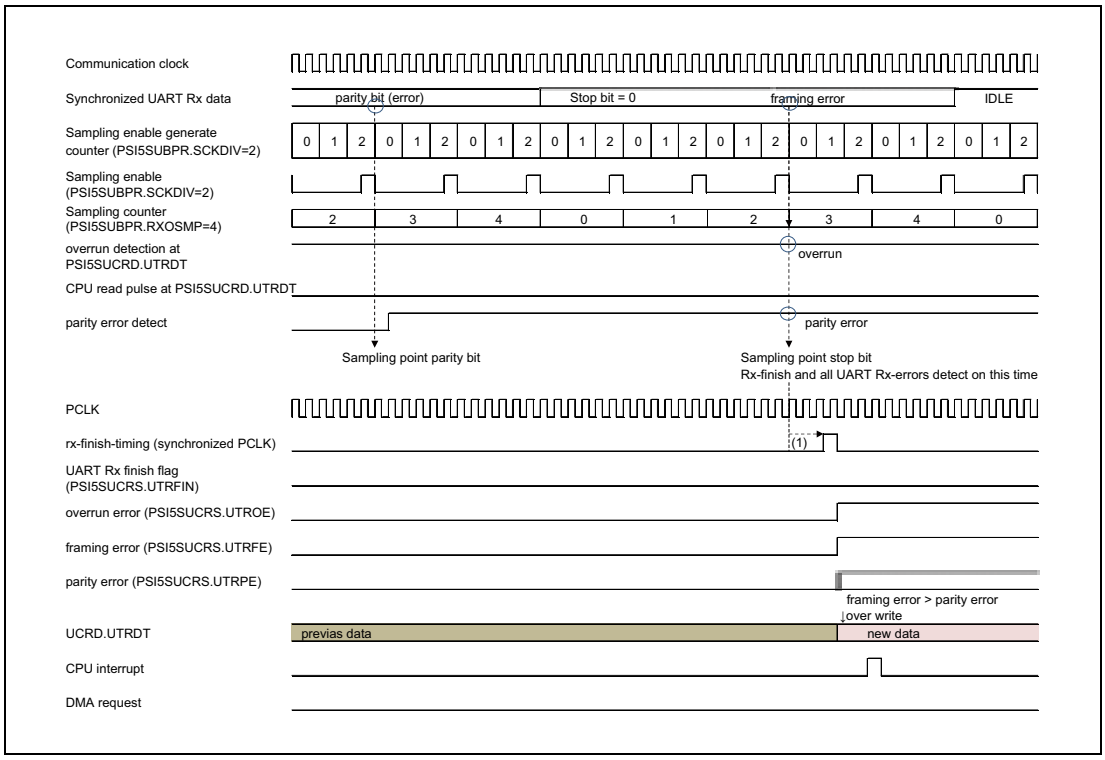


Figure 28.92 UART Rx (Received Error)

(1) The asynchronous transfer processing based on PCLK and the communication clock causes a delay of several clock cycles.

Figure 28.93 below describes an operation in which an overrun error is detected, as well as an example of an operation in which overrun error detection conflicts with receive data reading and an overrun error is not detected.

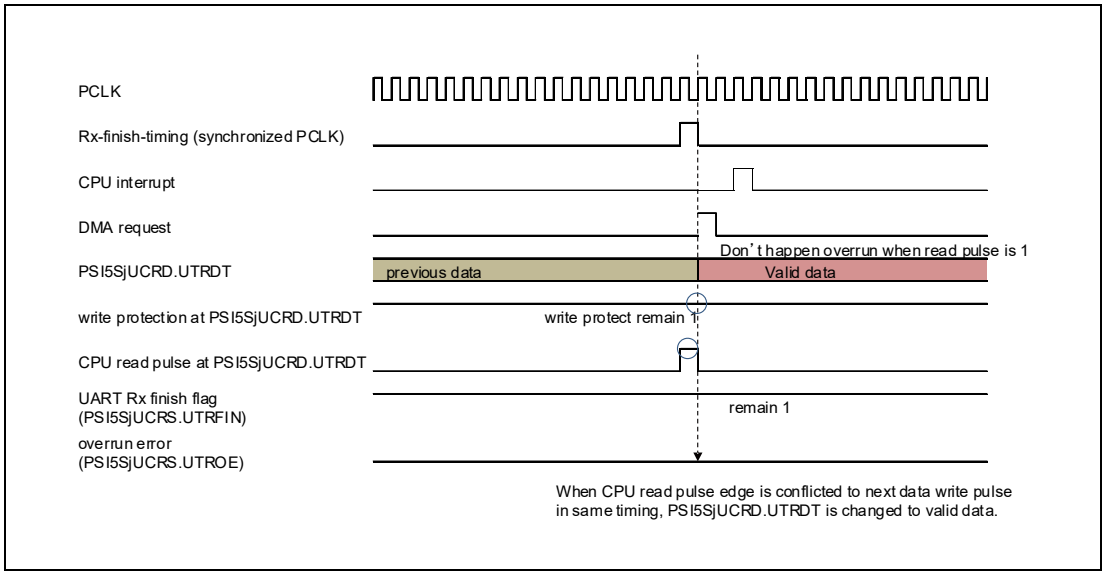


Figure 28.93 UART Rx (Overrun Error Is Not Detected)

The following figure shows an example of operation in which overrun error detection conflicts with receive data reading and an overrun error is detected.

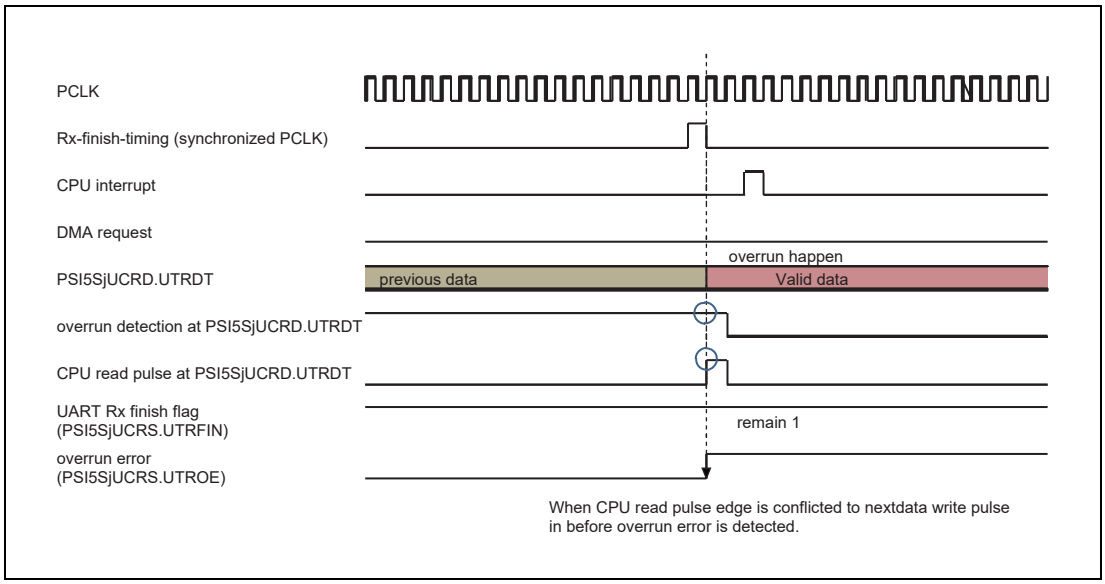


Figure 28.94 UART Rx (Overrun Error Is Detected)

28.6.3.4 Abnormal Transmission (Transmission Error)

This section describes the transmission errors that can occur in UART mode.

A transmission error is detected if new transmit data is set while PSI5S is busy with UART transmission.

When occurrence of an overwrite error is determined, the data written subsequently is not transmitted.

Already set data is not affected by the overwrite error, and the data is transmitted.

The interrupt status is caused by the occurrence of an overwrite error. The interrupt status can be cleared by writing 1 to the interrupt status clear register.

Table 28.105 UART Communication Tx Error

Error	Status	Status Clear	Detection Condition
Overwrite error	PSI5SjUCTS.UTTOWE	PSI5SjUCTSC.UTTOWECL	CPU writes (PSI5SjUCTD.UTTDT[7:0]), while PSI5SjUCTS.UTTBBF is asserted.

The following figure shows an example of operation in which transmission errors occur.

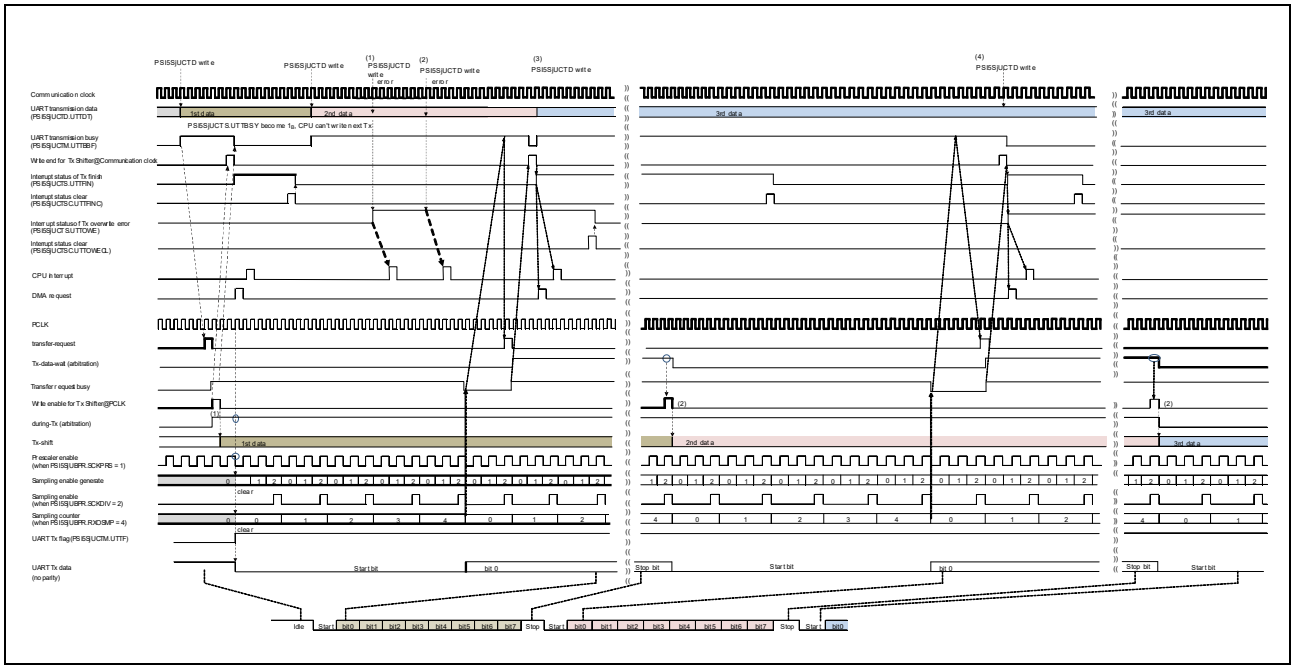


Figure 28.95 UART Tx (Overwrite Error)

(1) If data is written to the PSI5SjUCTD register with the PSI5SjUCTM.UTTBBF signal set to high level, the PSI5SjUCTM.UTTOWE signal is set to high level (indicating an overwrite error), and an interrupt is output.

(2) If data is written again to the PSI5SjUCTD register while the PSI5SjUCTS.UTTOWE signal is high level (indicating an overwrite error), an interrupt is outputs again.

(3) Even in the overwrite error status (PSI5SjUCTS.UTTOWE = high level), the PSI5SjUCTS.UTTFIN signal is set to high level and interrupts and DMA requests are output.

(4) This timing is a boundary for the occurrence of an overwrite error (PSI5SjUCTS.UTTOWE = high level). The overwrite error occurs together with transmission end (PSI5SjUCTS.UTTFIN = 1). In this case, both status signals are set to high level, and interrupts and DMA requests are output.

In configuration mode, even if data is written to the PSI5SjUCTD register, the PSI5S does not start the transmission and the PSI5S does not detect overwrite error.

28.6.3.5 Interrupt Output

PSI5S can output eight interrupt signals.

In UART mode, interrupts are output to channels 0 and 1.

Interrupts can be output only when interrupt output is enabled. If an interrupt factor occurs when interrupt output is disabled, no interrupt is output, but the interrupt status is indicated.

Table 28.106 below lists the factors of interrupts that are output in UART mode.

Table 28.106 Interrupt Output Timings

Output Signal	Interrupt Output Timing
int_psis_ch0	When UART data is received
int_psis_ch1	When UART data is stored in the Tx shifter.
int_psis_ch1	When UART data is written before the Tx shifter data is read.

Table 28.107 below describes the interrupt statuses and clearing of interrupt statuses.

To clear an interrupt status, write 1 to the bit corresponding to the interrupt status in the interrupt status clear register.

Table 28.107 Interrupt Status

Equipped Ch	Interrupt Status
Ch0	UART Rx parity error
Ch0	UART Rx framing error
Ch0	UART Rx overrun error
Ch0	UART Rx finish
Ch1	UART Tx overwrite error
Ch1	UART Tx finish

Figure 28.92, **Figure 28.93**, **Figure 28.94** and **Figure 28.95** show interrupt output operations.

28.6.3.6 DMA Request Output

PSI5S can output 15 DMA requests (eight receive requests and seven transmit requests).

In UART mode, DMA requests are output to channel 7 for reception and transmission.

If a request output factor occurs, one PCLK cycle of pulse signal outputs.

DMA requests are output only when DMA request output is enabled. DMA requests are not output when DMA request output is disabled.

Table 28.108 below lists the factors of DMA requests that are output in UART mode.

Table 28.108 Timings of DMA Transfer Request Output

Operation Mode	Equipped Ch	Rx/Tx	DMA Request Output Factor
UART mode	Ch7	Rx	When UART data is received (no error)
	Ch7	Tx	When UART data is stored in the Tx shifter

Figure 28.90 and **Figure 28.91** show the DMA request output operations.

28.6.3.7 Transition to Configuration Mode

The operation of PSI5S to return to the configuration mode is described in **Section 28.4, Operation Modes**.

Figure 28.96 below describes the transition to the configuration mode during reception and transmission.

When CPU writes PSI5SjPUOEB.OPEN to 0 during reception, PSI5S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5S isn't sending a frame.

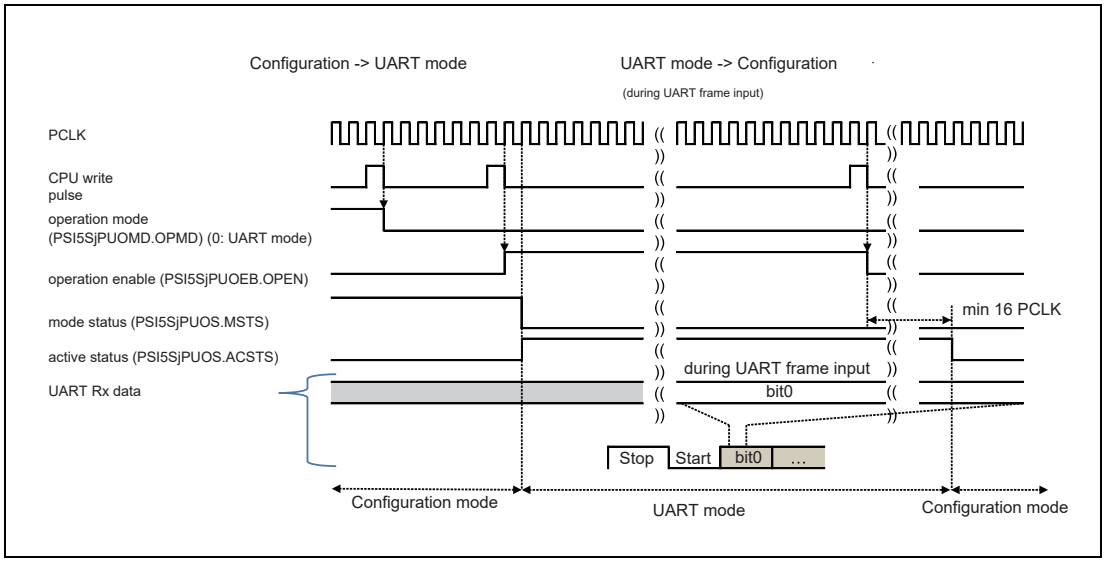


Figure 28.96 Move from UART Mode to Configuration Mode (Rx)

When CPU writes PSI5SjPUOEB.OPEN to 0 during transmission, PSI5S waits for 16 PCLK cycles and then transitions to configuration mode if the PSI5S isn't sending a frame. If PSI5S is sending a frame, it will transition to configuration mode after it completes to send the frame.

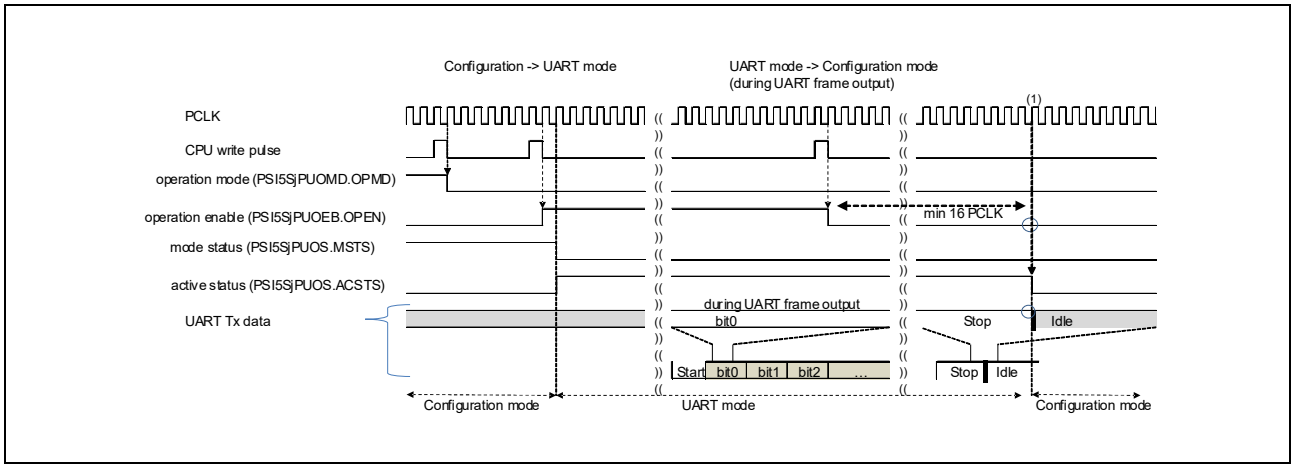


Figure 28.97 Move from UART Mode to Configuration Mode (Tx)

NOTE

In a case that the command to transceiver has been accepted in UART transmission module in PSI5S when CPU writes PSI5SjPUOEB.OPEN to 0, PSI5S wait for completion of the UART transmission. The maximum period of the mode transition time is a period of 2 UART frames.

28.6.4 Clearing Status Signals

Each status signal can be cleared by writing 1 to the corresponding bit in the status clear register. Note, however, that the following statuses are cleared automatically when their clearing conditions are met:

PSI5SjPUOS.SWSTS: Cleared when the software reset period has passed (**Figure 28.26**)

PSI5SjPUOS.ACSTS: Cleared when PSI5S returns to the configuration mode

PSI5SjPTFS.TXSTS: Cleared when the last byte of data is stored in the Tx shifter (**Figure 28.65**)

PSI5SjUCTM.UTTF: Cleared when stop bits are transferred in UART mode.

PSI5SjUCTM.UTTBBF: Cleared when transmit data is loaded into the transmit buffer in UART mode (**Figure 28.91**)

PSI5SjPDDS(1-7).DDSRSTS: Cleared when the last ECU-to-sensor data is stored in the Tx shifter in PSI5S mode (**Figure 28.68**) or when a DDSR transmission stop request is generated (**Figure 28.70**)

If setting and clearing conditions are both met at the same time, setting conditions have priority over clearing conditions.

28.7 Note

28.7.1 Notes

When the UART receive data pin is “Low” level at the transition to UART mode, the PSI5S may detect a UART parity error or a UART framing error.

28.7.2 Restrictions

- Baud rate tolerance must be within 1.5%.
- The minimum interval is 60 PCLK cycles from transition to configuration mode until set operation to enable (PSI5SjPUOEB.OPEN = 1).

Section 29 Renesas High-Speed Serial I/F (RHSIF)

This section contains a generic description of the Renesas high-speed Serial I/F (RHSIF). The first part in this section describes all this product specific properties, such as the number of units and register base addresses. The remainder of this section describes the functions and registers of the RHSIF.

29.1 Features RHSIF for RH850/U2A-EVA

29.1.1 Number of Units and Channels

This microcontroller has the following number of Renesas high-speed Serial I/F (RHSIF) units.

Table 29.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	—	—	—	—
Name	RHSIFn									

Note: Regarding the unit index “n” in this section, the individual RHSIF units are identified by the index “n”.

29.1.2 Register Base Addresses

RHSIFn base addresses are listed in the following table. RHSIFn register addresses are given as offsets from the base address in general.

Table 29.2 Register Base Addresses

Base Address Name	Base Address	Bus Group
<RHSIF0_L1_base>	FF00 0000_H	P-Bus Group 9
<RHSIF0_L2_base>	1000 0000_H	H-Bus Group 0

29.1.3 Clock Supply

Clock supply by and to RHSIFn is listed in the following table.

Table 29.3 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RHSIFn (L2)	H-Bus clock (Register access/Operation clock)	CLK_HBUS
RHSIFn (L1)	PCLK (Register access/Operation clock)	CLK_HBUS
	CLK_MOSC Communication clock (Internal)	CLK_MOSC
	CLK_LSB Communication clock (Internal)	CLK_LSB
	RHSIF0_REFCLK Communication clock (External)	HSIF0_REFCLK

29.1.4 Interrupt Requests and Error Notifications

RHSIFn interrupt requests are listed in the following table.

Table 29.4 Interrupt and DMA/DTS Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
RHSIF0					
INTRHSIF0TXCMP	RHSIFnTXCMP (n = 0)	Transmit complete interrupt	612	—	—
INTRHSIF0TXEX	RHSIFnTXERR (n = 0)	Transmit exception interrupt	613	—	—
INTRHSIF0RXCMP	RHSIFnRXCMP (n = 0)	Receive complete interrupt	614	—	—
INTRHSIF0RXEX	RHSIFnRXERR (n = 0)	Receive exception interrupt	615	—	—
INTRHSIF0ICLCR	RHSIFnRXICLC (n = 0)	ICLC receive interrupt	616	—	—
INTRHSIF0INTCH0	int_hsif_ch0	Channel 0 interrupt	617	Group1-120	Group2-40
INTRHSIF0INTCH1	int_hsif_ch1	Channel 1 interrupt	618	Group1-121	Group2-41
INTRHSIF0INTCH2	int_hsif_ch2	Channel 2 interrupt	619	Group1-122	Group2-42
INTRHSIF0INTCH3	int_hsif_ch3	Channel 3 interrupt	620	Group1-123	Group2-43
INTRHSIF0STR	int_hsif_str	Stream interrupt	621	—	—
INTRHSIF0ERR	int_hsif_err / int_hsif_sec	Error interrupt / Security interrupt	622	—	—

This module has no error notifications.

29.1.5 Reset Sources

RHSIFn reset sources are shown below. RHSIFn is initialized by the following reset sources.

Table 29.5 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RHSIFn	All registers	√	√	√	√	√	√	—

29.1.6 External Input/Output Signals

External input/output signals of RHSIF units are listed below.

Table 29.6 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RHSIF0		
RHSIF0_REFCLK	RHSIF0 Reference clock input/output	HSIF0_REFCLK
RHSIF0_RXDN	RHSIF0 receive data differential input	HSIF0_RXDN
RHSIF0_RXDP	RHSIF0 receive data differential input	HSIF0_RXDP
RHSIF0_TXDN	RHSIF0 transmission data differential output	HSIF0_TXDN
RHSIF0_TXDP	RHSIF0 transmission data differential output	HSIF0_TXDP

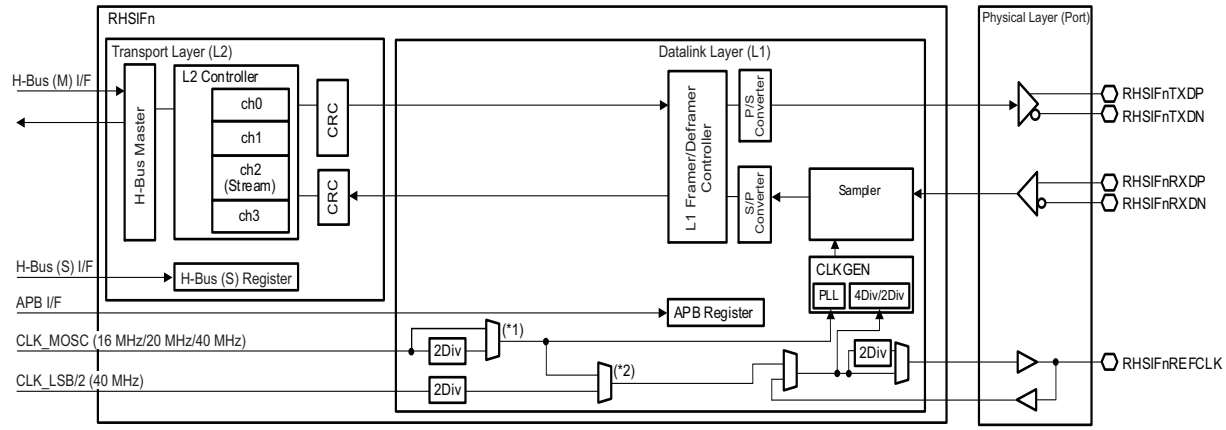
29.2 Overview

29.2.1 Functional Overview

- Four channels, including one channel with data streaming capability
- Bus master interface used by target node to access shared memory
- H-Bus bus master supporting “Write/Read” command
  - Writing a single 8/16/32 bit data value into the register of a target device
  - Reading a single data from 8/16/32 bit register of a target device
  - Support of 32-bit address range
- Built-in DMA controller supporting “Stream write” command
  - 128 or 256-bit data transfer on streaming channel
  - Up to two outstanding requests of “Stream write” are supported by the initiator
- Transfers protected by CRC16
- Programmable time outs for detection of blocked answer transfers
- Automatic frame transfer ID generation for detection of dropped frames
- Fixed priority channel selection
- Remote trigger of event/interrupt in the target device by the initiator
- Access protection from an external master
  - Can configure four access windows
  - Can configure of access rule (Prohibited, R, W, RW) at each window
- Security function
  - Authentication at the time of initial communication
  - ID authentication
  - Challenge and response authentication
- Point-to-point high speed serial communication between two devices
- Full duplex communication
- Supports dual mode (register configurable Master/Slave).
- Supports asynchronous data transfer at data rates up to a maximum of 320 M Baud
- Transmits and receives data, CTS, ICLC and unsolicited frames
- Supports automatic ping response generation in slave mode
- Supports detection of unsupported channel number and unsupported payload sizes
- The interface is based on IEEE 1596.3-1996 reduced range link LVDS IOs
- Built-in self test functions

29.2.2 Block Diagram

The following figure shows a top level block diagram of the RHSIF module.



Note 1. When CLK_MOSC frequency is 16 MHz / 20 MHz, it selects CLK_MOSC.
When CLK_MOSC frequency is 40 MHz, it selects CLK_MOSC divided into 2: CLK_MOSC/2

Note 2. When CLK_MOSC frequency is 20 MHz / 40 MHz, it selects clock outputted from **Note 1.**: CLK_MOSC or CLK_MOSC/2.
When CLK_MOSC frequency is 16 MHz, it selects CLK_LSB divided into 2: CLK_LSB/2

Figure 29.1 Module Block Diagram

CAUTION

When CLK_MOSC frequency is 24MHz, RHSIF can NOT operate normally.

29.2.3 Communication Protocol

29.2.3.1 Protocol Definitions

Table 29.7 Protocol Definitions

Item	Definition	Comment
Master	The module that supplies the clock and controls Link initialization on the Slave interface.	L1
Slave	The module that receives the clock.	L1
Initiator	The module that sends an L2 frame with either the Write, Read, Stream, Event, or ID command	L2
Target	The module that sends an L2 frame with either the Read Answer, ACK, or NACK command	L2
TxLink	Point-to-point connection executing transfer from Master to Slave.	L1
RxLink	Point-to-point connection executing transfer from Slave to Master.	L1

29.2.3.2 Connection

The **Figure 29.2** shows the connection between master and slave of RHSIF.

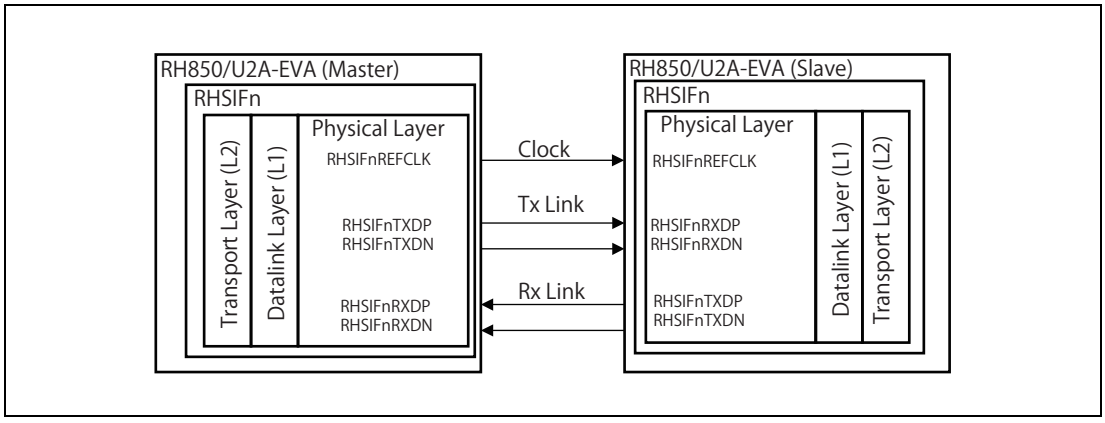
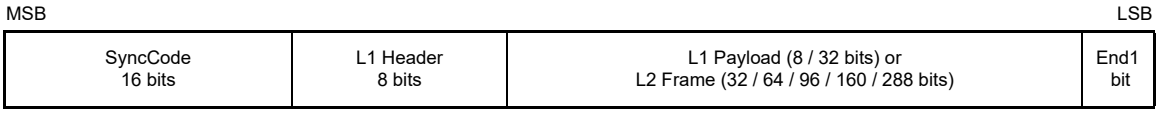


Figure 29.2 Connection between Master and Slave of RHSIF

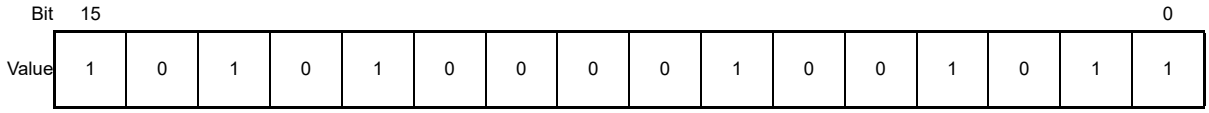
29.2.3.3 L1 Frame

The L1 Frame Format is shown below.



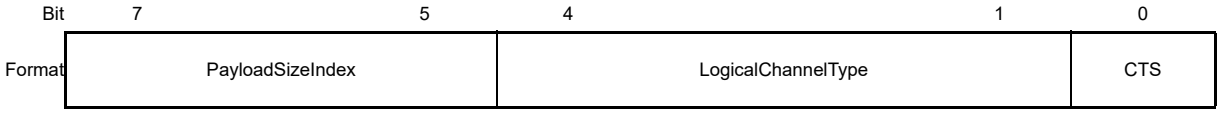
(1) SyncCode

The value in this field is assigned to SyncCode, and the value is fixed in $A84B_H$



(2) L1 Header Format

The value in this field is assigned to L1 Header.



(a) List of Payload Size Index

The **Table 29.8**. shows the details of the payload size index values.

Table 29.8 List of Payload Size Index

Payload Size Index	Payload Size	Total Frame Size*1	Comment
000	8	32	Interface Control
001	32	56	—
010	64	88	—
011	96	120	—
100	128	152	Not used
101	256	280	Not used
110	160	184	—
111	288	312	—

Note 1. “Endbit” is excluded from Total Frame Size.

(b) List of Logical Channel Types

The **Table 29.9** shows the details of Logical Channel Type values.

Table 29.9 List of Logical Channel Types

Logical Channel Type	Logical Channel Type Master	Logical Channel Type Slave
0000	Interface Control	Interface Control PING answer (32 bit value)
0001	Reserved for future use	
0010	Reserved for future use	
0011	CTS frame for flow control	
0100	Data Channel A	
0101	Data Channel B	
0110	Data Channel C	
0111	Data Channel D	
1000-1111	Reserved for future use	

(c) List of CTS

The **Table 29.10** shows the details of CTS values.

Table 29.10 List of CTS

CTS	Function
0	BackPressure
1	Ready

(3) L1 Payload

The value in this field is assigned to L1 Payload as ICLC Command.

(a) List of ICLC Commands

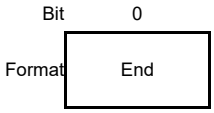
The **Table 29.11** shows the details of ICLC command values.

Table 29.11 List of ICLC Commands

Value (hex)	Function
00_H	PING (Sends by master interface. Slave sends back a fixed 32-bit payload result ($ABCDEF01_H$))
02_H	Slave interface PLL start (in preparation for high speed mode)
04_H	Slave interface PLL stop (after fallback from high speed mode)
08_H	Selects the Slow Speed mode for transfers from the Master interface to the Slave interface
10_H	Selects the Fast Speed mode for transfers from the Master interface to the Slave interface
20_H	Selects the Slow Speed mode for transfers from the Slave interface to the Master interface
80_H	Selects the Fast Speed mode for transfers from the slave interface to the master interface
31_H	Enables the Slave interface transmitter
32_H	Disables the Slave interface transmitter
34_H	Turn on test mode (Send 101010 .. on RX line continuously using actual configured RX line data rate; cancelled by issuing “test mode off” command.)
38_H	Turns off the test mode (Cancels the clock test mode and payload loopback)
FF_H	Turns on the payload loopback (Incoming frames at Slave interface are looped back until cancelled by a Frame containing “test mode off”.)

(4) Endbit

The value in this field is assigned to Endbit.



(a) List of Endbit

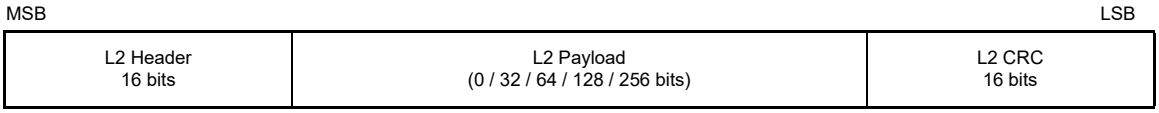
The **Table 29.12** shows the details of the Endbit values.

Table 29.12 List of Endbit

Endbit	Function
0	Normal
1	Sleep

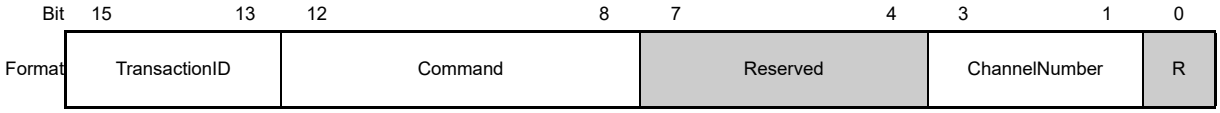
29.2.3.4 L2 Frame

The L2 Frame Format is shown below.



(1) L2 Header Format

The value in this field is assigned to L2 Header.



(a) Transaction ID

The value in this field is assigned to the outgoing frame. Afterwards, it is automatically increased and the new value is assigned to the next outgoing frame. It wraps around.

(b) List of L2 Commands

The **Table 29.13** shows the details of the L2 Command values.

Table 29.13 List of L2 Commands

Header Bit Number					Command
12	11	10	9	8	
0	0	0	0	0	Reads 8 bits
0	0	0	0	1	Reads 16 bits
0	0	0	1	0	Reads 32 bits
0	0	0	1	1	Reserved
0	0	1	0	0	Writes 8 bits
0	0	1	0	1	Writes 16 bits
0	0	1	1	0	Writes 32 bits
0	0	1	1	1	Reserved
0	1	0	0	0	ACK
0	1	0	0	1	NACK (ACK Error)
0	1	0	1	0	Read Answer
0	1	0	1	1	Reserved
0	1	1	0	0	Event Command (Trigger)
0	1	1	0	1	Reserved
0	1	1	1	0	Reserved
0	1	1	1	1	Reserved
1	0	0	0	0	Reserved
1	0	0	0	1	Reserved
1	0	0	1	0	ID Read (Read JTAG ID)*1
1	0	0	1	1	Reserved
1	0	1	0	0	Reserved
1	0	1	0	1	Reserved
1	0	1	1	0	Reserved
1	0	1	1	1	Stream
1	1	0	0	0	Reserved
1	1	0	0	1	Reserved
1	1	0	1	0	Reserved
1	1	0	1	1	Reserved
1	1	1	0	0	Reserved
1	1	1	0	1	Reserved
1	1	1	1	0	Reserved
1	1	1	1	1	Reserved

Note 1. Refer to **Section 53, Boundary Scan**, **Table 53.5** for details.

(c) List of Channel Number

The **Table 29.14** shows the details of the Channel Number values.

Table 29.14 List of Channel Number

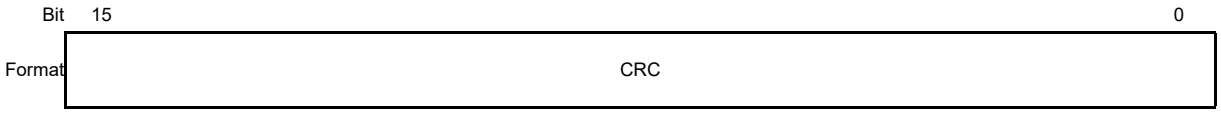
HSSL Channel Name	Channel Number Coding in Header (s)			Comment
	HSSL Header (Binary)		Data Link Layer Header (Binary)	
	Code Table I	Code Table II		
0 (channel A)	000	100	0100	In use
1 (channel B)	001	101	0101	In use
2 (channel C)	010	110	0110	In use
3 (channel D)	011	111	0111	In use
4 (channel E)	100	000	1000	Reserved
5 (channel F)	101	001	1001	Reserved
6 (channel G)	110	010	1010	Reserved
7 (channel H)	111	011	1011	Reserved

(2) L2 Payload

The value in this field is assigned to actual data.

(3) L2 CRC

The value in this field is assigned to CRC.



The last field of the L2 frame contains the CRC field. To setup the CRC value, this protocol uses the following algorithm:

- CRC-16-CCITT polynomial x16 + x12 + x5 + 1

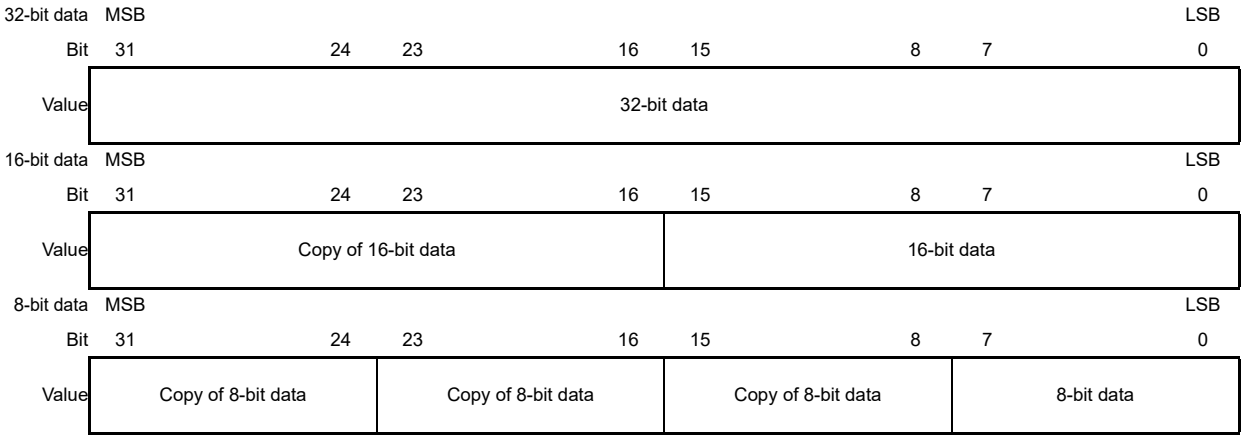
with the following standard properties:

- seed: $FFFF_H$
- calculation direction: Most significant bit first (most significant bit to least significant bit in the header, and then the most significant bit to least significant bit in the payload)
- CRC result direction: Most significant bit first

29.2.3.5 Data Format

(1) Non Stream data format

Data is mapped into a 32-bit wide frame for 8 to 16 bits with no specific data frames defined. The following pictures show the mapping of 8/16/32 bit data into the 32 bit wide data frame.

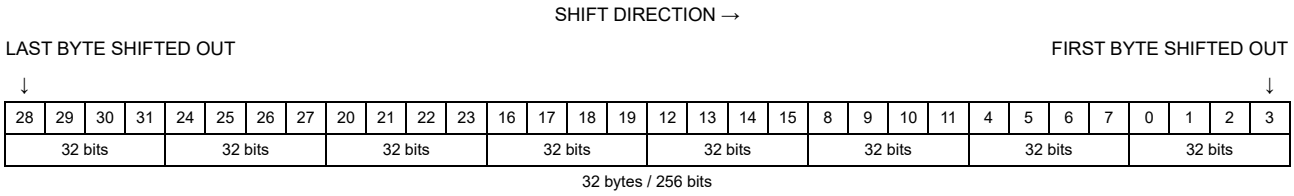


(2) Stream data format

(a) 256-bit Data Stream Format

Data is transmitted in a stream frame as a sequence of 8 times a 32-bit value. The initiator and target modules must be able to accept the data sequence in this format.

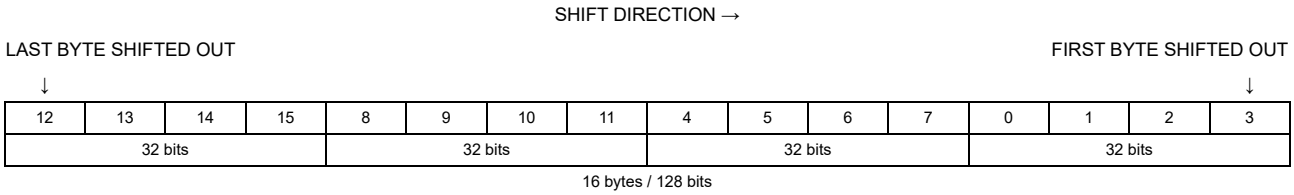
The bytes are sent with the most significant bit sent first.



(b) 128-bit Data Stream Format

Data is transmitted in a stream frame as a sequence of 4 times a 32-bit value. The initiator and target modules must be able to accept the data sequence in this format.

The bytes are sent with the most significant bit sent first.



29.2.3.6 All Sending and Receiving Frames

The **Figure 29.3** shows the all sending and receiving frames.

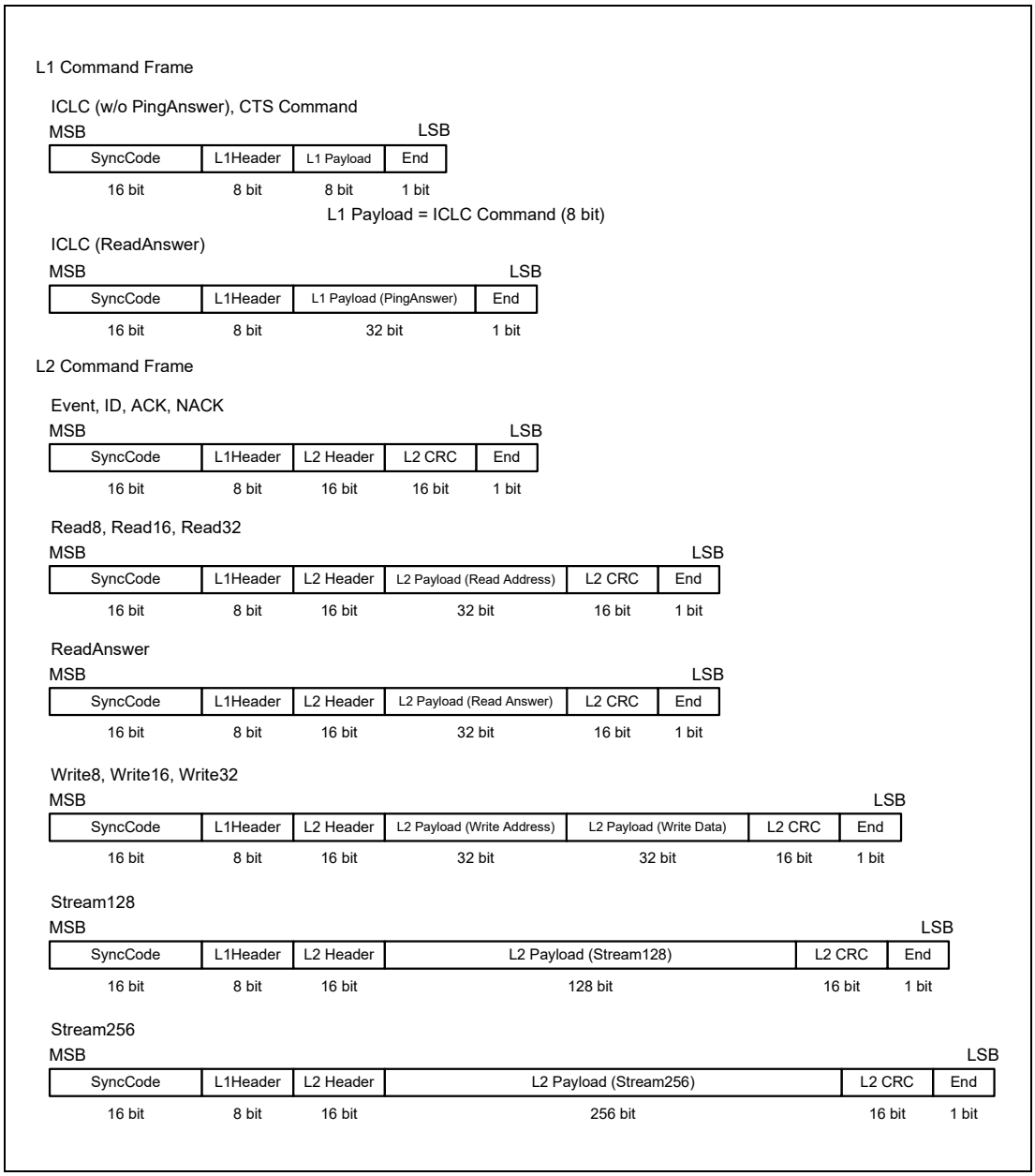


Figure 29.3 All Sending and Receiving Frames

29.2.3.7 Number of Transfer Bits in each Frame

Table 29.15 Number of Transfer Bits in each Frame List

#	Layer	L1			L2			L1	Total Bit	Command	Direction
		Sync	Header	Payload	Header	Payload	CRC	END			
1	L1	16	8	8	—			1	33	ICLC (w/o PingAnswer)	Master → Slave
2										CTS*1	
3				32					57	ICLC PingAnswer	Slave → Master
4	L2	16	8	—	16	—	16	1	57	ID, Event	Initiator → Target
5										ACK, NACK	Target → Initiator
6						32			89	Read8, Read16, Read32	Initiator → Target
7										ReadAnswer	Target → Initiator
8						64			121	Write8, Write16, Write32	Initiator → Target
9						128			185	Stream128	Initiator → Target
10						256			313	Stream256	Initiator → Target

Note 1. payload size is fixed to 8-bit data when transmitting and can be any size when receiving.

29.2.3.8 List of Command and Response

Table 29.16 List of Command and Response

#	Layer	Command Issue	Response	
			Normal	Abnormal
1	L1	Ping	PingAnswer	NoResponse
2	L2	ID	ReadAnswer	NACK
3		Event	ACK	NoResponse
4		Read8, Read16, Read32	ReadAnswer	NACK
5		Write8, Write16, Write32	ACK	NACK
6		Stream128, Stream256	ACK	NACK

29.2.3.9 Command Pipelining

The protocol does not allow pipelining of commands. The initiator shall not send any new command before the target has returned an answer (ACK, NACK, Read Answer). If the target receives a command before it is able to answer the previous command, it shall ignore the command.

There is one exception to this rule. Stream Commands can be pipelined. This feature is only available for Stream Commands. The initiator is allowed to send the next command before it has received an answer from the target linked per Transaction ID to the command sent before. The pipeline depth is limited to a sequence of two commands in the current protocol.

In summary, the initiator is allowed to send a Stream Command, which can be immediately followed by a second Stream Command. After the second command, the initiator has to stop and wait for an answer from the target.

The target shall be capable of receiving pipelined Stream Commands up to a pipeline depth of two Stream Commands.

29.3 Operation

29.3.1 Operation Flow

The RHSIF can perform data transmission/reception after the initial setting is configured according to the setup flow shown in **Figure 29.4** below.

For details about the initial setting for L2 and L1, refer to **Section 29.4, Transport Layer (L2)** and **Section 29.5, Datalink Layer (L1)**, respectively.

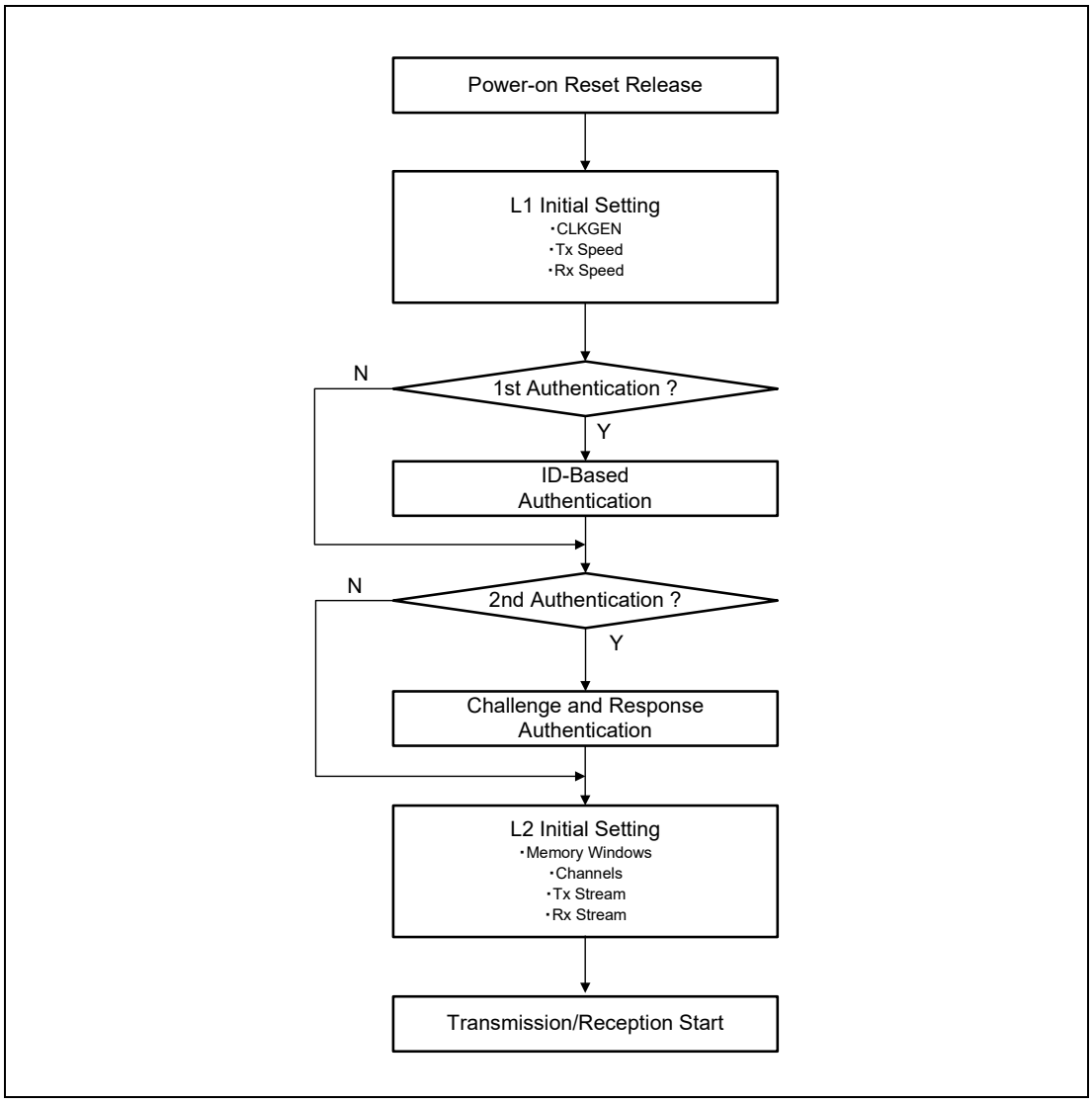


Figure 29.4 Operation Flow

29.4 Transport Layer (L2)

29.4.1 Overview

29.4.1.1 Functional Overview

The Renesas High-speed Serial Interface Transport Layer IP (hereinafter referred to as “L2”) is compliant with following specifications.

L2 supports following features / functions.

- Supports 4 channels
  - Supports stream transfer on channel 2
  - Channel arbitration: Fixed (Higher) Ch.0 → Ch.1 → Ch.2 → Ch.3 (lower)
- Generates header
  - Manages transaction-ID
  - Supports both channel number code table I and II
- Generates and validates CRC-16
- Supports both initiator node and target node
- Initiator node function
  - Generates and transmit following request Commands
    - Read Command (8 bits, 16 bits, 32 bits)
    - Write Command (8 bits, 16 bits, 32 bits)
    - Event Command
    - ID Command
    - Stream Command (256 bits, 128 bits)
  - Number of outstanding requests: Non-Stream = 1 / channel, Stream = 2 (channel 2 only)
  - Detects reply timeout
    - The timeout value is programmable by register setting
  - Implements the DMA controller for stream transmission
- Target node function
  - Number of acceptable requests: Non-Stream = 1 / channel, Stream = 2 (channel 2 only)
  - Translates HSSL Commands to H-Bus master requests
    - Refer to **Section 44, Functional Safety** about the H-Bus SPID setting.
  - Supports the access protect function from an external master
    - Supports 4 areas, independent of channel
      - Can configure access rule (prohibited, read only, write only, read/write)
      - Can configure memory access window
    - Supports special authentication based on the security policy of this product
      - 256-bit ID authentication
      - “Challenge and response” authentication

– Generates and transmits the following reply Commands as target node
  - ACK/NACK Command
  - Read Answer Command
– Implements the DMA controller for stream reception
– Generates interrupt by receiving Event Command

29.4.1.2 Block Diagram

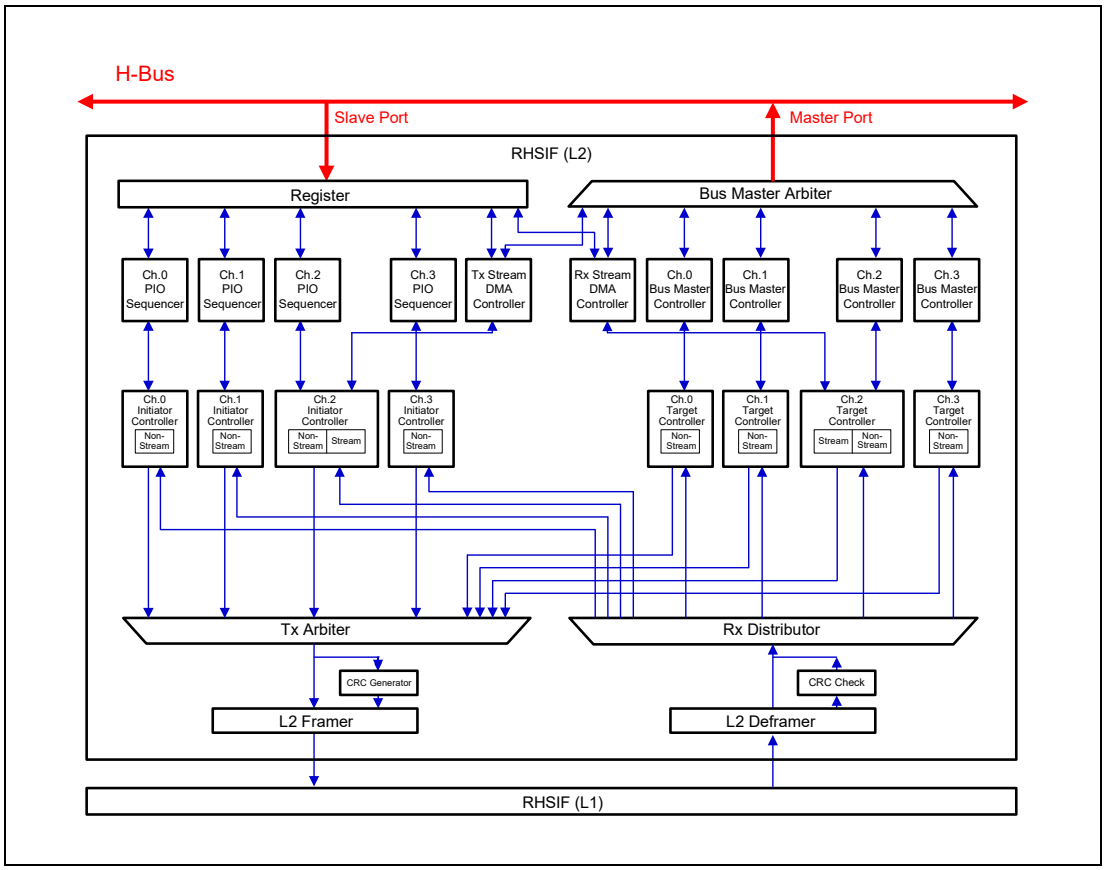


Figure 29.5 Block Diagram of L2

29.4.2 Registers

This section describes the L2 registers in detail.

In case of a concurrent update of the same register bit, L2 module write (hardware write) has higher priority than CPU write (software write).

29.4.2.1 List of Registers

The register map of an L2 module is shown in **Table 29.17**.

The L2 module remains in reset while RESET.

The values shown in **Table 29.17** in the registers description under “Values after Reset” are related to this reset state.

For further information about the L2 operation modes, refer to **Section 29.4.3, Operation**.

The specific base address of a module for an L2 module needs to be added to each of the specified offset addresses.

Table 29.17 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					HBG	Other
RHSIFn	Module mode register	RHSIFnMMD	<RHSIFn_L2_base> + 000_H	32, 16, 8	HBG91	-
	Module control register	RHSIFnMCT	<RHSIFn_L2_base> + 008_H	32, 16, 8	HBG91	(*1)
	Module status register	RHSIFnMST	<RHSIFn_L2_base> + 010_H	32, 16, 8	HBG91	-
	Module interrupt status register	RHSIFnMIST	<RHSIFn_L2_base> + 018_H	32, 16, 8	HBG91	-
	Module reply timeout time register	RHSIFnMRT	<RHSIFn_L2_base> + 020_H	32, 16, 8	HBG91	-
	Module memory window A start address register	RHSIFnMWAA	<RHSIFn_L2_base> + 040_H	32, 16, 8	HBG91	(*1)
	Module memory window A size register	RHSIFnMWAS	<RHSIFn_L2_base> + 048_H	32, 16, 8	HBG91	(*1)
	Module memory window B start address register	RHSIFnMWBA	<RHSIFn_L2_base> + 050_H	32, 16, 8	HBG91	(*1)
	Module memory window B size register	RHSIFnMWBS	<RHSIFn_L2_base> + 058_H	32, 16, 8	HBG91	(*1)
	Module memory window C start address register	RHSIFnMWCA	<RHSIFn_L2_base> + 060_H	32, 16, 8	HBG91	(*1)
	Module memory window C size register	RHSIFnMWCS	<RHSIFn_L2_base> + 068_H	32, 16, 8	HBG91	(*1)
	Module memory window D start address register	RHSIFnMWDA	<RHSIFn_L2_base> + 070_H	32, 16, 8	HBG91	(*1)
	Module memory window D size register	RHSIFnMWDS	<RHSIFn_L2_base> + 078_H	32, 16, 8	HBG91	(*1)
	Stream Tx mode register	RHSIFnSTMD	<RHSIFn_L2_base> + 200_H	32, 16, 8	HBG91	-
	Stream Tx control register	RHSIFnSTCT	<RHSIFn_L2_base> + 208_H	32, 16, 8	HBG91	-
	Stream Tx status register	RHSIFnSTST	<RHSIFn_L2_base> + 210_H	32, 16, 8	HBG91	-
	Stream Tx status clear register	RHSIFnSTSC	<RHSIFn_L2_base> + 218_H	32, 16, 8	HBG91	-
	Stream Tx interrupt enable register	RHSIFnSTIE	<RHSIFn_L2_base> + 220_H	32, 16, 8	HBG91	-
	Stream Tx source address register	RHSIFnSTSA	<RHSIFn_L2_base> + 228_H	32, 16, 8	HBG91	-
	Stream Tx byte count register	RHSIFnSTBC	<RHSIFn_L2_base> + 230_H	32, 16, 8	HBG91	-
	Stream Rx mode register	RHSIFnSRMD	<RHSIFn_L2_base> + 280_H	32, 16, 8	HBG91	-
	Stream Rx control register	RHSIFnSRCT	<RHSIFn_L2_base> + 288_H	32, 16, 8	HBG91	-

Table 29.17 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					HBG	Other
RHSIFn	Stream Rx status register	RHSIFnSRST	<RHSIFn_L2_base> + 290_H	32, 16, 8	HBG91	-
	Stream Rx status clear register	RHSIFnSRSC	<RHSIFn_L2_base> + 298_H	32, 16, 8	HBG91	-
	Stream Rx interrupt enable register	RHSIFnSRIE	<RHSIFn_L2_base> + $2A0_H$	32, 16, 8	HBG91	-
	Stream Rx destination area start address register	RHSIFnSRDA	<RHSIFn_L2_base> + $2A8_H$	32, 16, 8	HBG91	-
	Stream Rx destination area size register	RHSIFnSRDS	<RHSIFn_L2_base> + $2B0_H$	32, 16, 8	HBG91	-
	Stream Rx byte count register	RHSIFnSRBC	<RHSIFn_L2_base> + $2B8_H$	32, 16, 8	HBG91	-
	Stream Rx write pointer register	RHSIFnSRWP	<RHSIFn_L2_base> + $2C0_H$	32, 16, 8	HBG91	-
	Stream Rx read pointer register	RHSIFnSRRP	<RHSIFn_L2_base> + $2C8_H$	32, 16, 8	HBG91	-
	Authentication ID data register 0	RHSIFnAID0	<RHSIFn_L2_base> + 300_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 1	RHSIFnAID1	<RHSIFn_L2_base> + 308_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 2	RHSIFnAID2	<RHSIFn_L2_base> + 310_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 3	RHSIFnAID3	<RHSIFn_L2_base> + 318_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 4	RHSIFnAID4	<RHSIFn_L2_base> + 320_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 5	RHSIFnAID5	<RHSIFn_L2_base> + 328_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 6	RHSIFnAID6	<RHSIFn_L2_base> + 330_H	32, 16, 8	HBG91	(*2)
	Authentication ID data register 7	RHSIFnAID7	<RHSIFn_L2_base> + 338_H	32, 16, 8	HBG91	(*2)
	Authentication answer data register 0	RHSIFnAAD0	<RHSIFn_L2_base> + 340_H	32, 16, 8	HBG91	(*3)
	Authentication answer data register 1	RHSIFnAAD1	<RHSIFn_L2_base> + 348_H	32, 16, 8	HBG91	(*3)
	Authentication answer data register 2	RHSIFnAAD2	<RHSIFn_L2_base> + 350_H	32, 16, 8	HBG91	(*3)
	Authentication answer data register 3	RHSIFnAAD3	<RHSIFn_L2_base> + 358_H	32, 16, 8	HBG91	(*3)
	Authentication response data register 0	RHSIFnARD0	<RHSIFn_L2_base> + 360_H	32, 16, 8	HBG91	(*2)
	Authentication response data register 1	RHSIFnARD1	<RHSIFn_L2_base> + 368_H	32, 16, 8	HBG91	(*2)
	Authentication response data register 2	RHSIFnARD2	<RHSIFn_L2_base> + 370_H	32, 16, 8	HBG91	(*2)
	Authentication response data register 3	RHSIFnARD3	<RHSIFn_L2_base> + 378_H	32, 16, 8	HBG91	(*2)
	Authentication error status register	RHSIFnAEST	<RHSIFn_L2_base> + $3E0_H$	32, 16, 8	HBG91	-
	Authentication error status clear register	RHSIFnAESC	<RHSIFn_L2_base> + $3E8_H$	32, 16, 8	HBG91	-
	Authentication error interrupt enable register	RHSIFnAEIE	<RHSIFn_L2_base> + $3F0_H$	32, 16, 8	HBG91	-
	Channel 0 mode register	RHSIFnCMD0	<RHSIFn_L2_base> + 400_H	32, 16, 8	HBG91	-
	Channel 0 control register	RHSIFnCCT0	<RHSIFn_L2_base> + 408_H	32, 16, 8	HBG91	-
	Channel 0 status register	RHSIFnCST0	<RHSIFn_L2_base> + 410_H	32, 16, 8	HBG91	-
	Channel 0 status clear register	RHSIFnCSC0	<RHSIFn_L2_base> + 418_H	32, 16, 8	HBG91	-
	Channel 0 interrupt enable register	RHSIFnCIE0	<RHSIFn_L2_base> + 420_H	32, 16, 8	HBG91	-
	Channel 0 read/write address register	RHSIFnCAR0	<RHSIFn_L2_base> + 428_H	32, 16, 8	HBG91	-

Table 29.17 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					HBG	Other
RHSIFn	Channel 0 write data register	RHSIFnCWD0	<RHSIFn_L2_base> + 430_H	32, 16, 8	HBG91	-
	Channel 0 read data register	RHSIFnCRD0	<RHSIFn_L2_base> + 438_H	32, 16, 8	HBG91	-
	Channel 1 mode register	RHSIFnCMD1	<RHSIFn_L2_base> + 480_H	32, 16, 8	HBG91	-
	Channel 1 control register	RHSIFnCCT1	<RHSIFn_L2_base> + 488_H	32, 16, 8	HBG91	-
	Channel 1 status register	RHSIFnCST1	<RHSIFn_L2_base> + 490_H	32, 16, 8	HBG91	-
	Channel 1 status clear register	RHSIFnCSC1	<RHSIFn_L2_base> + 498_H	32, 16, 8	HBG91	-
	Channel 1 interrupt enable register	RHSIFnCIE1	<RHSIFn_L2_base> + $4A0_H$	32, 16, 8	HBG91	-
	Channel 1 read/write address register	RHSIFnCAR1	<RHSIFn_L2_base> + $4A8_H$	32, 16, 8	HBG91	-
	Channel 1 write data register	RHSIFnCWD1	<RHSIFn_L2_base> + $4B0_H$	32, 16, 8	HBG91	-
	Channel 1 read data register	RHSIFnCRD1	<RHSIFn_L2_base> + $4B8_H$	32, 16, 8	HBG91	-
	Channel 2 mode register	RHSIFnCMD2	<RHSIFn_L2_base> + 500_H	32, 16, 8	HBG91	-
	Channel 2 control register	RHSIFnCCT2	<RHSIFn_L2_base> + 508_H	32, 16, 8	HBG91	-
	Channel 2 status register	RHSIFnCST2	<RHSIFn_L2_base> + 510_H	32, 16, 8	HBG91	-
	Channel 2 status clear register	RHSIFnCSC2	<RHSIFn_L2_base> + 518_H	32, 16, 8	HBG91	-
	Channel 2 interrupt enable register	RHSIFnCIE2	<RHSIFn_L2_base> + 520_H	32, 16, 8	HBG91	-
	Channel 2 read/write address register	RHSIFnCAR2	<RHSIFn_L2_base> + 528_H	32, 16, 8	HBG91	-
	Channel 2 write data register	RHSIFnCWD2	<RHSIFn_L2_base> + 530_H	32, 16, 8	HBG91	-
	Channel 2 read data register	RHSIFnCRD2	<RHSIFn_L2_base> + 538_H	32, 16, 8	HBG91	-
	Channel 3 mode register	RHSIFnCMD3	<RHSIFn_L2_base> + 580_H	32, 16, 8	HBG91	-
	Channel 3 control register	RHSIFnCCT3	<RHSIFn_L2_base> + 588_H	32, 16, 8	HBG91	-
	Channel 3 status register	RHSIFnCST3	<RHSIFn_L2_base> + 590_H	32, 16, 8	HBG91	-
	Channel 3 status clear register	RHSIFnCSC3	<RHSIFn_L2_base> + 598_H	32, 16, 8	HBG91	-
	Channel 3 interrupt enable register	RHSIFnCIE3	<RHSIFn_L2_base> + $5A0_H$	32, 16, 8	HBG91	-
	Channel 3 read/write address register	RHSIFnCAR3	<RHSIFn_L2_base> + $5A8_H$	32, 16, 8	HBG91	-
	Channel 3 write data register	RHSIFnCWD3	<RHSIFn_L2_base> + $5B0_H$	32, 16, 8	HBG91	-
	Channel 3 read data register	RHSIFnCRD3	<RHSIFn_L2_base> + $5B8_H$	32, 16, 8	HBG91	-

Note: Addresses other than those listed above are reserved. Do not access reserved addresses.

Note 1. Data cannot be written to these bits from the IP's master port. When the RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices. Initial value depends on specified setting. For details, refer to each section.

Note 2. Data can be read from or written to these bits only via the master port of RHSIF. When another master device such as a CPU, accesses these bits, these bits have the read-only (R) attribute.

Note 3. Data can be read from or written to these bits only from ICUMHA. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.

The access size mentioned in **Table 29.17** is required for write accesses. In addition all registers are readable by 8, 16 and 32 bit access.

The user should not access the registers marked as reserved in **Table 29.17**.

The user should not write bits to a value specified as "invalid" in the description of the registers.

29.4.2.2 Legend

This section explains the module state dependent abbreviations used for the L2 registers description.

Conditions:

- R/W: Bit is readable and writable
- R: Read-only bit; the user cannot write to this bit
- W: Write-only bit; Read value is always “0”

Indexes used for register and register Bit Names

The prefix of the "HSIFn" bit name is omitted.

M: The character “(M)” means “0”, “1”, “2”, “3”, “4”, “5”, “6” or “7”.

N: The character “(N)” means “0”, “1”, “2”, or “3”.

X: The character “(X)” means “A”, “B”, “C”, or “D”.

Reserved bits:

- Always read as 0
- Writing to these bits has no effect (hardware protection)

29.4.2.3 Detailed Register Descriptions

Detailed specifications of each register are described below.

(1) RHSIFnMMD — Module Mode Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHCT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.18 RHSIFnMMD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CHCT	This bit specifies the type of the channel number code table to used for L2. Note that this bit can only be changed at the time of initial setting. You cannot change this setting dynamically during system operation. 0: Table II (Ch.0 = 100_B, Ch.1 = 101_B, Ch.2 = 110_B, Ch.3 = 111_B) 1: Table I (Ch.0 = 000_B, Ch.1 = 001_B, Ch.2 = 010_B, Ch.3 = 011_B)

(2) RHSIFnMCT — Module Control Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	WDWE	WDRE	—	—	WCWE	WCRE	—	—	WBWE	WBRE	—	—	WAWE	WARE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	*1	*1	R	R	*1	*1	R	R	*1	*1	R	R	*1	*1

Note 1. Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices.

Table 29.19 RHSIFnMCT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	WDWE	This bit specifies whether to enable the write operation for memory window D. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
12	WDRE	This bit specifies whether to enable the read operation for memory window D. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	WCWE	This bit specifies whether to enable the write operation for memory window C. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
8	WCRE	This bit specifies whether to enable the read operation for memory window C. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	WBWE	This bit specifies whether to enable the write operation for memory window B. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).

Table 29.19 RHSIFnMCT Register Contents (2/2)

Bit Position	Bit Name	Function
4	WBRE	This bit specifies whether to enable the read operation for memory window B. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WAWE	This bit specifies whether to enable the write operation for memory window A. 0: Disables writing. 1: Enables writing. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).
0	WARE	This bit specifies whether to enable the read operation for memory window A. 0: Disables reading. 1: Enables reading. You can change this bit only when the initiator node function and target node function of all channels are disabled (RHSIFnCMD(N).INME(N)=0, TNME(N)=0).

(3) RHSIFnMST — Module Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AUTS1	AUTS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.20 RHSIFnMST Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	AUTS1	This bit indicates the status of the second authentication of the link partner. 0: Second authentication not completed. 1: Second authentication completed.
0	AUTS0	This bit indicates the status of the first authentication of the link partner. 0: First authentication not completed. 1: First authentication completed.

(4) RHSIFnMIST — Module Interrupt Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DBGS	—	—	AESS	—	—	SRES	STES	CERS3	CERS2	CERS1	CERS0	—	SRCS1	SRCS0	STCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TERS3	AKRS3	RARS3	—	TERS2	AKRS2	RARS2	—	TERS1	AKRS1	RARS1	—	TERS0	AKRS0	RARS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.21 RHSIFnMIST Register Contents (1/2)

Bit Position	Bit Name	Function
31	DBGS	When set to 1, this bit indicates that there is a debug interrupt factor.
30 to 29	Reserved	When read, the value after reset is returned.
28	AESS	When set to 1, this bit indicates that there is an error interrupt factor concerning authentication or window mis-hit.
27, 26	Reserved	When read, the value after reset is returned.
25	SRES	When set to 1, this bit indicates that there is an error interrupt factor concerning stream reception.
24	STES	When set to 1, this bit indicates that there is an error interrupt factor concerning stream transmission.
23	CERS3	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 3.
22	CERS2	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 2.
21	CERS1	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 1.
20	CERS0	When set to 1, this bit indicates that there is an error interrupt factor concerning channel 0.
19	Reserved	When read, the value after reset is returned.
18	SRCS1	When set to 1, this bit indicates that there is a factor for completing stream interrupt reception (of not less than the specified size).
17	SRCS0	When set to 1, this bit indicates that there is a factor for completing stream interrupt reception (of not less than the data size of one frame).
16	STCS	When set to 1, this bit indicates that there is a factor for completing stream interrupt transmission.
15	Reserved	When read, the value after reset is returned.
14	TERS3	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 3 interrupt.
13	AKRS3	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 3 interrupt.
12	RARS3	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 3 interrupt.
11	Reserved	When read, the value after reset is returned.

Table 29.21 RHSIFnMIST Register Contents (2/2)

Bit Position	Bit Name	Function
10	TERS2	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 2 interrupt.
9	AKRS2	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 2 interrupt.
8	RARS2	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 2 interrupt.
7	Reserved	When read, the value after reset is returned.
6	TERS1	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 1 interrupt.
5	AKRS1	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 1 interrupt.
4	RARS1	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 1 interrupt.
3	Reserved	When read, the value after reset is returned.
2	TERS0	When set to 1, this bit indicates that there is a factor for receiving an Event Command as a channel 0 interrupt.
1	AKRS0	When set to 1, this bit indicates that there is a factor for receiving an ACK Command as a channel 0 interrupt.
0	RARS0	When set to 1, this bit indicates that there is a factor for receiving a Read Answer Command as a channel 0 interrupt.

(5) RHSIFnMRT — Module Reply Timeout Time Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 020_H

Value after reset: 0000 $F3FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSCL[3:0]				—	—	RCNT[9:0]									
Value after reset	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.22 RHSIFnMRT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 12	RSCL[3:0]	These bits specify the reference clock for the reply timer. Note that these bits can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation. 0000: Communication clock (cclk) 0001: Communication clock (cclk) divided by 2 0010: Communication clock (cclk) divided by 4 0011: Communication clock (cclk) divided by 8 0100: Communication clock (cclk) divided by 16 0101: Communication clock (cclk) divided by 32 0110: Communication clock (cclk) divided by 64 0111: Communication clock (cclk) divided by 128 1000: Communication clock (cclk) divided by 256 1001: Communication clock (cclk) divided by 512 1010: Communication clock (cclk) divided by 1024 1011: Communication clock (cclk) divided by 2048 1100: Communication clock (cclk) divided by 4096 1101: Communication clock (cclk) divided by 8192 1110: Communication clock (cclk) divided by 16384 1111: Communication clock (cclk) divided by 32768
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	RCNT[9:0]	These bits specify the limit count for the reply timer. If the pulse count of the clock specified by the RSCL bits reaches the value specified in these bits, a reply timeout is determined. When 0 is specified in these bits, the reply timeout does not occur. If, however, no reply Command is detected with 0 specified in these bits, the initiator on the channel is disabled. Therefore, specifying 0 in these bits is prohibited if the system does not have a measure to recover from the status in which the initiator is disabled. Note that these bits can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.

(6) RHSIFnMW (A, B, C, D) A — Module Memory Window (A, B, C, D) Start Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnMWAA:<RHSIFn_L2_base> + 040_H
RHSIFnMWAB:<RHSIFn_L2_base> + 050_H
RHSIFnMWAC:<RHSIFn_L2_base> + 060_H
RHSIFnMWAD:<RHSIFn_L2_base> + 070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MW(X)A[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MW(X)A[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	R	R

Note 1. This bit is readable/writable (RW) or read-only (R). For details, see the description of the bit.

Table 29.23 RHSIFnMW (A, B, C, D) A Register Contents

Bit Position	Bit Name	Function
31 to 2	MW(X)A	These bits specify the start address of the window that is allowed to be accessed when RHSIF generates a H-Bus read or write request as bus master. Write 0 to all the bits lower than those corresponding to the valid size value set in RHSIFnMW(X)S. Example: When the valid size is 4 kilobytes, set the MW(X)A[11:2] bits to $0000\_0000\_00_B$. The MW(X)A[6:2] bits must always be 0. When RHSIFnMCT.W(X)RE = 1 or RHSIFnMCT.W(X)WE = 1, the value of these bits (MW(X)A) must not be changed. For details, see **Section 29.4.3.1(1), Setting Memory Windows**. Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have the read-only (R) attribute. These bits can be read from all master devices.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(7) RHSIFnMW (A, B, C, D) S — Module Memory Window (A, B, C, D) Size Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnMWAS: <RHSIFn_L2_base> + 048_H
RHSIFnMWBS:<RHSIFn_L2_base> + 058_H
RHSIFnMWCS:<RHSIFn_L2_base> + 068_H
RHSIFnMWDS:<RHSIFn_L2_base> + 078_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MW(X)S[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MW(X)S[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	R	R

Note 1. This bit is readable / writable (R/W) or read-only (R). For details, see the description of the bit.

Table 29.24 RHSIFnMW (A, B, C, D) S Register Contents

Bit Position	Bit Name	Function
31 to 2	MW(X)S	These bits specify the size of the window that is allowed to be accessed when RHSIF generates a H-Bus read or write request as bus master. Write 1 to all the bits lower than those corresponding to the valid size value, and write 0 to all the bits higher than those corresponding to the valid size value. Examples of settings are as follows: 0000_0000_0000_0000_0000_0000_0111_11b: 128 bytes 0000_0000_0000_0000_0000_0000_1111_11b: 256 bytes 0000_0000_0000_0000_0000_0001_1111_11b: 512 bytes … 0011_1111_1111_1111_1111_1111_1111_11b: 1 GB 0111_1111_1111_1111_1111_1111_1111_11b: 2 GB 1111_1111_1111_1111_1111_1111_1111_11b: 4 GB The MW(X)S[6:2] bits must always be 1. When RHSIFnMCT.W(X)RE = 1 or RHSIFnMCT.W(X)WE = 1, the value of these bits (MW(X)S) must not be changed. For details, see **Section 29.4.3.1(1), Setting Memory Windows**. Data cannot be written to these bits from RHSIF's master port. When RHSIF's master port writes to these bits, these bits have read-only (R) attribute. These bits can be read from all master devices.
1 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(8) RHSIFnSTMD — Stream Tx Mode Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 200_H

Value after reset: $0000\ 1110_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	STNK	—	—	—	STPS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.25 RHSIFnSTMD Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	STNK	This bit specifies the operation that is necessary when L2, operating as an initiator node, receives a NACK Command from a link partner in response to a Stream Command the initiator node transmitted. 0: Stops the DMAC. (Same as the operation that is necessary when an error occurs) 1: Continues the DMAC operation. (Same as the operation to be done when an ACK Command is received) Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	STPS	This bit specifies the data payload size of the frame when L2, operating as an initiator node, transmits a Stream Command. **Note:** Transmission / reception in 128 bits is only available for communication between the devices with the same STPS values. 0: 256 bits 1: 128 bits Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(9) RHSIFnSTCT — Stream Tx Control Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 208_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.26 RHSIFnSTCT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STDE	This bit specifies whether to enable the DMAC. Write 1 to this bit to start the DMAC when L2, operating as an initiator node, transmits a Stream Command. 0: Disables the DMAC. 1: Enables the DMAC. This bit is automatically cleared when the transfer ends (normally or abnormally). For details about abnormal cases, see **Section 29.4.3.4, Stream Command Transmission by Initiator Node**. During the transfer (while this bit is 1), 0 must not be written to this bit.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the STDE bit to 1.

(10) RHSIFnSTST — Stream Tx Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 210_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STE4	—	—	STE3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STE2	STE1	—	STE0	—	STC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.27 RHSIFnSTST Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	STE4	This bit is set to 1 when a memory window mis-hit occurs and the DMAC stops abnormally (because of the error) after RHSIF, operating as an initiator, has started the DMAC for Stream Command transmission.
22 to 21	Reserved	When read, the value after reset is returned.
20	STE3	This bit is set to 1 when a transfer error occurs in the H-Bus and the DMAC stops abnormally (because of the error) after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
19 to 6	Reserved	When read, the value after reset is returned.
5	STE2	This bit is set to 1 when a reply Command (ACK, NACK, or Read Answer) causing a transaction ID error is detected and the DMAC stops abnormally (because of the error) after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
4	STE1	This bit is set to 1 when the DMAC stops abnormally because it does not receive any response within the time set in the RHSIFnMRT bits after L2, operating as an initiator, has started the DMAC for Stream Command transmission.
3	Reserved	When read, the value after reset is returned.
2	STE0	This bit is set to 1 when an NACK Command is received and the DMAC stops abnormally after L2, operating as an initiator, has started the DMAC for Stream Command transmission. This bit is not set when RHSIFnSTMD.STNK = 1.
1	Reserved	When read, the value after reset is returned.
0	STC	This bit is set to 1 when the transfer of all data ends normally after L2, operating as an initiator, has started the DMAC for Stream Command transmission.

(11) RHSIFnSTSC — Stream Tx Status Clear Register

Access: This register is a write-only register that can be write in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 218_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEC4	—	—	STEC3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	R	W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STEC2	STEC1	—	STEC0	—	STCC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	R	W	R	W

Table 29.28 RHSIFnSTSC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When writing, write the value after reset.
23	STEC4	Writing 1 to this bit clears the RHSIFnSTST.STE4 bit. Writing 0 to this bit is ignored.
22 to 21	Reserved	When writing, write the value after reset.
20	STEC3	Writing 1 to this bit clears the RHSIFnSTST.STE3 bit. Writing 0 to this bit is ignored.
19 to 6	Reserved	When writing, write the value after reset.
5	STEC2	Writing 1 to this bit clears the RHSIFnSTST.STE2 bit. Writing 0 to this bit is ignored.
4	STEC1	Writing 1 to this bit clears the RHSIFnSTST.STE1 bit. Writing 0 to this bit is ignored.
3	Reserved	When writing, write the value after reset.
2	STEC0	Writing 1 to this bit clears the RHSIFnSTST.STE0 bit. Writing 0 to this bit is ignored.
1	Reserved	When writing, write the value after reset.
0	STCC	Writing 1 to this bit clears the RHSIFnSTST.STC bit. Writing 0 to this bit is ignored.

(12) RHSIFnSTIE — Stream Tx Interrupt Enable Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 220_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STEE4	—	—	STEE3	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	STEE2	STEE1	—	STEE0	—	STCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R	R/W

Table 29.29 RHSIFnSTIE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	STEE4	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE4 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE4 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE4 bit is 1.
22 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	STEE3	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE3 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE3 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE3 bit is 1.
19 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	STEE2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE2 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE2 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE2 bit is 1.
4	STEE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE1 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE1 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE1 bit is 1.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	STEE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STE0 bit. 0: Does not assert int_hsif_err when the RHSIFnSTST.STE0 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSTST.STE0 bit is 1.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STCE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSTST.STC bit. 0: Does not assert int_hsif_str when the RHSIFnSTST.STC bit is 1. 1: Asserts int_hsif_str when the RHSIFnSTST.STC bit is 1.

(13) RHSIFnSTSA — Stream Tx Source Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 228_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STSA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STSA[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 29.30 RHSIFnSTSA Register Contents

Bit Position	Bit Name	Function
31 to 4	STSA	These bits specify the start address of the area storing the data to be transferred when L2, operating as an initiator node, starts the DMAC for Stream Command transmission. When the data payload of the Stream Command to be transmitted is 256 bits (when RHSIFnSTMD.STPS = 0), the STSA[4] bit must always be 0. When RHSIFnSTCT.STDE = 1, the value of these bits (STSA) must not be changed. The settings for RHSIFnSTSA and RHSIFnSTBC which causes wrap around of address is not allowed.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(14) RHSIFnSTBC — Stream Tx Byte Count Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 230_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	STBC[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STBC[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 29.31 RHSIFnSTBC Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	STBC	These bits specify the total number of bytes of the data to be transferred when L2, operating as an initiator node, starts the DMAC for Stream Command transmission. When the data payload of the Stream Command to be transmitted is 256 bits (when RHSIFnSTMD.STPS = 0), the STBC[4] bit must always be 0. When 0 is set in these bits, 2^{25} bytes (32 MBytes) are transferred. The value read from these bits indicates the number of remaining bytes of the data. When RHSIFnSTCT.STDE = 1, the value of these bits (STBC) must not be changed. The setting of RHSIFnSTSA and RHSIFnSTBC that causes wrap around of address is not allowed.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

Write to unused Channel Register should not be executed.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the STBC bits to 1.

(15) RHSIFnSRMD — Stream Rx Mode Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 280_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SRMC	—	—	—	SRPS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.32 RHSIFnSRMD Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	SRMC	This bit specifies the operation to be done if an error occurs when L2, operating as a target node, receives a Stream Command. 0: Automatically clears the RHSIFnSRCT.SRDE bit automatically when an error occurs. 1: Does not automatically clear the RHSIFnSRCT.SRDE bit automatically when an error occurs. For the conditions of errors, see the description of the RHSIFnSRCT.SRDE bit. Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	SRPS	This bit specifies the frame data payload size that is expected when L2, operating as a target node, receives a Stream Command. 0: 256 bits 1: 128 bits Note that this bit can be changed only at the time of initial setting. You cannot change the setting dynamically during system operation.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(16) RHSIFnSRCT — Stream Rx Control Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 288_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRDE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.33 RHSIFnSRCT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRDE	This bit specifies whether to enable the DMAC. Write 1 to this bit to start the DMAC when L2, operating as a target node, receives a Stream Command. 0: Disables the DMAC (ignores the Stream Command that is received without any response). 1: Enables the DMAC (automatically transfers the Stream Command that is received to a storage area). When RHSIFnCMD2.TNME2 = 0, you must not set this bit to 1. When RHSIFnSRMD.SRMC = 0, this bit is cleared to 0 automatically when one of the following events occur: • An error occurs when Stream Command data is transferred to the H-Bus. • When a Stream Command is received, Stream Command data is discarded because the data storage area is full. • An error is detected in a stream Command that is received. • In channel 2, a Command causing a CRC error is detected. • H-Bus Write request's address does not hit any memory window.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the SRDE bit to 1.

(17) RHSIFnSRST — Stream Rx Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 290_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRA	—	—	—	—	—	—	—	SRE2	—	SRE1	SRE0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRTA	—	—	—	—	—	—	—	—	—	—	—	—	—	SRC1	SRC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.34 RHSIFnSRST Register Contents

Bit Position	Bit Name	Function
31	SRA	This bit is set to 1 if RHSIFnSRCT.SRDE is cleared automatically because L2, operating as a target node, detects errors related to the Stream Command that is received. For the conditions of errors, see the description of the RHSIFnSRCT.SRDE bit.
30 to 24	Reserved	When read, the value after reset is returned.
23	SRE2	This bit is set to 1 if a memory window mis-hit occurs when RHSIF, operating as a target node, transfers Stream Command data that is received to the H-Bus.
22	Reserved	When read, the value after reset is returned.
21	SRE1	This bit is set to 1 if Stream Command data is discarded because the data storage area is full when L2, operating as a target node, receives a Stream Command.
20	SRE0	This bit is set to 1 if an error occurs when L2, operating as a target node, transfers Stream Command data that is received to the H-Bus.
19 to 16	Reserved	When read, the value after reset is returned.
15	SRTA	This bit indicates the processing status of the received Stream Command when L2 is operating as a target node. 0: No Command is being processed. 1: Command(s) is being processed.
14 to 2	Reserved	When read, the value after reset is returned.
1	SRC1	This bit is set to 1 when the size of unprocessed data (in the received Stream Command when L2 is operating as a target node) is not less than the value set in RHSIFnSRBC.
0	SRC0	This bit is set to 1 when the size of unprocessed data (in the received Stream Command when L2 is operating as a target node) is not 0.

(18) RHSIFnSRSC — Stream Rx Status Clear Register

Access: This register is a write-only register that can be write in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 298_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRAC	—	—	—	—	—	—	—	SREC2	—	SREC1	SREC0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	W	R	W	W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.35 RHSIFnSRSC Register Contents

Bit Position	Bit Name	Function
31	SRAC	Writing 1 to this bit clears the RHSIFnSRST.SRA bit. Writing 0 to this bit is ignored.
30 to 24	Reserved	When writing, write the value after reset.
23	SREC2	Writing 1 to this bit clears the RHSIFnSRST.SRE2 bit. Writing 0 to this bit is ignored.
22	Reserved	When writing, write the value after reset.
21	SREC1	Writing 1 to this bit clears the RHSIFnSRST.SRE1 bit. Writing 0 to this bit is ignored.
20	SREC0	Writing 1 to this bit clears the RHSIFnSRST.SRE0 bit. Writing 0 to this bit is ignored.
19 to 0	Reserved	When writing, write the value after reset.

(19) RHSIFnSRIE — Stream Rx Interrupt Enable Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $2A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRAE	—	—	—	—	—	—	—	SREE2	—	SREE1	SREE0	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRCE1	SRCE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 29.36 RHSIFnSRIE Register Contents

Bit Position	Bit Name	Function
31	SRAE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRA bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRA bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRA bit is 1.
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	SREE2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE2 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE2 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE2 bit is 1.
22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	SREE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE1 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE1 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE1 bit is 1.
20	SREE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRE0 bit. 0: Does not assert int_hsif_err when the RHSIFnSRST.SRE0 bit is 1. 1: Asserts int_hsif_err when the RHSIFnSRST.SRE0 bit is 1.
19 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SRCE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRC1 bit. 0: Does not assert int_hsif_str when the RHSIFnSRST.SRC1 bit is 1. 1: Asserts int_hsif_str when the RHSIFnSRST.SRC1 bit is 1.
0	SRCE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnSRST.SRC0 bit. 0: Does not assert int_hsif_str when the RHSIFnSRST.SRC0 bit is 1. 1: Asserts int_hsif_str when the RHSIFnSRST.SRC0 bit is 1.

(20) RHSIFnSRDA — Stream Rx Destination Area Start Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $2A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SRDA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRDA[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 29.37 RHSIFnSRDA Register Contents

Bit Position	Bit Name	Function
31 to 4	SRDA	These bits specify the start address of the area to store data when L2, operating as a target node, receives a Stream Command. Write 0 to all the bits lower than those corresponding to the valid size value set in RHSIFnSRDS. Example: When the valid size is 4 kilobytes, set the SRDA[11:4] bits to $0000\_0000\_00_B$. When the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the SRDA[4] bit must always be 0. When RHSIFnSRCT.SRDE = 1, the value of these bits (SRDA) must not be changed.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(21) RHSIFnSRDS — Stream Rx Destination Area Size Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $2B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SRDS[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRDS[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 29.38 RHSIFnSRDS Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRDS	These bits specify the valid size of the area to store data when L2, operating as a target node, receives a Stream Command. Write 1 to all the bits lower than those corresponding to the valid size value, and write 0 to all the bits higher than those corresponding to the valid size value. Examples of settings are as follows: $0\_0000\_0000\_0000\_0000\_0000_B$: 16 bytes $0\_0000\_0000\_0000\_0000\_0001_B$: 32 bytes ... $0\_1111\_1111\_1111\_1111\_1111_B$: 16 Mbytes $1\_1111\_1111\_1111\_1111\_1111_B$: 32 Mbytes When the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the valid size must not be set to 16 bytes. When RHSIFnSRCT.SRDE = 1, the value of these bits (SRDS) must not be changed.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(22) RHSIFnSRBC — Stream Rx Byte Count Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L2_base> + $2B8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SRBC[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRBC[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 29.39 RHSIFnSRBC Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRBC	These bits specify the number of bytes that triggers a notification when L2, operating as a target node, receives a Stream Command. When the size of unprocessed data reaches or exceeds the value specified in these bits, the RHSIFnSRST.SRC1 bit is set to 1. If the data payload of the Stream Command to be received is 256 bits (when RHSIFnSRMD.SRPS = 0), the SRBC[4] bit must always be 0. If 0 is set in these bits, a notification is triggered when the size of unprocessed data reaches 2^{25} bytes (32 Mbytes). When RHSIFnSRCT.SRDE = 1, the value of these bits (SRBC) must not be changed.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(23) RHSIFnSRWP — Stream Rx Write Pointer Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L2_base> + $2C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SRWP[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRWP[15:4]												—	—	—	SRWT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.40 RHSIFnSRWP Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	SRWP	These bits indicate the value of the write pointer (offset value from start address) to the area to store received data when L2, operating as a target node, receives a Stream Command. The number of valid bits of the pointer corresponds to the value set in RHSIFnSRDS. All the bits higher than valid bits show 0. These bits are initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1. For details, see **Section 29.4.3.5, Stream Command Reception by Target Node**.
3 to 1	Reserved	When read, the value after reset is returned.
0	SRWT	This bit is a toggle bit for the SRWP bits. This bit is toggled each time the SRWP bits wrap around. This bit is initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1.

(24) RHSIFnSRRP — Stream Rx Read Pointer Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $2C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SRRP[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRRP[15:4]												—	—	—	SRRT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 29.41 RHSIFnSRRP Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 4	SRRP	These bits indicate the value of the read pointer (offset value from start address) to the area to store received data when L2, operating as a target node, receives a Stream Command. The number of valid bits of the pointer corresponds to the value set in RHSIFnSRDS. All the bits higher than valid bits must be 0. These bits are initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1. For details, see **Section 29.4.3.5, Stream Command Reception by Target Node**.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRRT	This bit is a toggle bit for the SRRP bits. This bit is toggled each time the SRRP bits wrap around. This bit is initialized when the RHSIFnSRCT.SRDE bit changes from 0 to 1.

NOTE

Note that when using a bit-manipulation instruction for writing to this register, the read-modify-write operation may inadvertently set the SRRP bits or the SRRT bit to 1.

(25) RHSIFnAID (0, 1, 2, 3, 4, 5, 6, 7) — Authentication ID Register (0, 1, 2, 3, 4, 5, 6, 7)

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnAID0: <RHSIFn_L2_base> + 300_H
RHSIFnAID1: <RHSIFn_L2_base> + 308_H
RHSIFnAID2: <RHSIFn_L2_base> + 310_H
RHSIFnAID3: <RHSIFn_L2_base> + 318_H
RHSIFnAID4: <RHSIFn_L2_base> + 320_H
RHSIFnAID5: <RHSIFn_L2_base> + 328_H
RHSIFnAID6: <RHSIFn_L2_base> + 330_H
RHSIFnAID7: <RHSIFn_L2_base> + 338_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AID(M)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AID(M)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 29.42 RHSIFnAID (0, 1, 2, 3, 4, 5, 6, 7) Register Contents

Bit Position	Bit Name	Function
31 to 0	AID(M)	These bits store the ID data for the first authentication (ID-based authentication). HSIFnAID0.AID0[31:0]: ID[31:0] HSIFnAID1.AID1[31:0]: ID[63:32] HSIFnAID2.AID2[31:0]: ID[95:64] HSIFnAID3.AID3[31:0]: ID[127:96] HSIFnAID4.AID4[31:0]: ID[159:128] HSIFnAID5.AID5[31:0]: ID[191:160] HSIFnAID6.AID6[31:0]: ID[223:192] HSIFnAID7.AID7[31:0]: ID[255:224] Data can be read from or written to these bits only via the master port of L2. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.

(26) RHSIFnAAD (0, 1, 2, 3) — Authentication Answer Data Register (0, 1, 2, 3)

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnAAD0: <RHSIFn_L2_base> + 340_H
RHSIFnAAD1: <RHSIFn_L2_base> + 348_H
RHSIFnAAD2: <RHSIFn_L2_base> + 350_H
RHSIFnAAD3: <RHSIFn_L2_base> + 358_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AAD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AAD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable/writable (R/W) or read-only (R). For details, see the description of the bit.

Table 29.43 RHSIFnAAD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	AAD(N)	These bits store the answer data (calculated from challenge data by the ICUMHA) for the second authentication (challenge-and-response authentication). RHSIFnAAD0.AAD0[31:0]: Answer Data[31:0] RHSIFnAAD1.AAD1[31:0]: Answer Data[63:32] RHSIFnAAD2.AAD2[31:0]: Answer Data[95:64] RHSIFnAAD3.AAD3[31:0]: Answer Data[127:96] Data can be read from or written to these bits only from ICUMHA. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.

(27) RHSIFnARD (0, 1, 2, 3) — Authentication Response Data Register (0, 1, 2, 3)

Access: This register can be read or written in 32-bit units.

Address: RHSIFnARD0: <RHSIFn_L2_base> + 360_H
RHSIFnARD1: <RHSIFn_L2_base> + 368_H
RHSIFnARD2: <RHSIFn_L2_base> + 370_H
RHSIFnARD3: <RHSIFn_L2_base> + 378_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Note 1. This bit is readable / writable (R/W) or read-only (R). For details, see the description of the bit.

Table 29.44 RHSIFnARD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	ARD(N)	These bits store the response data for the second authentication (challenge-and-response authentication). RHSIFnARD0.ARD0[31:0]: Response Data[31:0] RHSIFnARD1.ARD1[31:0]: Response Data[63:32] RHSIFnARD2.ARD2[31:0]: Response Data[95:64] RHSIFnARD3.ARD3[31:0]: Response Data[127:96] Data can be read from or written to these bits only via the master port of L2. When another master device such as a CPU accesses these bits, these bits have the read-only (R) attribute.

(28) RHSIFnAEST — Authentication Error Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $3E0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WES1	WES0	WEW3	WER3	WEW2	WER2	WEW1	WER1	WEW0	WER0	—	—	—	AES
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEW3	AER3	AEI3	AEE3	AEW2	AER2	AEI2	AEE2	AEW1	AER1	AEI1	AEE1	AEW0	AER0	AEI0	AEE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.45 RHSIFnAEST Register Contents

Bit Position	Bit Name	Function
31 to 30	Reserved	When read, the value after reset is returned.
29	WES1	This bit is set to 1 when RHSIF, as a target node, tries to issue H-Bus Write for Stream Command that does not hit any memory window.
28	WES0	This bit is set to 1 when RHSIF, as an initiator node, tries to issue H-Bus Read for Stream Command that does not hit any memory window.
27,25,23,21	WEW(N)	This bit is set to 1 when RHSIF, as a target node, receives Write Command on Channel (N) whose targeted address does not hit any memory window.
26,24,22,20	WER(N)	This bit is set to 1 when RHSIF, as a target node, receives Read Command on Channel (N) whose targeted address does not hit any memory window.
19 to 17	Reserved	When read, the value after reset is returned.
16	AES	This bit is set to 1 when RHSIF, as a target node, receives Stream Command on Channel (N) before authentication is completed.
15,11,7,3	AEW(N)	This bit is set to 1 when RHSIF, as a target node, receives Write Command on Channel (N) before authentication is completed.
14,10,6,2	AER(N)	This bit is set to 1 when RHSIF, as a target node, receives Read Command on Channel (N) before authentication is completed.
13,9,5,1,	AEI(N)	This bit is set to 1 when RHSIF, as a target node, receives ID Command on Channel (N) before authentication is completed.
12,8,4,0	AEE(N)	This bit is set to 1 when RHSIF, as a target node, receives Event Command on Channel (N) before authentication is completed.

(29) RHSIFnAESC — Authentication Error Status Clear Register

Access: This register is a write-only register that can be write in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $3E8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WESC1	WESC0	WEWC3	WERC3	WEWC2	WERC2	WEWC1	WERC1	WEWC0	WERC0	—	—	—	AESC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	W	W	W	W	W	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEWC3	AERC3	AEIC3	AEEC3	AEWC2	AERC2	AEIC2	AEEC2	AEWC1	AERC1	AEIC1	AEEC1	AEWC0	AERC0	AEIC0	AEEC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 29.46 RHSIFnAESC Register Contents

Bit Position	Bit Name	Function
31 to 30	Reserved	When writing, write the value after reset.
29	WESC1	Writing 1 to this bit clears the RHSIFnAEST.WES1 bit. Writing 0 to this bit is ignored.
28	WESC0	Writing 1 to this bit clears the RHSIFnAEST.WES0 bit. Writing 0 to this bit is ignored.
27,25,23,21	WEWC(N)	Writing 1 to this bit clears the RHSIFnAEST.WEW(N) bit. Writing 0 to this bit is ignored.
26,24,22,20	WERC(N)	Writing 1 to this bit clears the RHSIFnAEST.WER(N) bit. Writing 0 to this bit is ignored.
19 to 17	Reserved	When writing, write the value after reset.
16	AESC	Writing 1 to this bit clears the RHSIFnAEST.AES bit. Writing 0 to this bit is ignored.
15,11,7,3	AEWC(N)	Writing 1 to this bit clears the RHSIFnAEST.AEW(N) bit. Writing 0 to this bit is ignored.
14,10,6,2	AERC(N)	Writing 1 to this bit clears the RHSIFnAEST.AER(N) bit. Writing 0 to this bit is ignored.
13,9,5,1,	AEIC(N)	Writing 1 to this bit clears the RHSIFnAEST.AEI(N) bit. Writing 0 to this bit is ignored.
12,8,4,0	AEEC(N)	Writing 1 to this bit clears the RHSIFnAEST.AEE(N) bit. Writing 0 to this bit is ignored.

(30) RHSIFnAEIE — Authentication Error Interrupt Enable Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + $3F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	WESE1	WESE0	WEWE3	WERE3	WEWE2	WERE2	WEWE1	WERE1	WEWE0	WERE0	—	—	—	AESE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEWE3	AERE3	AEIE3	AEEE3	AEWE2	AERE2	AEIE2	AEEE2	AEWE1	AERE1	AEIE1	AEEE1	AEWE0	AERE0	AEIE0	AEEE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.47 RHSIFnAEIE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	WESE1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WES1 bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.WES1 bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.WES1 bit is 1.
28	WESE0	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WES0 bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.WES0 bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.WES0 bit is 1.
27,25,23,21	WEWE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WEW(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.WEW(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.WEW(N) bit is 1.
26,24,22,20	WERE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.WER(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.WER(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.WER(N) bit is 1.
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	AESE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AES bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.AES bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.AES bit is 1.
15,11,7,3	AEWE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEW(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.AEW(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.AEW(N) bit is 1.
14,10,6,2	AERE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AER(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.AER(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.AER(N) bit is 1.
13,9,5,1,	AEIE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEI(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.AEI(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.AEI(N) bit is 1.

Table 29.47 RHSIFnAEIE Register Contents (2/2)

Bit Position	Bit Name	Function
12,8,4,0	AEEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnAEST.AEE(N) bit. 0: Does not assert int_hsif_sec when the RHSIFnAEST.AEE(N) bit is 1. 1: Asserts int_hsif_sec when the RHSIFnAEST.AEE(N) bit is 1.

(31) RHSIFnCMD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Mode Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: <RHSIFn_L2_base> + 400_H
<RHSIFn_L2_base> + 480_H
<RHSIFn_L2_base> + 500_H
<RHSIFn_L2_base> + 580_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TNME (N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INME (N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.48 RHSIFnCMD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TNME(N)	This bit specifies whether to enable the target node function. 0: Disables reception of Read, Write, Event, ID, and Stream Commands (ignores these Commands without any response). 1: Enables reception of Read, Write, Event, ID, and Stream Commands. Even when this bit is 1, whether Read, Write, and Stream Commands can be received depends on other individual settings. For details, see **Section 29.4.3.3, Non-Stream Command Reception by Target Node** and **Section 29.4.3.5, Stream Command Reception by Target Node**.
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	INME(N)	This bit specifies whether to enable the initiator node function. 0: Disables transmission of Read, Write, Event, ID, and Stream Commands. 1: Enables transmission of Read, Write, Event, ID, and Stream Commands. This bit must not be cleared to 0 while L2 is transmitting data or waiting for a response by using the initiator node function.

(32) RHSIFnCCT (0, 1, 2, 3) — Channel (0, 1, 2, 3) Control Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: RHSIFnCCT0: <RHSIFn_L2_base> + 408_H
RHSIFnCCT1: <RHSIFn_L2_base> + 488_H
RHSIFnCCT2: <RHSIFn_L2_base> + 508_H
RHSIFnCCT3: <RHSIFn_L2_base> + 588_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CTYW (N)	—	—	CTY(N)[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	R	W	W	W	W	W

Table 29.49 RHSIFnCCT (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7	CTYW(N)	"1" must be written to this bit at the same time as writing data to the CTY(N) bits when L2, operating as an initiator node, transmits a request Command other than a Stream Command. When this bit is 0, the data written to the CTY(N) bits is ignored.
6, 5	Reserved	When writing, write the value after reset.
4 to 0	CTY(N)	A Command type must be set in these bits when L2, operating as an initiator node, transmits a request Command other than a Stream Command. When L2 transmits a Command, 1 must be written to the CTYW(N) bit at the same time as writing data to these bits. 00000: Read Command (8 bits) 00001: Read Command (16 bits) 00010: Read Command (32 bits) 00100: Write Command (8 bits) 00101: Write Command (16 bits) 00110: Write Command (32 bits) 01100: Event Command 10010: ID Command Only the above Command types can be set.

(33) RHSIFnCST (0, 1, 2, 3) — Channel (0, 1, 2, 3) Status Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: RHSIFnCST0: <RHSIFn_L2_base> + 410_H
RHSIFnCST1: <RHSIFn_L2_base> + 490_H
RHSIFnCST2: <RHSIFn_L2_base> + 510_H
RHSIFnCST3: <RHSIFn_L2_base> + 590_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRE(N)	—	—	—	—	—	—	—	—	—	—	BRE(N)	—	—	—	TER(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDY(N)	—	—	—	—	—	—	—	AOE(N)	—	IDE(N)	TOE(N)	—	AKE(N)	AKR(N)	RAR(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.50 RHSIFnCST (0, 1, 2, 3) Register Contents (1/2)

Bit Position	Bit Name	Function
31	CRE(N)	This bit is set to 1 when a Command causing a CRC error is detected in the channel corresponding to the bit number. For details, see **Section 29.4.3.6, CRC Error Handling**.
30 to 21	Reserved	When read, the value after reset is returned.
20	BRE(N)	This bit is set to 1 when an error has occurred during data transfer in the H-Bus by a Read or Write Command received by L2 operating as a target node.
19 to 17	Reserved	When read, the value after reset is returned.
16	TER(N)	This bit is set to 1 when L2, operating as a target mode, receives an Event Command. Note that L2 ignores an event if another Event Command is received when this bit is already set to 1.
15	RDY(N)	This bit indicates whether L2, operating as an initiator node, can transmit request Commands other than Stream Commands. 0: Request Commands cannot be transmitted (the initiator node function is disabled or L2 is waiting for a reply Command). 1: Request Commands can be transmitted. Transmission of request Commands (writing 1 to the RHSIFnCCT(N).CTYW(N) bit) must be done only when this bit is 1.
14 to 8	Reserved	When read, the value after reset is returned.
7	AOE(N)	This bit is set to 1 when an error that cannot be classified as the one indicated by the AKE or IDE bit is detected after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command. In particular, the following event corresponds to this error: • A Read Command transmitted by RHSIF has met the "Any Other" condition in *the Table 18 of the HSSL / SIPI Interprocessor Bus Protocol Specification*.
6	Reserved	When read, the value after reset is returned.
5	IDE(N)	This bit is set to 1 when a reply Command (ACK, NACK, or Read Answer) causing a transaction ID error is detected after L2, operating as an initiator node, has transmitted a Read, Write, Event or ID Command.
4	TOE(N)	This bit is set to 1 when L2 cannot receive a response within the time set in RHSIFnMRT after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command.
3	Reserved	When read, the value after reset is returned.

Table 29.50 RHSIFnCST (0, 1, 2, 3) Register Contents (2/2)

Bit Position	Bit Name	Function
2	AKE(N)	This bit is set to 1 when L2 receives an NACK Command after L2, operating as an initiator node, has transmitted a Read, Write, Event, or ID Command.
1	AKR(N)	This bit is set to 1 when L2 receives an ACK Command after L2, operating as an initiator node, has transmitted a Write or Event Command.
0	RAR(N)	This bit is set to 1 when L2 receives a Read Answer Command after L2, operating as an initiator node, has transmitted a Read or ID Command.

(34) RHSIFnCSC (0, 1, 2, 3) — Channel (0, 1, 2, 3) Status Clear Register

Access: This register is a write-only register that can be write in 8-, 16-, or 32-bit units.

Address: RHSIFnCSC0: <RHSIFn_L2_base> + 418_H
RHSIFnCSC1: <RHSIFn_L2_base> + 498_H
RHSIFnCSC2: <RHSIFn_L2_base> + 518_H
RHSIFnCSC3: <RHSIFn_L2_base> + 598_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CREC(N)	—	—	—	—	—	—	—	—	—	—	BREC(N)	—	—	—	TERC(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AOEC(N)	—	IDEC(N)	TOEC(N)	—	AKEC(N)	AKRC(N)	RARC(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	R	W	W	R	W	W	W

Table 29.51 RHSIFnCSC (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31	CREC(N)	Writing 1 to this bit clears the RHSIFnCST(N).CRE(N) bit. Writing 0 to this bit is ignored.
30 to 21	Reserved	When writing, write the value after reset.
20	BREC(N)	Writing 1 to this bit clears the RHSIFnCST(N).BRE(N) bit. Writing 0 to this bit is ignored.
19 to 17	Reserved	When writing, write the value after reset.
16	TERC(N)	Writing 1 to this bit clears the RHSIFnCST(N).TER(N) bit. Writing 0 to this bit is ignored.
15 to 8	Reserved	When writing, write the value after reset.
7	AOEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AOE(N) bit. Writing 0 to this bit is ignored.
6	Reserved	When writing, write the value after reset.
5	IDEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).IDE(N) bit. Writing 0 to this bit is ignored.
4	TOEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).TOE(N) bit. Writing 0 to this bit is ignored.
3	Reserved	When writing, write the value after reset.
2	AKEC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AKE(N) bit. Writing 0 to this bit is ignored.
1	AKRC(N)	Writing 1 to this bit clears the RHSIFnCST(N).AKR(N) bit. Writing 0 to this bit is ignored.
0	RARC(N)	Writing 1 to this bit clears the RHSIFnCST(N).RAR(N) bit. Writing 0 to this bit is ignored.

(35) RHSIFnCIE (0, 1, 2, 3) — Channel (0, 1, 2, 3) Interrupt Enable Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnCIE0: <RHSIFn_L2_base> + 420_H
RHSIFnCIE1: <RHSIFn_L2_base> + $4A0_H$
RHSIFnCIE2: <RHSIFn_L2_base> + 520_H
RHSIFnCIE3: <RHSIFn_L2_base> + $5A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CREE(N)	—	—	—	—	—	—	—	—	—	—	BREE(N)	—	—	—	TERE(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AOEE(N)	—	IDEE(N)	TOEE(N)	—	AKEE(N)	AKRE(N)	RARE(N)
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W

Table 29.52 RHSIFnCIE (0, 1, 2, 3) Register Contents (1/2)

Bit Position	Bit Name	Function
31	CREE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).CRE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).CRE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).CRE(N) bit is 1.
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	BREE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).BRE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).BRE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).BRE(N) bit is 1.
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TERE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).TER(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).TER(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).TER(N) bit is 1.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	AOEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AOE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).AOE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).AOE(N) bit is 1.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	IDEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).IDE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).IDE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).IDE(N) bit is 1.
4	TOEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).TOE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).TOE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).TOE(N) bit is 1.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 29.52 RHSIFnCIE (0, 1, 2, 3) Register Contents (2/2)

Bit Position	Bit Name	Function
2	AKEE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AKE(N) bit. 0: Does not assert int_hsif_err when the RHSIFnCST(N).AKE(N) bit is 1. 1: Asserts int_hsif_err when the RHSIFnCST(N).AKE(N) bit is 1.
1	AKRE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).AKR(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).AKR(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).AKR(N) bit is 1.
0	RARE(N)	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnCST(N).RAR(N) bit. 0: Does not assert int_hsif_ch(N) when the RHSIFnCST(N).RAR(N) bit is 1. 1: Asserts int_hsif_ch(N) when the RHSIFnCST(N).RAR(N) bit is 1.

(36) RHSIFnCAR (0, 1, 2, 3) — Channel (0, 1, 2, 3) Write/Read Address Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnCAR0: <RHSIFn_L2_base> + 428_H
RHSIFnCAR1: <RHSIFn_L2_base> + $4A8_H$
RHSIFnCAR2: <RHSIFn_L2_base> + 528_H
RHSIFnCAR3: <RHSIFn_L2_base> + $5A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAR(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CAR(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.53 RHSIFnCAR (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CAR(N)	These bits specify the address of the area that is the target of a Read or Write Command when L2, operating as an initiator node, transmits the Read or Write Command. When the request size is 16 bits, the CAR(N)[0] bit must always be 0. When the request size is 32 bits, the CAR(N)[1:0] bits must always be 00.

(37) RHSIFnCWD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Write Data Register

Access: This register can be read or written in 8-, 16-, or 32-bit units.

Address: RHSIFnCWD0: <RHSIFn_L2_base> + 430_H
RHSIFnCWD1: <RHSIFn_L2_base> + $4B0_H$
RHSIFnCWD2: <RHSIFn_L2_base> + 530_H
RHSIFnCWD3: <RHSIFn_L2_base> + $5B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CWD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CWD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.54 RHSIFnCWD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CWD(N)	These bits are used to set the data to be transmitted when L2, operating as an initiator node, transmits a Write Command. When the request size is 8 bits, set the data in the CDW(N)[7:0] bits. When the request size is 16 bits, set the data in the CDW(N)[15:0] bits. When the request size is 32 bits, set the data in the CDW(N)[31:0] bits.

(38) RHSIFnCRD (0, 1, 2, 3) — Channel (0, 1, 2, 3) Read Data Register

Access: This register is a read-only register that can be read in 8-, 16-, or 32-bit units.

Address: RHSIFnCRD0:<RHSIFn_L2_base> + 438_H
RHSIFnCRD1:<RHSIFn_L2_base> + $4B8_H$
RHSIFnCRD2:<RHSIFn_L2_base> + 538_H
RHSIFnCRD3:<RHSIFn_L2_base> + $5B8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRD(N)[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRD(N)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.55 RHSIFnCRD (0, 1, 2, 3) Register Contents

Bit Position	Bit Name	Function
31 to 0	CRD(N)	These bits contain the data that is received from a target node after L2, operating as an initiator node, has transmitted a Read or ID Command to the target node. When the request size is 8 bits, reference the CRD(N)[7:0] bits. When the request size is 16 bits, reference the CRD(N)[15:0] bits. When the request size is 32 bits, reference the CRD(N)[31:0] bits. These bits are valid only when RHSIFnCST(N).RAR(N) = 1.

29.4.3 Operation

29.4.3.1 Initial Settings

The procedure for the initial settings of L2 is as follows:

(1) Set the memory windows

(2) Set the channels

(3) Set the link partner authentication (only for systems that use the target node function and require authentication) *1

(4) Set the transmission stream (only for systems that transmit Stream Commands by using the initiator node function)

(5) Set the reception stream (only for systems that receive Stream Commands by using the target node function)

Note 1. To complete link partner authentication, you need to at least specify the settings below before step.(3).
At step.(1): Open memory window for the IP's register space.
for first authentication: RHSIFnAID0-7
for second authentication: RHSIFnARD0-3
At step.(2): Enable target node function on at least one channel.

The following sections describe how to specify settings with individual procedural steps:

(1) Setting Memory Windows

Figure 29.6 shows the flow to set memory windows in L2.

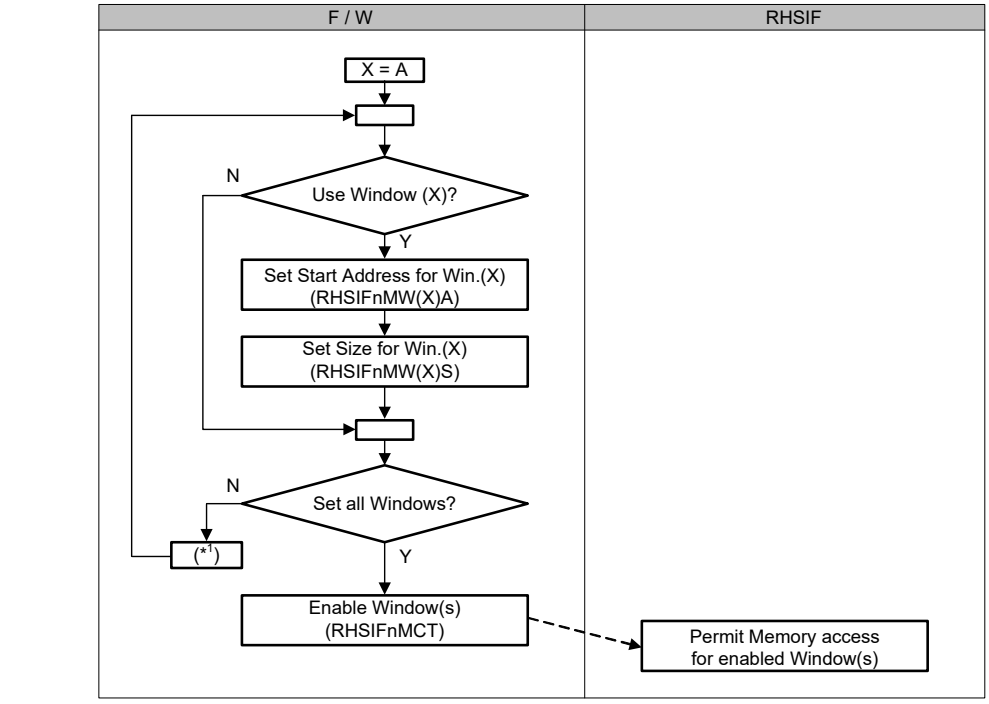
When the target node function of L2 is enabled, link partners can access (read and write) information in the product with L2 installed. If, however, unspecified link partners are permitted to access L2 without restriction, security problems can occur. Therefore, L2 has a function to enable you to specify the space (called a “window”) that is permitted to be accessed.

You can set up to four windows independently of channels. For each window, specify a start address and a size to prevent window areas from overlapping. You can also set read and write permissions separately for individual windows.

This window setting is valid for H-Bus requests which are issued by RHSIF as master. For details, refer to **Section 29.4.3.3, Non-Stream Command Reception by Target Node**, **Section 29.4.3.4, Stream Command Transmission by Initiator Node** and **Section 29.4.3.5, Stream Command Reception by Target Node**.

If you set windows at a time other than during the initial settings after a reset, you must disable the initiator node function on all channels (clear the RHSIFnCMD(N).INME(N) bit to 0), and the target node function on all channels (clear the RHSIFnCMD(N).TNME(N) bit to 0) before setting windows.

In normal mode, RHSIFnMCT, RHSIFnMW(X)A, and RHSIFnMW(X)S cannot be written to RHSIF’s master port. Reading data from these registers is possible from all master modules.



Note 1. Current Value = “A” → Change to “B”, ..., Current Value = “C” → Change to “D”

Figure 29.6 Setting Memory Windows

Figure 29.7 shows an example of setting memory windows.

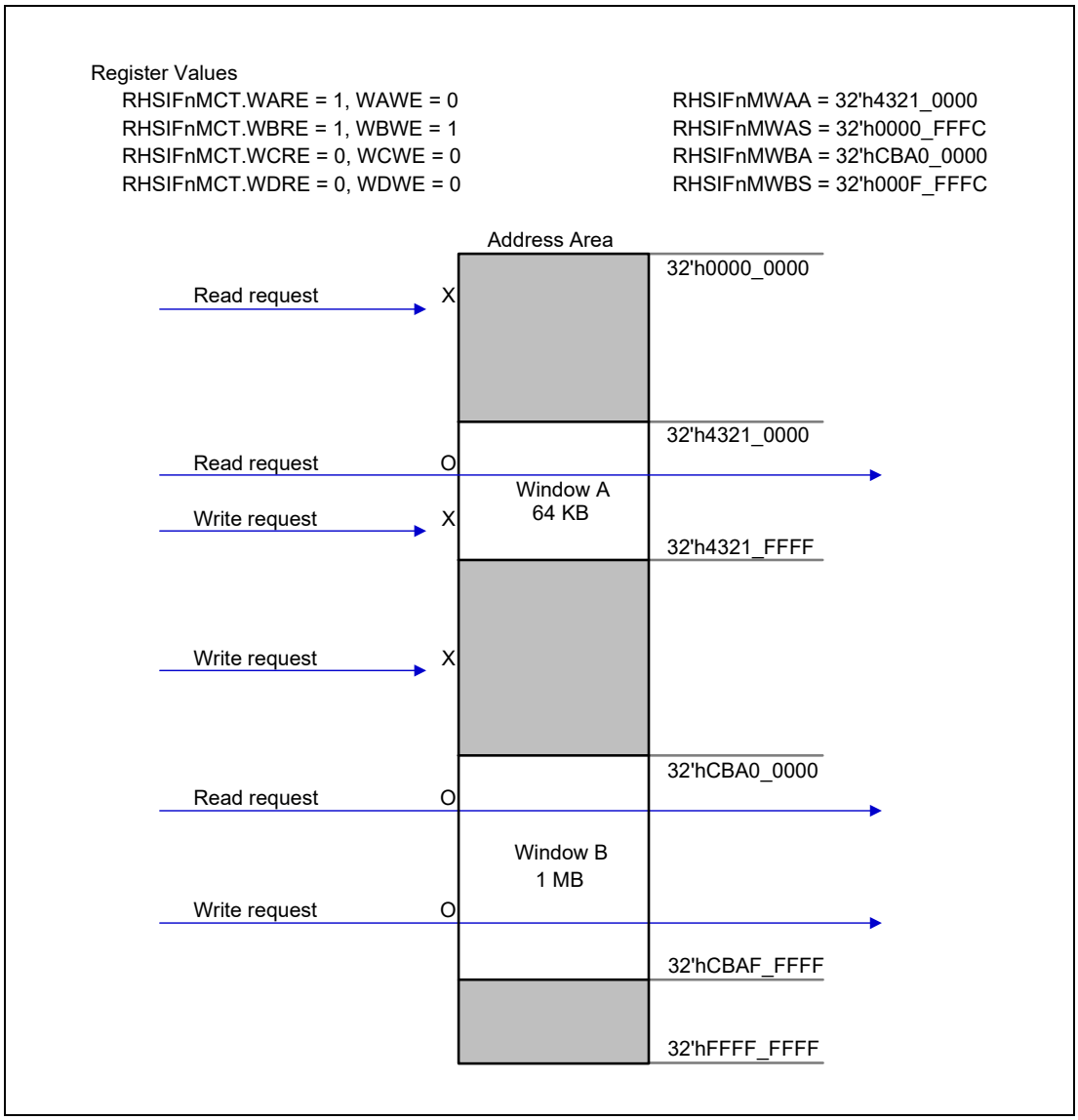


Figure 29.7 Example of Setting Memory Windows

(2) Setting Channels

Figure 29.8 shows the flow of setting channels for L2. Enable the functions to be used for individual channels.

When the RHSIFnCMD(N).INME(N) bit is set to 1 for a channel, L2 is enabled to transmit non-Stream Commands (Read, Write, Event, and ID Commands) through the channel. For details, see **Section 29.4.3.2, Non-Stream Command Transmission by Initiator Node**.

Also, specify a reply timeout time according to the system. The reply timer starts when L2 (Layer 2) finishes to transmit last symbol of a request Command. The reply timer stops when one of the following events occurs;

- L2 detects an expected reply Command.
- For Read Command, L2 detects reply Command with error (except CRC error).
- Reply timeout occurs

The reply timeout time that is specified applies to all channels.

When the RHSIFnCMD(N).TNME(N) bit is set to 1 for a channel, L2 is enabled to receive non-Stream Commands (Read, Write, Event, and ID Commands) through the channel. For details, see **Section 29.4.3.3, Non-Stream Command Reception by Target Node**.

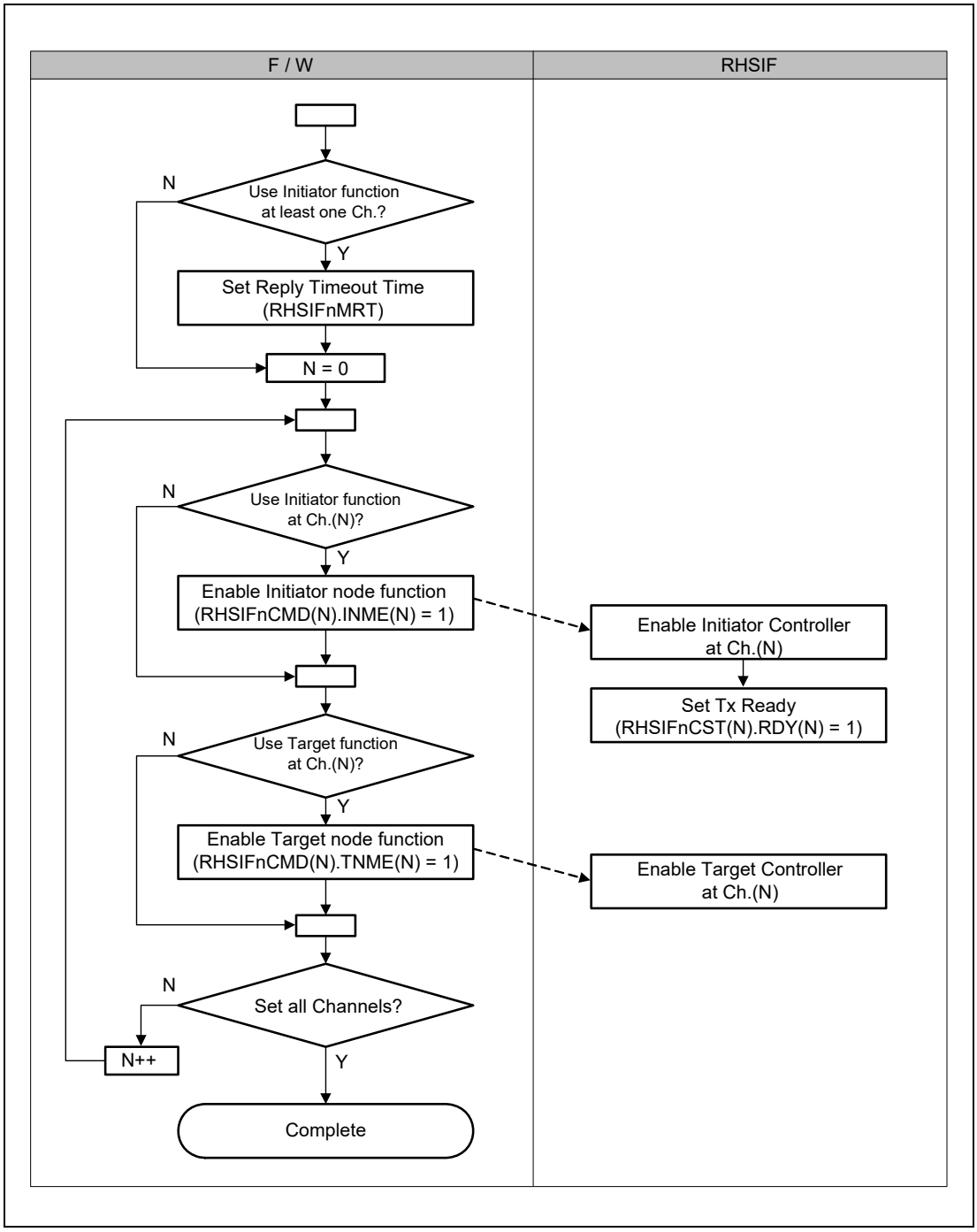


Figure 29.8 Setting Channels

(3) Setting Link Partner Authentication

When the target node function of L2 is enabled, link partners can access (read and write) information in the product with L2 installed. If, however, unspecified link partners are permitted to access L2, security problems can occur. Therefore, L2 has a function to authenticate link partners.

Link partner authentication is performed by using the following two steps:

1. Authentication based on a 256-bit key ID*1
2. Challenge-and-response authentication*2

You can enable or disable each type of authentication by using FLASH option byte. When link partner authentication is disabled, link partners can access the internal space of the product with L2 installed without performing the authentication procedure described in this section. Specify settings of the relevant FLASH option byte for RHSIF according to the security level required of the product with L2 installed.

Note 1. First Authentication (ID-based Authentication) needs setting.
Product shipping setting is invalid.
For the values of the FLASH option byte, refer to **Section 47, Basic Hardware Protection (BHP)**.

Note 2. Second Authentication (Challenge and Response Authentication) needs setting.
Product shipping setting is invalid.
For the values of the FLASH option byte for security, Refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

(a) First Authentication (ID-Based)

Figure 29.9 shows a concept of the first authentication (ID-based authentication), and the procedure for the authentication is described below.

(1) The link partner issues a Write Command (32 bits) for the address corresponding to the RHSIFnAID0 register in L2. Write data is the [31:0] bits of key ID (256 bits).

(2) Upon receiving the Write Command, the target controller of L2 checks the address in the Command.

A) When the address matches any of RHSIFnAID0 to RHSIFnAID7, the target controller determines the Write Command to be valid, and then issues a master write request to the H-Bus.

B) When the address does not match any of RHSIFnAID0 to RHSIFnAID7, the target controller ignores the Write Command without returning any reply Command.

(3) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.

A) When the master ID matches the ID of L2, L2 determines the request to be valid, and then writes data to RHSIFnAID0.

B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnAID0, and then returns an OKAY response.

(4) In the manner similar to steps (1) to (3), data is set in the RHSIFnAID1 to RHSIFnAID7 registers.

(5) When data is set in RHSIFnAID7, L2 compares the data (32 bits x 8 = 256 bits) in RHSIFnAID0 to RHSIFnAID7 with the data of FLASH option byte.

Refer to **Section 47, Basic Hardware Protection (BHP)** and **Table 47.24, RHSIFIDn Contents** for details.

A) If the data matches each other, L2 determines that the link partner passes the authentication. When only the first authentication is required, L2 determines whether the Read or Write Command from the link partner is valid according to the window setting described in **Section 29.4.3.1(1), Setting Memory Windows**

When the second authentication is also required, the procedure described in **Section 29.4.3.1(2), Setting Channels** is performed.

B) If the data does not match, L2 determines that the link partner fails the authentication. In such a case, the authentication procedure must be repeated beginning from step (1). If the link partner fails the authentication three consecutive times, L2 determines that the link partner is unauthorized, and will not let the link partner pass the authentication until a reset signal is asserted.

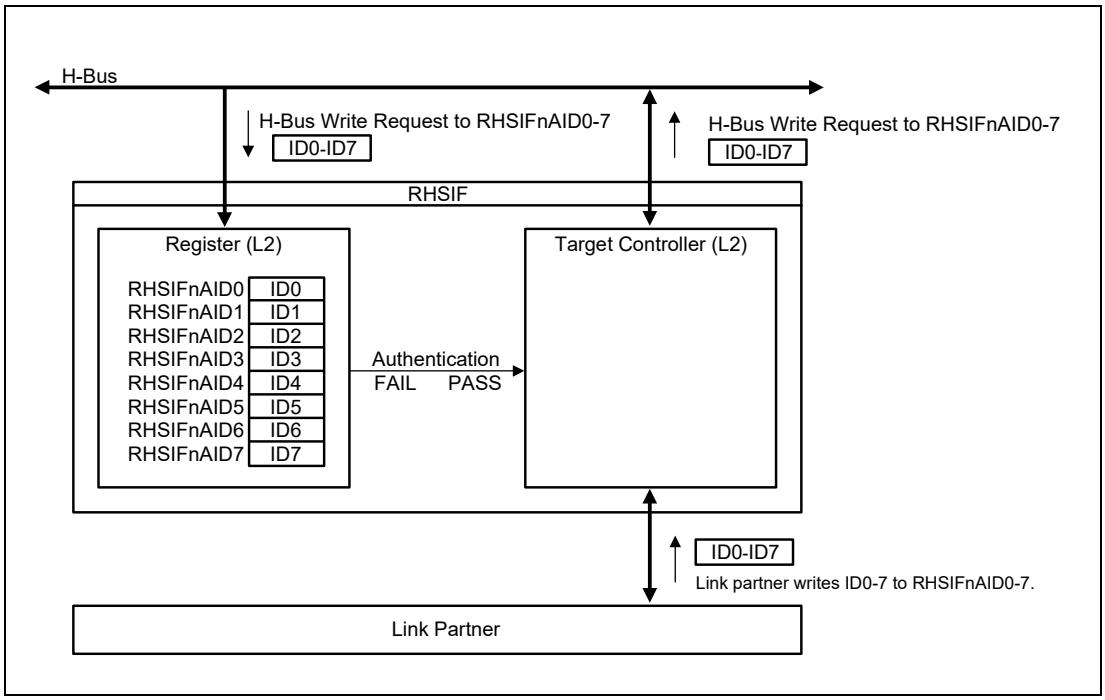


Figure 29.9 Concept of ID-Based Authentication

(b) Second Authentification (Challenge and Response)

Figure 29.10 and **Figure 29.11** show a concept of the second authentication (challenge-and-response authentication), and the procedure for the authentication is described below. Note that this manual describes only how to set the data to be used for challenge and response. For the procedure of challenge and response, see the manual for the product in which L2 is installed.

Step 1 Preparation of the challenge data (Figure 29.10)

(1) Prepare the challenge data, and send it to the Link Partner. The way to generate the challenge data is selectable.
In the example in this figure, challenge data is generated by ICUMHA and send it via RHSIF to the Link Partner.

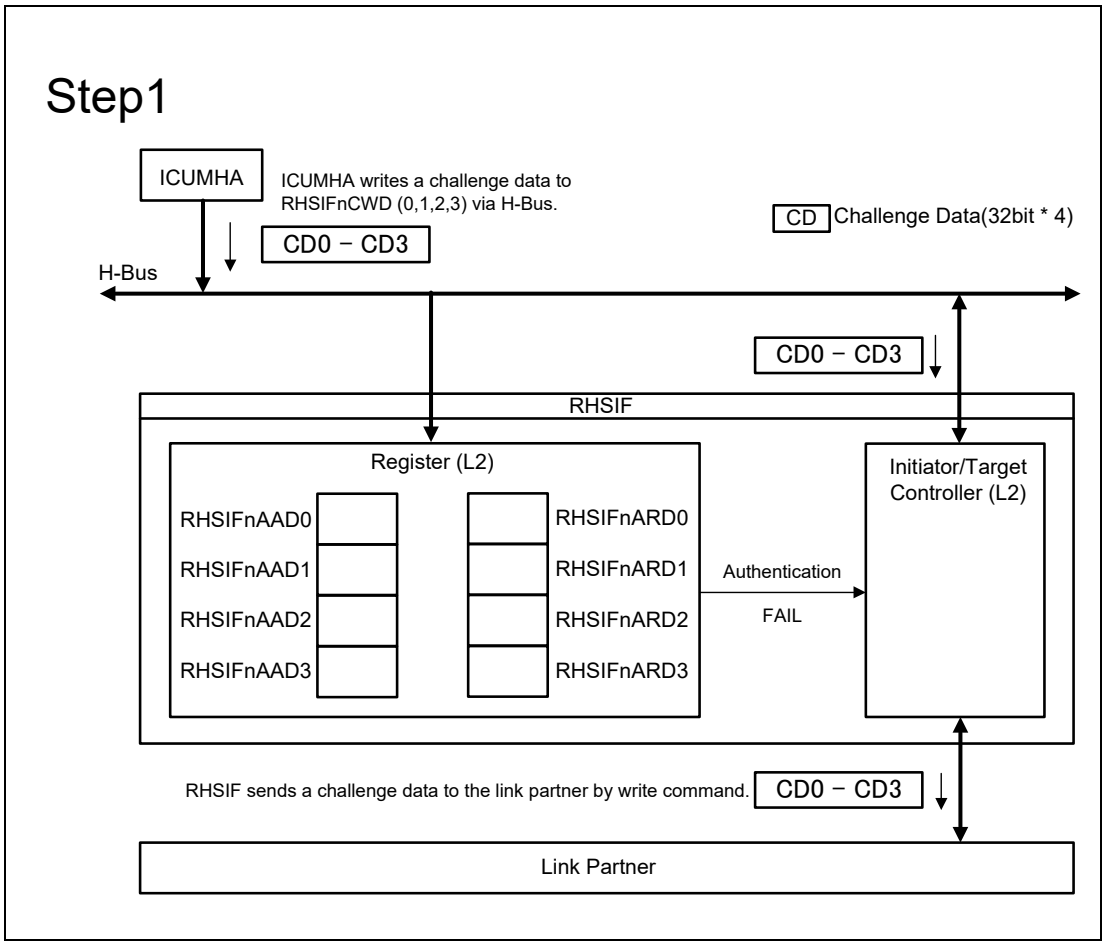


Figure 29.10 Concept of Challenge and Response Authentication (Preparation of the challenge data)

Step 2 Calculate the response data and compare it (Figure 29.11)

(1) The security module (ICUMHA) connected to the H-Bus calculates Answer (Response) data from Challenge data, and then sets the [31:0] bits of the Response data in the RHSIFnAAD0 register in L2.

(2) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.

A) When the master ID matches the one in the ICUMHA, L2 determines the request to be valid, and then writes data to RHSIFnAAD0.

B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnAAD0, and then returns an OKAY response.

(3) In the manner similar to steps (1) and (2), data is set in the RHSIFnAAD1 to RHSIFnAAD3 registers.

(4) The link partner issues a Write Command (32 bits) for the address corresponding to the RHSIFnARD0 register in L2. Write data is the [31:0] bits of Response data.

(5) Upon receiving the Write Command, the target controller of L2 checks the address in the Command.

A) When the address matches any of RHSIFnARD0 to RHSIFnARD3, the target controller determines the Write Command to be valid, and then issues a master write request to the H-Bus.

B) When the address does not match any of RHSIFnARD0 to RHSIFnARD3, the target controller ignores the Write Command without returning any reply Command.

(6) Upon receiving the H-Bus write request, the register unit of L2 checks the master ID in the request.

A) When the master ID matches the ID of L2, L2 determines the request to be valid, and then writes data to RHSIFnARD0.

B) When the master ID does not match, L2 does not execute the Write Command to RHSIFnARD0, and then returns an OKAY response.

(7) In the manner similar to steps (4) to (6), data is set in the RHSIFnARD1 to RHSIFnARD3 registers.

(8) When data is set in RHSIFnARD3, L2 compares the data (32 bits x 4 = 128 bits) in RHSIFnARD0 to RHSIFnARD3 with the data (32 bits x 4 = 128 bits) preset in RHSIFnAAD0 to RHSIFnAAD3.

A) If the data matches each other, L2 determines that the link partner passes the authentication. After that, L2 determines whether the read or Write Command from the link partner is valid according to the window setting described in **Section 29.4.3.1(1), Setting Memory Windows**.

B) If the data does not match, L2 determines that the link partner fails the authentication. In such a case, the authentication procedure must be repeated beginning from step (1). If the link partner fails the authentication three consecutive times, L2 determines that the link partner is unauthorized, and will not let the link partner pass the authentication until a reset signal is asserted.

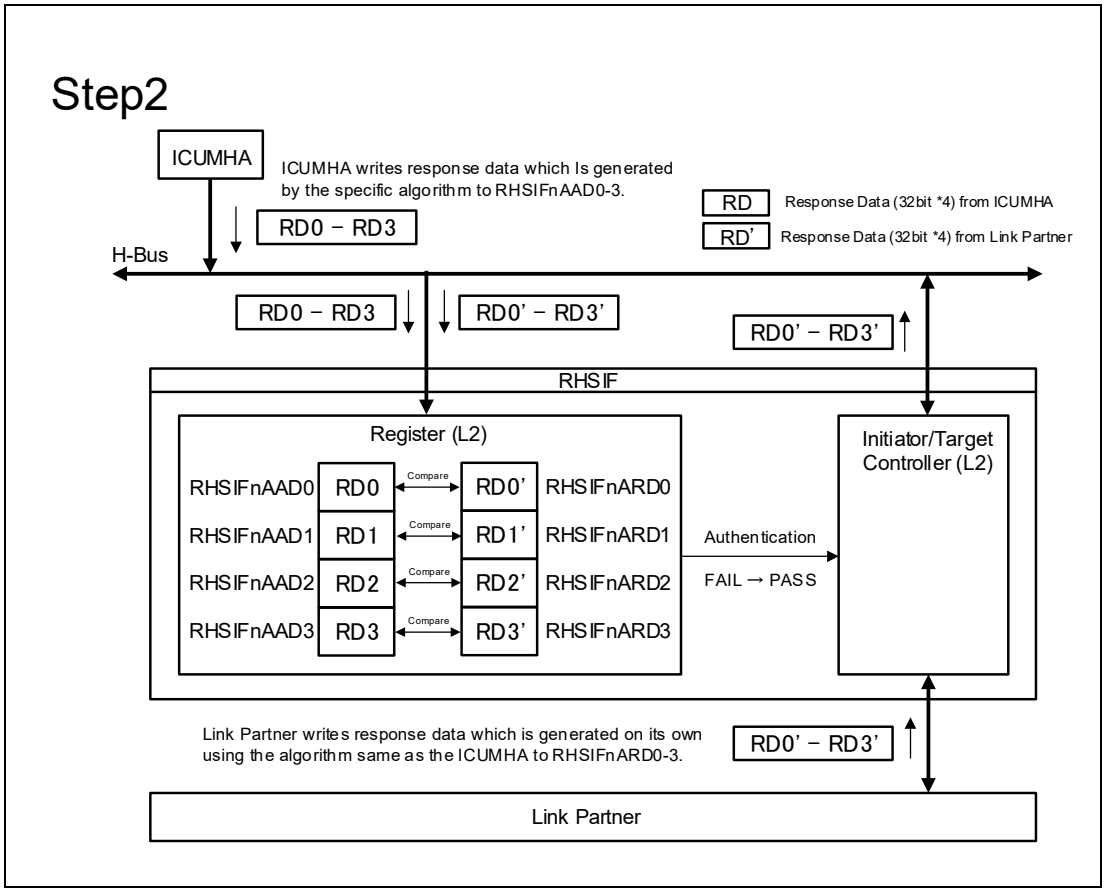


Figure 29.11 Concept of Challenge and Response Authentication (Calculates and Compares Response Data)

(4) Setting Tx Stream for Initiator Node

Figure 29.12 shows the flow of the initial settings of Stream Command transmission by L2.

RHSIF's protocol defines the data payload size of a Stream Command as 256 bits. You, however, can also set the data payload size to 128 bits for L2. Note that the data payload size can be changed only at the time of the initial settings. You cannot change this setting dynamically during system operation. Specify an appropriate value in the relevant parameter during the initial settings according to the system.

There are two modes for receiving NACK Command.

- RHSIFnSTMD.STNK = 0: L2 regards reception of the NACK Command as occurrence of an error.
- RHSIFnSTMD.STNK = 1: L2 regards reception of the NACK Command as reception of an ACK Command.

Select an appropriate mode according to the application program.

For details about the flow of Stream Command transmission, see **Section 29.4.3.4, Stream Command Transmission by Initiator Node**.

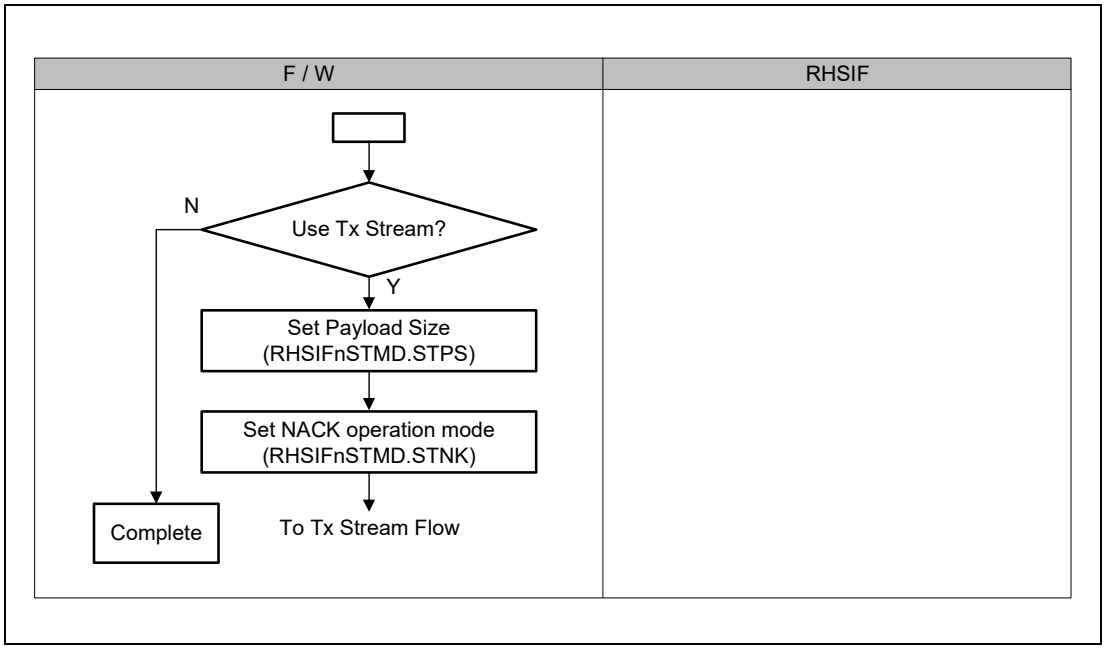


Figure 29.12 Initial Setting of Tx Stream

(5) Setting Rx Stream for Target Node

Figure 29.13 shows the flow of the initial settings of Stream Command reception by L2.

RHSIF defines the data payload size of a Stream Command as 256 bits. You, however, can also set the data payload size to 128 bits for L2. Note that the data payload size can be changed only at the time of the initial settings. You cannot change this setting dynamically during system operation. Specify an appropriate value in the relevant parameter during the initial settings according to the system.

How to handle an error occurring in a command that is received or during data transfer depends on the application program. L2 allows you to select the operation that is necessary when an error occurs in the following two modes:

- RHSIFnSRMD.SRMC = 0: The RHSIFnSRCT.SRDE bit is cleared automatically when an error occurs.
  Select this mode when you want to stop data storage immediately after the error occurs.
  Note that, because L2 can receive two Stream Commands concurrently, if an H-Bus error occurs, L2 might store the data up to the data causing the error.
- RHSIFnSRMD.SRMC = 1: The RHSIFnSRCT.SRDE bit is not cleared automatically when an error occurs.
  Select this mode when you want to let the firmware determine the processing that is necessary if an error occurs.
  To stop data storage after the processing is determined, clear the RHSIFnSRCT.SRDE bit by the firmware.

Select an appropriate mode according to the application program.

For details about the flow of Stream Command reception, see **Section 29.4.3.5, Stream Command Reception by Target Node**.

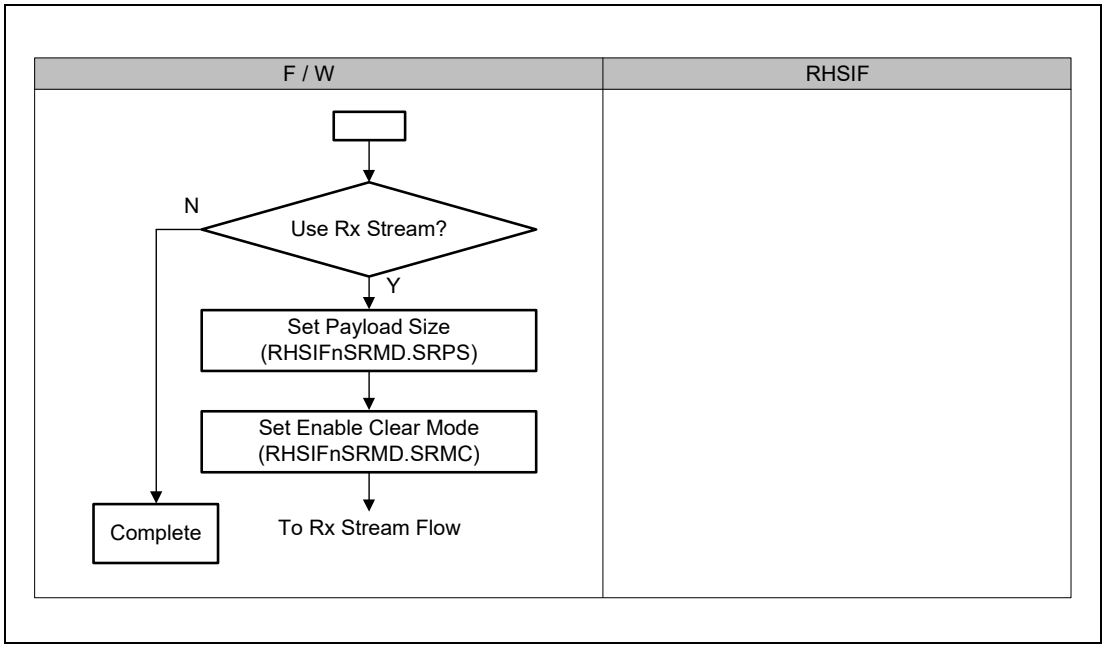


Figure 29.13 Initial Setting of Rx Stream

29.4.3.2 Non-Stream Command Transmission by Initiator Node

Figure 29.14 to **Figure 29.17** show the flows for transmitting non-stream requests (Read, Write, Event, and ID Commands) by using the initiator node function of L2.

Transmission of non-stream requests is performed under control of the firmware. A request is transmitted by accessing a register, and an interrupt is used as the trigger for checking response.

The following describes the transmission operation, using Read Command transmission (**Figure 29.14**) for example.

When a Command transmission request is found, whether transmission is enabled must be checked first. Transmission is enabled when the RHSIFnCST(N).RDY(N) bit is 1. Then, a target address must be set in the RHSIFnCAR(N) register, and a trigger for transmission must be issued by using the RHSIFnCCT(N) register. When the trigger is issued, L2 generates and transmits the Command.

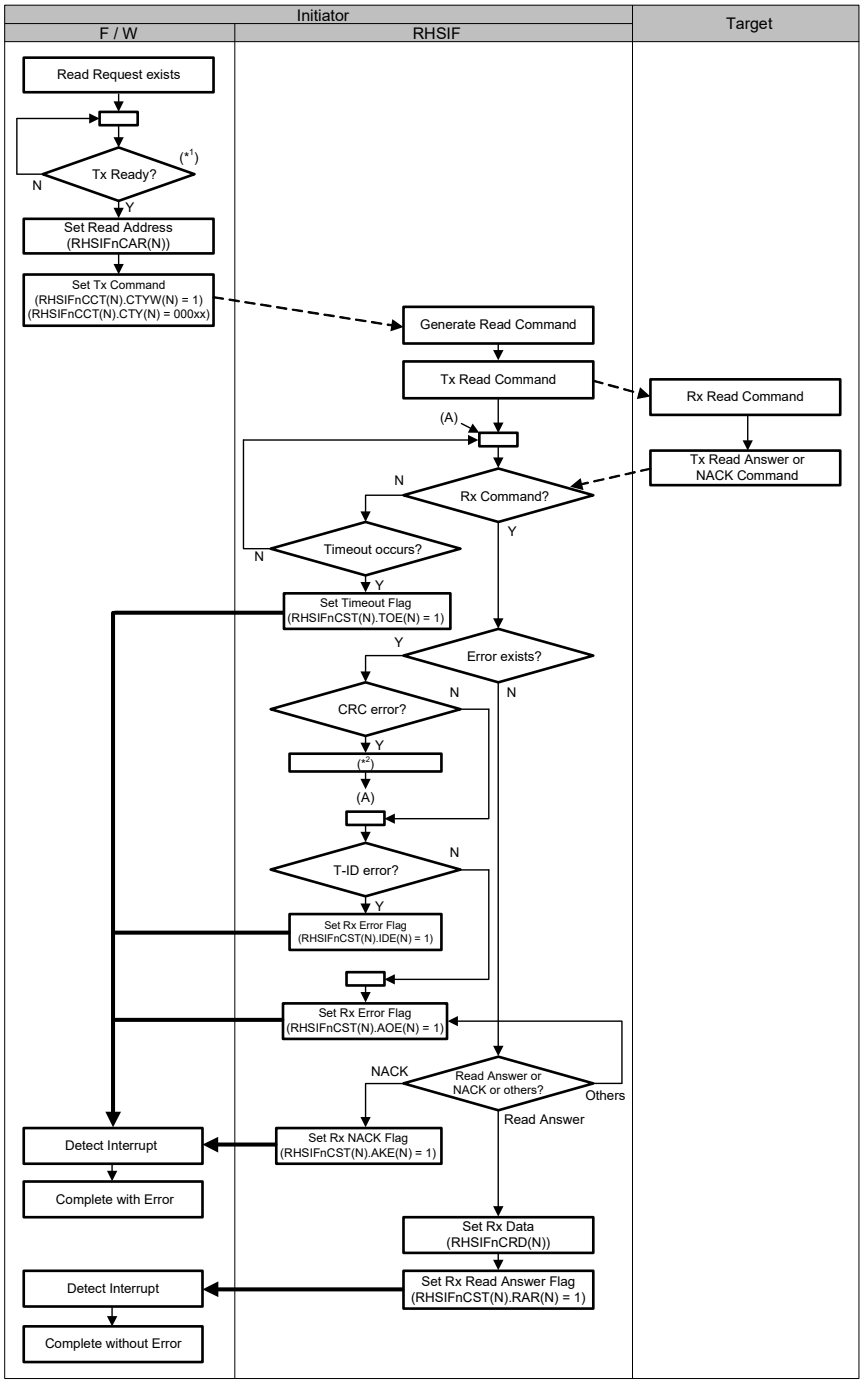
After transmitting the Command, L2 waits for a reply Command from the link partner. L2 subsequently performs one of the following four operations depending on the response of the link partner and whether a frame error occurs:

(1) When the link partner returns a Read Answer Command (normal case)
L2 sets the RHSIFnCST(N).RAR(N) bit, and (when interrupt is enabled) generates an int_hsif_ch(N) interrupt.
Checking of read data in the RHSIFnCRD(N) register and other necessary processing must be performed.

(2) When the link partner returns an NACK Command
L2 sets the RHSIFnCST(N).AKE(N) bit, and (when interrupt is enabled) generates an int_hsif_err interrupt.
Retransmission of the Command and other necessary processing must be performed.

(3) When the reply Command from the link partner involves an error
L2 sets the RHSIFnCST(N).IDE(N) or RHSIFnCST(N).AOE(N) bit according to the type of the error, and (when interrupt is enabled) generates an int_hsif_err interrupt.
Retransmission of the Command and other necessary processing must be performed.
In case reply Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

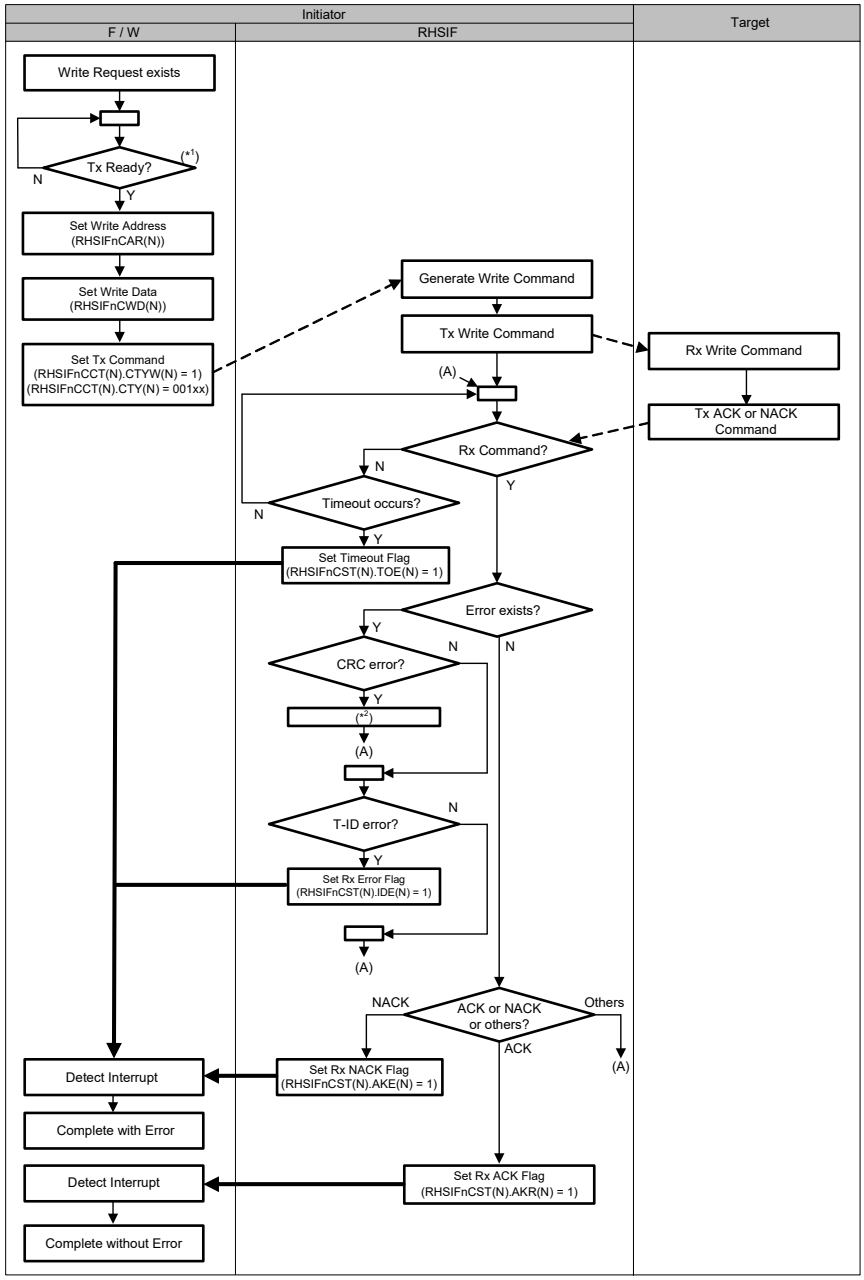
(4) When the link partner does not respond (reply timeout occurs)
If the reply Command from the link partner cannot be detected within the time set in the RHSIFnMRT register after Command transmission, L2 sets the RHSIFnCST(N).TOE(N) bit, and (when interrupt is enabled) generates an int_hsif_err interrupt.
Retransmission of the Command and other necessary processing must be performed.



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

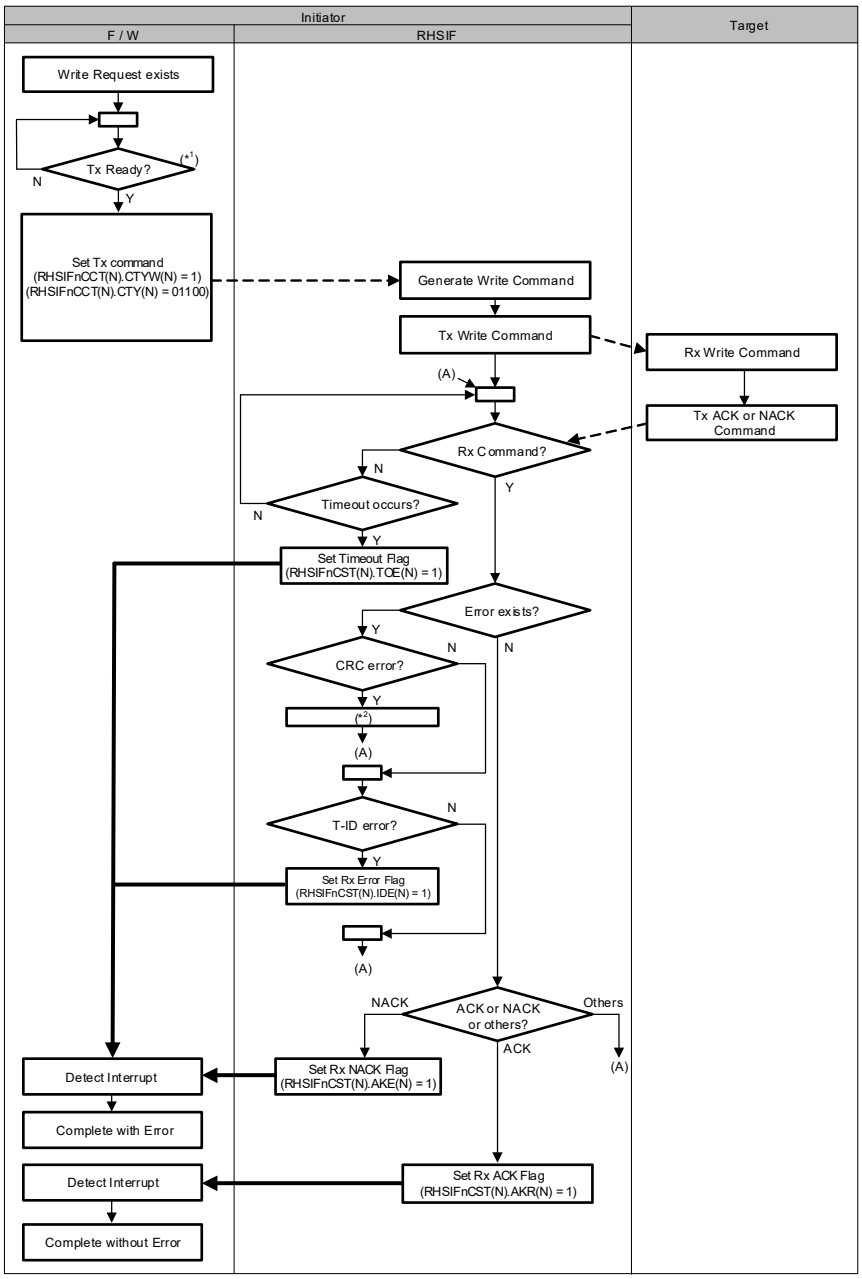
Figure 29.14 Tx Read Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

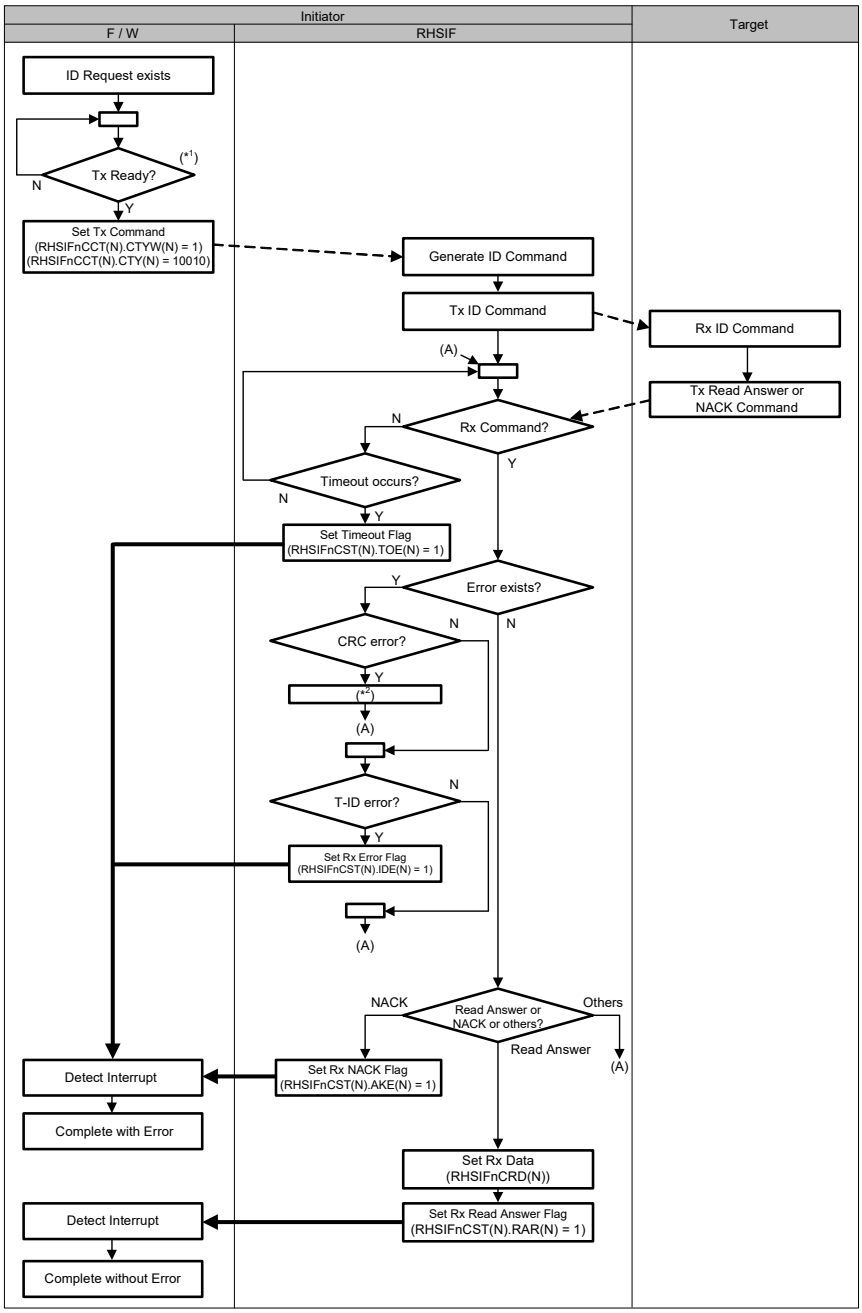
Figure 29.15 Tx Write Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

Figure 29.16 Tx Event Command Operation



Note 1. RHSIFnCST(N).RDY(N) = 1

Note 2. In case Rx Command has CRC error, RHSIF performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

Figure 29.17 Tx ID Command Operation

29.4.3.3 Non-Stream Command Reception by Target Node

Figure 29.18 to **Figure 29.21** show the flows for receiving non-stream requests (Read, Write, Event, and ID Commands) by using the target node function of L2.

Because the hardware of L2 automatically processes Commands that are received, the firmware of L2 does not need to perform any processing.

The following describes the reception operation, using the Read Command reception (**Figure 29.18**) for example.

Upon receiving a Command, L2 first checks whether an error has occurred. When an error is detected, L2 discards the command that is received without returning any reply Command. In the case the command that is received has a CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

When no error is detected, L2 checks the following items to determine whether read access is enabled:

(1) Whether the target node function has been enabled for the relevant channel

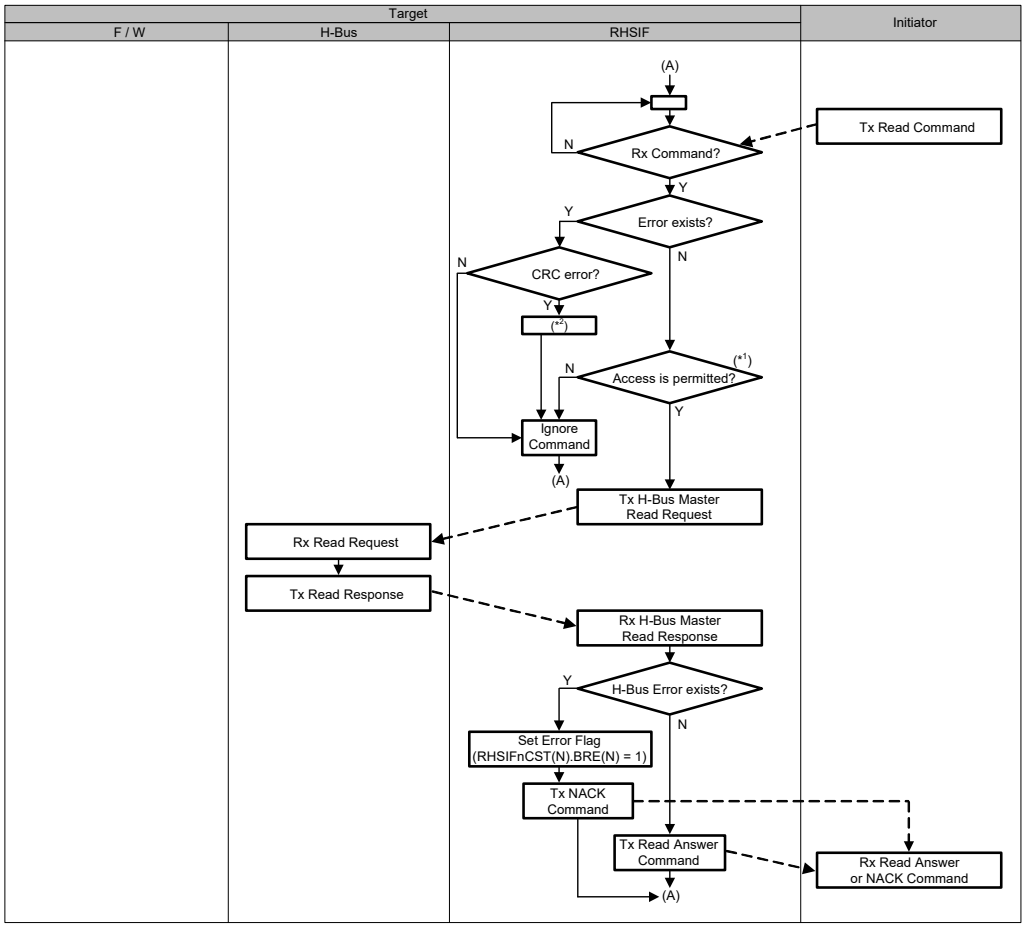
(2) Whether authentication has been performed (when authentication is required)

(3) Whether the target address corresponds to a valid memory window

(4) Whether reading of the window described in (3) above is permitted

If the result of any of the above checks is negative, L2 discards the command that is received without returning any reply Command.

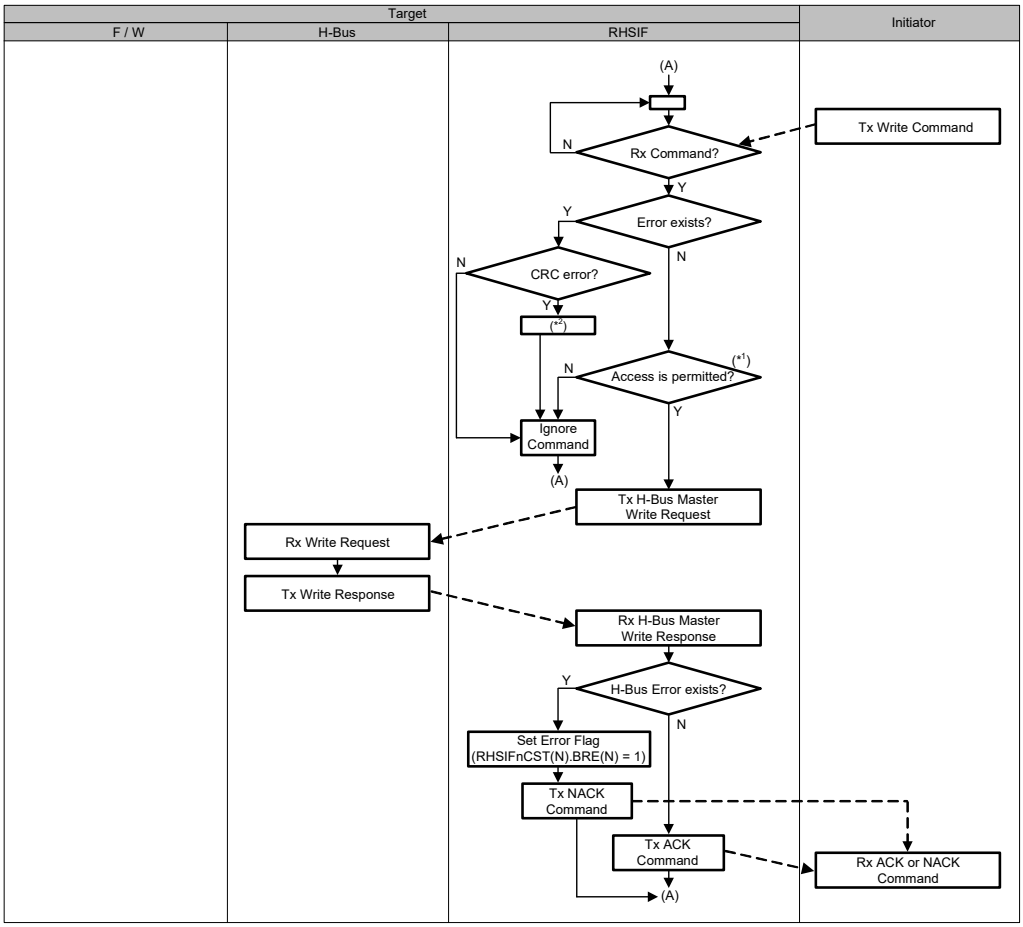
If the results of all of the above checks are positive, L2 issues a Master Read request to the H-Bus. After receiving a response from the H-Bus, L2 transmits a Read Answer Command (when no H-Bus response error is detected) or an NACK Command (when an H-Bus response error is detected) to the link partner.



Note 1. All of following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSed. (As required)
3. Address is inside of any Memory Windows which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
4. Read is allowed for a Memory Window whose address is hit (RHSIFnMCT.W(X)RE = 1)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

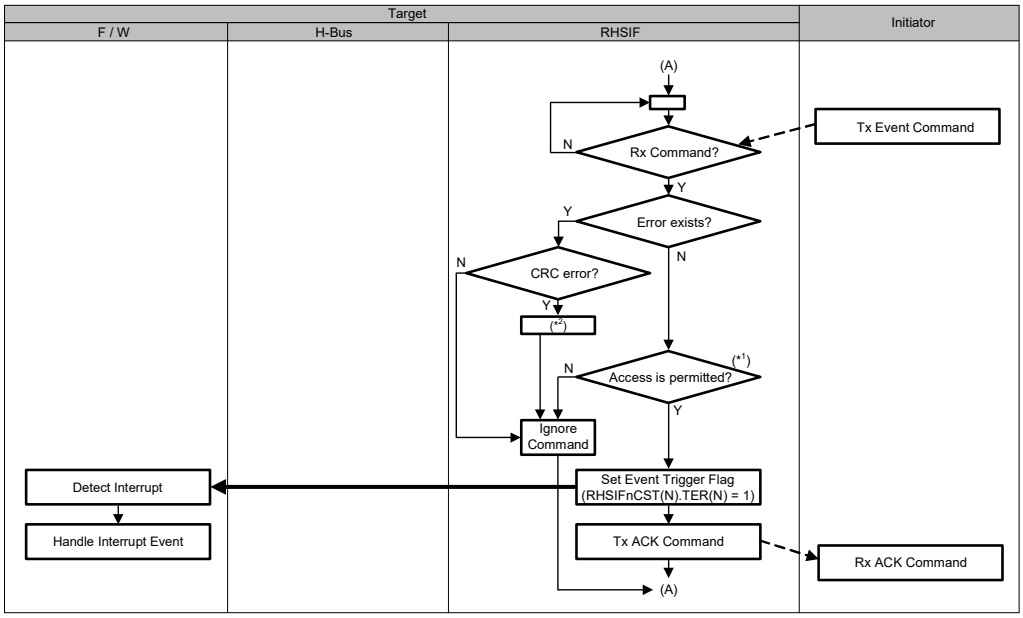
Figure 29.18 Rx Read Command Operation



Note 1. All of following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSed. (As required)
3. Address is inside of any Memory Windows which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
4. Write is allowed for a Memory Window whose address is hit (RHSIFnMCT.W(X)WE = 1)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

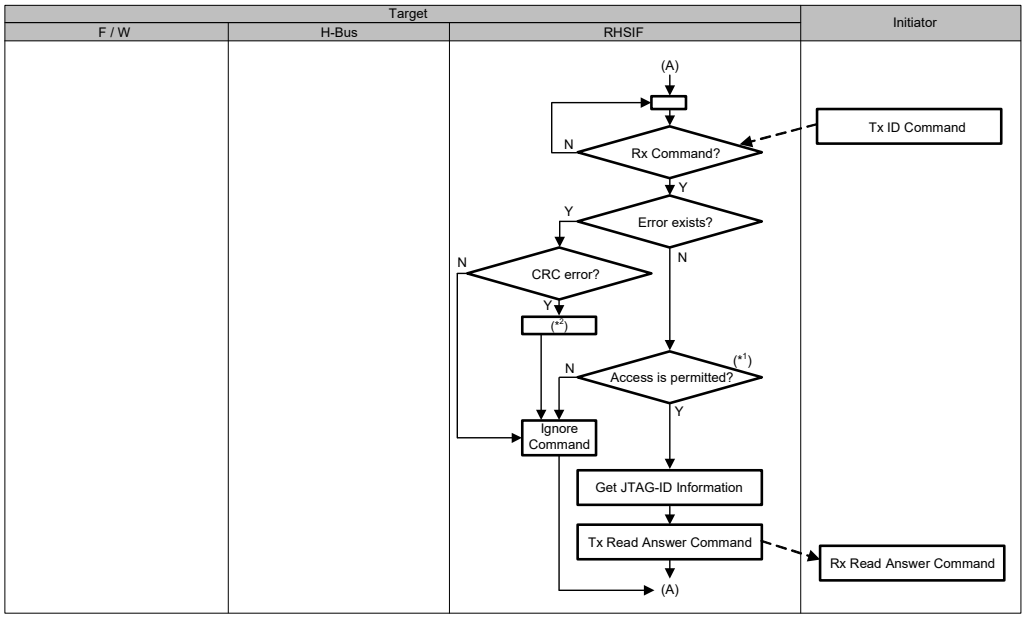
Figure 29.19 Rx Write Command Operation



Note 1. The following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSed. (As required)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

Figure 29.20 Rx Event Command Operation



Note 1. The following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSed. (As required)

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

Figure 29.21 Rx ID Command Operation

29.4.3.4 Stream Command Transmission by Initiator Node

Figure 29.22 and **Figure 29.23** show the flow for transmitting Stream Commands by using the initiator node function and internal DMAC of L2.

After preparing stream data, the firmware must set the start address of the area containing the data and the number of bytes to be transferred, and then enable the DMAC.

L2 internally has a 512-bit data buffer, and obtains data by issuing a Master Read request to the H-Bus with the maximum size is 256-bit. For L2, the maximum number of outstanding requests to the H-Bus is set to 2.

If source address does not hit any window or read operation is not allowed for a memory window whose address is hit RHSIF does not issue H-Bus read request. When RHSIF collects all the responses to transmitted H-Bus requests and transmitted request commands, it sets the RHSIFnSTST.STE4 bit, and then stops the DMAC.

If an error is detected in the response from the H-Bus, L2 collects all the responses to transmitted H-Bus requests and transmitted request commands, it sets the RHSIFnSTST.STE3 bit, and then stops the DMAC.

If no error is detected in the response from the H-Bus, L2 generates and transmits a Stream Command in units of the size specified by the RHSIFnSTMD.STPS bit. RHSIF allows up to two Stream Commands to be transmitted to a link partner before the link partner responds. Compliant with this specification, L2 can transmit a maximum of two outstanding requests to a link partner.

The Stream Command data to be transmitted is realigned in L2 so that the data alignment meets the specification. For the data alignment, see **Figure 29.24**.

After normally transmitting all data (specified number of bytes to be transferred), L2 sets the RHSIFnSTST.STC bit, and (when interrupt is enabled) generates an int_hsif_str interrupt.

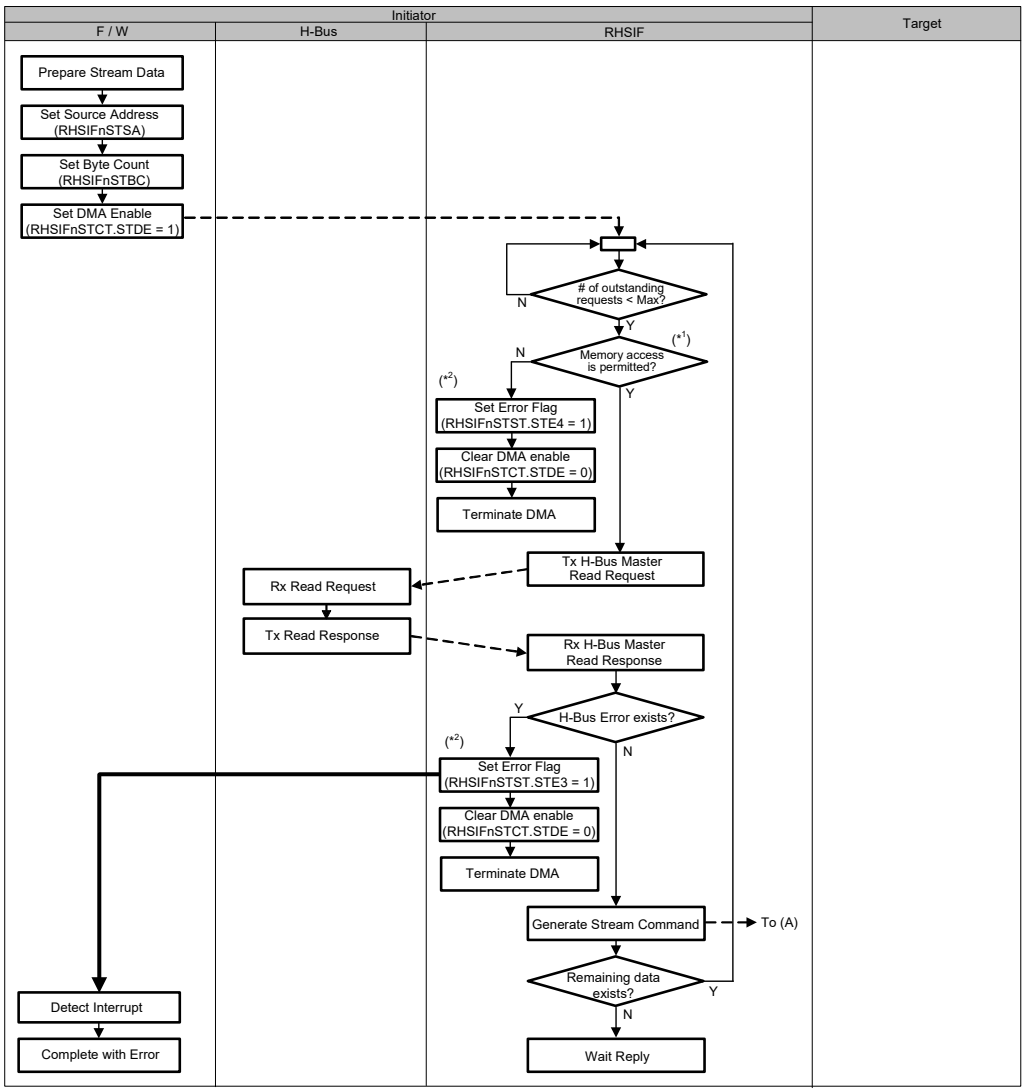
If an error (including timeout) is caused by a reply command during transmission, L2 collects all the responses to transmitted H-Bus requests and transmitted request commands, it sets the RHSIFnSTST.STE1 or RHSIFnSTST.STE2 bit, and then stops the DMAC.

Note that the operation L2 performs when it receives an NACK Command depends on the value of the RHSIFnSTMD.STNK bit. When the RHSIFnSTMD.STNK bit is 0, L2 regards reception of the NACK Command as occurrence of an error, it sets the RHSIFnSTST.STE0 bit, and then stops the DMA transfer. When the RHSIFnSTMD.STNK bit is 1, L2 regards reception of the NACK Command as reception of an ACK Command, and continues the DMA transfer. Specify an appropriate value in the RHSIFnSTMD.STNK bit according to the usage of the data to be transferred.

If more than one error (receives NACK, detects T-ID error, or reply timeout) occur during a DMA operation, only first error is set to RHSIFnSTST.STE0, STE1, or STE2.

In case reply Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

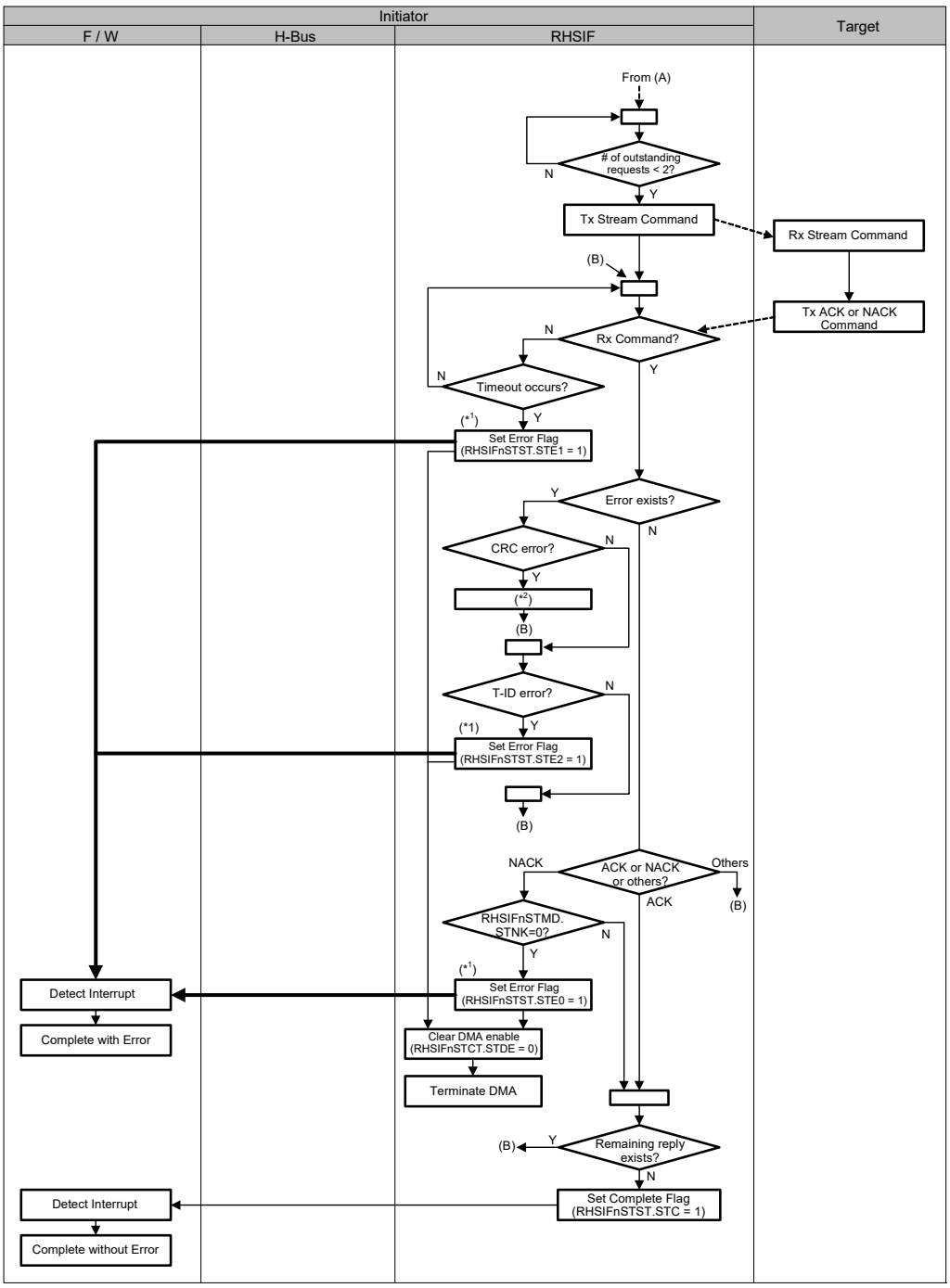
Note that Stream Commands must not be issued together with a non-Stream Command (Read, Write, Event, or ID Command). The firmware must be controlled so that non-Stream Commands are not transmitted while a Stream Command is being transmitted.



Note 1. All of following conditions are satisfied:
1. Source address is inside of any memory window which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
2. Read is allowed for a memory window whose address is hit (RHSIFnMCT.W(X)WE=1)

Note 2. This bit is set after all responses/replies are received for all H-Bus requests/request commands that are already transmitted.

Figure 29.22 Tx Stream Command Operation (1)



Note 1. This bit is set after all responses/replies are received for all H-Bus requests/request commands that are already transmitted.

Note 2. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

Figure 29.23 Tx Stream Command Operation (2)

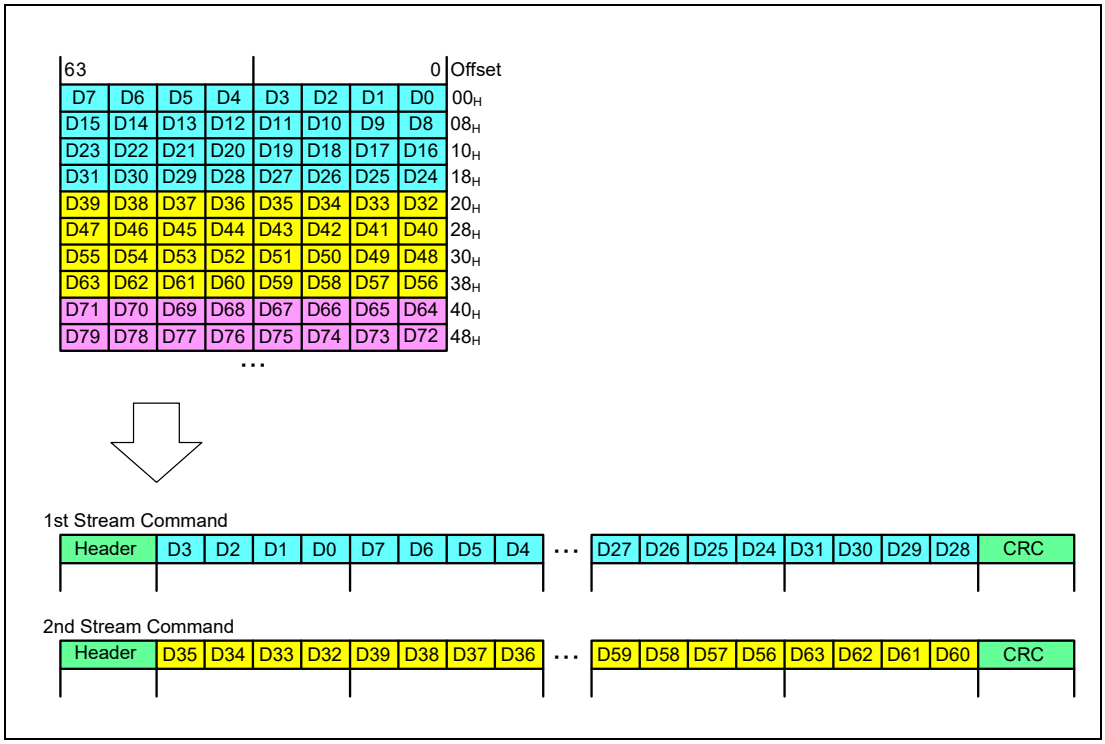


Figure 29.24 Tx Stream Data Alignment

29.4.3.5 Stream Command Reception by Target Node

Figure 29.25 and **Figure 29.26** show the flow for receiving Stream Commands by using the target node function and internal DMAC of L2.

First, the type of interrupt must be selected. L2 provides the following two status types:

(1) Status indicating that the size of unprocessed data is not 0 (RHSIFnSRST.SRC0)

(2) Status indicating that the size of unprocessed data is at least the specified size (RHSIFnSRST.SRC1)

An appropriate type must be selected according to the system.

Upon receiving a Stream Command, L2 first checks whether an error has occurred. When an error is detected, L2 discards the received Command without returning any reply Command. In the case the command that is received has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

When the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

When no error is detected, L2 checks the following items to determine whether stream transfer is enabled:

(1) Whether the target node function has been enabled for channel 2

(2) Whether authentication has been performed (when authentication is required)

(3) Whether DMA for stream reception has been enabled

If the result of any of the above checks is negative, L2 discards the command that is received without returning any reply Command.

If destination address does not hit any window or write operation is not allowed for a memory window whose address is hit, RHSIF discards the command that is received without returning any reply Command, and sets the RHSIFnSRST.STE2 bit to 1. When the RHSIFnSRMD.SRMC bit is 0, RHSIF automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

If the results of all the above checks are positive, L2 checks the size of the free space in the stream storage buffer. If the buffer does not have enough free space to store the data that is received, L2 discards the command that is received, returns a NACK Command, and sets the RHSIFnSRST.SRE1 bit to 1. When the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

If the buffer has enough free space to store received data, L2 issues a Master Write request to the H-Bus. RHSIF allows up to two Stream Commands to be transmitted to a link partner before the link partner responds. Compliant with this specification, L2 can receive up to two Stream Commands concurrently. The maximum number of outstanding requests to the H-Bus is 2.

The data alignment of the Stream Commands that are received complies with the specifications. Therefore, the Stream Command data to be transmitted is realigned in L2 before transmission to the H-Bus. For the data alignment, see **Figure 29.24**.

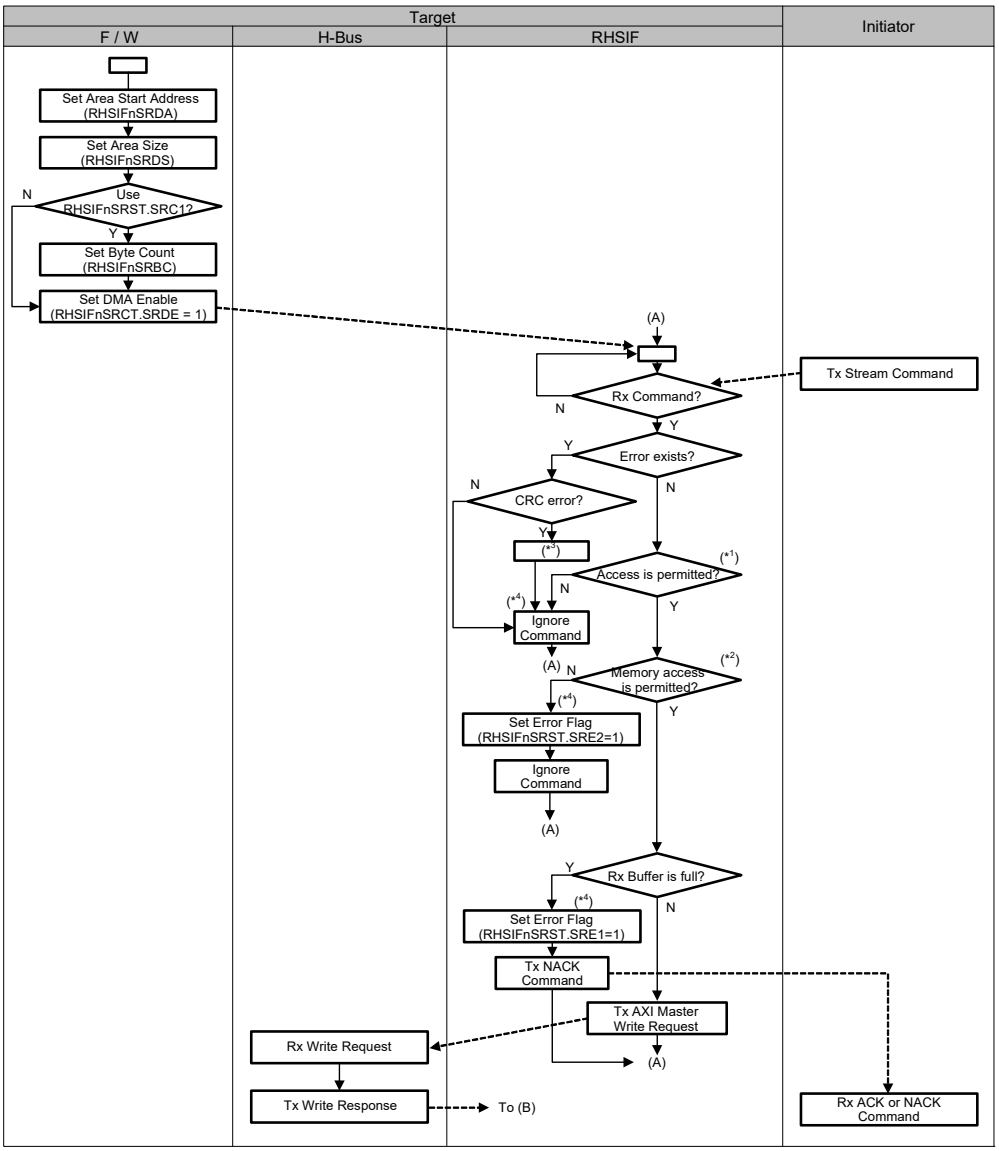
After receiving a response from the H-Bus, L2 subsequently performs one of the following operations depending on whether an error occurs in the response from the H-Bus:

(1) When no H-Bus response error is detected
L2 updates the write pointer (RHSIFnSRWP) for the data payload size of the command that is received, and returns an ACK Command to the link partner.

(2) When an H-Bus response error is detected
L2 updates the write pointer (RHSIFnSRWP) for the data payload size of the command that is received, returns a NACK Command to the link partner, and sets the RHSIFnSRST.SRE0 bit to 0. When the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit to 0 and sets the RHSIFnSRST.SRA to prevent extra data from being stored subsequently.

If the values of write pointer and read pointer differ from each other, an int_hsif_str interrupt is generated (when the RHSIFnSRIE.SRCE0 bit is 1). Also, if the size of unprocessed data reaches or exceeds the size specified in the RHSIFnSRBC register, an int_hsif_str interrupt is generated (when the RHSIFnSRIE.SRCE1 bit is 1).

The firmware must check the data size with the write pointer as needed, and process the data in the data storage area. After data processing ends, the firmware must update the read pointer for the size of processed data. Updating of the read pointer must always be performed in units of the size specified by the RHSIFnSRMD.SRPS bit. **Figure 29.27** shows an example of updating the pointer.



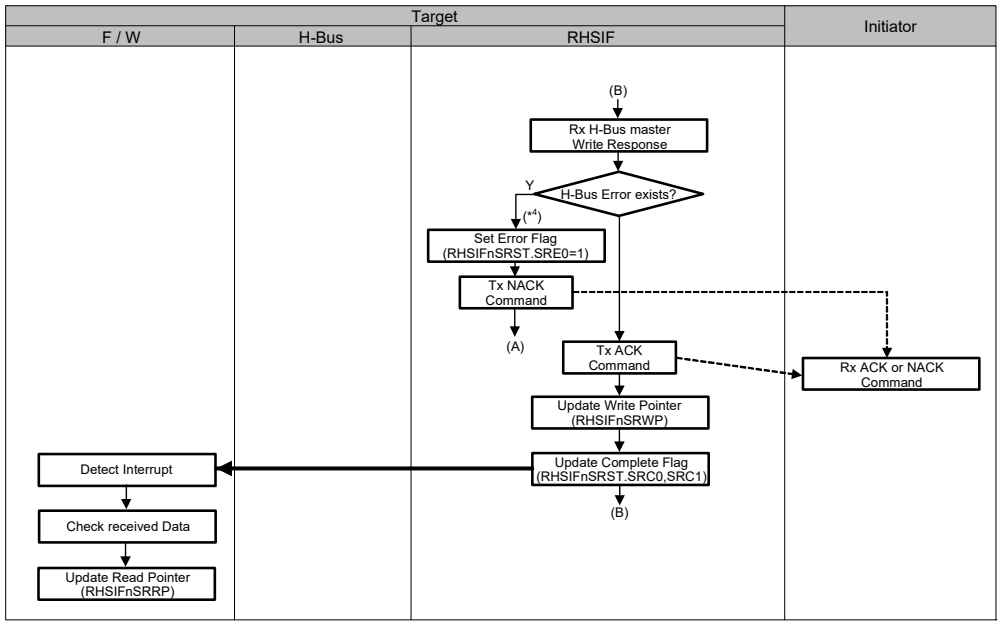
Note 1. The following conditions are satisfied:
1. RHSIFnCMD(N).TNME(N) = 1
2. Authentication(s) is PASSed. (As required)
3. DMA is enabled (RHSIFnSRCT.SRDE = 1)

Note 2. All of following conditions are satisfied:
1. Destination address is inside of any memory window which are defined by RHSIFnMW(X)A and RHSIFnMW(X)S
2. Write is allowed for a memory window whose address is hit (RHSIFnMCT.W(X)WE=1)

Note 3. In case Rx Command has CRC error, L2 performs special action. For details, see **Section 29.4.3.6, CRC Error Handling**.

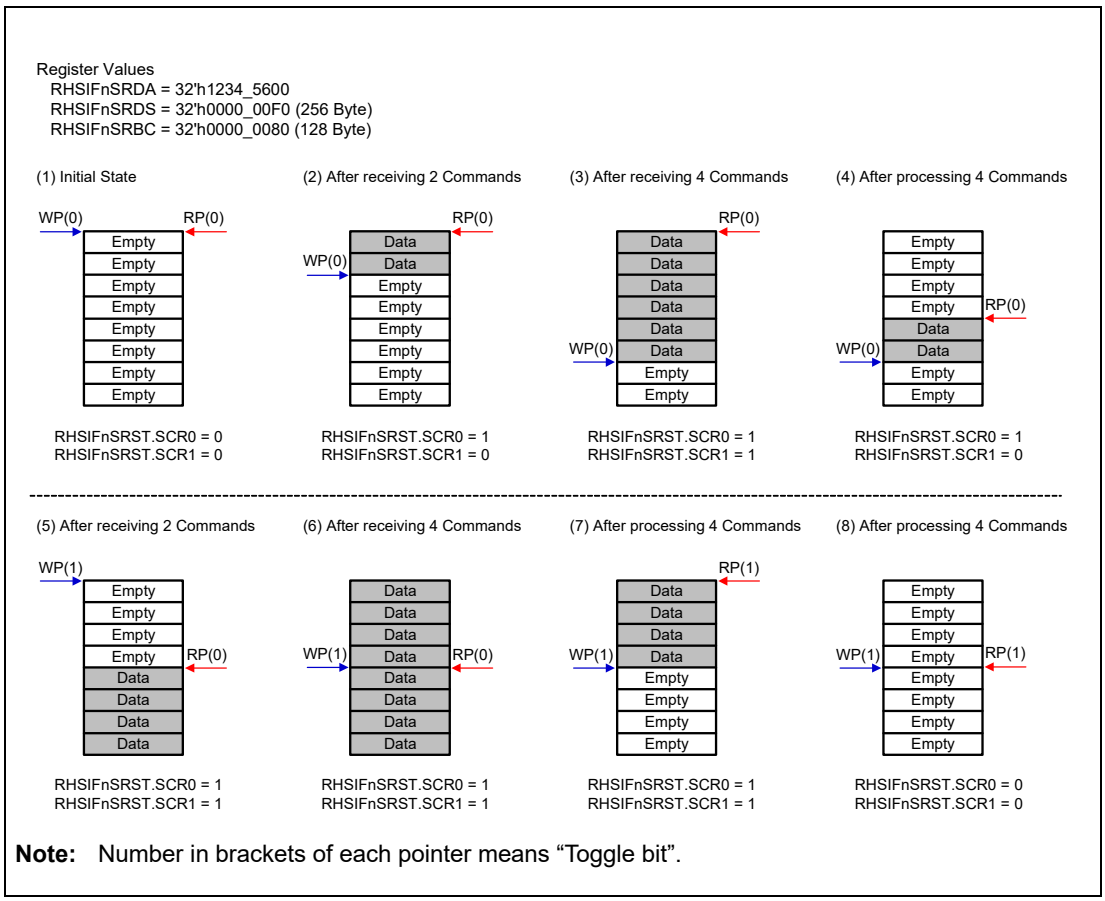
Note 4. If DMA is enabled (RHSIFnSRCT.SRDE = 1), clear DMA Enable (RHSIFnSRCT.SRDE) and set DMA abort (RHSIFnSRST.SRA) if RHSIFnSRMD.SRMC = 0.

Figure 29.25 Rx Stream Command Operation(1)



Note 4. If DMA is enabled (RHSIFnSRCT.SRDE = 1), clear DMA Enable (RHSIFnSRCT.SRDE) and set DMA abort (RHSIFnSRST.SRA) if RHSIFnSRMD.SRMC = 0.

Figure 29.26 Rx Stream Command Operation(2)



Note: Number in brackets of each pointer means “Toggle bit”.

Figure 29.27 Rx Stream Pointer Operation Example

If an error occurs during Stream Command reception when the RHSIFnSRMD.SRMC bit is 0, L2 automatically clears the RHSIFnSRCT.SRDE bit and stops receiving any more Commands. (For detailed conditions, see the description of the RHSIFnSRCT.SRDE bit.)

If reception of Stream Commands is to be resumed in this status, the following procedure must be followed:

(1) Detect interrupt by the RHSIFnSRST.SRA bit. (If RHSIFnSRIE.SRAE bit is 1.)

(2) Check that the RHSIFnSRCT.SRDE bit is 0.

(3) Wait until L2 can check that the RHSIFnSRST.SRTA bit is 0.

(4) Reset the RHSIFnSRCT.SRDE bit to 1.

This procedure also applies when the firmware has cleared the RHSIFnSRCT.SRDE bit to 0 because of the system. (In this case, RHSIFnSRST.SRA is not set.)

29.4.3.6 CRC Error Handling

If a CRC error is detected in a command that is received, the header information also cannot be trusted, and, therefore, the command that is received cannot be determined to be a request or a reply. For this reason, the operation of L2 differs from those other errors.

Upon receiving a Command involving a CRC error, L2 performs the following operations:

- Sets the RHSIFnCST(N).CRE(N) bit
- Discards the command that is received
- Clears the RHSIFnSRCT.SRDE bit when the error has occurred in channel 2 and the RHSIFnSRMD.SRMC bit is 0

Based on the above operations, necessary processing must be executed when a CRC error is detected.

29.4.4 Interrupt Specification

29.4.4.1 Interrupt Signals and Corresponding Interrupt Sources

Table 29.56 shows a list of interrupt signals, interrupt status register fields, interrupt source register fields and interrupt enable register fields. The interrupt signal is output as long as the corresponding interrupt enable bit is set to 1 and the corresponding status / flag bit is set to 1.

All interrupt signals are synchronized to ACLK, level signal, and active high.

Table 29.56 Interrupt Signal List (1/2)

Signal	Status	Source	Enable	Description
int_hsif_ch0	RHSIFnMIST.RARS0	RHSIFnCST0.RAR0	RHSIFnCIE0.RARE0	Read Answer received at Ch.0
	RHSIFnMIST.AKRS0	RHSIFnCST0.AKR0	RHSIFnCIE0.AKRE0	ACK received at Ch.0
	RHSIFnMIST.TERS0	RHSIFnCST0.TER0	RHSIFnCIE0.TERE0	Event received at Ch.0
int_hsif_ch1	RHSIFnMIST.RARS1	RHSIFnCST1.RAR1	RHSIFnCIE1.RARE1	Read Answer received at Ch.1
	RHSIFnMIST.AKRS1	RHSIFnCST1.AKR1	RHSIFnCIE1.AKRE1	ACK received at Ch.1
	RHSIFnMIST.TERS1	RHSIFnCST1.TER1	RHSIFnCIE1.TERE1	Event received at Ch.1
int_hsif_ch2	RHSIFnMIST.RARS2	RHSIFnCST2.RAR2	RHSIFnCIE2.RARE2	Read Answer received at Ch.2
	RHSIFnMIST.AKRS2	RHSIFnCST2.AKR2	RHSIFnCIE2.AKRE2	ACK received at Ch.2
	RHSIFnMIST.TERS2	RHSIFnCST2.TER2	RHSIFnCIE2.TERE2	Event received at Ch.2
int_hsif_ch3	RHSIFnMIST.RARS3	RHSIFnCST3.RAR3	RHSIFnCIE3.RARE3	Read Answer received at Ch.3
	RHSIFnMIST.AKRS3	RHSIFnCST3.AKR3	RHSIFnCIE3.AKRE3	ACK received at Ch.3
	RHSIFnMIST.TERS3	RHSIFnCST3.TER3	RHSIFnCIE3.TERE3	Event received at Ch.3
int_hsif_str	RHSIFnMIST.STCS	RHSIFnSTST.STC	RHSIFnSTIE.STCE	Tx stream completed
	RHSIFnMIST.SRCS0	RHSIFnSRST.SRC0	RHSIFnSRIE.SRCE0	Rx stream completed (more than 1 Stream Command data size)
	RHSIFnMIST.SRCS1	RHSIFnSRST.SRC1	RHSIFnSRIE.SRCE1	Rx stream completed (more than RHSIFnSRBC)
int_hsif_err	RHSIFnMIST.CERS(N)	RHSIFnCST0.AKE(N)	RHSIFnCIE0.AKEE(N)	NACK received at Ch.(N)
		RHSIFnCST0.TOE(N)	RHSIFnCIE0.TOEE(N)	Reply timeout detected at Ch.(N)
		RHSIFnCST0.IDE(N)	RHSIFnCIE0.IDEE(N)	Transaction-ID error detected at Ch.(N)
		RHSIFnCST0.AOE(N)	RHSIFnCIE0.AOEE(N)	Any other error detected at Ch.(N)
		RHSIFnCST0.BRE(N)	RHSIFnCIE0.BREE(N)	Bus error detected at Ch.(N)
		RHSIFnCST0.CRE(N)	RHSIFnCIE0.CREE(N)	CRC error detected at Ch.(N)
	RHSIFnMIST.STES	RHSIFnSTST.STE0	RHSIFnSTIE.STEE0	NACK received at Tx stream Ch.
		RHSIFnSTST.STE1	RHSIFnSTIE.STEE1	Reply timeout detected at Tx stream Ch
		RHSIFnSTST.STE2	RHSIFnSTIE.STEE2	Transaction-ID error detected at Tx stream Ch.
		RHSIFnSTST.STE3	RHSIFnSTIE.STEE3	Bus error detected at Tx stream Ch.
		RHSIFnSTST.STE4	RHSIFnSTIE.STEE4	Window mis-hit detected at Tx stream Ch.
	RHSIFnMIST.SRES	RHSIFnSRST.SRE0	RHSIFnSRIE.SREE0	Bus error detected at Rx stream Ch.
		RHSIFnSRST.SRE1	RHSIFnSRIE.SREE1	Buffer overflow detected at Rx stream Ch.
		RHSIFnSRST.SRE2	RHSIFnSRIE.SREE2	Window mis-hit detected at Rx stream Ch.
		RHSIFnSRST.SRA	RHSIFnSRIE.SRAE	Rx stream aborted by error.

Table 29.56 Interrupt Signal List (2/2)

Signal	Status	Source	Enable	Description
int_hsif_sec	RHSIFnMIST.AESS	RHSIFnAEST.AEE(N)	RHSIFnAEIE.AEEE(N)	Event received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AEI(N)	RHSIFnAEIE.AEIE(N)	ID received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AER(N)	RHSIFnAEIE.AERE(N)	Read received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AEW(N)	RHSIFnAEIE.AEWE(N)	Write received at Ch.(N) before authentication is completed.
		RHSIFnAEST.AES	RHSIFnAEIE.AESE	Stream received at Ch.2 before authentication is completed.
		RHSIFnAEST.WER(N)	RHSIFnAEIE.WERE(N)	Received Read at Ch.(N) does not hit any memory window.
		RHSIFnAEST.WEW(N)	RHSIFnAEIE.WEWE(N)	Received Write at Ch.(N) does not hit any memory window.
		RHSIFnAEST.WES0	RHSIFnAEIE.WESE0	H-Bus Read for Stream does not hit any window.
		RHSIFnAEST.WES1	RHSIFnAEIE.WESE1	H-Bus Write for Stream does not hit any window.

Figure 29.28 shows the logic for interrupt generation. The corresponding bit in the source register / field is ANDed with the corresponding enable bit in the interrupt enable register / field. Moreover, respective factors are ORed to an interrupt signal.

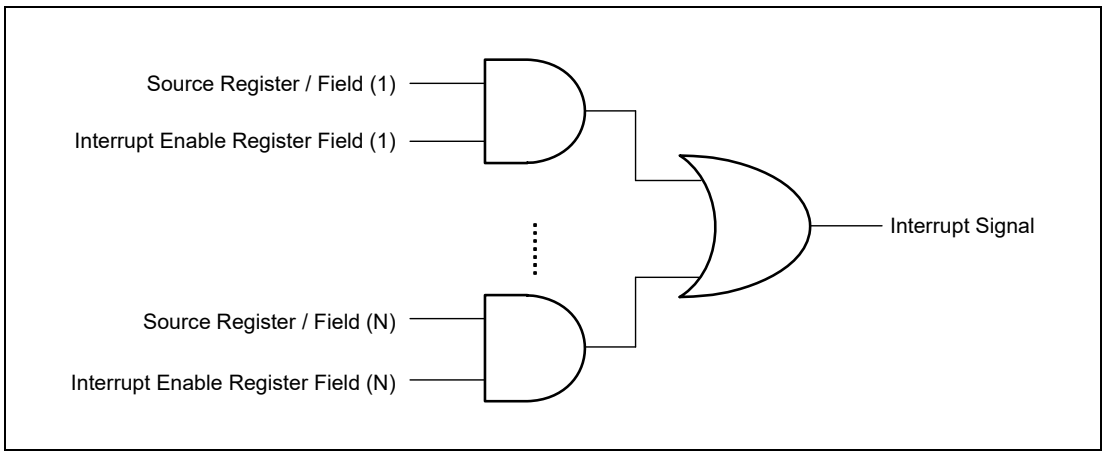


Figure 29.28 Interrupt Signal Generation Logic

29.4.4.2 Interrupt Signal Status Register

L2 has a register to monitor the interrupt factor state. This register (RHSIFnMIST) indicates the current state of the interrupt factors. Whether an interrupt is active can be checked by a single register read.

29.5 Datalink Layer (L1)

29.5.1 Overview

Regarding the basic functions and specifications, refer to the protocol specifications.

29.5.1.1 Functional Overview

The Renesas High-speed Serial Interface Datalink Layer IP (henceforth, called “L1”) is compliant with following specifications.

(1) L1

- Addition of L1 header information to the L2 frame for data transmission
- Removal of the L1 header from the receive data and transmission to L2
- Generation and transmission of Interface Control Logical Channel (hereafter, ICLC) commands
- Reception and execution of ICLC commands
- Generation of interrupt sources for transmission completion, transmission error, reception completion, reception error, and ICLC command reception
- Generation and transmission of CTS commands
- Frame arbitration: Fixed (Higher) PingAnswer → CTS Command → ICLC Command → L2 Frame (lower)
- Execution of flow control
- Execution of Loopback test mode
- Execution of toggle pattern test mode

(2) Physical Interface

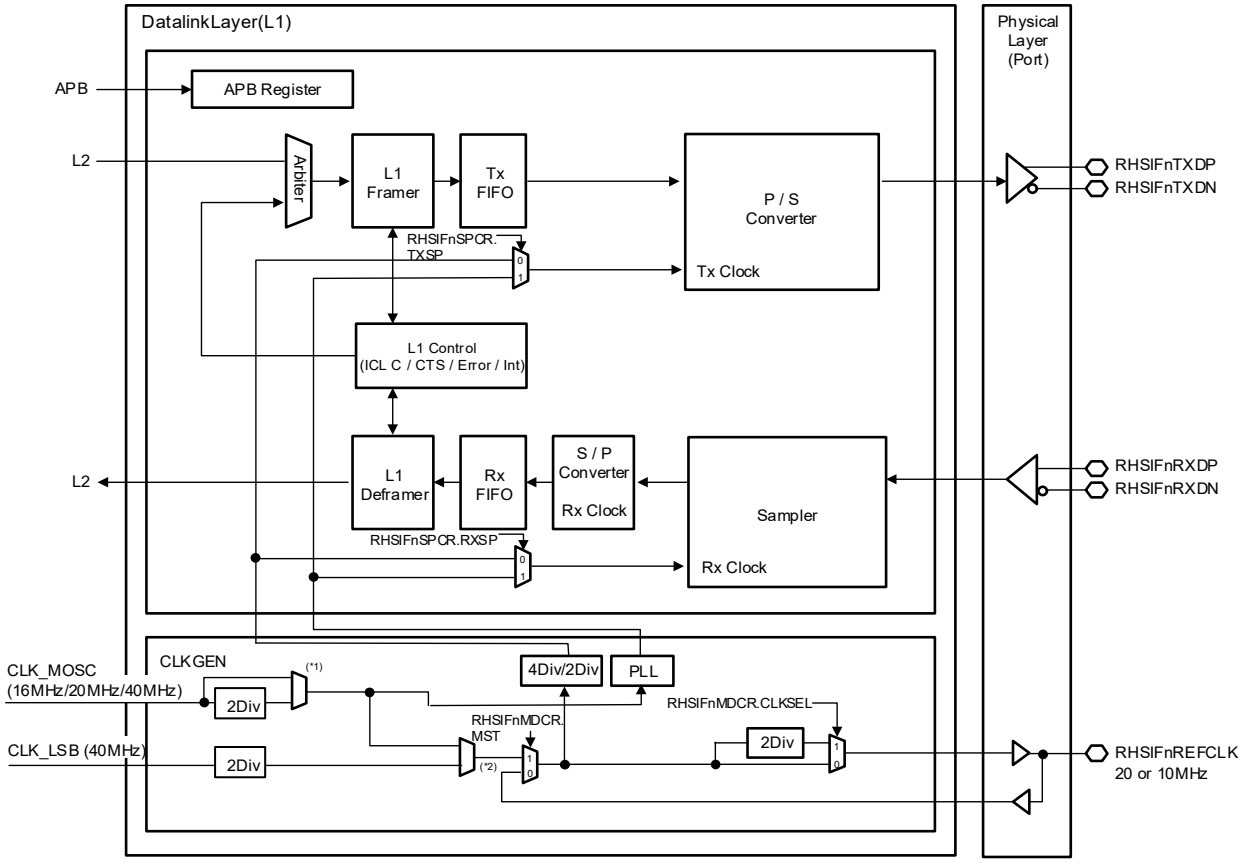
This layer consists of LVDS differential interface (input and output) and the reference clock. Regarding the basic functions and specifications, refer to the protocol specification.

Table 29.57 Physical Interface

External Port	Specification Item	Specification
Clock (RHSIFnREFCLK)	Electrical Characteristics	TTL / CMOS
	Frequency	20 MHz,10 MHz
Data (RHSIFnTXDP, RHSIFnTXDN, RHSIFnRXDP, RHSIFnRXDN)	Electrical Characteristics	LVDS (based on IEEE1596.3-1996 reducedrangelink)
	BaudRate (FastMode) Using PLL	320 M Baud, 160 M Baud, 80 M Baud selectable
	BaudRate (SlowMode) Using REFCLK	5 M Baud

29.5.1.2 Block Diagram

(1) L1



Note 1. When CLK_MOSC frequency is 16MHz / 20MHz, it selects CLK_MOSC.
When CLK_MOSC frequency is 40MHz, it selects CLK_MOSC divided into 2: CLK_MOSC/2

Note 2. When CLK_MOSC frequency is 20MHz / 40MHz, it selects clock outputted from **Note 1.**: CLK_MOSC or CLK_MOSC/2.
When CLK_MOSC frequency is 16MHz, it selects CLK_LSB divided into 2: CLK_LSB/2

Figure 29.29 Block Diagram of L1 and Physical Interface

CAUTION

When CLK_MOSC frequency is 24MHz, RHSIF can NOT operate normally.

29.5.2 Registers

This section describes the L1 registers in detail.

In case of a concurrent update of the same register bit, L1 module write (hardware write) has higher priority than CPU write (software write).

29.5.2.1 List of Registers

The register map of an L1 module is shown in **Table 29.58**.

The L1 module remains in reset while RESET.

The values shown in **Table 29.58** in the L1 registers description under “Values after Reset” are related to this reset state.

For further information about the L1 operation modes, refer to **Section 29.5.3, Operation**.

The module specific base address of an L1 module needs to be added to each of the specified offset addresses.

Table 29.58 Registers Address Map

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					PBG	Other
RHSIFn	Mode Control Register	RHSIFnMDCR	<RHSIFn_L1_base> + 000_H	32	PBG90#2	—
	Sleep Mode Control Register	RHSIFnSMCR	<RHSIFn_L1_base> + 004_H	32	PBG90#2	—
	PLL Control Register	RHSIFnPCR	<RHSIFn_L1_base> + 008_H	32	PBG90#2	—
	Speed Control Register	RHSIFnSPCR	<RHSIFn_L1_base> + $00C_H$	32	PBG90#2	—
	Test Mode Control Register	RHSIFnTMDCR	<RHSIFn_L1_base> + 010_H	32	PBG90#2	—
	L1 Status Register	RHSIFnL1SR	<RHSIFn_L1_base> + 018_H	32	PBG90#2	—
	Last Tx Frame L1 Result Register	RHSIFnLTXFRMRL1	<RHSIFn_L1_base> + 030_H	32	PBG90#2	—
	Last Rx Frame L1 Result Register	RHSIFnLRXFRMRL1	<RHSIFn_L1_base> + 034_H	32	PBG90#2	—
	Last Tx Frame L2 Result Register	RHSIFnLTXFRMRL2	<RHSIFn_L1_base> + 038_H	32	PBG90#2	—
	Last Rx Frame L2 Result Register	RHSIFnLRXFRMRL2	<RHSIFn_L1_base> + $03C_H$	32	PBG90#2	—
	TX/RX Control Register	RHSIFnTXRXCR	<RHSIFn_L1_base> + 050_H	32	PBG90#2	—
	ICLC Command Control Register	RHSIFnICCR	<RHSIFn_L1_base> + 060_H	32	PBG90#2	—
	CTS Frame Control Register	RHSIFnCCR	<RHSIFn_L1_base> + 070_H	32	PBG90#2	—
	Tx Complete Status Register	RHSIFnTXCMPST	<RHSIFn_L1_base> + 080_H	32	PBG90#2	—
	Tx Complete Status Clear Register	RHSIFnTXCMPSC	<RHSIFn_L1_base> + 084_H	32	PBG90#2	—
	Tx Complete Interrupt Enable Register	RHSIFnTXCMPIE	<RHSIFn_L1_base> + 088_H	32	PBG90#2	—
	Tx Error Status Register	RHSIFnTXERRST	<RHSIFn_L1_base> + 090_H	32	PBG90#2	—
	Tx Error Status Clear Register	RHSIFnTXERRSC	<RHSIFn_L1_base> + 094_H	32	PBG90#2	—
	Tx Error Interrupt Enable Register	RHSIFnTXERRIE	<RHSIFn_L1_base> + 098_H	32	PBG90#2	—
	Rx Complete Status Register	RHSIFnRXCMPST	<RHSIFn_L1_base> + $0A0_H$	32	PBG90#2	—
	Rx Complete Status Clear Register	RHSIFnRXCMPSC	<RHSIFn_L1_base> + $0A4_H$	32	PBG90#2	—
	Rx Complete Interrupt Enable Register	RHSIFnRXCMPIE	<RHSIFn_L1_base> + $0A8_H$	32	PBG90#2	—
	Rx Error Status Register	RHSIFnRXERRST	<RHSIFn_L1_base> + $0B0_H$	32	PBG90#2	—
	Rx Error Status Clear Register	RHSIFnRXERRSC	<RHSIFn_L1_base> + $0B4_H$	32	PBG90#2	—
	Rx Error Interrupt Enable Register	RHSIFnRXERRIE	<RHSIFn_L1_base> + $0B8_H$	32	PBG90#2	—
	Rx ICLC Command Status Register	RHSIFnRXICST	<RHSIFn_L1_base> + $0C0_H$	32	PBG90#2	—
	Rx ICLC Command Status Clear Register	RHSIFnRXICSC	<RHSIFn_L1_base> + $0C4_H$	32	PBG90#2	—
	Rx ICLC Command Interrupt Enable Register	RHSIFnRXICIE	<RHSIFn_L1_base> + $0C8_H$	32	PBG90#2	—

29.5.2.2 Legend

This section explains the module state dependent abbreviations used for the L1 registers description.

Conditions:

- R/W: Bit is readable and writable
- R: Read-only bit; the user cannot write to this bit
- W: Write-only bit; Read value is always “0”

Reserved bits:

- Always read as 0
- Writing to these bits has no effect (hardware protection)

29.5.2.3 Detailed Register Description

Detailed specifications of each register are described below.

(1) RHSIFnMDCR — Mode Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 000_H

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTSEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTSV	—	—	—	—	—	—	CLKSEL	MST
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 29.59 RHSIFnMDCR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	CTSEN	This bit specifies whether to enable the flow control. 0: Disable automatic flow control. 1: Enable automatic flow control. Set this bit to “1” to enable flow control and stop next frame transfer. See **Section 29.5.3.5, CTS Commands**.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CTSV	This bit specifies CTS Value of L1 header for all frames. 0: Frame unreceivable. 1: Frame receivable.
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CLKSEL	This bit specifies REFCLK frequency. 0: 20MHz. 1: 10MHz.
0	MST	This bit specifies whether Master or Slave. 0: Slave. 1: Master.

(2) RHSIFnSMCR — Sleep Mode Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SLP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.60 RHSIFnSMCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SLP	Specify whether to set the sleep mode during data transmission or in the next transmission frame. 0: L1 End bit = 0 (Normal mode for the link partner) 1: L1 End bit = 1 (Transitioning to sleep mode for the link partner)

(3) RHSIFnPCR — PLL Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 008_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLSTBY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.61 RHSIFnPCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PLLSTBY	This bit specifies whether PLL Standby or PLL Active. This bit is updated only when writing directly. It is not updated when ICLC (Slave interface PLL start, Slave interface PLL stop) command reception. 0: PLL Active. 1: PLL Standby.

(4) RHSIFnSPCR — Speed Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + $00C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	FMBR[1:0]		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TXSP	—	—	—	—	—	—	—	RXSP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 29.62 RHSIFnSPCR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 24	FMBR1	These bits specify Fast Speed Mode Baud Rate. 00: 80MBaud 01: 160MBaud 10: 320MBaud 11: Setting prohibited
23 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TXSP*1	This bit specifies whether Slow Speed Mode or Fast Speed Mode for data transmission. This bit is updated only when writing directly. It is not updated when ICLC (Select Slow Speed mode for transfers from the Slave interface to the Master interface, Select Fast Speed mode for transfers from the Slave interface to the Master interface) command reception. 0: Slow Speed Mode. 1: Fast Speed Mode.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RXSP*1	This bit specifies whether Slow Speed Mode or Fast Speed Mode for data reception. This bit is updated only when writing directly. It is not updated when ICLC (Select Slow Speed mode for transfers from the Master interface to the Slave interface, Select Fast Speed mode or transfers from the Master interface to the Slave interface) command reception. 0: Slow Speed Mode. 1: Fast Speed Mode.

Note 1. In case of setting '1' to RHSIFnSPCR.TXSP or RHSIFnSPCR.RXSP, set '0' to RHSIFnPCR.PLLSTBY for activating PLL.

(5) RHSIFnTMDCR — Test Mode Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CKTM	CKTMSEL	—	—	—	—	—	—	TXLPBK	RXLPBK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 29.63 RHSIFnTMDCR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	CKTM*1	This bit is specified for selecting clock test mode. This bit is updated when ICLC command is for "Turn on testmode" or "Turn off test mode" as well as writing to this bit directly. 0: Disable. 1: Clock test mode.
8	CKTMSEL	Clock Selection of Clock test mode 0: RX Baud Rate/2 Clock 1: TX Baud Rate/2 Clock
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TXLPBK*1*3	This bit is specified for selecting TX-RX loopback test mode. 0: Disable. 1: TX-RX loopback testmode.
0	RXLPBK*1*2	This bit is specified for selecting RX-TX loopback test mode. This bit is updated when ICLC command is for "Turn on payload loopback" or "Turn off test mode" as well as writing to this bit directly. 0: Disable. 1: RX-TX loopback test mode.

Note 1. Setting 2 bits or more to 1 at the same time is prohibited.

Note 2. In Rx-Tx loopback testmode, received ICLC Turn off command is output by looping back.

Note 3. In case of setting '1' to RHSIFnTMDCR.TXLPBK, set the same value to RHSIFnSPCR.TXSP and RHSIFnSPCR.RXSP. Unless the same Baud Rate is set to both transmitter/receiver, the communication will not be done normally.

(6) RHSIFnL1SR — L1 Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 018_H

Value after reset: $0001\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TFFUL	TFEMP	—	—	—	—	—	—	RFFUL	RFEMP
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.64 RHSIFnL1SR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	RCTS	This bit indicates the CTS value that is received. 0: Link partner is not receivable 1: Link partner is receivable
15 to 10	Reserved	When read, the value after reset is returned.
9	TFFUL	This bit indicates TX FIFO full status. 0: Not full. 1: Full.
8	TFEMP	This bit indicates TX FIFO empty status. 0: Not empty. 1: Empty.
7 to 2	Reserved	When read, the value after reset is returned.
1	RFFUL	This bit indicates RX FIFO full status. 0: Not full. 1: Full.
0	RFEMP	This bit indicates RX FIFO empty status. 0: Not empty. 1: Empty.

(7) RHSIFnLTXFRMRL1 — Last Tx Frame L1 Result Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TL1E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TL1P[7:0]								TL1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.65 RHSIFnLTXFRMRL1 Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	TL1E	This bit indicates last Tx Endbit.
15 to 8	TL1P[7:0]	These bits indicate Last Tx L1 Payload.
7 to 0	TL1H[7:0]	These bits indicate Last Tx L1 Header.

(8) RHSIFnLRXFRMRL1 — Last Rx Frame L1 Result Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RL1E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RL1P[7:0]								RL1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.66 RHSIFnLRXFRMRL1 Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	RL1E	This bit indicates last Rx Endbit.
15 to 8	RL1P[7:0]	These bits indicate Last Rx L1 Payload.
7 to 0	RL1H[7:0]	These bits indicate Last Rx L1 Header.

(9) RHSIFnLTXFRMRL2 — Last Tx Frame L2 Result Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	T2L2H[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	T2L1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.67 RHSIFnLTXFRMRL2 Register Contents

Bit Position	Bit Name	Function
31 to 16	T2L2H[15:0]	These bits indicate Last Tx L2 Header of L2 frame.
15 to 8	Reserved	When read, the value after reset is returned.
7 to 0	T2L1H[7:0]	These bits indicate Last Tx L1 Header of L2 frame.

(10) RHSIFnLRXFRMRL2 — Last Rx Frame L2 Result Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + $03C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	R2L2H[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	R2L1H[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.68 RHSIFnLRXFRMRL2 Register Contents

Bit Position	Bit Name	Function
31 to 16	R2L2H[15:0]	These bits indicate Last Rx L2 Header of L2 frame.
15 to 8	Reserved	When read, the value after reset is returned.
7 to 0	R2L1H[7:0]	These bits indicate Last Rx L1 Header of L2 frame.

(11) RHSIFnTXRXCR — TX/RX Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 29.69 RHSIFnTXRXCR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TXEN	This bit specifies whether to enable transmission. This bit is updated when the ICLC command is for "Enable Slave interface transmitter" or "Disable Slave interface transmitter" as well as writing to this bit directly. 0: Disable. 1: Enable.
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RXEN	This bit specifies whether to enable reception. 0: Disable. 1: Enable.

(12) RHSIFnICCR — ICLC Command Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ITRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PLD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.70 RHSIFnICCR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	ITRG	ICLC Command Tx Trigger. This bit is cleared automatically after ICLC command transmission is completed. 0: Disable. 1: Trigger.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	PLD[7:0]*1	ICLC Command (Payload)

Note 1. Do not specify any commands except the commands described in **Table 29.11**.

(13) RHSIFnCCR — CTS Frame Control Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CTS	PLD[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.71 RHSIFnCCR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	CTRG	CTS Frame Tx Trigger. This bit is cleared automatically after CTS Frame transmission is completed. 0: Disable. 1: Trigger.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CTS	CTS Value
7 to 0	PLD[7:0]	CTS Frame (Payload)

(14) RHSIFnTXCMPST — Tx Complete Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 080_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCCT	—	—	—	—	—	—	—	TCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.72 RHSIFnTXCMPST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	TCL2	This bit indicates the status of L2 frame transmission. 0: L2 frame transmission not completed. 1: L2 frame transmission completed.
15 to 9	Reserved	When read, the value after reset is returned.
8	TCCT	This bit indicates the status of L1 CTS frame transmission. 0: L1 CTS frame transmission not completed. 1: L1 CTS frame transmission completed.
7 to 1	Reserved	When read, the value after reset is returned.
0	TCIC	This bit indicates the status of L1 ICLC command transmission. 0: L1 ICLC command transmission not completed. 1: L1 ICLC command transmission completed.

(15) RHSIFnTXCMPSC — Tx Complete Status Clear Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: <RHSIFn_L1_base> + 084_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCCCT	—	—	—	—	—	—	—	TCCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	W

Table 29.73 RHSIFnTXCMPSC Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When writing, write the value after reset.
16	TCCL2	Writing 1 to this bit clears the RHSIFnTXCMPST.TCL2 bit. Writing 0 to this bit is ignored.
15 to 9	Reserved	When writing, write the value after reset.
8	TCCCT	Writing 1 to this bit clears the RHSIFnTXCMPST.TCCT bit. Writing 0 to this bit is ignored.
7 to 1	Reserved	When writing, write the value after reset.
0	TCCIC	Writing 1 to this bit clears the RHSIFnTXCMPST.TCIC bit. Writing 0 to this bit is ignored.

(16) RHSIFnTXCMPIE — Tx Complete Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 088_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCEL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCECT	—	—	—	—	—	—	—	TCEIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 29.74 RHSIFnTXCMPIE Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TCEL2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST. TCL2 bit.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TCECT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST.TCCT bit.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TCEIC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXCMPST.TCIC bit.

(17) RHSIFnTXERRST — Tx Error Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + 090_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERSZ	—	—	—	—	—	—	—	—	—	—	TERSZI5	TERSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERCTF	TERCTE	TERCTD	TERCTC	TERCTB	TERCTA	TERCT9	TERCT8	—	—	—	—	—	TERCT2	TERCT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.75 RHSIFnTXERRST Register Contents (1/2)

Bit Position	Bit Name	Function
31	TERSZ	This bit indicates the status of payload size error of transmission. 0: Payload size error has not occurred. 1: Payload size error occurred.
30 to 21	Reserved	When read, the value after reset is returned.
20	TERSZI5	This bit indicates that the payload size of transmission is 3'b101. 0: Payload size index is not 3'b101. 1: Payload size index is 3'b101.
19	TERSZI4	This bit indicates that the payload size of transmission is 3'b100. 0: Payload size index is not 3'b100. 1: Payload size index is 3'b100.
18 to 16	Reserved	When read, the value after reset is returned.
15	TERCTF	This bit indicates that the logical channel type of transmission is 4'b1111. 0: Logical channel type of transmission is not 4'b1111. 1: Logical channel type of transmission is 4'b1111.
14	TERCTE	This bit indicates that the logical channel type of transmission is 4'b1110. 0: Logical channel type of transmission is not 4'b1110. 1: Logical channel type of transmission is 4'b1110.
13	TERCTD	This bit indicates that the logical channel type of transmission is 4'b1101. 0: Logical channel type of transmission is not 4'b1101. 1: Logical channel type of transmission is 4'b1101.
12	TERCTC	This bit indicates that the logical channel type of transmission is 4'b1100. 0: Logical channel type of transmission is not 4'b1100. 1: Logical channel type of transmission is 4'b1100.
11	TERCTB	This bit indicates that the logical channel type of transmission is 4'b1011. 0: Logical channel type of transmission is not 4'b1011. 1: Logical channel type of transmission is 4'b1011.
10	TERCTA	This bit indicates that the logical channel type of transmission is 4'b1010. 0: Logical channel type of transmission is not 4'b1010. 1: Logical channel type of transmission is 4'b1010.
9	TERCT9	This bit indicates that the logical channel type of transmission is 4'b1001. 0: Logical channel type of transmission is not 4'b1001. 1: Logical channel type of transmission is 4'b1001.
8	TERCT8	This bit indicates that the logical channel type of transmission is 4'b1000. 0: Logical channel type of transmission is not 4'b1000. 1: Logical channel type of transmission is 4'b1000.

Table 29.75 RHSIFnTXERRST Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	TERCT2	This bit indicates that the logical channel type of transmission is 4'b0010. 0: Logical channel type of transmission is not 4'b0010. 1: Logical channel type of transmission is 4'b0010.
1	TERCT1	This bit indicates that the logical channel type of transmission is 4'b0001. 0: Logical channel type of transmission is not 4'b0001. 1: Logical channel type of transmission is 4'b0001.
0	Reserved	When read, the value after reset is returned.

(18) RHSIFnTXERRSC — Tx Error Status Clear Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: <RHSIFn_L1_base> + 094_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERCSZ	—	—	—	—	—	—	—	—	—	—	TERCSZI5	TERCSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERCCTF	TERCCTE	TERCCTD	TERCCTC	TERCCTB	TERCCTA	TERCCT9	TERCCT8	—	—	—	—	—	TERCCT2	TERCCT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	R	R	R	W	W	R

Table 29.76 RHSIFnTXERRSC Register Contents

Bit Position	Bit Name	Function
31	TERCSZ	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZ bit. Writing 0 to this bit is ignored.
30 to 21	Reserved	When writing, write the value after reset.
20	TERCSZI5	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZI5 bit. Writing 0 to this bit is ignored.
19	TERCSZI4	Writing 1 to this bit clears the RHSIFnTXERRST.TERSZI4 bit. Writing 0 to this bit is ignored.
18 to 16	Reserved	When writing, write the value after reset.
15	TERCCTF	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTF bit. Writing 0 to this bit is ignored.
14	TERCCTE	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTE bit. Writing 0 to this bit is ignored.
13	TERCCTD	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTD bit. Writing 0 to this bit is ignored.
12	TERCCTC	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTC bit. Writing 0 to this bit is ignored.
11	TERCCTB	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTB bit. Writing 0 to this bit is ignored.
10	TERCCTA	Writing 1 to this bit clears the RHSIFnTXERRST.TERCTA bit. Writing 0 to this bit is ignored.
9	TERCCT9	Writing 1 to this bit clears the RHSIFnTXERRST.TERCT9 bit. Writing 0 to this bit is ignored.
8	TERCCT8	Writing 1 to this bit clears the RHSIFnTXERRST.TERCT8 bit. Writing 0 to this bit is ignored.
7 to 3	Reserved	When writing, write the value after reset.
2	TERCCT2	Writing 1 to this bit clears the RHSIFnTXERRST.TERCT2 bit. Writing 0 to this bit is ignored.
1	TERCCT1	Writing 1 to this bit clears the RHSIFnTXERRST.TERCT1 bit. Writing 0 to this bit is ignored.
0	Reserved	When writing, write the value after reset.

(19) RHSIFnTXERRIE — Tx Error Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + 098_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TERESZ	—	—	—	—	—	—	—	—	—	—	TERESZI5	TERESZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TERECTF	TERECTE	TERECTD	TERECTC	TERECTB	TERECTA	TERECT9	TERECT8	—	—	—	—	—	TERECT2	TERECT1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R

Table 29.77 RHSIFnTXERRIE Register Contents (1/2)

Bit Position	Bit Name	Function
31	TERESZ	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZ bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZ bit is 1.
30 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	TERESZI5	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZI5 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZI5 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZI5 bit is 1.
19	TERESZI4	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERSZI4 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZI4 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERSZI4 bit is 1.
18 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	TERECTF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTF bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTF bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTF bit is 1.
14	TERECTE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTE bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTE bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTE bit is 1.
13	TERECTD	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTD bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTD bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTD bit is 1.
12	TERECTC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTC bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTC bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTC bit is 1.
11	TERECTB	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTB bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTB bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTB bit is 1.

Table 29.77 RHSIFnTXERRIE Register Contents (2/2)

Bit Position	Bit Name	Function
10	TERECTA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCTA bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTA bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCTA bit is 1.
9	TERECT9	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT9 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT9 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT9 bit is 1.
8	TERECT8	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT8 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT8 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT8 bit is 1.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TERECT2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT2 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT2 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT2 bit is 1.
1	TERECT1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnTXERRST.TERCT1 bit. 0: Does not assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT1 bit is 1. 1: Assert RHSIFnTXERR when the RHSIFnTXERRST.TERCT1 bit is 1.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(20) RHSIFnRXCMPST — Rx Complete Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + $0A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCCT	—	—	—	—	—	—	—	RCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.78 RHSIFnRXCMPST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	RCL2	This bit indicates the status of L2 frame reception. 0: L2 frame reception not completed. 1: L2 frame reception completed.
15 to 9	Reserved	When read, the value after reset is returned.
8	RCCT	This bit indicates the status of L1 CTS frame reception. 0: L1 CTS frame reception not completed. 1: L1 CTS frame reception completed.
7 to 1	Reserved	When read, the value after reset is returned.
0	RCIC	This bit indicates the status of L1 ICLC command reception. 0: L1 ICLC command reception not completed. 1: L1 ICLC command reception completed.

(21) RHSIFnRXCMPSC — Rx Complete Status Clear Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: <RHSIFn_L1_base> + $0A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCCL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCCCT	—	—	—	—	—	—	—	RCCIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	W

Table 29.79 RHSIFnRXCMPSC Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When writing, write the value after reset.
16	RCCL2	Writing 1 to this bit clears the RHSIFnRXCMPST.RCL2 bit. Writing 0 to this bit is ignored.
15 to 9	Reserved	When writing, write the value after reset.
8	RCCCT	Writing 1 to this bit clears the RHSIFnRXCMPST.RCCT bit. Writing 0 to this bit is ignored.
7 to 1	Reserved	When writing, write the value after reset.
0	RCCIC	Writing 1 to this bit clears the RHSIFnRXCMPST.RCIC bit. Writing 0 to this bit is ignored.

(22) RHSIFnRXCMPIE — Rx Complete Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + $0A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCEL2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RCECT	—	—	—	—	—	—	—	RCEIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 29.80 RHSIFnRXCMPIE Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RCEL2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCL2 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCL2 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCL2 bit is 1.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	RCECT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCCT bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCCT bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCCT bit is 1.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RCEIC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXCMPST.RCIC bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXCMPST.RCIC bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXCMPST.RCIC bit is 1.

(23) RHSIFnRXERRST — Rx Error Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + $0B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERSZ	—	—	—	—	—	—	—	—	—	—	RERSZI5	RERSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RERCTF	RERCTE	RERCTD	RERCTC	RERCTB	RERCTA	RERCT9	RERCT8	—	—	—	—	—	RERCT2	RERCT1	RERIPV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.81 RHSIFnRXERRST Register Contents (1/2)

Bit Position	Bit Name	Function
31	RERSZ	This bit indicates that the status of payload size error of reception. 0: Payload size error not occurred. 1: Payload size error occurred.
30 to 21	Reserved	When read, the value after reset is returned.
20	RERSZI5	This bit indicates that the payload size of reception is 101_B. 0: Payload size index is not 101_B. 1: Payload size index is 101_B.
19	RERSZI4	This bit indicates that the payload size of reception is 100_B. 0: Payload size index is not 100_B. 1: Payload size index is 100_B.
18 to 16	Reserved	When read, the value after reset is returned.
15	RERCTF	This bit indicates that the logical channel type of reception is 1111_B. 0: Logical channel type of reception is not 1111_B. 1: Logical channel type of reception is 1111_B.
14	RERCTE	This bit indicates that the logical channel type of reception is 1110_B. 0: Logical channel type of reception is not 1110_B. 1: Logical channel type of reception is 1110_B.
13	RERCTD	This bit indicates that the logical channel type of reception is 1101_B. 0: Logical channel type of reception is not 1101_B. 1: Logical channel type of reception is 1101_B.
12	RERCTC	This bit indicates that the logical channel type of reception is 1100_B. 0: Logical channel type of reception is not 1100_B. 1: Logical channel type of reception is 1100_B.
11	RERCTB	This bit indicates that the logical channel type of reception is 1011_B. 0: Logical channel type of reception is not 1011_B. 1: Logical channel type of reception is 1011_B.
10	RERCTA	This bit indicates that the logical channel type of reception is 1010_B. 0: Logical channel type of reception is not 1010_B. 1: Logical channel type of reception is 1010_B.
9	RERCT9	This bit indicates that the logical channel type of reception is 1001_B. 0: Logical channel type of reception is not 1001_B. 1: Logical channel type of reception is 1001_B.
8	RERCT8	This bit indicates that the logical channel type of reception is 1000_B. 0: Logical channel type of reception is not 1000_B. 1: Logical channel type of reception is 1000_B.

Table 29.81 RHSIFnRXERRST Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	RERCT2	This bit indicates that the logical channel type of reception is 0010_B. 0: Logical channel type of reception is not 0010_B. 1: Logical channel type of reception is 0010_B.
1	RERCT1	This bit indicates that the logical channel type of reception is 0001_B. 0: Logical channel type of reception is not 0001_B. 1: Logical channel type of reception is 0001_B.
0	RERIPV	This bit indicates whether or not the command undefined in **Table 29.11** is received. 0: Command defined in the table is received. 1: Command undefined in the table is received.

(24) RHSIFnRXERRSC — Rx Error Status Clear Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: <RHSIFn_L1_base> + $0B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERCSZ	—	—	—	—	—	—	—	—	—	—	RERCSZI5	RERCSZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RERCCTF	RERCCTE	RERCCTD	RERCCTC	RERCCTB	RERCCTA	RERCCT9	RERCCT8	—	—	—	—	—	RERCCT2	RERCCT1	RERCIPV
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	R	R	R	W	W	W

Table 29.82 RHSIFnRXERRSC Register Contents

Bit Position	Bit Name	Function
31	RERCSZ	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZ bit. Writing 0 to this bit is ignored.
30 to 21	Reserved	When writing, write the value after reset.
20	RERCSZI5	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZI5 bit. Writing 0 to this bit is ignored.
19	RERCSZI4	Writing 1 to this bit clears the RHSIFnRXERRST.RERSZI4 bit. Writing 0 to this bit is ignored.
18 to 16	Reserved	When writing, write the value after reset.
15	RERCCTF	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTF bit. Writing 0 to this bit is ignored.
14	RERCCTE	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTE bit. Writing 0 to this bit is ignored.
13	RERCCTD	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTD bit. Writing 0 to this bit is ignored.
12	RERCCTC	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTC bit. Writing 0 to this bit is ignored.
11	RERCCTB	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTB bit. Writing 0 to this bit is ignored.
10	RERCCTA	Writing 1 to this bit clears the RHSIFnRXERRST.RERCTA bit. Writing 0 to this bit is ignored.
9	RERCCT9	Writing 1 to this bit clears the RHSIFnRXERRST.RERCT9 bit. Writing 0 to this bit is ignored.
8	RERCCT8	Writing 1 to this bit clears the RHSIFnRXERRST.RERCT8 bit. Writing 0 to this bit is ignored.
7 to 3	Reserved	When writing, write the value after reset.
2	RERCCT2	Writing 1 to this bit clears the RHSIFnRXERRST.RERCT2 bit. Writing 0 to this bit is ignored.
1	RERCCT1	Writing 1 to this bit clears the RHSIFnRXERRST.RERCT1 bit. Writing 0 to this bit is ignored.
0	RERCIPV	Writing 1 to this bit clears the RHSIFnRXERRST.RERIPV bit. Writing 0 to this bit is ignored.

(25) RHSIFnRXERRIE — Rx Error Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + $0B8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RERESZ	—	—	—	—	—	—	—	—	—	—	RERESZI5	RERESZI4	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REREC TF	REREC TE	REREC TD	REREC TC	REREC TB	REREC TA	REREC T9	REREC T8	—	—	—	—	—	REREC T2	REREC T1	REREIP V
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 29.83 RHSIFnRXERRIE Register Contents (1/2)

Bit Position	Bit Name	Function
31	RERESZ	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZ bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZ bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZ bit is 1.
30 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	RERESZI5	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZI5 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI5 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI5 bit is 1.
19	RERESZI4	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERSZI4 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI4 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERSZI4 bit is 1.
18 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	RERECTF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTF bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTF bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTF bit is 1.
14	RERECTE	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTE bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTE bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTE bit is 1.
13	RERECTD	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTD bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTD bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTD bit is 1.

Table 29.83 RHSIFnRXERRIE Register Contents (2/2)

Bit Position	Bit Name	Function
12	RERECTC	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTC bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTC bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTC bit is 1.
11	RERECTB	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTB bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTB bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTB bit is 1.
10	RERECTA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCTA bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTA bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCTA bit is 1.
9	RERECT9	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCT9 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT9 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT9 bit is 1.
8	RERECT8	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCT8 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT8 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT8 bit is 1.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	RERECT2	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCT2 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT2 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT2 bit is 1.
1	RERECT1	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERCT1 bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT1 bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERCT1 bit is 1.
0	REREIPV	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXERRST.RERIPV bit. 0: Does not assert RHSIFnRXERR when the RHSIFnRXERRST.RERIPV bit is 1. 1: Assert RHSIFnRXERR when the RHSIFnRXERRST.RERIPV bit is 1.

(26) RHSIFnRXICST — Rx ICLC Command Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RHSIFn_L1_base> + $0C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RITOL	RITOF	RITON	RIDT	RIET	RIFT	RIST	RIFR	RISR	RIPSTP	RIPSRT	RIPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.84 RHSIFnRXICST Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	RIPA	This bit indicates the status of ICLC ping answer. 0: Ping answer not received. 1: Ping answer received.
15 to 12	Reserved	When read, the value after reset is returned.
11	RITOL	This bit indicates the status of ICLC turn on payload loopback. 0: Turn on payload loopback not received. 1: Turn on payload loopback received.
10	RITOF	This bit indicates the status of ICLC turn off payload loopback. 0: Turn off payload loopback not received. 1: Turn off payload loopback received.
9	RITON	This bit indicates the status of ICLC turn on test mode. 0: Turn on test mode not received. 1: Turn on test mode received.
8	RIDT	This bit indicates the status of ICLC disable Tx port. 0: Disable Tx port not received. 1: Disable Tx port received.
7	RIET	This bit indicates the status of ICLC enable Tx port. 0: Enable Tx port not received. 1: Enable Tx port received.
6	RIFT	This bit indicates the status of ICLC select FastMode Tx port. 0: Select FastMode Tx port not received. 1: Select FastMode Tx port received.
5	RIST	This bit indicates the status of ICLC select SlowMode Tx port. 0: Select SlowMode Tx port not received. 1: Select SlowMode Tx port received.
4	RIFR	This bit indicates the status of ICLC select FastMode Rx port. 0: Select FastMode Rx port not received. 1: Select FastMode Rx port received.
3	RISR	This bit indicates the status of ICLC select SlowMode Rx port. 0: Select SlowMode Rx port not received. 1: Select SlowMode Rx port received.
2	RIPSTP	This bit indicates the status of ICLC PLL Stop. 0: PLL Stop not received. 1: PLL Stop received.

Table 29.84 RHSIFnRXICST Register Contents (2/2)

Bit Position	Bit Name	Function
1	RIPSRT	This bit indicates the status of ICLC PLL Start. 0: PLL Start not received. 1: PLL Start received.
0	RIPG	This bit indicates the status of ICLC Ping. 0: Ping not received. 1: Ping received.

(27) RHSIFnRXICSC — Rx ICLC Command Status Clear Register

Access: This register is a write-only register that can be write in 32-bit units.

Address: <RHSIFn_L1_base> + $0C4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RICPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RICTOL	RICTOF	RICTON	RICDT	RICET	RICFT	RICST	RICFR	RICSR	RICPSTP	RICPSRT	RICPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W	W

Table 29.85 RHSIFnRXICSC Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When writing, write the value after reset.
16	RICPA	Writing 1 to this bit clears the RHSIFnRXICST.RIPA bit. Writing 0 to this bit is ignored.
15 to 12	Reserved	When writing, write the value after reset.
11	RICTOL	Writing 1 to this bit clears the RHSIFnRXICST.RITOL bit. Writing 0 to this bit is ignored.
10	RICTOF	Writing 1 to this bit clears the RHSIFnRXICST.RITOF bit. Writing 0 to this bit is ignored.
9	RICTON	Writing 1 to this bit clears the RHSIFnRXICST.RITON bit. Writing 0 to this bit is ignored.
8	RICDT	Writing 1 to this bit clears the RHSIFnRXICST.RIDT bit. Writing 0 to this bit is ignored.
7	RICET	Writing 1 to this bit clears the RHSIFnRXICST.RIET bit. Writing 0 to this bit is ignored.
6	RICFT	Writing 1 to this bit clears the RHSIFnRXICST.RIRF bit. Writing 0 to this bit is ignored.
5	RICST	Writing 1 to this bit clears the RHSIFnRXICST.RIRS bit. Writing 0 to this bit is ignored.
4	RICFR	Writing 1 to this bit clears the RHSIFnRXICST.RITF bit. Writing 0 to this bit is ignored.
3	RICSR	Writing 1 to this bit clears the RHSIFnRXICST.RITS bit. Writing 0 to this bit is ignored.
2	RICPSTP	Writing 1 to this bit clears the RHSIFnRXICST.RIPSTP bit. Writing 0 to this bit is ignored.
1	RICPSRT	Writing 1 to this bit clears the RHSIFnRXICST.RIPSRT bit. Writing 0 to this bit is ignored.
0	RICPG	Writing 1 to this bit clears the RHSIFnRXICST.RIPG bit. Writing 0 to this bit is ignored.

(28) RHSIFnRXICIE — Rx ICLC Command Interrupt Enable Register

Access: This register can be read or written in 32-bit units.

Address: <RHSIFn_L1_base> + $0C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIEPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RIETOL	RIETOF	RIETON	RIEDT	RIEET	RIEFT	RIEST	RIEFR	RIESR	RIEPSTP	RIEPSRT	RIEPG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.86 RHSIFnRXICIE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	RIEPA	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPA bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPA bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPA bit is 1.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	RIETOL	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITOL bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITOL bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITOL bit is 1.
10	RIETOF	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITOF bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITOF bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITOF bit is 1.
9	RIETON	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RITON bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RITON bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RITON bit is 1.
8	RIEDT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIDT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIDT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIDT bit is 1.
7	RIEET	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIET bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIET bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIET bit is 1.
6	RIEFT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIFT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIFT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIRF bit is 1.
5	RIEST	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIST bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIST bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIST bit is 1.

Table 29.86 RHSIFnRXICIE Register Contents (2/2)

Bit Position	Bit Name	Function
4	RIEFR	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIFR bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIFR bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIFR bit is 1.
3	RIESR	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RISR bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RISR bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RISR bit is 1.
2	RIEPSTP	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPSTP bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSTP bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSTP bit is 1.
1	RIEPSRT	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPSRT bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSRT bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPSRT bit is 1.
0	RIEPG	This bit specifies whether to enable the interrupt that is triggered by the RHSIFnRXICST.RIPG bit. 0: Does not assert RHSIFnRXICLC when the RHSIFnRXICST.RIPG bit is 1. 1: Assert RHSIFnRXICLC when the RHSIFnRXICST.RIPG bit is 1.

29.5.3 Operation

29.5.3.1 Initial Settings

(1) Clocks generated by CLKGEN and transfer mode

The CLKGEN generates the clock for data transmission / reception. **Table 29.87** below shows the correspondence between the clocks to be generated and the transfer mode. The switching between master mode and slave mode, the transfer mode, and the baud rate for fast transfer mode are specified by setting the registers.

During operation in slave mode, the transfer mode can be switched between fast and slow by the ICLC command transmitted from the master.

Table 29.87 Clocks Generated by CLKGEN

Data Transfer Modes	Operation Modes	Clock Sources	Sampling Clock Frequency	Baud Rate
Fast Speed mode	Master / slave	Internal PLL	320, 160, 80 MHz	320, 160, 80 M Baud
Slow Speed mode	Master	System clock	20 MHz	5 M Baud
	Slave	External clock	10, 20 MHz	

(2) L1 settings

(1) Master / slave setting
Specify the master or slave operation by setting the RHSIFnMDCR.MST register. When the master operation is specified, also specify the frequency of the clock to be output to the communication device by setting the RHSIFnMDCR.CLKSEL.

(2) Flow control setting
Specify the automatic flow control setting by setting the RHSIFnMDCR.CTSEN register.

(3) RxD enable setting
Enable or disable RxD reception by setting the RHSIFnTXRXCR.RXEN register.

(4) TxD enable setting
Enable or disable TxD transmission by setting the RHSIFnTXRXCR.TXEN.

(5) Interrupt enable setting
Enable or disable interrupts by setting the RHSIFn*IE registers for L2 and L1, respectively.

(6) Transmission / reception speed setting for fast transfer mode
Specify the transmission / reception speed (baud rate) by setting the RHSIFnSPCR.FMBR register.

(7) Transfer mode setting for transmission / reception
Specify the transfer speed mode for transmission and reception by setting the RHSIFnSPCR.TXSP and RHSIFnSPCR.RXSP registers, respectively.

(8) Procedure for releasing CLKGEN standby
After power-on reset release, the PLL in the CLKGEN is in the standby (OFF) state. It is necessary to release the PLL from the standby state (ON) to perform data transmission / reception in fast transfer mode. When switching from PLL OFF to PLL ON, be sure to release the standby state of the PLL by using the procedures of steps (a) to (e) described below.
If data transmission/reception is performed in fast transfer mode without following the procedure, the baud rate is unstable and normal operation may not be achieved.

(a) Insert waits by software so that the external reference clock (RHSIF0_REFCLK) becomes stable 1 µs before the PLL is released from the standby state (indicated by step (d) below). During master operation, however, no wait is required because the internal clock is used as the reference clock for PLL at this time.

(b) Set the following registers as required.
RHSIFnMDCR.MST
RHSIFnMDCR.CLKSEL
RHSIFnSPCR.FMBR
RHSIFnSPCR.TXSP
RHSIFnSPCR.RXSP
Be sure to update these registers when the PLL is in the standby state.

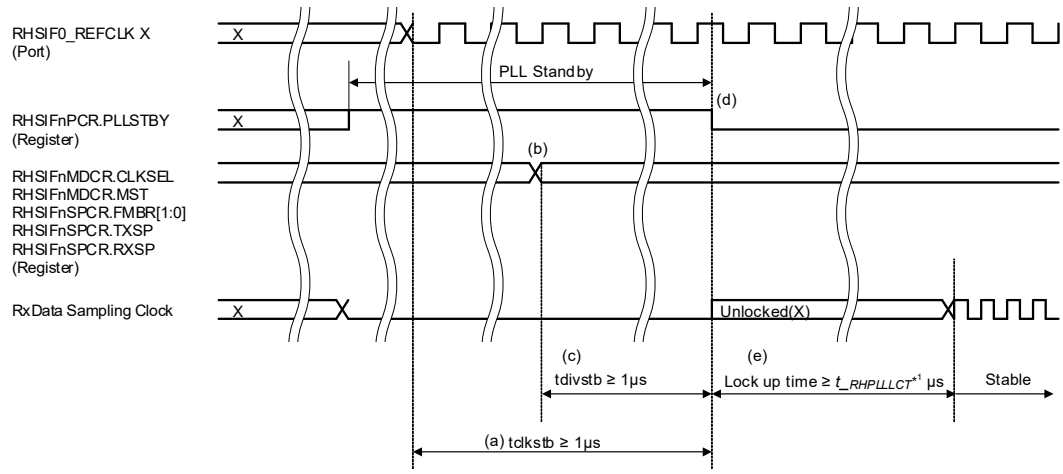
(c) Insert waits for at least 1 µs by software.

(d) Clear the RHSIFnPCR.PLLSTBY bit to “0”. The PLL released from the standby starts running.

(e) Insert waits for at least tRHPLLLCT µs by software until the PLL locks up. For details of PLL locked time, refer to **Section 55, Electrical Characteristics**.

Note: When switching from PLL ON to PLL OFF, set the RHSIFnPCR.PLLSTBY bit to “1” first to set the PLL in standby state, and then change the registers described in step (b) above. If setting values of these registers are changed while the PLL is running, the PLL may not operate normally.

Figure 29.30 below shows a timing chart for the procedure of steps (a) to (e) above.



Note 1. For details of PLL locked time, refer to **Section 55, Electrical Characteristics**.

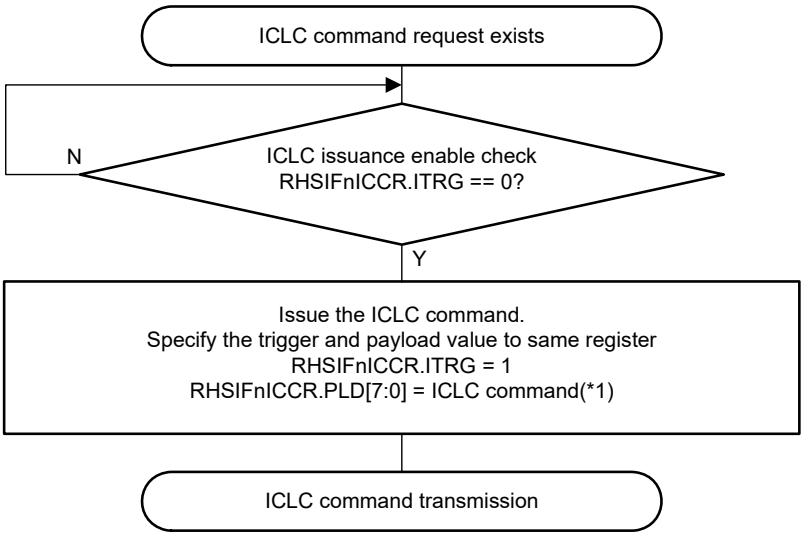
Figure 29.30 Timing Chart for Releasing CLKGEN from Reset State

29.5.3.2 ICLC Command Control Method

(1) Basic operation: ICLC command issuance

An ICLC command can be issued from the master to a slave. An ICLC command request is transmitted by firmware. Transmit a request by accessing the register and check the response by using an interrupt as trigger.

Figure 29.31 below shows the procedure flow for issuing an ICLC command.



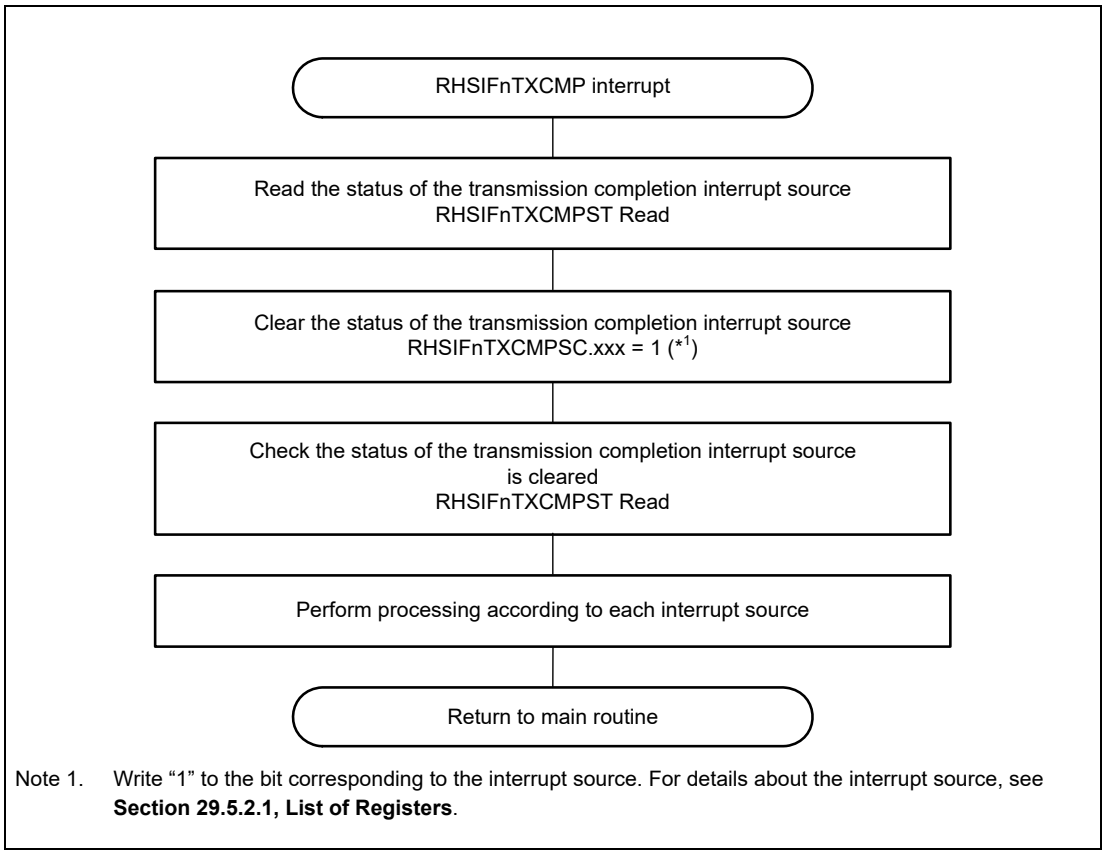
Note 1. See **Table 29.11** for details about the payload values for the ICLC command.

Figure 29.31 Procedure Flow for Issuing ICLC Command

For details about the ICLC commands supported by this product, see **Table 29.11 List of ICLC Commands**.

(2) Basic Operation of Interrupt

Figure 29.32 below shows the procedure flow for processing an interrupt, using transmission completion interrupt as example.



Note 1. Write “1” to the bit corresponding to the interrupt source. For details about the interrupt source, see **Section 29.5.2.1, List of Registers**.

Figure 29.32 Procedure Flow for Processing Transmission Completion Interrupt

The processing is similar to the other interrupts. See **Table 29.88** below for the status registers, status clear registers and interrupt enable registers for the other interrupts.

Table 29.88 List of Interrupt Related Registers

Interrupt	Status Register	Status Clear Register	Interrupt Enable Register
RHSIFnTXCMP	RHSIFnTXCMPST	RHSIFnTXCMPSC	RHSIFnTXCMPIE
RHSIFnTXERR	RHSIFnTXERRST	RHSIFnTXERRSC	RHSIFnTXERRIE
RHSIFnRXCMP	RHSIFnRXCMPST	RHSIFnRXCMPSC	RHSIFnRXCMPIE
RHSIFnRXERR	RHSIFnRXERRST	RHSIFnRXERRSC	RHSIFnRXERRIE
RHSIFnRXICLC	RHSIFnRXICST	RHSIFnRXICSC	RHSIFnRXICIE

(3) Basic operation: Execution of ICLC “PING” command

Figure 29.33 below shows a flow for transmitting an ICLC “PING” command and receiving an ICLC “PING Answer”. The ICLC “PING Answer” can automatically be transmitted without firmware.

After completion of transmission, the device waits for a response from the link partner. The following five cases can be considered according to the responding state of the link partner and whether there is a frame error.

(a) When the device operates as the master and the link partner sends back an ICLC “PING Answer” (normal case)
When the device completes transmission of a PING command, a transmission completion interrupt is generated.
Wait until the device receives the ICLC “PING Answer”.
When the device receives the “PING Answer”, a reception completion interrupt and an ICLC reception interrupt are generated.

(b) Transmission errors when the device operates as the master
If an error occurs after the device transmits an ICLC “PING” command, a transmission error interrupt is generated.

(c) Reception errors when the device operates as the master
If an error occurs after the device receives the ICLC “PING Answer”, a reception error interrupt is generated.

(d) Reception errors when the device operates as a slave
If an error occurs after the device receives the ICLC “PING” command, a reception error interrupt is generated.

(e) Transmission errors when the device operates as a slave
If an error occurs after the device transmits an ICLC “PING Answer”, a transmission error interrupt is generated.

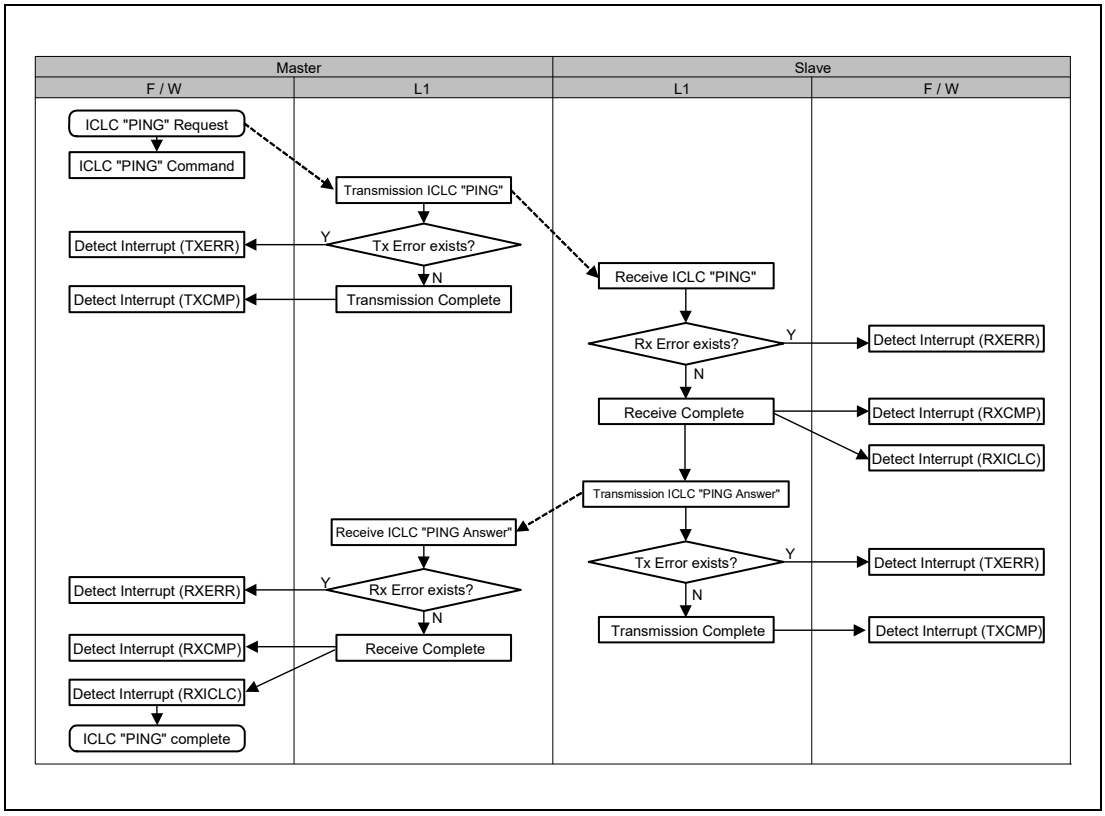


Figure 29.33 Flow for ICLC “PING” Command Execution

For the ICLC commands other than “PING”, the processing is the same except the slave does not send back an ICLC “PING Answer”.

29.5.3.3 Changing Communication Speed Using ICLC Commands

(1) Changing from slow transfer mode to fast transfer mode

Figure 29.34 to **Figure 29.40** show the procedure for changing the communication speed (transfer mode) from slow transfer mode to fast transfer mode.

For fast transfer mode, the transfer rate (baud rate) can be selected from 320, 160 and 80 M Baud. Be sure to specify the transfer rate in advance, by setting the RHSIFnSPCR register.

Also, be sure to enable transmission for the master.

(a) Enable transmission for the RX link slave.
Transmit the ICLC “Enable Slave interface transmitter” command.

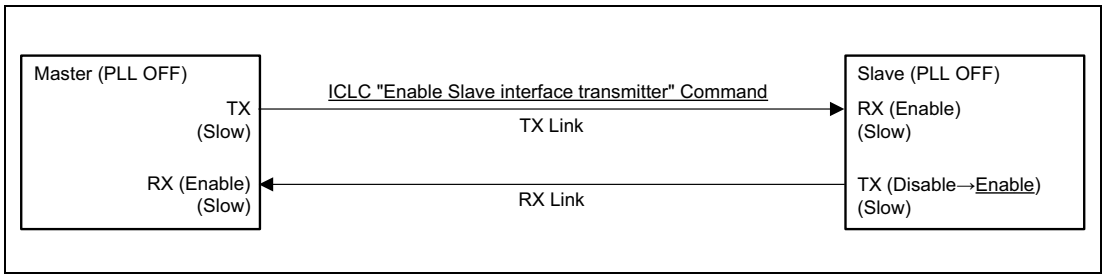
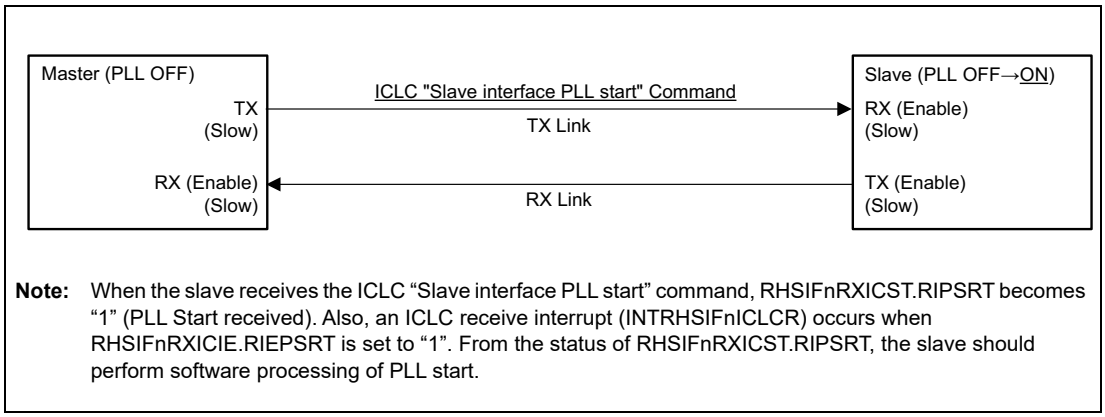


Figure 29.34 Enabling Transmission for RX Link Slave

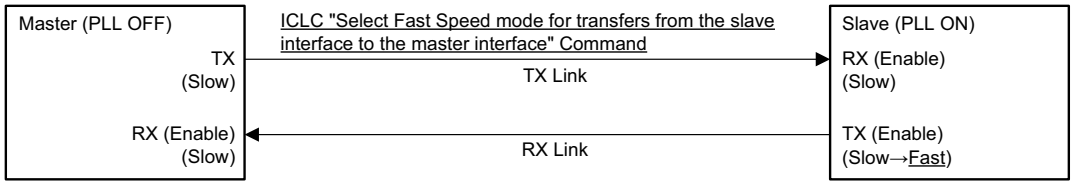
(b) Start operation of the PLL in slave
Transmit the ICLC “Slave interface PLL start” command. Refer to **(8), Procedure for releasing CLKGEN standby** in **Section 29.5.3.1(2), L1 settings** for details.



Note: When the slave receives the ICLC “Slave interface PLL start” command, RHSIFnRXICST.RIPSRT becomes “1” (PLL Start received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIEPSRT is set to “1”. From the status of RHSIFnRXICST.RIPSRT, the slave should perform software processing of PLL start.

Figure 29.35 Starting Operation of PLL in Slave

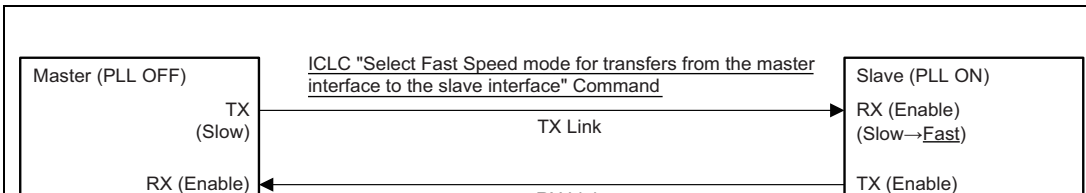
(c) Set the transmission speed mode for Rx link slave to fast mode
Transmit the ICLC “Select Fast Speed mode for transfers from the slave interface to the master interface” command.



Note: When the slave receives the ICLC “Select Fast Speed mode for transfers from the Slave interface to the Master interface” command, RHSIFnRXICST.RIFT becomes "1" (Select FastMode Tx port received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIEFT is set to "1". From the status of RHSIFnRXICST.RIFT, the slave should perform software processing of fast speed mode for data transmission.

Figure 29.36 Setting Transmission Speed Mode for Slave to Fast

(d) Set the reception speed mode for Tx link slave to fast mode
Transmit the ICLC “Select Fast Speed mode for transfers from the Master interface to the Slave interface” command.



Note: When the slave receives the ICLC “Select Fast Speed mode for transfers from the Master interface to the Slave interface” command, RHSIFnRXICST.RIFR becomes “1” (Select FastMode Rx port received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIEFR is set to “1”. From the status of RHSIFnRXICST.RIFR, the slave should perform software processing of fast speed mode for data reception.

Figure 29.37 Setting Reception Speed for Slave to Fast

(e) Start operation of the PLL in master
Release the PLL from the standby state and start operation of the PLL by setting the APB registers. Refer to **(8), Procedure for releasing CLKGEN standby**, in **Section 29.5.3.1(2), L1 settings** for details.

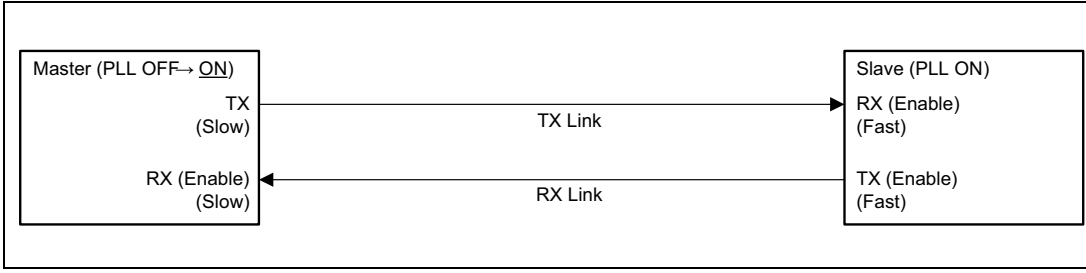


Figure 29.38 Starting Operation of PLL in Master

(f) Set the reception speed mode for Rx link and the transmission speed mode for Tx link of master to fast mode
Set the Rx and Tx speed modes to fast mode by setting RHSIFnSPCR.TXSP to “1” and RHSIFnSPCR.RXSP to “1”.

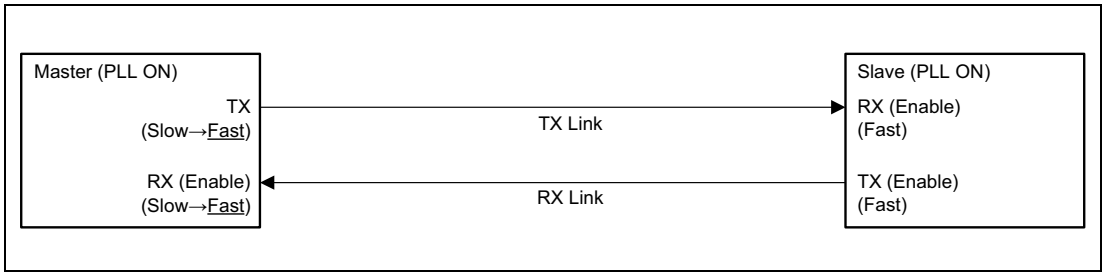


Figure 29.39 Setting Rx and Tx Speed Modes for Master to Fast

(g) Check the link establishment
Transmit the ICLC “PING” command and make sure that a “PING ANSWER” is sent back.

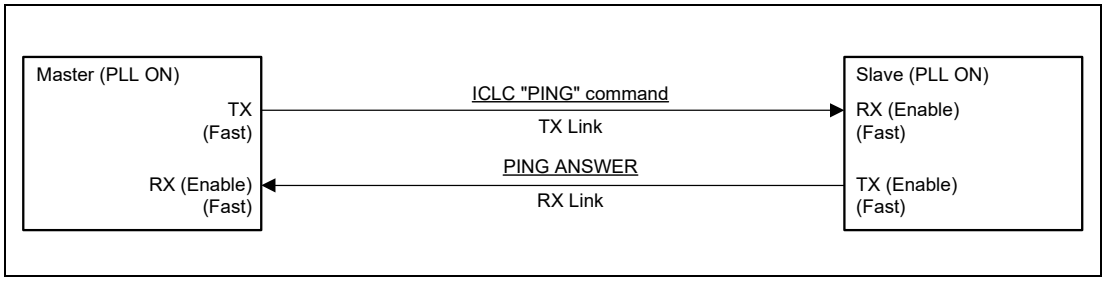


Figure 29.40 Checking Link Establishment

(2) Changing from fast transfer mode to slow transfer mode

Figure 29.41 to **Figure 29.47** show the procedure for changing the communication speed (transfer mode) from fast transfer mode to slow transfer mode.

Be sure to enable transmission for the master in advance

(a) Enable transmission for the RX link slave
Transmit the ICLC “Enable Slave interface transmitter” from the master to slave.

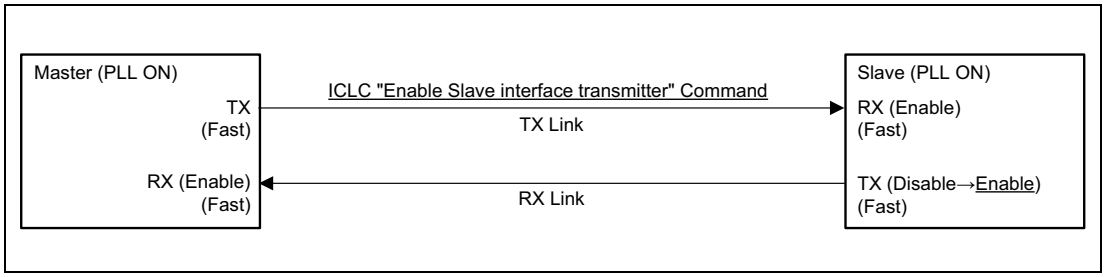
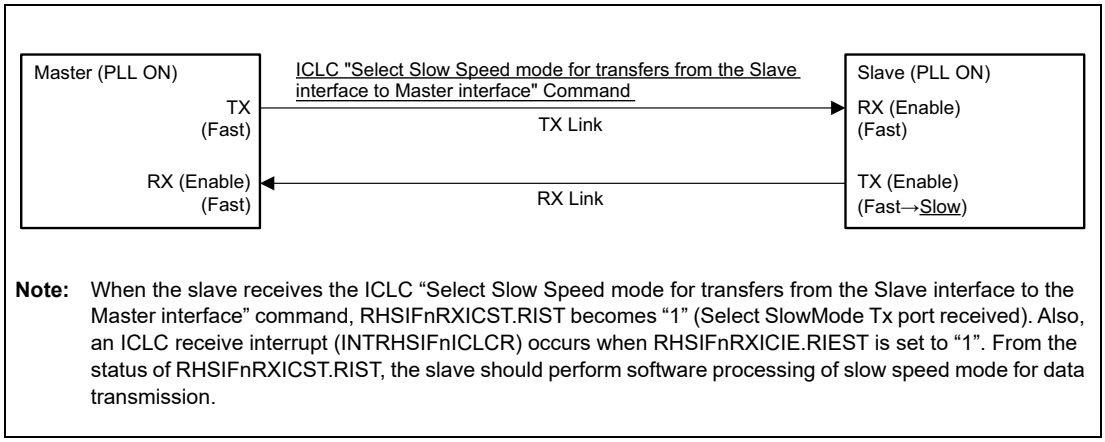


Figure 29.41 Enabling Transmission for RX Link Slave

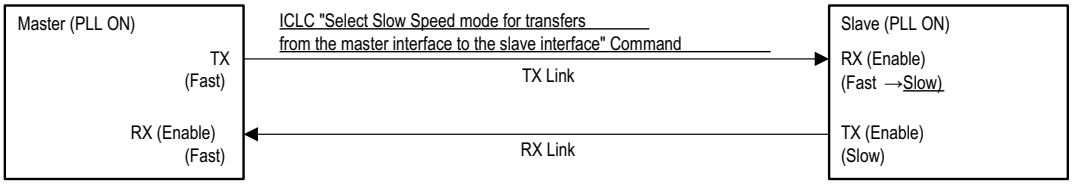
(b) Set the transmission speed mode for slave to slow mode
Transmit the ICLC “Select Slow Speed mode for transfers from the Slave interface to Master interface” command from the master to slave.



Note: When the slave receives the ICLC “Select Slow Speed mode for transfers from the Slave interface to the Master interface” command, RHSIFnRXICST.RIST becomes “1” (Select SlowMode Tx port received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIEST is set to “1”. From the status of RHSIFnRXICST.RIST, the slave should perform software processing of slow speed mode for data transmission.

Figure 29.42 Setting Transmission Speed Mode for Slave to Slow

(c) Set the reception speed mode for slave to slow mode
Transmit the ICLC “Select Slow Speed mode for transfers from the Master interface to the Slave interface” command from the master to slave.



Note: When the slave receives the ICLC “Select Slow Speed mode for transfers from the Master interface to the Slave interface” command, RHSIFnRXICST.RISR becomes "1" (Select SlowMode Rx port received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIESR is set to “1”. From the status of RHSIFnRXICST.RISR, the slave should perform software processing of slow speed mode for data reception.

Figure 29.43 Setting Reception Speed for Slave to Slow

(d) Set the reception and transmission speed modes for master to slow mode
Set the Rx and Tx speed modes to slow by setting RHSIFnSPCR.RXSP to “0” and RHSIFnSPCR.TXSP to “0”.

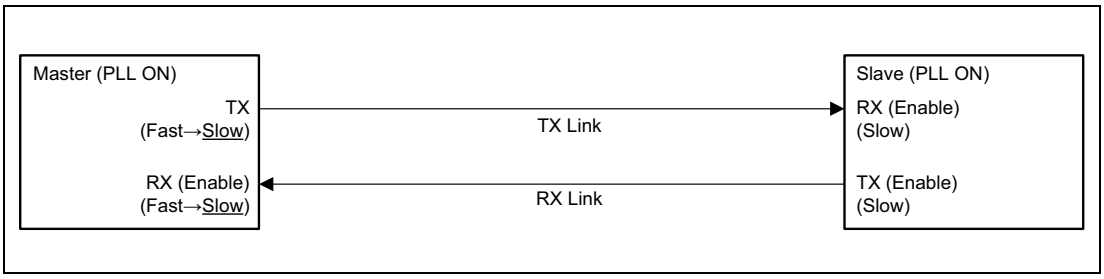
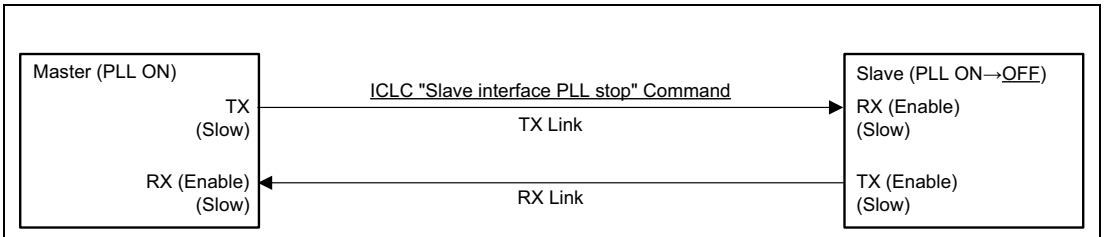


Figure 29.44 Setting Rx and Tx Speed Modes for Master to Slow

(e) Stop the operation of the PLL in slave
Transmit the ICLC “Slave interface PLL stop” command.



Note: When the slave receives the ICLC “Slave interface PLL stop” command, RHSIFnRXICST.RIPSTP becomes "1" (PLL Stop received). Also, an ICLC receive interrupt (INTRHSIFnICLCR) occurs when RHSIFnRXICIE.RIEPSTP is set to “1”. From the status of RHSIFnRXICST.RIPSTP, the slave should perform software processing of PLL stop.

Figure 29.45 Stopping Operation of PLL in Slave

(f) Stop the operation of the PLL in master
Set RHSIFnPCR.PLLSTBY to “1” to stop the PLL operation in master.

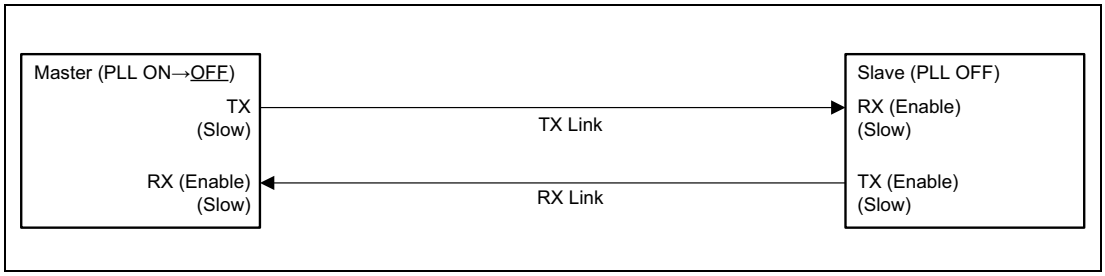


Figure 29.46 Stopping PLL Operation in Master

(g) Check the link establishment
Transmit the ICLC “PING” command from the master to slave, and make sure that a “PING ANSWER” is sent back.

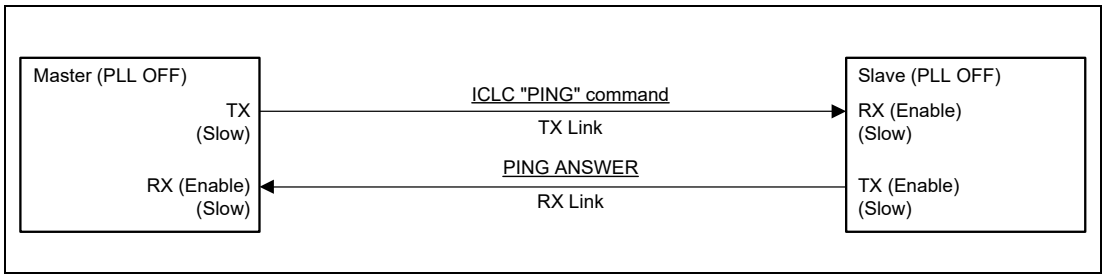


Figure 29.47 Checking Link Establishment

29.5.3.4 Transiting Slave State Using ICLC Commands

When the device is operating as a slave, the state transits according to the ICLC command that is received as follows.

(1) Transmission enabled / disabled state

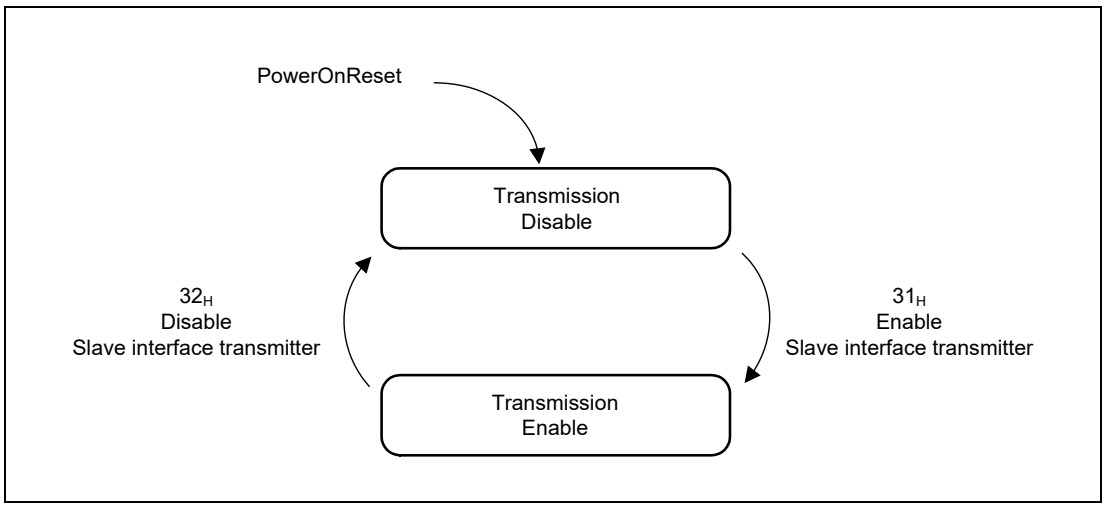
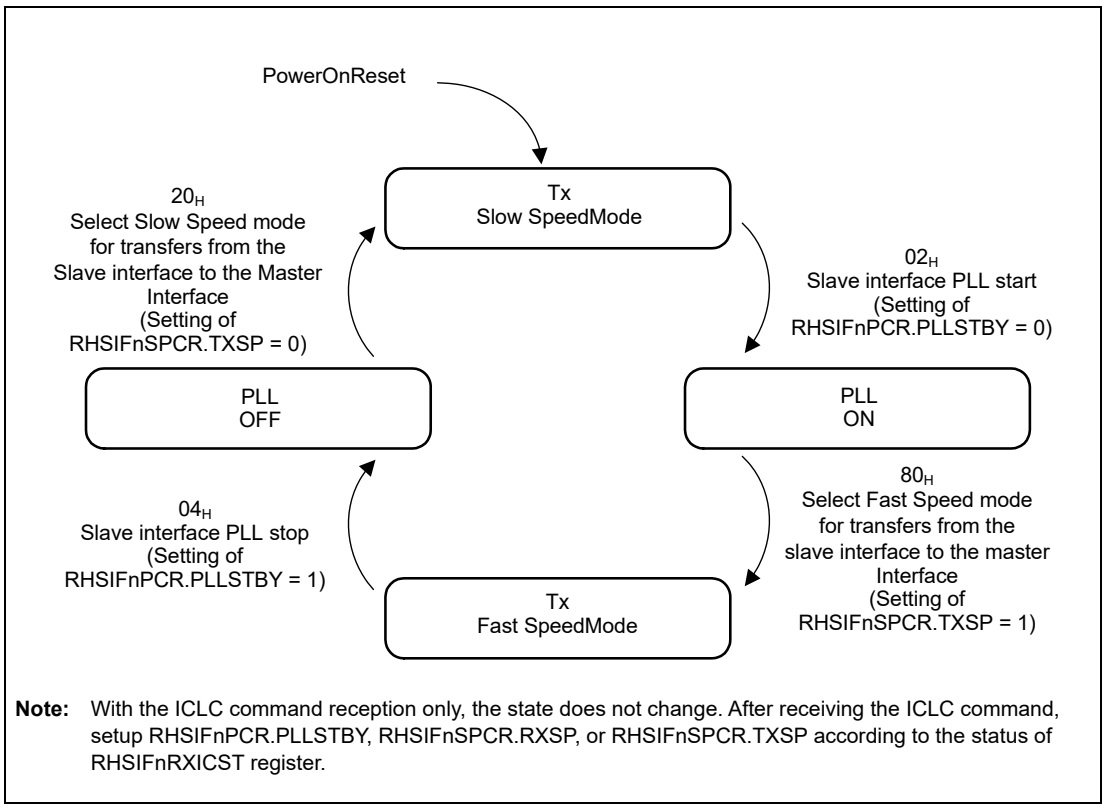


Figure 29.48 Transition between Transmission Enabled and Disabled States

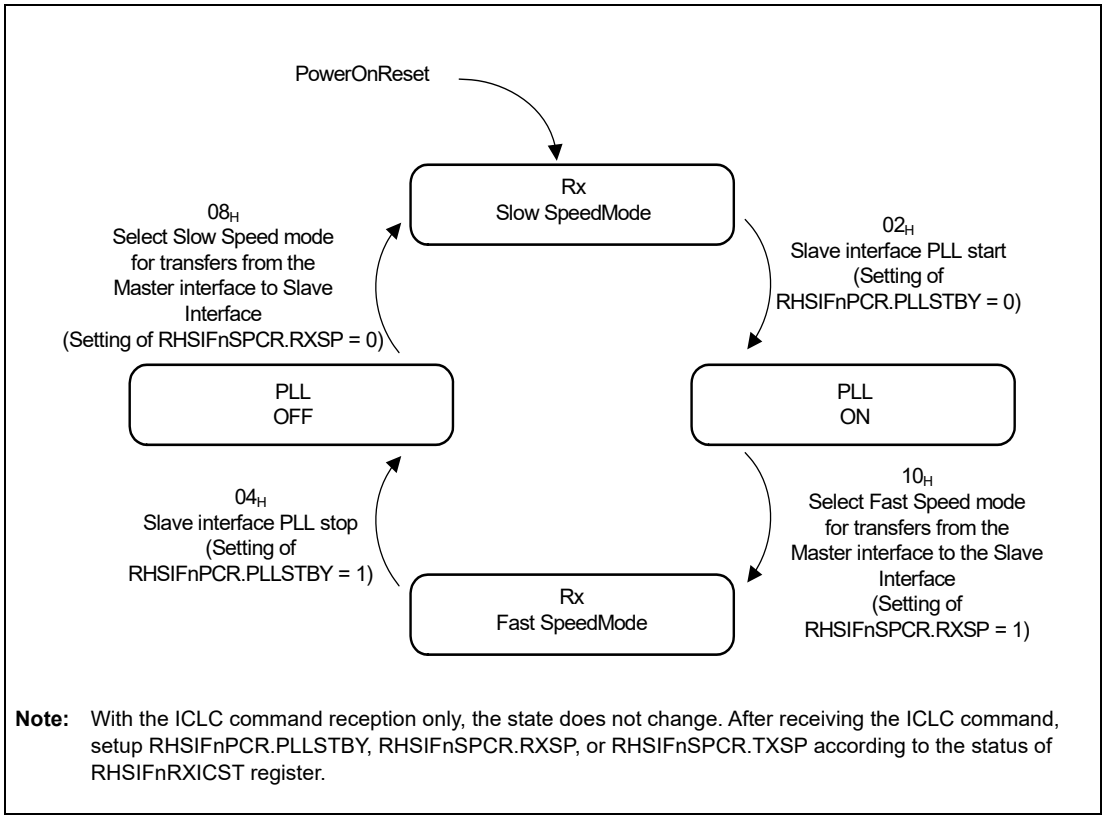
(2) Tx Speed Mode Slow / Fast



Note: With the ICLC command reception only, the state does not change. After receiving the ICLC command, setup RHSIFnPCR.PLLSTBY, RHSIFnSPCR.RXSP, or RHSIFnSPCR.TXSP according to the status of RHSIFnRXICST register.

Figure 29.49 Transition between Tx Speed Modes Slow and Fast

(3) Rx Speed Mode Slow / Fast



Note: With the ICLC command reception only, the state does not change. After receiving the ICLC command, setup RHSIFnPCR.PLLSTBY, RHSIFnSPCR.RXSP, or RHSIFnSPCR.TXSP according to the status of RHSIFnRXICST register.

Figure 29.50 Transition between the Slow and Fast RX Speed Modes

(4) Test Mode

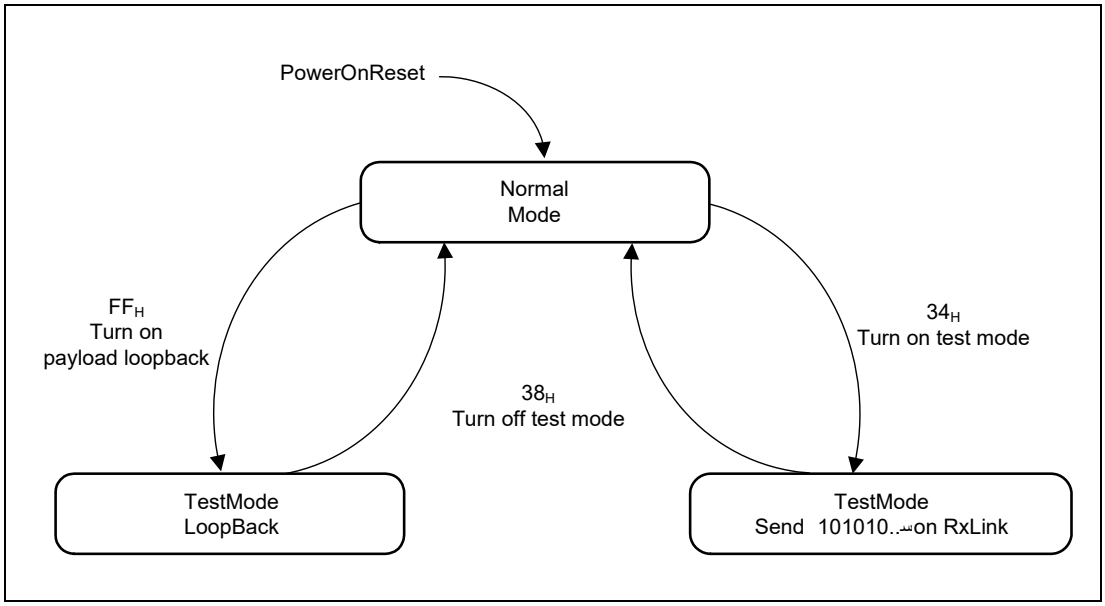


Figure 29.51 Transition between Test Modes

29.5.3.5 CTS Commands

(1) For Flow Control by CTS

The automatic flow control is enabled after PowerOnReset.

The automatic flow control is disabled by the RHSIFnMDCR.CTSEN=0.

(a) When the Automatic Flow Control is Enabled

(i) Link partner is not ready

Link partner: Transmits a frame with a CTS value =0.

This Product: Receives a frame and reflects the CTS value to RHSIFnL1SR.RCTS

This Product: Suppresses the next frame transmission

Link partner: Transmits a frame with a CTS value = 1 after becoming receivable

This Product: Receives a frame and reflects the CTS value to RHSIFnL1SR.RCTS

This Product: Resumes the transmission of the next frame.

(ii) This product is not ready

This Product: When RxDFIFO is full, transmits a CTS command frame of CTS value = 0.

This Product: When RxDFIFO is not full, transmits a CTS command frame with a CTS value = 1

CAUTION

Do not write '0' to RHSIFnMDCR.CTSV while Automatic Flow Control is enabled.

(b) When the Automatic Flow Control is Disabled

By using the following register, it is possible to control the flow manually.

(i) reception status of Link partner

RHSIFnL1SR.RCTS

(ii) reception status of This Product

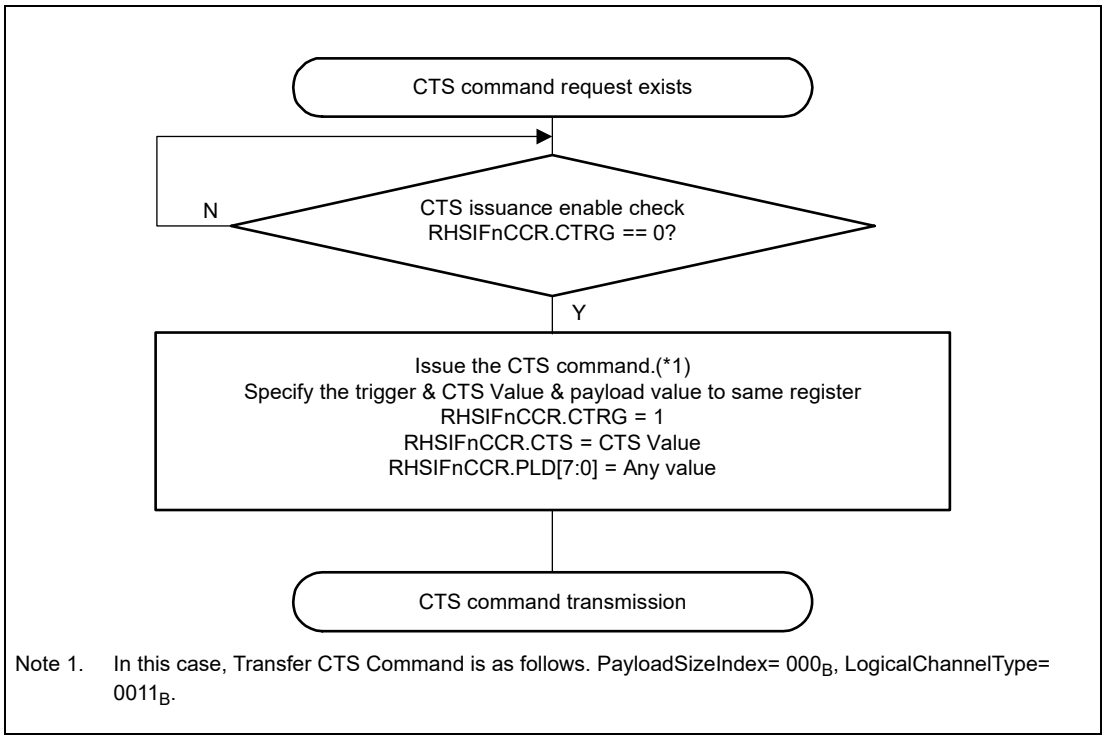
RHSIFnL1SR.RFFUL

RHSIFnMDCR.CTSV

(2) Basic Operation: CTS Command Issuance

A CTS command request is transmitted by firmware. Transmit a request by accessing the register.

Figure 29.52 below shows the procedure flow for issuing a CTS command.



Note 1. In this case, Transfer CTS Command is as follows. PayloadSizeIndex= 000_B, LogicalChannelType= 0011_B.

Figure 29.52 Procedure Flow for Issuing CTS Command

29.5.3.6 Sleep Mode

(1) Transitioning to Sleep Mode for the Link Partner

If an L2 frame, ICLC command, or CTS command is transmitted after RHSIFnSMCR.SLP is set to “1”, the L1 END bit continues to transmit “1”. This makes the Link Partner to transit to sleep mode.

(2) Recovering from Sleep Mode

Enter either of the following signals to the reception data pin to recover the device from sleep mode.

- In fast transfer mode: Zero-level signal of 8-bit rate or more
- In slow transfer mode: Zero-level signal of 1-bit rate or more

This makes the device to restore from sleep mode.

29.5.4 Interrupt Specification

29.5.4.1 Interrupt Source List

The tables below list the error interrupt sources detected in L1.

Table 29.89 Transmission Completion Interrupt

No	Register		Description
	Register Name	Bit Name	
1	RHSIFnTXCMP	TCL2	Transmission frame L2 completion interrupt
2		TCCT	Transmission frame CTS completion interrupt
3		TCIC	Transmission frame ICLC completion interrupt

Table 29.90 Transmission Error Interrupts

No	Register		Description
	Register Name	Bit Name	
1	RHSIFnTXERRST	TERSZ	This bit indicates the status of payload size error of transmission.
2		TERSZI5	Transmission frame L1 header payload code error (undefined code 5_H detected)
3		TERSZI4	Transmission frame L1 header payload code error (undefined code 4_H detected)
4		TERCTF	Transmission frame L1 header channel type code error (undefined code F_H detected)
5		TERCTE	Transmission frame L1 header channel type code error (undefined code E_H detected)
6		TERCTD	Transmission frame L1 header channel type code error (undefined code D_H detected)
7		TERCTC	Transmission frame L1 header channel type code error (undefined code C_H detected)
8		TERCTB	Transmission frame L1 header channel type code error (undefined code B_H detected)
9		TERCTA	Transmission frame L1 header channel type code error (undefined code A_H detected)
10		TERCT9	Transmission frame L1 header channel type code error (undefined code 9_H detected)
11		TERCT8	Transmission frame L1 header channel type code error (undefined code 8_H detected)
12		TERCT2	Transmission frame L1 header channel type code error (undefined code 2_H detected)
13		TERCT1	Transmission frame L1 header channel type code error (undefined code 1_H detected)

Table 29.91 Reception Completion Interrupt

No	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXCMPST	RCL2	Reception frame L2 completion interrupt
2		RCCT	Reception frame CTS completion interrupt
3		RCIC	Reception frame ICLC completion interrupt

Table 29.92 Reception Error Interrupts

No	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXERRST	RERSZ	This bit indicates the status of payload size error of reception.
2		RERSZI5	Reception frame L1 header payload code error (undefined code 5_H detected)
3		RERSZI4	Reception frame L1 header payload code error (undefined code 4_H detected)
4		RERCTF	Reception frame L1 header channel type code error (undefined code F_H detected)
5		RERCTE	Reception frame L1 header channel type code error (undefined code E_H detected)
6		RERCTD	Reception frame L1 header channel type code error (undefined code D_H detected)
7		RERCTC	Reception frame L1 header channel type code error (undefined code C_H detected)
8		RERCTB	Reception frame L1 header channel type code error (undefined code B_H detected)
9		RERCTA	Reception frame L1 header channel type code error (undefined code A_H detected)
10		RERCT9	Reception frame L1 header channel type code error (undefined code 9_H detected)
11		RERCT8	Reception frame L1 header channel type code error (undefined code 8_H detected)
12		RERCT2	Reception frame L1 header channel type code error (undefined code 2_H detected)
13		RERCT1	Reception frame L1 header channel type code error (undefined code 1_H detected)
14		RERIPV	Reception ICLC command undefined error.

Table 29.93 ICLC Reception Completion Interrupt

No	Register		Description
	Register Name	Bit Name	
1	RHSIFnRXICST	RIPA	Reception of PING answer frame
2		RITOL	Reception of turn on payload loopback frame
3		RITOF	Reception of turn off test mode frame
4		RITON	Reception of turn on test mode frame
5		RIDT	Reception of Disable slave interface transmitter frame
6		RIET	Reception of Enable slave interface transmitter frame
7		RIFT	Reception of Select Fast Speed mode for transfers from the Slave interface to the Master interface frame
8		RIST	Reception of Select Slow Speed mode for transfers from the Slave interface to the Master interface frame
9		RIFR	Reception of Select Fast Speed mode for transfers from the Master interface to the Slave interface frame
10		RISR	Reception of Select Slow Speed mode for transfers from the Master interface to the Slave interface frame
11		RIPSTP	Reception of Slave interface PLL stop
12		RIPSRT	Reception of Slave interface PLL start
13		RIPG	Reception of PING frame

29.5.4.2 Error detect conditions

The tables below list the error detect conditions.

Table 29.94 The combination of Transmission Payload size index and Transmission Logical channel type

			Logical Channel Type						
			ICLC(*2)	CTS(*2)	L2Frame				
			0000_B	0011_B	0100_B	0101_B	0110_B	0111_B	others(*3)
Payload_Size_Index	8bit	000_B	TCIC	TCCT	TERSZ	TERSZ	TERSZ	TERSZ	TERSZ
	32bit	001_B	TCIC	-(*1)	TCL2	TCL2	TCL2	TCL2	TERSZ
	64bit	010_B	-(*1)	-(*1)	TCL2	TCL2	TCL2	TCL2	TERSZ
	96bit	011_B	-(*1)	-(*1)	TCL2	TCL2	TCL2	TCL2	TERSZ
	Not used	100_B	-(*1)	-(*1)	TERSZI4	TERSZI4	TERSZI4	TERSZI4	TERSZ TERSZI4(*4)
		101_B	-(*1)	-(*1)	TERSZI5	TERSZI5	TERSZI5	TERSZI5	TERSZ TERSZI5(*4)
	160bit	110_B	-(*1)	-(*1)	TCL2	TCL2	TCL2	TCL2	TERSZ
	288bit	111_B	-(*1)	-(*1)	TCL2	TCL2	TCL2	TCL2	TERSZ

Note 1. These conditions will not occur.

Note 2. In L1 Frame, error bits are generated only when Logical Channel Type is 0000_B(ICLC) or 0011_B(CTS).

Note 3. In L2 Frame, the generated error bit is always TERSZ when the Logical Channel Type is other than b01xx.

Note 4. Either TERSZI4 or TERSZI5 is simultaneously generated with TERSZ according to the Payload Size Index value.

Table 29.95 The combination of Reception Payload size index and Reception Logical channel type (1 of 2)

			Logical_Channel_Type							
			ICLC	Reserved		CTS	L2Frame			
			0000_B	0001_B	0010_B	0011_B	0100_B	0101_B	0110_B	0111_B
Payload_Size_Index	8bit	000_B	RCIC	RERCT1	RERCT2	RCCT	RERSZ	RERSZ	RERSZ	RERSZ
	32bit	001_B	RCIC RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	64bit	010_B	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	96bit	011_B	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	Not used	100_B	RERSZI4	RERCT1 RERSZI4(*1)	RERCT2 RERSZI4(*1)	RERSZI4	RERSZI4	RERSZI4	RERSZI4	RERSZI4
		101_B	RERSZI5	RERCT1 RERSZI5(*1)	RERCT2 ERSZI5(*1)	RERSZI5	RERSZI5	RERSZI5	RERSZI5	RERSZI5
	160bit	110_B	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2
	288bit	111_B	RERSZ	RERCT1	RERCT2	RCCT	RCL2	RCL2	RCL2	RCL2

Note 1. Either RERSZI4 or RERSZI5 is simultaneously generated with RERCT1 or RERCT2 according to the Payload Size Index value.

Table 29.96 The combination of Reception Payload size index and Reception Logical channel type (2 of 2)

			Logical_Channel_Type							
			Reserved							
			1000_B	1001_B	1010_B	1011_B	1100_B	1101_B	1110_B	1111_B
Payload_Size_Index	8bit	000_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	32bit	001_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	64bit	010_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	96bit	011_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	Not used	100_B	RERCT8 RERSZI4(*1)	RERCT9 RERSZI4(*1)	RERCTA RERSZI4(*1)	RERCTB RERSZI4(*1)	RERCTC RERSZI4(*1)	RERCTD RERSZI4(*1)	RERCTE RERSZI4(*1)	RERCTF RERSZI4(*1)
		101_B	RERCT8 RERSZI5(*1)	RERCT9 RERSZI5(*1)	RERCTA RERSZI5(*1)	RERCTB RERSZI5(*1)	RERCTC RERSZI5(*1)	RERCTD RERSZI5(*1)	RERCTE RERSZI5(*1)	RERCTF RERSZI5(*1)
	160bit	110_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF
	288bit	111_B	RERCT8	RERCT9	RERCTA	RERCTB	RERCTC	RERCTD	RERCTE	RERCTF

Note 1. Either RERSZI4 or RERSZI5 is simultaneously generated with RERCT8~RERCTF according to the Payload Size Index value.

29.5.4.3 Operation Upon Error Occurrence

The operations upon error occurrence in L1 consist only of the following.

- Assertion of error interrupt
- Indication of error interrupt status by register

29.5.5 Debug

Table 29.97 List of Debug Function

Purpose	Item	Content		APB Register
Debug	ICLC(34_H)	Turn on test mode (Send 101010.. Txline data rate)		RHSIFnTMDCR.CKTMSEL
	Loopback	TxD->RxD		RHSIFnTMDCR.TXLPBK
	Sent Frame information	Last sent frame information display	L1 frame	RHSIFnLTXFRMRL1.TL1E
				RHSIFnLTXFRMRL1.TL1P
				RHSIFnLTXFRMRL1.TL1H
			L2 frame	RHSIFnLTXFRMRL2.T2L2H
				RHSIFnLTXFRMRL2.T2L1H
	Received frame information	Last received frame information display	L1 frame	RHSIFnLRXFRMRL1.RL1E
				RHSIFnLRXFRMRL1.RL1P
				RHSIFnLRXFRMRL1.RL1H
			L2 frame	RHSIFnLRXFRMRL2.R2L2H
				RHSIFnLRXFRMRL2.R2L1H

29.5.6 Usage Note

29.5.6.1 About Fast Mode of Slave

A fast mode of Slave (It's sampled using a reference clock.) recommends following correspondence because there is a possibility that BER (the bit error rate) deteriorates by influence of a jitter.

- Use Slave's fast mode only when sending / receiving various ICLC commands (during communication establishment).
- After confirming the establishment of communication by the PING command of ICLC when communicating by a fast mode, begin to change to Master (sampling by MOSC) and communicate from tRHPLLLCT µs later (PLL locked time of a clock change). For details of PLL locked time, refer to **Section 55, Electrical Characteristics**.

Section 30 Clock Extension Peripheral Interface (CXPI)

This section contains a generic description of the clock extension peripheral interface (CXPI).

The first part in this section describes the RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the CXPI.

30.1 Features of CXPI for RH850/U2A-EVA

30.1.1 Number of Units and Channels

This microcontroller has the following number of CXPI units.

Each CXPI unit has a single channel interface. “Number of channels” therefore has the same meaning as “number of units” in this section.

Table 30.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	—	—	—	—
Name	CXP1n									

Table 30.2 Index

Index	Description
n	Throughout this section, the individual CXPI function A units are identified by the index “n” (n = 0 to 3); for example, CXP1nMODE indicates the CXP1n mode selection register.

30.1.2 Register Base Addresses

CXP1n base addresses are listed in the following table.

CXP1n register addresses are given as offsets from the base addresses.

Table 30.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<CXP10_base>	FFD4 0000_H	P-Bus Group 3
<CXP11_base>	FFD4 1000_H	P-Bus Group 3
<CXP12_base>	FFD4 2000_H	P-Bus Group 3
<CXP13_base>	FFD4 3000_H	P-Bus Group 3

30.1.3 Clock Supply

The CXP1n clock supply is shown in the following table.

Table 30.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
CXP1n	CXPI Communication clock (PCLK)	CLK_HSB	Timer count clock
	Register access clock	CLK_HSB	Bus clock

30.1.4 Interrupt Request and Error Notifications

CXP1n interrupt requests are listed in the following table.

Table 30.5 Interrupt Request

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CXP10			
INTCXP10TI	CXPI transfer interrupt	673	—
INTCXP10RI	CXPI receive interrupt	674	—
INTCXP10SI	CXPI status interrupt	675	—
CXP11			
INTCXP11TI	CXPI transfer interrupt	676	—
INTCXP11RI	CXPI receive interrupt	677	—
INTCXP11SI	CXPI status interrupt	678	—
CXP12			
INTCXP12TI	CXPI transfer interrupt	679	—
INTCXP12RI	CXPI receive interrupt	680	—
INTCXP12SI	CXPI status interrupt	681	—
CXP13			
INTCXP13TI	CXPI transfer interrupt	682	—
INTCXP13RI	CXPI receive interrupt	683	—
INTCXP13SI	CXPI status interrupt	684	—

This module has no error notifications.

30.1.5 Reset Sources

CXP1n reset sources are listed in the following table. CXP1n is initialized by these reset sources.

Table 30.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
CXP1n	All registers	√	√	√	√	√	√	—

30.1.6 External Input/Output Signals

External input/output signals of CXP1n are listed below.

Table 30.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
CXP10		
pt_cxpi_rxd	CXPI receive data Input	CXP10RX
cxpi_pt_txd	CXPI transmit data Output	CXP10TX
CXP11		
pt_cxpi_rxd	CXPI receive data Input	CXP11RX
cxpi_pt_txd	CXPI transmit data Output	CXP11TX
CXP12		
pt_cxpi_rxd	CXPI receive data Input	CXP12RX
cxpi_pt_txd	CXPI transmit data Output	CXP12TX
CXP13		
pt_cxpi_rxd	CXPI receive data Input	CXP13RX
cxpi_pt_txd	CXPI transmit data Output	CXP13TX

30.1.7 Combinations of Pins and Ports

Combinations of CXPI pins and ports are listed in the following table.

Table 30.8 Combinations of Pins and Ports

Function	Pin name	Port name
CXP10	CXP10RX	P2_9/P4_9/P10_5/P10_8/P20_2/P20_14
	CXP10TX	P2_10/P4_8/P10_0/P10_4/P10_6/P10_9
CXP11	CXP11RX	P2_4/P4_4
	CXP11TX	P2_5/P4_5
CXP12	CXP12RX	P2_1/P3_2/P21_4
	CXP12TX	P2_0/P3_3/P21_5
CXP13	CXP13RX	P2_11/P4_15
	CXP13TX	P2_12/P4_14

30.2 Overview

The Clock Extension Peripheral Interface (CXPI) supports the following standard specification (JASO D015) functions:

- Baud rate can be chosen
  - 9.6 kbps/10.4 kbps/19.2 kbps/20.0 kbps
- Operation mode
  - Master mode/Slave mode
- Access method
  - Event trigger method/Polling method
- PWM encoding/decoding function is used to output and sample the PWM waveform

30.2.1 Block Diagram

The figures below are block diagrams of the CXPI.

- When CXP1nMODE.CXP1nMD = 0_B (in CXPI-PWM mode) (The IP (MCU) handles encoding and decoding.)

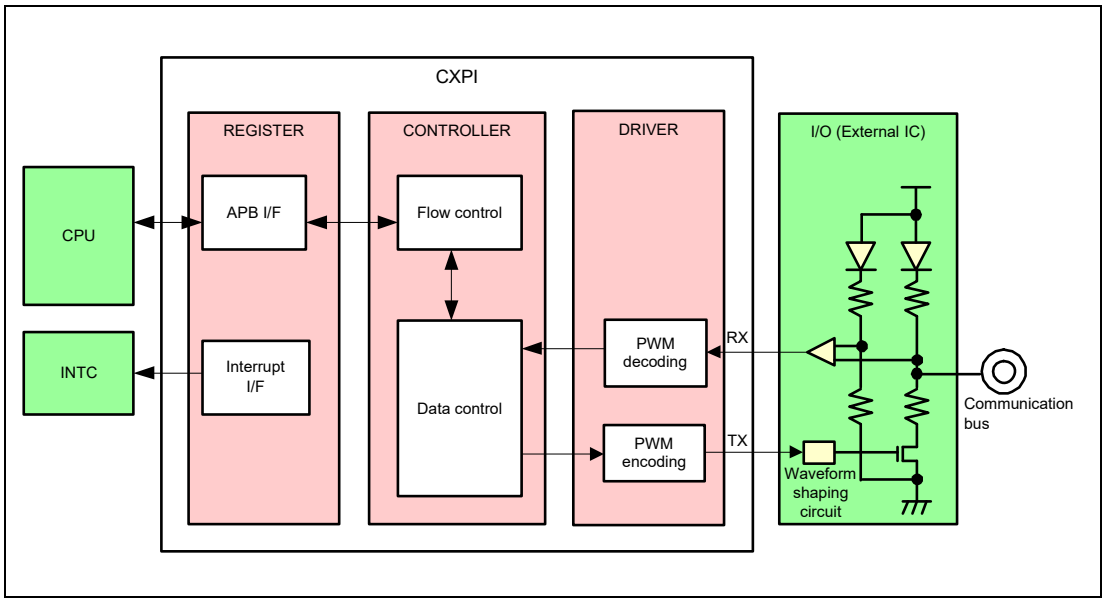


Figure 30.1 Block Diagram (in CXPI-PWM Mode)

- When CXP1nMODE.CXP1nMD = 1_B (in CXPI-NRZ mode) (The external IC handles encoding and decoding.)

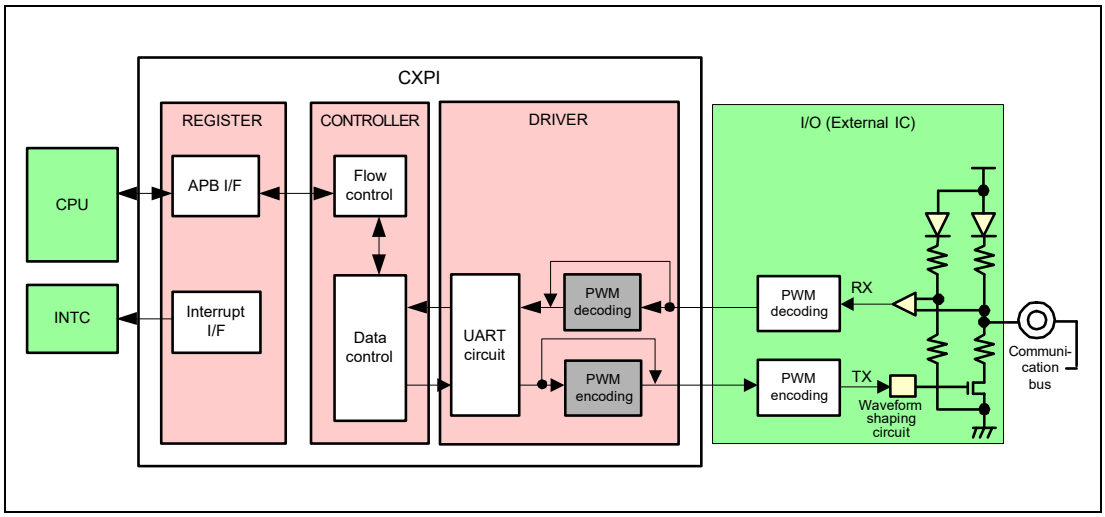


Figure 30.2 Block Diagram (in CXPI-NRZ Mode)

30.3 List of Registers

30.3.1 List of CXP1n Control Registers

The table below lists the CXP1n control registers.

Table 30.9 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
CXP1n	CXPI Operating Mode Select Register	CXP1nMODE	<CXP1n_base> + 000_H	8, 16, 32	*1	—
	CXPI Baud Rate Setting Register	CXP1nBRT	<CXP1n_base> + 004_H	8, 16, 32	*1	—
	CXPI Logical Value 1 Setting Register	CXP1nV1LW	<CXP1n_base> + 008_H	8, 16, 32	*1	—
	CXPI Logical Value 0 Setting Register	CXP1nV0LW	<CXP1n_base> + $00C_H$	8, 16, 32	*1	—
	CXPI Sampling Setting Register	CXP1nSAMP	<CXP1n_base> + 010_H	8, 16, 32	*1	—
	CXPI Input Filter Setting Register	CXP1nFIL	<CXP1n_base> + 014_H	8, 16, 32	*1	—
	CXPI Wakeup Pulse Setting Register	CXP1nWU	<CXP1n_base> + 018_H	8, 16, 32	*1	—
	CXPI Inter-Frame Setting Register	CXP1nFRMW	<CXP1n_base> + $01C_H$	8, 16, 32	*1	—
	CXPI Sleep Setting Register	CXP1nSLP	<CXP1n_base> + 020_H	8, 16, 32	*1	—
	CXPI Retransmission Setting Register	CXP1nREP	<CXP1n_base> + 024_H	8, 16, 32	*1	—
	CXPI Wakeup Control Register	CXP1nWUP	<CXP1n_base> + 028_H	8, 16, 32	*1	—
	CXPI Transmission Control Register	CXP1nSND	<CXP1n_base> + $02C_H$	8, 16, 32	*1	—
	CXPI Interrupt Mask Register	CXP1nIMSK	<CXP1n_base> + 030_H	8, 16, 32	*1	—
	CXPI Interrupt Source Register	CXP1nINT	<CXP1n_base> + 034_H	8, 16, 32	*1	—
	CXPI Flag Register 1	CXP1nFLG1	<CXP1n_base> + 038_H	8, 16, 32	*1	—
	CXPI Flag Register 2	CXP1nFLG2	<CXP1n_base> + $03C_H$	8, 16, 32	*1	—
	CXPI Flow Register 1	CXP1nFLW1	<CXP1n_base> + 040_H	8, 16, 32	*1	—
	CXPI Flow Register 2	CXP1nFLW2	<CXP1n_base> + 044_H	8, 16, 32	*1	—
	CXPI Error Register	CXP1nERR	<CXP1n_base> + $04C_H$	8, 16, 32	*1	—
	CXPI received PID register	CXP1nRPID	<CXP1n_base> + 060_H	8, 16, 32	*1	—
	CXPI transmission PID register	CXP1nSPID	<CXP1n_base> + 070_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 101	CXP1nRPIDF101	<CXP1n_base> + 080_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 102	CXP1nRPIDF102	<CXP1n_base> + 084_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 103	CXP1nRPIDF103	<CXP1n_base> + 088_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 104	CXP1nRPIDF104	<CXP1n_base> + $08C_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 105	CXP1nRPIDF105	<CXP1n_base> + 090_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 106	CXP1nRPIDF106	<CXP1n_base> + 094_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 107	CXP1nRPIDF107	<CXP1n_base> + 098_H	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 108	CXP1nRPIDF108	<CXP1n_base> + $09C_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 109	CXP1nRPIDF109	<CXP1n_base> + $0A0_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 110	CXP1nRPIDF110	<CXP1n_base> + $0A4_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 111	CXP1nRPIDF111	<CXP1n_base> + $0A8_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 112	CXP1nRPIDF112	<CXP1n_base> + $0AC_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 201	CXP1nRPIDF201	<CXP1n_base> + $0B0_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 202	CXP1nRPIDF202	<CXP1n_base> + $0B4_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 203	CXP1nRPIDF203	<CXP1n_base> + $0B8_H$	8, 16, 32	*1	—
	CXPI Received Frame ID Setting Register 204	CXP1nRPIDF204	<CXP1n_base> + $0BC_H$	8, 16, 32	*1	—

Table 30.9 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
CXP1n	CXPI Automatic Response PID Setting Register	CXP1nARPID	<CXP1n_base> + $0C0_H$	8, 16, 32	*1	—
	CXPI Received Frame Information Register	CXP1nRFRI	<CXP1n_base> + 120_H	8, 16, 32	*1	—
	CXPI Transmission Frame Information Register	CXP1nSFRI	<CXP1n_base> + 130_H	8, 16, 32	*1	—
	CXPI Automatic Response Frame Information Register	CXP1nARFRI	<CXP1n_base> + 140_H	8, 16, 32	*1	—
	CXPI Received CRC Register	CXP1nRCRC	<CXP1n_base> + $1A0_H$	8, 16, 32	*1	—
	CXPI Transmission CRC Register	CXP1nSCRC	<CXP1n_base> + $1B0_H$	8, 16, 32	*1	—
	CXPI Automatic Response CRC Register	CXP1nARCRC	<CXP1n_base> + $1C0_H$	8, 16, 32	*1	—
	CXPI Received Data Register 01	CXP1nRDATA01	<CXP1n_base> + 200_H	8, 16, 32	*1	—
	CXPI Received Data Register 02	CXP1nRDATA02	<CXP1n_base> + 204_H	8, 16, 32	*1	—
	CXPI Received Data Register 03	CXP1nRDATA03	<CXP1n_base> + 208_H	8, 16, 32	*1	—
	CXPI Received Data Register 04	CXP1nRDATA04	<CXP1n_base> + $20C_H$	8, 16, 32	*1	—
	CXPI Received Data Register 05	CXP1nRDATA05	<CXP1n_base> + 210_H	8, 16, 32	*1	—
	CXPI Received Data Register 06	CXP1nRDATA06	<CXP1n_base> + 214_H	8, 16, 32	*1	—
	CXPI Received Data Register 07	CXP1nRDATA07	<CXP1n_base> + 218_H	8, 16, 32	*1	—
	CXPI Received Data Register 08	CXP1nRDATA08	<CXP1n_base> + $21C_H$	8, 16, 32	*1	—
	CXPI Transmission Data Register 01	CXP1nSDATA01	<CXP1n_base> + 300_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 02	CXP1nSDATA02	<CXP1n_base> + 304_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 03	CXP1nSDATA03	<CXP1n_base> + 308_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 04	CXP1nSDATA04	<CXP1n_base> + $30C_H$	8, 16, 32	*1	—
	CXPI Transmission Data Register 05	CXP1nSDATA05	<CXP1n_base> + 310_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 06	CXP1nSDATA06	<CXP1n_base> + 314_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 07	CXP1nSDATA07	<CXP1n_base> + 318_H	8, 16, 32	*1	—
	CXPI Transmission Data Register 08	CXP1nSDATA08	<CXP1n_base> + $31C_H$	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 01	CXP1nARDATA01	<CXP1n_base> + 400_H	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 02	CXP1nARDATA02	<CXP1n_base> + 404_H	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 03	CXP1nARDATA03	<CXP1n_base> + 408_H	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 04	CXP1nARDATA04	<CXP1n_base> + $40C_H$	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 05	CXP1nARDATA05	<CXP1n_base> + 410_H	8, 16, 32	*1	—
	CXPI Automatic Response Data Register 06	CXP1nARDATA06	<CXP1n_base> + 414_H	8, 16, 32	*1	—

Note 1. n = 0: PBG31#1
n = 1: PBG31#2
n = 2: PBG31#3
n = 3: PBG31#4

30.4 Function Details

30.4.1 Register Descriptions

30.4.1.1 CXP1nMODE —CXPI Operating Mode Select Register

This register sets the operation of the CXPI controller.

Write access to this register is only enabled in bus sleep mode*2. It is disabled in other modes.

Note 2. CXP1nFLW1.CXP1nCXMD = 01_B

Access: CXP1nMODE register can be read or written in 32-bit units.
CXP1nMODEL register can be read or written 16-bit units.
CXP1nMODELL, CXP1nMODELH registers can be read or written 8-bit units.

Address: CXP1nMODE: <CXP1n_base> + 000_H
CXP1nMODEL: <CXP1n_base> + 000_H
CXP1nMODELL: <CXP1n_base> + 000_H,
CXP1nMODELH: <CXP1n_base> + 001_H

Value after reset: 0000 0200_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CXP1nARESEN	CXP1nREPOP	CXP1nACC	—	—	—	—	CXP1nMS	—	CXP1nMD	CXP1nEN
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W	R/W

Table 30.10 CXP1nMODE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	CXP1nARESEN	Automatic response enable 0: Disables automatic response 1: Enables automatic response
9	CXP1nREPOP	Retransmission setting 0: Retransmission is not judged (unlimited number of retransmissions) 1: Retransmission is judged (the number of retransmissions is set by the CXP1nREP register)
8	CXP1nACC	CXPI access method setting 0: Event trigger method 1: Polling method
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CXP1nMS	Node setting 0: Slave node 1: Master node
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CXP1nMD	Transfer mode setting 0: CXPI-PWM mode 1: CXPI-NRZ mode

Table 30.10 CXP1nMODE Register Contents (2/2)

Bit Position	Bit Name	Function
0	CXP1nEN	Operation enable 0: Disables operation (forced reset issued) 1: Enables operation

CXP1nARESEN Bit

This bit enables/disables automatic response.
For details, see **Section 30.5.4.4(1), PID Reception to Response Transmission**.

When automatic response is enabled (1_B), the PID received from another node is compared with the automatic response PID set in the CXP1nARPID register.
If they match, a frame of the response field is transmitted automatically.

NOTE

This function is defined as automatic response.

CXP1nREPOP Bit

This bit specifies whether or not retransmission is determined.
For details, see **Section 30.5.7.3, Retransmission Processing**.

[What is retransmission processing?]
This function retransmits a frame if the received data differs from the transmitted data due to a loss in arbitration, error, etc.
The node which sent the PID retransmits the same PID. Retransmission must be done after IFS has been detected.

- When CXP1nREPOP = 1_B:
  The number of retransmissions is the value set in the CXP1nREP register.
  For details, see **Section 30.4.1.10, CXP1nREP — CXPI Retransmission Setting Register**.
- When CXP1nREPOP = 0_B:
  The number of retransmissions is unlimited.
  Note that setting CXP1nREPOP to 0_B initializes the CXP1nFLG2 register, disabling writing to the register.

CXP1nACC Bit

This bit specifies the access method of the CXPI.
For details, see **Section 30.5.2, CXPI Operation Overview**.

- When CXP1nACC = 0_B:
  The CXPI operates in the event trigger method.
  Each node can transmit frames freely upon detecting an idle state.
- When CXP1nACC = 1_B:
  The CXPI operates in the polling method.
  The slave node transmits a response for the request from the master node.
  It can transmit any frame when receiving the PTYPE field that represents an event request from the master node.

The master node operates in the same way as in the event trigger method.

CXP1nMS Bit

This bit specifies whether the node is the master node or a slave node.

- When CXP1nMS = 0_B:
  The node operates as a slave node.
  A slave node operates in synchronization with the CXPI clock signal from the master node.
- When CXP1nMS = 1_B:
  The node operates as the master node.
  The master node transmits CXPI clock signals to the communication bus.

CXP1nMD Bit

This bit specifies the transfer mode of the CXPI.
For details, see **Section 30.5.5.5, PWM Encoding/Decoding**.

- When CXP1nMD = 0_B:
  The CXPI operates in CXPI-PWM mode.
  In CXPI-PWM mode, the CXPI uses PWM encoding/decoding by the IP to transmit and receive on a PWM basis.
- When CXP1nMD = 1_B:
  The CXPI operates in CXPI-NRZ mode.
  In CXPI-NRZ mode, the CXPI handles transfer to the transceiver in the UART format.
  The transceiver encodes/decodes the UART protocol into PWM and outputs the resulting data as CXPI communications (PWM encoding/decoding by the IP is turned off).

CXP1nEN Bit

This bit enables/disables the CXPI operation.

- When CXP1nEN = 1_B (enabling operation):
  Various settings and operations are enabled.
  Note that the mode sequencer[*1] transitions from initializing mode to bus sleep mode.
- When CXP1nEN = 0_B (disabling operation):
  Various settings and operations are disabled.
  Note that the mode sequencer[*1] transitions to initializing mode.

Setting CXP1nEN to 0_B issues a forced reset.
A forced reset initializes all registers except for this register. In the slave node, the CXP1nFLG1.CXP1nFAIL bit and the CXP1nERR.CXP1nERRCNT bit are not initialized.

In the master node, the CXP1nFLG1.CXP1nFAIL bit and the CXP1nERR.ERRCNT bit are also initialized.

Note 1. For details, see **Section 30.5.8.7, Software Reset**.

If the application found a physical bus error, a forced reset is issued to initialize the IP if sleep mode is enabled.
For details about the mode sequencer, see **Section 30.4.1.17, CXP1nFLW1 — CXP1 Flow Register 1**.

30.4.1.2 CXP1nBRT — CXPI Baud Rate Setting Register

This register sets the CXPI nominal bit (Tbit) frequency of the CXPI bus.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Access: CXP1nBRT register can be read or written in 32-bit units.
CXP1nBRTL register can be read or written 16-bit units.
CXP1nBRTLL, CXP1nBRTLH registers can be read or written 8-bit units.

Address: CXP1nBRT: <CXP1n_base> + 004_H
CXP1nBRTL: <CXP1n_base> + 004_H
CXP1nBRTLL: <CXP1n_base> + 004_H,
CXP1nBRTLH: <CXP1n_base> + 005_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CXP1nBRP											—	CXP1nPRS		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 30.11 CXP1nBRT Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 4	CXP1nBRP	CXPI nominal bit time setting (1 Tbit width) The CXPI nominal bit (Tbit) frequency is calculated by using the following equation: • CXPI nominal bit (Tbit) frequency = operating reference frequency (divclk) / CXP1nBRP
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CXP1nPRS	CXPI basic division ratio setting The operating reference frequency is calculated by using the following equation: • Operating reference frequency (divclk) = input frequency (PCLK) / (CXP1nPRS + 1)

CXP1nBRP Bits

These bits set the division ratio of the operating reference frequency (divclk).
The divided clock of the operating reference frequency (divclk) is defined as the CXPI nominal bit (Tbit) frequency.
This clock is used to generate the communication baud rate and wakeup pulses.

CXP1nPRS Bits

These bits set the division ratio of PCLK to generate the operating reference frequency (divclk).
The operating reference frequency (divclk) is used to generate the communication waveform and sampling timing.

The conditions for setting and clearing of the CXP1nBRT register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Recommended Communication Baud Rate Settings

The table below lists the recommended communication baud rate settings.

Table 30.12 List of Communication Baud Rate Settings

Macro Operating Frequency (PCLK)	Target Communication Baud Rate	CXP1nPRS	CXP1nBRP	Generated Communication Baud Rate
40 MHz	20 kbps	000_B	$7D0_H$	20 kbps
	19.2 kbps	001_B	412_H	19.19 kbps
	10.4 kbps	001_B	783_H	10.40 kbps
		010_B	502_H	10.40 kbps
	9.6 kbps	010_B	$56D_H$	9.60 kbps
		011_B	412_H	9.60 kbps

Relationship between Clocks and Functions

This IP operates based on the following three clocks:

- Bus clock (PCLK)
- Operating reference frequency (divclk)
- CXPI nominal bit (Tbit) frequency

The figure below shows the relationship between the clocks and functions (registers).

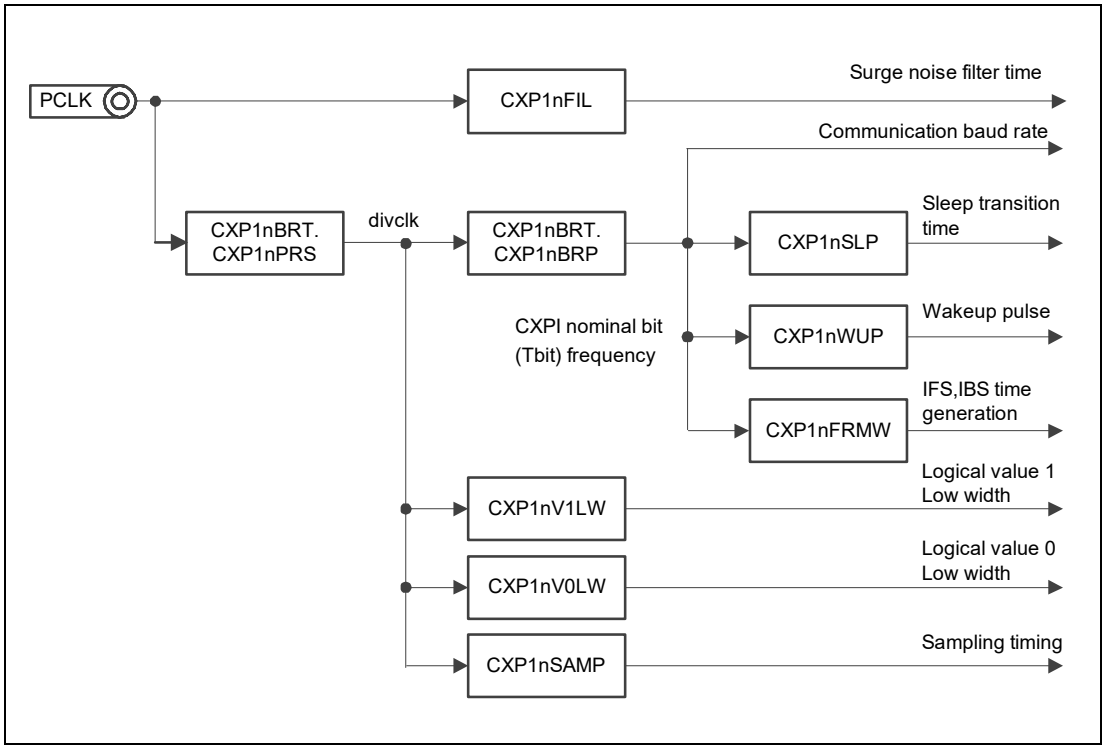


Figure 30.3 Relationship between Clocks and Functions (CXPI-PWM)

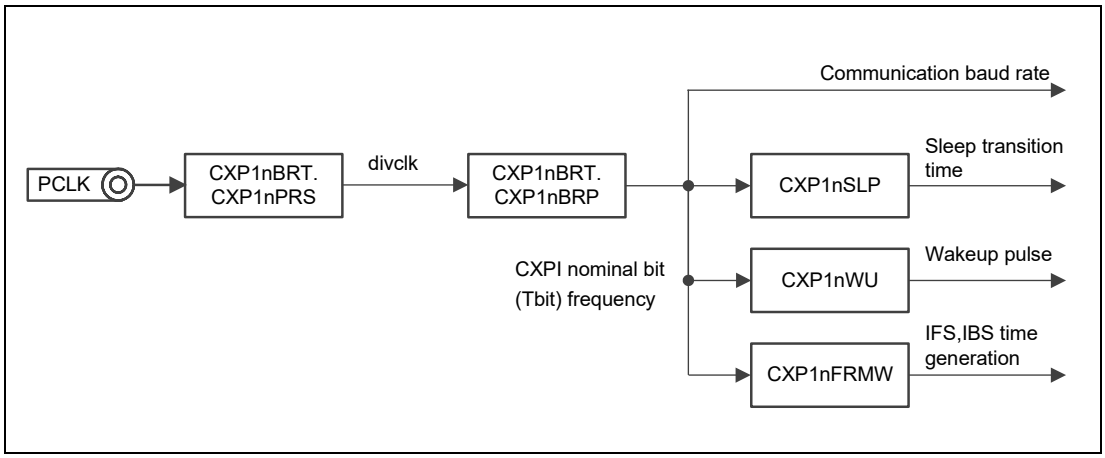


Figure 30.4 Relationship between Clocks and Functions (CXPI-NRZ)

30.4.1.3 CXP1nV1LW — CXPI Logical Value 1 Setting Register

This register sets the low width (startup timing) of logical value 1 in CXPI-PWM mode.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Writing to this register is prohibited in CXPI-NRZ mode.

Access: CXP1nV1LW register can be read or written in 32-bit units.
CXP1nV1LWL register can be read or written 16-bit units.
CXP1nV1LWLL, CXP1nV1LWLH registers can be read or written 8-bit units.

Address: CXP1nV1LW: <CXP1n_base> + 008_H
CXP1nV1LWL: <CXP1n_base> + 008_H
CXP1nV1LWLL: <CXP1n_base> + 008_H,
CXP1nV1LWLH: <CXP1n_base> + 009_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CXP1nV1LCNT									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.13 CXP1nV1LW Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	CXP1nV1LCNT	Logical value 1 low width setting These bits set the low width of logical value 1 in CXPI-PWM mode. The low width of logical value 1 is set by using the following equation: • Low width of logical value 1 = 1 / operating reference frequency (divclk) × CXP1nV1LCNT Range of allowable settings: 001_H to $3FF_H$

CXP1nV1LCNT Bits

These bits set the low width of logical value 1 in CXPI-PWM mode.
The logical value 1 waveform of the CXPI communication bus is generated by this setting.

The table below lists the range of allowable settings.

Table 30.14 Allowable Settings of CXP1nV1LCNT

Macro Operating Frequency (PCLK)	Communication Baud Rate (CXPI Nominal Bit (Tbit) Frequency)	CXP1nPRS	CXP1nBRP	CXP1nV1LCNT*3	
				MIN*1	MAX*2
40 MHz	20 kbps	0000_B	$7D0_H$	$0DC_H$	$30C_H$
	19.2 kbps	0001_B	412_H	073_H	196_H
	10.4 kbps	0001_B	783_H	$0D4_H$	$2ED_H$
		0010_B	502_H	$08E_H$	$1F3_H$
	9.6 kbps	0010_B	$56D_H$	099_H	$21D_H$
		0011_B	412_H	073_H	196_H

Note 1. The setting range of MIN is determined by the following specified value:
Ttx_1_lo_dom: 0.11 × CXPI nominal bit (Tbit) time

Note 2. The setting range of MAX is determined by the following specified value:
Ttx_1_lo_rec: 0.39 × CXPI nominal bit (Tbit) time + 0.6 × τ
τ (time constant of the entire system): 1 µs to 5 µs (excluded from the calculation of the range of allowable settings)

Note 3. In the master node, the startup timing in synchronization with the internally-generated baud rate counter

The conditions for setting and clearing of the CXP1nV1LW register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.15 CXP1nV1LCNT Bit Access Conditions

Access Conditions				Writing	Reading (Read Value)
CXP1nMODE			CXP1nFLW1. CXP1nCXMD		
CXP1nEN	CXP1nMD	CXP1nMS			
0_B	x_B	x_B	xx_B	Not possible	Initial value
1_B	0_B (CXPI-PWM)	0_B (Slave)	xx_B	Disabled	Setting value
		1_B (Master)	01_B (Bus Sleep)	Enabled	Setting value
			Other than above	Disabled	Setting value
	1_B (in CXPI-NRZ mode)	X_B	XX_B	Prohibited	Initial value

Note: "Writing not possible" means that writing is ignored.
"Writing disabled" is a limitation to specifications. Not guarded by hardware.

30.4.1.4 CXP1nV0LW — CXPI Logical Value 0 Setting Register

This register sets the low width (timing) of logical value 0 in CXPI-PWM mode.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Writing to this register is prohibited in CXPI-NRZ mode.

Access: CXP1nV0LW register can be read or written in 32-bit units.
CXP1nV0LWL register can be read or written 16-bit units.
CXP1nV0LWLL, CXP1nV0LWLH registers can be read or written 8-bit units.

Address: CXP1nV0LW: <CXP1n_base> + $00C_H$
CXP1nV0LWL: <CXP1n_base> + $00C_H$
CXP1nV0LWLL: <CXP1n_base> + $00C_H$,
CXP1nV0LWLH: <CXP1n_base> + $00D_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CXP1nV0LCNT										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.16 CXP1nV0LW Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	CXP1nV0LCNT	Logical value 0 low width setting These bits set the low width of logical value 0 in CXPI-PWM mode. • When CXP1nMODE.CXP1nMD = 0_B (CXPI-PWM mode) The low width of logical value 0 is set by using the following equation: • Low width of logical value 0 = 1 / operating reference frequency (divclk) × CXP1nV0LCNT Range of allowable settings: 001_H to $7FF_H$ **NOTE:** Writing to these bits is prohibited in CXPI-NRZ mode.

CXP1nV0LCNT Bits

These bits set the low width of logical value 0 in CXPI-PWM mode.
The logical value 0 waveform of the CXPI communication bus is generated by this setting.

The table below lists the range of allowable settings.

Table 30.17 Allowable Settings of CXP1nV0LCNT

Macro Operating Frequency (PCLK)	Communication Baud Rate (CXPI Nominal Bit (Tbit) Frequency)	CXP1nPRS	CXP1nBRP	CXP1nV0LCNT*3	
				MIN*1	MAX*2
40 MHz	20 kbps	000_B	$7D0_H$	CXP1nV1LCNT + 078_H	758_H
	19.2 kbps	001_B	412_H	CXP1nV1LCNT + $03F_H$	$3D3_H$
	10.4 kbps	001_B	783_H	CXP1nV1LCNT + 074_H	$70F_H$
		010_B	502_H	CXP1nV1LCNT + $04D_H$	$4B5_H$
	9.6 kbps	010_B	$56D_H$	CXP1nV1LCNT + 054_H	519_H
		011_B	412_H	CXP1nV1LCNT + $03F_H$	$3D3_H$

Note 1. The setting range of MIN is determined by the following specified value:
Ttx_0_lo_dom: Ttx_1_lo_dom + 0.06 × CXPI nominal bit (Tbit) time
Ttx_0_lo_rec: Ttx_1_lo_rec + 0.06 × CXPI nominal bit (Tbit) time
Note that the MAX values are assumed for Ttx_1_lo_*** and the time constant for the entire system is not reflected in the specified value.

Note 2. MAX is calculated based on:
Range of allowable settings: CXPI nominal bit (Tbit) time – 0.06 × CXPI nominal bit (Tbit) time
Time when the condition that the receiving node is at a high level should be detected: 0.06 × CXPI nominal bit (Tbit) time

Note 3. In the master node, the startup timing in synchronization with the internally-generated baud rate counter.
In the slave node, the startup timing in synchronization with the sampling counter starting upon detection of the falling edge of the bus.
Make settings so that the condition CXP1nV1LCNT < CXP1nV0LCNT is satisfied.

The conditions for setting and clearing of the CXP1nV0LW register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.18 CXP1nV0LCNT Access Conditions

Access Conditions			Writing	Reading (Read Value)
CXP1nMODE		CXP1nFLW1. CXP1nCXMD		
CXP1nEN	CXP1nMD			
0_B	x_B	xx_B	Not possible	Initial value
1_B	0_B (CXPI-PWM)	01_B (Bus Sleep)	Enabled	Initial value
	1_B (in CXPI-NRZ mode)	XX_B	Prohibited	Initial value

Note: "Writing not possible" means that writing is ignored.
"Writing disabled" is a limitation to specifications. Not guarded by hardware.

30.4.1.5 CXP1nSAMP — CXPI Sampling Setting Register

This register sets the sampling timing in CXPI-PWM mode.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Access: CXP1nSAMP register can be read or written in 32-bit units.
CXP1nSAMPL register can be read or written 16-bit units.
CXP1nSAMPLL, CXP1nSAMPLH registers can be read or written 8-bit units.

Address: CXP1nSAMP: <CXP1n_base> + 010_H
CXP1nSAMPL: <CXP1n_base> + 010_H
CXP1nSAMPLL: <CXP1n_base> + 010_H,
CXP1nSAMPLH: <CXP1n_base> + 011_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1n STYPE	—	—	—	—	CXP1nSMP										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.19 CXP1nSAMP Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	CXP1nSTYPE.	Sampling timing method setting This bit sets the sampling timing method in CXPI-PWM mode. 0: Offset sampling 1: Absolute position sampling **NOTE:** Writing to this bit is prohibited in CXPI-NRZ mode.
14 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	CXP1nSMP	Sampling timing setting These bits set the sampling timing in CXPI-PWM mode. Range of allowable settings: 001_H to $7FF_H$ **NOTE:** Writing to these bits is prohibited in CXPI-NRZ mode.

CXP1nSTYPE and CXP1nSMP Bits

This register sets the sampling method and sampling timing.

- When CXP1nMODE.CXP1nMD = 0_B (in CXPI-PWM mode)
  - When CXP1nSTYPE = 0 (offset sampling)
    Offset sampling starts from the rising edge of RX (logical value 1).[*1]
    Data is counted from the starting point at divclk and sampled at the timing specified by the CXP1nSMP bit.

Note 1. The starting point is always updated when receiving logical value 1.

– When CXP1nSTYPE = 1 (absolute position sampling)
In absolute position sampling, data is counted starting from the rising edge of RX at divclk and sampled at the timing specified by the CXP1nSMP bit.

- When CXP1nMODE.CXP1nMD = 1_B (in CXPI-NRZ mode)
Writing to this register is prohibited (limitation to specifications).

The table below lists the range of allowable settings.

Table 30.20 Allowable Settings of CXP1nSMP

CXP1nSTYPE	CXP1nSMP	
	MIN.	MAX.
0_B Offset sampling	001_H	(CXP1nV0LCNT - CXP1nV1LCNT) – 1
1_B Absolute position sampling	CXP1nV1LCNT + 1	CXP1nV0LCNT – 1

The conditions for setting and clearing of the CXP1nSAMP register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access (Common to the CXP1nSTYPE and CXP1nSMP Bits)

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.21 CXP1nSMP Access Conditions

Access Conditions			Writing	Reading (Read Value)
CXP1nMODE		CXP1nFLW1. CXP1nCXMD		
CXP1nEN	CXP1nMD			
0_B	x_B	xx_B	Disabled	Initial value
1_B	0_B (CXPI-PWM)	01_B (Bus Sleep)	Enabled	Initial value
	1_B (in CXPI-NRZ mode)	XX_B	Prohibited	Initial value

30.4.1.6 CXP1nFIL — CXPI Input Filter Setting Register

This register sets the surge noise filter time.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Writing to this register is prohibited in CXPI-NRZ mode.

Access: CXP1nFIL register can be read or written in 32-bit units.
CXP1nFILL register can be read or written 16-bit units.
CXP1nFILLL, CXP1nFILLH registers can be read or written 8-bit units.

Address: CXP1nFIL: <CXP1n_base> + 014_H
CXP1nFILL: <CXP1n_base> + 014_H
CXP1nFILLL: <CXP1n_base> + 014_H,
CXP1nFILLH: <CXP1n_base> + 015_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CXP1nSURGE								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.22 CXP1nFIL Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	CXP1nSURGE	Surge noise filter time setting Eliminate the surge not exceeding the filter time (up to 12.775 µs when PCLK = 40 MHz) in CXPI-PWM mode. 000_H: Surge filter OFF 001_H to $1FF_H$: Specifies a surge noise filter time. Setting range: 001_H to $1FF_H$ Surge noise filter time = CXP1nSURGE × PCLK **NOTE:** Writing to these bits is prohibited in CXPI-NRZ mode.

CXP1nSURGE Bits

These bits eliminate the surge not exceeding the filter time (up to 12.775 µs when PCLK = 40 MHz) in CXPI-PWM mode.

(1) Noise filter time isn't supposed to exceed PWM low width and high width.

(2) In the slave node, PWM low width becomes big by noise filter time.

The conditions for setting and clearing of the CXP1nFIL register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.23 CXP1nSURGE Access Conditions

Access Conditions			Writing	Reading (Read Value)
CXP1nMODE		CXP1nFLW1. CXP1nCXMD		
CXP1nEN	CXP1nMD			
0_B	x_B	xx_B	Not possible	Initial value
1_B	0_B (CXPI-PWM)	01_B (Bus Sleep)	Enabled	Setting value
		Other than above	Disabled	Setting value
	1_B (CXPI-NRZ)	xx_B	Disabled	Setting value

Note: "Writing disabled" is a limitation to specifications. Not guarded by hardware.
"Writing not possible" means that writing is ignored.

30.4.1.7 CXP1nWU — CXPI Wakeup Pulse Setting Register

This register sets the timing of the transmission wakeup pulse.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Access: CXP1nWU register can be read or written in 32-bit units.
CXP1nWUL register can be read or written 16-bit units.
CXP1nWULL, CXP1nWULH registers can be read or written 8-bit units.

Address: CXP1nWU: <CXP1n_base> + 018_H
CXP1nWUL: <CXP1n_base> + 018_H
CXP1nWULL: <CXP1n_base> + 018_H,
CXP1nWULH: <CXP1n_base> + 019_H

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CXP1nWTXDOM				CXP1nWTXREC							
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.24 CXP1nWU Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	CXP1nWTXDOM	Wakeup pulse transmission dominant setting The dominant level time of the wakeup pulse from a slave node is set by using the following equation: Wakeup pulse dominant level time = CXPI nominal bit (Tbit) time × CXP1nWTXDOM **NOTE:** Writing to these bits is prohibited for the master node because it cannot transmit a wakeup pulse.
7 to 0	CXP1nWTXREC	Wakeup pulse transmission recessive setting The recessive level time of the wakeup pulse from a slave node is set by using the following equation: Wakeup pulse recessive level interval = CXPI nominal bit (Tbit) time × CXP1nWTXREC **NOTE:** Writing to these bits is prohibited for the master node because it cannot transmit a wakeup pulse.

CXP1nWTXDOM and CXP1nWTXREC Bits

These bits set the dominant width and recessive width of the wakeup pulse.
For details, see **Section 30.5.4.10, Wakeup Pulse Transmission Operation**.

The range of settings is as follows depending on the specified values for wakeup pulse transmission.

Specified dominant width: Ttx_wakeup (min: 250 µs/max: 2500 µs)
Specified recessive width: Ttx_wakeup_space (min: 5 ms/max: 10 ms)

Table 30.25 Allowable Settings of CXP1nWU

CXPI Nominal Bit (Tbit) Frequency	Allowable Settings of CXP1nWTXDOM	Wakeup Pulse Dominant Width by Setting Value	Allowable Settings of CXP1nWTXREC	Wakeup Pulse Recessive Width by Setting Value
20 kbps	5_H to F_H	250 µs to 750 µs	64_H to $C8_H$	5 ms to 10 ms
19.2 kbps	5_H to F_H	260 µs to 781 µs	60_H to $C0_H$	5 ms to 10 ms
10.4 kbps	3_H to F_H	288 µs to 1442 µs	34_H to 68_H	5 ms to 10 ms
9.6 kbps	3_H to F_H	313 µs to 1563 µs	30_H to 60_H	5 ms to 10 ms

The conditions for setting and clearing of the CXP1nWU register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.26 CXP1nWU Access Conditions

Access Conditions			Writing	Reading (Read Value)
CXP1nMODE		CXP1nFLW1. CXP1nCXMD		
CXP1nEN	CXP1nMS			
0_B	x_B	xx_B	Not possible	Initial value
1_B	0_B (slave node)	01_B	Enabled	Setting value
		Other than above	Disabled	Setting value
	1_B (master node)	xx_B	Disabled	Setting value

Note: "Writing disabled" is a limitation to specifications. Not guarded by hardware.
"Writing not possible" means that writing is ignored.

30.4.1.8 CXP1nFRMW — CXP1 Inter-Frame Setting Register

This register sets the CXPI inter-frame timing.

Write access to this register is only enabled in bus sleep mode[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B

Access: CXP1nFRMW register can be read or written in 32-bit units.
CXP1nFRMWL register can be read or written 16-bit units.
CXP1nFRMWLL, CXP1nFRMWLH registers can be read or written 8-bit units.

Address: CXP1nFRMW: <CXP1n_base> + $01C_H$
CXP1nFRMWL: <CXP1n_base> + $01C_H$
CXP1nFRMWLL: <CXP1n_base> + $01C_H$,
CXP1nFRMWLH: <CXP1n_base> + $01D_H$

Value after reset: 0000 1422_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CXP1nIFS					CXP1nIBS2				CXP1nIBS1			
Value after reset	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.27 CXP1nFRMW Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	CXP1nIFS	IFS time setting Set the CXPI nominal bit (Tbit) time for the invalid period between transmission and reception. Setting range: 11_H to $1E_H$ (setting of any other value is prohibited) IFS time: (Value of CXP1nIFS + 1) × nominal bit (Tbit) time
7 to 4	CXP1nIBS2	Inter-transmission frame timing 2 setting Set the CXPI nominal bit (Tbit) time for the interval between response frames. CXPI-PWM mode setting range: 1_H to 9_H (setting of any other value is prohibited) CXPI-NRZ mode setting range: 1_H to 8_H (setting of any other value is prohibited) IBS time: Value of CXP1nIBS2 × nominal bit (Tbit) time
3 to 0	CXP1nIBS1	Inter-transmission frame timing 1 setting Set the CXPI nominal bit (Tbit) time for the interval between the PID frame and response frame transmission. CXPI-PWM mode setting range: 1_H to 9_H (setting of any other value is prohibited) CXPI-NRZ mode setting range: 1_H to 8_H (setting of any other value is prohibited) IBS time: Value of CXP1nIBS1 × nominal bit (Tbit) time

CXP1nIFS Bits

These bits set the invalid period between transmission and reception.

- CXP1nMODE.CXP1nMS = $0/1_B$ (common to the master node and the slave node)
  Sets the CXPI nominal bit (Tbit) time for the invalid period between transmission and reception.
  - Transmission and reception
    The IFS time is counted at the CXPI nominal bit (Tbit) time after CRC frame transmission and reception is completed, and it continues until the counter value matches the setting value of the CXP1nIFS bit + 1.
    The CXPI operation state transitions to the idle state after the IFS period has passed, enabling communications.

CXP1nIBS2 Bits

These bits set the CXPI nominal bit (Tbit) time for the IBS time between response frames.

- CXP1nMODE.CXP1nMS = $0/1_B$ (common to the master node and the slave node)
  - Transmission
    The IBS time is counted at the CXPI nominal bit (Tbit) time after detecting the stop bit of the UART frame, and the next transmission is started upon detection of a match with the setting of the CXP1nIBS2 bit.
  - Reception
    This function does not operate.

NOTE

A data length error occurs if the IBS lasts for 10 Tbits or longer.

CXP1nIBS1 Bits

These bits set the CXPI nominal bit (Tbit) time for the interval between the PID frame and response frame transmission.

- CXP1nMODE.CXP1nMS = $0/1_B$ (common to the master node and the slave node)
  Sets the CXPI nominal bit (Tbit) time for the interval between the PID frame and response frame transmission.
  - Transmission
    The IBS time is counted at the CXPI nominal bit (Tbit) time after detecting the stop bit of the UART frame, and the next transmission is started upon detection of a match with the setting of the CXP1nIBS1 bit.
  - Reception
    This function does not operate.

NOTE

A data length error occurs if the IBS lasts for 10 Tbits or longer.

The conditions for setting and clearing of the CXP1nFRMW register are as follows:

[HW setting condition]

- None

[HW clearing condition]

– CXP1nMODE.CXP1nEN = 0_B (forced reset)

Not allowed setting of 0_H.

30.4.1.9 CXP1nSLP — CXPI Sleep Setting Register

This register sets the time for transition to sleep mode.

Write access to this register is only enabled in bus sleep mode and slave node[*1]. It is disabled in other modes.

Note 1. CXP1nFLW1.CXP1nCXMD = 01_B & CXP1nMODE.CXP1nMS = 0_B

Access: CXP1nSLP register can be read or written in 32-bit units.
CXP1nSLPL register can be read or written 16-bit units.
CXP1nSLPLL, CXP1nSLPLH registers can be read or written 8-bit units.

Address: CXP1nSLP: <CXP1n_base> + 020_H
CXP1nSLPL: <CXP1n_base> + 020_H
CXP1nSLPLL: <CXP1n_base> + 020_H,
CXP1nSLPLH: <CXP1n_base> + 021_H

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CXP1nSLPWAIT									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.28 CXP1nSLP Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	CXP1nSLPWAIT	Sleep transition time setting Sleep transition time = CXP1nSLPWAIT × CXPI nominal bit (Tbit) time

CXP1nSLPWAIT Bits

These bits set the sleep transition time from the reception of the sleep frame.
For details, see **Section 30.5.4.9, Sleep Frame Reception Operation**.

- When CXP1nMODE.CXP1nMS = 0 (slave node)
  Counting starts after the sleep frame has been transmitted and proceeds at the CXPI nominal bit (Tbit) time. A sleep transition completion interrupt is generated when the counter value matches the setting value of the CXP1nSLPWAIT bit.

  Sleep transition time = CXP1nSLPWAIT × CXPI nominal bit (Tbit) time

The range of settings of these bits is as follows depending on the specified values of the sleep transition time.

Table 30.29 Allowable Settings of CXP1nSLPWAIT (in Slave Mode)

CXPI Nominal Bit (Tbit) Frequency	Allowable Settings of CXP1nSLPWAIT	Sleep Transition Completion Time by Setting Value
20 kbps	$1F4_H$ to $3E8_H$	25 ms to 50 ms
19.2 kbps	$1E0_H$ to $3C0_H$	25 ms to 50 ms
10.4 kbps	104_H to 208_H	25 ms to 50 ms
9.6 kbps	$0F0_H$ to $1E0_H$	25 ms to 50 ms

The conditions for setting and clearing of the CXP1nSLP register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.10 CXP1nREP — CXPI Retransmission Setting Register

This register sets the CXPI retransmission.

This register is only valid while CXP1nMODE.CXP1nREPOP = 1.

Access: CXP1nREP register can be read or written in 32-bit units.
CXP1nREPL register can be read or written 16-bit units.
CXP1nREPLL register can be read or written 8-bit units.

Address: CXP1nREP: <CXP1n_base> + 024_H
CXP1nREPL: <CXP1n_base> + 024_H
CXP1nREPLL: <CXP1n_base> + 024_H

Value after reset: 0000 $00FA_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1nAIREP		CXP1nARREP		CXP1nEIREP		CXP1nERREP	
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.30 CXP1nREP Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7, 6	CXP1nAIREP	Loss-of-arbitration retransmission count setting 2 (CXP1nSND.CXP1nREPSEL = 10_B) Set the number of retransmissions when arbitration is lost. 00_B: No retransmission 01_B: One round of retransmission 10_B: Two rounds of retransmission 11_B: Unlimited
5, 4	CXP1nARREP	Loss-of-arbitration retransmission count setting 1 (CXP1nSND.CXP1nREPSEL = 01_B) Set the number of retransmissions when arbitration is lost. 00_B: No retransmission 01_B: One round of retransmission 10_B: Two rounds of retransmission 11_B: Unlimited
3, 2	CXP1nEIREP	Error retransmission count setting 2 (CXP1nSND.CXP1nREPSEL = 10_B) Set the number of retransmissions when an error is detected. 00_B: No retransmission 01_B: One round of retransmission 10_B: Two rounds of retransmission 11_B: Unlimited
1, 0	CXP1nERREP	Error retransmission count setting 1 (CXP1nSND.CXP1nREPSEL = 01_B) Set the number of retransmissions when an error is detected. 00_B: No retransmission 01_B: One round of retransmission 10_B: Two rounds of retransmission 11_B: Unlimited

CXP1nAIREP Bits

The setting value of these bits is reflected in the CXP1nFLG2.CXP1nAREP bit.
For details, see **Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**.

CXP1nARREP Bits

The setting value of these bits is reflected in the CXP1nFLG2.CXP1nAREP bit.
For details, see **Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**.

CXP1nEIREP Bits

The setting value of these bits is reflected in the CXP1nFLG2.CXP1nEREP bit.
For details, see **Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**.

CXP1nEREP Bits

The setting value of these bits is reflected in the CXP1nFLG2.CXP1nEREP bit.
For details, see **Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**.

The conditions for setting and clearing of the CXP1nREP register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.11 CXP1nWUP — CXPI Wakeup Control Register

This register sets the CXPI wakeup request.

Access: CXP1nWUP register can be read or written in 32-bit units.
CXP1nWUPL register can be read or written 16-bit units.
CXP1nWUPLL, CXP1nWUPLH registers can be read or written 8-bit units.

Address: CXP1nWUP: <CXP1n_base> + 028_H
CXP1nWUPL: <CXP1n_base> + 028_H
CXP1nWUPLL: <CXP1n_base> + 028_H,
CXP1nWUPLH: <CXP1n_base> + 029_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1n STCLR	—	—	—	—	—	—	—	—	—	CXP1n WUP REQ	—	—	—	CXP1n CLK Off	CXP1n CLK On
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W

Table 30.31 CXP1nWUP Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	CXP1nSTCLR	CXPI state clear request flag 0: — (Writing 0 is ignored) 1: CXPI state clear (simple reset) The read value is 0.
14 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CXP1nWUPREQ	Wakeup pulse transmission request flag This bit issues a wakeup pulse transmission request in a slave node. 0: — (Writing 0 is ignored) 1: Wakeup pulse transmission The read value is 0.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CXP1nCLKOFF	CXPI clock stop request flag This bit issues a CXPI clock stop request or notification of non-detection of the CXPI clock. • In CXPI-PWM mode 0: — (Writing 0 is ignored) 1: CXPI clock output stop The read value is 0. • In CXPI-NRZ mode 0: — (Writing 0 is ignored) 1: Notification of non-detection of the CXPI clock requested The read value is 0.

Table 30.31 CXP1nWUP Register Contents (2/2)

Bit Position	Bit Name	Function
0	CXP1nCLKON	CXPI clock output request flag This bit issues a CXPI clock output or CXPI clock detection notification. • CXPI-PWM mode 0: — (Writing 0 is ignored) 1: CXPI clock output The read value is 0. • CXPI-NRZ mode 0: — (Writing 0 is ignored) 1: CXPI clock detection notification request The read value is 0.

CXP1nSTCLR Bit

Writing 1 to this bit issues a simple reset.
CXPI state clear (write 1 to CXP1nWUP.CXP1nSTCLR) is prohibited in offset sampling PWM mode (CXP1nSTYPE = 0_B and CXP1nMD = 1_B).
For details, see **Section 30.5.8.7, Software Reset**.

If the application found a physical bus error, a simple reset is issued to restore the communication when CXP1nSFRI.CXP1nSSLP = 0_B (sleep prohibited).

A simple reset initializes the following:

- CXPI transmission control register (CXP1nSND)
- CXPI flag register 2 (CXP1nFLG2)
- CXPI flow register 1 (CXP1nFLW1) (CXP1nFLW1.CXP1nCXMD is excluded)
- CXPI flow register 2 (CXP1nFLW2)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.32 CXP1nSTCLR Bit Access Conditions

Access Conditions			
CXP1nMODE		CXP1nSFRI. CXP1nSSLP	Writing of 1
CXP1nEN	CXP1nMS		
0_B	x_B	x_B	Not possible
1_B	0_B	x_B	Disabled
	1_B	0_B	Enabled
		1_B	Disabled

Note: "Writing not possible" means that writing is ignored.
"Writing disabled" is a limitation to specifications. (Not guarded by hardware.)
The read value is always 0.

CXP1nWUPREQ Bit

This bit issues a wakeup pulse transmission request in a slave node.
When 1 is written to this bit, the CXPI transmits a wakeup pulse and enters send wakeup mode.
For details, see **Section 30.5.4.10, Wakeup Pulse Transmission Operation**.

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.33 CXP1nWUPREQ Bit Access Conditions

Access Condition			
CXP1nMODE		CXP1nFLG1. CXP1nCLKDET	Writing of 1
CXP1nEN	CXP1nMS		
0_B	x_B	x_B	Not possible
1_B	0_B Slave	0_B CXPI clock not detected	Enabled
		1_B CXPI clock detected	Not possible
	1_B Master	x_B	Disable

CXP1nCLKOFF bit

Writing 1 to this bit stops the CXPI clock (CXPI-PWM mode), or issues a notification of non-detection of the CXPI clock (CXPI-NRZ mode).

1. CXP1nMODE.CXP1nMD = 0_B (CXPI-PWM mode)

- CXP1nMODE.CXP1nMS = 1_B (master node)
  Writing 1 to this bit stops the CXPI clock output.
  The CXPI enters bus sleep mode after detecting that the CXPI clock has been stopped (CXP1nFLG1.CXP1nCLKDET = 0_B).
  - CXPI clock stop timing
    When the transmission pin level is low: The CXPI clock is stopped after a low level has been output.
    When the transmission pin level is high: The CXPI clock is stopped immediately.
- CXP1nMODE.CXP1nMS = 0_B (slave node)
  Writing to this bit is prohibited (limitation to specifications).

2. CXP1nMODE.CXP1nMD =1_B (CXPI-NRZ mode)

- CXP1nMODE.CXP1nMS = 1_B (master node) / When CXP1nMODE.CXP1nMS = 0_B (slave node)
  Writing 1 to this bit transitions the CXPI to bus sleep mode.

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.34 CXP1nCLKOFF Bit Access Conditions (1/2)

Access Conditions					
CXP1nMODE			CXP1nFLW1.		
CXP1nEN	CXP1nMD	CXP1nMS	CXP1nCXMD	CXP1nCXST	Writing of 1
0_B	x_B	x_B	xx_B	xxx_B	Not possible

Table 30.34 CXP1nCLKOFF Bit Access Conditions (2/2)

Access Conditions					
CXP1nMODE			CXP1nFLW1.		
CXP1nEN	CXP1nMD	CXP1nMS	CXP1nCXMD	CXP1nCXST	Writing of 1
1_B	0_B (CXPI-PWM)	0_B Slave	xx_B	xxx_B	Disabled
		1_B Master	11_B Normal	010_B Transmission/ reception stopped (IDLE)	Enabled
				Other than above	Disabled
			Other than above	xxx_B	Disabled
	1_B (CXPI-NRZ)	$0_B/1_B$ Slave/Master	11_B Normal	010_B Transmission/ reception stopped (IDLE)	Enabled
				Other than above	Disabled
			Other than above	xxx_B	Disabled

Note: "Writing not possible" means that writing is ignored.
"Writing disabled" is a limitation to specifications. Not guarded by hardware.
The read value is 0.

CXP1nCLKON Bit

Writing 1 to this bit in bus sleep mode outputs the CXPI clock (CXPI-PWM mode), or issues a notification of detection of the CXPI clock (CXPI-NRZ mode).

1. CXP1nMODE.CXP1nMD = 0_B (CXPI-PWM mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node)
  Writing 1 to this bit outputs the CXPI clock.
  The CXPI enters normal mode after detecting the CXPI clock.
- When CXP1nMODE.CXP1nMS = 0_B (slave node)
  Writing to this bit is disabled.

2. CXP1nMODE.CXP1nMD = 1_B (CXPI-NRZ mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)
  Writing 1 to this bit leads to transition to normal mode.

NOTE

Setting of the CXP1nCLKON and CXP1nCLKOFF bits should be mutually exclusive.
If 1_B is set in both bits at the same time, writing to these bits is ignored and the output request is not issued.

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.35 CXP1nCLKON Bit Access Conditions

Access Conditions					
CXP1nMODE			CXP1nFLW1. CXP1nCXMD	CXP1nCLKOFF. Write Value	Writing of 1
CXP1nEN	CXP1nMD	CXP1nMS			
0_B	x_B	x_B	xx_B	x_B	Not possible
1_B	0_B (CXPI-PWM)	0_B Slave	xx_B	x_B	Disabled
		1_B Master	01_B Bus Sleep	0_B	Enabled
				1_B	Not possible
			Other than above	x_B	Not possible
1_B	1_B (CXPI-NRZ)	$0_B/1_B$ Slave/Master	11_B Normal	0_B	Enabled
				1_B	Not possible
			Other than above	x_B	Disabled

Note: "Writing not possible" means that writing is ignored.
"Writing disabled" is a limitation to specifications. Not guarded by hardware.
The read value is always 0.

30.4.1.12 CXP1nSND — CXPI Transmission Control Register

This register sets the CXPI transmission request.

Access: CXP1nSND register can be read or written in 32-bit units.
CXP1nSNDL register can be read or written 16-bit units.
CXP1nSNDLL register can be read or written 8-bit units.

Address: CXP1nSND: <CXP1n_base> + $02C_H$
CXP1nSNDL: <CXP1n_base> + $02C_H$
CXP1nSNDLL: <CXP1n_base> + $02C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CXP1n REPSEL		CXP1n SPT REQ	CXP1n SRS REQ	CXP1n SFR REQ	CXP1n SID REQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.36 CXP1nSND Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	CXP1nREPSEL	Retransmission setting flag Specify the setting for retransmission when PID transmission or frame transmission is requested. 00_B: No retransmission 01_B: Setting 1 10_B: Setting 2 11_B: Setting prohibited
3	CXP1nSPTREQ	PTYPE transmission request flag Requests transmission of PTYPE in polling mode. 0: No PTYPE transmission request (transmission canceled) 1: PTYPE transmission requested
2	CXP1nSRSREQ	Response transmission request flag 0: — (Writing 0 is ignored) 1: Response transmission requested The read value is 0.
1	CXP1nSFRREQ	Frame transmission request flag Requests frame transmission. 0: No frame transmission request (transmission canceled) 1: Frame transmission requested
0	CXP1nSIDREQ	PID transmission request Requests PID transmission. 0: No PID transmission request (transmission canceled) 1: PID transmission requested

CXP1nREPSEL Bit

These bits specify the setting for retransmission when PID transmission or frame transmission is requested.
These bits are only valid while the CXP1nMODE.CXP1nREPOP bit is set to 1_B.

[HW setting condition]

- None

[HW clearing conditions]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)
- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B

CXP1nSPTREQ Bit

Requests transmission of PTYPE in polling mode.

Writing 1 to this bit triggers the CXPI to transmit PTYPE when the following condition is satisfied. (This bit is cleared at the start of transmission.)

1. CXP1nMODE.CXP1nACC = 1_B (polling mode)

- CXP1nMODE.CXP1nMS = 1_B (master node)
  The PTYPE transmission can be canceled by writing 0 to this bit when the condition is not satisfied.

  After PTYPE has been transmitted, the CXPI enters the state of waiting for setting response judgment/response data (CXP1nFLW1.CXP1nCXST = 100_B).

[HW setting condition]

- None

[HW clearing conditions]

- PTYPE transmission is started.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)
- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B (simple reset)

The table below lists the data transmitted in PTYPE transmission.

Table 30.37 Configuration of PTYPE

Parity Bit	Frame TYPE
1 bit (odd parity bit) Fixed to 1_B	7 bits (fixed to 00_H)

2. Mode other than above

Writing to this bit is disabled. Writing to this bit is ignored.

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.38 CXP1nSPTREQ Access Conditions

Access Conditions			Writing	Reading (Read Value)
CXP1nMODE				
CXP1nEN	CXP1nACC	CXP1nMS		
0_B	x_B	x_B	Not possible	Initial value
1_B	0_B Event trigger	x_B	Not possible	Initial value
	1_B Polling	1_B Master	Enabled*1	State
		0_B Slave	Not possible	Initial value

Note 1. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.

CXP1nSRSREQ bit

This bit initiates data transmission of the response field when the received PID is judged to be a response.

When 1_B is written to this bit while CXP1nFLW1.CXST = 100_B (state of waiting for setting response judgment/response data), the CXPI transmits data in the response field.

Data transmission is not performed if 1 is written to this bit while CXP1nFLW1.CXST is not 100_B.

Table 30.39 Data for Response of a Normal Frame

Frame Information CXP1nSFRI	Frame Information (Extension DLC) CXP1nSFRI	Data CXP1nSDATAm	CRC CXP1nSCRC
1 byte	—	0 to 12 bytes	1 byte

Table 30.40 Data for Response of a Long Frame

Frame Information CXP1nSFRI	Frame Information (Extension DLC) CXP1nSFRI	Data CXP1nSDATAm*1	CRC CXP1nSCRC
1 byte	1 byte	0 to 255 bytes (Extension DLC)	2 bytes

Note 1. Transfer proceeds while updating the data for transmission in units of 8 bytes by using the transmission data setting request interrupt.

CXP1nSFRREQ Bit

When 1 is written to this bit, the CXPI performs data transmission*2 when the following conditions are satisfied.

Note 2. For details, see **Section 30.5.7.1, Transmission**.

1. CXP1nMODE.CXP1nACC = 0_B (event trigger method)

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)
  - CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))

2. CXP1nMODE.CXP1nACC = 1_B (polling method)

- When CXP1nMODE.CXP1nMS = 1_B (master node)

– CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))

- When CXP1nMODE.CXP1nMS = 0_B (slave node)
  – CXP1nFLW1.CXP1nSPIDEN = 1_B (slave transmit allowed) and CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))

Data for Transmission

Table 30.41 Frame Structure of a Normal Frame

PID Field CXP1nSPID	Frame Information CXP1nSFRI	Data CXP1nSDATAm	CRC CXP1nSCRC
1 byte	1 byte	0 to 12 bytes	1 byte

Table 30.42 Frame Structure of a Long Frame

PID Field CXP1nSPID	Frame Information CXP1nSFRI	Frame Information (Extension DLC) CXP1nSFRI	Data CXP1nSDATAm*1	CRC CXP1nSCRC
1 byte	1 byte	1 byte	0 to 255 bytes	2 bytes

Note 1. Transfer proceeds while updating the data for transmission in units of 8 bytes by using the transmission data setting request interrupt.

Transmission is canceled by writing 0 to this bit.
The conditions for cancellation are as follows:

- When CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  Transmission can be canceled before transmission starts.
  Retransmission can be canceled during transmission.
- When CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)
  Transmission can be canceled before transmission starts.

[HW setting condition]

- None

[HW clearing conditions]

- A fatal error occurred.
- Transmission started (CXP1nMODE.CXP1nREPOP = 1 (automatic retransmission judged))
- Normal transmission has completed (CXP1nMODE.CXP1nREPOP = 0 (no automatic retransmission judgment))
- CXP1nWUP.CXP1nSTCLR = 1_B
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.43 CXP1nSFRREQ Bit Access Conditions

Access Conditions				Writing	Reading (Read Value)
		CXP1nSND			
CXP1nMODE.CXP1nEN	CXP1nFLG1.CXP1nFAIL	CXP1nSPTREQ Write Value	CXP1nSIDREQ Write Value		
0_B	x_B	x_B	x_B	Not possible	Initial value
1_B	0_B	0_B	0_B	Enabled*1	State
			1_B	Not possible	Initial value
		1_B	x_B	Not possible	Initial value
	1_B	x_B	x_B	Not possible	Initial value

Note 1. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.

CXP1nSIDREQ Bit

When 1 is written to this bit, the CXPI performs data transmission when the following conditions are satisfied.

1. CXP1nMODE.CXP1nACC = 0_B (event trigger method)

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)
  - CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))

2. CXP1nMODE.CXP1nACC = 1_B (polling method)

- CXP1nMODE.CXP1nMS = 1_B (master node)
  - CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))
- CXP1nMODE.CXP1nMS = 0_B (slave node)
  - CXP1nFLW1.CXP1nSPIDEN = 1_B (slave transmit allowed) and CXP1nFLW1.CXP1nCXST = 010_B (transmission/reception stopped (IDLE))

Data for transmission

Table 30.44 PID Field

PID Field CXP1nSPID
1 byte

Transmission is canceled by writing 0 to this bit.
The conditions for cancellation are as follows:

- When CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  - Transmission can be canceled before transmission starts.
  - Retransmission can be canceled during transmission.
- When CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)
  - Transmission can be canceled before transmission starts.

[HW setting condition]

- None

[HW clearing conditions]

- A fatal error occurred.
- Transmission started (CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged))
- Normal transmission has completed (CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment))
- CXP1nWUP.CXP1nSTCLR = 1_B
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.45 CXP1nSIDREQ Bit Access Conditions

Access Conditions			Writing	Reading (Read Value)
		CXP1nSND		
CXP1nMODE.CXP1nEN	CXP1nFLG1.CXP1nFAIL	CXP1nSPTREQ. Write Value		
0_B	x_B	x_B	Not possible	Initial value
1_B	0_B	0_B	Enabled*1	State
		1_B	Not possible	Initial value
	1_B	x_B	Not possible	Initial value

Note 1. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.

30.4.1.13 CXP1nIMSK — CXPI Interrupt Mask Register

This register sets masking of the CXPI interrupt. It masks the interrupt request to the CPU.

Interrupt sources are reflected in the CXP1nINT register even if interrupts are masked in this register.

Access: CXP1nIMSK register can be read or written in 32-bit units.
CXP1nIMSKL register can be read or written 16-bit units.
CXP1nIMSKLL, CXP1nIMSKLH registers can be read or written 8-bit units.

Address: CXP1nIMSK: <CXP1n_base> + 030_H
CXP1nIMSKL: <CXP1n_base> + 030_H
CXP1nIMSKLL: <CXP1n_base> + 030_H,
CXP1nIMSKLH: <CXP1n_base> + 031_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1n ERR MSK	CXP1n SLP MSK	CXP1n REX MSK	CXP1n RFQ MSK	CXP1n RPT MSK	—	CXP1n RFR MSK	CXP1n RID MSK	—	—	—	CXP1n SFQ MSK	CXP1n SPT MSK	—	CXP1n SFR MSK	CXP1n SID MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Table 30.46 CXP1nIMSK Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	CXP1nERRMSK	Error detection interrupt mask setting 0: Enables interrupts 1: Disables interrupts
14	CXP1nSLPMSK	Sleep transition completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
13	CXP1nREXMSK	PTYPE transmission completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
12	CXP1nRFQMSK	Received data save request interrupt mask setting 0: Enables interrupts 1: Disables interrupts
11	CXP1nRPTMSK	PTYPE reception completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	CXP1nRFRMSK	Frame reception completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
8	CXP1nRIDMSK	PID reception completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 30.46 CXP1nIMSK Register Contents (2/2)

Bit Position	Bit Name	Function
4	CXP1nSFQMSK	Transmission data setting request interrupt mask setting 0: Enables interrupts 1: Disables interrupts
3	CXP1nSPTMSK	PTYPE transmission completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CXP1nSFRMSK	Frame transmission completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts
0	CXP1nSIDMSK	PID transmission completion interrupt mask setting 0: Enables interrupts 1: Disables interrupts

For details on the CXP1nIMSK register, see **Section 30.5.9, Interrupts**.

The conditions for setting and clearing of the CXP1nIMSK register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.14 CXP1nINT — CXPI Interrupt Source Register

This is a status register that indicates interrupt sources.

Access: CXP1nINT register can be read or written in 32-bit units.
CXP1nINTL register can be read or written 16-bit units.
CXP1nINTLL, CXP1nINTLH registers can be read or written 8-bit units.

Address: CXP1nINT: <CXP1n_base> + 034_H
CXP1nINTL: <CXP1n_base> + 034_H
CXP1nINTLL: <CXP1n_base> + 034_H,
CXP1nINTLH: <CXP1n_base> + 035_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1n ERR INT	CXP1n SLP INT	CXP1n REX INT	CXP1n RFQ INT	CXP1n RPT INT	—	CXP1n RFR INT	CXP1n RID INT	—	—	—	CXP1n SFQ INT	CXP1n SPT INT	—	CXP1n SFR INT	CXP1n SID INT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Table 30.47 CXP1nINT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	CXP1nERRINT	Error detection interrupt source flag Indicates that an error detection interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
14	CXP1nSLPINT	Sleep transition completion interrupt source flag Indicates that a sleep transition completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
13	CXP1nREXINT	PTYPE valid period completion interrupt source flag Indicates that a PTYPE valid period completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
12	CXP1nRFQINT	Received data save request interrupt source flag Indicates that a received data save request interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
11	CXP1nRPTINT	PTYPE reception completion interrupt source flag Indicates that a PTYPE reception completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	CXP1nRFRINT	Frame reception completion interrupt source flag Indicates that a frame reception completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated

Table 30.47 CXP1nINT Register Contents (2/2)

Bit Position	Bit Name	Function
8	CXP1nRIDINT	PID reception completion interrupt source flag Indicates that a PID reception completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CXP1nSFQINT	Transmission data setting request interrupt source flag Indicates that a transmission data setting request interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
3	CXP1nSPTINT	PTYPE transmission completion interrupt source flag Indicates that a PTYPE transmission completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CXP1nSFRINT	Frame transmission completion interrupt source flag Indicates that a frame transmission completion interrupt request was generated. 0: No interrupt source 1: Interrupt source generated
0	CXP1nSIDINT	PID transmission completion interrupt source flag Indicates that a PID transmission completion interrupt request was generated. 0: No interrupt source 1: Interrupt source is generated

CXP1nERRINT Bit

This bit indicates whether or not an error interrupt is generated.
For details, see **Section 30.5.9.3(1), Error Detection**.

[HW setting conditions]

An error occurred (One of CXP1nERR[7:0] is 1).
A fatal error occurred (CXP1nFTERR = 1_B).
A state error occurred (CXP1nSTERR = 1_B).

[HW clearing conditions]

The error register was cleared (CXP1nERR[7:0] = 00_H).
CXP1nMODE.CXP1nEN = 0_B (forced reset)

The CXP1nSLPINT Bit

This bit indicates whether or not a sleep transition completion interrupt is generated. Process flow should be such that it is cleared by software when interrupt occurs.
For details, see **Section 30.5.9.3(2), Sleep Transition Completion**.

- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- The sleep transition time[*1] has passed after a sleep frame has been received.

Note 1. Sleep transition time = CXP1nSLPWAIT × CXPI nominal bit (Tbit) time

[HW clearing conditions]

- The CXPI makes the transition from bus sleep mode to normal mode (Other than Send Wakeup mode to Normal mode)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nREXINT Bit

This bit indicates that the PTYPE valid period has passed.
For details, see **Section 30.5.9.2(5), PTYPE Valid Period Completion**.

1. When CXP1nMODE.CXP1nACC = 0_B (event trigger mode)
   This interrupt is not generated.
2. When CXP1nMODE.CXP1nACC = 1_B (polling mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node)

[HW setting condition]

- PID reception has been accepted in response to PTYPE transmission.

[HW clearing conditions]

- PTYPE transmission is completed.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- PID transmission/reception has been accepted in response to PTYPE reception.

[HW clearing conditions]

- PTYPE reception is completed.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRFQINT Bit

This bit indicates whether or not a received data save request interrupt is generated.
For details, see **Section 30.5.9.2(3), Received Data Save Request**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- Eight bytes of data have been received in a long frame.

[HW clearing conditions]

- The CXPI started to transmit PID, or received PID.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRPTINT Bit

This bit indicates whether or not a PTYPE reception completion interrupt is generated.
For details, see **Section 30.5.9.2(1), PTYPE Reception Completion**.

1. When CXP1nMODE.CXP1nACC = 0_B (event trigger mode)
   This interrupt is not generated.
2. When CXP1nMODE.CXP1nACC = 1_B (polling mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node)
  This interrupt is not generated.
- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- PTYPE reception is completed.

[HW clearing conditions]

- The CXPI started to receive the PID.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRFRINT Bit

This bit indicates whether or not a frame reception completion interrupt is generated.
For details, see **Section 30.5.9.2(4), Frame Reception Completion**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- A frame has been received.

[HW clearing conditions]

- The CXPI started to transmit PID, or received PID.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRIDINT Bit

This bit indicates whether or not a PID reception completion interrupt is generated.
For details, see **Section 30.5.9.2(2), PID Reception Completion**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- PID has been received.

[HW clearing conditions]

- The CXPI started to receive the ID.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSFQINT Bit

This bit indicates whether or not a transmission data setting request interrupt is generated.
For details, see **Section 30.5.9.1(3), Transmission Data Setting Request**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- Eight bytes of data have been transmitted in a long frame.

[HW clearing conditions]

- The CXPI started to transmit PID, or received PID.
  CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSPTINT Bit

This bit indicates whether or not a PTYPE transmission completion interrupt is generated.
For details, see **Section 30.5.9.1(1), PTYPE Transmission Completion**.

1. When CXP1nMODE.CXP1nACC = 1_B (polling mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node)

[HW setting condition]

- PTYPE transmission is completed.

[HW clearing conditions]

- PTYPE transmission is started.
  CXP1nMODE.CXP1nEN = 0_B (forced reset)

- When CXP1nMODE.CXP1nMS = 0_B (slave node)
  This interrupt is not generated.

2. When CXP1nMODE.CXP1nACC = 0_B (event trigger mode)
   This interrupt is not generated.

CXP1nSFRINT Bit

This bit indicates whether or not a frame transmission completion interrupt is generated.
For details, see **Section 30.5.9.1(4), Frame Transmission Completion**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting conditions]

- Frame transmission is completed.

[HW clearing condition]

- The CXPI started to transmit PID
- PID received PID (CXP1nINT.CXP1nRIDINT=1_B & CXP1nFLW1.CXP1nCXST=110_B).
  However, exclude cases where received PID does not match tag.
  However, excluding the case of automatic response target.
- CXP1nMODE.CXP1nEN = 0_B (forced reset).

CXP1nSIDINT Bit

This bit indicates whether or not a PID transmission completion interrupt is generated.
For details, see **Section 30.5.9.1(2), PID Transmission Completion**.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- CXP1nFLW1.CXP1nCXST = 101_B (PID transmission) and reception is completed.
- The transmitted data and received data match in PID transmission loopback.

NOTE

This bit is not set if a bit error or framing error occurs. The PID includes a sleep frame.

[HW clearing conditions]

- PID transmission is started.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.15 CXP1nFLG1 — CXPI Flag Register 1

This register is a status register that indicates the state of the CXPI.

Access: CXP1nFLG1 register can be read or written in 32-bit units.
CXP1nFLG1L register can be read or written 16-bit units.
CXP1nFLG1LL register can be read or written 8-bit units.

Address: CXP1nFLG1: <CXP1n_base> + 038_H
CXP1nFLG1L: <CXP1n_base> + 038_H
CXP1nFLG1LL: <CXP1n_base> + 038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1n FAIL	CXP1n FTERR	CXP1n STERR	—	—	CXP1n RFT FID	CXP1n CLK DET	CXP1n EXREQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R	R/W

Table 30.48 CXP1nFLG1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	CXP1nFAIL	Failure management flag 0: Normal state 1: Transmission prohibited
6	CXP1nFTERR	Fatal error status 0: No error 1: An error occurred
5	CXP1nSTERR	State error status 0: No error 1: An error occurred
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CXP1nRFTFID	Frame TYPE/frame ID reception flag 0: Not received 1: Received
1	CXP1nCLKDET	CXPI clock detection status 0: CXPI clock not detected 1: CXPI clock detected
0	CXP1nEXREQ	External request flag 0: Internal request transfer 1: External request transfer Only 0 can be written to this bit. Writing 1 to this bit is ignored.

CXP1nFAIL Bit

Writing 0 to this bit clears the CXP1nERR.CXP1nERRCNT bit at the same time.

Writing 1 to this bit prohibits writing to CXP1nSND.CXP1nSIDREQ (PID transmission request) and CXP1nSND.CXP1nSFRREQ (frame transmission request) and CXP1nSND.CXP1nSPTREQ (PTYPE transmission request) and CXP1nSND.CXP1nSRSREQ (response request) to prohibit transmission requests.
For details, see **Section 30.5.7.6, Failure Management Function**.

- When CXP1nMODE.CXP1nMS = 1_B (master node)

[HW setting condition]

- An error[*1] is detected when CXP1nERR.CXP1nERRCNT > $F7_H$.

[HW clearing conditions]

- CXPI clock is detected (CXP1nFLG1.CXP1nCLKDET is set) when CXP1nFLW1.CXP1nCXMD = 01 (Bus Sleep mode)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- An error[*1] is detected when CXP1nERR.CXP1nERRCNT > $F7_H$.

[HW clearing conditions]

- 1 is written to CXP1nWUP.CXP1nWUPREQ (Wakeup pulse transmission request flag)
- The CXPI clock is detected when CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep).

Note 1. The following errors apply:
CXP1nERR.CXP1nBITERR (bit error)
CXP1nDLNERR (data length error)
CXP1nFRMERR (framing error)

CXP1nFTERR Bit

This bit indicates whether or not a fatal error occurred.
A fatal error is caused by a mechanical breakdown.
For details, see **Section 30.5.7.4(10), Fatal Error (Not Defined in CXPI Specifications)**.

[HW setting condition]

- A mechanical breakdown occurred.

[HW clearing conditions]

- Transmission is started. (PID or Frame or Sleep or PTYPE) (including retransmission).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSTERR Bit

This bit indicates whether or not a state error occurred.
For details, see **Section 30.5.7.4(9), State Error (Not Defined in CXPI Specifications)**.

1. When CXP1nMODE.CXP1nACC = 0_B (event trigger mode)

- PTYPE is detected.

2. When CXP1nMODE.CXP1nMS = 1_B (master node)

- PTYPE is detected.

3. When CXP1nMODE.CXP1nACC = 1_B (polling mode) and CXP1nMODE.CXP1nMS = 0_B (slave node)

- When the IP in a slave node starts PID transmission because it receives a PTYPE field in polling mode, it receives the PTYPE field again and loses an arbitration.

[HW clearing conditions]

- Transmission is started. (PID or Frame or Sleep or PTYPE) (including retransmission)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRFTFID Bit

This bit is a status bit indicating whether or not the frame TYPE/frame ID has been received.
The frame TYPE/frame ID is judged to have been received even if a parity error occurs.
For details, see **Section 30.5.8.8, Status**.

[HW setting condition]

- Frame TYPE or frame ID is received (Send loop-back is included)

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nCLKDET Bit

This bit indicates whether or not the clock has been detected.

1. When CXP1nMODE.CXP1nMD = 0_B (CXPI-PWM mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- The CXPI clock was detected.
  After detecting a wakeup pulse, the CXPI detects the CXPI clock when detecting the rising edge four or five times on the receive pin.
  However, it initializes the falling edge counter if the CXPI nominal bit (Tbit) frequency is slower than the frequency set by CXP1nBRT.

[HW clearing conditions]

- The CXPI clock was not detected.
  If the CXPI nominal bit (Tbit) frequency is slower than the frequency set by CXP1nBRT, the CXPI clock is determined not to have been detected.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)
- Bus sleep mode transition is completed (transition by a sleep frame).

2. When CXP1nMODE.CXP1nMD = 1_B (CXPI-NRZ mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- CXP1nWUP.CXP1nCLKON = 1_B is written.

[HW clearing conditions]

- CXP1nWUP.CXP1nCLKOFF = 1_B is written.
- Bus sleep mode transition is completed (transition by a sleep frame).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nEXREQ Bit

This bit indicates an external request or internal request.

- When CXP1nMODE.CXP1nMS = 1_B (master node) or CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- An externally requested transfer is completed. (termination by error is included)

[HW clearing conditions]

- An internally requested transmission is started. (PID or Frame or Sleep or PTYPE) (including retransmission)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.16 CXP1nFLG2 — CXPI Flag Register 2

This register is a status register that indicates the status of the CXPI.

Access: CXP1nFLG2 register can be read or written in 32-bit units.
CXP1nFLG2L register can be read or written 16-bit units.
CXP1nFLG2LL register can be read or written 8-bit units.

Address: CXP1nFLG2: <CXP1n_base> + $03C_H$
CXP1nFLG2L: <CXP1n_base> + $03C_H$
CXP1nFLG2LL: <CXP1n_base> + $03C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CXP1n SRSV	CXP1n RRSV	CXP1n AREP		CXP1n EREP	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.49 CXP1nFLG2 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CXP1nSRSV	Frame retransmission reservation flag*1 Indicates whether or not frame retransmission is reserved. 0: No retransmission reservation (retransmission canceled) 1: Retransmission is reserved
4	CXP1nRRSV	PID retransmission reservation flag*1 Indicates whether or not PID retransmission is reserved. 0: No retransmission reservation (retransmission canceled) 1: Retransmission is reserved
3, 2	CXP1nAREP	Loss-of-arbitration retransmission remaining count flag*2 Indicate the number of remaining retransmissions when arbitration is lost. 00_B: No retransmission remaining (retransmission canceled) 01_B: One round of retransmission remaining 10_B: Two rounds of retransmission remaining 11_B: Unlimited number of retransmissions remaining
1, 0	CXP1nEREP	Error retransmission remaining count flag*2 Indicate the number of remaining retransmissions when an error is detected. 00_B: No retransmission remaining (retransmission canceled) 01_B: One round of retransmission remaining 10_B: Two rounds retransmission remaining 11_B: Unlimited number of retransmissions remaining

Note 1. Not allowed changing from 0_B to 1_B.

Note 2. Not allowed changing from "No retransmission remaining" to "Retransmission remaining".

To cancel retransmission, clear all CXP1nFLG2 bits.

CXP1nSRSV Bit

This bit indicates whether or not frame retransmission is reserved.
For details, see **Section 30.5.7.3, Retransmission Processing**.

- CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  - This bit is fixed to 0.
- CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)

[HW setting conditions]

- When an error is detected when CXP1nEREP ≠ 00_B (the number of retransmissions when detecting an error is other than 0) and a frame transmission request is set
- When an error is detected when CXP1nAREP ≠ 00_B (the number of retransmissions when arbitration is lost is other than 0) and a frame transmission request is set

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- Transmission is started. (PID or Frame or Sleep or PTYPE) (including retransmission)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.50 CXP1nSRSV Bit Access Conditions

Access Conditions			
CXP1nMODE.			
CXP1nEN	CXP1nREPOP	Writing	Reading (Read Value)
0_B	x_B	Not possible	Initial value
1_B	0_B	Not possible	Initial value
1_B	1_B	Enabled	Setting value

The CXP1nRRSV bit

This bit indicates whether or not PID retransmission is reserved.
For details, see **Section 30.5.7.3, Retransmission Processing**.

- CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  - This bit is fixed to 0.
- CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)

[HW setting conditions]

- When an error is detected when CXP1nEREP ≠ 00_B (the number of retransmissions when detecting an error is other than 0) and a PID transmission request is set
- When an error is detected when CXP1nEREP ≠ 00_B (the number of retransmissions when arbitration is lost other than 0) and a PID transmission request is set

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- Transmission is started. (PID or Frame or Sleep or PTYPE) (including retransmission)
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.51 CXP1nRRSV Bit Access Conditions

Access Conditions		Writing	Reading (Read Value)
CXP1nMODE.			
CXP1nEN	CXP1nREPOP		
0_B	x_B	Not possible	Initial value
1_B	0_B	Not possible	Initial value
1_B	1_B	Enabled	Setting value

The CXP1nAREP Bit

This bit indicates the number of remaining retransmissions when arbitration is lost*1.
For details, see **Section 30.5.7.3, Retransmission Processing**.

Note 1. CXP1nFLG2.CXP1nAREP is used when detecting CXPI-NRZ mode and PID start bit error.

- CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  - This bit is fixed to 0.
- CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)

[HW setting conditions]

- The number of retransmissions is decremented when a loss in arbitration is detected.
- The number of retransmissions is decremented when detecting CXPI-NRZ mode and PID start bit error.
- When transmission is started (CXP1nSND.CXP1nREPSEL = 10_B (setting 2)), CXP1nREP.CXP1nAIREP is set.
- Transmission is started (CXP1nSND.CXP1nREPSEL = 01_B (setting 1)). CXP1nREP.CXP1nARREP is set.

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to this bit and operations when accessed.

Table 30.52 CXP1nAREP Bit Access Conditions

Access Conditions		Writing	Reading (Read Value)
CXP1nMODE.			
CXP1nEN	CXP1nREPOP		
0_B	X_B	Not possible	Initial value
1_B	0_B	Not possible	Initial value
1_B	1_B	Enabled	Setting value

The CXP1nEREP Bit

These bits indicate the number of remaining retransmissions when an error is detected.
For details, see **Section 30.5.7.3, Retransmission Processing**.

- CXP1nMODE.CXP1nREPOP = 0_B (no automatic retransmission judgment)
  - This bit is fixed to 0.
- CXP1nMODE.CXP1nREPOP = 1_B (automatic retransmission judged)

[HW setting conditions]

- The number of retransmissions is decremented when detecting an error.
- When transmission is started (CXP1nSND.CXP1nREPSEL = 10_B (setting 2)), CXP1nREP.CXP1nEIREP is set.
- When transmission is started (CXP1nSND.CXP1nREPSEL = 01_B (setting 1)), CXP1nREP.CXP1nERREP is set.

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

Access

The table below lists the conditions for access to these bits and operations when accessed.

Table 30.53 CXP1nEREP Bit Access Conditions

Access Conditions		Writing	Reading (Read Value)
CXP1nMODE.			
CXP1nEN	CXP1nREPOP		
0_B	X_B	Not possible	Initial value
1_B	0_B	Not possible	Initial value
1_B	1_B	Enabled	Setting value

30.4.1.17 CXP1nFLW1 — CXP1 Flow Register 1

This is a status register that indicates the CXP1 operating state.

Access: CXP1nFLW1 register is a read-only register that can be read in 32-bit units.
CXP1nFLW1L register is a read-only register that can be read in 16-bit units.
CXP1nFLW1LL, CXP1nFLW1LH registers are read-only registers that can be read in 8-bit units.

Address: CXP1nFLW1: <CXP1n_base> + 040_H
CXP1nFLW1L: <CXP1n_base> + 040_H
CXP1nFLW1LL: <CXP1n_base> + 040_H,
CXP1nFLW1LH: <CXP1n_base> + 041_H

Value after reset: 0000 0020_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	CXP1nCXMD		CXP1nCXST			—	CXP1n SLP SEQ	CXP1n RXIDLE	CXP1n ESEQ	CXP1n SPIDEN	CXP1n SRES SEQ	CXP1n SSEQ	CXP1n RSEQ
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.54 CXP1nFLW1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12, 11	CXP1nCXMD	CXPI operating mode status Indicate the CXPI operating mode. 00_B: Initializing 01_B: Bus Sleep 10_B: Send Wakeup 11_B: Normal
10 to 8	CXP1nCXST	CXPI operating state Indicate the CXPI operating state. 000_B: Waiting for IFS to elapse 001_B: Waiting for IFS to elapse after transfer is suspended due to an error 010_B: Transmission/reception stopped (IDLE) 011_B: Transmitting PTYPE NOTE: Only when CXP1nMODE.CXP1nACC = 1_B (polling mode) and CXP1nMODE.CXP1nMS = 1_B 100_B: Waiting for response judgment/response data to be set 101_B: PID transmission 110_B: Response (frame information/extension DLC/data/CRC) transmission/reception 111_B: Waiting for the end of the frame (IBS elapsing)
7	Reserved	When read, the value after reset is returned.
6	CXP1nSLPSEQ	Sleep frame status This bit indicates the state of reception to a sleep frame. 0: No sleep reception 1: Sleep reception in progress
5	CXP1nRXIDLE	CXPI bus idle status 0: Not idle 1: Idle
4	CXP1nESEQ	Received PID (External factor) status 0: No received PID (No external factor) 1: Received PID (Externally requested transfer in progress)

Table 30.54 CXP1nFLW1 Register Contents (2/2)

Bit Position	Bit Name	Function
3	CXP1nSPIDEN	Slave node transmission enable status 0: Not allowed slave node transmission 1: Allowed slave node transmission
2	CXP1nSRESSEQ	Non-automatic response transmission status 0: No transmission factor 1: Transmission in progress (transmitting node)
1	CXP1nSSEQ	Transmission factor status 0: No transmission factor 1: Transmission in progress (transmitting node)
0.	CXP1nRSEQ	Reception factor status 0: No reception factor 1: Reception in progress (receiving node)

CXP1nCXMD Bits

These bits indicate the CXPI operating mode.
For details, see **Section 30.5.8.8, Status**.

[HW setting condition]

- State transition (mode sequencer)

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nCXST Bits

These bits indicate the CXPI operating status.

The read data of these bits is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 000_B when CXP1nCXMD is not 11_B (other than normal mode).
For details, see **Section 30.5.8.8, Status**.

[HW setting conditions]

- State transition (flow sequencer)

  These bits change to 001_B (waiting for IFS to elapse after transfer is suspended due to an error) if one of the following condition is satisfied.

- A fatal error is detected.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSLPSEQ Bit

This bit indicates the state of reception to a sleep frame.

The read data of this bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- PID = $1F_H$ is received.

[HW clearing conditions]

- The sleep transition time[*1] has passed after a sleep frame has been received.

Note 1. Sleep transition time = CXP1nSLPWAIT × CXPI nominal bit (Tbit) time

- Received DLC is not 8_H.
- Received 1st byte DATA is not 00_H.
- CXP1nWUP.CXP1nSTCLR = 1_B is written.
- A fatal error is detected.
- A transfer error is detected.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRXIDLE Bit

This bit indicates the CXPI bus idle or not idle.

[HW setting condition]

- After detected stop bit by CXPI bus.
- A fatal error is detected.
- CXP1nWUP.CXP1nSTCLR = 1B is written.
- CXP1nMODE.CXP1nEN = 0B (forced reset)

[HW clearing conditions]

- Detected start bit by CXPI bus.

CXP1nESEQ Bit

This bit indicates the received PID (External factor) status.
The read data of this bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

[HW setting condition]

- An external PID is detected when CXP1nCXST = 010_B (transmission/reception stopped (IDLE))/000_B (waiting for IFS to elapse)/101_B (PID transmission).

[HW clearing conditions]

- An external PID is detected and the IBS time has passed.
- An external PID is detected and disagree with CXP1nRPIDF.
- An external PID is detected and CXP1nFLG1.CXP1nFAIL = 1_B.
- Response Frame transmission/reception is completed
- A fatal error is detected.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSPIDEN Bit

This bit indicates slave node transmission enable status.

The read data of this bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

1. CXP1nMODE.CXP1nACC = 0_B (event trigger mode)
   This bit is fixed to 0.
2. CXP1nMODE.CXP1nACC = 1_B (polling mode)

- When CXP1nMODE.CXP1nMS = 1_B (master node)

[HW setting condition]

- PTYPE is detected when CXP1nCXST = 011_B (transmitting PTYPE).

[HW clearing conditions]

- The IBS time has passed.
- An external PID is detected.
- The external PID acceptance time ends (9 Tbits or more passed while CXP1nCXST = 100_B (waiting for response judgment/response data setting))
- A fatal error is detected.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

- When CXP1nMODE.CXP1nMS = 0_B (slave node)

[HW setting condition]

- PTYPE is detected while CXP1nCXST = 000_B (waiting for IFS to elapse)/010_B (transmission/reception stopped (IDLE)).

[HW clearing conditions]

- The IBS time has passed.
- An external PID is detected.
- PID transmission or frame transmission is performed.
- The external PID acceptance time ends (9 Tbits or more passed while CXP1nCXST = 100_B (waiting for response judgment/response data setting)).
- A fatal error is detected.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSRESSEQ Bit

This bit indicates manual response transmission status.
This bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

[HW setting condition]

- Transfer is started.

[HW clearing conditions]

- Transfer ends.
- A fatal error is detected.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSSEQ Bit

This bit indicates transmission factor status.
The read data of this bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

[HW setting conditions]

- A frame is transmitted.
- A frame is retransmitted.
- An external PID is detected and response (start bit) is transmitted.

[HW clearing conditions]

- Response transmission is completed.
- A fatal error is detected.
- Loss-of-Arbitration.
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- CXP1nMODE.CXP1nEN = 0_B (forced reset).

CXP1nRSEQ Bit

This bit indicates reception factor status
The read data of this bit is only valid while CXP1nCXMD = 11_B (normal mode).
Fixed to 0 when CXP1nCXMD is not 11_B (other than normal mode).

[HW setting conditions]

- The PID is transmitted.
- The PID is retransmitted.
- An external PID is detected and response (start bit) is received.

[HW clearing conditions]

- A response has been received.
- A fatal error is detected.
- Loss-of-Arbitration
- A transfer error is detected.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.18 CXP1nFLW2 — CXPI Flow Register 2

This is a status register that indicates the CXPI transfer frame.

Access: CXP1nFLW2 register is a read-only register that can be read in 32-bit units.
CXP1nFLW2L register is a read-only register that can be read in 16-bit units.
CXP1nFLW2LL, CXP1nFLW2LH registers are read-only registers that can be read in 8-bit units.

Address: CXP1nFLW2: <CXP1n_base> + 044_H
CXP1nFLW2L: <CXP1n_base> + 044_H
CXP1nFLW2LL: <CXP1n_base> + 044_H,
CXP1nFLW2LH: <CXP1n_base> + 045_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CXP1n FRMPHS		CXP1n DATANO							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.55 CXP1nFLW2 Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9, 8	CXP1nFRMPHS	Response frame phase status 00: Frame information or extension DLC 01: Data 10: CRC 11: Undefined CXP1nFLW1.CXP1nRSEQ = 1 (receiving node): Response frame phase status being received. CXP1nFLW1.CXP1nSSEQ=1 (transmitting node): Response frame phase status of next Transmission data.
7 to 0	CXP1nDATANO	Transmit/receive data number status For data frames: CXP1nFLW1.CXP1nRSEQ = 1 (receiving node): Data number being received CXP1nFLW1.CXP1nSSEQ =1 (transmitting node): Data number of next Transmission data. When transfer is completed normally: Quantity of transfer data When transfer ends in an error: Suspended data number

CXP1nFRMPHS Bits

[HW setting condition]

- When receiving data (Including loop-back reception on transmitting node)

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- Transfer is started.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nDATANO Bits

This is status that indicates the number of data.

[HW setting condition]

- When receiving data (Including loop-back reception on transmitting node).

[HW clearing conditions]

- A fatal error occurred.
- CXP1nWUP.CXP1nSTCLR = 1_B is written (simple reset).
- Transfer is started.
- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.19 CXP1nERR — CXPI Error Register

This is a status register that indicates the CXPI operating state.

Access: CXP1nERR register can be read or written in 32-bit units.
CXP1nERRL register can be read or written 16-bit units.
CXP1nERRLL, CXP1nERRLH registers can be read or written 8-bit units.

Address: CXP1nERR: <CXP1n_base> + $04C_H$
CXP1nERRL: <CXP1n_base> + $04C_H$
CXP1nERRLL: <CXP1n_base> + $04C_H$,
CXP1nERRLH: <CXP1n_base> + $04D_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nERRCNT								CXP1n FRM ERR	CXP1n ORN ERR	—	CXP1n DLN ERR	—	CXP1n PTY ERR	CXP1n CRC ERR	CXP1n BIT ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Table 30.56 CXP1nERR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	CXP1nERRCNT	Error counter The error counter indicates the update condition and the result of addition or subtraction. When detecting an error with the error counter value greater than 248, CXP1nFLG1.CXP1nFAIL is set to 1, and PID transmission and frame transmission is stopped.
7	CXP1nFRMERR	Framing error flag An error is detected if the stop bit of the UART frame received by the receiving node is 0. 0: Error not detected 1: Error detected
6	CXP1nORNERR	Overrun error flag An error is detected if the long frame received data is overwritten. 0: Error not detected 1: Error detected
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CXP1nDLNERR	Data length error flag The receiving node compares the DLC of the frame information with the data length of the received frame. A data length error is detected if they do not match. 0: Error not detected 1: Error detected
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 30.56 CXP1nERR Register Contents (2/2)

Bit Position	Bit Name	Function
2	CXP1nPTYERR	Parity error flag The receiving node calculates the number of logical 1s in the received PID field and PTYPE field. It detects a parity error if the number is an even number. The received PID and PTYPE are invalid when detecting an error. 0: Error not detected 1: Error detected
1	CXP1nCRCERR	CRC error flag The receiving node compares the result of automatic CRC calculation and the received CRC value. It detects a CRC error if they do not match. 0: Error not detected 1: Error detected
0	CXP1nBITERR	Bit error flag The value of the transmitted bit and the value of the received bit are compared. A bit error is detected if they do not match. Transmission processing is suspended if a bit error occurs. 0: Error not detected 1: Error detected

CXP1nERRCNT Bits

These bits indicate the error counter value.
For details, see **Section 30.5.7.6, Failure Management Function**.

[HW setting conditions]

- Error counter + 8 when the transmitted PTYPE field has an error
- Error counter + 8 when the transmitted PID field has an error (including the received PID has a framing error after arbitration lost)
- Error counter + 8 when the transmitted response field has an error (excluding the underrun (with data length error) on long frame)
- Error counter – 1 when the transmitted PTYPE field has no error
- Error counter – 1 when the transmitted PID field has no error
- Error counter – 1 when the transmitted response field has no error

NOTE

The upper limit of the error counter is 255.
The lower limit of the error counter is 0.
The following errors apply:

- CXP1nBITERR (bit error)
- CXP1nDLNERR (data length error)
- CXP1nFRMERR (framing error)

[HW clearing conditions]

- Cleared by setting CXP1nMODE.CXP1nEN = 0_B (forced reset) only when CXP1nMODE.CXP1nMS = 1_B (master node)
- Only when CXP1nMODE.CXP1nMS = 0_B (slave node)
- When CXP1nFLG1.CXP1nCLKDET = 0_B and CXP1nWUP.CXP1nWUPREQ = 1_B are written
- When CXP1nFLG1.CXP1nCLKDET = 1_B while CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep mode)
- When CXP1nFLG1.CXP1nFAIL = 0_B is written

CXP1nFRMERR Bit

This bit indicates whether or not a framing error occurred.
For details, see **Section 30.5.7.4(8), Framing Error**.

[HW setting condition]

- The stop bit of the received UART frame is 0.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nORNERR Bit

This bit indicates whether or not an overrun error occurred.
For details, see **Section 30.5.7.4(7), Overrun Error**.

- Long frame (received DLC = F_H)

[HW setting condition]

- A frame reception completion interrupt is generated while CXP1nRFQINT = 1_B (received data save request interrupt source flag).

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

- Normal frame (received DLC ≠ F_H)
  This interrupt is not generated.

CXP1nDLNERR Bit

This bit indicates whether or not a data length error occurred.
For details, see **Section 30.5.7.4(5), Data Length Error**.

[HW setting condition]

- Data length error occurred.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nPTYERR Bit

This bit indicates whether or not a parity error occurred.
For details, see **Section 30.5.7.4(3), Parity Error**.

[HW setting condition]

- A parity error is detected.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nCRCERR Bit

This bit indicates whether or not a CRC error occurred.
For details, see **Section 30.5.7.4(2), CRC Error**.

[HW setting condition]

- A CRC error is detected.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nBITERR Bit

This bit indicates whether or not a bit error occurred.
For details, see **Section 30.5.7.4(1), Bit Error**.

[HW setting conditions]

- Bit error occurred.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.20 CXP1nRPID — CXPI Received PID Register

This register indicates the received PID.

Access: CXP1nRPID register is a read-only register that can be read in 32-bit units.
CXP1nRPIDL register is a read-only register that can be read in 16-bit units.
CXP1nRPIDLL register is a read-only register that can be read in 8-bit units.

Address: CXP1nRPID: <CXP1n_base> + 060_H
CXP1nRPIDL: <CXP1n_base> + 060_H
CXP1nRPIDLL: <CXP1n_base> + 060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1n RPRT	CXP1nRFID						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.57 CXP1nRPID Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	CXP1nRPRT	Parity status Indicates the received parity bit.
6 to 0	CXP1nRFID	Received frame ID status Indicate the received frame ID.

CXP1nRPRT Bit

This bit indicates the parity of the received frame ID or transmitted frame ID.

[HW setting condition]

- The PID is received or transmitted.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRFID Bits

These bits indicate the received frame ID or transmitted frame ID.
For details, see **Section 30.5.6.1, PID**.

[HW setting condition]

- The PID is received or transmitted.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.21 CXP1nSPID — CXPI Transmission PID Register

This register sets the PID for transmission.

Access: CXP1nSPID register can be read or written in 32-bit units.
CXP1nSPIDL register can be read or written 16-bit units.
CXP1nSPIDLL register can be read or written 8-bit units.

Address: CXP1nSPID: <CXP1n_base> + 070_H
CXP1nSPIDL: <CXP1n_base> + 070_H
CXP1nSPIDLL: <CXP1n_base> + 070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1nSPRT	CXP1nSFID						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.58 CXP1nSPID Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	CXP1nSPRT	Parity setting Sets the parity bit of the frame ID.
6 to 0	CXP1nSFID	Transmission frame ID setting Set the frame ID for transmission. Range of allowable settings: 01_H to $7F_H$

CXP1nSPRT Bit

This bit sets the parity bit of the frame ID.

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSFID Bits

These bits set the frame ID.

There are several reserved frame IDs for which the usage is specified.
The table below lists the reserved frame IDs.

Reserved Frame ID	Name	Usage
00_H	Event transmission enable (PTYPE)	Used when CXP1nMODE.CXP1nACC = 1_B
$2F_H$	Inspection request (for ECU supplier)	Inspection
$6F_H$	Inspection response (for ECU supplier)	Inspection
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$	For a response frame	Diagnostic response
$3F_H$	Reserved for future extension	—
$7F_H$	Reserved for future extension	—

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.22 CXP1nRPIDF1m — CXPI Received Frame ID Setting Register 1m (m = 01 to 12)

This register sets the frame ID for reception.

Access: CXP1nRPIDF1m register can be read or written in 32-bit units.
CXP1nRPIDF1mL register can be read or written 16-bit units.
CXP1nRPIDF1mLL register can be read or written 8-bit units.

Address: CXP1nRPIDF1m: <CXP1n_base> + $080_H + 4_H \times (m - 1)$
CXP1nRPIDF1mL: <CXP1n_base> + $080_H + 4_H \times (m - 1)$
CXP1nRPIDF1mLL: <CXP1n_base> + $080_H + 4_H \times (m - 1)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CXP1nFID						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.59 CXP1nRPIDF1m Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CXP1nFID	Frame ID setting Set the frame ID for reception. 00_H: The setting of this register is invalid (The frame ID corresponding to the data of own node is not specified) 01_H to $7F_H$: The frame ID corresponding to the data of own node can be specified

CXP1nFID Bits

These bits set the frame ID for reception of the response field and for manual response.

There are several reserved frame IDs for which the usage is specified.
The table below lists the reserved frame IDs.

Reserved Frame ID	Name	Usage
00_H	Event transmission enable (PTYPE)	Used when CXP1nMODE.CXP1nACC = 1_B
$2F_H$	Inspection request (for ECU supplier)	Inspection
$6F_H$	Inspection response (for ECU supplier)	Inspection
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$	For a response frame	Diagnostic response
$3F_H$	Reserved for future extension	—
$7F_H$	Reserved for future extension	—

The following frames are received irrespective of the settings of this register.

Frame ID	Name	Usage
00_H	Event transmission enable	Used for the polling method
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$*1	For a response frame	Diagnostic response

Note 1. Only for master node setting

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.23 CXP1nRPIDF2m —CXPI Received Frame ID Setting Register 2m (m = 01 to 04)

This register sets the frame ID for reception. A frame can be received with a partial match of bits of the frame ID by using mask bits.

Access: CXP1nRPIDF2m register can be read or written in 32-bit units.
CXP1nRPIDF2mL register can be read or written 16-bit units.
CXP1nRPIDF2mLL, CXP1nRPIDF2mLH registers can be read or written 8-bit units.

Address: CXP1nRPIDF2m: <CXP1n_base> + $0B0_H$ + 4_H × (m − 1)
CXP1nRPIDF2mL: <CXP1n_base> + $0B0_H$ + 4_H × (m − 1)
CXP1nRPIDF2mLL: <CXP1n_base> + $0B0_H$ + 4_H × (m − 1),
CXP1nRPIDF2mLH: <CXP1n_base> + $0B0_H$ + 4_H × (m − 1) + 1_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CXP1nFIDMSK							—	CXP1nFID						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.60 CXP1nRPIDF2m Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	CXP1nFIDMSK	Received frame ID mask setting Mask the frame ID of the set bits. Range of allowable settings: 00_H to $7F_H$
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CXP1nFID	Frame ID setting Set the frame ID for reception. 00_H: The setting of this register is invalid (The frame ID corresponding to the data of own node is not specified) 01_H to $7F_H$: The frame ID corresponding to the data of own node can be specified

CXP1nFIDMSK Bits

These bits mask the frame ID value for the bits being set to 1.

- Setting example: CXP1nFID = 35_H, CXP1nFIDMSK = 03_H
  Received FID = 33_H → Discard
  Received FID = 34_H → Receive
  Received FID = 36_H → Receive
  Received FID = 37_H → Receive

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nFID Bits

These bits set the frame ID for reception of the response field and for manual response.
For details, see **Section 30.5.6.1, PID**.

There are several reserved frame IDs for which the usage is specified.
The table below lists the reserved frame IDs.

Reserved Frame ID	Name	Usage
00_H	Event transmission enable (PTYPE)	Used when CXP1nMODE.CXP1nACC = 1_B
$2F_H$	Inspection request (for ECU supplier)	Inspection
$6F_H$	Inspection response (for ECU supplier)	Inspection
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$	For a response frame	Diagnostic response
$3F_H$	Reserved for future extension	—
$7F_H$	Reserved for future extension	—

The following frames are received irrespective of the settings of this register.

Fame ID	Name	Usage
00_H	Event transmission enable	Used for the polling method
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$*1	For a response frame	Diagnostic response

Note 1. Only for master node setting

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.24 CXP1nARPID —CXPI Automatic Response PID Setting Register

This register sets the PID for automatic response.

Access: CXP1nARPID register can be read or written in 32-bit units.
CXP1nARPIDL register can be read or written 16-bit units.
CXP1nARPIDLL register can be read or written 8-bit units.

Address: CXP1nARPID: <CXP1n_base> + $0C0_H$
CXP1nARPIDL: <CXP1n_base> + $0C0_H$
CXP1nARPIDLL: <CXP1n_base> + $0C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CXP1nARFID						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.61 CXP1nARPID Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CXP1nARFID	Automatic response frame ID setting Set the frame ID for automatic response. 00_H: The setting of this register is invalid (The frame ID corresponding to the data of own node is not specified) 01_H to $7F_H$: The frame ID corresponding to the data of own node can be specified.

CXP1nARFID Bits

These bits set the frame ID for automatic response.
For details, see **Section 30.5.6.1, PID**.

There are several reserved frame IDs for which the usage is specified.
The table below lists the reserved frame IDs.

Reserved Frame ID	Name	Usage
00_H	Event transmission enable (PTYPE)	Used when CXP1nMODE.CXP1nACC = 1_B
$2F_H$	Inspection request (for ECU supplier)	Inspection
$6F_H$	Inspection response (for ECU supplier)	Inspection
$1F_H$	For a request frame	Sleep frame or diagnostic request
$5F_H$	For a response frame	Diagnostic response
$3F_H$	Reserved for future extension	—
$7F_H$	Reserved for future extension	—

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.25 CXP1nRFRI —CXPI Received Frame Information Register

This register stores the received frame information.

Access: CXP1nRFRI register is a read-only register that can be read in 32-bit units.
CXP1nRFRIL register is a read-only register that can be read in 16-bit units.
CXP1nRFRILL, CXP1nRFRILH registers are read-only registers that can be read in 8-bit units.

Address: <CXP1nRFRI: <CXP1n_base> + 120_H
CXP1nRFRIL: <CXP1n_base> + 120_H
CXP1nRFRILL: <CXP1n_base> + 120_H,
CXP1nRFRILH: <CXP1n_base> + 121_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nRDLC2								CXP1nRDLC1				CXP1nRWUP	CXP1nRSLP	CXP1nRCT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.62 CXP1nRFRI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	CXP1nRDLC2	Extended frame data length status Indicate the received extension DLC. Setting range: 00_H to FF_H
7 to 4	CXP1nRDLC1	Data length status Indicate the length of the received data. 0_H to C_H: Data length D_H to E_H: None F_H: Long frame
3	CXP1nRWUP	Wakeup indicator status Indicates the received wakeup indicator. 0: No wakeup by the own node as a trigger 1: Wakeup by the own node as a trigger
2	CXP1nRSLP	Sleep indicator status Indicates the received sleep indicator. 0: Disables sleep mode 1: Enables sleep mode
1, 0	CXP1nRCT	Counter status Indicate the received counter value.

CXP1nRDLC2 Bits

Indicate the extension DLC value received in a long frame.

[HW setting condition]

- Frame information (extension DLC) is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRDLC1 Bits

These bits indicate the received DLC value.

[HW setting condition]

- Frame information is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRWUP Bit

This bit indicates the received wakeup indicator value.

[HW setting condition]

- Frame information is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRSLP Bit

This bit indicates the received sleep indicator value.

[HW setting condition]

- Frame information is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRCT Bits

These bits indicate the received counter value.

[HW setting condition]

- Frame information is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.26 CXP1nSFRI — CXPI Transmission Frame Information Register

This register sets the transmission frame information.

Access: CXP1nSFRI register can be read or written in 32-bit units.
CXP1nSFRIL register can be read or written 16-bit units.
CXP1nSFRILL, CXP1nSFRILH registers can be read or written 8-bit units.

Address: CXP1nSFRI: <CXP1n_base> + 130_H
CXP1nSFRIL: <CXP1n_base> + 130_H
CXP1nSFRILL: <CXP1n_base> + 130_H,
CXP1nSFRILH: <CXP1n_base> + 131_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nSDLC2								CXP1nSDLC1				CXP1n SWUP	CXP1n SSLP	CXP1nSCT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.63 CXP1nSFRI Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	CXP1nSDLC2	Extended frame data length setting Set the extension DLC in a long frame. Setting range: 00_H to FF_H
7 to 4	CXP1nSDLC1	Data length setting 0_H to C_H: Data length D_H to E_H: Setting prohibited F_H: Long frame
3	CXP1nSWUP	Wakeup indicator setting 0: No wakeup by the own node as a trigger 1: Wakeup by the own node as a trigger
2	CXP1nSSLP	Sleep indicator setting 0: Disables sleep mode 1: Enables sleep mode
1, 0	CXP1nSCT	Counter setting Indicate the continuity of frames. Set these bits considering the continuity of frames by the application. Range of allowable settings: 0_H to 3_H

The conditions for setting and clearing of the CXP1nSFRI register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.27 CXP1nARFRI — CXPI Automatic Response Frame Information Register

This register sets the automatic response frame information.

Access: CXP1nARFRI register can be read or written in 32-bit units.
CXP1nARFRIL register can be read or written 16-bit units.
CXP1nARFRILL register can be read or written 8-bit units.

Address: CXP1nARFRI: <CXP1n_base> + 140_H
CXP1nARFRIL: <CXP1n_base> + 140_H
CXP1nARFRILL: <CXP1n_base> + 140_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1nASDLC				CXP1nASWUP	CXP1nASSLP	CXP1nASCT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.64 CXP1nARFRI Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	CXP1nASDLC	Data length setting Set the data length of frames for automatic response. Range of allowable settings: 00_H to $0C_H$ (0 byte to 12 bytes) Other than above: Setting prohibited
3	CXP1nASWUP	Wakeup indicator setting Sets the wakeup indicator of frames for automatic response. 0: No wakeup by the own node as a trigger 1: Wakeup by the own node as a trigger
2	CXP1nASSLP	Sleep indicator setting Sets the sleep indicator of frames for automatic response. 0: Disables sleep mode 1: Enables sleep mode
1, 0	CXP1nASCT	Counter setting Set the counter of frames for automatic response. Used to indicate the continuity of frames. Set these bits considering the continuity of frames by the application. Range of allowable settings: 00_H to 03_H

The conditions for setting and clearing of the CXP1nARFRI register are as follows:

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.28 CXP1nRCRC — CXPI Received CRC Register

This register indicates the CRC of the received frame.

Access: CXP1nRCRC register is a read-only register that can be read in 32-bit units.
CXP1nRCRCL register is a read-only register that can be read in 16-bit units.
CXP1nRCRCLL, CXP1nRCRCLH registers are read-only registers that can be read in 8-bit units.

Address: CXP1nRCRC: <CXP1n_base> + $1A0_H$
CXP1nRCRCL: <CXP1n_base> + $1A0_H$
CXP1nRCRCLL: <CXP1n_base> + $1A0_H$,
CXP1nRCRCLH: <CXP1n_base> + $1A1_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nRCRC2								CXP1nRCRC1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.65 CXP1nRCRC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	CXP1nRCRC2	Upper CRC status These bits indicate the upper CRC of the extended received frame.
7 to 0	CXP1nRCRC1	Lower CRC status These bits indicate the lower CRC of the extended received frame.

CXP1nRCRC2 Bits

These bits indicate the CRC value (upper byte) received in a long frame.
Data in these bits is invalid while CXP1nRFRI.CXP1nRDLC1 = 00_H to $0C_H$ (normal frame).

NOTE

The upper byte is indicated when data is received in an extended frame.

[HW setting condition]

- A frame is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nRCRC1 Bit

These bits indicate the received CRC value.

NOTE

The lower byte is indicated when data is received in an extended frame.

[HW setting condition]

- A frame is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.29 CXP1nSCRC — CXPI Transmission CRC Register

This register indicates CRC of the frame for transmission. It stores the CRC value generated by the controller.

Access: CXP1nSCRC register is a read-only register that can be read in 32-bit units.
CXP1nSCRCL register is a read-only register that can be read in 16-bit units.
CXP1nSCRCLL, CXP1nSCRCLH registers are read-only registers that can be read in 8-bit units.

Address: CXP1nSCRC: <CXP1n_base> + $1B0_H$
CXP1nSCRCL: <CXP1n_base> + $1B0_H$
CXP1nSCRCLL: <CXP1n_base> + $1B0_H$,
CXP1nSCRCLH: <CXP1n_base> + $1B1_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nSCRC2								CXP1nSCRC1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.66 CXP1nSCRC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	CXP1nSCRC2	Upper CRC status Indicate the upper CRC of the extended transmission frame.
7 to 0	CXP1nSCRC1	Lower CRC status Indicate the lower CRC of the extended transmission frame.

CXP1nSCRC2 Bits

These bits indicate the CRC value for transmission in a long frame.
Data in these bits is invalid while CXP1nSFRI.CXP1nSDLC1 = 00_H to $0C_H$ (normal frame).

NOTE

The upper byte is indicated when data is transmitted in an extended frame.

[HW setting condition]

- A frame is transmitted.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

CXP1nSCRC1 Bits

These bits indicate the CRC value for transmission.

NOTE

The lower byte is indicated when data is transmitted in an extended frame.

[HW setting condition]

- A frame is transmitted.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.30 CXP1nARCRC — CXPI Automatic Response CRC Register

This register indicates the CRC value of the automatic response frame. It stores the CRC value generated by the controller.

Access: CXP1nARCRC register is a read-only register that can be read in 32-bit units.
CXP1nARCRCL register is a read-only register that can be read in 16-bit units.
CXP1nARCRCLL register is a read-only register that can be read in 8-bit units.

Address: CXP1nARCRC: <CXP1n_base> + $1C0_H$
CXP1nARCRCL: <CXP1n_base> + $1C0_H$
CXP1nARCRCLL: <CXP1n_base> + $1C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CXP1nASCRC							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.67 CXP1nARCRC Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	CXP1nASCRC	CRC status This register indicates the CRC value of the automatic response frame.

CXP1nASCRC Bits

These bits indicate the CRC value for transmission in an automatic response.

[HW setting condition]

- A frame is transmitted.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.31 CXP1nRDATAm — CXPI Received Data Register m (m = 01 to 08)

This register indicates the received data.

Access: CXP1nRDATAm register is a read-only register that can be read in 32-bit units.
CXP1nRDATAmL register is a read-only register that can be read in 16-bit units.
CXP1nRDATAmLL, CXP1nRDATAmLH registers are read-only registers that can be read in 8-bit units.

Address: CXP1nRDATAm: <CXP1n_base> + 200_H+ 4_H × (m - 1)
CXP1nRDATAmL: <CXP1n_base> + 200_H + 4_H × (m - 1)
CXP1nRDATAmLL: <CXP1n_base> + 200_H + 4_H × (m - 1),
CXP1nRDATAmLH: <CXP1n_base> + 200_H + 4_H × (m - 1) + 1_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nRDAT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.68 CXP1nRDATAm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	CXP1nRDAT	Received data status Indicate the received data.

CXP1nRDAT Bits

These bits indicate the received data.

[HW setting condition]

- A frame is received.

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.32 CXP1nSDATAm — CXPI Transmission Data Register m (m = 01 to 08)

This register sets the data for transmission.

Access: CXP1nSDATAm register can be read or written in 32-bit units.
CXP1nSDATAmL register can be read or written 16-bit units.
CXP1nSDATAmLL, CXP1nSDATAmLH registers can be read or written 8-bit units.

Address: CXP1nSDATAm: <CXP1n_base> + 300_H + 4_H × (m - 1)
CXP1nSDATAmL: <CXP1n_base> + 300_H + 4_H × (m - 1)
CXP1nSDATAmLL: <CXP1n_base> + 300_H + 4_H × (m - 1),
CXP1nSDATAmLH: <CXP1n_base> + 300_H + 4_H × (m - 1) + 1_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nSDAT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.69 CXP1nSDATAm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CXP1nSDAT	Transmission data setting Set the data for transmission. Setting range: 0000_H to $FFFF_H$

CXP1nSDAT Bits

These bits indicate the data for transmission.

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.4.1.33 CXP1nARDATAm — CXPI Automatic Response Data Register m (m = 01 to 06)

This register sets the transmission data.

Access: CXP1nARDATAm register can be read or written in 32-bit units.
CXP1nARDATAmL register can be read or written 16-bit units.
CXP1nARDATAmLL, CXP1nARDATAmLH registers can be read or written 8-bit units.
Note that writing is only possible while the CXP1nMODE.CXP1nEN is 1.
While CXP1nMODE.CXP1nEN = 0, the initial value is read.

Address: CXP1nARDATAm: <CXP1n_base> + 400_H + 4_H × (m - 1)
CXP1nARDATAmL: <CXP1n_base> + 400_H + 4_H × (m - 1)
CXP1nARDATAmLL: <CXP1n_base> + 400_H + 4_H × (m - 1),
CXP1nARDATAmLH: <CXP1n_base> + 400_H + 4_H × (m - 1) + 1_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CXP1nARDAT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.70 CXP1nARDATAm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CXP1nARDAT	Automatic response data setting Set the automatic response data. Setting range: 0000_H to $FFFF_H$

CXP1nARDAT Bits

These bits indicate the data for transmission in an automatic response.

[HW setting condition]

- None

[HW clearing condition]

- CXP1nMODE.CXP1nEN = 0_B (forced reset)

30.5 Operation

30.5.1 CXPI System Configuration

The transfer mode for this IP can be selectable from CXPI-PWM mode and CXPI-NRZ mode.

Since different functions are assigned to the IP and the CXPI transceiver depending on the transfer mode, the external connections for the LSI are also different.

In addition, since some functions are not supported by the IP, they must be supported externally.

The following describes the system configuration in each transfer mode.

(1) CXPI-PWM mode (CXP1nMODE.CXP1nMD = 0_B)

Figure 30.5, System Configuration in CXPI-PWM Mode shows the CXPI system configuration when CXPI-PWM mode is selected.

NOTE

It has the same system configuration in the master node and the slave node.

Since this IP performs PWM encoding/decoding in CXPI-PWM mode, it communicates with the CXPI transceiver in the PWM waveform.

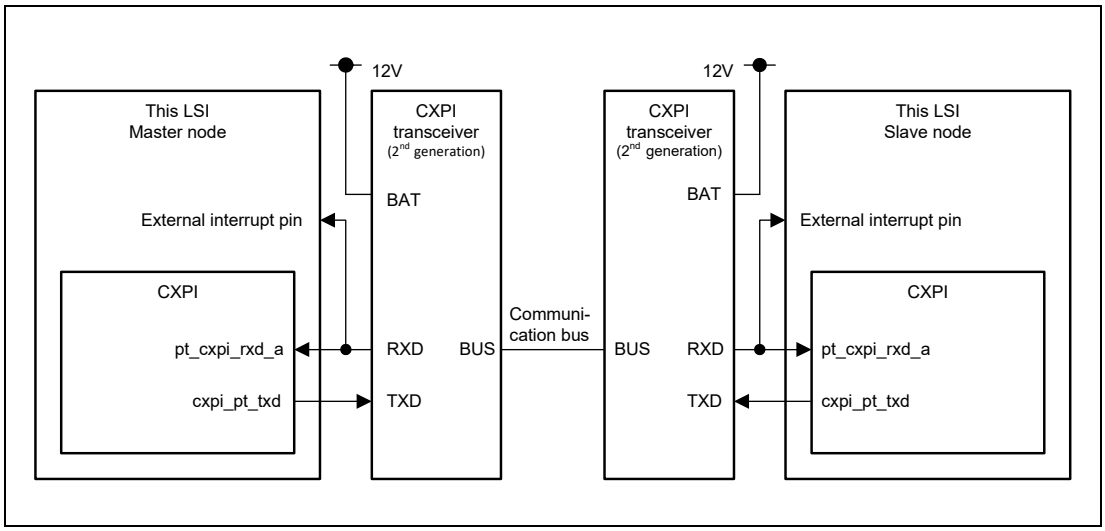


Figure 30.5 System Configuration in CXPI-PWM Mode

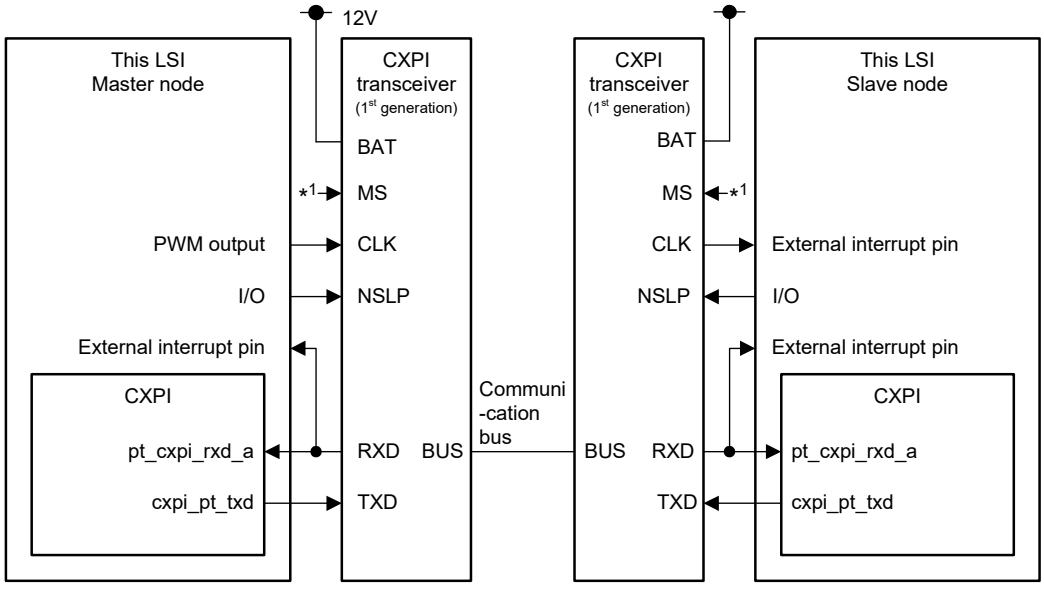
[Pins to be provided by the LSI:]

- External interrupt pin (connected to RXD)
  This IP does not have the wakeup pulse detection function.
  The CXPI transceiver detects wakeup pulses.
  The CXPI transceiver outputs a detection signal to the LSI after detecting a wakeup pulse.
  This LSI detects a detection signal at the external interrupt pin and returns from LSI standby mode.
- Transmit pin (cxpi_pt_txd) (connected to TXD)
  This is the CXPI transmit pin.
- Receive pin (pt_cxpi_rxd_a) (connected to RXD)
  This is the CXPI receive pin.

(2) CXPI-NRZ mode (CXP1nMODE.CXP1nMD = 1_B)

Figure 30.6 shows the CXPI system configuration when CXPI-NRZ mode is selected for this IP.

In CXPI-NRZ mode, the CXPI transceiver performs PWM encoding/decoding.
The communication interface with the CXPI transceiver is UART.
For this reason, the following CXPI transceiver pins must be controlled.



Note 1. The specifications for the MS pin differs from one CXPI transceiver manufacturer to another. Handle the pin based on the manufacturer's specifications.

Figure 30.6 System Configuration in CXPI-NRZ Mode

[Pins to be provided by the LSI]

Common to the master node and slave nodes

- External interrupt pin (connected to RXD)
  This IP does not have the wakeup pulse detection function.
  The CXPI transceiver detects wakeup pulses.
  The CXPI transceiver outputs a detection signal to the LSI after detecting a wakeup pulse.
  This LSI detects a detection signal at the external interrupt pin and returns from LSI standby mode.
- I/O (connected to NSLP)
  This is the pin to reduce the current consumption of the CXPI transceiver. (Only the wakeup pulse receive circuit operates.)
  Setting NSLP = 0 reduces the current consumption.
  Setting NSLP = 1 changes the CXPI transceiver to operating mode.
- — (connected to the MS pin)
  This pin is used to select from the master node and the slave node.
  Since the specifications for the MS pin differ from one manufacturer to another, handle the pin based on the specifications.
- Transmit pin (cxpi_pt_txd) (connected to TXD)
  This is the CXPI transmit pin. The transceivers communicate with each other in UART mode.
- Receive pin (pt_cxpi_rxd_a) (connected to RXD)
  This is the CXPI receive pin. The transceivers communicate with each other in UART mode.

Master node

- Duty 50 clock (PWM output) (connected to CLK)
  The LSI outputs bit rate clock cycles so that the CXPI transceiver can perform PWM encoding/decoding. Timer resources are needed this. Be based upon the specification of the transceiver about the output waveform.

Slave node

- External interrupt pin (connected to CLK/RXD)
  Since the transceivers communicate with each other in UART mode in CXPI-NRZ mode, the slave node cannot detect the CXPI clock at the pins for use in the communication. Therefore, the CXPI clock shall be detected at the external interrupt pin of the LSI to switch the CXPI operating mode.
  (Since the specifications for the clock detection function of the transceiver differ from one manufacturer to another, handle the pin based on the specifications. The clock signal is output from the CLK pin or the RXD pin.)

30.5.2 CXPI Operation Overview

The access method can be selected from the event trigger method and the polling method.

The functions and operations differ between the access methods.

The following overviews the operation in each access method.

30.5.2.1 Event Trigger Method (CXP1nMODE.CXP1nACC = 0_B)

In the event trigger method, each node can transmit a PID freely when it is in the transmission/ reception stopped (IDLE) state.

The master node outputs CXPI clock signals. A slave node operates in synchronization with the CXPI clock signal sent from the master node.

The figure below shows the sequence in the event trigger method.

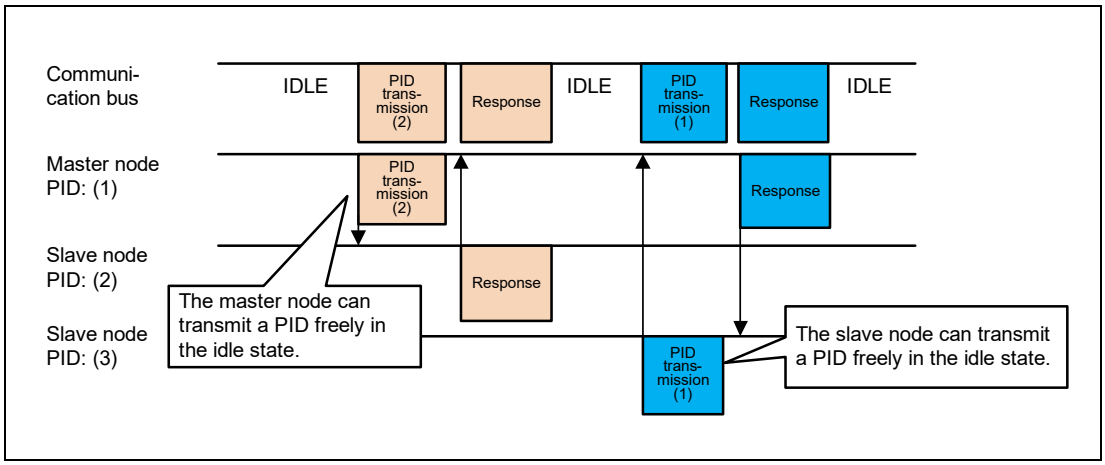


Figure 30.7 Sequence in the Event Trigger Method

30.5.2.2 Polling Method (CXP1nMODE.CXP1nACC = 1_B)

In the polling method, the master node can transmit frames freely when it is in the transmission/ reception stopped (IDLE) state, as in the event trigger method.

When the master node has transmitted a PTYPE field, the slave node can transmit PID freely.

Any slave node that has not received the PTYPE field cannot transmit a PID by hardware protection.

The figure below shows the sequence in the polling method.

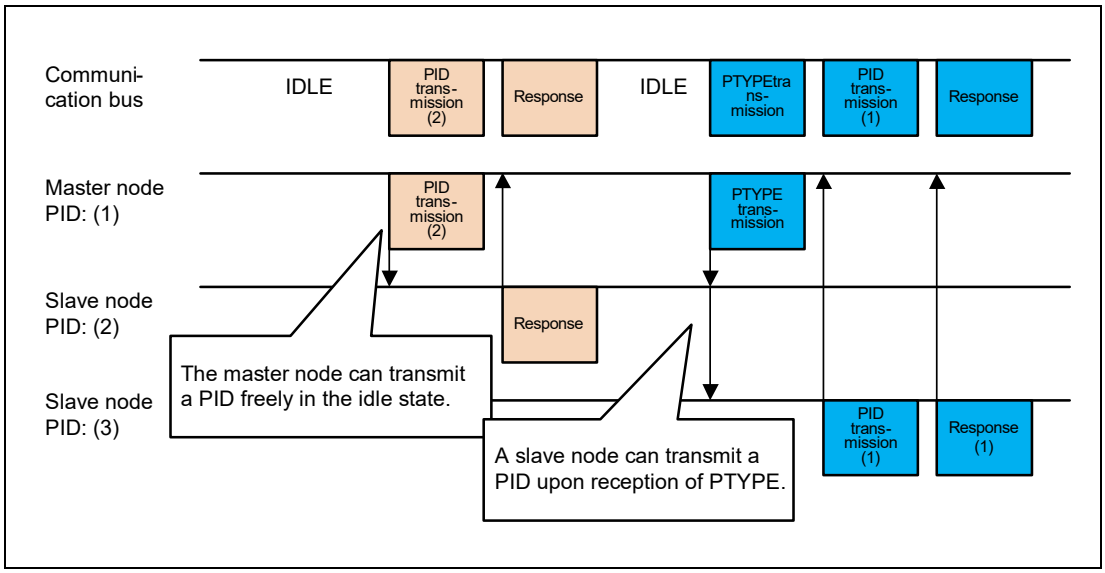


Figure 30.8 Sequence in the Polling Method

30.5.3 Functions

The available functions of this IP differ depending on the transfer mode, access method, and master node/slave node selection. The table below lists the functions (features) that are valid in each setting.

Table 30.71 Functions

Condition			CXPI-PWM Mode				CXPI-NRZ Mode			
			Event Trigger		Polling		Event Trigger		Polling	
			Master	Slave	Master	Slave	Master	Slave	Master	Slave
Basic functions	Protocol		Compliant with *the JASO D015 specification*							
	Baud rate		9.6 kbits/s / 10.4 kbits/s / 19.2 kbits/s / 20 kbits/s							
	Clock sharing		Supported				Not supported (supported by the transceiver)			
	PWM encoding/decoding		Supported				Not supported (supported by the transceiver)			
Data functions	PID		Provided with registers in which frame IDs can be set. Transmission: 1 Reception: 16 (4 registers have a frame ID filtering function.) Automatic response: 1							
	Data buffer		Transmission: 16 bytes / reception: 16 bytes / automatic response: 12 bytes							
	Frame structure		Normal frame, long frame, sleep frame							
	Number of data byes in the response field		Normal frame: variable from 0 to 12 bytes / Long frame: variable from 0 to 255 bytes Sleep frame: fixed to 8 bytes							
	Error detection method		Normal frame and sleep frame: 8-bit CRC Long frame: 16-bit CRC							
Control functions	Transmission	Clock transmission	Supported	Not supported	Supported	Not supported	Not supported			
		CXPI clock detection notification	Not supported				Supported			
		PID transmission	Supported							
		Frame transmission	Supported							
		Automatic response	Supported							
		Manual response	Supported							
		PTYPE transmission	Not supported		Supported	Not supported	Not supported		Supported	Not supported
	Arbitration		Bit basis				Byte basis **NOTE:** Bit-basis arbitration is performed by the transceiver.			
	IFS and IBS setting function		Supported							
	Error detection		Supported							
	Confinement on errors		Supported							
	Wake up / Sleep functions	Wakeup pulse transmission	Not supported	Supported	Not supported	Supported	Not supported	Supported	Not supported	Supported
		Wakeup pulse reception	Not supported (supported by the transceiver)							
		Sleep frame request	Supported	Not supported	Supported	Not supported	Supported	Not supported	Supported	Not supported
		Sleep frame reception	Not supported	Supported	Not supported	Supported	Not supported	Supported	Not supported	Supported
	Bus monitoring		Supported							
	Failure management		Supported							
	Error bit function		Not supported							
Extended functions	Frame ID filtering function		Supported							
	Noise filter		Supported				Not supported			
	Diagnostic function		A readable error counter is installed.							
	Software reset		A forced reset is possible by writing to a register. A simple reset is possible by writing to a register.							
	Status		The operating status of each hardware unit can be read.							
Interrupts			Supported							

30.5.4 Overall Operation

The CXPI operates differently depending on the wakeup factor.

The following describes the CXPI operation for each wakeup factor.

The hardware and software behaviors are different according to each operation. Operations triggered by each wakeup factor are outlined in the following pages.

Table 30.72 Operation Triggered by Each Wakeup Factor

No.	Wakeup Factor	Operation		Described in Section
		Operation after Wakeup	Continued Operation	
1	CXPI activation	Initial Setting	-	**30.5.10.2**
2	PID transmission (CXP1nSND.CXP1nSIDREQ = 1)	PID transmission	Response reception	**30.5.4.1(1)**
			No response from the target node	**30.5.4.1(2)**
3	Frame transmission (CXP1nSND.CXP1nSFRREQ = 1)	PID transmission	Response transmission	**30.5.4.2(1)**
4	Manual response (CXP1nSND.CXP1nSRSREQ = 1)	PID reception	Response transmission	**30.5.4.3(1)**
			Own node is not the target.	**30.5.4.3(2)**
5	Automatic response (Automatically judged when receiving a PID)	PID reception	Response transmission	**30.5.4.4(1)**
			Own node is not the target.	**30.5.4.4(2)**
6	PID reception (frame reception) (Automatically judged when receiving a PID)	PID reception	Response reception	**30.5.4.5(1)**
			Own node is not the target.	**30.5.4.5(2)**
7	PTYPE transmission (CXP1nSND.CXP1nSPTREQ = 1)	PTYPE transmission	No. 4 to No. 6	**30.5.4.6(1)**
			No response from the target node	**30.5.4.6(2)**
8	PTYPE reception	PTYPE reception	No. 2 and No. 3	**30.5.4.7(1)**
			PID not transmitted	**30.5.4.7(2)**
9	Sleep frame transmission	PID transmission	Response transmission	**30.5.4.8**
10	Sleep frame reception	PID reception	Response reception	**30.5.4.9**
11	Wakeup pulse transmission (CXP1nWUP.CXP1nWUPREQ = 1)	Wakeup pulse transmission	Starts another node.	**30.5.4.10**
12	Wakeup pulse reception	Wakeup pulse reception	Own node starts up.	**30.5.4.11**

30.5.4.1 PID Transmission

(1) PID Transmission to Response Reception

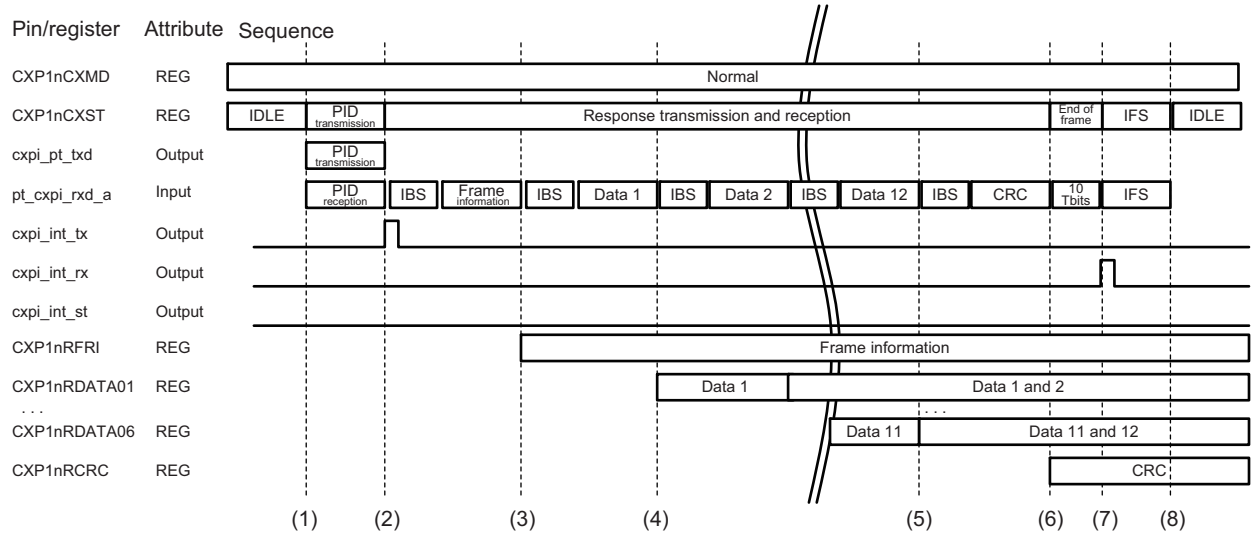
The CXPI can receive data of another node by transmitting a PID.

A normal frame or a long frame is received as a response according to the frame information (CXP1nRFRI.CXP1nRDLC1).

This section provides an overview of the hardware and software operation in each frame structure.

(a) Normal Frame

The following provides an overview of the hardware and software operation when the frame structure is the normal frame structure.



Note: IBS: IBS controlled by another node

Figure 30.9 Operation Sequence from PID Transmission to Response Reception (Normal Frame)

Operation

Operation setting: Frame structure: Normal frame
($CXP1nRFRI.CXP1nRDLC1 \neq F_H$)*1

Number of data bytes in the response: 12 bytes
($CXP1nRFRI.CXP1nRDLC1 = C_H$)*1

Note 1. The frame structure and the number of data bytes are determined based on the received data.

(1) Writes 1 to CXP1nSND.CXP1nSIDREQ and makes a PID transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because $CXP1nFLW1.CXP1nCXST = 010_B$ (IDLE).
(If $CXP1nFLW1.CXP1nCXST$ is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PID transmission completion interrupt is generated. CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) Frame information is received, and is stored in CXP1nRFRI.

(4) The first-byte data is received. The received data is stored in CXP1nRDATA01.

NOTE

After that, data is stored in CXP1nRDATAm (m = 01 to 06) as each byte is received.

(5) The 12th-byte data is received. The received data is stored in CXP1nRDATA06.

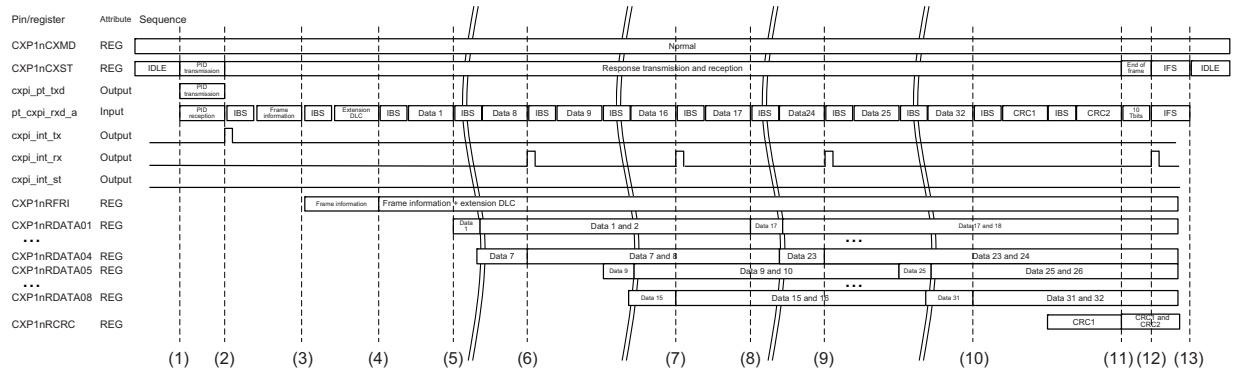
(6) The CRC value is received. The CRC value is stored in CXP1nRCRC.

(7) 10 Tbits (fixed value determined by the hardware) have passed after receiving the CRC value.
A frame reception completion interrupt is generated and CXP1nINT.CXP1nRFRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFRINT to 0.

(8) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time from CRC value is received, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

(b) Long frame

The following provides an overview of the hardware and software operation when the frame structure is the long frame structure.



Note: IBS: IBS controlled by another node

Figure 30.10 Operation Sequence from PID Transmission to Response Reception (Long Frame)

Operation

Operation setting: Frame structure: Long frame
($CXP1nRFRI.CXP1nRDLC1 = F_H$)*1

Number of data bytes in the response: 32 bytes
($CXP1nRFRI.CXP1nRDLC\ 2 = 20_H$)*1

Note 1. The frame structure and the number of data bytes are determined based on the received data

(1) Writes 1 to CXP1nSND.CXP1nSIDREQ and makes a PID transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because $CXP1nFLW1.CXP1nCXST = 010_B$ (IDLE).
(If $CXP1nFLW1.CXP1nCXST$ is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PID transmission completion interrupt is generated and CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) Frame information is received, and is stored in CXP1nRFRI.

(4) Extension DLC is received, and is stored in CXP1nRFRI.

(5) The first-byte data is received. The received data is stored in CXP1nRDATA01.

(6) The 8th-byte data is received. The received data is stored in CXP1nRDATA04.
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
The software reads CXP1nRDATA01 to CXP1nRDATA04 and saves the received data.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFQINT to 0.

(7) The 16th-byte data is received. The received data is stored in CXP1nRDATA08.
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
The software reads CXP1nRDATA05 to CXP1nRDATA08 and saves the received data since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(8) The 17th-byte data is received. The received data is stored (overwritten) in CXP1nRDATA01.

(9) The 24th-byte data is received. The received data is stored in CXP1nRDATA04.
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFQINT to 0.
The software reads CXP1nRDATA01 to CXP1nRDATA04 and saves the received data.

(10) The 32nd-byte data is received. The received data is stored in CXP1nRDATA08.
No received data save request interrupt is generated because this is the last data.

(11) The CRC value is received. The received CRC is stored in CXP1nRCRC.

(12) 10 Tbits (fixed value determined by the hardware) have passed after receiving the CRC value.
A frame reception completion interrupt is generated and CXP1nINT.CXP1nRFRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFRINT to 0.

(13) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time from CRC value is received, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

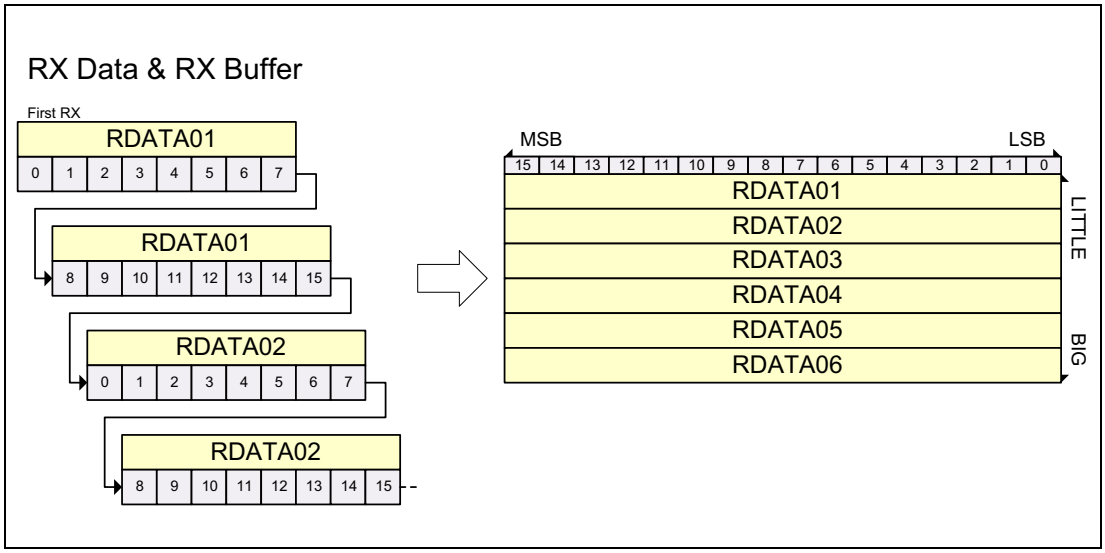


Figure 30.11 Received DATA and Received BUFFER

(2) PID Transmission to No Response from the Target Node

The CXPI may not receive a response from the target node after transmitting a PID.

This section provides an overview of the hardware and software operation when no response is received from the target node after transmitting a PID.

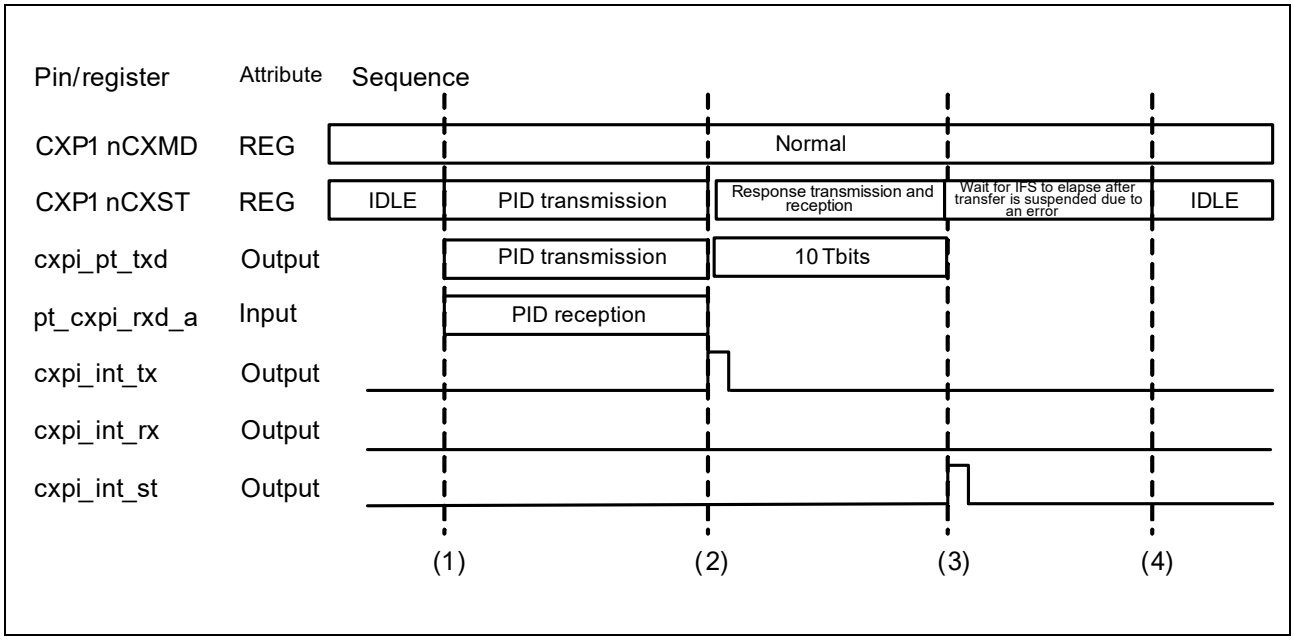


Figure 30.12 PID Transmission to No Response from the Target Node

Operation

(1) Writes 1 to CXP1nSND.CXP1nSIDREQ and makes a PID transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because CXP1nFLW1.CXP1nCXST = 010_B (IDLE).
(If CXP1nFLW1.CXP1nCXST is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PID transmission completion interrupt is generated and CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) 10 Tbits (fixed value determined by the hardware) have passed (a time longer than IBS has passed).
An error detection interrupt (data length error or bit error) is generated and CXP1nINT.CXP1nERRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nERRINT to 0.

(4) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after detect error, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

30.5.4.2 Frame Transmission

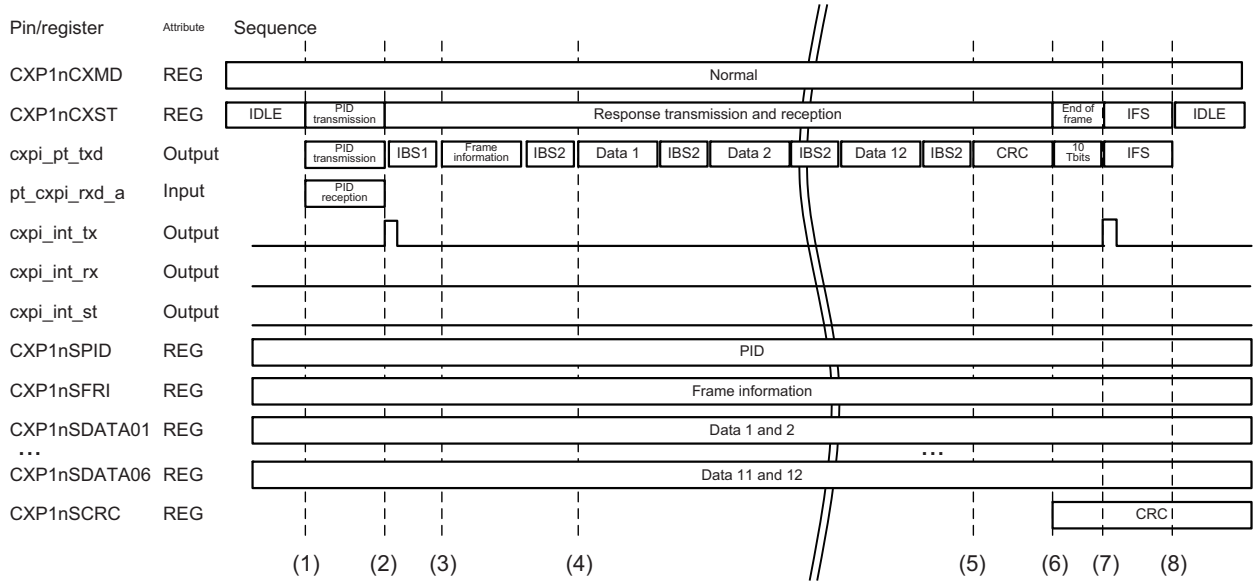
(1) PID Transmission to Response Transmission

The CXPI can transmit a response to the target node by transmitting a PID.

A normal frame or a long frame can be transmitted as a response according to the frame information (DLC) setting. This section provides an overview of the hardware and software operation in each frame structure.

(a) Normal Frame

The following provides an overview of the hardware and software operation when the frame structure is the normal frame structure.



Note: IBS: IBS controlled by another node
IBS1: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 bit. Then, the next data is transmitted.
IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS2 bit. Then, the next data is transmitted.

Figure 30.13 Operation Sequence from PID Transmission to Response Transmission (Normal Frame)

Operation

Operation setting: Frame structure: Normal frame
($CXP1nSFRI.CXP1nSDLC1 \neq F_H$)

Number of data bytes in the response: 12 bytes
($CXP1nSFRI.CXP1nSDLC1 = C_H$)

(1) Writes 1 to CXP1nSND.CXP1nSFRREQ and makes a frame transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because $CXP1nFLW1.CXP1nCXST = 010_B$ (IDLE).
(If $CXP1nFLW1.CXP1nCXST$ is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PID transmission completion interrupt is generated. CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) Frame information is transmitted. (Data in CXP1nSFRI[7:0] is transmitted.)

(4) The first-byte data is transmitted. (Data in CXP1nSDATA01[7:0] is transmitted.)
Note: Then, data in CXP1nSDATAm (m = 01 to 06) is transmitted on a byte basis.

(5) The CRC value is transmitted. (The CRC value calculated within the circuit is transmitted.)

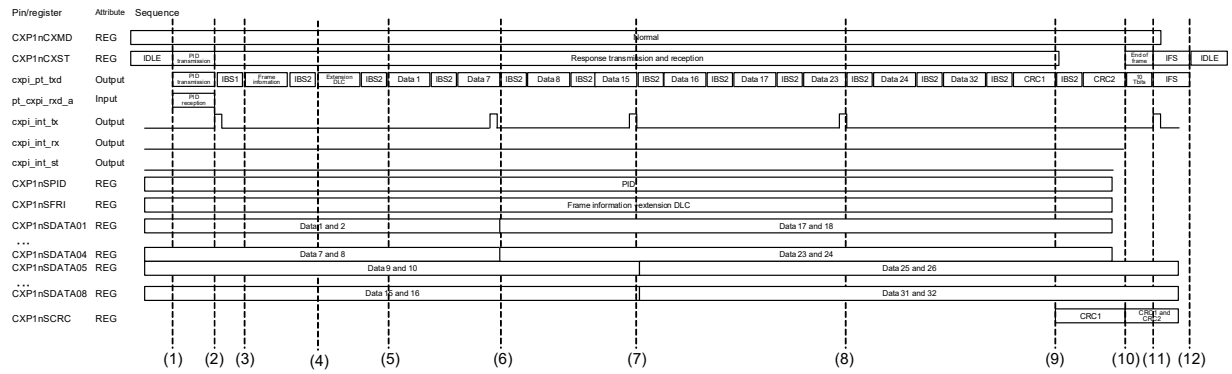
(6) CRC value transmission is completed. The transmitted CRC value is stored in CXP1nSCRC.

(7) 10 Tbits (fixed value determined by the hardware) have passed after transmitting the CRC value.
A frame transmit completion interrupt is generated and CXP1nINT.CXP1nSFRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSFRINT to 0.

(8) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

(b) Long Frame

The following provides an overview of the hardware and software operation when the frame structure is the long frame structure.



Note: IBS: IBS controlled by another node
IBS1: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 bit.
Then, the next data is transmitted.
IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS2 bit.
Then, the next data is transmitted.

Figure 30.14 Operation Sequence from PID Transmission to Response Transmission (Long Frame)

Operation

Operation setting: Frame structure: Long frame
(CXP1nSFRI.CXP1nSDLC1 = F_H)

Number of data bytes in the response: 32 bytes
(CXP1nSFRI.CXP1nSDLC 2 = 20_H)

(1) Writes 1 to CXP1nSND.CXP1nSFRREQ and makes a frame transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because CXP1nFLW1.CXP1nCXST = 010_B (IDLE).
(If CXP1nFLW1.CXP1nCXST is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PID transmission completion interrupt is generated. CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) Frame information is transmitted. (Data in CXP1nSFRI[7:0] is transmitted.)

(4) Extension DLC is transmitted. (Data in CXP1nSFRI[15:0] is transmitted.)

(5) The first-byte data is transmitted. (Data in CXP1nSDATA01[7:0] is transmitted.)

NOTE

Then, data in CXP1nSDATAm (m = 01 to 08) is transmitted on a byte basis.

(6) The 7 th-byte data transmitted.
A received data save request interrupt is generated and CXP1nSFQINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nSFQINT to 0.
The 16th-byte to 24th-byte data is set in CXP1nSDATA01 to CXP1nSDATA04.

(7) The 15 th-byte data transmitted.
A transmission data setting request interrupt is generated and CXP1nSFQINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nSFQINT to 0.
The 25th-byte to 32nd-byte data is set in CXP1nSDATA05 to CXP1nSDATA08.

(8) The 23th-byte data transmitted.
A transmission data setting request interrupt is generated and CXP1nSFQINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nSFQINT to 0.

NOTE

This processing flow handles 32-byte transmission. It does not set the transmission data.

(9) CRC1 transmission is completed. (The CRC value calculated within the circuit is transmitted.)
The transmitted CRC value is stored in CXP1nSCRC.

(10) CRC2 transmission is completed. (The CRC value calculated within the circuit is transmitted.)
The transmitted CRC value is stored in CXP1nSCRC.

(11) 10 Tbits (fixed value determined by the hardware) have passed after transmitting the CRC value.
A frame transmission completion interrupt is generated and CXP1nINT.CXP1nSFRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSFRINT to 0.

(12) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after CRC value transmission is completed, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

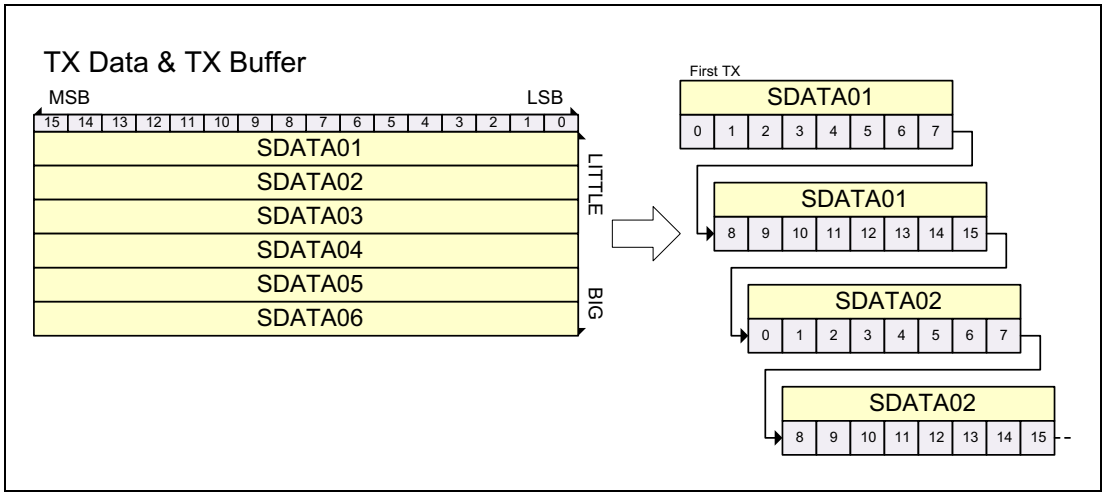


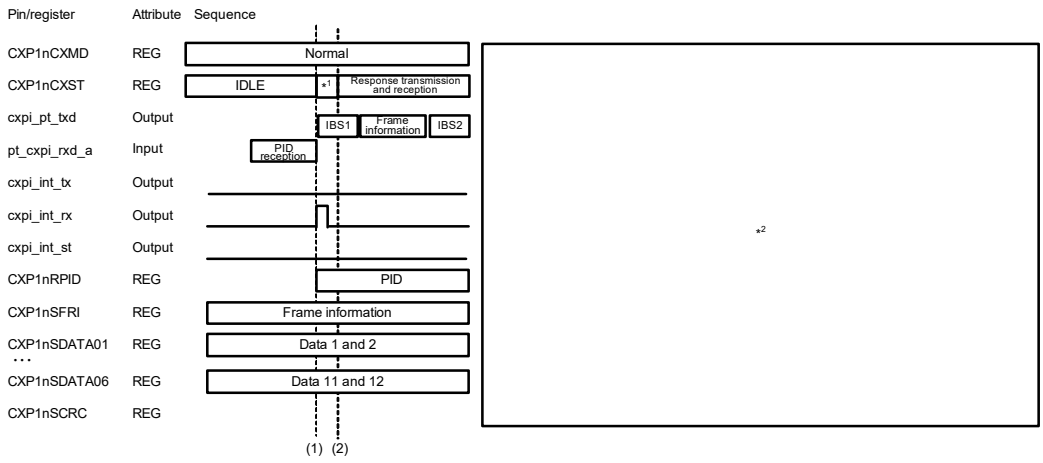
Figure 30.15 Transmit DATA and Transmit BUFFER

30.5.4.3 Manual Response

(1) PID Reception — Response Transmission

The CXP1 can transmit a response through software judgment when the PID of its own node was specified by another node. A normal frame or a long frame can be transmitted as a response according to the frame information (DLC) setting.

The following provides an overview of the hardware and software operation in the sequence from PID reception to response transmission (manual response).



Note: IBS: IBS controlled by another node
IBS1: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 bit. Then, the next data is transmitted.
IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS2 bit. Then, the next data is transmitted.

Note 1. Waiting for response judgment/response data setting.
Note 2. The same operation as step (3) and the rest in **Figure 30.13**.
The same operation as step (3) and the rest in **Figure 30.14**.

Figure 30.16 PID Reception — Response Transmission (Manual Response)

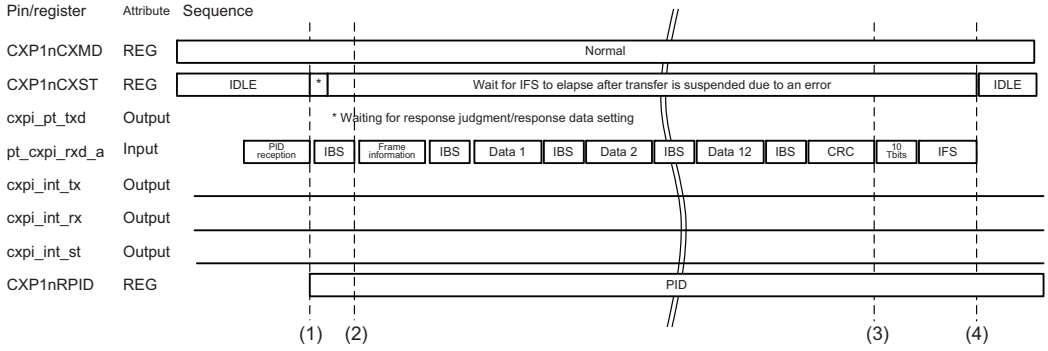
Operation

(1) A PID is received.
When the received PID matches the PID set in CXP1nRPIDF1m (m = 01 to 12) or CXP1nRPIDF2m (m = 01 to 04), a PID reception completion interrupt is generated and CXP1nINT.CXP1nRIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRIDINT to 0.
The software determines if the received PID is to be responded.
If the PID is to be responded, the software sets CXP1nSFRI (frame information) and CXP1nSDATA01 to CXP1nSDATA08 (transmission data), writes 1 to CXP1nSND.CXP1nSRSREQ, and makes a response transmission request.
(If the PID is to be received, the software receives the response.) For details, see **Figure 30.19** and **Figure 30.20**.

(2) Transmission of response data is started.

(2) PID Reception — Own Node is Not the Target

The following provides an overview of the hardware operation performed when the received PID is not for the own node and a response is not made or received. This sequence does not involve software processing. (No notification is made from the hardware.)



Note: IBS: IBS controlled by another node

Figure 30.17 Operation Sequence from PID Reception when Own Node is Not the Target

Operation

(1) A PID is received. No PID reception completion interrupt is generated because the received PID does not match the frame ID in CXP1nRPIDF1m (m = 01 to 12) or CXP1nRPIDF2m (m = 01 to 04).

(2) Response reception is started.
The received data is not stored in CXP1nRFRI and CXP1nRDATAm (m = 01 to 06).

(3) CRC reception is completed.
The received CRC is not stored in CXP1nRCRC.

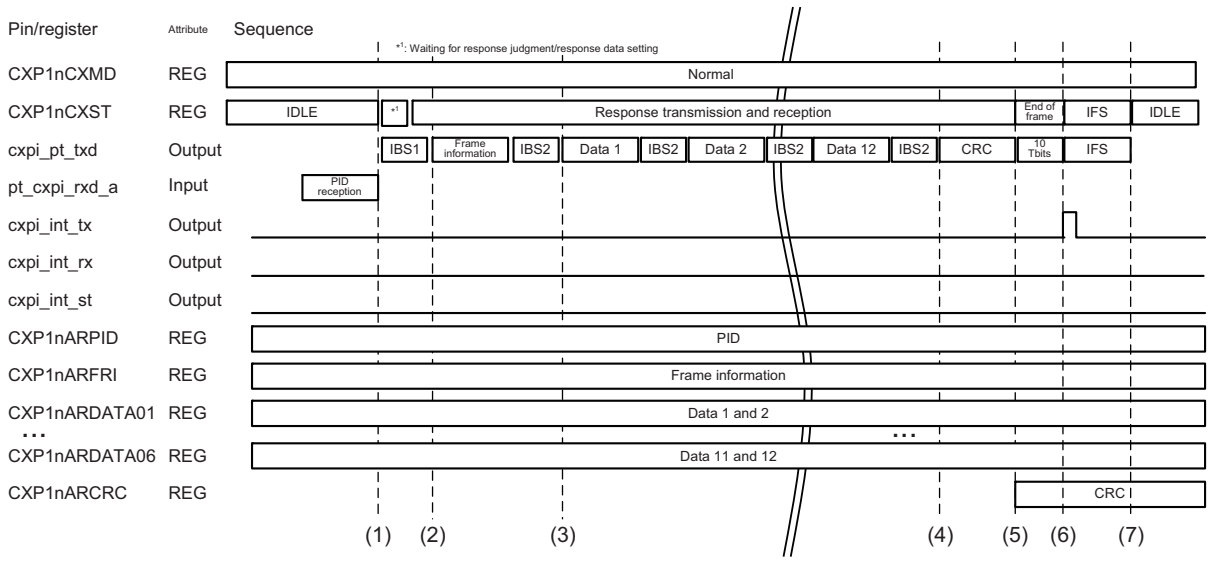
(4) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time from CRC value is received, and the state changes to the idle state (CXP1nFLW1.CXP1nCXST=010_B) after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed. Transition is possible.

30.5.4.4 Automatic Response

(1) PID Reception to Response Transmission

The CXPI can makes an automatic response upon receiving a PID by enabling the automatic response (CXP1nMODE.CXP1nARESEN = 1_B) and setting the automatic response PID in CXP1nARPID.

Only the normal frame is used as the frame structure of automatic response. The long frame is not supported.



Note: IBS: IBS controlled by another node
IBS1: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 bit. Then, the next data is transmitted.
IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS2 bit. Then, the next data is transmitted.

Figure 30.18 PID Reception — Response Transmission (Automatic Response)

Operation

Operation setting: Automatic response enabled: CXP1nMODE.CXP1nARESEN = 1_B

Frame structure: Normal frame (CXP1nARFRI.CXP1nASDLC ≠ F_H)

Number of data bytes in the response: 12 bytes (CXP1nARFRI.CXP1nASDLC = C_H)

(1) A PID is received. When the received PID matches the PID set in CXP1nARPID, automatic response is started.
No PID reception completion interrupt is generated when the received PID matches the PID set in CXP1nARPID.

(2) Frame information is transmitted. (Data in CXP1nARFRI[7:0] is transmitted.)

(3) The first-byte data is transmitted. (Data in CXP1nARDATA01[7:0] is transmitted.)

NOTE

Then, data in CXP1nARDATAn (n = 01 to 06) is transmitted on a byte basis.

(4) The CRC value is transmitted. (The CRC value calculated within the circuit is transmitted.)

(5) CRC value transmission is completed. The transmitted CRC value is stored in CXP1nARCRC.

(6) 10 Tbits have passed after transmitting the CRC value.
A frame retransmission completion interrupt is generated and CXP1nINT.CXP1nSFRINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSFRINT to 0.

(7) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after CRC value transmission is completed, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed. Communication is possible after the state transitions to the idle state.

(2) PID Reception — Own Node is Not the Target

The operation is the same as for manual response.

For details, see **Section 30.5.4.3(2), PID Reception — Own Node is Not the Target**.

30.5.4.5 PID Reception

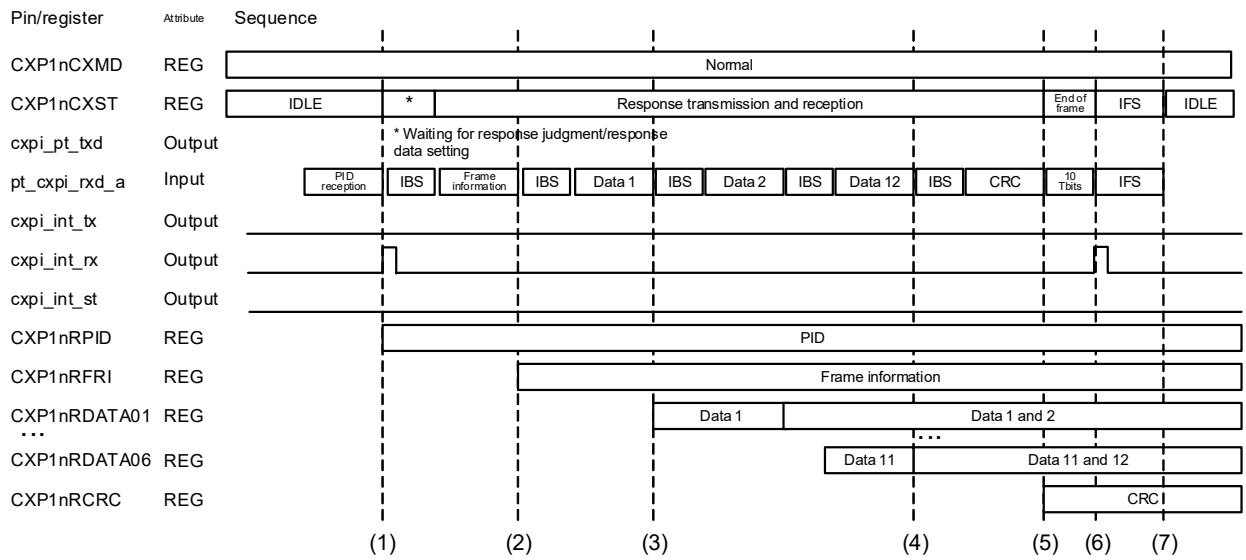
(1) PID Reception — Response Reception (Frame Reception)

The CXPI can receive a response from another node when the PID of its own node was specified by another node. A normal frame or a long frame is received as a response according to the received frame information (DLC).

This section provides an overview of the hardware and software operation in each frame structure.

(a) Normal Frame

The following provides an overview of the hardware and software operation when the frame structure is the normal frame structure.



Note: IBS: IBS controlled by another node

Figure 30.19 Operation Sequence from PID Reception to Response Reception (Normal Frame)

Operation

Operation setting: Frame structure: Normal frame
($CXP1nRFRI.CXP1nRDLC1 \neq F_H$)*1

Number of data bytes in the response: 12 bytes
($CXP1nRFRI.CXP1nRDLC1 = C_H$)*1

Note 1. The frame structure and the number of data bytes are determined based on the received data.

(1) A PID is received. When the received PID matches the PID set in CXP1nRPIDF1m (m = 01 to 12) or CXP1nRPIDF2m (m = 01 to 04), a PID reception completion interrupt is generated and CXP1nINT.CXP1nRIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRIDINT to 0.

(2) Frame information is received, and is stored in CXP1nRFRI.
The frame structure and the number of data bytes (12) are determined based on the received data (Frame information).

(3) The first-byte data is received. The received data is stored in CXP1nRDATA01.

NOTE

After that, data is stored in CXP1nRDATAm (m = 01 to 06) as each byte is received.

(4) The 12th-byte data is received. The received data is stored in CXP1nRDATA06.

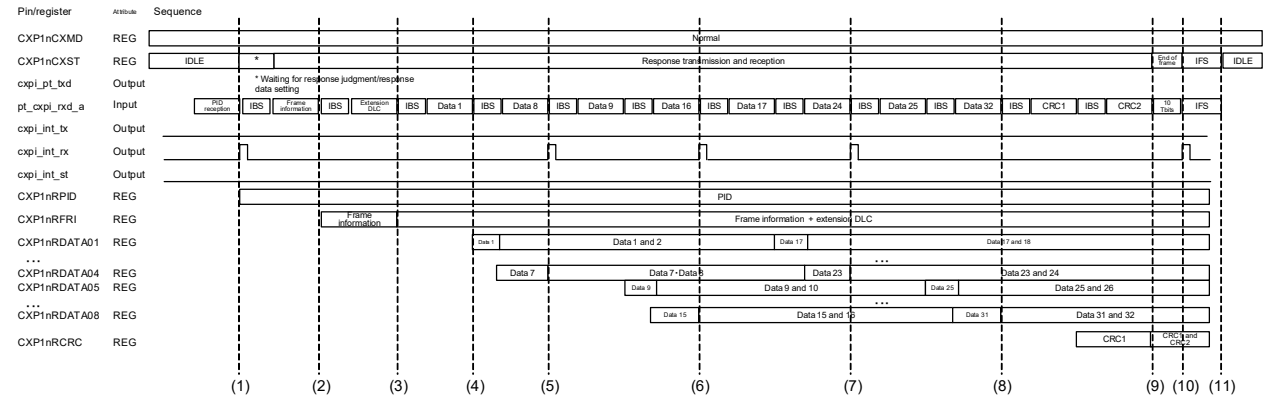
(5) The CRC value is received. The CRC value is stored in CXP1nRCRC.

(6) 10 Tbits (fixed value determined by the hardware) have passed after receiving the CRC value.
A frame reception completion interrupt is generated and CXP1nINT.CXP1nRFRINT is set to 1.
Reception is possible.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFRINT to 0.
Frame information can be read from CXP1nRFRI.
Received data can be read from CXP1nRDATAm (m = 01-06).

(7) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time from CRC value is received, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Transition is possible.

(b) Long Frame

The following provides an overview of the hardware and software operation when the frame structure is the long frame structure.



Note: IBS: IBS controlled by another node

Figure 30.20 Operation Sequence from PID Reception to Response Reception (Long Frame)

Operation

Operation setting: Frame structure: Long frame
(CXP1nRFRI.CXP1nRDLC1 = F_H)*1

Number of data bytes in the response: 32 bytes
(CXP1nRFRI.CXP1nRDLC2 = 20_H)*1

Note 1. The frame structure and the number of data bytes are determined based on the received data.

(1) A PID is received. When the received PID matches the PID set in CXP1nRPIDF1m (m = 01 to 12) or CXP1nRPIDF2m (m = 01 to 04), a PID reception completion interrupt is generated and CXP1nINT.CXP1nRIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRIDINT to 0.

(2) Frame information is received, and is stored in CXP1nRFRI.

(3) Extension DLC is received, and is stored in CXP1nRFRI[15:8].
The frame structure (long frame) and the number of data bytes (32) are determined based on the received data (Frame information).

(4) The first-byte data is received. The received data is stored in CXP1nRDATA01[7:0].
Then, data is stored in CXP1nRDATAm (m = 01 to 04) as each byte is received.
(See **Figure 30.11, Received DATA and Received BUFFER**)

(5) The 8th-byte data is received. The received data is stored in CXP1nRDATA04[15:8].
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
Then, data is stored in CXP1nRDATAm (m = 05 to 08) as each byte is received.
(See **Figure 30.11, Received DATA and Received BUFFER**)
The software reads CXP1nRDATA01 to CXP1nRDATA04 and saves the received data.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFQINT to 0.

(6) The 16th-byte data is received. The received data is stored in CXP1nRDATA08[15:8].
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
Then, data is stored in CXP1nRDATAm (m = 01 to 04) as each byte is received.
(See **Figure 30.11, Received DATA and Received BUFFER**)
The software reads CXP1nRDATA05 to CXP1nRDATA08 and saves the received data.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFQINT to 0.

(7) The 24th-byte data is received. The received data is stored in CXP1nRDATA04[15:8].
A received data save request interrupt is generated and CXP1nINT.CXP1nRFQINT is set to 1.
Then, data is stored in CXP1nRDATAm (m = 05 to 08) as each byte is received.
(See **Figure 30.11, Received DATA and Received BUFFER**)
The software reads CXP1nRDATA01 to CXP1nRDATA04 and saves the received data.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFQINT to 0.

(8) The 32nd-byte data is received. The received data is stored in CXP1nRDATA08[15:8].
No received data save request interrupt is generated because this is the last data.

(9) The CRC value is received. The received CRC is stored in CXP1nRCRC.

(10) 10 Tbits have passed after receiving the CRC value.
A frame reception completion interrupt is generated and CXP1nINT.CXP1nRFRINT is set to 1.
The software reads CXP1nRDATAm(05-08) and frame information (CXP1nRFRI).
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRFRINT to 0.

(11) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time from CRC value is received, and the state changes to the idle state (CXP1nFLW1.CXP1nCXST=010B) after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Transition is possible.

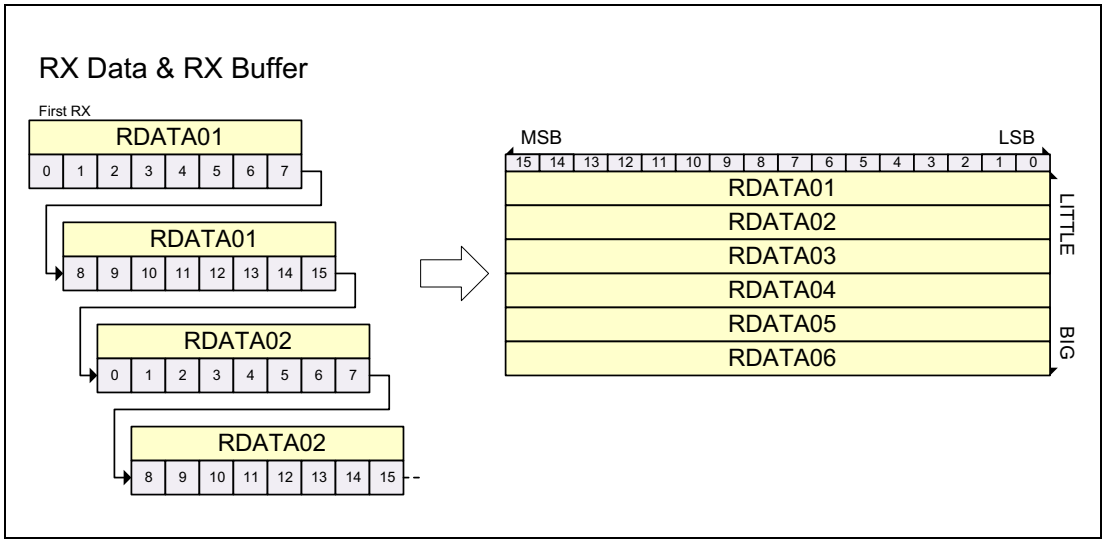


Figure 30.21 Received DATA and Received BUFFER

(2) PID Reception — Own Node is Not the Target

The operation is the same as that for manual response.

For details, see **Section 30.5.4.3(2), PID Reception — Own Node is Not the Target**.

30.5.4.6 PTYPE Transmission

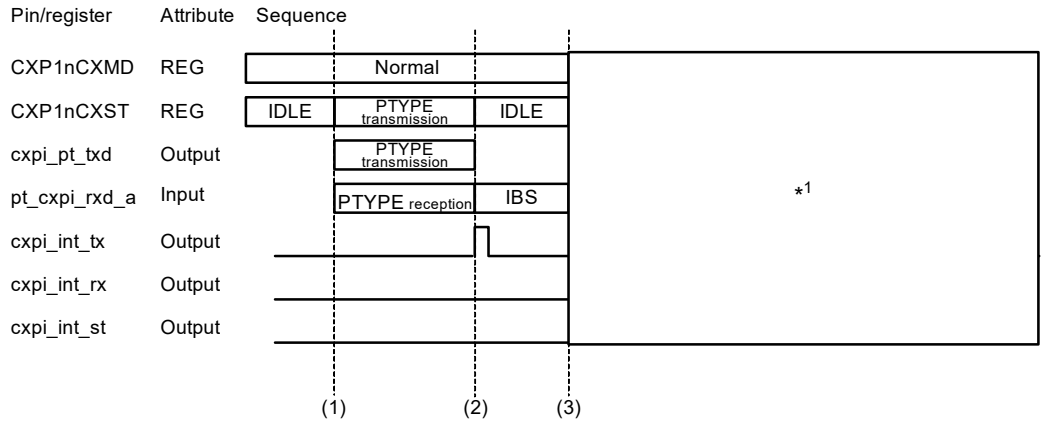
The PTYPE field is used to allow the master node to enable a slave node to transmit a PID in the polling method. The slave node can transmit a PID only during the IBS time immediately after receiving the PTYPE field.

In the event trigger method, the PTYPE field cannot be transmitted because it is protected by the hardware. Also, the slave node cannot transmit the PTYPE field because it is protected by the hardware. This section provides an overview of the hardware and software operation performed in each frame structure when the master node transmits a PTYPE field.

(1) When There is a Response from the Slave Node after Transmitting PTYPE

The following provides an overview of the hardware and software operation for PTYPE transmission.

The following figure shows the sequence when there is a response from the slave node after transmitting the PTYPE field.



Note: IBS: IBS controlled by another node

Note 1. The same operation as in **Figure 30.16**, **Figure 30.18**, **Figure 30.19**, or **Figure 30.20**.

Figure 30.22 Operation Sequence of PTYPE Transmission (When There is a Response)

Operation

(1) Writes 1 to CXP1nSND.CXP1nSPTREQ and makes a PTYPE transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because CXP1nFLW1.CXP1nCXST = 010_B (IDLE).
(If CXP1nFLW1.CXP1nCXST is not 010_B (IDLE), the hardware puts transmission on hold.)

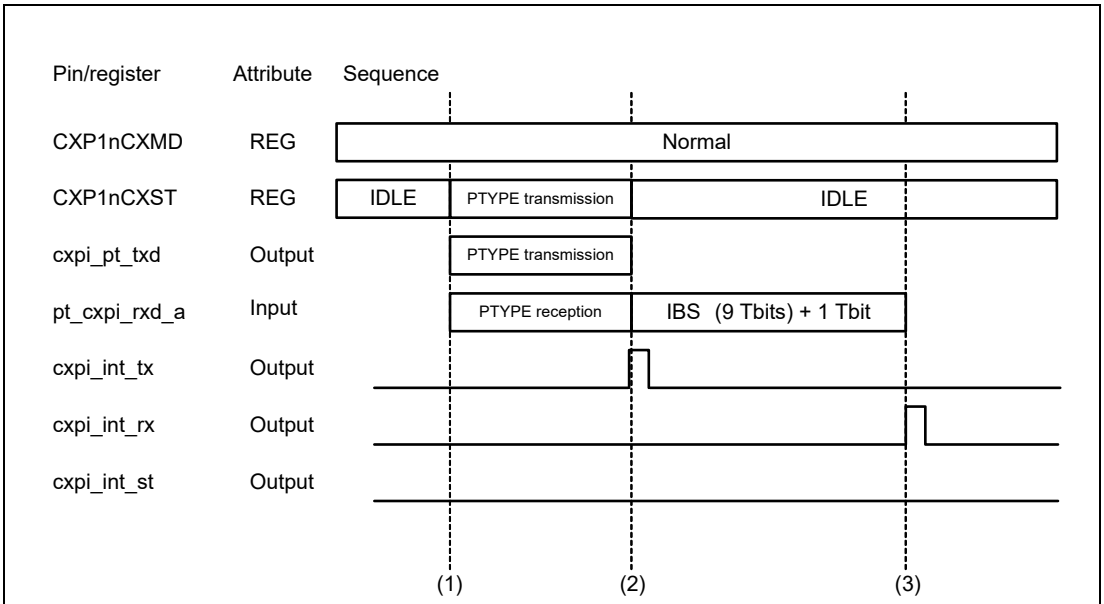
(2) A PTYPE transmission completion interrupt is generated and CXP1nINT.CXP1nSPTINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSPTINT to 0.

(3) The sequence after * is the same as shown in the timing charts indicated below the figure.

(2) When There is No Response from the Slave Node after Transmitting PTYPE

The following provides an overview of the hardware and software operation for PTYPE transmission.

The following figure shows the sequence when there is no response from the slave node after the master node transmits a PTYPE field.



Note: IBS: IBS controlled by another node

Figure 30.23 Operation Sequence of PTYPE Transmission (When There is No Response)

Operation

(1) Writes 1 to CXP1nSND.CXP1nSPTREQ and makes a PTYPE transmission request. For control, see **Section 30.6.2.4, Restriction of Transmission Request**.
It starts transmission immediately because CXP1nFLW1.CXP1nCXST = 010_B (IDLE).
(If CXP1nFLW1.CXP1nCXST is not 010_B (IDLE), the hardware puts transmission on hold.)

(2) A PTYPE transmission completion interrupt is generated and CXP1nINT.CXP1nSPTINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSPTINT to 0.

(3) If the start bit of the PID is not received within 10 Tbits after PTYPE transmission is completed, a PTYPE valid period completion interrupt is generated and CXP1nINT.CXP1nREXINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nREXINT to 0.

After detecting PTYPE validity completion interrupt after a PTYPE transmission, the transmission the IDLE state saves and by which PID and a frame is possible by a CXPI controller. Therefore send PID or a frame after a IFS period on the bus is measured by software.

30.5.4.7 PTYPE Reception

A PTYPE field is used to allow the master node to enable a slave node to transmit a PID in the polling method. The slave node can transmit a PID only during the IBS time immediately after receiving a PTYPE field.

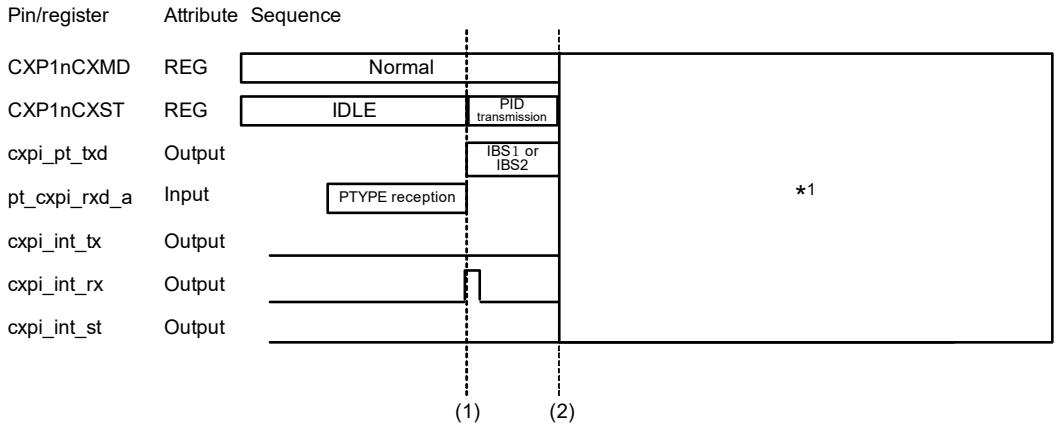
It is not used in the event trigger method.

This section provides an overview of the hardware and software operation performed in each frame structure when the slave node receives a PTYPE field.

(1) When Transmitting PID after Receiving PTYPE

The following provides an overview of the hardware and software operation for PTYPE reception.

The figure below shows the sequence of PID transmission after receiving a PTYPE field.



Note: IBS1 or IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 or CXP1nFRMW.CXP1nIBS2 bit. Then, the next data is transmitted.

Note 1. The selection of IBS1 or IBS2 is as follows:
IBS2 to transmit a frame in *1 (CXP1nSFRREQ = 1)
IBS1 to transmit a PID in *1 (CXP1nPIDREQ = 1)
*1: The same operation as in **Figure 30.9**, **Figure 30.10**, **Figure 30.13**, or **Figure 30.14**.

Figure 30.24 Operation Sequence of PTYPE Reception (PID is Transmitted)

Operation

(1) PTYPE is received.
A PTYPE reception completion interrupt is generated and CXP1nINT.CXP1nRPTINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRPTINT to 0.

(2) The sequence after *1 is the same as shown in the timing charts indicated below the figure.

(2) When Not Transmitting PID after Receiving PTYPE

The following provides an overview of the hardware and software operation for PTYPE reception.

The figure below shows the sequence of not transmitting PID after receiving a PTYPE field.

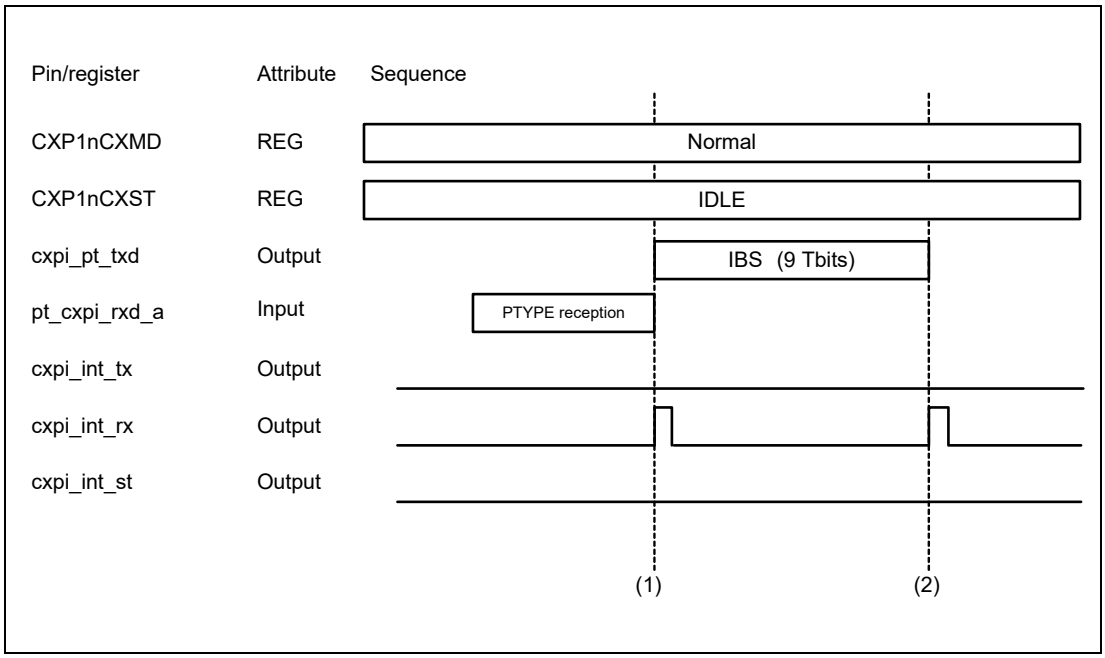


Figure 30.25 Operation Sequence of PTYPE Reception (PID is Not Transmitted)

Operation

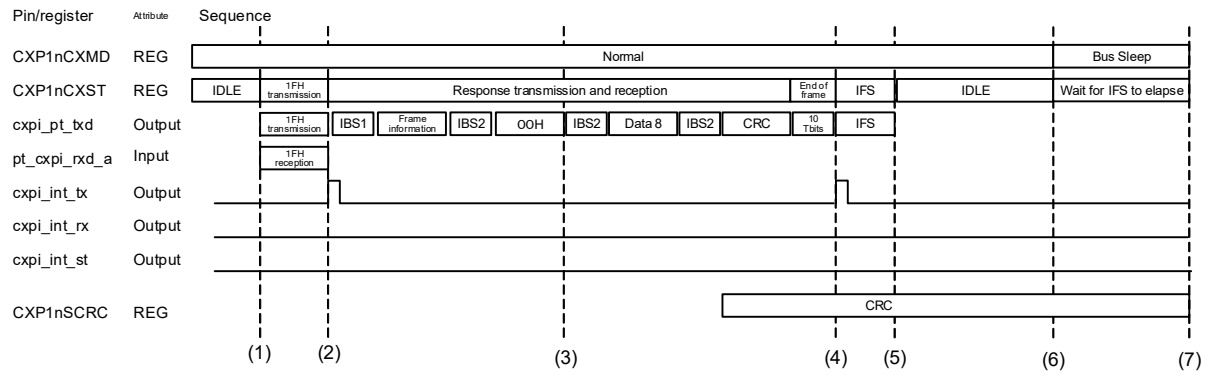
(1) A PTYPE field is received. A PTYPE reception completion interrupt is generated and CXP1nINT.CXP1nRPTINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRPTINT to 0.

(2) If no PID transmission request is made within 9 Tbits (fixed value determined by the hardware) after PTYPE reception is completed, a PTYPE valid period completion interrupt is generated and CXP1nINT.CXP1nREXINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nREXINT to 0.

30.5.4.8 Sleep Frame Transmission Operation

The master node uses a sleep frame to transition each slave node from normal mode to bus sleep mode. After a sleep frame is transmitted and received, the master node and each slave node enter bus sleep mode.

This section provides an overview of the hardware and software operation performed in each frame structure when the master node transmits a sleep frame.



Note: IBS: IBS controlled by another node
IBS1: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS1 bit. Then, the next data is transmitted.
IBS2: Counting is performed at the CXPI nominal bit (Tbit) time, and logical 1 is output for the time specified by the CXP1nFRMW.CXP1nIBS2 bit. Then, the next data is transmitted.

Figure 30.26 Operation Sequence of Sleep Frame Transmission

Operation

(1) Setting sleep frame as following.
Writes 1 to CXP1nSND.CXP1nSFRREQ, and makes a sleep frame transmission request.
PID transmission.
(If CXP1nFLW1.CXP1nCXST is not 010_B (IDLE), the hardware puts transmission on hold.)

PID Field	Frame Information	Data	CRC
1 byte **NOTE**: The parity bit is the same as the normal frame. The frame ID is $1F_H$ (fixed).	1 byte **NOTE**: Same as the normal frame. DLC is fixed to 1000_B.	8 bytes **NOTE**: The first byte is 00_H (fixed). The second and following bytes are FF_H (fixed).	1 byte **NOTE**: Same as the normal frame.

(2) A PID transmission completion interrupt is generated and CXP1nINT.CXP1nSIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSIDINT to 0.

(3) Data 1 (00_H) is transmitted. At this time, the receiving side completes sleep frame reception.

(4) 10 Tbits (fixed value determined by the hardware) have passed after transmitting the CRC value.
A frame transmit completion interrupt is generated and CXP1nINT.CXP1nSFRINT is set to 1.
Reception is possible.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSFRINT to 0.

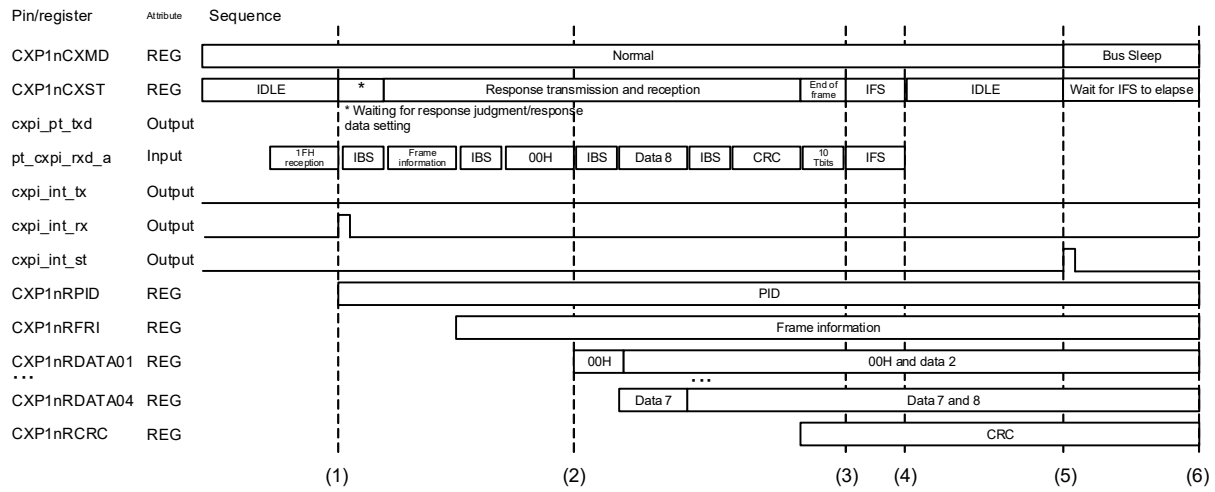
(5) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after CRC value transmission is completed, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

(6) Stop clock.
Write CXP1nWUP.CXP1nCLKOFF to 1, and the CXPI enters bus sleep mode.

30.5.4.9 Sleep Frame Reception Operation

The master node uses a sleep frame to transition each slave node from normal mode to bus sleep mode. After a sleep frame is transmitted and received, the master node and each slave node enter bus sleep mode.

This section provides an overview of the hardware and software operation performed in each frame structure when the slave node receives a sleep frame.



Note: IBS: IBS controlled by another node

Figure 30.27 Operation Sequence of Sleep Frame Reception

Operation

(1) A PID is received. If PID = $1F_H$ is received, a PID reception completion interrupt is generated even if there is no matched PID in CXP1nRPIDF1m (m = 01 to 12) and CXP1nRPIDF2m (m = 01 to 04).
A PID reception completion interrupt is generated and CXP1nINT.CXP1nRIDINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nRIDINT to 0.

(2) The first-byte data (00_H) is received. At this time, it is determined that a sleep frame has been received.

NOTE

Data is stored in CXP1nRDATAm (m = 01 to 04) as each byte is received.
For the second and following bytes, an arbitrary value is received because no data value is specified.

(3) The CRC value is received. The CRC value is stored in CXP1nRCRC.
A sleep frame reception is completed. (No frame reception completion interrupt is generated.)

(4) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.

(5) The time specified in the CXP1nSLP.CXP1nSLPWAIT bits has passed since (3).
A sleep transition completion interrupt is generated and CXP1nINT.CXP1nSLPINT is set to 1.
Since an interrupt is output at a level, the software clears CXP1nINT.CXP1nSLPINT to 0.
(If it is a frame reception completion interrupt instead of a sleep transition completion interrupt, process it as a diagnostic request.)
The CXPI transitions from bus sleep mode to normal mode.

(6) Transit transceiver to the standby mode. (See transceiver specification).
The software outputs LOW to the NSLP pin of the CXPI transceiver. Implemented by an external function outside this IP.

NOTES

1. If a PID is received between (3) and (5), operation for reception proceeds, but the CXPI enters bus sleep mode forcibly after the sleep transition time has passed.
2. Don't transmit request or write 1 to CXP1nWUP.CXP1nCLKOFF between (3) and (5).

30.5.4.10 Wakeup Pulse Transmission Operation

The CXPI can wake up other nodes by sending a wakeup pulse.

The wakeup pulse transmission operation differs depending on the transfer mode and master node/ slave node selection.

This section describes the wakeup pulse transmission operation under each operating condition.

(1) CXPI-PWM Mode

This section describes the wakeup pulse transmission operation in CXPI-PWM mode.

(a) Master Node

The CXPI in the master node wakes up a slave node by transmitting the clock signal.

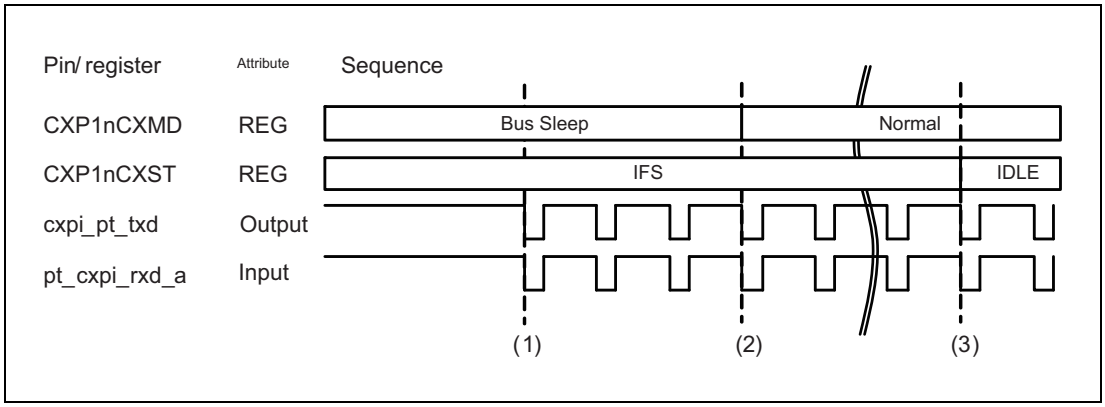


Figure 30.28 Wakeup Pulse Transmission Operation (Master Node in CXPI-PWM Mode)

Operation

(1) The software verifies that CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep mode).
It then writes 1 to the CXP1nWUP.CXP1nCLKON bit to request clock transmission.
It transmits the clock signal immediately after requesting clock transmission.

(2) The CXPI transitions from bus sleep mode to normal mode by detecting the looped-back clock signal after transmitting the clock signal. (Four or five falling edges of pt_cxpi_rxd_a are detected.)

(3) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after entering normal mode, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

(b) Slave Node

A slave node wakes up other nodes by sending a wakeup pulse.

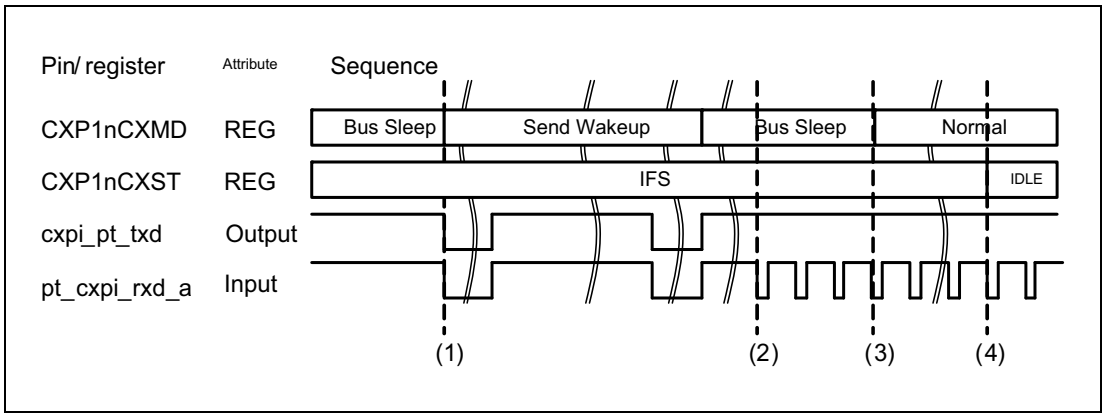


Figure 30.29 Wakeup Pulse Transmission Operation (Slave Node in CXPI-PWM Mode)

The pulse width of the wakeup pulse is specified by the CXP1nWU register.

Width of dominant level: CXPI nominal bit (Tbit) time × CXP1nWU.CXP1nWTXDOM

Width of recessive level: CXPI nominal bit (Tbit) time × CXP1nWU.CXP1nWTXREC

Operation

(1) The software verifies that CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep mode).
Then, it writes 1 to CXP1nWUP.CXP1nWUPREQ and makes a wakeup pulse transmission request.
It transmits a wakeup pulse immediately after requesting wakeup pulse transmission.
It keeps polling until CXP1nFLW1.CXP1nCXMD = 11_B (normal mode).

(2) The clock signal is input from the master node that has detected a wakeup pulse.

NOTE

If this IP does not enter normal mode within the specified time, restart from (1).

(3) This IP transitions from bus sleep mode to normal mode by detecting the clock signal. (Four falling edges of pt_cxpi_rxd_a are detected.)

NOTE

If detecting a clock signal while transmitting a wakeup pulse, the IP transitions from send wakeup mode to normal mode and stops the wakeup pulse transmission.

(4) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after entering normal mode, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

(2) CXPI-NRZ Mode

This section describes the wakeup pulse transmission operation in CXPI-NRZ mode.

(a) Master Node

In CXPI-NRZ mode, a wakeup pulse is transmitted by transmitting the clock signal to the CLK pin of the CXPI transceiver from the PWM output pin of the LSI.

The following provides an overview of the hardware and software operation under control of the LSI and CXPI transceiver.

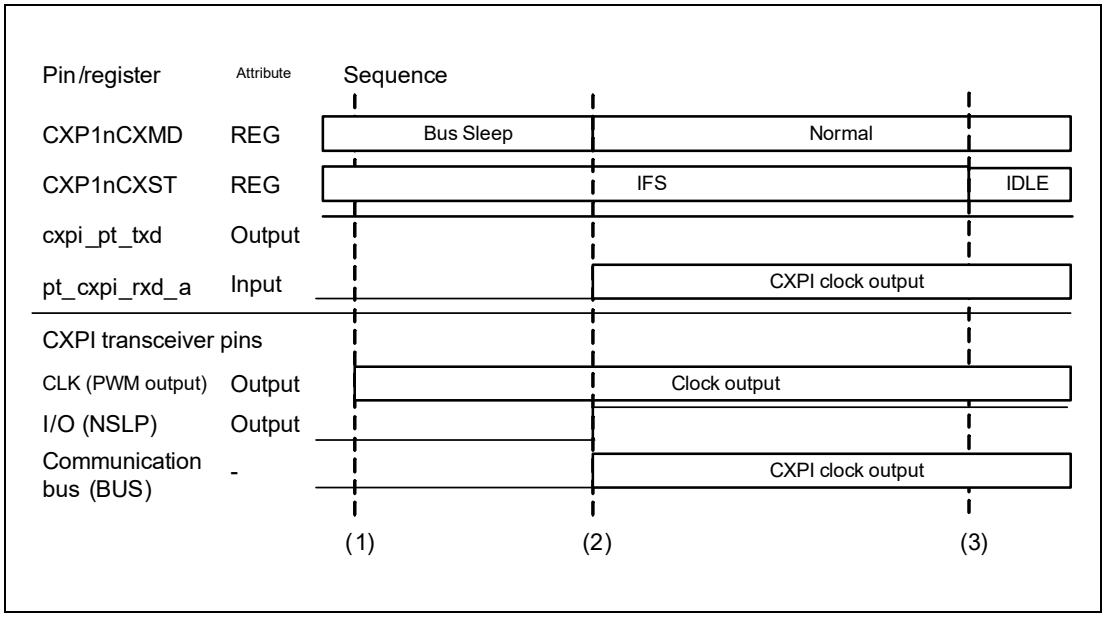


Figure 30.30 Wakeup Pulse Transmission Operation (Master Node in CXPI-NRZ Mode)

Operation

(1) The software starts the PWM macro and outputs bit rate cycles from the PWM output pin.*1

(2) The software outputs HIGH to the NSLP pin of the CXPI transceiver. *1
The CXPI transceiver outputs the CXPI clock to the communication bus because the NSLP pin went high.
The software verifies that CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep mode).
It then writes 1 to the CXP1nWUP.CXP1nCLKON bit to notify clock detection.
This IP transitions from bus sleep mode to normal mode by notifying the clock detection.

(3) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after entering normal mode, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed.
Communication is possible after the state transitions to the idle state.

Note 1. Implemented by a module other than this IP.

(b) Slave Node

In CXPI-NRZ mode, the CXPI transceiver detects the clock on the communication bus and the IP detects the clock by the clock signal output from the CLK pin or RXD pin.

The following provides an overview of the hardware and software operation under control of the LSI and CXPI transceiver.

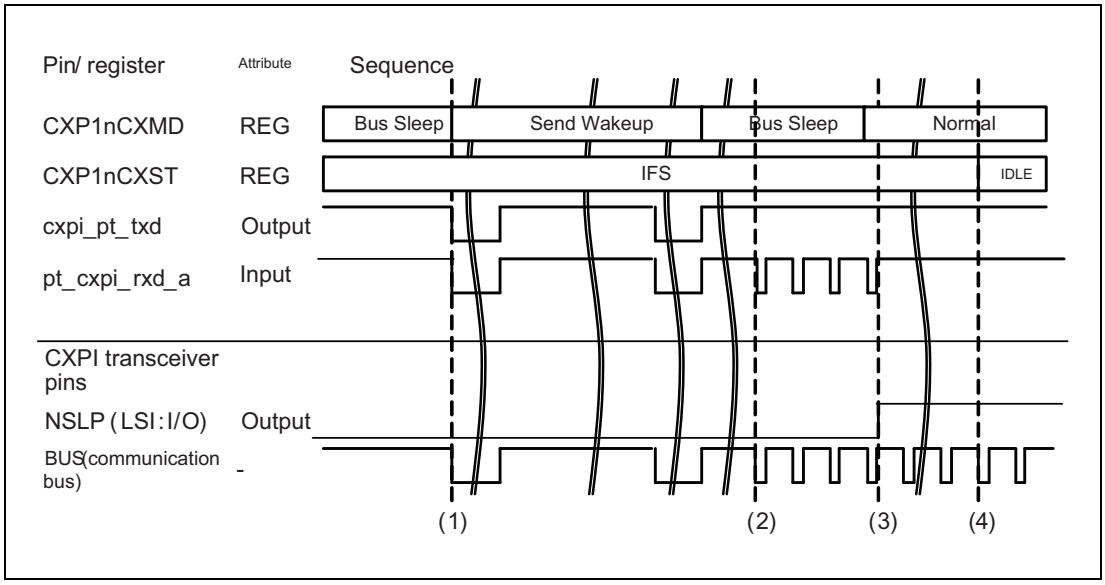


Figure 30.31 Wakeup Pulse Transmission Operation (Slave Node in CXPI-NRZ Mode)

The pulse width of the wakeup pulse is specified by the CXP1nWU register.

Width of dominant level: CXPI nominal bit (Tbit) time × CXP1nWU.CXP1nWTXDOM

Width of recessive level: CXPI nominal bit (Tbit) time × CXP1nWU.CXP1nWTXREC

Operation

(1) The software verifies that CXP1nFLW1.CXP1nCXMD = 01_B (bus sleep mode).
Then, it writes 1 to CXP1nWUP.CXP1nWUPREQ and makes a wakeup pulse transmission request.
It transmits a wakeup pulse immediately after requesting wakeup pulse transmission.

(2) The clock signal is input from the master node that has detected a wakeup pulse.

NOTE

If this IP does not enter normal mode within the specified time, restart from (1).

(3) An external function outside this IP detects the CXPI clock (the clock detection sequence differs from one system to another).
The software writes 1 to the CXP1nWUP.CXP1nCLKON bit to notify clock detection.
This IP transitions from bus sleep mode to normal mode after notifying clock detection.(Four or five falling edges of pt_cxpi_rxd_a are detected.)

NOTE

The software outputs High to the NSLP pin of the CXPI transceiver (this is implemented by a module other than this IP).

(4) Wait for IFS to elapse.
Counting is started at the CXPI nominal bit (Tbit) time after entering normal mode, and the state changes to the idle state after the time specified by the CXP1nFRMW.CXP1nIFS bit has passed. Communication is possible after the state transitions to the idle state.

30.5.4.11 Wakeup Pulse Reception Operation

This IP does not have the wakeup pulse reception function. It detects a wakeup pulse by using the CXPI transceiver and a module other than this IP. For the detailed operation, see **Section 30.5.1, CXPI System Configuration**.

30.5.5 Basic Functions

This section describes the basic functions defined in the Protocol Specification.

30.5.5.1 Baud Rate

The CXPI baud rate can be selected from the following. For the detailed baud rate setting, see **Section 30.4.1.2, CXP1nBRT — CXPI Baud Rate Setting Register**.

- 9.6 kbits/s
- 10.4 kbits/s
- 19.2 kbits/s
- 20 kbits/s

30.5.5.2 Node Selection

The CXP1nMODE.CXP1nMS bit allows node selection from the master node and the slave node.

For the hardware function differences by node selection, see **Section 30.5.3, Functions**.

30.5.5.3 Clock Sharing

For details on clock sharing, see *Section 5.1.4* in *the JASO D015-3 Protocol Specification*.

30.5.5.4 Access Method

The CXPI access method can be selected from the event trigger method and the polling method.

For details on the access method, see **Section 30.5.2, CXPI Operation Overview**.

30.5.5.5 PWM Encoding/Decoding

The PWM encoding/decoding function is used to output and sample the PWM waveform.

The table below lists the functions for PWM encoding/decoding.

Table 30.73 List of Functions for PWM Encoding/Decoding

No.	Function		Description
1	PWM waveform generation	Generation of logical value 1	For details, see **Section 30.4.1.3, CXP1nV1LW — CXPI Logical Value 1 Setting Register**.
2		Generation of logical value 0	For details, see **Section 30.4.1.4, CXP1nV0LW — CXPI Logical Value 0 Setting Register**.
3	PWM waveform sampling		For details, see **Section 30.4.1.5, CXP1nSAMP — CXPI Sampling Setting Register**.

Note that PWM encoding/decoding does not work as hardware in CXPI-NRZ mode.

The communication operation in CXPI-NRZ mode is as follows:

In CXPI-NRZ mode, UART communication is performed and the CXPI transceiver handles PWM encoding/decoding.

The UART reception operation in the CXPI-NRZ mode performs sampling at the center of the baud rate set by the CXP1nBRT register.

NOTE

The CXP1nSAMP register in which the sampling timing is set is valid only in CXPI-PWM mode.
It sets the sampling timing for the PWM waveform.

30.5.6 Data Functions

This section describes the PID and data buffers defined in the Protocol Specification.

30.5.6.1 PID

The table below lists the registers related with the PID. For detailed operation, see the relevant section.

Table 30.74 Registers Related to PID

Register Name	Description	Described in Section
CXP1nSPID	Sets the transmitting PID when transmitting a PID and frame.	**30.4.1.21**
CXP1nRPID	Stores the received PID when a PID is received.	**30.4.1.20**
CXP1nRPIDF1m (m = 01 to 12)	Sets the PID to be received and responded to. Sets the PID to be transmitting.	**30.4.1.22**
CXP1nRPIDF2m (m = 01 to 04)	Sets the PID to be received and responded to. **NOTE:** A frame ID filtering function is provided.	**30.4.1.23**
CXP1nARPID	Sets the PID for automatic response.	**30.4.1.24**

30.5.6.2 Data Buffer

The table below lists the data buffers.

Table 30.75 Data Buffers

Type	No. of Bytes	Register Name	Usage
Transmission	16 bytes	CXP1nSDATAm (m = 01 to 08)	Normal frame (see **Section 30.5.4.2(1)(a)**) Long frame (see **Section 30.5.4.2(1)(b)**)
Reception	16 bytes	CXP1nRDATAm (m = 01 to 08)	Normal frame (see **Section 30.5.4.5(1)(a)**) Long frame (see **Section 30.5.4.5(1)(b)**)
Automatic Response	12 bytes	CXP1nARDATAm (m = 01 to 06)	Normal frame (see **Section 30.5.4.4(1)**) The long frame is not supported.

30.5.6.3 Number of Data Byes in the Response

The table below lists the number of data byes in the response for each type of communication.

Table 30.76 Number of Data Byes in the Response

Type	Frame Structure	Number of Data Byes in the Response
Transmission/manual response	Normal frame	0 byte to 12 bytes
	Long frame	0 byte to 255 bytes
	Sleep frame	Fixed to 8 bytes
Automatic response	Normal frame	0 byte to 12 bytes
Reception	Normal frame	0 byte to 12 bytes
	Long frame	0 byte to 255 bytes
	Sleep frame	Fixed to 8 bytes

30.5.7 Control Functions

This section describes the functions to control the CXPI.

30.5.7.1 Transmission

The table below lists the functions provided by this IP for transmission control.

For how to use them, see the relevant section.

Table 30.77 Control Functions

Transmission Control Functions	Register that Makes a Transmission Request	Described in Section
CXPI clock transmission	CXP1nWUP.CXP1nCLKON	**30.5.4.10**
PID transmission	CXP1nSND.CXP1nPIDREQ	**30.5.4.1**
Frame transmission	CXP1nSND.CXP1nSFRREQ	**30.5.4.2**
Automatic response of the response when receiving a PID	— (Automatic response is enabled when the received PID matches the value in CXP1nARPID.)	**30.5.4.4**
Manual response of the response when receiving a PID	CXP1nSND.CXP1nSRSREQ	**30.5.4.3**
PTYPE transmission	CXP1nSND.CXP1nSPTREQ	**30.5.4.6**

30.5.7.2 IFS and IBS Setting Function

For details on the IFS and IBS setting function, see **Section 30.4.1.8, CXP1nFRMW — CXP1 Inter-Frame Setting Register**.

For the actual operation, see **Section 30.5.4, Overall Operation**.

30.5.7.3 Retransmission Processing

The IP module can perform retransmission processing automatically in response to a loss in arbitration or an error. It handles retransmission when a retransmission factor occurs in the communications listed in the table below.

Table 30.78 Targets and Factors of Retransmission Processing

No.	Target Communication	Retransmission Factor	
1	Frame transmission (PID transmission → response reception)	During PID transmission	Transmission stopped because of a loss in arbitration
2			Transmission stopped because a bit error occurred
3			Transmission stopped because a parity error occurred
4			Transmission stopped because a framing error occurred
5		During response transmission	Transmission stopped because a bit error occurred
6			Transmission stopped because a framing error occurred
7			Transmission stopped because a data length error occurred
8	PID transmission (PID transmission → response reception)	During PID transmission	Transmission stopped because of a loss in arbitration
9			Transmission stopped because a bit error occurred
10			Transmission stopped because a parity error occurred
11			Transmission stopped because a framing error occurred
12		During response reception	Reception stopped because a data length error occurred
13			Reception stopped because a CRC error occurred
14			Reception stopped because a framing error occurred
15			Reception stopped because an overrun error occurred

The table below lists the functions for retransmission processing.

Table 30.79 Functions for Retransmission Processing

No.	Retransmission Function	Related Registers
1	Two modes of retransmission not judged (mode in which the number of retransmissions (retries) is fixed to unlimited) and retransmission judged (mode in which Nos. 2 and 3 below are valid)	CXP1nMODE.CXP1nREPOP
2	The number of retransmissions (retries) can be specified. As the number of retries, 0, 1, 2 or unlimited can be specified.	CXP1nSND.CXP1nREPSEL CXP1nREP
3	Indication of retransmission status (remaining number of retransmissions, etc.)	CXP1nFLG2
4	Retransmission can be canceled during communication.	CXP1nSND CXP1nFLG2

For the specifications for the related registers, see **Section 30.4.1.1, CXP1nMODE —CXPI Operating Mode Select Register**, **Section 30.4.1.10, CXP1nREP — CXPI Retransmission Setting Register**, **Section 30.4.1.12, CXP1nSND — CXPI Transmission Control Register**, and **Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**. For the retransmission processing setting procedure through software, see **Section 30.5.10.2, Initial Settings**.

The hardware operation overview is provided below for four different retransmission operations.

- Retransmission after a loss in arbitration (retransmission not judged)
- Retransmission after a loss in arbitration (retransmission judged)
- Retransmission after an error occurred (retransmission not judged)
- Retransmission after an error occurred (retransmission judged)

NOTE

If a bit error occurred in PID/PTYPE transmission in PWM slave mode, the CXPI transmits a PID/PTYPE only once after retransmission was canceled.

(1) Retransmission after a Loss in Arbitration (Retransmission Not Judged)

The figure below shows the operation sequence of retransmission after a loss in arbitration (retransmission not judged) during frame transmission (No. 1 in **Table 30.78**).

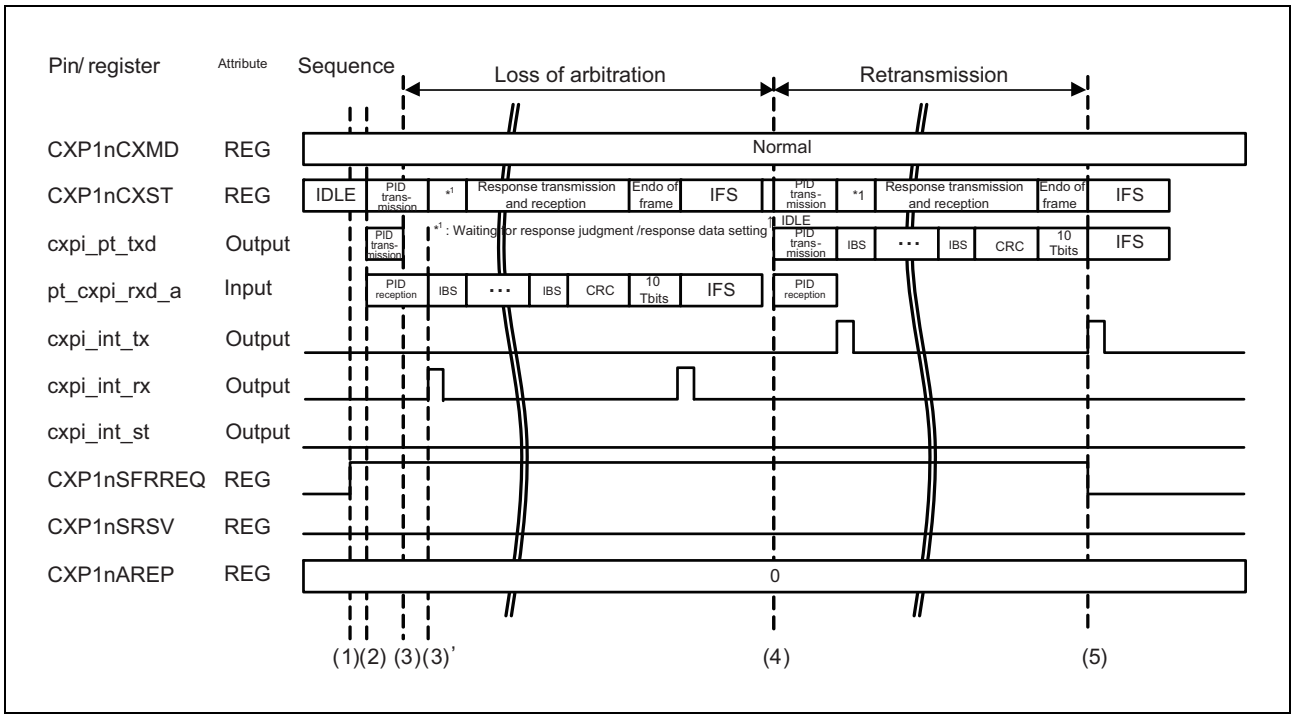


Figure 30.32 Retransmission Operation Sequence (Retransmission after a Loss in Arbitration (Retransmission Not Judged))

Operation

Conditions: The IP module loses in arbitration while transmitting a frame with the setting of "retransmission not judged". Frame transmission succeeds in the first round of retransmission.
($CXP1nMODE.CXP1nREPOP = 0_B$)

(1) 1 is written to CXP1nSND.CXP1nSFRREQ to request frame transmission.

(2) Frame transmission is started. CXP1nSND.CXP1nSFRREQ is not cleared at the start of transmission with the setting of "retransmission not judged".

(3) Arbitration was lost.
(Only in CXPI-PWM mode, the cxpi_pt_txd output is stopped because of a loss in arbitration during PID transmission as shown in the figure. In CXPI-NRZ mode, arbitration is made on a byte basis, so cxpi_pt_txd is output up to (3)'.)
A PID is received.
After that, it is equivalent to the processing flow of PID reception. Confirm the necessity of re-transmission cancellation.

(4) CXP1nCXST indicates an idle state after communication has been completed in response to a loss in arbitration and IFS has passed.
If 1 is written to CXP1nSND.CXP1nSFRREQ when CXP1nCXST indicates an idle state, retransmission is started immediately.
(Retransmission is not performed immediately in a slave node in polling mode, but performed after the IBS time has passed after PTYPE reception.)
After that, it is equivalent to the processing flow of frame transmission.

(5) When frame transmission initiated by retransmission has been normally completed, CXP1nSND.CXP1nSFRREQ is cleared to 0. The IFS time has passed, CXP1nCXST indicates an idle state. (No more retransmission is performed because frame transmission was successful.)

CXP1nSRSV and CXP1nAREP do not operate because retransmission is not judged.

- Retransmission cancellation
  When 0 is written to TCXP1nSND.CXP1nSFRREQ when CXP1nSND.CXP1nSFRREQ = 1, the next and following retransmissions are canceled.
- PID retransmission (No. 8 in **Table 30.78, Targets and Factors of Retransmission Processing**)
  The retransmission operation sequence of PID transmission is the same as the operation sequence for frame transmission except:
  - Read CXP1nSFRREQ as CXP1nSIDREQ.
  - Read CXP1nSRSV as CXP1nRRSV.
  - Read transmission as reception in the response operation during retransmission described for cxpi_pt_txd and pt_cxpi_rxd_a.

(2) Retransmission after a Loss in Arbitration (Retransmission Judged)

The figure below shows the operation sequence of retransmission after a loss in arbitration (retransmission judged) during frame transmission (No. 1 in **Table 30.78**).

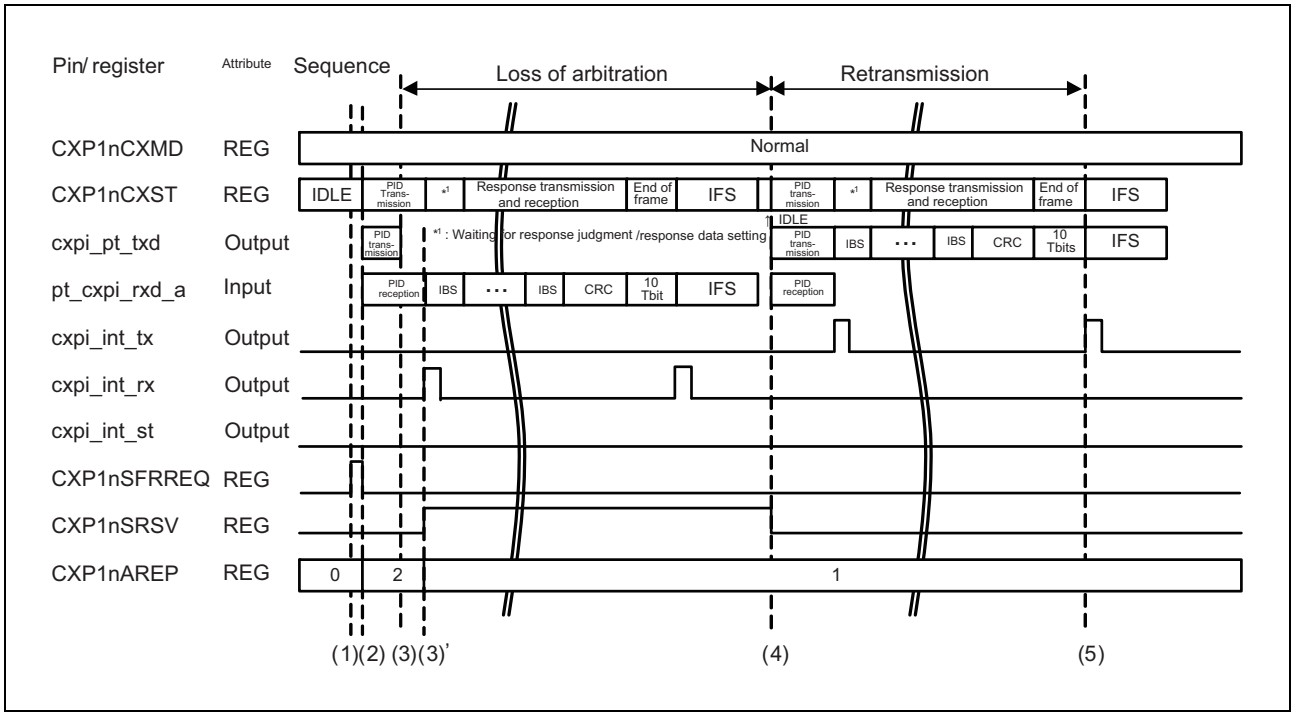


Figure 30.33 Retransmission Operation Sequence (Retransmission after a Loss in Arbitration (Retransmission Judged))

Operation

Conditions: The IP module loses in arbitration while transmitting a frame with the settings of "retransmission judged" and "the number of retransmissions is 2". Frame transmission succeeds in the first round of retransmission.
(CXP1nMODE.CXP1nREPOP = 1_B, CXP1nREP.CXP1nAIREP = 10_B or CXP1nREP.CXP1nARREP = 10_B)

(1) 1 is written to CXP1nSND.CXP1nSFRREQ to request frame transmission.

(2) Frame transmission is started. When "retransmission judged" is set, CXP1nSND.CXP1nSFRREQ is cleared at the start of transmission. The value set in the CXP1nREP.CXP1nAIREP bit or the CXP1nREP.CXP1nARREP bit is transferred to CXP1nFLG2.CXP1nAREP. (The transfer destination is determined by the value set in CXP1nSND.CXP1nREPSEL at the time of frame transmission request.)

(3) Arbitration was lost. If CXP1nFLG2.CXP1nAREP indicates 1 or more, CXP1nFLG2.CXP1nSRSV is set to make reservation for retransmission. Also, if CXP1nFLG2.CXP1nAREP is 1 or 2, the number of retransmissions is decremented.
(If CXP1nFLG2.CXP1nAREP indicates 3, the number of retransmissions is not decremented but retransmissions are carried out unlimitedly.)
(When CXP1nFLG2.CXP1nAREP is 0, CXP1nFLG2.CXP1nSRSV is not set because retransmission is not performed.)
(Only in CXPI-PWM mode, the cxpi_pt_txd output is stopped because of a loss in arbitration during PID transmission as shown in the figure. In CXPI-NRZ mode, arbitration is made on a byte basis, so cxpi_pt_txd is output up to (3)'. Also, CXP1nFLG2.CXP1nAREP and CXP1nFLG2.CXP1nSRSV change at (3)'.)
A PID is received.

After that, it is equivalent to the processing flow of PID reception. Confirm the necessity of re-transmission cancellation.

(4) CXP1nCXST indicates an idle state after communication has been completed in response to a loss in arbitration and IFS has passed.
If 1 is written to CXP1nFLG2.CXP1nSRSV when CXP1nCXST indicates an idle state, retransmission is started immediately and, at the same time, CXP1nFLG2.CXP1nSRSV is cleared to 0.
(Retransmission is not performed immediately in a slave node in polling mode, but performed after the IBS time has passed after PTYPE reception.)
After that, it is equivalent to the processing flow of frame transmission.

(5) When frame transmission initiated by retransmission has been normally completed and the IFS time has passed, CXP1nCXST indicates an idle state. No retransmission is initiated because CXP1nFLG2.CXP1nSRSV = 0.

- Retransmission cancellation
  - When 0 is written to CXP1nFLG2.CXP1nSRSV when CXP1nFLG2.CXP1nSRSV = 1, the next and following retransmissions are canceled.
  - When 0 is written to CXP1nFLG2.CXP1nEREP when CXP1nFLG2.CXP1nSRSV = 0, the next and following retransmissions are canceled.
  - When 0 is written to CXP1nFLG2.CXP1nEREP when CXP1nFLG2.CXP1nSRSV = 1, the next retransmission is initiated but the following retransmissions are canceled.
- PID retransmission (No. 8 in **Table 30.78**)
  The retransmission operation sequence of PID transmission is the same as the operation sequence for frame transmission except:
  - Read CXP1nSFRREQ as CXP1nSIDREQ.
  - Read CXP1nSRSV as CXP1nRRSV.
  - Read transmission as reception in the response operation during retransmission described for cxpi_pt_txd and pt_cxpi_rxd_a.

(3) Retransmission after an Error Occurred (Retransmission Not Judged)

The figure below shows the operation sequence of retransmission after an error occurred (retransmission not judged) during frame transmission (No. 5 in **Table 30.78**).

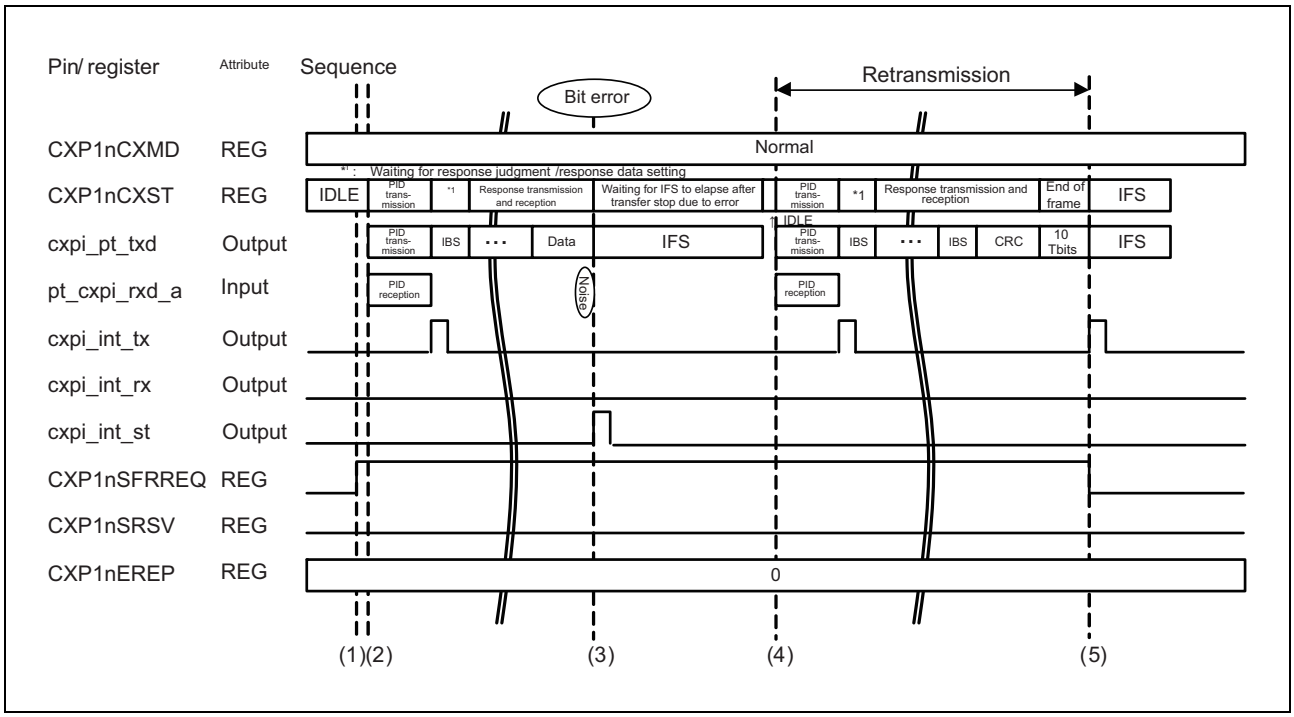


Figure 30.34 Retransmission Operation Sequence (Retransmission after an Error Occurred (Retransmission Not Judged))

Operation

Conditions: The IP has a bit error while transmitting a frame with the setting of “retransmission not judged”. Frame transmission succeeds in the first round of retransmission. (CXP1nMODE.CXP1nREPOP = 0_B)

(1) 1 is written to CXP1nSND.CXP1nSFRREQ to request frame transmission.

(2) Frame transmission is started. CXP1nSND.CXP1nSFRREQ is not cleared at the start of transmission with the setting of “retransmission not judged”.

(3) A bit error occurred.
An error detection interrupt is generated. CXP1nINT.CXP1nERRINT and CXP1nERR.CXP1nBITERR are set to 1.
Error status is set in CXP1nFLG1 and CXP1nERR register.
To clear them, the software write 0 to CXP1nINT.CXP1nERRINT and CXP1nERR.CXP1nBITERR.

(4) CXP1nCXST indicates an idle state after communication is stopped due to a bit error and IFS has passed.
If CXP1nSND.CXP1nSFRREQ = 1 when CXP1nCXST indicates an idle state, retransmission is started immediately.
(Retransmission is not performed immediately in a slave node in polling mode, but performed after the IBS time has passed after PTYPE reception.)

(5) When frame transmission initiated by retransmission is normally completed and the IFS time has passed, CXP1nCXST indicates an idle state and, at the same time, CXP1nSND.CXP1nSFRREQ

is cleared to 0. (No more retransmission is performed because frame transmission was successful.)

CXP1nSRSV and CXP1nEREP do not operate because retransmission is not judged.

- Retransmission cancellation
  When 0 is written to CXP1nSND.CXP1nSFRREQ when it is 1, the next and following retransmissions are canceled.
- Retransmission operation when another error occurred in frame transmission (Nos. 3 to 7 in **Table 30.78**)
  Even if a different error occurred, the IP retransmits the frame when CXP1nCXST indicates an idle state the IFS time after the error detection, as with No. 5. (For a bit error in PID, the IP may become idle state before the IFS time has passed.) The operation timing of CXP1nSND.CXP1nSFRREQ is the same as that for No. 5.
- PID retransmission (Nos. 9 to 15 in **Table 30.78**)
  The retransmission operation sequence of PID transmission is the same as the operation sequence for frame transmission except:
  - Read CXP1nSFRREQ as CXP1nSIDREQ.
  - Read CXP1nSRSV as CXP1nRRSV.
  - Read transmission as reception in the response operation during retransmission described for cxpi_pt_txd and pt_cxpi_rxd_a.

  If an error occurred during response reception, CXP1nCXST indicates an idle state and the IP performs retransmission the IFS time after having received the remaining data after the error.

(4) Retransmission after an Error Occurred (Retransmission Judged)

The figure below shows the operation sequence of retransmission after an error occurred (retransmission judged) during frame transmission (No. 5 in **Table 30.78**).

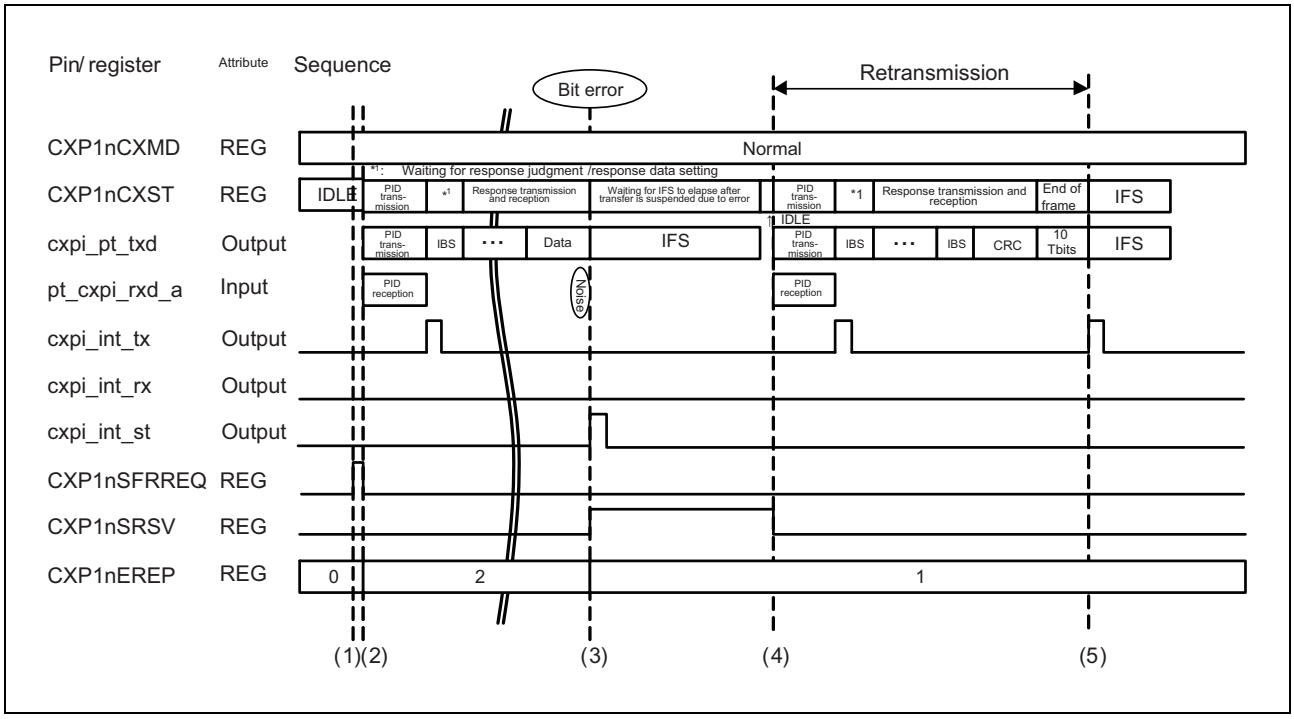


Figure 30.35 Retransmission Operation Sequence (Retransmission after an Error Occurred (Retransmission Judged))

Operation

Conditions: The IP has a bit error while transmitting a frame with the settings of “retransmission judged” and “the number of retransmissions is 2”. Frame transmission succeeds in the first round of retransmission.
(CXP1nMODE.CXP1nREPOP = 1_B, CXP1nREP.CXP1nAIREP = 10_B or CXP1nREP.CXP1nARREP = 10_B)

(1) 1 is written to CXP1nSND.CXP1nSFRREQ to request frame transmission.

(2) Frame transmission is started. When “retransmission judged” is set, CXP1nSND.CXP1nSFRREQ is cleared at the start of transmission. The value set in the CXP1nREP.CXP1nEIREP bit or the CXP1nREP.CXP1nERREP bit is transferred to CXP1nFLG2.CXP1nEREP. (The transfer destination is determined by the value set in CXP1nSND.CXP1nREPSEL at the time of frame transmission request.)

(3) A bit error occurred. If CXP1nFLG2.CXP1nEREP indicates 1 or more, CXP1nFLG2.CXP1nSRSV is set to make reservation for retransmission. Also, if CXP1nFLG2.CXP1nEREP is 1 or 2, the number of retransmissions is decremented.
(If CXP1nFLG2.CXP1nEREP indicates 3, the number of retransmissions is not decremented but retransmissions are carried out unlimitedly.)
(When CXP1nFLG2.CXP1nEREP is 0, CXP1nFLG2.CXP1nSRSV is not set because retransmission is not performed.)
An error detection interrupt is generated. CXP1nINT.CXP1nERRINT and CXP1nERR.CXP1nBITERR are set to 1.
Error status is set in CXP1nFLG1 and CXP1nERR register.
To clear them, the software write 0 to CXP1nINT.CXP1nERRINT and CXP1nERR.CXP1nBITERR.

(4) CXP1nCXST indicates an idle state after communication is stopped due to a bit error and IFS has passed.
If CXP1nFLG2.CXP1nSRSV = 1 when CXP1nCXST indicates an idle state, retransmission is started immediately and, at the same time, CXP1nFLG2.CXP1nSRSV is cleared to 0.
(Retransmission is not performed immediately in a slave node in polling mode, but performed after the IBS time has passed after PTYPE reception.)

(5) When frame transmission initiated by retransmission is normally completed and the IFS time has passed, CXP1nCXST indicates an idle state. No retransmission is initiated because CXP1nFLG2.CXP1nSRSV = 0.

- Retransmission cancellation
  - When 0 is written to CXP1nFLG2.CXP1nSRSV when CXP1nFLG2.CXP1nSRSV = 1, the next and following retransmissions are canceled.
  - When 0 is written to CXP1nFLG2.CXP1nEREP when CXP1nFLG2.CXP1nSRSV = 0, the next and following retransmissions are canceled.
  - When 0 is written to CXP1nFLG2.CXP1nEREP when CXP1nFLG2.CXP1nSRSV = 1, the next retransmission is initiated but the following retransmissions are canceled.
- Retransmission operation when another error occurred in frame transmission (No. 3 to 7 in **Table 30.78, Targets and Factors of Retransmission Processing**)
  Even if a different error occurred, the IP retransmits the frame when CXP1nCXST indicates an idle state the IFS time after the error detection, as with No. 5. (For a bit error in PID, the IP may become idle before the IFS time has passed.) The operation timing of

CXP1nSND.CXP1nSFRREQ, CXP1nFLG2.CXP1nSRSV and CXP1nFLG2.CXP1nEREP is the same as for No. 5.

- PID retransmission (Nos. 9 to 15 in **Table 30.78**)
  The retransmission operation sequence of PID transmission is the same as the operation sequence for frame transmission except:
  - Read CXP1nSFRREQ as CXP1nSIDREQ.
  - Read CXP1nSRSV as CXP1nRRSV.
  - Read transmission as reception in the response operation during retransmission described for cxpi_pt_txd and pt_cxpi_rxd_a.

  If an error occurred during response reception, CXP1nCXST indicates an idle state and the IP performs retransmission after the IFS time has passed following the reception of the remaining data after the error.

30.5.7.4 Error Detection

This section describes the error detection mechanism, the operation of major bits in the CXP1nFLW1 register when detecting an error, and the flow from communication suspension to communication restart.

For software processing procedure for errors, see **Section 30.5.10.15, Error Handling**.

[The operation of an error detection]

(a) The following condition is satisfied CXP1nFLW1.CXP1nCXST = 001_B (MSK_IFS state) and start time the IFS when detected UART stop bit, and detect value 1 for the ten time in a row, and transfers to CXP1nFLW1.CXP1nCXST = 000_B (IFS state). When MSK_IFS state, transmit operation be stopped and receive operation cannot be performed. If detect start bit when MSK_IFS, restart waiting IFS from stop bit.

(b) It performs counting at the CXPI nominal bit (Tbit) time and transitions to CXP1nFLW1.CXP1nCXST = 010_B (IDLE state) after the time (specified by CXP1nFRMW.CXP1nIFS – 10Tbit) has passed. PTYPE reception and PID reception become possible in IFS state.

(c) Communicate (SW Transmission or HW Retransmission) become possible in IDLE state.

(1) Bit Error

Since the bit error detection mechanism differs depending on the mode (CXPI-PWM mode or CXPI-NRZ mode), it will be described separately.

CXPI-PWM mode

In CXPI-PWM mode, bit errors are detected and transmission is stopped on a bit basis.

During transmission, the value of the transmit bit and the value of the received bit are compared. A bit error is detected if they do not match: The CXP1nERR.CXP1nBITERR bit is set when the next bit is transmitted (at the falling edge of cxpi_pt_txd) and an error detection interrupt is generated. Any other interrupt that indicates a transmission completion is not generated.

The error detection node, the range of the error detection is start bit of PTYPE or PID field. The other range of bit error detection are shown in the following.

- Loss-of-arbitration: Frame ID or Frame TYPE.
- Parity error: Parity bit.
- Framing error: Stop bit.

The value of the transmit bit and the value of the received bit do not match in response field: The IP detect a framing error, not bit error

Table 30.80 and **Table 30.81** summarizes the flow from transmission stop to restart which differs depending on the bit error location.

[Detected bit error by start bit of PID or PTYPE]

The following condition is satisfied CXP1nFLW1.CXP1nCXST = 000_B (IFS) when detected bit error. Communication is possible immediately without waiting IFS (The following condition is satisfied CXP1nFLW1.CXP1nCXST = 010_B (IDLE)).

[Detected bit error by other than those above]

The following condition is satisfied CXP1nFLW1.CXP1nCXST = 001_B (MSK_IFS).

For more information, see above explanation at [The operation of an error detection]

CXPI-NRZ mode

In CXPI-NRZ mode, the CXPI transceiver handles bit-basis bit error detection and transmission stop, and the IP performs byte-basis detection. In CXPI-NRZ mode, transmission stop is delayed because loopback to the IP is not made within 1 Tbit, so the transceiver handles bit-basis detection.

During transmit PTYPE or PID, a bit error is detected if the IBS time has passed and looped-back all bit is high level (recessive).

During transmit response, the looped-back stop bit is sampled and transmit data and received data are compared on a byte basis. A bit error is detected if they do not match.

A bit error is detected if the IBS time has passed and looped-back all bit is high level (recessive).

If bit error is detected, the CXP1nERR.CXP1nBITERR bit is set and an error detection interrupt is generated. Any other interrupt that indicates a transmission completion is not generated.

Table 30.80 and **Table 30.81** summarizes the flow from transmission stop to restart which differs depending on the bit error location.

The following condition is satisfied CXP1nFLW1.CXP1nCXST = 001_B (MSK_IFS). For more information, see above explanation at [The operation of an error detection].

Table 30.80 Flow from Transmission Stop to be available for transmission After Detecting a Bit Error

No.	Transfer Mode	Access Method	Bit Error Detection Bit		When Transmit is Enabled after Communication Stop (Transmission by software instruction, and retransmission)
1	CXPI-PWM mode	Event trigger method	PID	Start bit	Error → transmission stop → Transmit enabled from the next bit (first bit of data)
2			Other than above		Error → transmission stop → after IFS passed from the errored bit
3		Polling method	PTYPE	Start bit	Error → transmission stop → Transmit enabled from the next bit (first bit of data)
4			PID	Start bit	Error → transmission stop → after IFS passed (max)
5			PID after PTYPE	Start bit	Prohibited.
6			Other than above		Error → transmission stop → after IFS passed from the errored bit
7	CXPI-NRZ mode	All	All		Error at the stop bit → communication stop→ after IFS passed

Table 30.81 Flow from Transmission Stop to be available for receive After Detecting a Bit Error

No.	Transfer Mode	Access Method	Bit Error Detection Bit		When Receive is Enabled after Communication Stop (PID reception and PTYPE reception by hardware are enabled)
1	CXPI-PWM mode	Event trigger method	PID	Start bit	Error → transmission stop → Receive enabled from the next bit (first bit of data)
2			Other than above		Error → transmission stop → after 10 Tbit passed from the errored bit *1
3		Polling method	PTYPE	Start bit	Error → transmission stop → Receive enabled from the next bit (first bit of data)
4			PID	Start bit	Error → transmission stop → after 10 Tbit passed (max)
5			PID after PTYPE	Start bit	Prohibited.
6			Other than above		Error → transmission stop → after 10 Tbit passed from the errored bit *1
7	CXPI-NRZ mode	All	All		Error → transmission stop → after 10 Tbit passed *1

Note 1. When a bit error is detected, all received data are discarded before 10 Tbit passed (not stored in registers CXP1nRPID, CXP1nRFRI, CXP1nRDATAm (m = 01-08), CXP1nRCRC).

(2) CRC Error

The receiving node compares the result of CRC calculation done by the IP with the received CRC value. It detects a CRC error if they do not match. Even if an error occurred, the received CRC value is stored in the CXP1nRCRC register. 10 Tbits after the CRC stop bit reception is completed, the CXP1nERR.CXP1nCRCERR bit is set and an error detection interrupt is generated. No frame reception completion interrupt is generated.

When an interrupt is generated (CRC error detection), the CXP1nFLW1.CXP1nCXST bit is set to 000_B(IFS). For more information, see above explanation at [The operation of an error detection].

(3) Parity Error

The IP in a receiving node checks the number of bits with logical value “1” in the received PID and PTYPE fields, detects a parity error if it is an even number, sets the CXP1nERR.CXP1nPTYERR bit during stop bit reception, and generates an error detection interrupt. No PTYPE reception completion or PID reception completion interrupt is generated. If the PTYPE field results in an error, the succeeding PID transmission is impossible. If the PID field results in an error, it is stored in the CXP1nRPID register, but no response is transmitted afterward, and all received responses are discarded (not stored in registers).

When the IP is a transmitting node, it detects a parity error on the PID of the looped-back received data. It stops transmission if detecting a parity error. The other operations are the same as those already described for the receiving node.

When a parity error is detected, the CXP1nFLW1.CXP1nCXST bit is set to 001_B(MSK_IFS). For more information, see above explanation at [The operation of an error detection].

(4) Physical Bus Error (Not Supported)

The hardware does not detect physical bus errors. The software shall monitor the communication status by checking the IP interrupt status and performing polling to registers to detect physical bus errors.

(5) Data Length Error

The following describes the data length error generation mechanisms in the transmitting node and the receiving node.

Transmitting node

The IP detects a data length error based on the following two factors:

- Factor 1 (the received data is longer than the DLC)
  The IP detects a data length error if it receives a start bit within 10 Tbits after transmitting the CRC value in the response field. It sets the CXP1nERR.CXP1nDLNERR bit when it samples the start bit to generate an error detection interrupt. No frame transmit completion interrupt is generated. The CXP1nFLW1.CXP1nCXST bit is set to 001_B (MSK_IFS) when detecting a data length error. For more information, see above explanation at [The operation of an error detection].
- Factor 2 (an underrun occurred during long frame transmission)
  If the CXP1nINT.CXP1nSFQINT bit was not cleared when another 7-byte data have been transmitted after transmitting 7-byte data and the CXP1nINT.CXP1nSFQINT was set during long frame transmission, the IP detects a data length error, judging transmit data was not set in time. During stop bit transmission of the 8th-byte data, the CXP1nERR.CXP1nDLNERR bit is set and an error detection interrupt is generated. When a data length error is detected, the CXP1nFLW1.CXP1nCXST bit is set to 001_B (MSK_IFS). For more information, see above explanation at [The operation of an error detection].

- Factor 3 (the received loopback data is shorter than the DLC)
  CXPI-NRZ mode: The IP detects a data length error if the received loopback data is shorter than the DLC.
  CXPI-PWM mode: Substituted by detection of bit error.
  The IP detects a data length error if it does not receive a start bit in the response field in the IBS time (10 Tbits) of the PID field (Local node isn't a target.) or it does not receive a start bit in the IBS time (10 Tbits) of the response field. It sets the CXP1nERR.CXP1nDLNERR bit in the ending of the IBS to generate an error detection interrupt. If the CXP1nFLW1.CXP1nCXST bit is set to 000_B (IFS) when detecting a data length error. For more information, see above explanation at [The operation of an error detection].

Receiving node

The IP detects a data length error based on the following two factors:

- Factor 1 (the received data is longer than the DLC)
  The IP detects a data length error if it receives a start bit within 10 Tbits after receiving the CRC value in the response field. It sets the CXP1nERR.CXP1nDLNERR bit when it samples the start bit to generate an error detection interrupt. If the CXP1nFLW1.CXP1nCXST bit is set to 001_B (MSK_IFS) when detecting a data length error. For more information, see above explanation at [The operation of an error detection].
- Factor 2 (the received data is shorter than the DLC, or the response field is not received)
  After transmission of PID, the IP detects a data length error if it does not receive a start bit in the response field in the IBS time (10 Tbits) of the PID field (the response field is no received) or it does not receive a start bit in the IBS time (10 Tbits) of the response field. After reception of PID, the IP detects a data length error if it does not receive a start bit in the IBS time (10 Tbits) of the response field (Local node isn't a target.). It sets the CXP1nERR.CXP1nDLNERR bit in the ending of the IBS to generate an error detection interrupt. If the CXP1nFLW1.CXP1nCXST bit is set to 000_B (IFS) when detecting a data length error. For more information, see above explanation at [The operation of an error detection].

(6) Counter Error (Not Supported)

This IP does not detect counter errors. The software shall monitor the status of the CXP1nRFRI.CXP1nRCT bit to detect counter errors.

(7) Overrun Error

If the CXP1nINT.CXP1nRFQINT bit was not cleared when receiving another 8-byte data was received after receiving 8-byte data and the CXP1nINT.CXP1nRFQINT was set during long frame reception, the IP detects an overrun error, judging that received data was not saved in time. When the 8th-byte received data is sampled, the CXP1nERR.CXP1nORNERR bit is set and an error detection interrupt is generated. The data that has been received before the CXP1nERR.CXP1nORNERR bit is set is stored in CXP1nRDATAm (m = 01 to 08) and the subsequently received data is not stored in registers. If the CXP1nFLW1.CXP1nCXST bit is set to 001_B when detecting an overrun error. For more information, see above explanation at [The operation of an error detection].

(8) Framing Error

The following describes the framing error generation mechanisms in the transmitting node and the receiving node.

Transmitting node

The IP detects a framing error if the logical value of the stop bit which is looped back and received during transmission is 0. It sets the CXP1nERR.CXP1nFRMERR bit when it samples the start bit to generate an error detection interrupt. Any other interrupt that indicates a transmission completion is not generated. When a framing error is detected, the CXP1nFLW1.CXP1nCXST bit is set to 001_B. For more information, see above explanation at [The operation of an error detection].

Receiving node

The IP detects a framing error if the logical value of the received stop bit is 0. It sets the CXP1nERR.CXP1nFRMERR bit when it samples the start bit to generate an error detection interrupt. Any other interrupt that indicates a reception completion is not generated. If the PTYPE field results in an error, the succeeding PID transmission is impossible. If the PID field results in an error, it is stored in the CXP1nRPID register, but no response is transmitted afterward. Also, all the received responses are discarded (not stored in registers). If the response results in an error, the received data with an error and the subsequently received data are discarded (not stored in registers). When detecting a framing error, the IP sets the CXP1nFLW1.CXP1nCXST bit to 001_B. For more information, see above explanation at [The operation of an error detection].

(9) State Error (Not Defined in CXPI Specifications)

A state error occurs due to the following factors:

- Factor 1
  A PTYPE field is received in event trigger mode.
- Factor 2
  A PTYPE field is received by the master node.
- Factor 3
  When the IP in a slave node starts PID transmission because it receives a PTYPE field in polling mode, it receives the PTYPE field again and loses an arbitration.

For each factor, the CXP1nFLG1.CXP1nSTERR bit is set and an error detection interrupt is generated when the stop bit of PTYPE is sampled. No PTYPE reception completion interrupt is generated. After state error is detected, write 0_B to CXP1nMODE.CXP1nEN initializes the internal circuits.

(10) Fatal Error (Not Defined in CXPI Specifications)

This error only occurs due to an IP failure or a bit error of an IP internal signal.

Perform a reset when detecting a fatal error. If the error persists, the IP may be malfunctioning.

As soon as the condition is satisfied, the CXP1nFLG1.CXP1nFTERR bit is set and an error detection interrupt is generated.

The possible factors for fatal errors are as follows:

After fatal error is detected, write 0_B to CXP1nMODE.CXP1nEN initializes all.

- Factor 1
  CXP1nFLW2.CXP1nFRMPHS becomes 11_B (undefined) during response transmission or reception.

- Factor 2
  The frame counter in the IP (frm_cnt_reg: incremented as each byte is transmitted or received) indicates 4 or more while transmitting or receiving a response and CXP1nFLW2.CXP1nFRMPHS = 00_B (frame information/extension DLC) or 10_B (CRC).

30.5.7.5 Bus Monitoring

It is possible to receive all PIDs on the communication bus. This generates a PID reception completion interrupt each time a PID is received. This also enables a response to be transmitted or received after receiving each PID. The received response is stored in CXP1nRFRI, CXP1nRDATAm (m = 01 to 08) and CXP1nRCRC.

Setting the CXP1nRPIDF2m.CXP1nFIDMSK bit (m = 01 to 04) to 1111111_B enables bus monitoring setting. For the register specifications, see **Section 30.4.1.23, CXP1nRPIDF2m —CXPI Received Frame ID Setting Register 2m (m = 01 to 04)**.

30.5.7.6 Failure Management Function

For the failure management function, the IP has the CXP1nERR.CXP1nERRCNT bit (error counter) and the CXP1nFLG1.CXP1nFAIL bit (which indicates the transmission disabled state). For the register specifications, see **Section 30.4.1.15, CXP1nFLG1 — CXPI Flag Register 1** and **Section 30.4.1.19, CXP1nERR — CXPI Error Register**.

If detecting a bit error, a framing error or a data length error during transmission, the IP increments the error counter by 8 when a relevant bit of the CXP1nERR register is set. When the IP transmits the PTYPE field, the PID field or a response normally, it decrements the error counter by 1 when a relevant bit of CXP1nINT register is set (when a transmission completion interrupt source is generated). If an error occurs when the error counter value is 248 or more, the error counter is set to 255 and the CXP1nFLG1.CXP1nFAIL bit is set to disable transmission.

The table below lists the operations which are disabled in the transmission disabled state. Clock transmission is not stopped.

Table 30.82 Transmission Operations Disabled When CXP1nFLG1.CXP1nFAIL = 1

No.	Transmission Operations Disabled When CXP1nFLG1.CXP1nFAIL = 1
1	Frame transmission
2	PID transmission
3	Response transmission
4	Automatic response
5	PTYPE transmission
6	Sleep frame transmission
7	Retransmission of frame transmission
8	Retransmission of PID transmission

The figure below shows the release from failure (The transmission disabled) state.

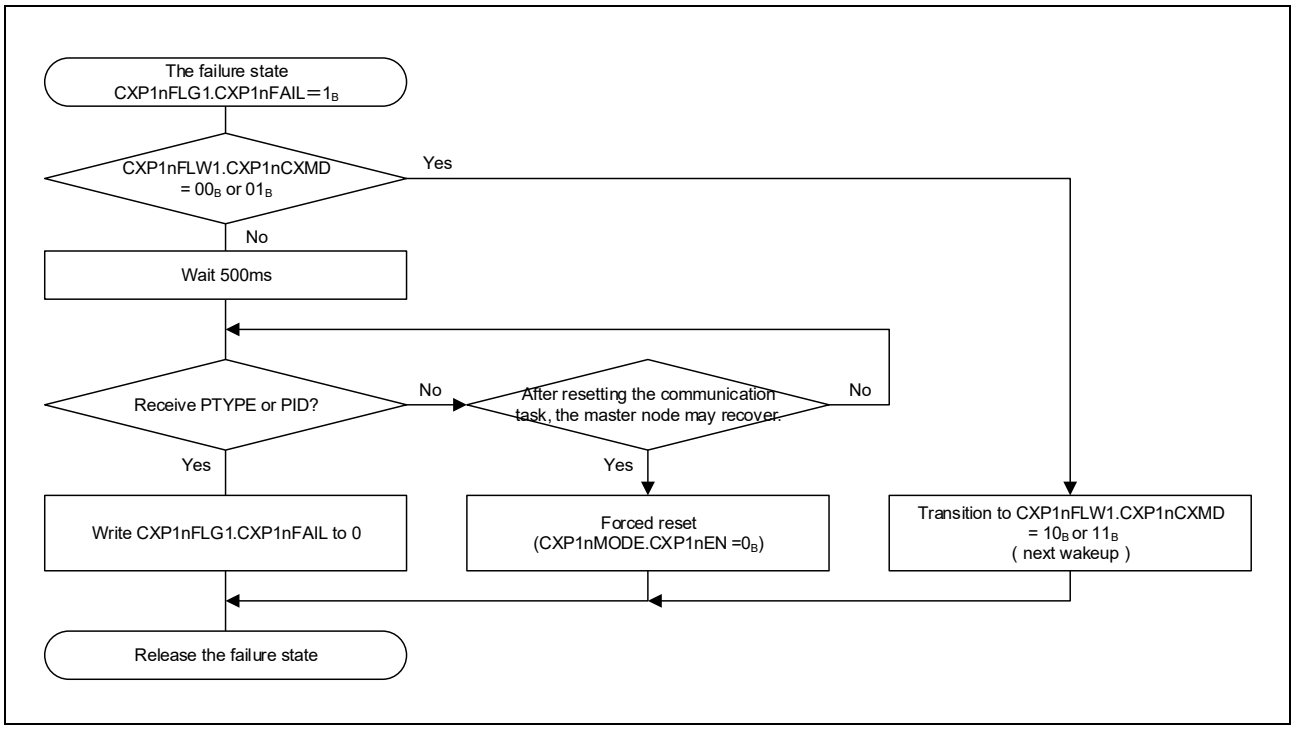


Figure 30.36 Release the failure state

30.5.7.7 Error Bit Function (Not Supported)

This IP does not support the error bit function.

The software shall check whether or not an error occurred when transmitting or receiving a frame.

30.5.8 Extension Functions

This section describes the extended functions which are not defined in the Protocol Specification but installed solely in the IP.

30.5.8.1 Frame ID Filtering Function

The frame ID filtering function sets an ID mask value for the target PID of reception or manual response.

No PID comparison is made for the ID-masked bit. For details on the frame ID filtering function, see **Section 30.4.1.23, CXP1nRPIDF2m —CXPI Received Frame ID Setting Register 2m (m = 01 to 04)**.

30.5.8.2 Noise Filter

The receive pin (pt_cxpi_rxd_a) of the CXPI is provided with a noise filter that can eliminate noise for up to 12.775 µs (PCLK = 40 MHz). For details on the noise filter, see **Section 30.4.1.6, CXP1nFIL — CXPI Input Filter Setting Register**.

The receive pin (pt_cxpi_rxd_a) has a chattering removal function of two logic coincidence on a PCLK standard.

30.5.8.3 Offset Sampling

CXPI has auto remediation function for offset sampling.

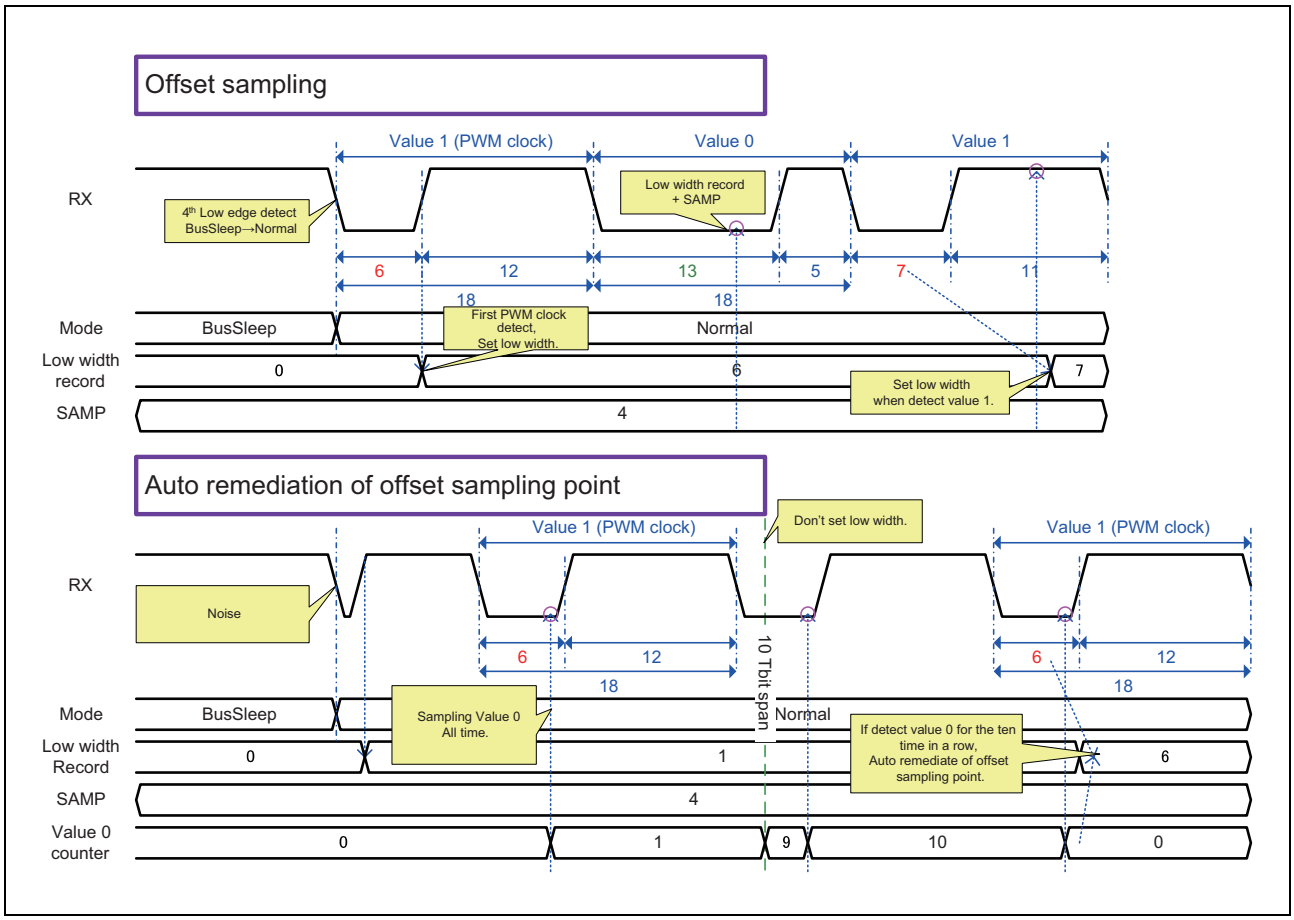


Figure 30.37 Offset sampling function

Note that if noise occurs during transmission operation in offset sampling and CXPI-PWM mode, normal receive operation cannot be performed for maximum 10Tbit + IFS period because, initialization of internal sequence and detection of PWM clock for 10Tbit is required after reset release.

Also, framing error might be detected at the time of initialization period completion.

1. Issue noise during transmission.
2. To initialize sampling counter, re-write threshold value (Low width record) of offset sampling. Here, threshold value becomes extremely small due to state clear timing and it is judged as logical value 0, irrespective of logical value 0/1 on transmission bus.
3. Receive logical value 0 for 10Tbit period (write-back threshold value to initial value)
4. Wait for IFS period from here
5. Transmission possible (error process is required if framing error is detected)

When the extremely large Low width (Serge noise filter + Low period of logic value 0 + Sampling timing setup (CXP1nSAMP.CXP1nSMP) more than or equal to 1Tbit period) has been recorded, since it becomes impossible to sample normally, be careful of a sampling timing setup (CXP1nSAMP.CXP1nSMP).

30.5.8.4 PWM Encoding/Decoding ON/OFF Function

The PWM encoding/decoding ON/OFF function enables or disables PWM encoding/decoding.

For details on the PWM encoding/decoding ON/OFF function, see **Section 30.5.1, CXPI System Configuration**.

30.5.8.5 Control Pins for NRZ Mode (Not Supported)

In CXPI-NRZ mode, a pin that controls the CXPI transceiver is required. Since the IP supports only communication pins, the control pins for CXPI-NRZ mode must be supported by a module other than this IP. For the details about the control pins for CXPI-NRZ mode, see **Section 30.5.1, CXPI System Configuration**.

30.5.8.6 Diagnostic Function

Since the diagnostic function is the same as the failure management function, see **Section 30.5.7.6, Failure Management Function**.

30.5.8.7 Software Reset

Two types of reset are available: a forced reset that initializes the entire IP, and a simple reset that resets some functions after detecting an error and returns the IP to the communication enabled state.

- Forced reset:
  Setting CXP1nMODE.CXP1nEN to 0 initializes the internal circuits related to the associated registers.
  For details on the associated registers, see **Table 30.83**.
  As long as CXP1nMODE.CXP1nEN = 0, the IP remains initialized.
- Simple reset:
  Writing 1 to CXP1nWUP.CXP1nSTCLR initializes the internal circuits related to the associated registers.
  For details on the associated registers, see **Table 30.83**.
  Initialization occurs only once when 1 is written to CXP1nWUP.CXP1nSTCLR.

When the software has detected a physical bus error and transition to bus sleep mode is prohibited[*1], a simple reset is issued to initialize some functions of the IP.

Note 1. The software manages the prohibition of transition to bus sleep mode.

The table below shows the ranges covered by the two types of reset.

Table 30.83 Ranges Affected by Software Resets

Target to Be Initialized	Forced Reset CXP1nMODE.CXP1nEN = 0_B	Simple Reset CXP1nWUP.CXP1nSTCLR = 1_B
CXP1nMODE	No	No
CXP1nBRT	Yes	No
CXP1nV1LW	Yes	No
CXP1nV0LW	Yes	No
CXP1nSAMP	Yes	No
CXP1nFIL	Yes	No
CXP1nWU	Yes	No
CXP1nFRMW	Yes	No
CXP1nSLP	Yes	No
CXP1nREP	Yes	No
CXP1nWUP	Yes	No
CXP1nSND	Yes	Yes
CXP1nIMSK	Yes	No
CXP1nINT	Yes	No
CXP1nFLG1	*1	No
CXP1nFLG2	Yes	Yes
CXP1nFLW1	Yes	Yes (Only CXP1nFLW1.CXP1nCXMD is not initialized)
CXP1nFLW2	Yes	Yes
CXP1nERR	*2	No
CXP1nRPID	Yes	No
CXP1nSPID	Yes	No
CXP1nRPIDF1m (m = 01 to 12)	Yes	No
CXP1nRPIDF2m (m = 01 to 04)	Yes	No
CXP1nARPID	Yes	No
CXP1nRFRI	Yes	No
CXP1nSFRI	Yes	No
CXP1nARFRI	Yes	No
CXP1nRCRC	Yes	No
CXP1nSCRC	Yes	No
CXP1nARCRC	Yes	No
CXP1nRDATAm (m = 01 to 08)	Yes	No
CXP1nSDATAm (m = 01 to 08)	Yes	No
CXP1nARDATAm (m = 01 to 06)	Yes	No

Note: Yes: Initialized
No: Not initialized

Note 1. The CXP1nFLG1.CXP1nFAIL bit is initialized only in the IP in the master node.

Note 2. The CXP1nERR.CXP1nERRCNT bit is initialized only in the IP in the master node.

30.5.8.8 Status

The CXPI has several statuses to indicate the operating mode and operation state.

This section describes the operating mode and the state transition of operation states.

(1) Operating Mode (CXP1nFLW1.CXP1nCXMD)

The CXPI controls the bus sleep/wakeup mode and communication enabled state by operating mode.

The figure below shows the state transition of operating mode. Each state (1) to (4), and transition condition are described below.

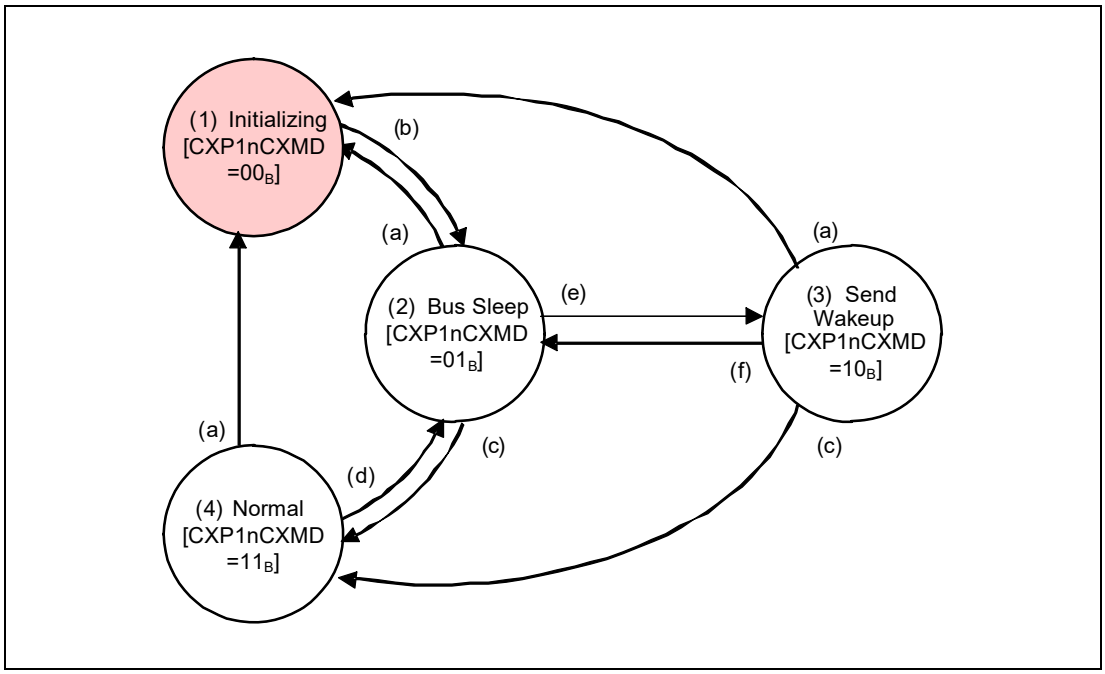


Figure 30.38 State Transition of Operating Mode

(1) Initializing mode

In initializing mode, the CXPI is not in an operational state. This is the state in which a reset is being released or is the first state after a reset has been released.

The following are transition conditions (the following transition conditions are listed in the order of priority).

(b) Enabling the CXPI operation (CXP1nMODE.CXP1nEN = 1) triggers the transition to bus sleep mode.

(2) Bus sleep mode

Bus sleep mode is power-saving mode in which data transmission and reception are stopped. This mode is used to initialize the IP.

The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) The CXPI transitions to initializing mode when a reset = 0 or when a forced reset is issued (CXP1nMODE.CXP1nEN = 0_B).

(c) This transition condition differs depending on the transfer mode. The following are transition conditions for each transfer mode.

CXPI-PWM mode: The CXPI transitions to normal mode when detecting the clock 4 or 5 times.

CXPI-NRZ mode: Writing 1_B to CXP1nWUP.CXP1nCLKON triggers a transition to normal mode.

(e) A slave node transitions to send wakeup mode at the start of wakeup pulse transmission (CXP1nWUP.CXP1nWUPREQ = 1_B).

(3) Send wakeup mode

Send wakeup mode indicates that the slave node is transmitting a wakeup pulse.
Note that send wakeup mode is part of standby mode specified in the Protocol Specification.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) The CXPI transitions to initializing mode when a reset = 0 or when a forced reset is issued (CXP1nMODE.CXP1nEN = 0_B).

(c) This transition condition differs depending on the transfer mode. The following are transition conditions for each transfer mode.

CXPI-PWM mode: The CXPI transitions to normal mode when detecting the clock.

CXPI-NRZ mode: Writing 1_B to CXP1nWUP.CXP1nCLKON triggers a transition to normal mode.

(f) Upon completion of wakeup pulse transmission, the CXPI transitions to bus sleep mode.

(4) Normal mode

Normal mode indicates that the CXPI can communicate.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) The CXPI transitions to initializing mode when a reset = 0 or when a forced reset is issued (CXP1nMODE.CXP1nEN = 0_B).

(d) This transition condition differs depending on the setting of each operation. The following are transition conditions depending on each operation setting.

Master node: Transitions to bus sleep mode when writing 1 to CXP1nWUP.CXP1nCLKOFF.

Slave node: Transitions to bus sleep mode the sleep transition time after receiving a sleep frame.

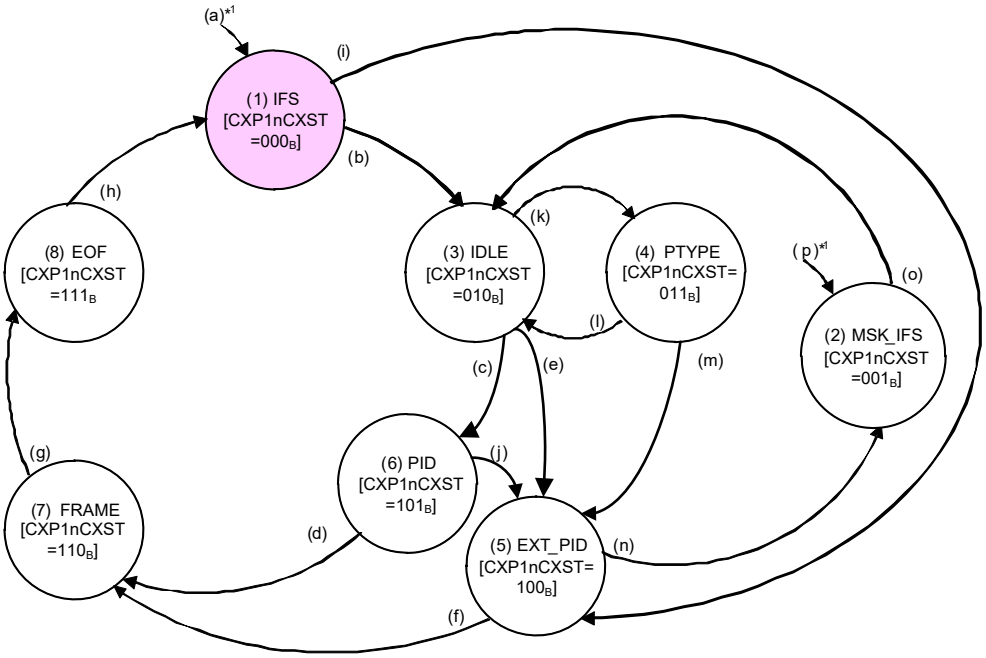
CXPI-PWM mode: The CXPI transitions to bus sleep mode when not detecting the clock signal.

CXPI-NRZ mode: The CXPI transitions to bus sleep mode when 1 is written to CXP1nWUP.CLKOFF.

(2) Operating State (CXP1nFLW1.CXP1nCXST)

The CXPI controls transmission and reception according to the operating state.

This section describes the operating states. Each state (1) to (8), and transition condition are described below.



Note 1. Transition occurs when conditions are satisfied in each state.

Figure 30.39 State Transition of Operating State

(1) Waiting for IFS to elapse (IFS in the figure)

The CXPI first transitions to this state after a reset is released and CXP1nCXMD = 11_B (normal mode).

This state waits for the IFS time to elapse. The CXPI transitions to the idle state after the IFS time has passed.

Note that it transitions to another state if detecting a PID or PTYPE before the IFS time has passed.

Then IP isn't possible to receive the just after IFS period (CXP1nIFS set value +1) when transferred to CXP1nCXMD = BusSleep in CXP1nCXMD = Normal.

The following are transition conditions (the following transition conditions are listed in the order of priority).

(p) The CXPI transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error*1 occurred.

Note 1. The following are included in the transfer error. (This applies to other states.)
Bit error (other than start bit of PID or PTYPE)
Parity error
Data length error (other than not received response)
Framing error
Overrun error
State error
Fatal error

(i) The CXPI transitions to the EXT_PID state when receiving a PID.

(b) It performs counting at the CXPI nominal bit (Tbit) time and transitions to the idle state after the time specified by CXP1nFRMW.CXP1nIFS has passed.
It transitions to the idle state when receiving a PTYPE field.

(2) Waiting for IFS to elapse after transfer is suspended due to an error (MSK_IFS in the figure)
This state waits for the IFS time to elapse when an error occurred or after a simple reset was issued (CXP1nSTCLR = 1). If detected CRC error, (o) cause immediate.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 is issued.

(o) Detect value 1 for the ten time in a row, change to IFS state.

(3) Transmission/reception stopped (IDLE) (IDLE in the figure)
In the transmission/reception stopped (IDLE) state, the CXPI can communicate.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 or bit error detected by start bit of PID or PTYPE is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(e) It transitions to the EXT_PID state when receiving a PID.

(k) It transitions to the PTYPE state when transmitting a PTYPE field (CXP1nSND.SPTREQ = 1).

(c) It transitions to the PID state when transmitting a PID (when transmission is accepted after writing 1 to CXP1nSND.SIDREQ).

(4) Transmitting PTYPE (PTYPE in the figure)
In the “transmitting PTYPE” state, the CXPI is transmitting a PTYPE field.
It transitions to the idle state after it has transmitted a PTYPE field.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(m) It transitions to the EXT_PID state when receiving a PID.

(l) It transitions to the idle state when detecting a PTYPE field in the received data looped back for PTYPE transmission.

(5) Waiting for response judgment/response data setting (EXT_PID in the figure)

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 or bit error detected by start bit of PID or PTYPE is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(n) It transitions to the MSK_IFS state when the following conditions are satisfied:

- The IBS time has passed.
- The received PID does not match the PID set in CXP1nRPIDF1m (m = 01 to 12), CXP1nRPIDF2m (m = 01 to 04) or CXP1nARPID.
- CXP1nFLG1.CXP1nFAIL = 1
- When 9 Tbits have passed after receiving a PID, if the following condition of (f) is not satisfied.

(f) It transitions to the FRAME state at the start of manual response (CXP1nSND.CXP1nSRESERQ = 1). Or, it transitions to the FRAME state at the start of response reception (when detecting a start bit).

(6) PID transmission (PID in the figure)
In the “PTYPE transmission” state, the CXPI is transmitting a PID.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(j) It transitions to the EXT_PID state when the received PID matches the PID set in CXP1nRPIDF1m (m = 01 to 12), CXP1nRPIDF2m (m = 01 to 04) or CXP1nARPID.

(d) It transitions to the FRAME state when it has transmitted the PID.

(7) Response (frame information/extension DLC/data/CRC) transmission/reception (FRAME in the figure)
In the response state, the CXPI is transmitting or receiving a response.
The following are transition conditions (the following transition conditions are listed in the order of priority).

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(g) It transitions to the EOF state when it has transmitted or received a response.

(8) Frame completion (waiting for IBS to elapse) (EOF in the figure)

(a) It transitions to the IFS when a forced reset (CXP1nMODE.CXP1nEN = 0_B) or a reset = 0 is issued.

(p) It transitions to the MSK_IFS state if a simple reset is issued (CXP1nWUP.CXP1nSTCLR = 1) or a transfer error occurred.

(h) It transitions to the IFS state 10 Tbits (fixed value determined by the hardware) after it has transmitted or received a response.

30.5.9 Interrupts

30.5.9.1 Transmission

(1) PTYPE Transmission Completion

When the CXPI has transmitted a PTYPE field, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.6, PTYPE Transmission**.

(2) PID Transmission Completion

When the CXPI has transmitted a PID field, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.1, PID Transmission** or **Section 30.5.4.2, Frame Transmission**.

(3) Transmission Data Setting Request

When the CXPI transmits a normal frame, or a long frame for manual response, it outputs an interrupt request at a level each time it has transmitted eight bytes. For detailed operation, see **Section 30.5.4.2, Frame Transmission**, or **Section 30.5.4.3, Manual Response**.

(4) Frame Transmission Completion

When the CXPI has transmitted a response, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.2, Frame Transmission**, **Section 30.5.4.3, Manual Response**, or **Section 30.5.4.4, Automatic Response**.

30.5.9.2 Reception

(1) PTYPE Reception Completion

When the CXPI has received a PTYPE field, it outputs an interrupt request at a level.

For details, see **Section 30.5.4.7, PTYPE Reception**.

(2) PID Reception Completion

When the CXPI has received a PID field, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.5, PID Reception**, or **Section 30.5.4.3, Manual Response**.

(3) Received Data Save Request

When the CXPI has received a long-frame response, it outputs an interrupt request at a level each time it receives eight bytes.

For detailed operation, see **Section 30.5.4.5, PID Reception**, or **Section 30.5.4.1, PID Transmission**.

(4) Frame Reception Completion

When the CXPI has received a response, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.5, PID Reception**, or **Section 30.5.4.1, PID Transmission**.

(5) PTYPE Valid Period Completion

When the CXPI has transmitted a PTYPE field or the valid period after detecting a PTYPE field is ended, it outputs an interrupt request at a level.

For detailed operation, see **Section 30.5.4.6(2), When There is No Response from the Slave Node after Transmitting PTYPE**, or **Section 30.5.4.7(2), When Not Transmitting PID after Receiving PTYPE**.

(6) Wakeup Pulse Detection (Not Supported)

This IP does not support the wakeup pulse detection interrupt. It receives a wakeup pulse by using the CXPI transceiver and a module other than this IP.

30.5.9.3 Status

(1) Error Detection

When the CXPI has detected an error, it outputs an interrupt request at a level.

For details, see **Section 30.5.7.4, Error Detection**.

(2) Sleep Transition Completion

The CXPI outputs an interrupt request at a level the sleep transition time after it has transmitted or received a sleep frame. For detailed operation, see **Section 30.5.4.8, Sleep Frame Transmission Operation**, or **Section 30.5.4.9, Sleep Frame Reception Operation**.

30.5.10 Software Processing Flow

This section describes the software processing flow of the CXPI.

30.5.10.1 Overall Correlation Diagram

Each operation is designed by combining software processes. This section describes how software processes are combined for each operation.

(1) At Startup

The figure below is an overall correlation diagram at startup.

(a) Startup by Own Node

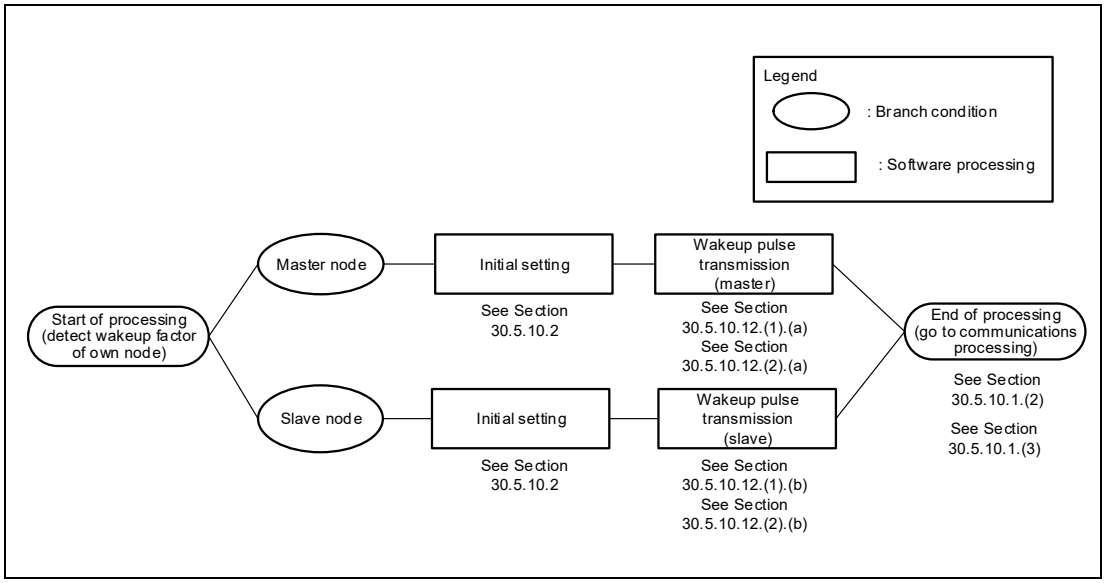


Figure 30.40 Overall Correlation Diagram at Startup by Own Node

(b) Startup by Other Node

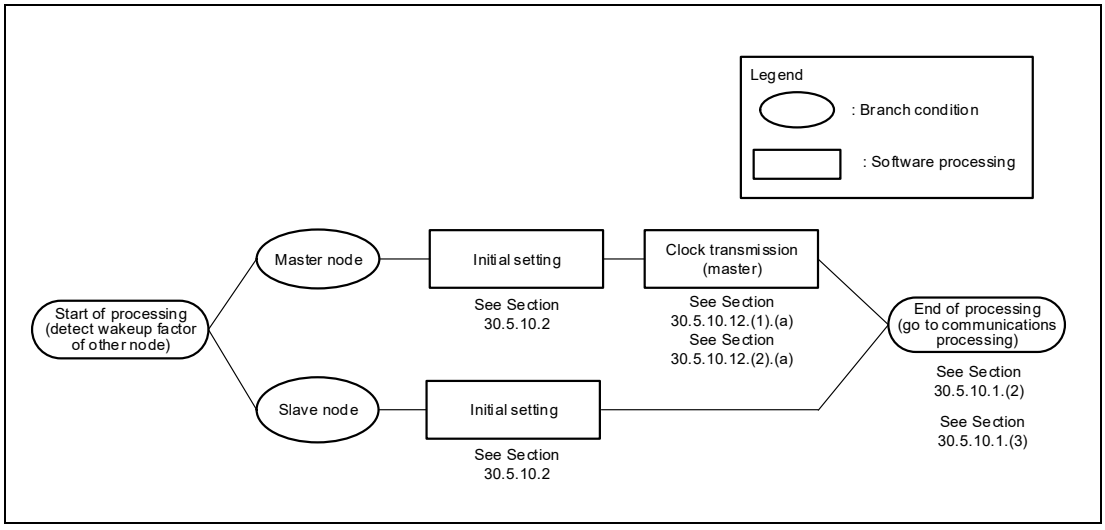
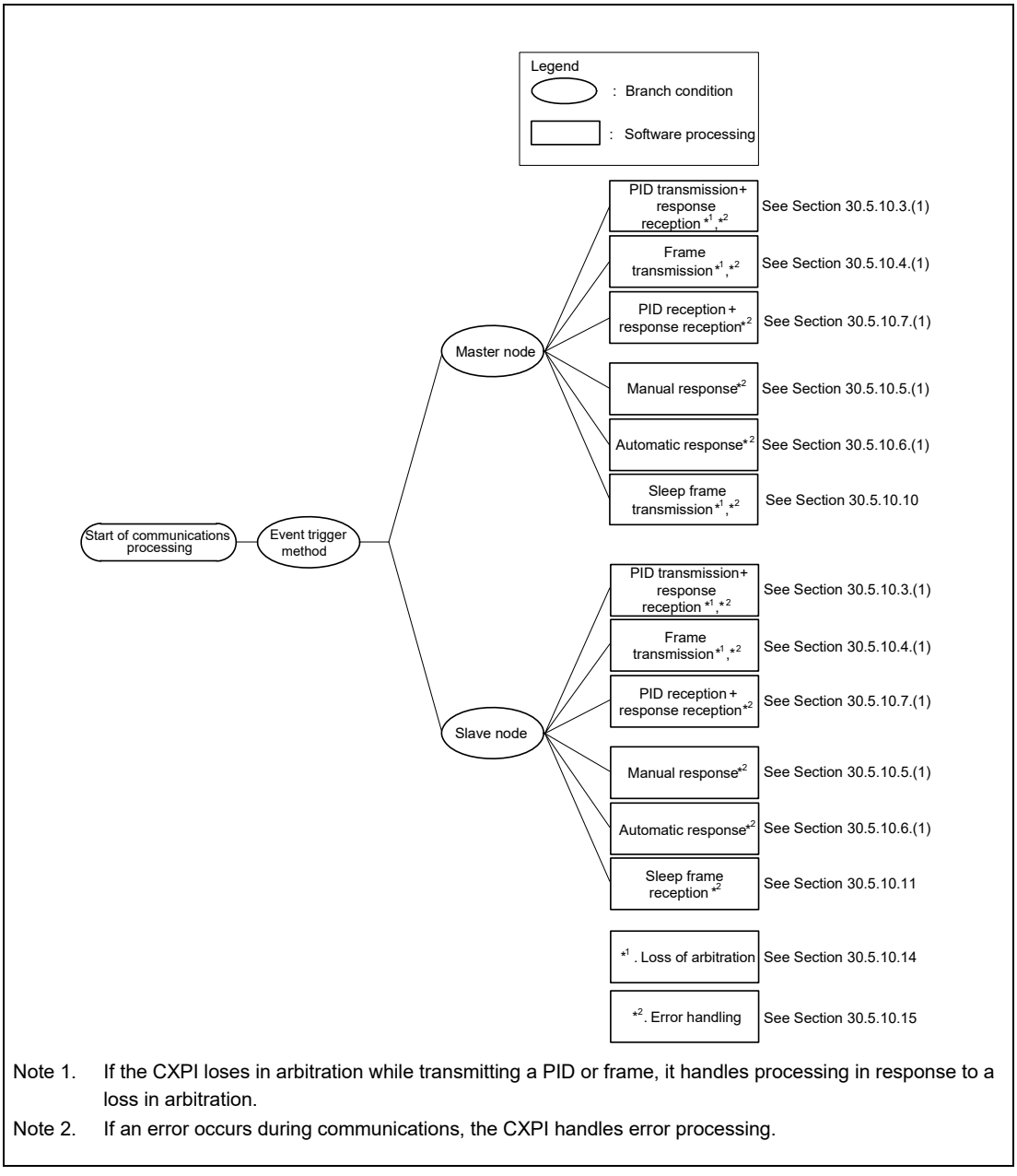


Figure 30.41 Overall Correlation Diagram at Startup by Other Node

(2) Communication Flow in the Event Trigger Method (CXP1nMODE.CXP1nACC = 0_B)

The figure below shows the communication flow in the event trigger method.



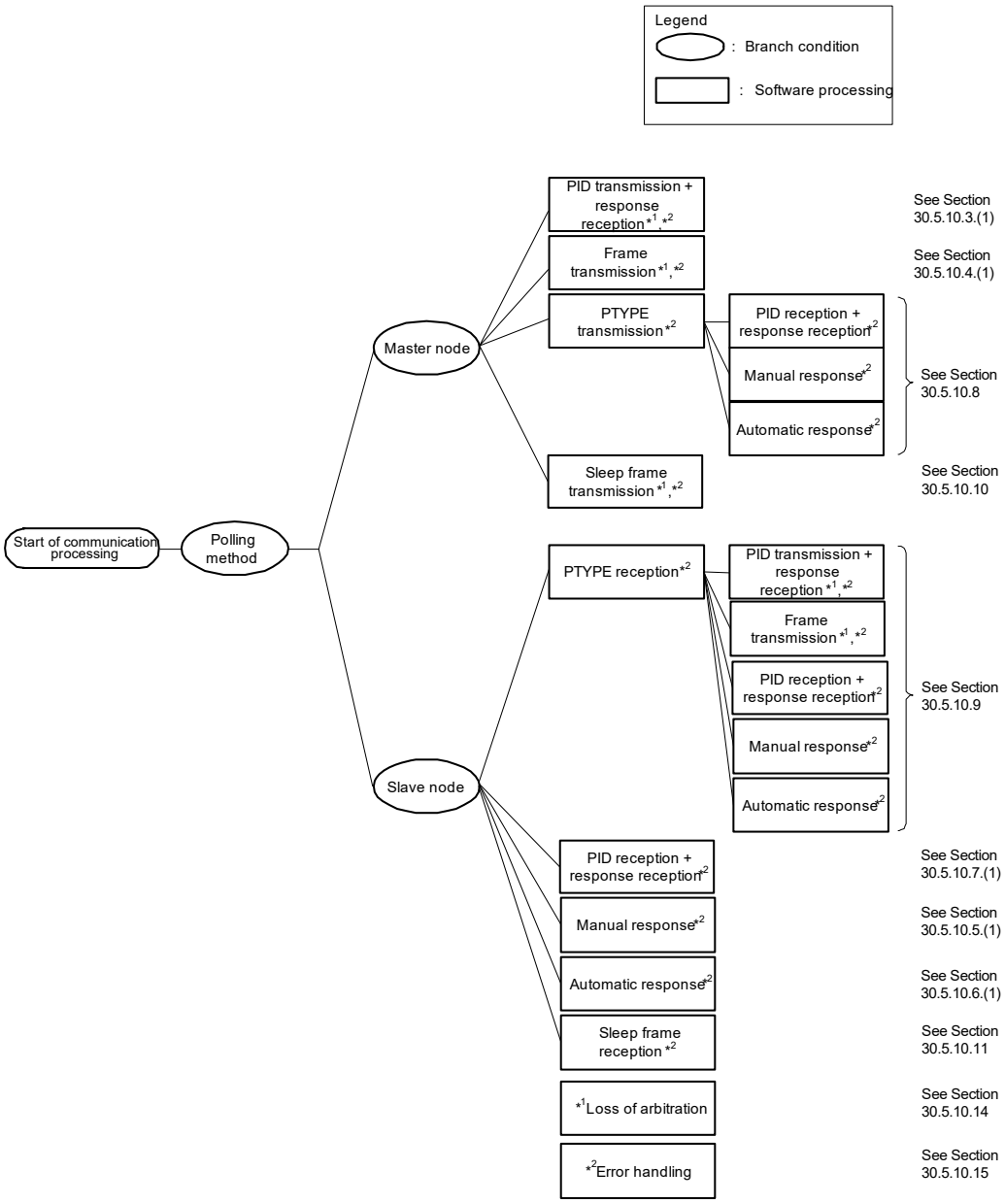
Note 1. If the CXPI loses in arbitration while transmitting a PID or frame, it handles processing in response to a loss in arbitration.

Note 2. If an error occurs during communications, the CXPI handles error processing.

Figure 30.42 Overall Correlation Diagram in the Event Trigger Method

(3) Communication Flow in the Polling Method (CXP1nMODE.CXP1nACC = 1_B)

The figure below shows the communication flow in the polling method.



Note 1. If the CXPI loses in arbitration while transmitting a PID or frame, it handles processing in response to a loss in arbitration.

Note 2. If an error occurs during communications, the CXPI handles error processing.

Figure 30.43 Overall Correlation Diagram in the Polling Method

30.5.10.2 Initial Settings

The initial settings differ depending on the transfer mode. This section describes the initial settings in each transfer mode.

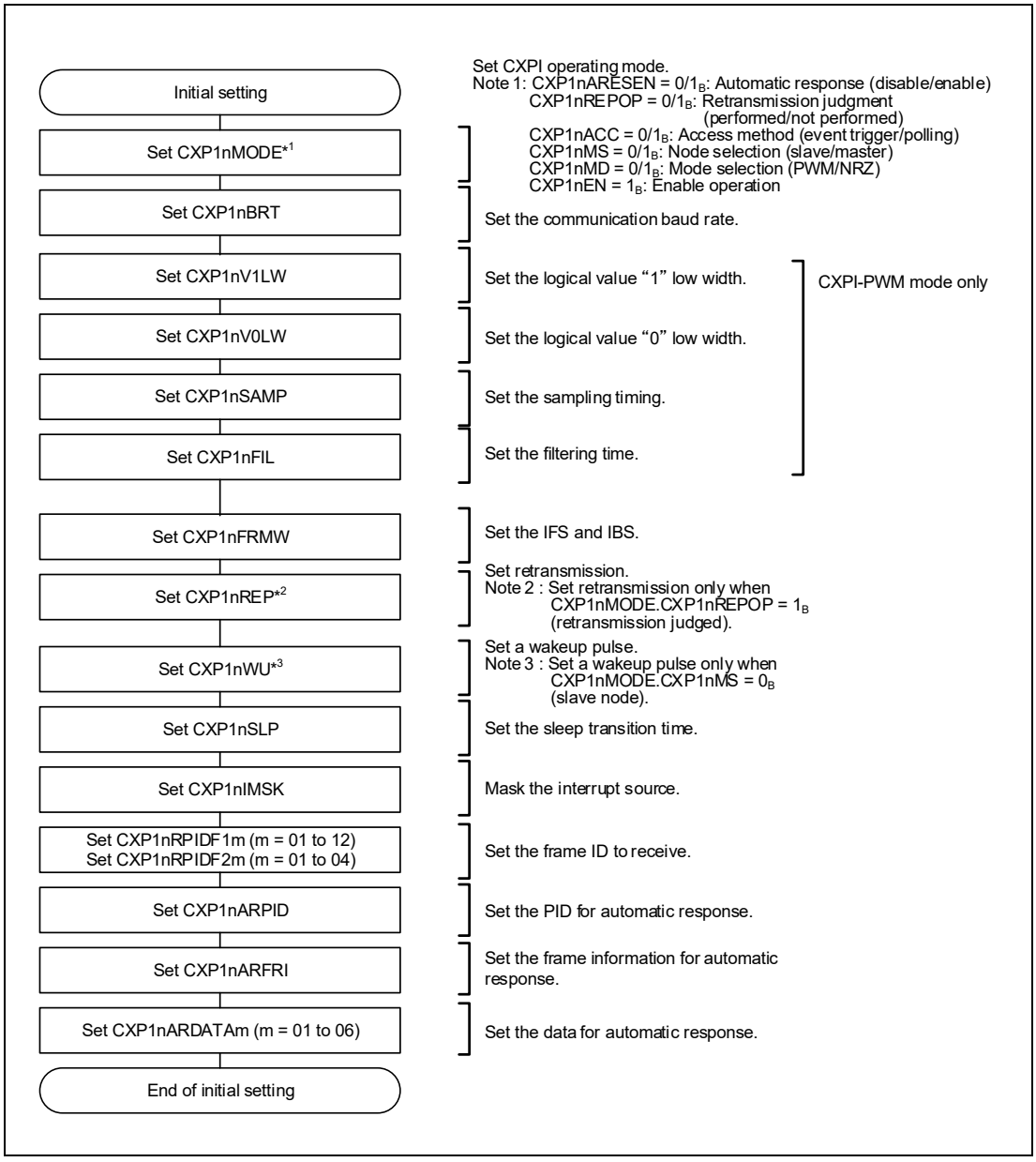


Figure 30.44 Initial Processing Flow

30.5.10.3 PID Transmission

(1) PID Transmission to Response Reception

(a) Normal Frame

The figure below shows the processing flow from PID transmission to response reception (for a normal frame).

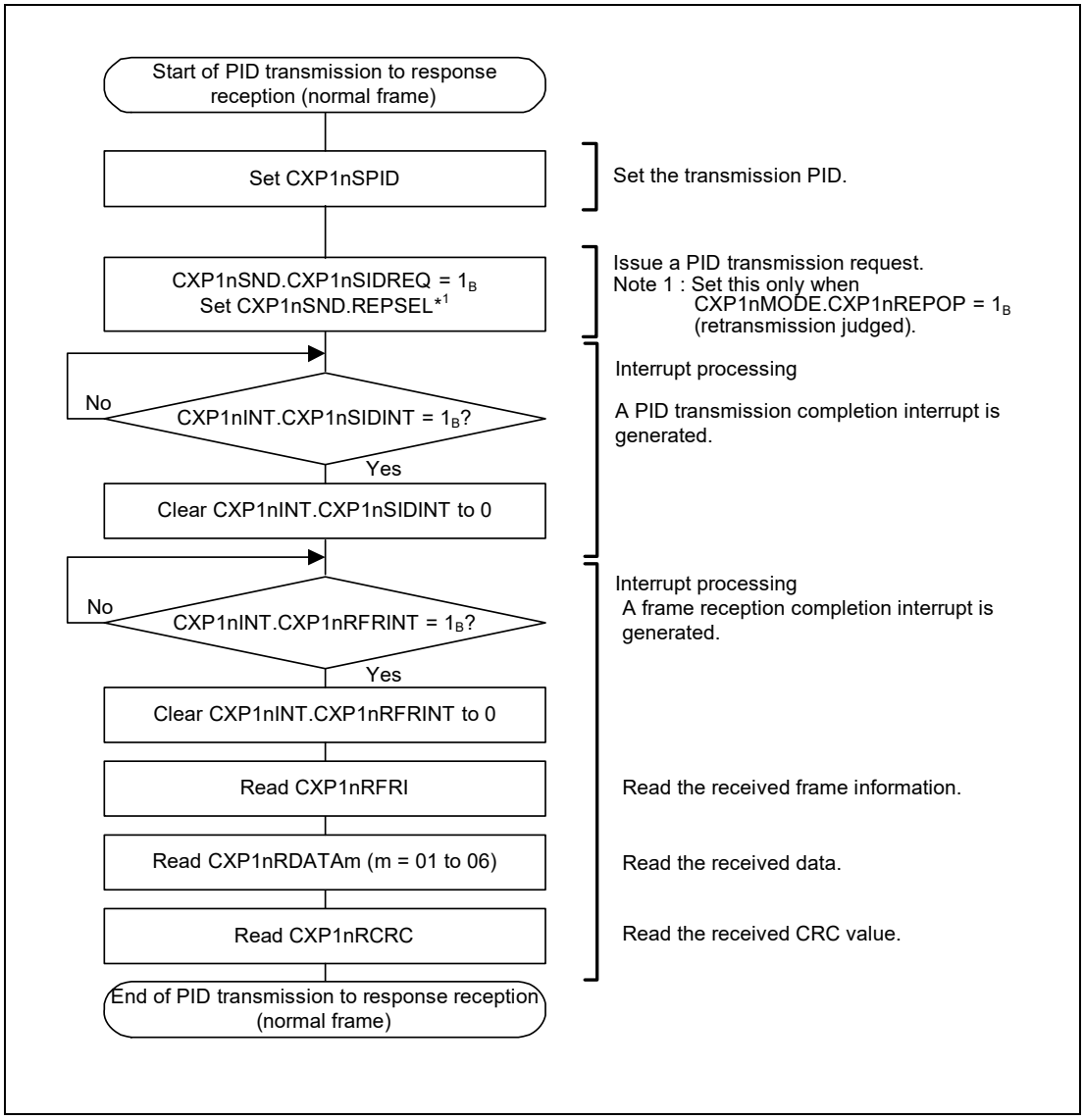


Figure 30.45 Processing Flow from PID Transmission to Response Reception (Normal Frame)

(b) Long Frame

The figure below shows the processing flow from PID transmission to response reception (for a long frame).

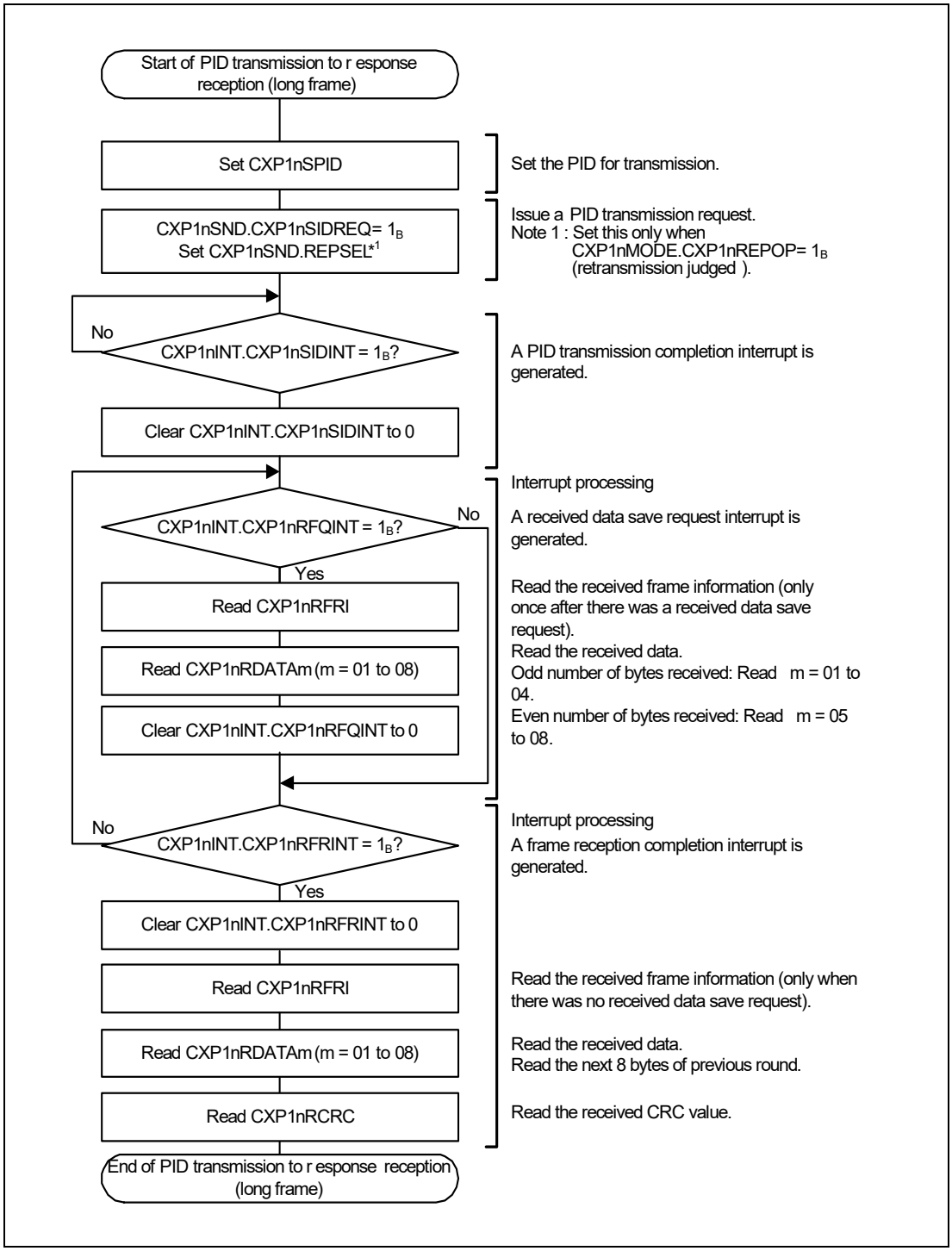


Figure 30.46 Processing Flow from PID Transmission to Response Reception (Long Frame)

30.5.10.4 Frame Transmission

(1) PID Transmission to Response Transmission

(a) Normal frame

The figure below shows the frame transmission processing flow from PID reception to response transmission (for a normal frame).

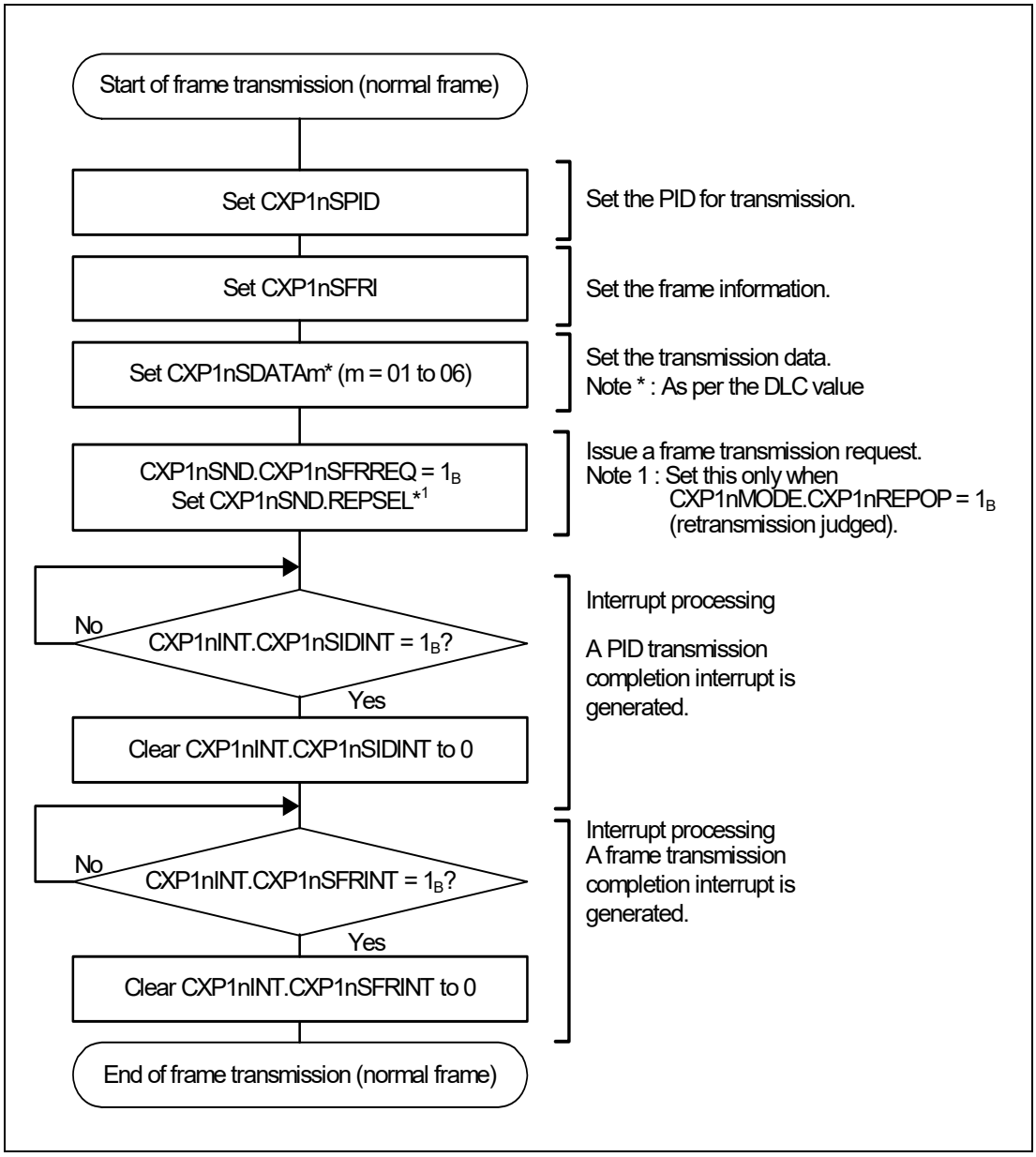


Figure 30.47 Frame Transmission Flow (Normal Frame)

(b) Long frame

The figure below shows the frame transmission processing flow from PID reception to response transmission (for a long frame).

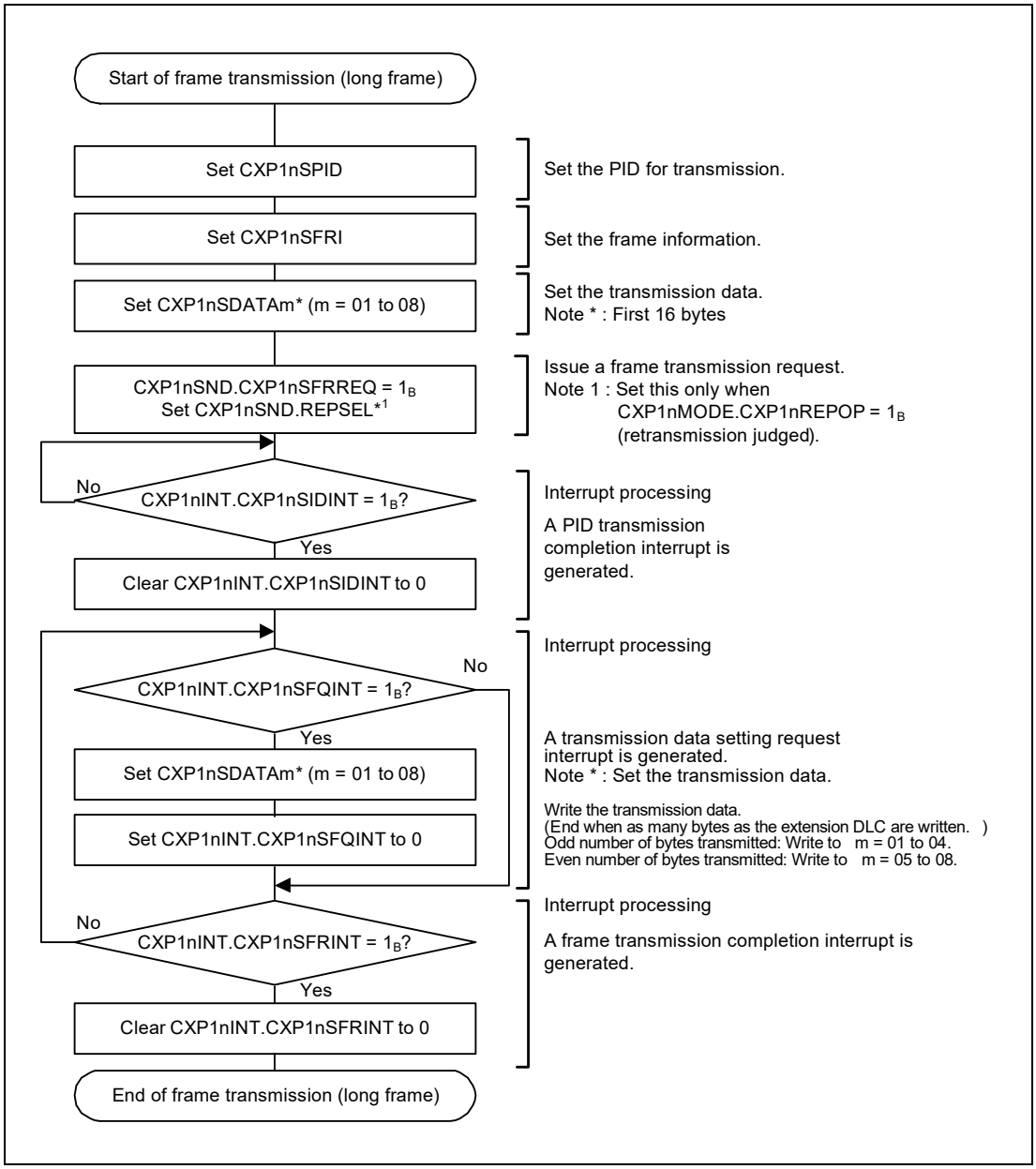


Figure 30.48 Frame Transmission Flow (Long Frame)

30.5.10.5 Manual Response

(1) PID Reception to Response Transmission

(a) Normal Frame

The figure below shows the manual response processing flow from PID reception to response transmission (for a normal frame).

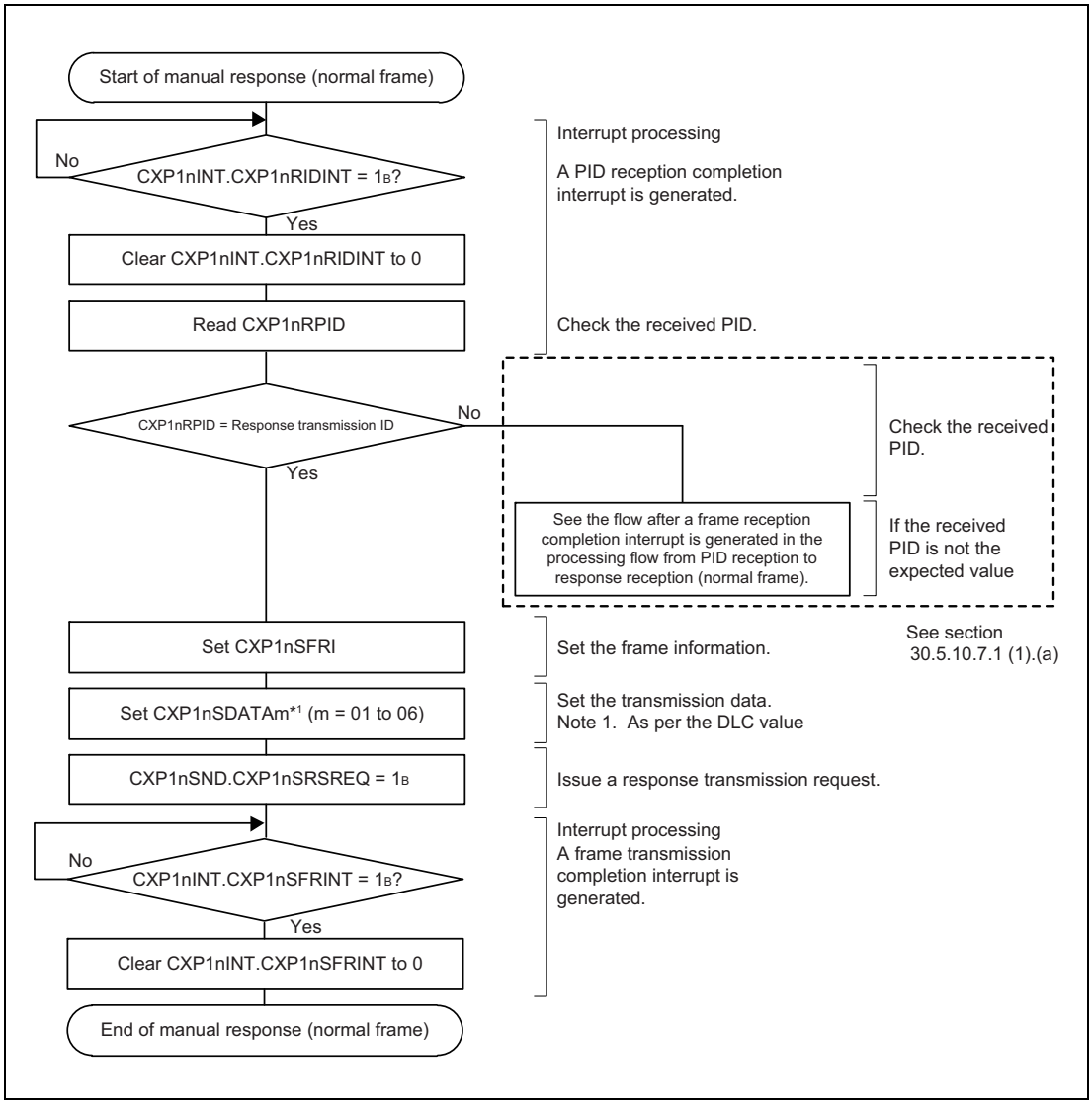


Figure 30.49 Manual Response Flow (Normal Frame)

(b) Long Frame

The figure below shows the manual response processing flow from PID reception to response transmission (for a long frame).

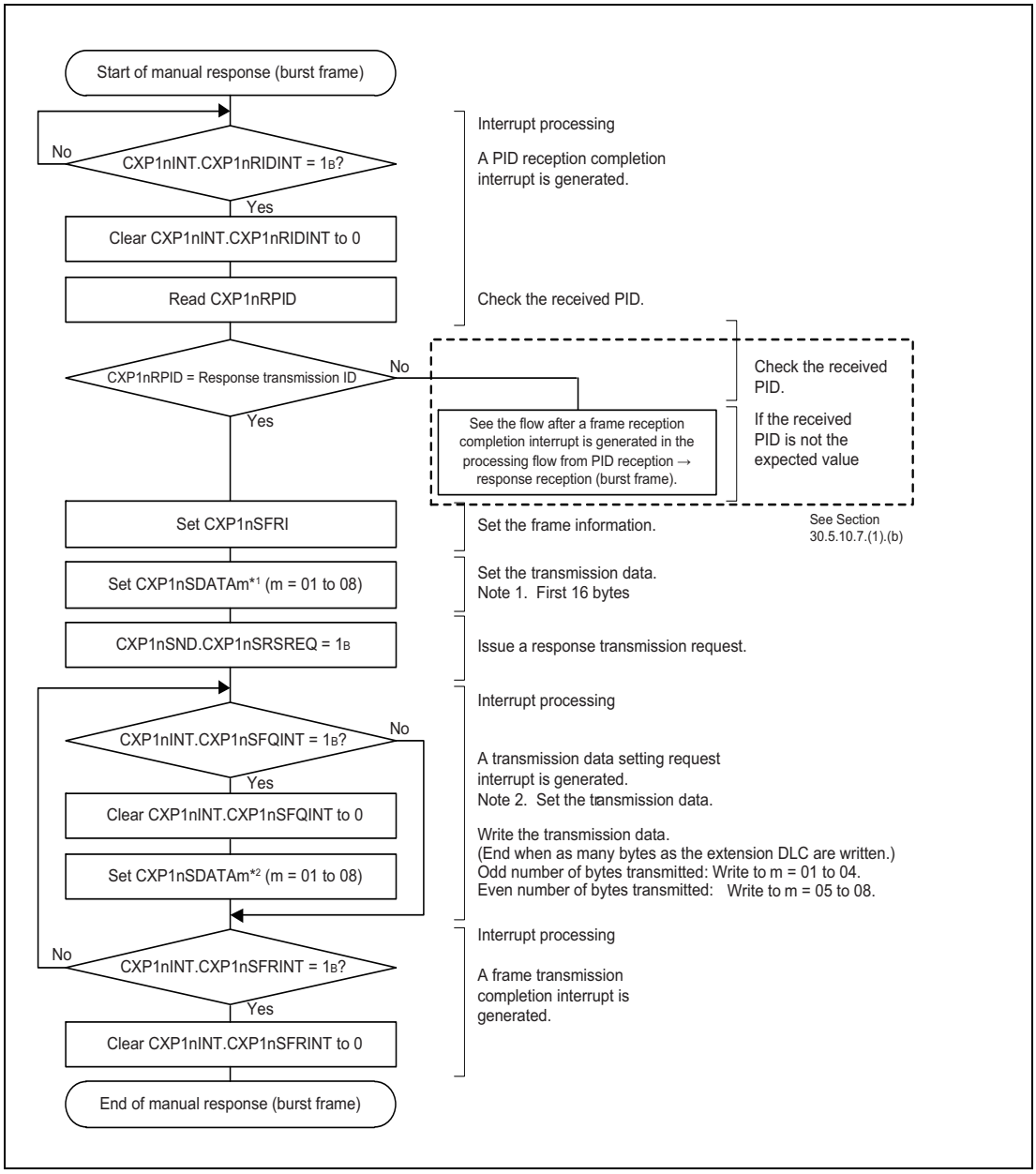


Figure 30.50 Manual Response Flow (Long Frame)

30.5.10.6 Automatic Response

(1) PID Reception to Response Transmission

The figure below shows the automatic response processing flow from PID reception to response transmission (for a normal frame).

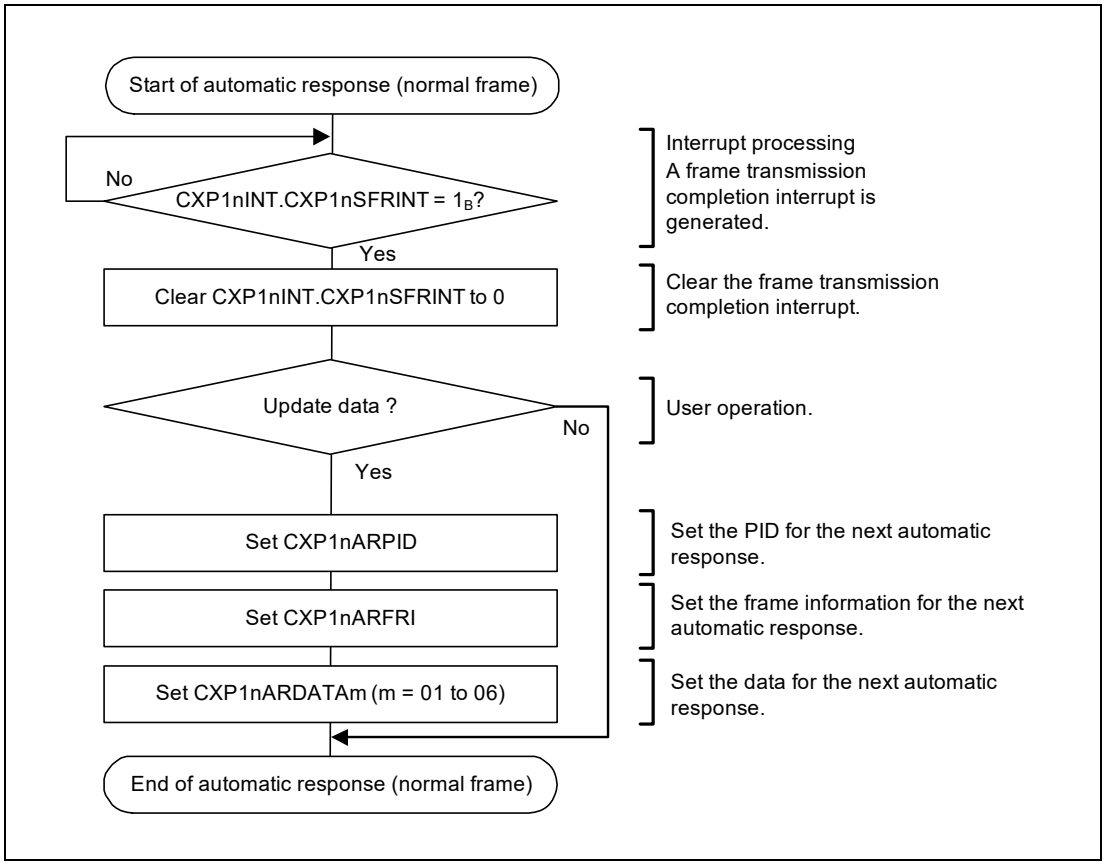


Figure 30.51 Automatic Response Flow (Normal Frame)

30.5.10.7 PID Reception

(1) PID Reception to Response Reception (Frame Reception)

(a) Normal Frame

The figure below shows the processing flow from PID reception to response reception (for a normal frame).

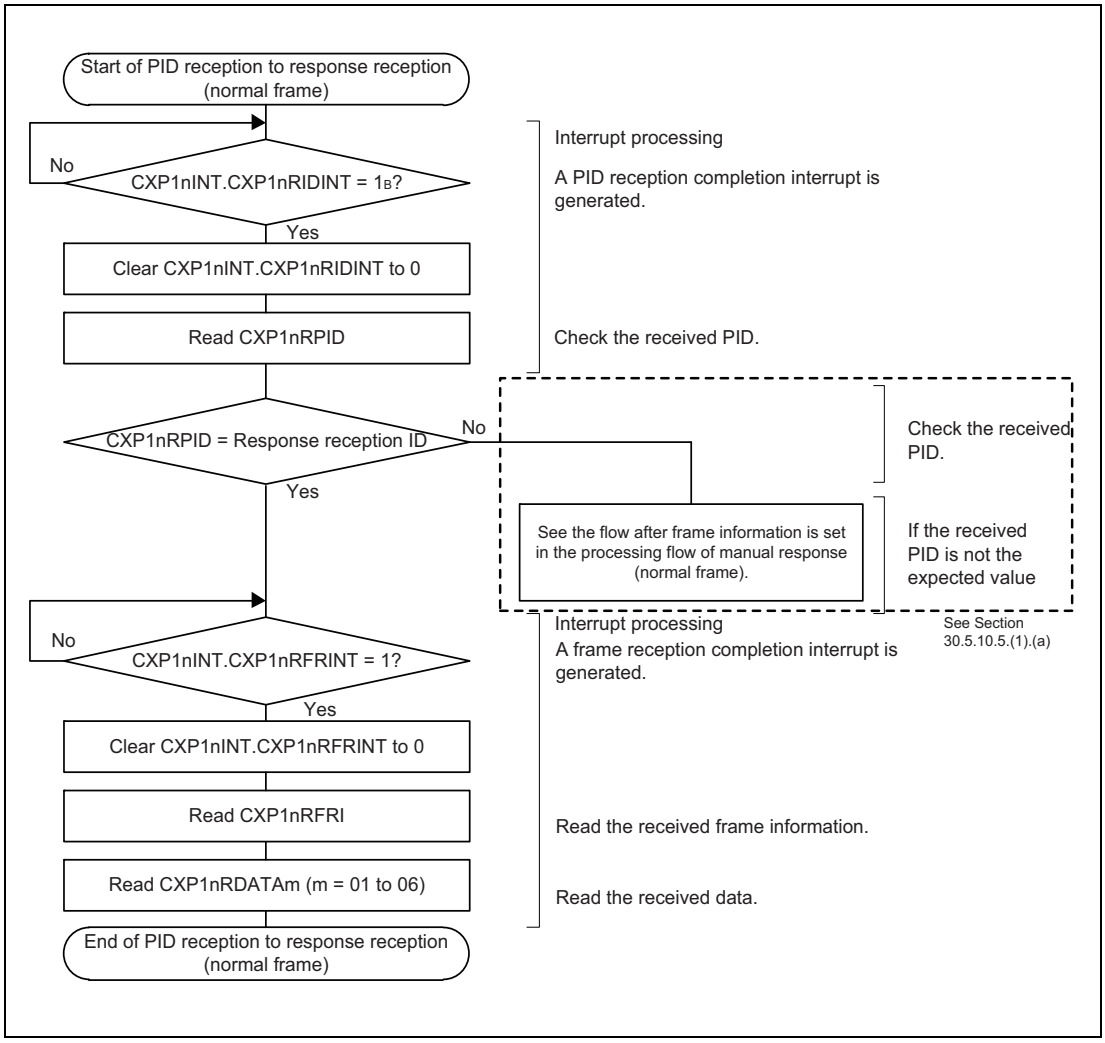


Figure 30.52 Processing Flow from PID Reception to Response Reception (Normal Frame)

(b) Long Frame

The figure below shows the processing flow from PID reception to response reception (for a long frame).

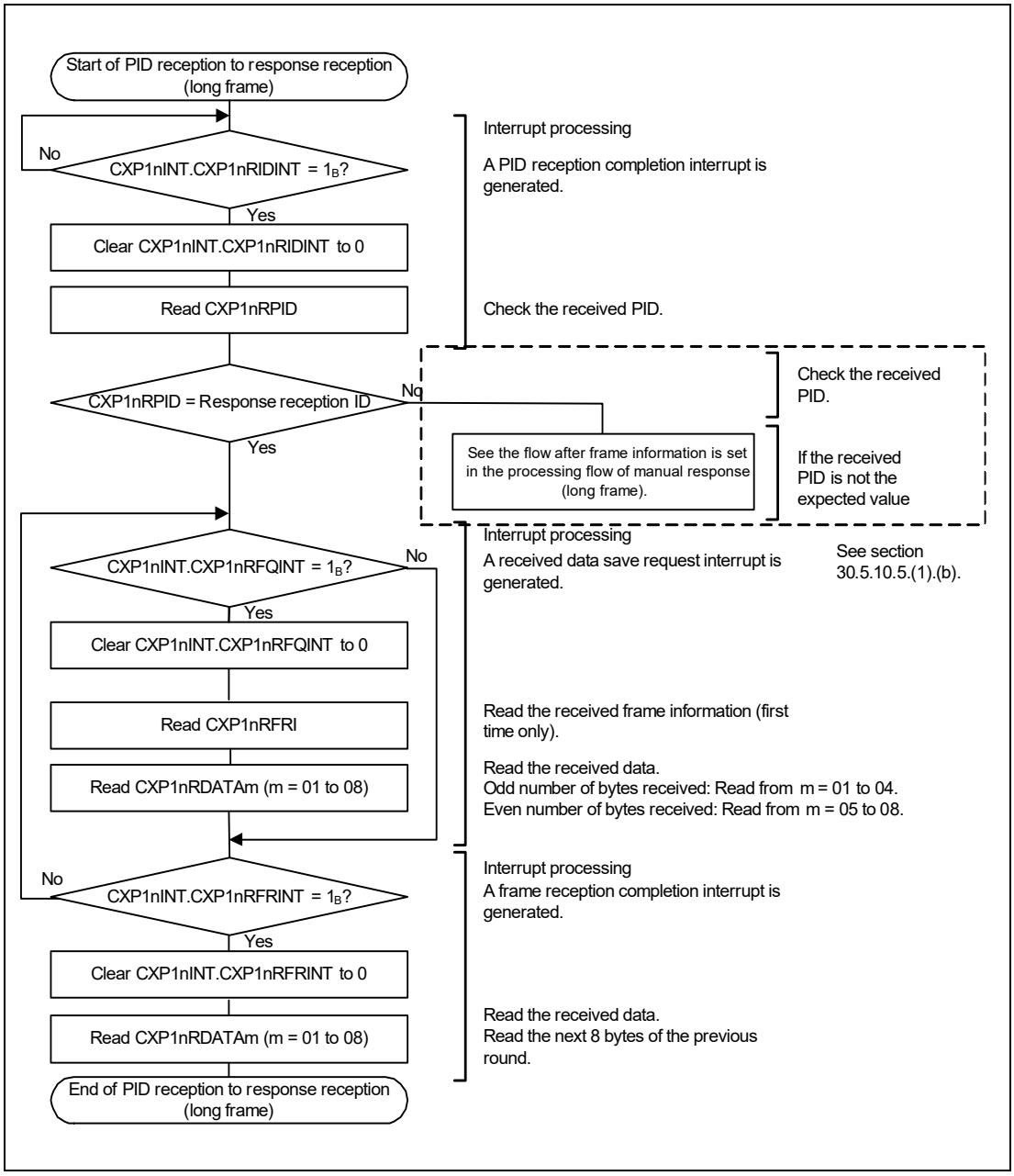
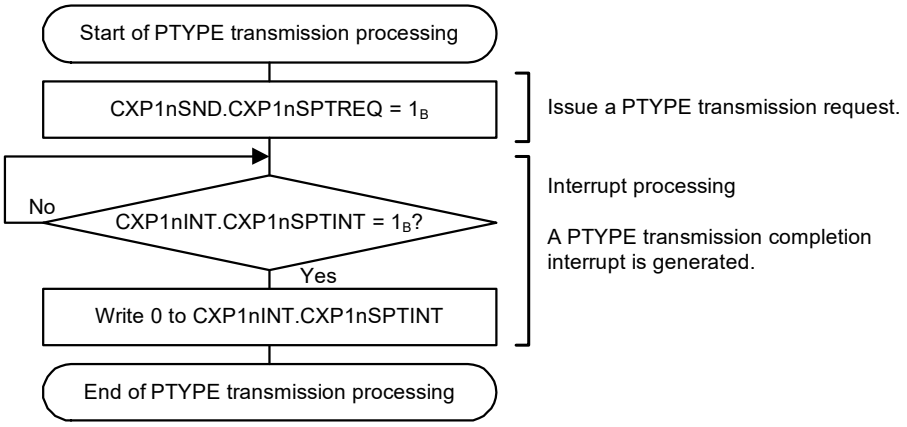


Figure 30.53 Processing Flow from PID Reception to Response Reception (Long Frame)

30.5.10.8 PTYPE Transmission

The figure below shows the processing flow for PTYPE transmission.

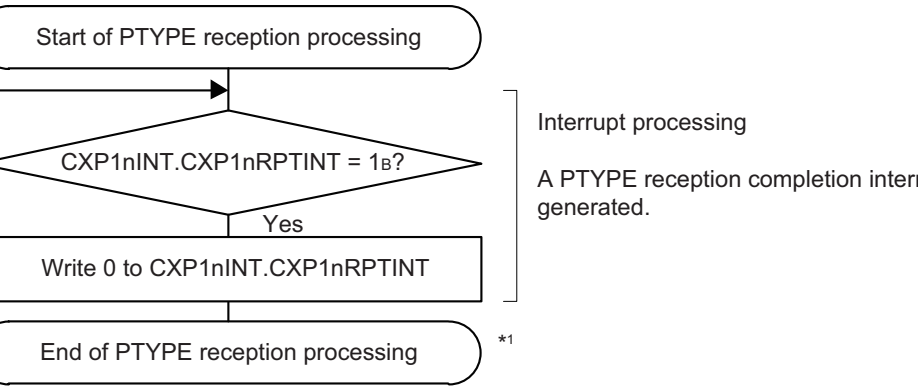


Note 1. If there is a response from the slave node after PTYPE transmission, any of the processing flows described in the following sections follows.
Section 30.5.10.5, Manual Response, **Section 30.5.10.6, Automatic Response**, or **Section 30.5.10.7, PID Reception**.

Figure 30.54 PTYPE Transmission Flow

30.5.10.9 PTYPE Reception

The figure below shows the processing flow for PTYPE reception.



Note 1. If the CXPI transmits a PID after receiving a PTYPE field and the PTYPE valid period hasn't passed (not CXP1nINT.CXP1nREXINT = 1), any of the processing flows follows.
Section 30.5.10.3, PID Transmission, **Section 30.5.10.4, Frame Transmission**, **Section 30.5.10.5, Manual Response**, **Section 30.5.10.6, Automatic Response**, or **Section 30.5.10.7, PID Reception**.

Figure 30.55 PTYPE Reception Flow

30.5.10.10 Sleep Frame Transmission

The figure below shows the processing flow for sleep frame transmission.

Don’t retransmit for sleep frame transmission.

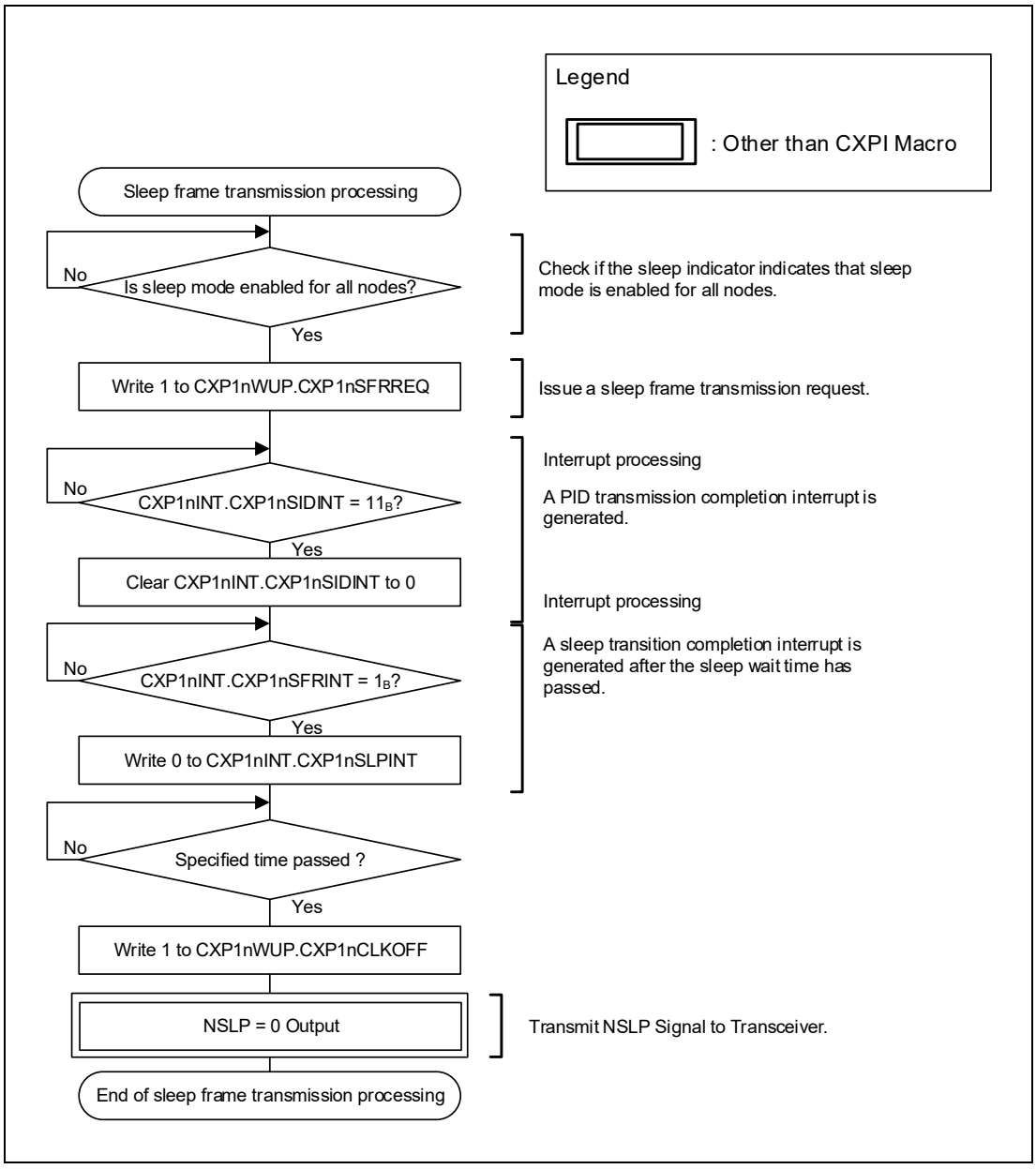


Figure 30.56 Sleep Frame Transmission Flow

30.5.10.11 Sleep Frame Reception

The figure below shows the processing flow for sleep frame reception.

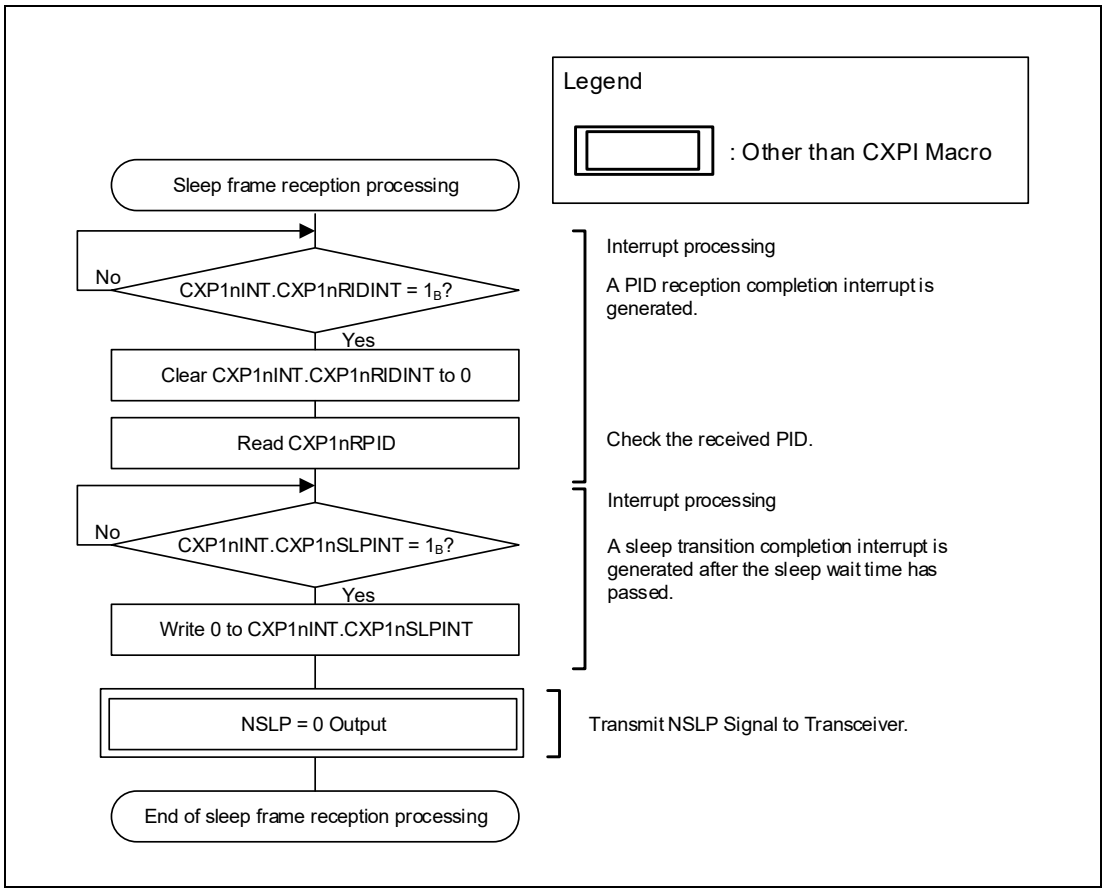


Figure 30.57 Sleep Frame Reception Flow

30.5.10.12 Wakeup Pulse Transmission

The wakeup pulse transmission operation differs depending on the transfer mode and master node/ slave node selection.

This section describes the wakeup pulse transmission operation under each operating condition.

(1) CXPI-PWM Mode

(a) Master Node

The figure below shows the processing flow for wakeup pulse transmission (by the master node).

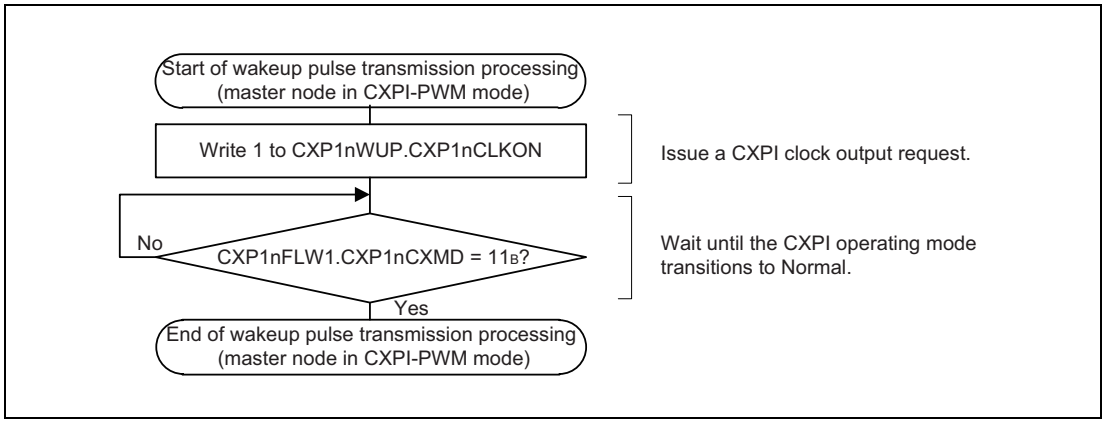


Figure 30.58 Flow of Wakeup Pulse Transmission (by Master Node in CXPI-PWM Mode)

(b) Slave Node

The figure below shows the processing flow for wakeup pulse transmission (by a slave node).

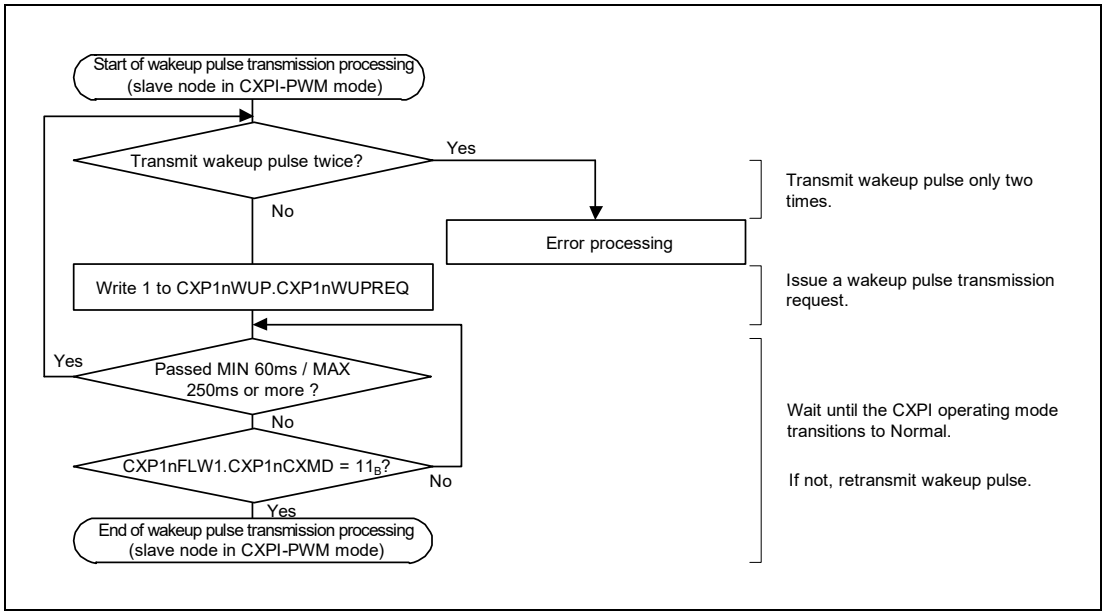


Figure 30.59 Flow of Wakeup Pulse Transmission (by Slave Node in CXPI-PWM Mode)

(2) CXPI-NRZ Mode

In CXPI-NRZ mode, the IP module controls the wakeup pulse transmission by controlling the CXPI transceiver and a module other than this IP. For the detailed operation, see **Section 30.5.1, CXPI System Configuration**.

(a) Master Node

The figure below shows the processing flow for wakeup pulse transmission (by the master node).

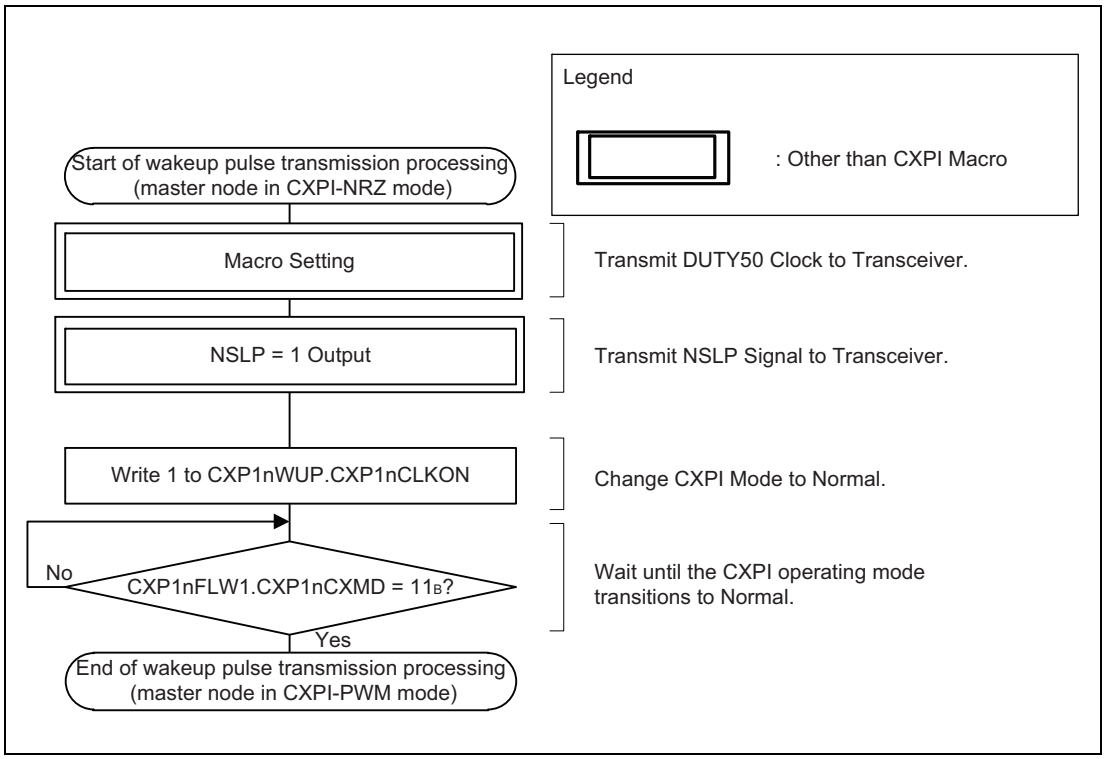


Figure 30.60 Flow of Wakeup Pulse Transmission (by Master Node in CXPI-NRZ Mode)

(b) Slave Node

The figure below shows the processing flow for wakeup pulse transmission (by a slave node).

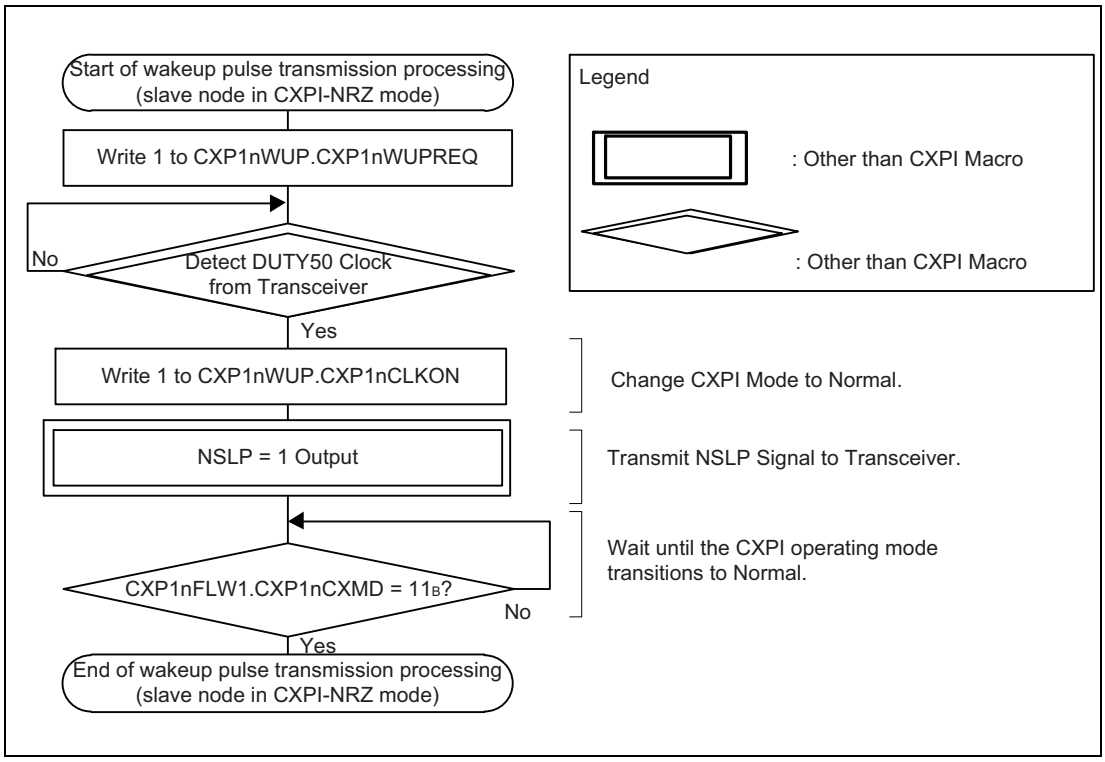


Figure 30.61 Flow of Wakeup Pulse Transmission (by Slave Node in CXPI-NRZ Mode)

30.5.10.13 Wakeup Pulse Reception

This IP does not have the wakeup pulse reception function. It detects a wakeup pulse by using the CXPI transceiver and an external function outside this IP. For the detailed operation, see **Section 30.5.1, CXPI System Configuration**.

30.5.10.14 Processing in Response to a Loss in Arbitration

The CXPI handles processing in response to a loss in arbitration if it has completed reception of a PID or frame while transmitting a PID.

The figure below shows the flow of processing in response to a loss in arbitration.

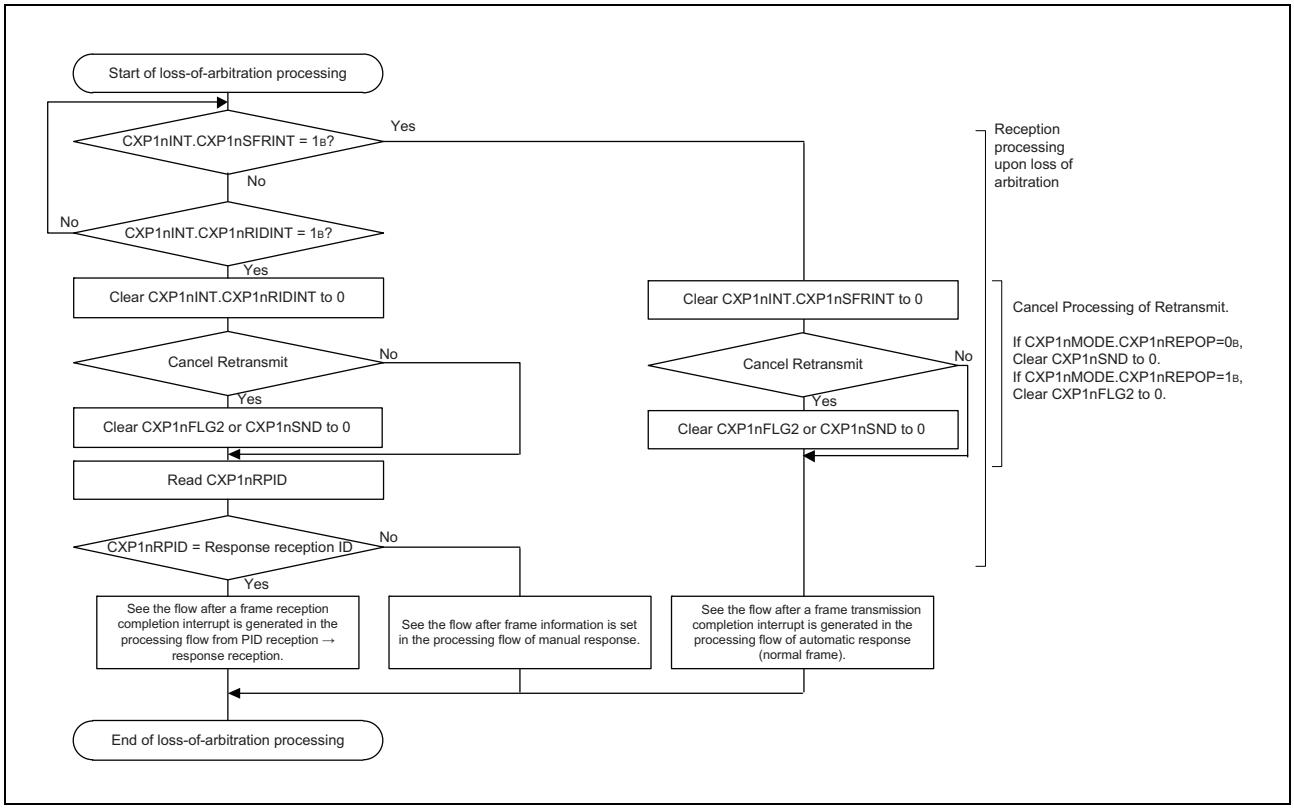


Figure 30.62 Flow of Processing in Response to a Loss in Arbitration

30.5.10.15 Error Handling

The figure below shows the error handling flow.

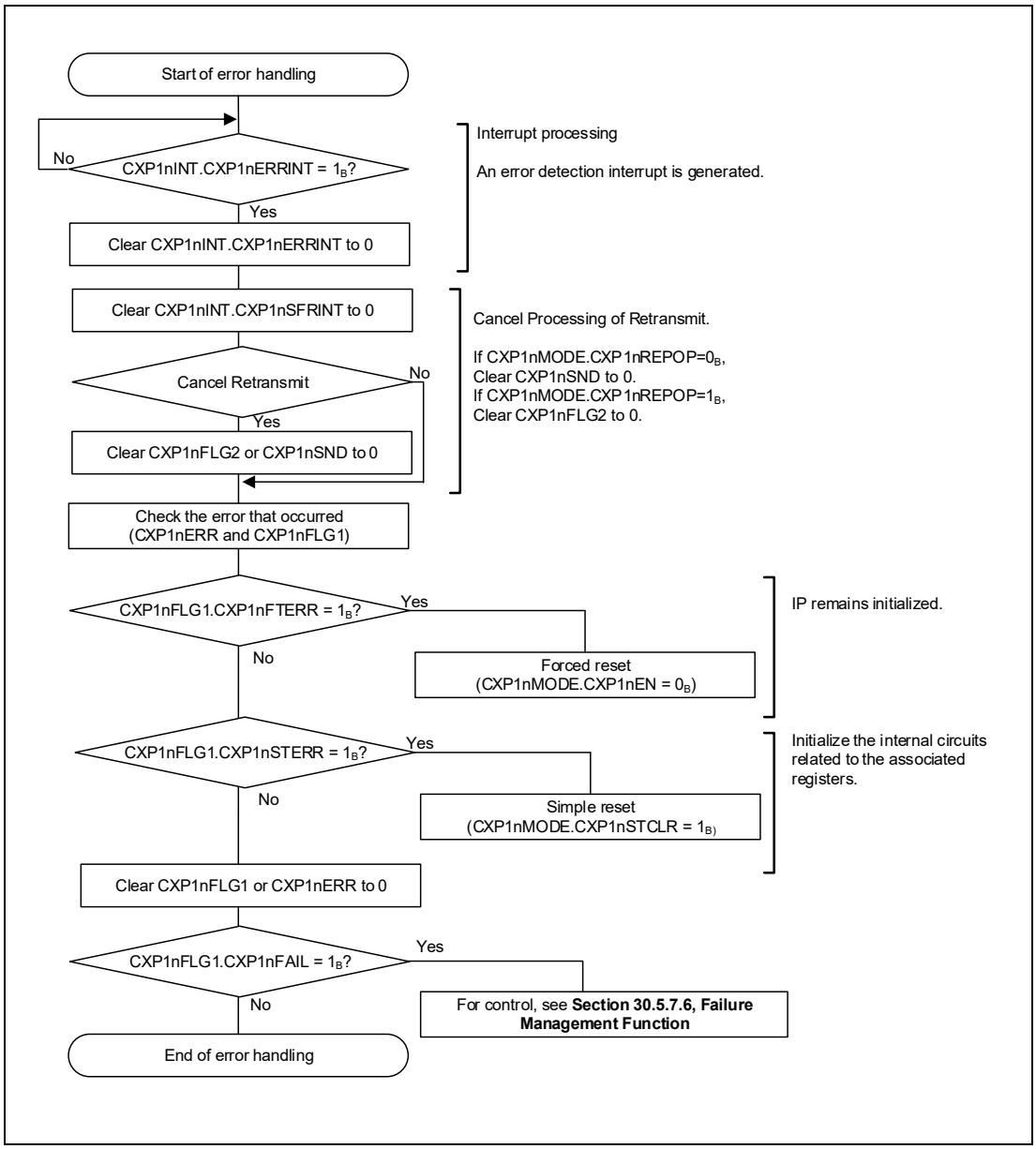


Figure 30.63 Error Handling Flow

30.6 Notes

30.6.1 Supplement Notes

30.6.1.1 On-Board Noise

The specification of the CXPI controller is on the assumption of negligible on-board noise between the LSI chip and the CXPI transceiver (the lines in the red broken-line boxes in the following figure). If on-board noise is present, take measures against it.

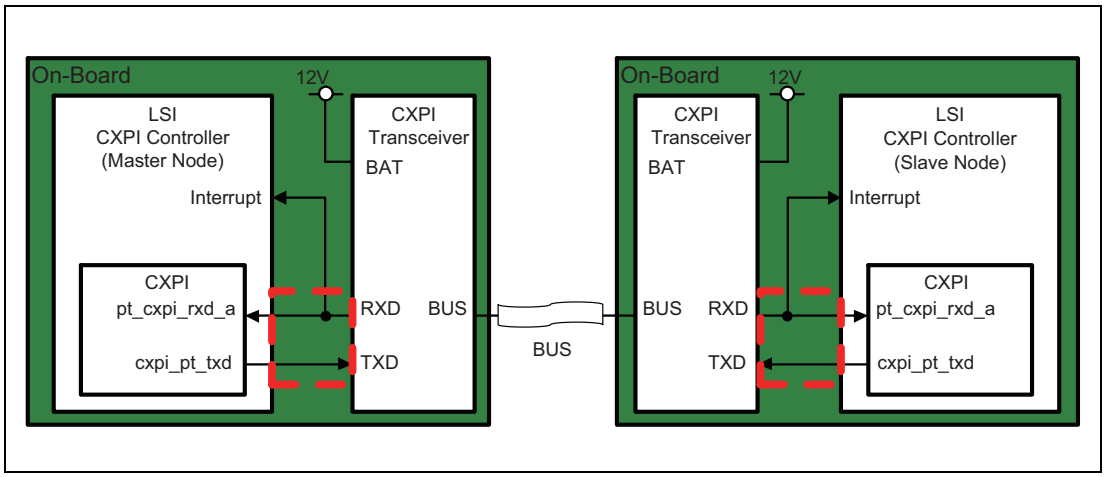


Figure 30.64 On-Board Noise

The following gives example of operations when on-board noise occurred between the LSI and the CXPI transceiver.

Operating conditions:

- CXPI-PWM mode
- Noise occurring during transmission of the start bit of a PID.
- Operation for retransmission being enabled.

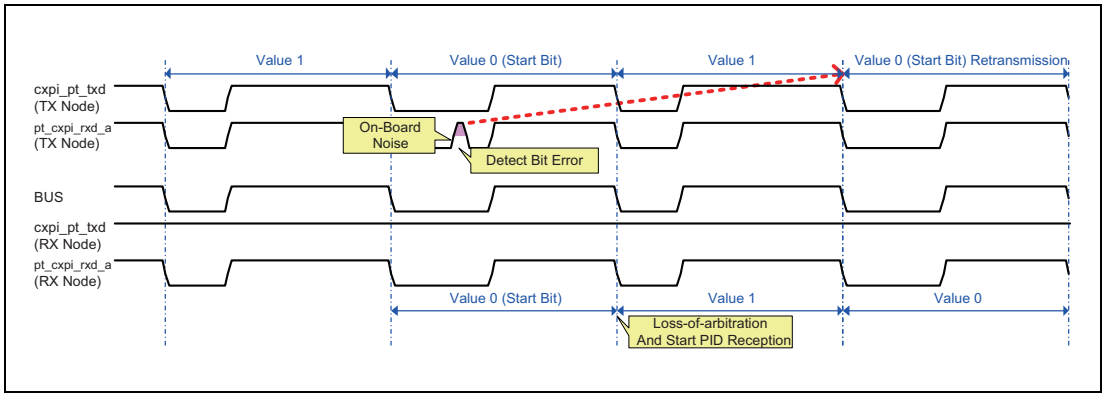


Figure 30.65 Example of Operations when On-Board Noise Occurred

If noise occurs in start bit of PID transmission in CXPI-PWM mode, local node detects bit error and performs re-transmission immediately. However, since logical value 0 (start bit) is transmitted on transmission bus, other nodes switch to receive operation from initial start bit. Therefore, other nodes receive in the order of start bits according to previous start bit + logical value 1 (interval until re-transmission) + re-transmission and receives unexpected PID.

As CXPI standard compliant operation, re-transmission should be performed at the time of bit error detection, after waiting until IFS period if logical value 0 is transmitted to transmission bus. But, because the specification does not take into consideration the board noise, this IP does not support this operation.

30.6.1.2 Loss in Arbitration in CXPI-PWM Mode

For CXPI standard arbitration, arbitration fail is judged if transmitted data and received data are not matching. Since this IP is also compliant with that standard, following abnormal operation may occur.

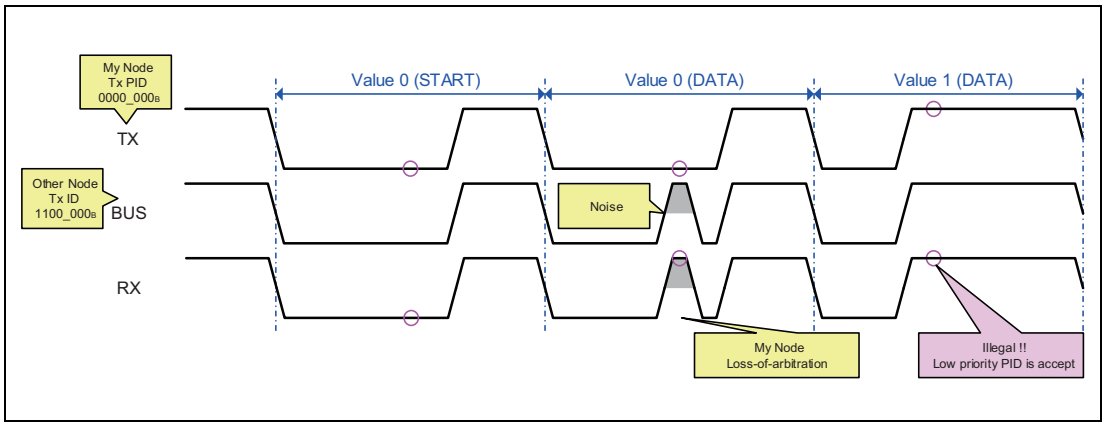


Figure 30.66 Low priority PID is transmitted case

When local node transmits dominant level output (high priority PID 0000000_B) and other node transmits recessive level output (low priority PID 1100010_B) PID, if recessive level noise occurs in transmission bus, local node judge's arbitration fail from mismatch of transmitted data and received data, stops transmission and starts receive. Other node continues PID transmission since transmitted data and received data are matching. Therefore, arbitration fail occurs for high priority PID but low priority PID is transmitted on transmission bus.

In addition, due to same reason, following unexpected communication may be established when only local node transmits PID.

If noise occurs in transmission bus during PID transmission, local node will judge arbitration fail and stop transmission (bit error is generated if noise occurs in start bit). Unexpected PID is transmitted on transmission bus because PID transmission is stopped and PID transmitted by other node is received normally (in case of data without parity error). Therefore, transmission may be established with unexpected PID but since this is considered as normal data transmission on communication bus, there is no problem with the operation.

30.6.2 Restrictions

30.6.2.1 Restriction of Register RMW (read-modify-write)

In case of RMW to a register with HW set clear conditions, values changed in HW may be overwritten unintentionally. Take care in case of RMW of following registers. It overwrites values not checked with SW read.

- CXPI Transmission Control Register (CXP1nSND)
  Observe the notes (**Section 30.6.2.5, Restriction of the Transmission Holding**).
- CXPI Interrupt Source Register (CXP1nINT)
  Since CXP1nINT.CXP1nERRINT and other bits change at the shortest 1 Tbit, complete RMW in the meantime.
- CXPI Flag Register 1 (CXP1nFLG1)
  When CXP1nFLG1.CXP1nRFTFID RMW, ignore the other bits. When referring to registers other than CXP1nFLG1.CXP1nRFTFID, do not RMW to CXP1nFLG1.CXP1nRFTFID.
- CXPI Flag Register 2 (CXP1nFLG2)
  Observe the notes (**Section 30.4.1.16, CXP1nFLG2 — CXPI Flag Register 2**).
- CXPI Error Register (CXP1nERR)
  Complete RMW within the interval (8 Tbit) at which each error is set.

30.6.2.2 Restriction of IFS Period and IBS Period Associated with Transceiver Delay

Although IFS period and IBS period are compliant with CXPI standards and throughout the manual they are explained with Tbit unit which is the basic bit, IFS period and IBS period are different per modes as given below.

- At the time of CXPI-PWM mode
  (CXP1nFRMW.CXP1nIFS setting value + 1) × IFS period of basic bit
- At the time of CXPI-NRZ mode
  (CXP1nFRMW.CXP1nIFS setting value + 1) × IFS period of basic bit
- At the time of CXPI-PWM mode
  CXP1nFRMW.CXP1nIBS1orIBS2 setting value × IBS period of basic bit
- At the time of CXPI-NRZ mode
  CXP1nFRMW.CXP1nIBS1orIBS2 setting value × IBS period of basic bit

Also, as shown in follow figure, IFS period and IBS period on transmission bus vary considerably due to transceiver delay. Note that the transceiver delay is not taken into consideration in this manual and the explanation is based on input output terminals of microcomputer.

Moreover, when the delay difference of less than 1 Tbit exists in TX and RX, per the delay difference, the reception start condition / PTYPE term of validity of an IFS period / IBS period / frame get mixed up 1 Tbit.

Build the system which secured the IFS period and the IBS period which is generous in consideration (more than or equal to 1 Tbit) of these delays.

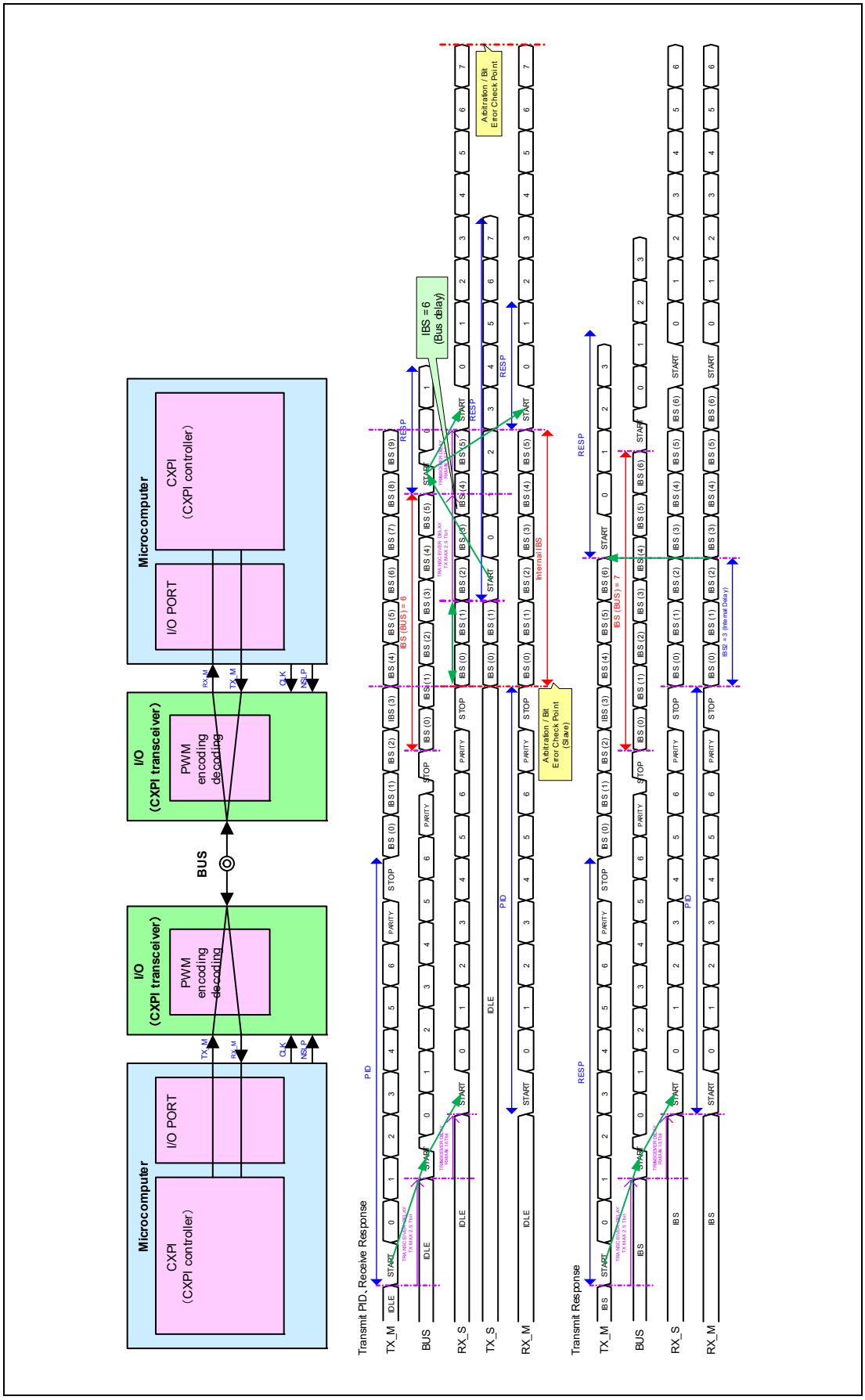


Figure 30.67 Effect of transceiver delay in NRZ mode

30.6.2.3 Restriction of PID Transmit after PTYPE Transmit and Receive

About the control after PTYPE transmission, note following that.

1. To Less than 10 Tbit after transition PTYPE, frame or PID transmit is prohibited.
2. After detecting PTYPE validity completion interrupt after a PTYPE transmission, the transmission the idle state saves and by which PID and a frame is possible by a CXPI controller. Therefore please send PID or a frame after a IFS period on the bus is measured by software.

About the control after PTYPE reception, note following that.

1. The PID transmitting operation after PTYPE reception should build a system with sufficient margin for IBS period.
2. When detecting PID reception complete before PTYPE transmission completion, please break the communication behavior. There is a possibility that a slave node is sending PID without PTYPE reception (though that does not follow CXPI standard spec), but this module has no function to detect it as an error though PID can receive normally.

30.6.2.4 Restriction of Transmission Request

When CXPI bus not idle (CXP1nFLW1.RXIDLE = 0_B) and IDLE state (CXP1nFLW1.CXP1nCXST = IDLE), transmission (write 1 to CXP1nSND. CXP1nSPTREQ CXP1nSND. CXP1nSFRREQ CXP1nSND. CXP1nSIDREQ) is prohibited.

30.6.2.5 Restriction of the Transmission Holding

This IP have the function of the transmission holding as follows: Write 1 to CXP1nSND register, Loss-of-arbitration retransmission remaining, Error retransmission remaining.

Note the more than one transmission holding cause difficulties. Be sure to set the transmission request of CXP1nSND one by one (one hot).

30.6.2.6 Restriction of Retransmission

Note follow these.

1. Both CXP1nFLG2.CXP1nEREP and CXP1nFLG2.CXP1nAREP are used for retransmission when detecting bit error with PID transmission of CXPI-NRZ mode.
2. Clear CXP1nFLG2.CXP1nEREP when frame or PID transmission is finished with “Retransmission After an Error Occurred (Retransmission Judged)”.

30.6.2.7 Restriction of Request Frame ID

The master node can transmit request frame ID (PID = $1F_H$). The slave node can receive request frame ID (PID = $1F_H$).

Use “Frame Transmission” if transmit request function ID.

When a response is not able to receipt by abnormalities after receive frame ID (PID = $1F_H$), be forced reset (CXP1nMODE.CXP1nEN = 0).

Section 31 Window Watchdog Timer (WDTB)

This section contains a generic description of the Window Watchdog Timer (WDTB).

The first part of this section describes properties specific to the RH850/U2A-EVA, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the WDTB.

31.1 Features WDTB for RH850/U2A-EVA

31.1.1 Number of Units and Channels

This microcontroller has the following number of WDTB units.

Each WDTB unit has one channel interface.

Table 31.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	5 (n = 0 to 3, A)	5 (n = 0 to 3, A)	5 (n = 0 to 3, A)	5 (n = 0 to 3, A)	3 (n = 0, 1, A)	3 (n = 0, 1, A)	3 (n = 0, 1, A)	3 (n = 0, 1, A)	3 (n = 0, 1, A)	3 (n = 0, 1, A)
Name	WDTBn/WDTBA									

Table 31.2 Index

Index	Description
n	In ISO domain, the individual units of window watchdog timer are identified by the index “n”, For example, WDTBnWDTE for the WDTBn enable register. Throughout this section, “n” will range from 0 to 3.
A	In AWO domain, the window watchdog timer unit is identified by the index “A”, For example, WDTBAWDTE for the WDTBA enable register.

31.1.2 Register Base Addresses

WDTBn/WDTBA base addresses are listed in the following table.

WDTBn/WDTBA register addresses are given as offsets from the base addresses.

Table 31.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<WDTB0_base>	FFBF 1000_H	P-Bus Group 5
<WDTB1_base>	FFBF 1100_H	P-Bus Group 5
<WDTB2_base>	FFBF 1200_H	P-Bus Group 5
<WDTB3_base>	FFBF 1300_H	P-Bus Group 5
<WDTBA_base>	FF9A 5000_H	P-Bus Group 2L

31.1.3 Clock Supply

The window watch dog timer uses WDTBTCKI and PCLK as a clock input.

Table 31.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
WDTBA	PCLK	CLK_LSB
	WDTBTCKI	CLKA_WDT
WDTBn	PCLK	CLK_HSB
	WDTBTCKI	CLK_WDT

Note: For details of the clock, see **Section 13, Clock Controller**.

31.1.4 Interrupt Requests and Error Notifications

WDTBn/WDTBA interrupt requests are listed in the following table.

Table 31.5 Interrupt Requests

Interrupt Symbol Name	Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Wake up
INTWDTB0TIT	WDTB0TIT	WDTB0 interrupt	22	—	—
INTWDTB1TIT	WDTB1TIT	WDTB1 interrupt	22	—	—
INTWDTB2TIT	WDTB2TIT	WDTB2 interrupt	22	—	—
INTWDTB3TIT	WDTB3TIT	WDTB3 interrupt	22	—	—
INTWDTBATIT	WDTBATIT	WDTBA interrupt	705	—	√

The error notifications of this module are listed in the following table.

Table 31.6 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
Watchdog timer error (PE0)	WDTB0 error	228	—
Watchdog timer error (PE1)	WDTB1 error	260	—
Watchdog timer error (PE2)	WDTB2 error	292	—
Watchdog timer error (PE3)	WDTB3 error	324	—

Note: WDTBA does not send error notification to ECM.

31.1.5 Reset Sources

WDTB reset sources are listed in the following table. WDTB is initialized by these reset sources.

Table 31.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
WDTBn	All registers	√	√	√	√	√	—	—
WDTBA	All registers	√	√	√	√	—	—	—

31.1.6 Correspondence of WDTBn/WDTBA and CPU

Table 31.8 Correspondence of WDTB and CPU

WDTB	CPU Unit										ISO/ AWO domain
	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)	
WDTB0	CPU0	CPU0	CPU0	CPU0	CPU0	CPU0	CPU0	CPU0	CPU0	CPU0	ISO
WDTB1	CPU1	CPU1	CPU1	CPU1	CPU1	CPU1	CPU1	CPU1	CPU1	CPU1	ISO
WDTB2	CPU2	CPU2	CPU2	CPU2	—	—	—	—	—	—	ISO
WDTB3	CPU3	CPU3	CPU3	CPU3	—	—	—	—	—	—	ISO
WDTBA	selectable	selectable	selectable	selectable	selectable	selectable	selectable	selectable	selectable	selectable	AWO

31.1.7 Internal Input/Output Signals

Table 31.9 Internal Input/Output Signals

Unit Signal Name	Description	Connection
WDTBATRES	WDTBA error	Reset Controller
WDTB0TRES	WDTB0 error	ECM
WDTB1TRES	WDTB1 error	ECM
WDTB2TRES	WDTB2 error	ECM
WDTB3TRES	WDTB3 error	ECM
WDTB0TNMI	WDTB0 error for non-maskable interrupt	PIC1
WDTB1TNMI	WDTB1 error for non-maskable interrupt	PIC1
WDTB2TNMI	WDTB2 error for non-maskable interrupt	PIC1
WDTB3TNMI	WDTB3 error for non-maskable interrupt	PIC1

31.2 Overview

31.2.1 Functional Overview

The window watchdog timer has the following functions:

- Selection of the operation mode after reset, by using the option bytes
  - Enabling/disabling of WDTB, starting/stopping of the counter after reset, setting of the counter overflow time, and enabling/disabling of the VAC function can be selected. WDTB startup options to be set by the option bytes.
- WDTB trigger function
  - Writing an activation code to the WDTB trigger register starts WDTB and restarts the counter. Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTB trigger register, which causes the counter to be restarted.
- Interrupt request signal
  - An interrupt request signal can be generated when the WDTB counter reaches a specific number or 75% of the overflow interval time (these two options can be set by WDTBnMD.WDTBnWMS/WDTBAMD.WDTBAWMS and WDTBnWIS/WDTBAWIS).
  - This function can be enabled/disabled by the setting of WDTBnMD.WDTBnWIE/ WDTBAMD.WDTBAWIE.
- Window function
  - The period during which writing to the WDTB trigger register is valid (window open period) can be set. Writing to the WDTB trigger register during a period other than the window open period causes an error.
- WDTB error detection function
  - When an error is detected, a non-maskable interrupt request or an internal reset is generated.

31.2.2 Block Diagram

Figure 31.1 and **Figure 31.2** show the main components of the WDTB.

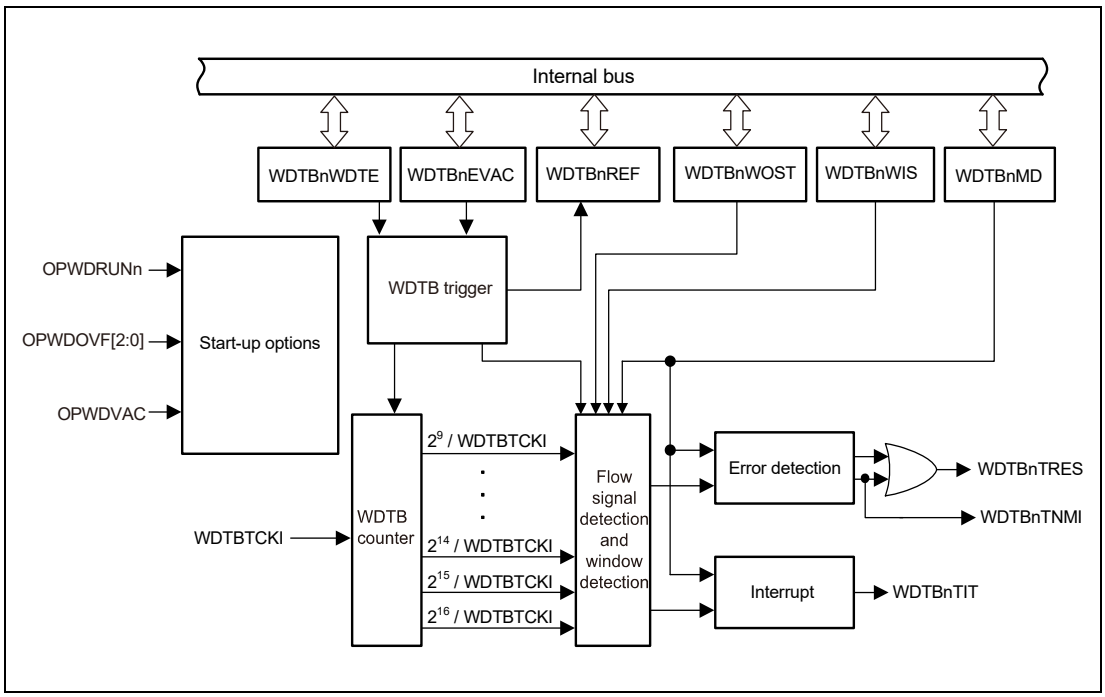


Figure 31.1 Block Diagram of WDTBn

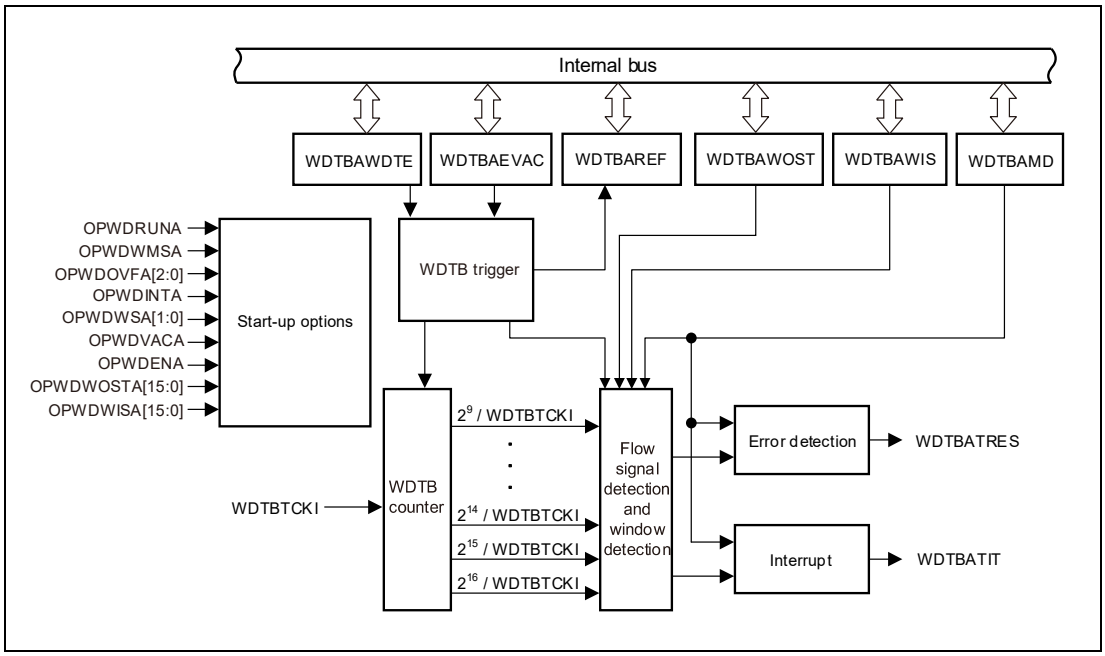


Figure 31.2 Block Diagram of WDTBA

31.3 Registers

The WDTB is controlled and operated by the following registers.

31.3.1 List of Registers

The following shows the list of registers and the memory address of WDTB.

For the base addresses, see **Table 31.3**.

The actual address can be obtained by adding the offset shown in the table to this base address.

Table 31.10 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
WDTB	WDTB Enable register	WDTBnWDTE	<WDTBn_base> + 0000_H	8	*1	—
	WDTB enable register for VAC	WDTBnEVAC	<WDTBn_base> + 0004_H	8	*1	—
	WDTB reference value register	WDTBnREF	<WDTBn_base> + 0008_H	8	*1	—
	WDTB Mode register	WDTBnMD	<WDTBn_base> + $000C_H$	8	*1	—
	WDTB Window Open Start register	WDTBnWOST	<WDTBn_base> + 0010_H	16	*1	—
	WDTB Interrupt Output Timing Setting register	WDTBnWIS	<WDTBn_base> + 0014_H	16	*1	—
	WDTB Enable register	WDTBAWDTE	<WDTBA_base> + 0000_H	8	PBG20#11	—
	WDTB enable register for VAC	WDTBAEVAC	<WDTBA_base> + 0004_H	8	PBG20#11	—
	WDTB reference value register	WDTBAREF	<WDTBA_base> + 0008_H	8	PBG20#11	—
	WDTB Mode register	WDTBAMD	<WDTBA_base> + $000C_H$	8	PBG20#11	—
	WDTB Window Open Start register	WDTBAWOST	<WDTBA_base> + 0010_H	16	PBG20#11	—
	WDTB Interrupt Output Timing Setting register	WDTBAWIS	<WDTBA_base> + 0014_H	16	PBG20#11	—
	WDTB Option Byte Monitor Register 0	WDTBAOPBMON0	FF9A 5208_H	32	PBG20#11	—
	WDTB Option Byte Monitor Register 1	WDTBAOPBMON1	FF9A $520C_H$	32	PBG20#11	—

Note 1. n = 0 PBG50#2
n = 1 PBG50#3
n = 2 PBG50#4
n = 3 PBG50#5

31.3.2 Detail of Registers

31.3.2.1 WDTBnWDTE/WDTBAWDTE — WDTB Enable Register

This register is the WDTB trigger register when the VAC function is not used.

Writing AC_H to this register generates a WDTB trigger and starts or restarts the WDTB counter.

Refer to **Section 31.4.2, WDTB Trigger** for details.

The value that can be written in this register is only AC_H.

Access: This register can be read or written in 8-bit units.

Address: <WDTBn_base> + 0000_H
<WDTBA_base> + 0000_H

Value after reset: The initial value depends on the start-up options (OPWDENA, OPWDRUNn/OPWDRUNA and OPWDVAC/OPWDVACA). See **Table 31.12, Values of WDTBnRUN[7]/WDTBARUN[7] after Reset**.

Bit	7	6	5	4	3	2	1	0
	WDTBnRUN[7:0]/WDTBARUN[7:0]							
Value after reset	0/1*1	0	1	0	1	1	0	0
R/W	R/W	R	R	R	R	R	R	R

Note 1. WDTBnRUN[7]/WDTBARUN[7] depends on start-up options.

Table 31.11 WDTBnWDTE/WDTBAWDTE Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTBnRUN[7:0]/ WDTBARUN[7:0]	Writing the fixed activation code (AC_H) generates the WDTB trigger and starts/ restarts the WDTB counter. Writing a value other than AC_H generates an error. The WDTB cannot be stopped once it is started.

The value after WDTBnRUN[7] /WDTBARUN[7] bit different according to the start-up option after reset is indicated in the following table.

Table 31.12 Values of WDTBnRUN[7]/WDTBARUN[7] after Reset

Start-Up Options			Value after Reset
OPWDVAC		**OPWDRUNn**	**WDTBnRUN[7]**
1		×	0
0		0	0
		1	1
OPWDENA	**OPWDVACA**	**OPWDRUNA**	**WDTBARUN[7]**
0	×	×	0
1	1	×	0
	0	0	0
		1	1

- If OPWDVAC/OPWDVACA = 0 (VAC is disabled), this register is enabled.
- When “1” is written on WDTBnRUN[7]/WDTBARUN[7], the watchdog timer starts counting.
- To clear (= trigger) the counter, AC_H must be written to this register.
- Writing wrong value or writing while window is closed leads to error detection.
- IF OPWDVAC/OPWDVACA = 1 (VAC is enabled), this register is disabled. Return Value is 2C (WDTBnRUN[7]/WDTBARUN[7] = 0).

NOTE

2 cycles are necessary with PCLK before initial value is reflected on the WDTBnRUN[7]/ WDTBARUN[7] bit after reset release.

31.3.2.2 WDTBnEVAC/WDTBAEVAC — WDTB Enable VAC Register

This register is a trigger register when VAC function is used.

Writing a correct activation code to this register generates a WDTB trigger and starts or restarts the WDTB counter.

Refer to **Section 31.4.2, WDTB Trigger** for details.

Access: This register can be read or written in 8-bit units.

Address: <WDTBn_base> + 0004_H
<WDTBA_base> + 0004_H

Value after reset: The initial value depends on the start-up options (OPWDENA, OPWDRUNn/OPWDRUNA and OPWDVAC/OPWDVACA). See **Table 31.14, Values of WDTBnEVAC[7]/WDTBAEVAC[7] after Reset**.

Bit	7	6	5	4	3	2	1	0
	WDTBnEVAC[7:0]/WDTBAEVAC[7:0]							
Value after reset	0/1*1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. WDTBnEVAC[7]/WDTBAEVAC[7] depends on start-up options.

Table 31.13 WDTBnEVAC/WDTBAEVAC Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTBnEVAC[7:0]/ WDTBAEVAC[7:0]	Writing a variable activation code generates the WDTB trigger and starts/ restarts the WDTB counter. Writing an incorrect activation code generates an error. The WDTB cannot be stopped once it is started.

The value after WDTBnEVAC[7]/WDTBAEVAC[7] bit different according to the start-up option after reset is indicated in the following table.

Table 31.14 Values of WDTBnEVAC[7]/WDTBAEVAC[7] after Reset

Start-Up Options			Value after Reset
OPWDVAC		OPWDRUNn	WDTBnEVAC[7]
0		×	0
1		0	0
		1	1
OPWDENA	**OPWDVACA**	**OPWDRUNA**	**WDTBAEVAC[7]**
0	×	×	0
1	0	×	0
	1	0	0
		1	1

- If OPWDVAC/OPWDVACA = 1 (VAC is enabled), this register is enabled.
- When “1” is written on WDTBnEVAC[7]/WDTBAEVAC[7], the watchdog timer starts counting.
- Writing the correct value on this register can clear the counter.
- Writing wrong value or writing while window is closed leads to error detection.
- When this register is enabled, the reading value just after reset is Initial value. After write to this register, the last time written value is returned.
- If OPWDVAC/OPWDVACA = 0 (VAC is disabled), this register is disabled. Return Value is $2C_H$.

NOTE

2 cycles are necessary with PCLK before initial value is reflected on the WDTBnEVAC[7]/ WDTBAEVAC[7] bit after reset release.

31.3.2.3 WDTBnREF/WDTBAREF — WDTB Reference Value Register

This register is Reference Value register for VAC function. It is automatically after every trigger operation. See **Section 31.4.6.2, Calculating an Activation Code when the VAC Function is Used**.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <WDTBn_base> + 0008_H
<WDTBA_base> + 0008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WDTBnREF[7:0]/WDTBAREF[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.15 WDTBnREF/WDTBAREF Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTBnREF[7:0]/ WDTBAREF[7:0]	Reference value for activation code calculation for the VAC function.

- If OPWDVAC/OPWDVACA = 1 (VAC is enabled), this register is enabled.
- The reference value of the start code operation to count trigger is indicated.
- Whenever the start code is written in the WDTBnEVAC/WDTBAEVAC register, the reference value calculated based on the expected value is returned.
- If OPWDVAC/OPWDVACA = 0 (VAC is disabled), this register is disabled. Return Value is reset value (00_H).

31.3.2.4 WDTBnMD/WDTBAMD — WDTB Mode Register

This register specifies the overflow interval time, enabling or disabling of the interrupt, window open function mode selection, and the window open period.

Access: This register can be read or written in 8-bit units.

Address: <WDTBn_base> + $000C_H$
<WDTBA_base> + $000C_H$

Value after reset: WDTBnMD: The initial value of WDTBnMD[6:4] bits depends on the start-up options (OPWDOVF).
WDTBAMD: The initial value depends on the start-up options (OPWDWMSA, OPWDOVFA, OPWDINTA and OPWDWSA).

Bit	7	6	5	4	3	2	1	0
	WDTBnWMS/ WDTBAWMS	WDTBnOVF[2:0]/WDTBAOVF[2:0]			WDTBnWIE/ WDTBAWIE	WDTBnERM/ —*5	WDTBnWS[1:0]/ WDTBAWS[1:0]	
Value after reset	0*1	1*2	1*2	1*2	1*3	1	1*4	1*4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. WDTBAWMS depends on OPWDWMSA.
Note 2. WDTBnOVF[2:0]/WDTBAOVF[2:0] depends on OPWDOVF[2:0]/OPWDOVFA[2:0].
Note 3. WDTBAWIE depends on OPWDINTA.
Note 4. WDTBAWS[1:0] depends on OPWDWSA[1:0].
Note 5. NMI function only support WDTBn.

Table 31.16 WDTBnMD/WDTBAMD Register Contents (1/2)

Bit Position	Bit Name	Function
7	WDTBnWMS/ WDTBAWMS	Window Open function mode selection 0: Window size of Window Open function is set by WDTBnWS[1:0]/ WDTBAWS[1:0]. WDTBnTIT/WDTBATIT outputs when the counter reaches 75% of the overflow setting defined by WDTBnMD.WDTBnOVF[2:0]/ WDTBAMD.WDTBAOVF[2:0]. 1: Window size of Window Open function is set by WDTBnWOST/ WDTBAWOST register. WDTBnTIT/WDTBATIT outputs when WDTBnWIS/ WDTBAWIS matches WDTB counter
6 to 4	WDTBnOVF[2:0]/ WDTBAOVF[2:0]	These bits select the overflow interval time.
3	WDTBnWIE/ WDTBAWIE	Enables/disables the interrupt request WDTBnTIT/WDTBATIT 0: WDTBnTIT/WDTBATIT is disabled. 1: WDTBnTIT/WDTBATIT is enabled.
2	WDTBnERM*1/ Reserved	WDTBn Error mode setting 0: NMI mode 1: Reset mode (Default) WDTBAMD[2]: When read, the value after reset is return. When writing, write the value after reset.

WDTBnOVF2/ WDTBAOVF2	WDTBnOVF1/ WDTBAOVF1	WDTBnOVF0/ WDTBAOVF0	Overflow Interval Time
0	0	0	2^9 / WDTBTCKI
0	0	1	2^{10} / WDTBTCKI
0	1	0	2^{11} / WDTBTCKI
0	1	1	2^{12} / WDTBTCKI
1	0	0	2^{13} / WDTBTCKI
1	0	1	2^{14} / WDTBTCKI
1	1	0	2^{15} / WDTBTCKI
1	1	1	2^{16} / WDTBTCKI

Table 31.16 WDTBnMD/WDTBAMD Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	WDTBnWS[1:0]/ WDTBAWS[1:0]	These bits select the period over which the window is open. If WDTBnWMS/WDTBAWMS = 1, this register is disabled.

WDTBnWS1/ WDTBAWS1	WDTBnWS0/ WDTBAWS0	Window Open Period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

Note 1. NMI function only support WDTBn

WDTBnMD/WDTBAMD register update timing

- Default start mode: WDTBnMD/WDTBAMD register can not be updated after any resets.
- Software trigger start mode: WDTBnMD/WDTBAMD register can be updated only once before first write to trigger register.

The WDTBnWS[1:0]/WDTBAWS[1:0] setting applies after the first trigger.

NOTE

3 cycles are necessary with PCLK before initial value is reflected on each bit after reset release.

31.3.2.5 WDTBnWOST/WDTBAWOST — WDTB Window Open Start Register

This register is Window Open Start register for setting the start timing of the window open.

Refer to **Section 31.4.5, Window Function** for details.

Access: This register can be read or written in 16-bit units.

Address: <WDTBn_base> + 0010_H
<WDTBA_base> + 0010_H

Value after reset: WDTBnWOST: 0010_H
WDTBAWOST: The initial value depends on the start-up options (OPWDWOSTA).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBnWOST[15:0]/WDTBAWOST[15:0]															
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	1*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. WDTBAWOST[15:0] depends on OPWDWOSTA[15:0].

Table 31.17 WDTBnWOST/WDTBAWOST Register Contents

Bit Position	Bit Name	Function
15 to 0	WDTBnWOST [15:0] / WDTBAWOST [15:0]	If WDTBnMD.WDTBnWMS/WDTBAMD.WDTBAWMS = 1, this register is enabled. When WDTB counter value ≥ WDTBnWOST/WDTBAWOST, WDT window open.

NOTE

WDTBnWOST/WDTBAWOST register update timing

- Default start mode: WDTBnWOST/WDTBAWOST register can not be updated after any resets.
- Software trigger start mode: WDTBnWOST/WDTBAWOST register can be updated only once before first WDTB trigger.

WDTBnWOST/WDTBAWOST register must be set more than 0010_H (When WDTBnMD.WDTBnWMS/WDTBAMD.WDTBAWMS = 1).

WDTBnWOST/WDTBAWOST register setting applies after the first trigger.

31.3.2.6 WDTBnWIS/WDTBAWIS — WDTB Interrupt Output Timing Setting Register

This register configures the timing of the WDTB interrupt (WDTBnTIT/WDTBATIT) generation.

Access: This register can be read or written in 16-bit units.

Address: <WDTBn_base> + 0014_H
<WDTBA_base> + 0014_H

Value after reset: WDTBnWIS: 0000_H
WDTBAWIS: The initial value depends on the start-up options (OPWDWISA).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDTBnWIS[15:0]/WDTBAWIS[15:0]															
Value after reset	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1	0*1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. WDTBAWIS[15:0] depends on OPWDWISA[15:0].

Table 31.18 WDTBnWIS/WDTBAWIS Register Contents

Bit Position	Bit Name	Function
15 to 0	WDTBnWIS [15:0] / WDTBAWIS [15:0]	If WDTBnMD.WDTBnWMS/WDTBAMD.WDTBAWMS = 1 and WDTBnMD.WDTBnWIE/WDTBAMD.WDTBAWIE = 1, this register is enabled. When WDTB counter value = WDTBnWIS[15:0]/WDTBAWIS[15:0], interrupt request WDTBnTIT/WDTBATIT is generated.

NOTE

WDTBnWIS/WDTBAWIS register update timing

- Default start mode: WDTBnWIS/WDTBAWIS register can not be updated after any resets.
- Software trigger start mode: WDTBnWIS/WDTBAWIS register can be updated only once before first WDTB trigger.

CAUTIONS

The register value after reset is 0000_H.
If this register is enabled, it must be set more than 0001_H.

31.3.2.7 WDTBAOPBMON0 — WDTB Option Byte Monitor Register 0

This register is the option byte monitor of the WDTBA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 5208$_H$

Value after reset: The initial value depends on the start-up options (OPWDOVFA, OPWDWSA, OPWDENA, OPWDVACA, OPWDINTA, OPWDWMSA and OPWDRUNA).
The initial state value of OPWDVACA bit, OPWDWMSA bit and OPWDRUNA bit are set to 0_B; the values of OPWDOVFA[2:0] bits, OPWDWSA[1:0] bits, OPWDENA bit and OPWDINTA bit are set to 1_B.
Therefore, the register value after reset of WDTBAOPBMON0 at shipment is 0000 07D4$_H$.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OPWDOVFA[2:0]			OPWDWSA[1:0]		—	OPWD ENA	OPWD VACA	OPWD INTA	OPWD WMSA	OPWD RUNA
Value after reset	0	0	0	0	0	1	1	1	1	1	0	1	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.19 WDTBAOPBMON0 Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10 to 8	OPWDOVFA [2:0]	For monitor Select the overflow interval time option byte.
7, 6	OPWDWSA[1:0]	For monitor WDTBA window size setting option byte.
5	Reserved	When read, the value after reset is returned.
4	OPWDENA	For monitor WDTBA macro enable option byte.
3	OPWDVACA	For monitor WDTBA VAC enable option byte.
2	OPWDINTA	For monitor WDTBATIT enabled option byte.
1	OPWDWMSA	For monitor WDTBA Window mode select option byte.
0	OPWDRUNA	For monitor WDTBA Automatic start after reset option byte.

For details of the option bytes, see **Section 51, Flash Memory**.

31.3.2.8 WDTBAOPBMON1 — WDTB Option Byte Monitor Register 1

This register is the option byte monitor of the WDTBA.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 520C$_H$

Value after reset: The initial value depends on the start-up options (OPWDWOSTA and OPWDWISA).
The initial state values of OPWDWOSTA[15:0] bits at shipment is 0010$_H$.
The initial state values of OPWDWISA[15:0] bits at shipment is 0000$_H$
Therefore the register value after reset of WDTBAOPBMON1 at shipment is = 0010 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPWDWOSTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPWDWISA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.20 WDTBAOPBMON1 Register Contents

Bit Position	Bit Name	Function
31 to 16	OPWDWOSTA [15:0]	For monitor WDTBA setting window open start timing option byte.
15 to 0	OPWDWISA [15:0]	For monitor WDTBATIT output timing option byte.

For details of the option bytes, see **Section 51, Flash Memory**.

31.4 Operation

31.4.1 WDTB after Reset Release

31.4.1.1 Start Modes

The WDTB provided two modes for the counter start after release from the reset state:

- Software trigger start mode
  The counter value remains 0000_H after reset release.
  The counter is started with the first WDTB trigger.
- Default start mode
  The counter starts automatically after release from the reset state.
  However, default start mode is disabled in serial programming mode even if OPWDRUNn/OPWDRUNA is set to 1_B for the start-up options.

31.4.1.2 Start Mode Selection

The start mode can be selected by the start-up options.

The following **Table 31.21** shows the selection of the start mode.

Table 31.21 Start Mode Selection

Start-up Options OPWDRUNn/OPWDRUNA	Start Mode	WDTB Settings
0	Software trigger	Can be changed only once before the first WDTB trigger
1	Default	Can not be changed after any resets

31.4.1.3 WDTB Settings after Reset Release

The following table shows the setting of WDTB during the reset release and the first trigger generation.

Table 31.22 Setting of WDTB after the Reset Release (1/2)

Bit Name	Function	Setting after WDTBn is Reset	Setting after WDTBA is Reset
—	Start mode	Specified by OPWDRUNn	Specified by OPWDRUNA
WDTBnOVF[2:0]/ WDTBAOVF[2:0]	Overflow interval time	Specified by OPWDOVF[2:0]	Specified by OPWDOVFA[2:0]
WDTBnWIE/ WDTBAWIE	Interrupt mode	1_B: Interrupt enabled	Specified by OPWDINTA
WDTBnWS[1:0]/ WDTBAWS[1:0]	Window open period	11_B: 100%	Specified by OPWDWSA[1:0]
WDTBnWMS/ WDTBAWMS	Window open function mode selection	0_B: Window size of Window Open function is set by WDTBnMD.WDTBnWS[1:0] and interrupt is generated when the counter reaches 75% of the overflow setting.	Specified by OPWDWMSA[1:0]
WDTBnWOST/ WDTBAWOST	Counter value of window open start	0010_H	Specified by OPWDWOSTA[15:0]
WDTBnWIS/ WDTBAWIS	Counter value of WDTB interrupt output timing	0000_H*1	Specified by OPWDWISA[15:0]*1

Table 31.22 Setting of WDTB after the Reset Release (2/2)

Bit Name	Function	Setting after WDTBn is Reset	Setting after WDTBA is Reset
—	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing	Specified by OPWDVAC	Specified by OPWDVACA

Note 1. If WDTBnWIS/WDTBAWIS is used, it must be set more than 0001_H.

For details of the option bytes, see **Section 51, Flash Memory**.

31.4.1.4 Default Start Mode Timing

The following **Figure 31.3** shows the timing of the default start mode.

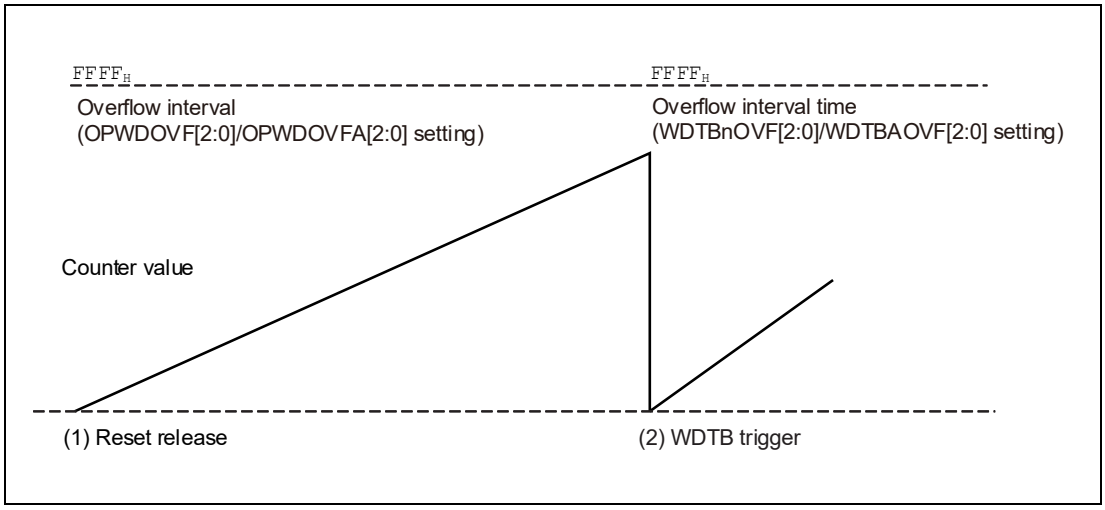


Figure 31.3 Timing Diagram of WDTB Start in Default Start Mode

The timing chart of **Figure 31.3** shows the following behaviors:

(1) In default start mode, the WDTB counter starts after release from the reset state. The overflow interval after release from the reset state is set by start-up options.

Example: overflow interval time after release from the reset state
$= 2^{16}$ / WDTBTCKI (OPWDOVF[2:0]/OPWDOVFA[2:0] = 111_B)

In default start mode, after reset release the WDTBnMD/WDTBAMD register can not be change because the WDTBnRUN/WDTBARUN = 1.

(2) Write to the WDTB trigger register before the WDTB counter overflows.
The WDTB counter restart with the WDTB trigger.

Example: overflow interval time after a WDTB trigger is generated
$= 2^{16}$ / WDTBTCKI (WDTBnOVF[2:0]/WDTBAOVF[2:0] = 111_B)

31.4.1.5 Software Trigger Start Mode Timing (Common to all WDTBn/WDTBA)

The following **Figure 31.4** shows the timing of the software trigger start mode and the change to the WDTB setting.

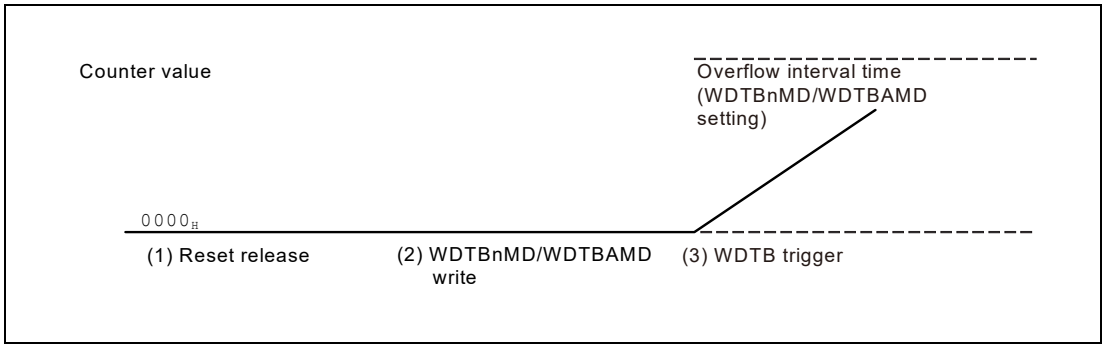


Figure 31.4 Timing Diagram of WDTB Start in Software Trigger Start Mode

The timing diagram in **Figure 31.4** shows the following:

(1) After release from the reset state, the counter remains 0000_H until the first WDTB trigger.
The overflow interval time is set by using the start-up options, but it does not have any effect.

(2) WDTBnMD/WDTBAMD is set before the first WDTB trigger. However, the settings are not applied immediately.

(3) The WDTB counter starts at the first WDTB trigger.
The overflow interval time and other settings specified in WDTBnMD/WDTBAMD are applied.

31.4.2 WDTB Trigger

The WDTB trigger is generated in WDTB enable register (WDTBnWDTE/WDTBAWDTE) and WDTB enable VAC register (WDTBnEVAC/WDTBAEVAC) by writing the specific value that is called a start code.

The WDTB trigger has the following functions:

- Starting the WDTB counter in software trigger start mode
- Restarting the WDTB counter
- Setting the WDTB mode specified by the WDTBnMD/WDTBAMD register in software trigger start mode (only for the first WDTB trigger after release from the reset state).

The WDTB trigger register, which generates a WDTB trigger, is specified by the start-up option OPWDVAC/OPWDVACA.

NOTE

6 PCLKs + 6 WDTBTCKI cycles are necessary until next counter clearing from counter clearing by WDTBnWDTE/WDTBAWDTE or WDTBnEVAC/WDTBAEVAC register writing.

Table 31.23 Trigger Register and Activation Code

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC/OPWDVACA = 0)	WDTBnWDTE/WDTBAWDTE	AC_H
Variable (OPWDVAC/OPWDVACA = 1)	WDTBnEVAC/WDTBAEVAC	For details, see **Section 31.4.6.2, Calculating an Activation Code when the VAC Function is Used**.

31.4.3 WDTB Error Detection

The WDTB detects an overflow of the WDTB counter and illegal operations as an error.

The conditions for error detection are:

- WDTB counter overflow
- Wrong activation code is written to the WDTB trigger register
- Writing to the trigger register at the time outside the window open period
  (For details, see **Section 31.4.5, Window Function**.)
- Illegal update of WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers

Following charts describes illegal update of WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers.

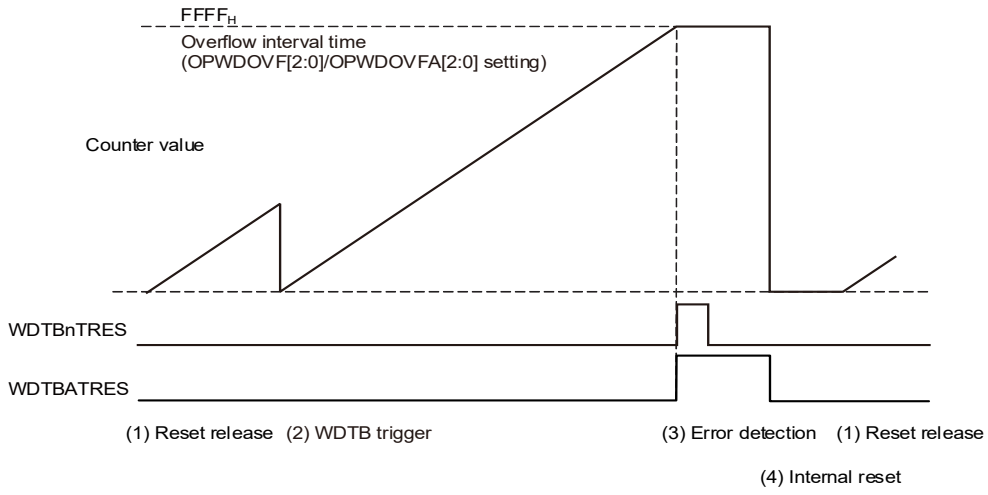
Table 31.24 Condition of WDTB Error Detection

Condition				
Start Mode	**WDTBnRUN/ WDTBARUN**	**Register Write First or Second**	**Write Value**	**Error Detection**
Default	0	This condition does not happen.		
	1	—	Same	Not Detect
		—	Another	Detect
Software Trigger	0	First	Same	Not Detect
			Another	Not Detect
		Second	Same	Not Detect
			Another	Detect
	1	—	Same	Not Detect
		—	Another	Detect

31.4.3.1 WDTB counter overflow

When an error is detected, the error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by the WDTBATRES signal.

The following **Figure 31.5** shows the generation of reset when the counter overflows when the default start mode has been selected.



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES.

Figure 31.5 Timing Diagram of WDTB Internal Reset Generation

The timing diagram of **Figure 31.5** shows the following:

(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

(2) The counter restarts with the WDTB trigger.

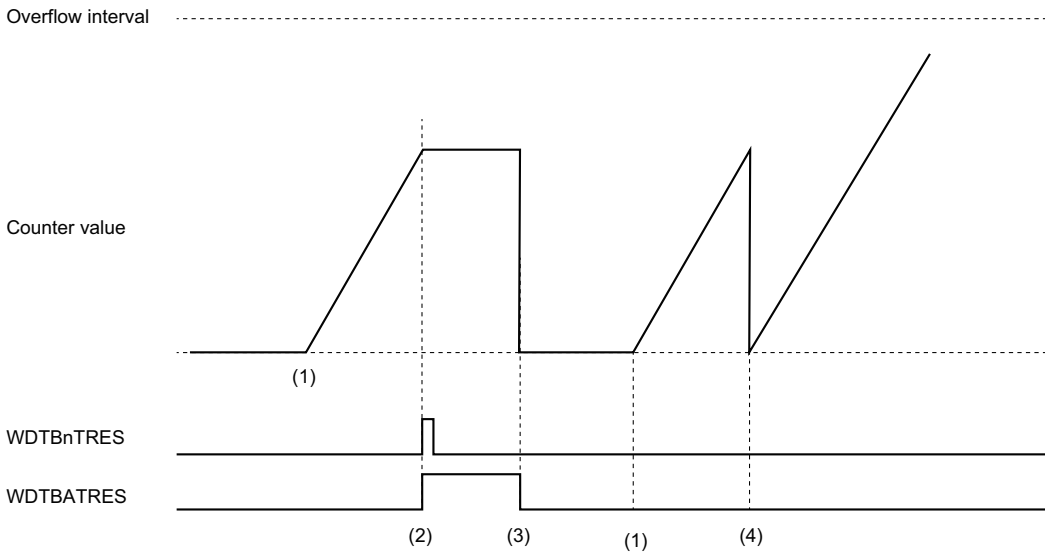
(3) The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(4) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

31.4.3.2 WDTB wrong activation code

When an error is detected , the error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by the WDTBATRES signal.

The following **Figure 31.6** shows the generation of reset when writing wrong activation code when the default start mode has been selected.



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES.

Figure 31.6 Timing Diagram of WDTB wrong activation code when the default start mode has been selected

The timing diagram of **Figure 31.6** shows the following:

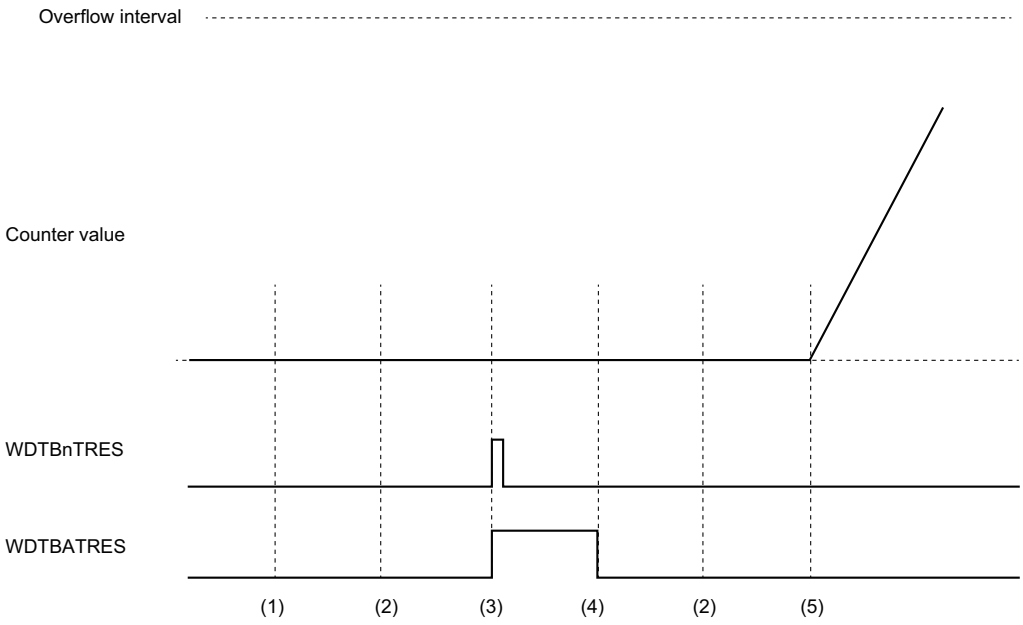
(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

(2) WDTB trigger with wrong activation code
The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(3) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

(4) The setting of WDTBnMD/WDTBAMD is applied with the WDTB trigger.

The following **Figure 31.7** shows the generation of reset when writing wrong activation code when the software start mode has been selected.



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES.

Figure 31.7 Timing Diagram of WDTB wrong activation code when the software start mode has been selected

The timing diagram of **Figure 31.7** shows the following:

(1) After release from the reset state, the counter remains 0000_H until the first WDTB trigger.

(2) WDTBnMD/WDTBAMD is set before the first WDTB trigger. However, the setting are not applied immediately.

(3) WDTB trigger with wrong activation code
The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(4) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

(5) The WDTB counter starts at the WDTB trigger.

The overflow interval time and other settings specified in WDTBnMD/WDTBAMD are applied.

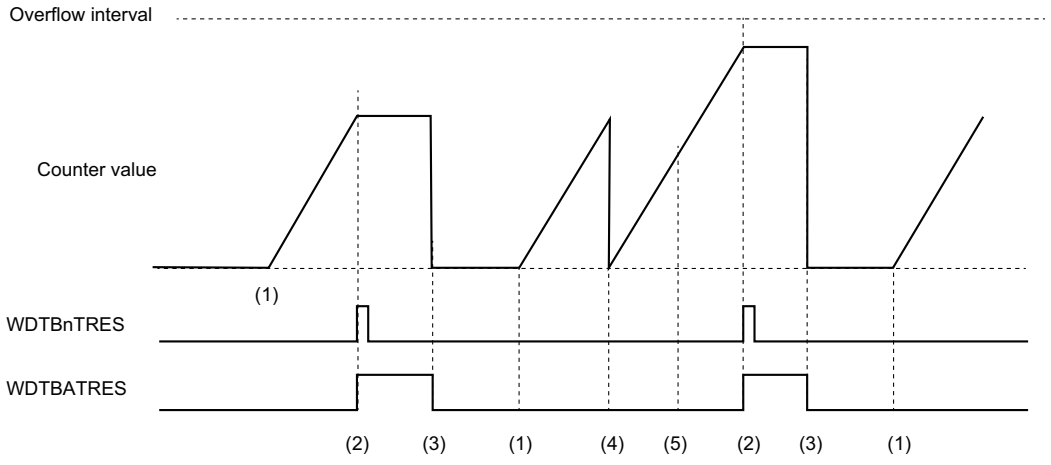
31.4.3.3 WDTB trigger counter outside the window open period

For details, see **Section 31.4.5, Window Function**.

31.4.3.4 WDTB illegal update register

When an error is detected , the error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by the WDTBATRES signal.

The following **Figure 31.8** shows the generation of reset when Illegal update register when the default start mode has been selected.



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES.

Figure 31.8 Timing Diagram of WDTB illegal register when the default start mode has been selected

The timing diagram of **Figure 31.8** shows the following:

(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

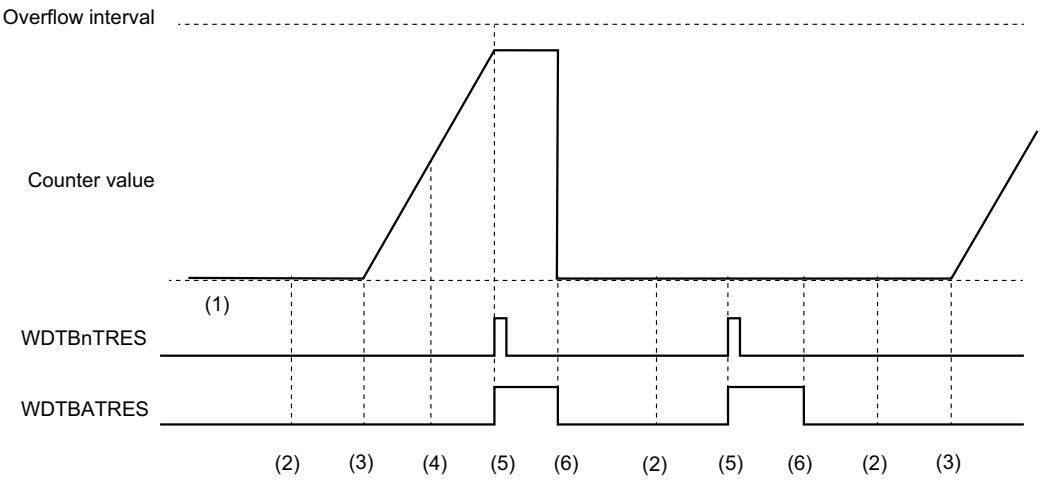
(2) Update WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers with value different with default value.
Ex: default value of WDTBnMD[7:0]/WDTBAMD[7:0] = 7F, write to WDTBnMD[7:0]/WDTBAMD[7:0] with value FF.
The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(3) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

(4) The setting of WDTBnMD/WDTBAMD is applied with the WDTB trigger.

(5) Update WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers with value same as default value. There is no thing happen.
Ex: default value of WDTBnMD[7:0]/WDTBAMD[7:0] = 7F, write to WDTBnMD[7:0]/WDTBAMD[7:0] with value 7F.

The following **Figure 31.9** shows the generation of reset when Illegal update register when the software start mode has been selected.



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES.

Figure 31.9 Timing Diagram of WDTB illegal update register when the software start mode has been selected

The timing diagram of **Figure 31.9** shows the following:

(1) After release from the reset state, the counter remains 0000_H until the first WDTB trigger.

(2) WDTBnMD/WDTBAMD is set before the first WDTB trigger. However, the setting are not applied immediately.

(3) The WDTB counter starts at the first WDTB trigger.
The overflow interval time and other settings specified in WDTBnMD/WDTBAMD are applied.

(4) Update WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers with value same as old value. There is no thing happen.
– Ex: old value of WDTBnMD[7:0]/WDTBAMD[7:0] = 7F, write to WDTBnMD[7:0]/WDTBAMD[7:0] with value 7F.

(5) Update WDTBnMD/WDTBAMD, WDTBnWOST/WDTBAWOST and WDTBnWIS/WDTBAWIS registers with value different with old value.
Ex: old value of WDTBnMD[7:0]/WDTBAMD[7:0] = 7F, write to WDTBnMD[7:0]/WDTBAMD[7:0] with value FF.
The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(6) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

31.4.4 WDTB Interrupt Output

An interrupt request signal WDTBnTIT/WDTBATIT is generated when:

- WDTB counter reaches 75% of overflow interval set time and WDTBnWMS/WDTBAWMS = 0.
- WDTB counter = WDTBnWIS/WDTBAWIS and WDTBnWMS/WDTBAWMS = 1.

This function can be enabled/disabled by the WDTBnMD.WDTBnWIE/WDTBAMD.WDTBAWIE register.

The following **Figure 31.10** and **Figure 31.11** show the generation of the interrupt request under the following condition:

- The default start mode is selected.
- Interrupt request is enabled after the first WDTB trigger is generated.
- WDTB overflow time of interval: 2^{16} / WDTBTCKI

31.4.4.1 For case WDTBnWMS/WDTBAWMS = 0:

When the WDTB counter reaches 75% of overflow interval set time and WDTBnWMS/WDTBAWMS = 0, interrupt request WDTBnTIT/WDTBATIT is generated.

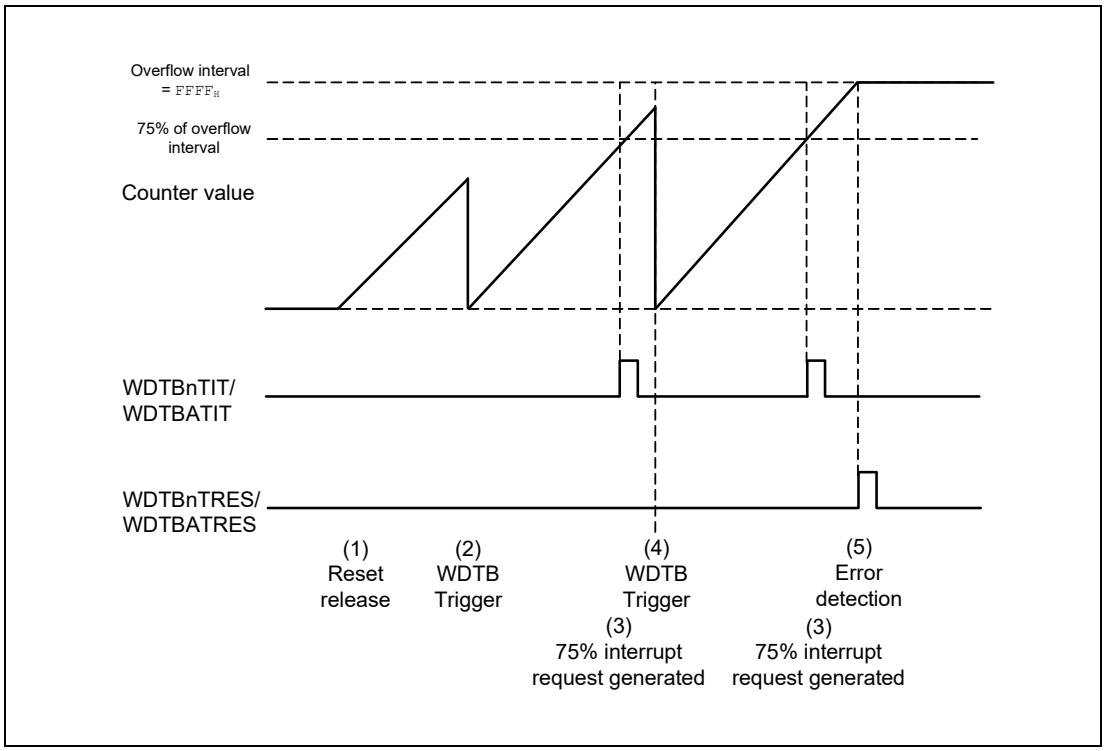


Figure 31.10 Timing Diagram of WDTB Interrupt Request Signals (WDTBnWMS/ WDTBAWMS = 0)

(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

(2) The counter restarts with the WDTB trigger.

(3) When the WDTB counter reaches 75% of overflow interval set time, interrupt request WDTBnTIT/WDTBATIT is generated.

(4) The count restarts with the WDTB trigger.

(5) The error is detected when the counter overflows.
The error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

31.4.4.2 For case WDTBnWMS/WDTBAWMS = 1:

When the WDTB counter = WDTBnWIS/WDTBAWIS and WDTBnWMS/WDTBAWMS = 1, an interrupt request WDTBnTIT/WDTBATIT is generated.

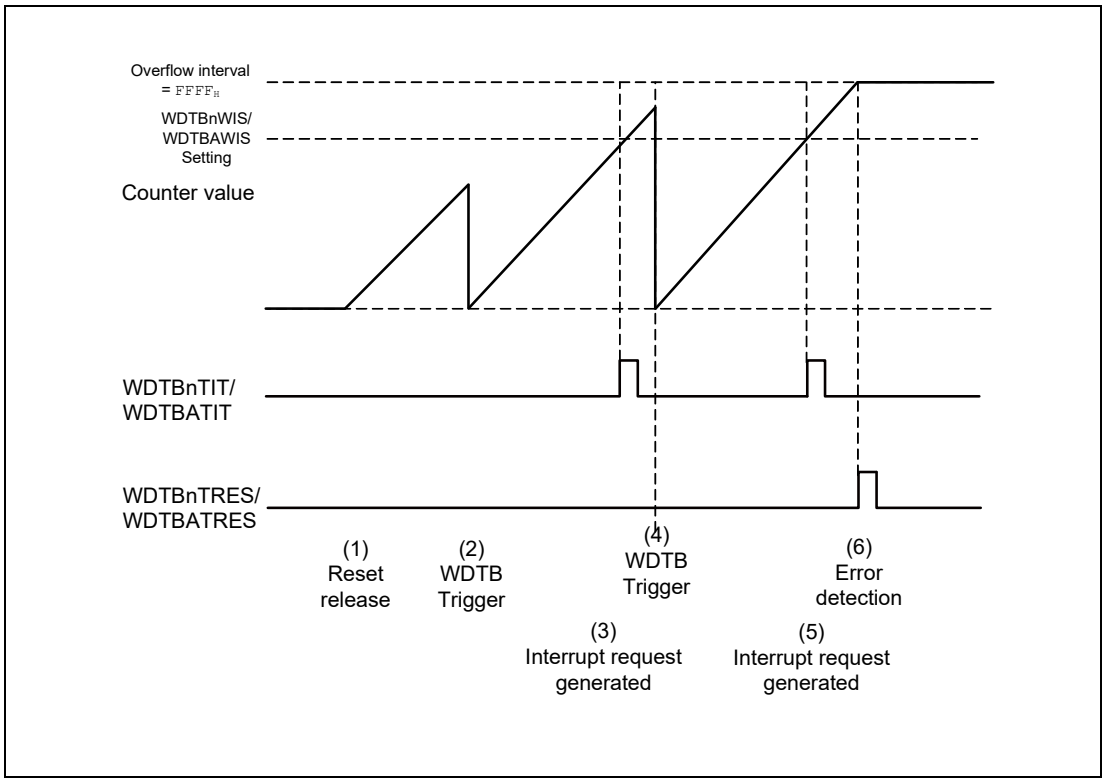


Figure 31.11 Timing Diagram of WDTB Interrupt Request Signals (WDTBnWMS/ WDTBAWMS = 1)

(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

(2) The counter restarts with the WDTB trigger.

(3) When the WDTB counter value = WDTBnWIS[15:0]/WDTBAWIS[15:0], an interrupt request WDTBnTIT/WDTBATIT is generated.

(4) The count restarts with the WDTB trigger.

(5) When the WDTB counter value = WDTBnWIS[15:0]/WDTBAWIS[15:0], an interrupt request WDTBnTIT/WDTBATIT is generated.

(6) The error is detected when the counter overflows.
The error notification is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until an internal reset is generated.

31.4.5 Window Function

The period when a WDTB trigger is valid (window open period) can be set. If the window open period is set to a value less than 100%, an error is occurred by the writing the WDTB trigger outside the window open period.

The window open period after reset release is 100%. The period is set to the value configured by the WDTBnMD.WDTBnWS[1:0] /WDTBAMD.WDTBAWS[1:0] or the WDTBnWOST/WDTBAWOST register setting after the first WDTB trigger is generated.

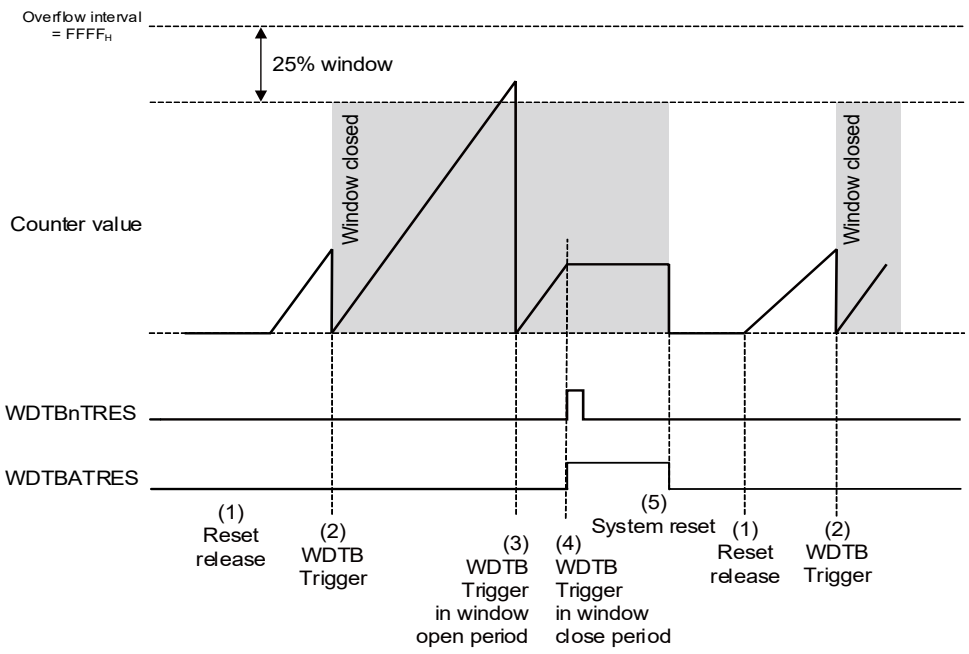
Figure 31.12 and **Figure 31.13** show the window function operation under the following condition.

- The default start mode is selected.
- 25% window opening period is effective after the first WDTB trigger (WDTBnWS[1:0]/ WDTBAWS[1:0] = 00_B).
- WDTB overflow interval: 2^{16} / WDTBTCKI

31.4.5.1 For case WDTBnWMS = 0:

The window open period is set to the value configured by the WDTBnMD.WDTBnWS[1:0]/ WDTBAMD.WDTBAWS[1:0] setting after the first WDTB trigger is generated.

This example setting WDTBnWS[1:0]/WDTBAWS[1:0] = 00_B has window open period is 25%, meaning WDTB trigger only allowed when counter value larger than 75% of overflow value (outside window closed period and inside window open period).



Note: In U2A-EVA, output width of WDTBATRES is same as WDTBnTRES

Figure 31.12 Timing Diagram of WDTB Window Function

(1) In default start mode, the WDTB counter starts after reset is released.
The overflow interval time period after reset is released is set by the start-up options.

(2) The WDTBnMD.WDTBnWS[1:0]/WDTBAMD.WDTBAWS[1:0] setting is applied at the WDTB trigger.

(3) The WDTB counter restarts at the WDTB trigger during the window open period.

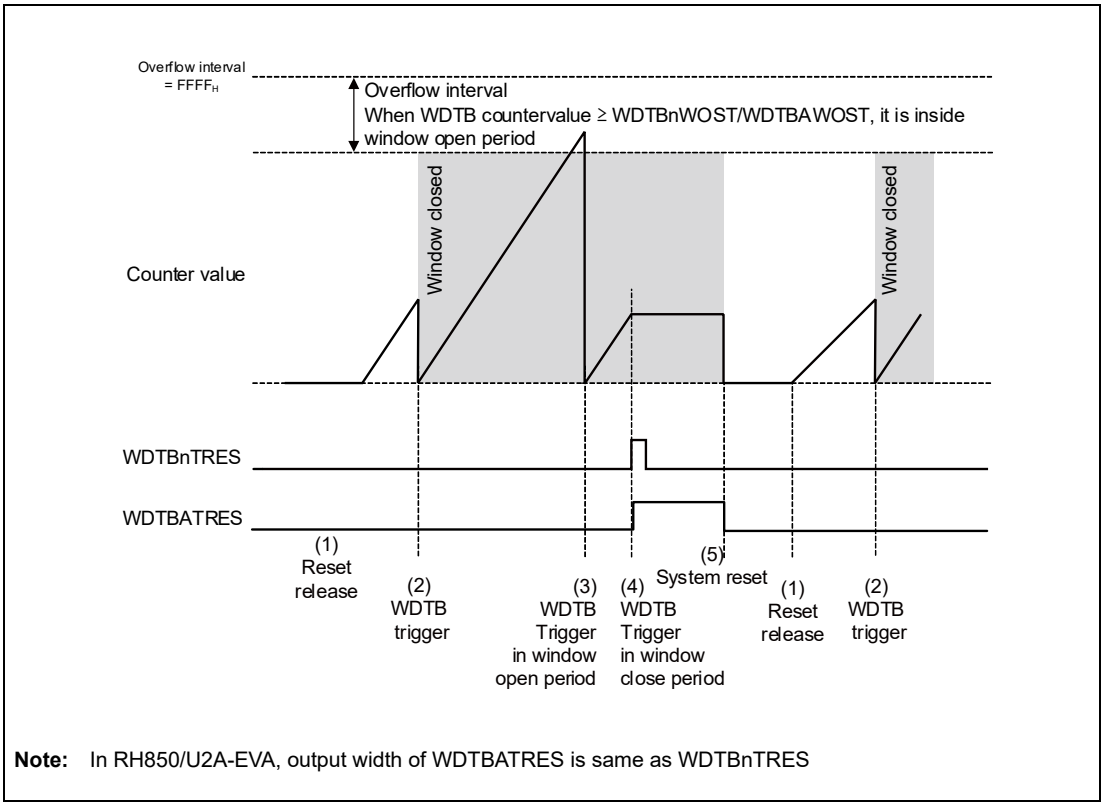
(4) An error is detected at the WDTB trigger during the window close period, the error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(5) If an internal reset occurs due to the ECM and other sources, the counter is cleared and stopped until release from the reset state.

31.4.5.2 For case WDTBnWMS/WDTBAWMS = 1:

The period is set to the value configured by WDTBnWOST[15:0]/WDTBAWOST[15:0] setting after the first WDTB trigger is generated.

In this case, WDTB trigger is only allowed when the counter value is equal or larger than the value of WDTBnWOST[15:0]/WDTBAWOST[15:0].



Note: In RH850/U2A-EVA, output width of WDTBATRES is same as WDTBnTRES

Figure 31.13 Timing Diagram of WDTB Window Function

(1) In default start mode, the WDTB counter starts after reset is released.
The overflow interval time after reset is released is set by the start-up options.

(2) The WDTBnWOST/WDTBAWOST register setting is applied at the WDTB trigger.

(3) The WDTB counter restarts at the WDTB trigger during the window open period.

(4) An error is detected at the WDTB trigger during the window close period, the error is notified to the ECM by the WDTBnTRES signal, to the PIC by the WDTBnTNMI signal, or to the Reset Controller by WDTBATRES signal. The counter value does not change until internal reset is generated.

(5) If an internal reset occurs due to the ECM and other sources, the counter is cleared and stopped until release from the reset state.

31.4.6 Varying Activation Code (VAC)

31.4.6.1 About VAC Function

If enabled by option byte, a fixed activation code (AC_H) is replaced by varying activation code.

This generates a sequence of 122 different numbers that have to be used to re-trigger the Watchdog timer. This calculation requires a computation in the CPU and therefore gives additional security against program and CPU malfunction.

If OPWDVAC/OPWDVACA = 1, VAC can be used. The WDTBnEVAC/WDTBAEVAC register in VAC module replaces WDTBnWDTE/WDTBAWDTE register's functionality. That means, the activation code must be written on WDTBnEVAC/WDTBAEVAC instead of WDTBnWDTE/ WDTBAWDTE. (In this case access to WDTBnWDTE /WDTBAWDTE is ignored.)

31.4.6.2 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (Expected WDTE) to be set in the WDTB trigger register (WDTBnEVAC/WDTBAEVAC) when the VAC function is used, by using the WDTB reference value register (WDTBnREF/WDTBAREF):

Expected WDTE = AC_H - WDTBnREF/WDTBAREF (previous)

Note that the value in the WDTBnREF/WDTBAREF register is updated every time a start-code is written to the trigger register WDTBnEVAC/WDTBAEVAC. Use the following expression to calculate the updated value of the WDTBnREF/WDTBAREF register.

WDTBnREF/WDTBAREF (following) = (rotate the value of Expected WDTE to the left by 1 bit)

Table 31.25 Expected Variable Activation code Development

No*1	WDTBnREF/WDTBAREF (Previous)		Expected WDTE (AC_H - WDTBnREF/WDTBAREF)		WDTBnREF/WDTBAREF (Following)	
0	0000 0000	00_H	1010 1100	AC_H	0101 1001	59_H
1	0101 1001	59_H	0101 0011	53_H	1010 0110	$A6_H$
2	1010 0110	$A6_H$	0000 0110	06_H	0000 1100	$0C_H$
...	...	...	...	...	...	...

Note 1. Number of triggers after reset

NOTE

Writing an incorrect activation code generates an error.

Even when the writing value to WDTBnEVAC/WDTBAEVAC is different from the expected value, the WDTBnREF/WDTBAREF value returns the calculation value from the expected value.

31.4.7 NMI or Reset Interrupt Output

When an error is detected, WDTBn can generate a non-maskable interrupt request or an internal reset. The output signals are configurated by WDTBnMD.WDTBnERM as follow:

- In NMI mode, WDTBnTNMI is output alongside with WDTBnTRES signal.
- In Reset mode, only WDTBnTRES is generated.

The output width of WDTBnTRES/WDTBnTNMI signal is equal to the pulse width of 1 WDTBTCKI.

For WDTBA, it will only run in Reset mode by default. When an error is detected, WDTBATRES signal will keep “high” until internal reset is generated.

The following **Figure 31.14** shows the generation of reset and NMI signals when the counter overflows in NMI mode and default start mode has been selected.

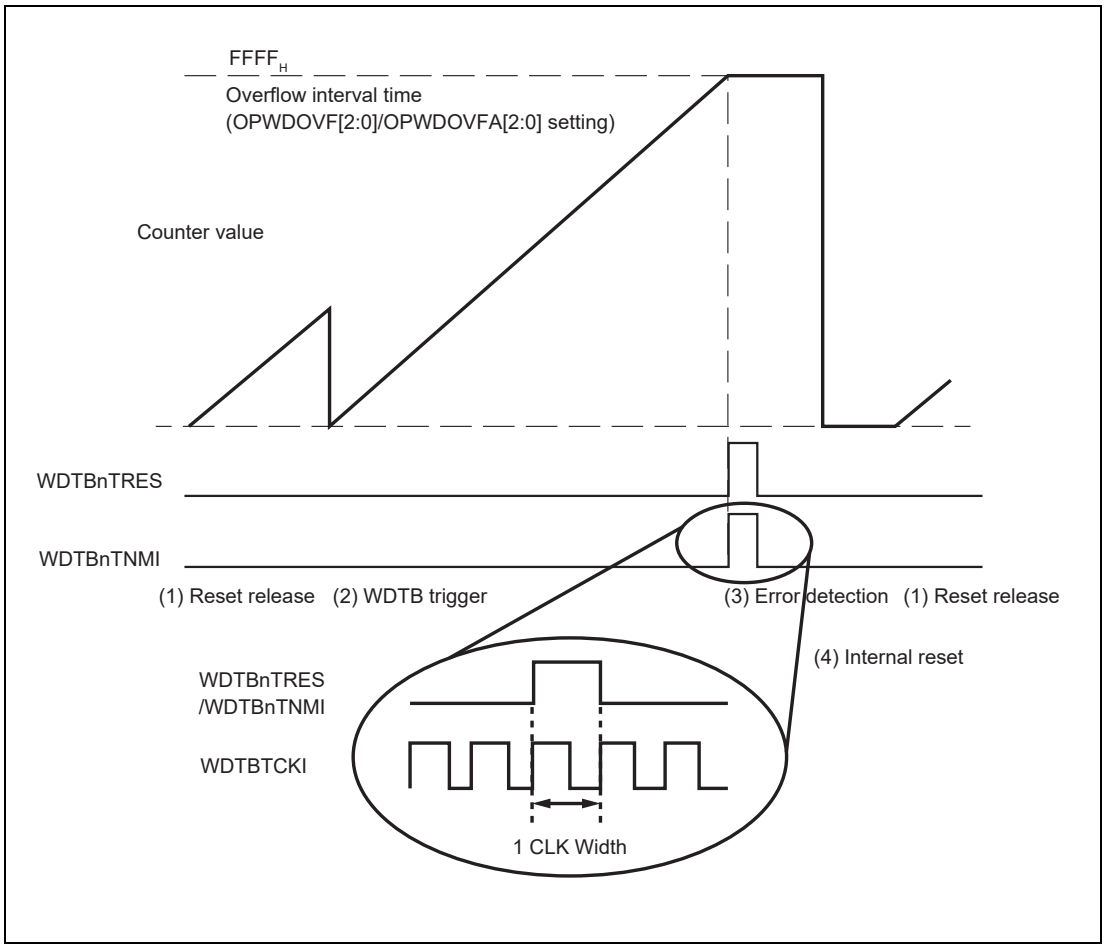


Figure 31.14 NMI or Reset Interrupt output timing of WDTBn

The timing diagram of **Figure 31.14** shows the following:

(1) After reset is released, the WDTB counter is started in the default start mode. The overflow interval time after released from the reset state is set by the start options.

(2) The counter restarts with the WDTB trigger.

(3) The error is detected when the counter overflows.
The error is notified to the ECM by the WDTBnTRES signal, and to PIC by the WDTBnTNMI signal. The counter value does not change until internal reset is generated.

(4) It stops until the counter is cleared when internal reset is generated by the factor such as ECM, and reset is released.

Section 32 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the OSTM.

32.1 Features OSTM for RH850/U2A-EVA

32.1.1 Number of Units

This microcontroller has the following number of OSTM units.

Each OSTM unit has one channel OSTM. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 32.1 Number of Units

Product	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144pins)
Number of Units	10 (n = 0 to 9)	10 (n = 0 to 9)	10 (n = 0 to 9)	10 (n = 0 to 9)	8 (n = 0 to 5, 8, 9)	8 (n = 0 to 5, 8, 9)	8 (n = 0 to 5, 8, 9)	8 (n = 0 to 5, 8, 9)	8 (n = 0 to 5, 8, 9)	8 (n = 0 to 5, 8, 9)
Name	OSTMn									

Table 32.2 Index

Index	Description
n	Throughout this section, the individual units of the OS timer are identified by the index "n" (n = 0 to 9), for example, OSTMnTO for the OS timer n output register.

32.1.2 Register Base Addresses

All OS timer register addresses are given as address offsets from the individual base addresses <OSTMn_base>.

The register base address of each OSTMn is listed in the following table.

Table 32.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<OSTM0_base>	FFBF 0000$_H$	P-Bus Group 5
<OSTM1_base>	FFBF 0100$_H$	P-Bus Group 5
<OSTM2_base>	FFBF 0200$_H$	P-Bus Group 5
<OSTM3_base>	FFBF 0300$_H$	P-Bus Group 5
<OSTM4_base>	FFBF 0400$_H$	P-Bus Group 5
<OSTM5_base>	FFBF 0500$_H$	P-Bus Group 5
<OSTM6_base>	FFBF 0600$_H$	P-Bus Group 5
<OSTM7_base>	FFBF 0700$_H$	P-Bus Group 5
<OSTM8_base>	FFBF 0800$_H$	P-Bus Group 5
<OSTM9_base>	FFBF 0900$_H$	P-Bus Group 5

32.1.3 Clock Supply

Clock supply by and to OSTMn is listed in the following table.

Table 32.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
OSTMn	PCLK	CLK_HSB
	Register access clock	CLK_HSB

32.1.4 Interrupt Requests and Error Notifications

OSTMn interrupt requests are listed in the following table.

Table 32.5 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTOSTM0TINT	OSTM0TINT	OSTM0 interrupt	199	Group0-157	Group3-82
INTOSTM1TINT	OSTM1TINT	OSTM1 interrupt	200	Group0-158	Group3-83
INTOSTM2TINT	OSTM2TINT	OSTM2 interrupt	201	Group0-159	Group3-84
INTOSTM3TINT	OSTM3TINT	OSTM3 interrupt	202	Group0-160	Group3-85
INTOSTM4TINT	OSTM4TINT	OSTM4 interrupt	203	Group0-161	Group3-86
INTOSTM5TINT	OSTM5TINT	OSTM5 interrupt	204	Group0-162	Group3-87
INTOSTM6TINT	OSTM6TINT	OSTM6 interrupt	205	Group0-163	Group3-88
INTOSTM7TINT	OSTM7TINT	OSTM7 interrupt	206	Group0-164	Group3-89
INTOSTM8TINT	OSTM8TINT	OSTM8 interrupt	207	Group0-165	Group3-90
INTOSTM9TINT	OSTM9TINT	OSTM9 interrupt	208	Group0-166	Group3-91

The error notifications of this module are listed in the following table.

Table 32.6 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
OS timer 1 interrupt	OSTM1 interrupt	56	—
OS timer 2 interrupt	OSTM2 interrupt	57	—
OS timer 3 interrupt	OSTM3 interrupt	58	—
OS timer 4 interrupt	OSTM4 interrupt	59	—
OS timer 5 interrupt	OSTM5 interrupt	60	—
OS timer 6 interrupt	OSTM6 interrupt	61	—
OS timer 7 interrupt	OSTM7 interrupt	62	—
OS timer 8 interrupt	OSTM8 interrupt	63	—
OS timer 9 interrupt	OSTM9 interrupt	64	—

NOTE

OSTM0 does not send error notification to ECM. However, its output can determine the output wave of ERROROUT_M and ERROROUT_C pin of ECM. For more detail, refer to **Section 45, Error Control Module (ECM)**.

32.1.5 Reset Sources

OSTM reset sources are listed in the following table.

Table 32.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
OSTMn	All registers	√	√	√	√	√	√	—

32.1.6 External Input/Output Signals

An OSTMn input/output signals are listed in the following table.

Table 32.8 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
OSTM8:			
OSTM8TTOUT	O	OSTM8 output	OSTM8O
OSTM9:			
OSTM9TTOUT	O	OSTM9 output	OSTM9O

32.2 Overview

32.2.1 Functional Overview

- OSTM has two operating modes.
  - Interval timer mode
  - Free-run compare mode
- All OS timers are running at the same frequency.
- Simultaneous starting of multiple units of the OS timer (timer synchronization of PIC) for OSTM8 and OSTM9.
- Interrupt and sDMA/DTS notification when counting by the counter is started, restarted, or ends
- Notifying the ECM of errors when an OSTMn (n = 1 to 9) interrupt occurs
- Generation of output waveform from the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins with the OSTM0TTOUT signal of OSTM0 when ECM is in dynamic mode
- OS timer is cleared by a reset source defined in **Table 32.7, Reset Sources**.
- Counter start and restart of OSTM can be performed by software.
- The counter value at counter operation start can be set by software.
- Interrupt request of OSTM will be notified to the corresponding CPU.
  - Interrupt request of OSTM0 assigned to all CPU
  - Interrupt request of OSTM1 assigned to all CPU
  - Interrupt request of OSTM2 assigned to all CPU
  - Interrupt request of OSTM3 assigned to all CPU
  - Interrupt request of OSTM4 assigned to all CPU
  - Interrupt request of OSTM5 assigned to all CPU
  - Interrupt request of OSTM6 assigned to all CPU
  - Interrupt request of OSTM7 assigned to all CPU
  - Interrupt request of OSTM8 assigned to all CPU
  - Interrupt request of OSTM9 assigned to all CPU

32.2.2 Block Diagram

The following block diagram shows the main components of the OSTMn.

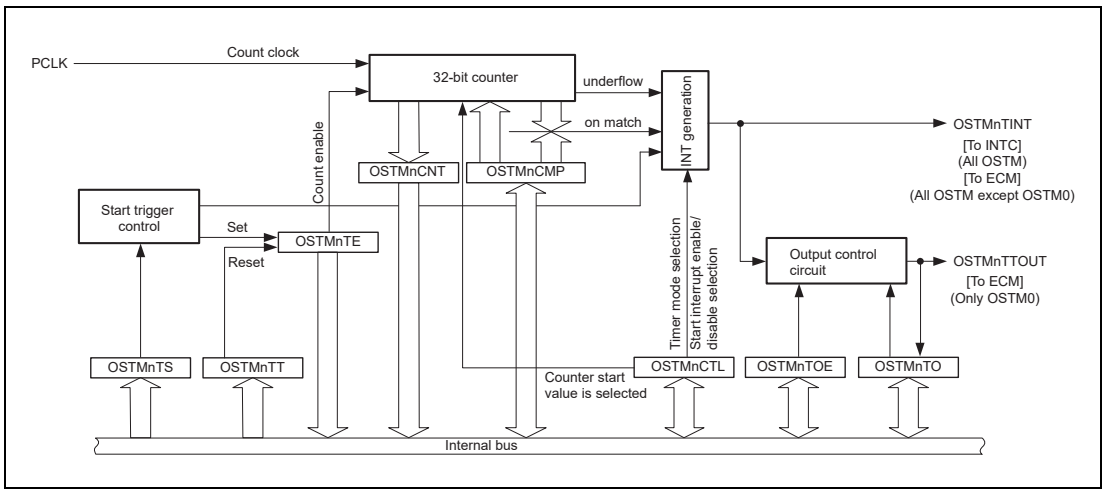


Figure 32.1 Block Diagram of the OSTMn (OSTM0 to OSTM7)

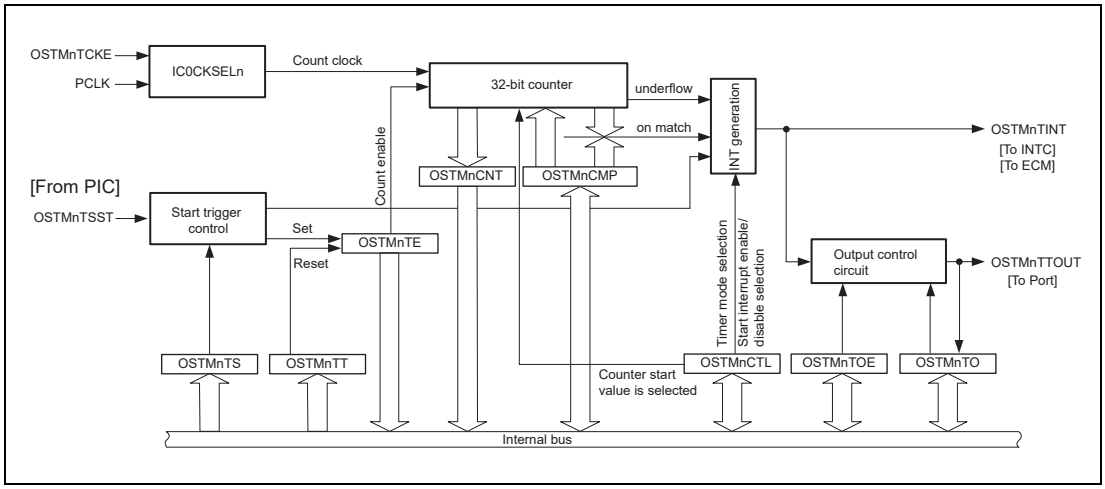


Figure 32.2 Block Diagram of the OSTMn (OSTM8, OSTM9)

32.2.3 Output Modes (only for OSTM0, OSTM8 and OSTM9)

The OSTMn has the following output modes. The mode is selected by the setting of the OSTMnTOE.OSTMnTOE bit.

- Software control mode (the OSTMnTOE.OSTMnTOE bit is 0): The value set in the OSTMnTO.OSTMnTO bit is output to OSTMnTTOUT.
- Timer-output toggling mode (the OSTMnTOE.OSTMnTOE bit is 1): The OSTMnTTOUT output is toggled each time an OSTMnTINT request is generated.

Both output modes are illustrated in the following figure.

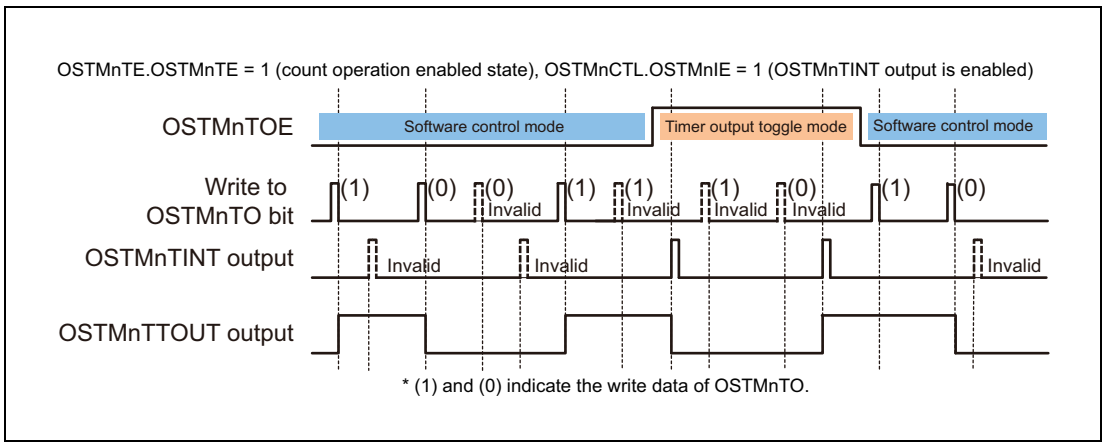


Figure 32.3 Timing Diagram of Output Modes

The above timing diagram shows the following operations.

- In software control mode, the level of the OSTMnTTOUT output changes in accord with the value set in the OSTMnTO.OSTMnTO bit.
- In timer-output toggling mode, the value of the OSTMnTO.OSTMnTO bit and level of the OSTMnTTOUT output are toggled each time an OSTMnTINT interrupt request is generated.

32.2.4 Interrupt Requests (OSTMnTINT)

An OSTMnTINT interrupt request is generated whenever the counter reaches 0000 0000_H (in interval timer mode) or matches the comparison value (in free-run compare mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

Since OSTMnTINT triggers toggling of the OSTMnTTOUT output in timer output toggling mode (OSTMnTOE.OSTMnTOE is 1), the setting of the OSTMnCTL.OSTMnMD0 bit also affects the output (OSTMnTTOUT).

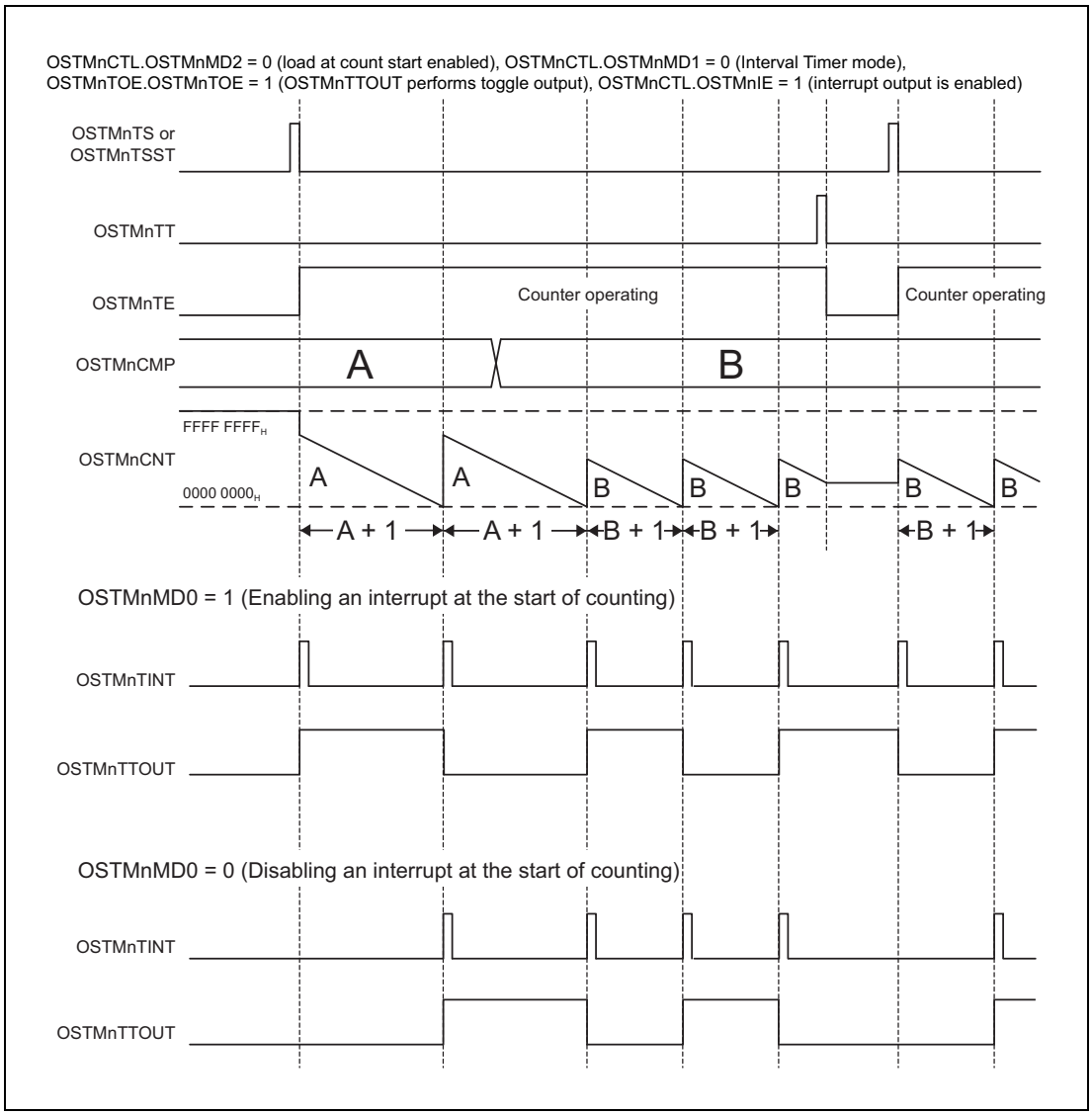


Figure 32.4 Generating an Interrupt when Counting Starts (Interval Timer Mode)

32.2.5 Counter clock

The clock signals for counting by OSTM8 and OSTM9 are defined by PCLK and OSTMnTCKE in the following ways. The count signals for counting by OSTM0 to OSTM7 are always defined by PCLK.

- If IC0CKSELn.IC0TMENn = 0, the OSTMnTCKE signal being at the high level (logical 1) selects counting of cycles of PCLK.
- IC0CKSELn.IC0TMENn being set to 1 selects the TAUDn or TAUJn signal as the counter-clock-enable signal, OSTMnTCKE. The cycles of the clock signals for counting by OSTMn are synchronized with the cycles of the clock signals generated as TAUDn or TAUJn (any from among CK0 to CK3).

This is illustrated in the following figures.

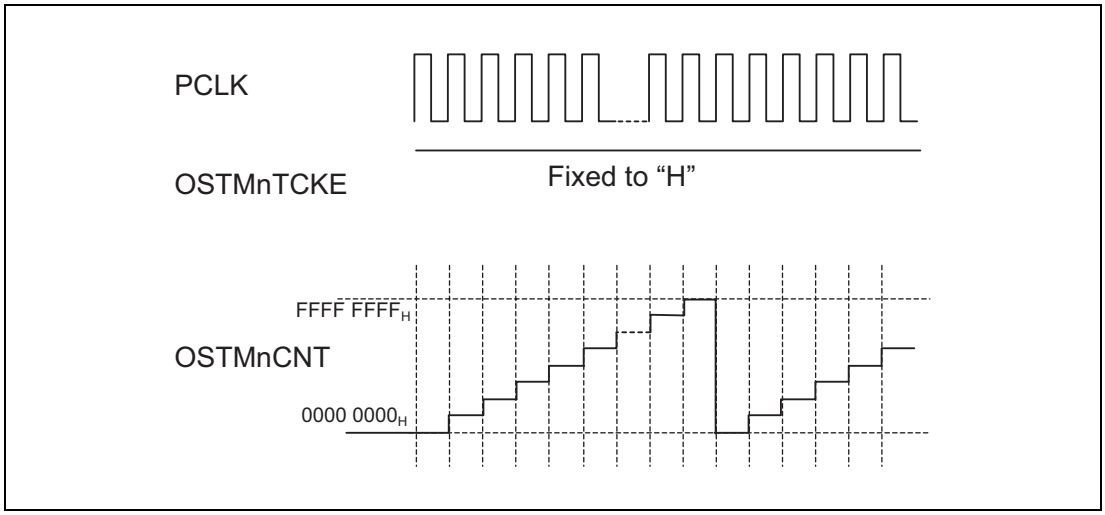


Figure 32.5 Counting when OSTMnTCKE is Fixed to H

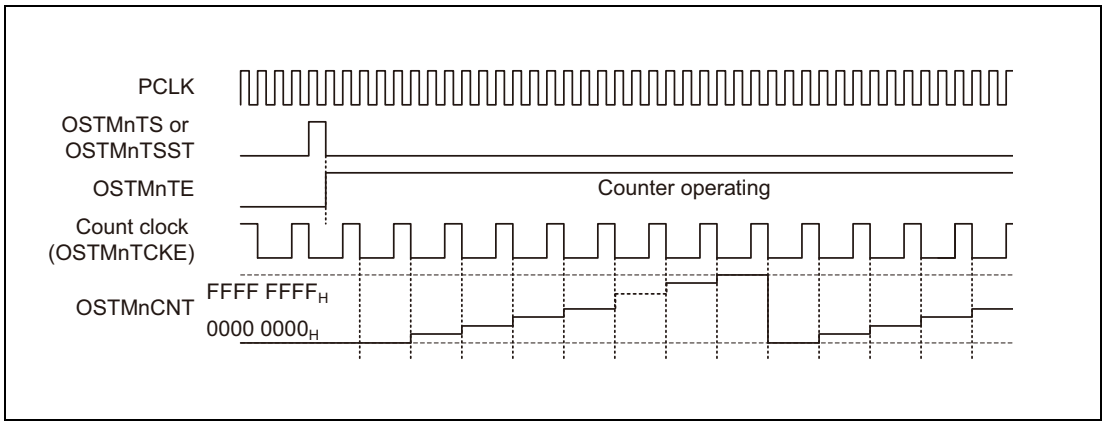


Figure 32.6 Counting in Response to Transitions of the OSTMnTCKE Input Signal

The counter-clock-enable signal (OSTMnTCKE) for each OSTMn can be selected by the IC0CKSEL8 or IC0CKSEL9 register from among 40 channels: 16 channels from each of TAUD0 and TAUD1, and 4 channels from each of TAUJ0 and TAUJ1.

For the counter-clock-enable signals for TAUDn and TAUJn (TAUDnTCKENm and TAUJnTCKENm) and the clock signals generated in TAUDn and TAUJn (CK0 to CK3), see **Figure 33.2, Block Diagram of the TAUD** and **Figure 34.2, Block Diagram of the TAUJ**.

CAUTIONS

1. **Select a counter-clock-enable signal for OSTMn while the operation of OSTMn is stopped (the OSTMnTE.OSTMnTE bit is 0).**
2. **After using the IC0CKSELn.IC0TMSELn[1:0], IC0CKSELn.IC0CKSELn3[1:0], IC0CKSELn.IC0CKSELn2[1:0], IC0CKSELn.IC0CKSELn1[3:0], and IC0CKSELn.IC0CKSELn0[3:0] bits to select the counter-clock-enable signal for OSTMn, set the IC0CKSELn.IC0TMENn bit to 1.**
3. **If TAUDn or TAUJn is selected as the source of the counter-clock-enable signal for OSTMn (the IC0CKSELn.IC0TMENn bit is 1), do not change the operation of TAUDn or TAUJn while the OSTMn is operating (the OSTMnTE.OSTMnTE bit is 1).**

   [Setting procedure]

   (1) Confirm that the OSTMnTE.OSTMnTE bit is 0 (OSTMn operation is stopped).
   (2) Set the IC0CKSELn.IC0TMSELn[1:0], IC0CKSELn.IC0CKSELn3[1:0], IC0CKSELn.IC0CKSELn2[1:0], IC0CKSELn.IC0CKSELn1[3:0], and IC0CKSELn.IC0CKSELn0[3:0] bits to select the counter-clock enable signal for the OSTMn.
   (3) Set the IC0CKSELn.IC0TMENn bit to 1.
   (4) Enable OSTMn operation by setting the OSTMnTS.OSTMnTS bit to 1.
4. **Operating system timers OSTM0 to OSTM7 do not have IC0CKSELn registers. Only PCLK is available as the clock for counting.**

32.3 Registers

The OS timers are controlled and operated by the following registers.

32.3.1 List of Registers

The list of OSTMn (n = 0 to 9) registers and the memory addresses are as follows.

For the base addresses, see **Table 32.3**.

Table 32.9 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
OSTM	OSTMn compare register	OSTMnCMP	<OSTMn_base> + 00_H	32	*1	—
	OSTMn counter register	OSTMnCNT	<OSTMn_base> + 04_H	32	*1	—
	OSTMn output register	OSTMnTO	<OSTMn_base> + 08_H	8	*1	—
	OSTMn output enable register	OSTMnTOE	<OSTMn_base> + $0C_H$	8	*1	—
	OSTMn count enable status register	OSTMnTE	<OSTMn_base> + 10_H	8	*1	—
	OSTMn count start trigger register	OSTMnTS	<OSTMn_base> + 14_H	8	*1	—
	OSTMn count stop trigger register	OSTMnTT	<OSTMn_base> + 18_H	8	*1	—
	OSTMn control register	OSTMnCTL	<OSTMn_base> + 20_H	8	*1	—
	OSTM8 clock select register	IC0CKSEL8	FFBF $08C0_H$	16	PBG 50#15	—
	OSTM9 clock select register	IC0CKSEL9	FFBF $09C0_H$	16	PBG 51#0	—

Note 1. n = 0: PBG50#7
n = 1: PBG50#8
n = 2: PBG50#9
n = 3: PBG50#10
n = 4: PBG50#11
n = 5: PBG50#12
n = 6: PBG50#13
n = 7: PBG50#14
n = 8: PBG50#15
n = 9: PBG51#0

32.3.2 Details of OSTMn Registers

32.3.2.1 OSTMnCMP — OSTMn Compare Register

Depending on the mode of operation, this register holds the start value for the down-counter or the value for comparison with that of the counter.

Access: This register can be read or written in 32-bit units.

Address: <OSTMn_base> + 00_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.10 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP [31:0]	• In interval timer mode: Start value of the down-counter • In free-run compare mode: Value for comparison

32.3.2.2 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register can be read or written in 32-bit units. Writing can only proceed when the counter is disabled (OSTMnTE.OSTMnTE = 0).

Address: <OSTMn_base> + 04_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.11 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT [31:0]	32-bit counter value

When the operation mode of the counter is changed with OSTMnCTL.OSTMnMD2 = 1, write any value to OSTMnCNT register before counter operation start.

Table 32.12 lists the correspondence between the OSTM operating mode, counting direction, and start value.

The start value indicates the value to be read after the operating mode is changed.

Table 32.12 Correspondence between Operating Mode, Counting Direction and Start Value

Timer operating mode	OSTMnCTL.OSTMnMD1	Counting direction	Start value*2
Interval timer mode	0*1	Down count	FFFF $FFFF_H$
Free-run compare mode	1	Up count	0000 0000_H

Note 1. Value after reset.

Note 2. The start value is the value that is read after the operating mode has been changed following reset input.
When OSTMnTE.OSTMnTE = 0 (counter stop status), the OSTMnCNT register can be written.
When the operation mode of the counter is changed with OSTMnCTL.OSTMnMD2 = 1, write any value to the OSTMnCNT register before the counter operation starts.

32.3.2.3 OSTMnTO — OSTMn Output Register

The OSTM output register is used to specify and read the level of an OSTMnTTOUT output signals. The setting of this register is only valid in OSTM0, OSTM8 and OSTM9.

Access: This register can be read or written in 8-bit units. Writing can only proceed when the software control mode is selected (OSTMnTOE.OSTMnTOE = 0).

Address: <OSTMn_base>+ 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTO
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 32.13 OSTMnTO Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OSTMnTO	This bit specifies/reads the level of OSTMnTTOUT output signals. 0: Low level 1: High level

32.3.2.4 OSTMnTOE — OSTMn Output Enable Register

The OSTM output enable register specifies OSTMnTTOUT output mode. The setting of this register is only valid in OSTM0, OSTM8 and OSTM9.

Access: This register can be read or written in 8-bit units.

Address: <OSTMn_base>+ $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 32.14 OSTMnTOE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OSTMnTOE	This bit specifies the OSTMnTTOUT output mode. 0: Software control mode: The level corresponding to the setting of OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Timer-output toggling mode: OSTMnTTOUT output is toggled whenever an OSTMnTINT interrupt request is generated.

32.3.2.5 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <OSTMn_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.15 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OSTMnTE	This bit indicates whether the counter is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1, or to the OSTMnTSST signal (when timer synchronization of PIC is in use) becoming 1. This bit is reset to 0 when the OSTMnTT.OSTMnTT bit is set to 1.

NOTE

When OSTMnTE.OSTMnTE = 0, the counter retains its value.

If the counter is restarted when the OSTMnCTL.OSTMnMD2 = 0,

- It restarts counting down from the value in the OSTMnCMP register in interval timer mode
- It restarts counting up from the counter value 0000 0000_H in free-run compare mode

If OSTMnCTL.OSTMnMD2 = 1, the counter is not initialized.

32.3.2.6 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.16 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTS	This bit starts the counter. 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. • In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

32.3.2.7 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <OSTMn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.17 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTT	Stops the counter: 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

32.3.2.8 OSTMnCTL — OSTMn Control Register

This register specifies the operating mode for the counter and controls the generation of OSTMnTINT interrupt requests when counting starts.

Although this register is readable and writable, writing to it is only possible when OSTMnTE.OSTMnTE = 0; that is, the register becomes read-only when OSTMnTE.OSTMnTE = 1.

Access: This register can be read or written in 8-bit units.

Address: <OSTMn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	OSTMnIE	—	—	—	—	OSTMnMD2	OSTMnMD1	OSTMnMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R/W	R/W	R/W

Table 32.18 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
7	OSTMnIE	OSTM Interrupt enable This bit is the enable bit of the OSTM Interrupt (OSTMnTINT) 0: OSTMn interrupt request is disabled. 1: OSTMn interrupt request is enabled.
6 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	OSTMnMD2	Control of the counter operation at counter operation start This bit controls the counter operation at counter operation start regardless of whether the counter currently is disabled or enabled (OSTMnTE.OSTMnTE = 0 to 1). 0: OSTMnCNT is loaded into at counter operation start The following describes the value of the counter at counter operation start. – Interval timer mode: The OSTMnCMP[31:0] value is loaded to OSTMnCNT[31:0] at counter start. – Free-run compare mode: The value of $0000\ 0000_H$ is loaded to OSTMnCNT[31:0] at counter operation start. 1: OSTMnCNT is not loaded into at counter operation start
1	OSTMnMD1	Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls enabling/disabling of OSTMnTINT interrupt requests when counting starts. 0: Disables the interrupts when counting starts. 1: Enables the interrupts when counting starts.

32.3.2.9 IC0CKSEL8 — OSTM8 Clock Select Register

This register selects the counter-clock-enable signal as the clock source to drive counting for the OSTM8 counter.

Access: This register can be read or written in 16-bit units.

Address: FFBF $08C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC0TMEN8	—	IC0TMSEL8[1:0]		IC0CKSEL83[1:0]		IC0CKSEL82[1:0]		IC0CKSEL81[3:0]				IC0CKSEL80[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.19 IC0CKSEL8 Register Contents (1/2)

Bit Position	Bit Name	Function
15	IC0TMEN8	Selects the counter-clock-enable signal for OSTM8. 0: The counter-clock-enable signal is fixed to 1 (PCLK is selected as the source of the clock signal for counting) 1: Selects the peripheral module selected by IC0TMSEL8[1:0] as the counter-clock-enable signal.
14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	IC0TMSEL8 [1:0]	Select the counter-clock-enable signal for the OSTM8 counter (Enabled only when IC0TMEN8 = 1). **IC0TMSEL8[1:0] / Selected Peripheral Function** 00_B TAUD0 01_B TAUD1 10_B TAUJ0 11_B TAUJ1
11, 10	IC0CKSEL83 [1:0]	Select the counter-clock-enable signal for the OSTM8 counter (Enabled only when IC0TMEN8 = 1 and IC0TMSEL8[1:0] = 11_B). **IC0CKSEL83[1:0] / Selected TAUJ1 Channel** 00_B TAUJ1 channel 0 01_B TAUJ1 channel 1 10_B TAUJ1 channel 2 11_B TAUJ1 channel 3
9, 8	IC0CKSEL82 [1:0]	Select the counter-clock-enable signal for the OSTM8 counter (Enabled only when IC0TMEN8 = 1 and IC0TMSEL8[1:0] = 10_B). **IC0CKSEL82[1:0] / Selected TAUJ0 Channel** 00_B TAUJ0 channel 0 01_B TAUJ0 channel 1 10_B TAUJ0 channel 2 11_B TAUJ0 channel 3

Table 32.19 IC0CKSEL8 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>IC0CKSEL81 [3:0]</td><td>Select the counter-clock-enable signal for the OSTM8 counter (Enabled only when IC0TMEN8 = 1 and IC0TMSEL8[1:0] = 01_B).
<table>
<tr><th>IC0CKSEL81[3:0]</th><th>Selected TAUD1 Channel</th></tr>
<tr><td>0000_B</td><td>TAUD1 channel 0</td></tr>
<tr><td>0001_B</td><td>TAUD1 channel 1</td></tr>
<tr><td>0010_B</td><td>TAUD1 channel 2</td></tr>
<tr><td>:</td><td></td></tr>
<tr><td>1101_B</td><td>TAUD1 channel 13</td></tr>
<tr><td>1110_B</td><td>TAUD1 channel 14</td></tr>
<tr><td>1111_B</td><td>TAUD1 channel 15</td></tr>
</table></td></tr>
<tr><td>3 to 0</td><td>IC0CKSEL80 [3:0]</td><td>Select the counter-clock-enable signal for the OSTM8 counter (Enabled only when IC0TMEN8 = 1 and IC0TMSEL8[1:0] = 00_B).
<table>
<tr><th>IC0CKSEL80[3:0]</th><th>Selected TAUD0 Channel</th></tr>
<tr><td>0000_B</td><td>TAUD0 channel 0</td></tr>
<tr><td>0001_B</td><td>TAUD0 channel 1</td></tr>
<tr><td>0010_B</td><td>TAUD0 channel 2</td></tr>
<tr><td>:</td><td></td></tr>
<tr><td>1101_B</td><td>TAUD0 channel 13</td></tr>
<tr><td>1110_B</td><td>TAUD0 channel 14</td></tr>
<tr><td>1111_B</td><td>TAUD0 channel 15</td></tr>
</table></td></tr>
</table>

32.3.2.10 IC0CKSEL9 — OSTM9 Clock Select Register

This register selects the counter-clock-enable signal as the clock source to drive counting for the OSTM9 counter.

Access: This register can be read or written in 16-bit units.

Address: FFBF 09C0$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IC0TMEN9	—	IC0TMSEL9 [1:0]		IC0CKSEL93 [1:0]		IC0CKSEL92 [1:0]		IC0CKSEL91 [3:0]				IC0CKSEL90 [3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.20 IC0CKSEL9 Register Contents (1/2)

Bit Position	Bit Name	Function
15	IC0TMEN9	Selects the counter-clock-enable signal for OSTM9. 0: The counter-clock-enable signal is fixed to 1 (PCLK is selected as the source of the clock signal for counting) 1: Selects the peripheral module selected by IC0TMSEL9[1:0] as the counter-clock-enable signal.
14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	IC0TMSEL9 [1:0]	Select the counter-clock-enable signal for the OSTM9 counter (Enabled only when IC0TMEN9 = 1). **IC0TMSEL9[1:0] / Selected Peripheral Function** 00_B TAUD0 01_B TAUD1 10_B TAUJ0 11_B TAUJ1
11, 10	IC0CKSEL93 [1:0]	Select the counter-clock-enable signal for the OSTM9 counter (Enabled only when IC0TMEN9 = 1 and IC0TMSEL9[1:0] = 11_B). **IC0CKSEL93[1:0] / Selected TAUJ1 Channel** 00_B TAUJ1 channel 0 01_B TAUJ1 channel 1 10_B TAUJ1 channel 2 11_B TAUJ1 channel 3
9, 8	IC0CKSEL92 [1:0]	Select the counter-clock-enable signal for the OSTM9 counter (Enabled only when IC0TMEN9 = 1 and IC0TMSEL9[1:0] = 10_B). **IC0CKSEL92[1:0] / Selected TAUJ0 Channel** 00_B TAUJ0 channel 0 01_B TAUJ0 channel 1 10_B TAUJ0 channel 2 11_B TAUJ0 channel 3

Table 32.20 IC0CKSEL9 Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 4	IC0CKSEL91 [3:0]	Select the counter-clock-enable signal for the OSTM9 counter (Enabled only when IC0TMEN9 = 1 and IC0TMSEL9[1:0] = 01_B). **IC0CKSEL91[3:0]** / **Selected TAUD1 Channel** 0000_B TAUD1 channel 0 0001_B TAUD1 channel 1 0010_B TAUD1 channel 2 : 1101_B TAUD1 channel 13 1110_B TAUD1 channel 14 1111_B TAUD1 channel 15
3 to 0	IC0CKSEL90 [3:0]	Select the counter-clock-enable signal for the OSTM9 counter (Enabled only when IC0TMEN9 = 1 and IC0TMSEL9[1:0] = 00_B). **IC0CKSEL90[3:0]** / **Selected TAUD0 Channel** 0000_B TAUD0 channel 0 0001_B TAUD0 channel 1 0010_B TAUD0 channel 2 : 1101_B TAUD0 channel 13 1110_B TAUD0 channel 14 1111_B TAUD0 channel 15

32.4 Operation

Each OS timer is a 32-bit timer/counter.

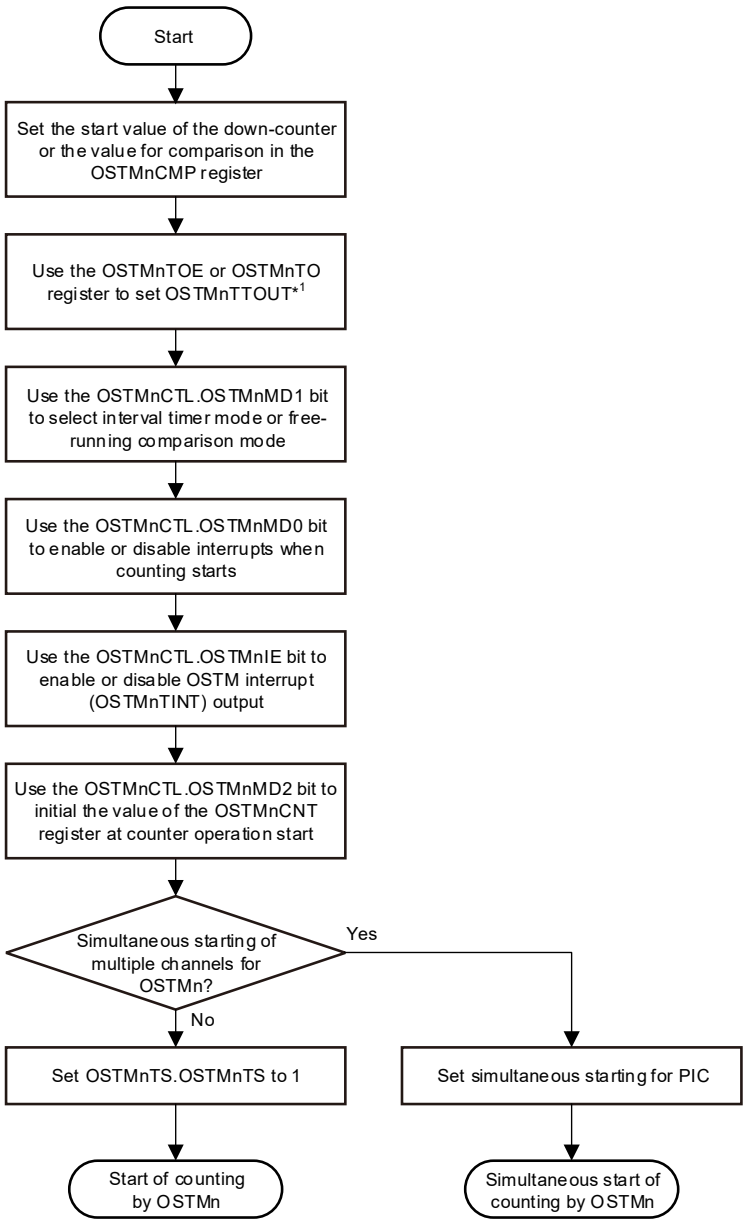
The settings for the operating mode specify the direction of counting (up or down) and the generation of interrupt requests.

32.4.1 Starting and Stopping OSTMn

The OSTMn is started and stopped as follows:

Starting the timer

Figure 32.7, Flow of Starting the OS Timer shows the flow of starting the counter of OSTMn. For details on the operations of the timer in interval timer mode and free-run compare mode in a given step of the figure, see the descriptions in **Section 32.4.2, Interval Timer Mode** and **Section 32.4.3, Free-Run Compare Mode**.



Note 1. For generating an output waveform on the $\overline{\text{ERROROUT\_M}}$ pin or $\overline{\text{ERROROUT\_C}}$ pin and output waveform for OSTM8 and OSTM9, make settings for only OSTM0, OSTM8 and OSTM9 to control OSTM0TTOUT, OSTM8TTOUT and OSTM9TTOUT.
This setting is not required for OSTMn (n = 1 to 7).

Figure 32.7 Flow of Starting the OS Timer

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the timer and clears the OSTMnTE.OSTMnTE status flag bit.

Timer Synchronization (PIC)

The OSTMnTSST signal can be used to start multiple timers at the same time. For detailed connections, refer to **Section 41, Peripheral Interconnect (PIC)**.

32.4.2 Interval Timer Mode

Select the interval timer mode when an OS timer is to be used as a reference timer for generating interrupt requests at a fixed interval.

32.4.2.1 Basic Operation in Interval Timer Mode

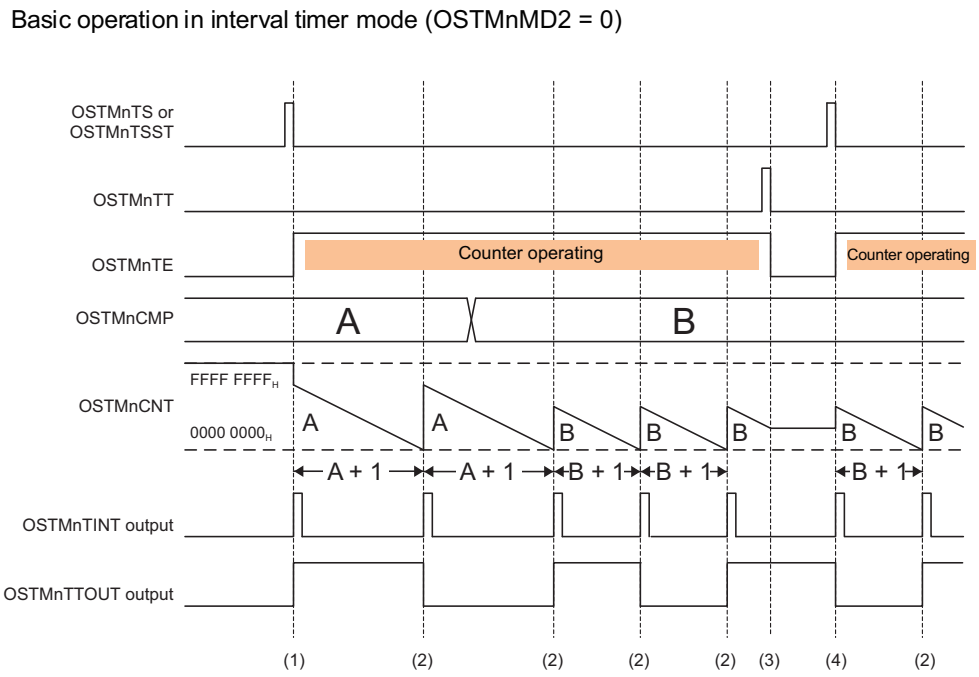
The interval timer mode is set when count mode selection bit OSTMnMD1 is set to “0”. In the interval timer mode, the OSTM can be used as the reference timer generating the OSTMnTINT interrupt at fixed intervals. Upon occurrence of an interrupt, OSTMnTTOUT performs a toggle operation and outputs a square wave. The periods of OSTMnTINT and OSTMnTTOUT are as follows.

- OSTMnTINT generation cycle = Count-clock cycle × (OSTMnCMP + 1)
- OSTMnTTOUT output cycle = OSTMnTINT generation period × 2

OSTMnCMP can be rewritten at any timing, and if it is rewritten during OSTMnCNT operation (OSTMnTE = 1), OSTMnCNT reloads the new OSTMnCMP value upon the next 0000 0000H match detection, and the count operation continues. Then, when “1” is written to OSTMnTT to clear OSTMnTE to “0”, the counter and OSTMnTO stops operation while holding their values as they are.

When OSTMnMD2 = 0, OSTMnCMP[31:0] value is stored to OSTMnCNT at counter operation start.

When OSTMnMD2 = 1, OSTMnCMP[31:0] value is not stored to OSTMnCNT at counter operation start.



Note 1. OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled)

Figure 32.8 Timing Diagram of OSTMn in Interval Timer Mode (OSTMnMD2 = 0)

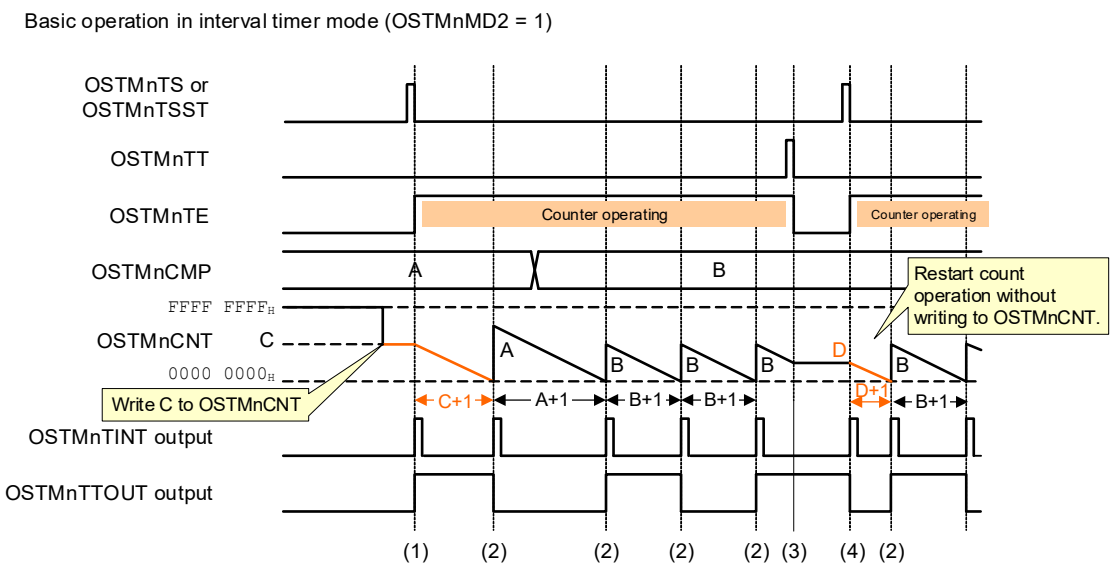
The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting-down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 is 1, OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT signal is toggled. The OSTMnCNT register indicates the counter value.

(2) When the counter reaches 0000 0000_H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST = 1), the counter loads the new start value from OSTMnCMP and starts counting down.



Note: OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled)

Figure 32.9 Timing Diagram of OSTMn in Interval Timer Mode (OSTMnMD2 = 1)

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting-down from the value written to OSTMnCNT.
If OSTMnCTL.OSTMnMD0 is 1, OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT signal is toggled. The OSTMnCNT register contains the current value as the counter.

(2) When the counter reaches 0000 0000_H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

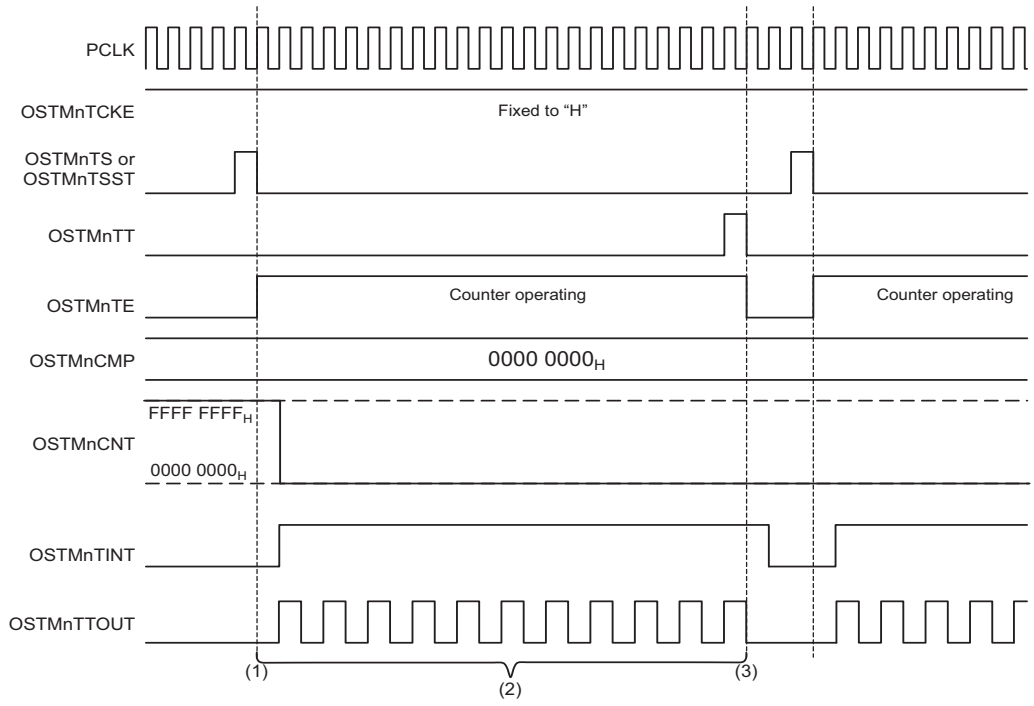
(4) When counting is restarted (OSTMnTS.OSTMnTS = 1, or OSTMnTSST = 1), the counter keeps the value when it was stopped and starts counting down.

32.4.2.2 Operation when OSTMnCMP = 0000 0000$_H$

When PCLK is selected as the source of the clock signal for counting and OSTMnCMP = 0000 0000$_H$, the OS timer behaves as follows.

If the counter is enabled, the OSTMnTINT interrupt request will always be set to 1. However, the timer (OSTMnTTOUT) output signal can still be used. Timer (OSTMnTTOUT) output using timer output toggling mode results in OSTMnTTOUT being toggled on every cycle of the counter clock.

The following figure shows operations of the OSTMn when PCLK is selected as the source of the clock signal for counting and OSTMnCMP = 0000 0000$_H$, the counter start interrupt is enabled (OSTMnCTL.OSTMnMD0 = 1), and the timer is in timer output toggling mode, where OSTMnTTOUT is toggled (OSTMnTOE.OSTMnTOE = 1).



Note: OSTMnMD2 = 0 (load at count start enabled), OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled)

Figure 32.10 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000$_H$ is retained in OSTMnCNT.

(2) While the OSTMnTINT interrupt request signal is continuously asserted, OSTMnTTOUT is toggled (in **Figure 32.10**, OSTMnTINT is fixed to the high level because PCLK is selected as the counter clock).

(3) After the counter stops, the OSTMnTINT interrupt request signal is deasserted and the output level of the OSTMnTTOUT signal is maintained.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the counter clock is not generated at the start timing of the counter.

32.4.2.3 Forced Restart Operation in Interval Timer Mode

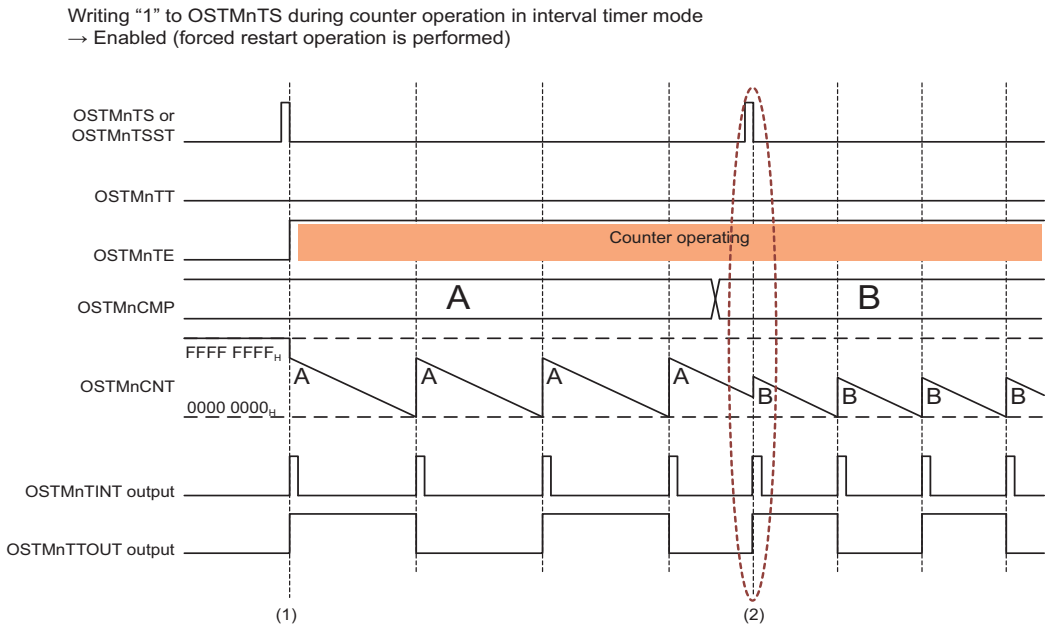
Forced restart operation is performed by writing “1” to the OSTMnTS count start register during the count operation (OSTMnTE = 1) in the interval timer mode. At this time, the OSTMnCNT counter immediately reloads the OSTMnCMP compare register and the count operation continues.

CAUTION

If the OSTMnTS.OSTMnTS bit for a desired channel is set to 1 while simultaneous starting of multiple channels of the timer is in progress, the individual channels are forcibly restarted, so synchronous operation between the channels is not guaranteed.

If simultaneous starting of PIC is used while multiple channels of the timer start individually, the timers set for simultaneous starting are forcibly restarted at the same time.

For OSTM0 to OSTM7, forced restart operation is performed regardless of the value of OSTMnCTL.OSTMnMD2.
For OSTM8 to OSTM9, forced restart operation is performed regardless of the value of OSTMnCTL.OSTMnMD2 when IC0CKSELn.IC0TMENn = 0. Otherwise, forced restart operation is performed before count start in OSTMnCTL.OSTMnMD2 = 1, OSTMnCNT not loads OSTMnCMP value, but starts counting at any value. See **Figure 32.12, Forced Restart Operation while OSTMnMD2 = 1 in Interval Timer Mode (for OSTM8 and OSTM9 only)**.



Note: OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled).

Figure 32.11 Forced Restart Operation in Interval Timer Mode

The operations shown in the above timing diagram are as follows.

(1) The counter is started and stopped as described under **Figure 32.8** or **Figure 32.9**.

(2) Setting OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMnTINT interrupt request is generated when counting starts and the OSTMnTTOUT signal is toggled.

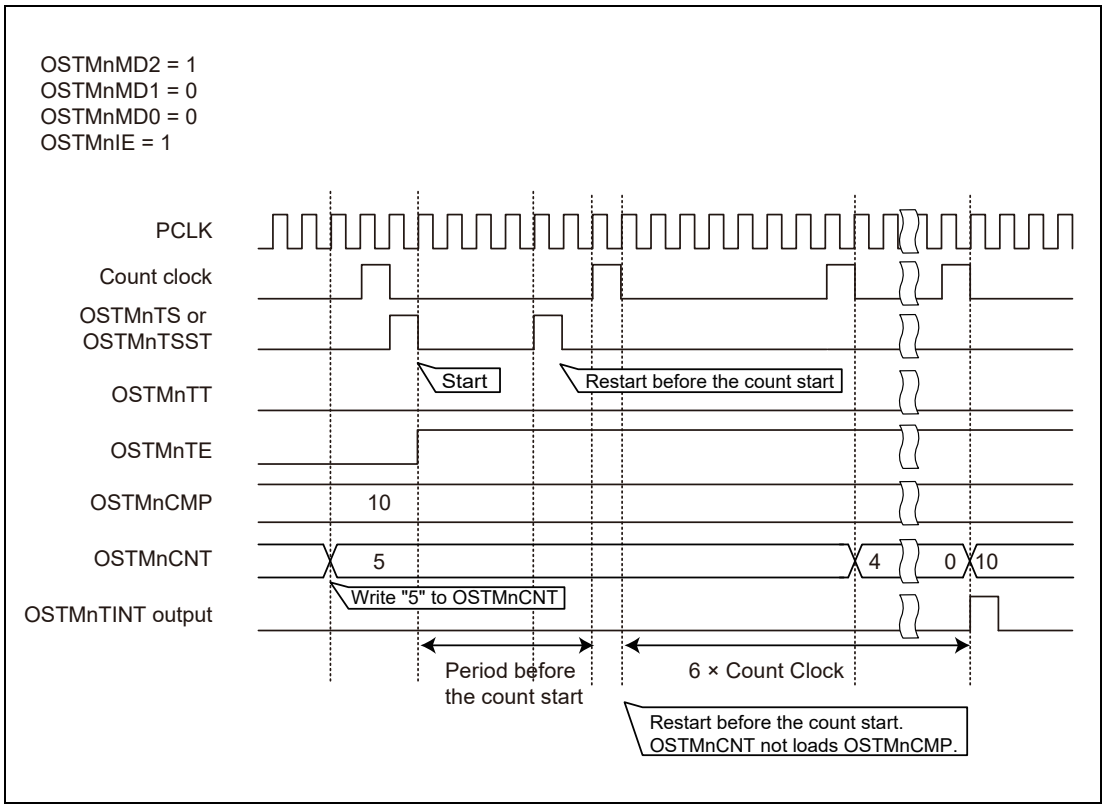


Figure 32.12 Forced Restart Operation while OSTMnMD2 = 1 in Interval Timer Mode (for OSTM8 and OSTM9 only)

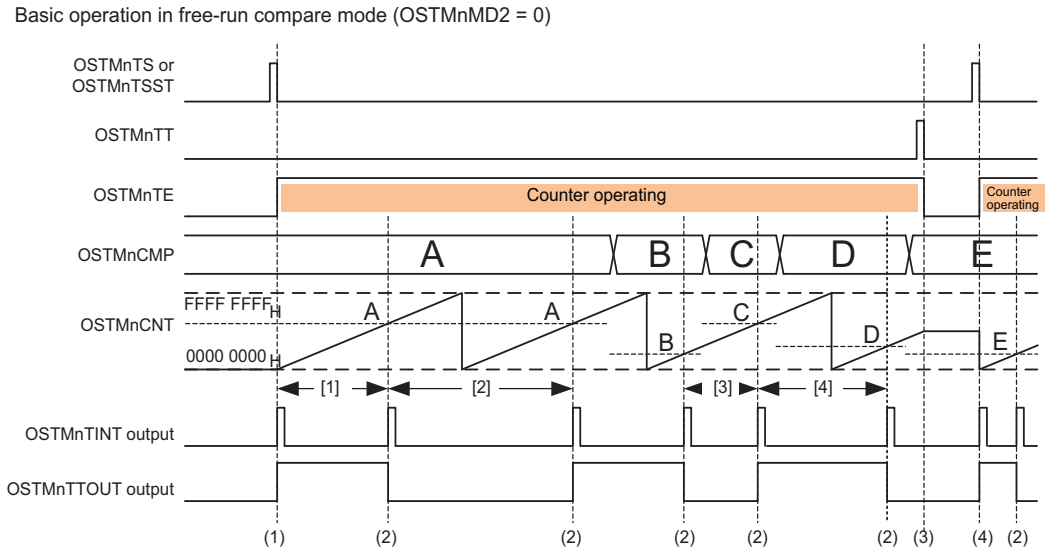
32.4.3 Free-Run Compare Mode

32.4.3.1 Basic Operation in Free-Run Compare Mode

The free-run compare mode is set when count mode selection bit OSTMnMD1 is set to “1”. In the free-run compare mode, the OSTMnTINT interrupt is output when the value of the OSTMnCMP compare register and the count value match. OSTMnTTOUT performs a toggle operation.

When OSTMnMD2 = 0, OSTMnCNT[31:0] is cleared to 0000 0000_H at counter operation start.

When OSTMnMD2 = 1, OSTMnCNT is not cleared at counter operation start.



Note: OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled).

Figure 32.13 Timing Diagram of OSTMn in Free-Run Compare Mode (OSTMnMD2 = 0)

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000_H to FFFF $FFFF_H$. The OSTMnCNT register indicates the counter value. When the OSTMnCTL.OSTMnMD0 set to 1, the interrupt request OSTMnTINT is generated when the counting starts.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

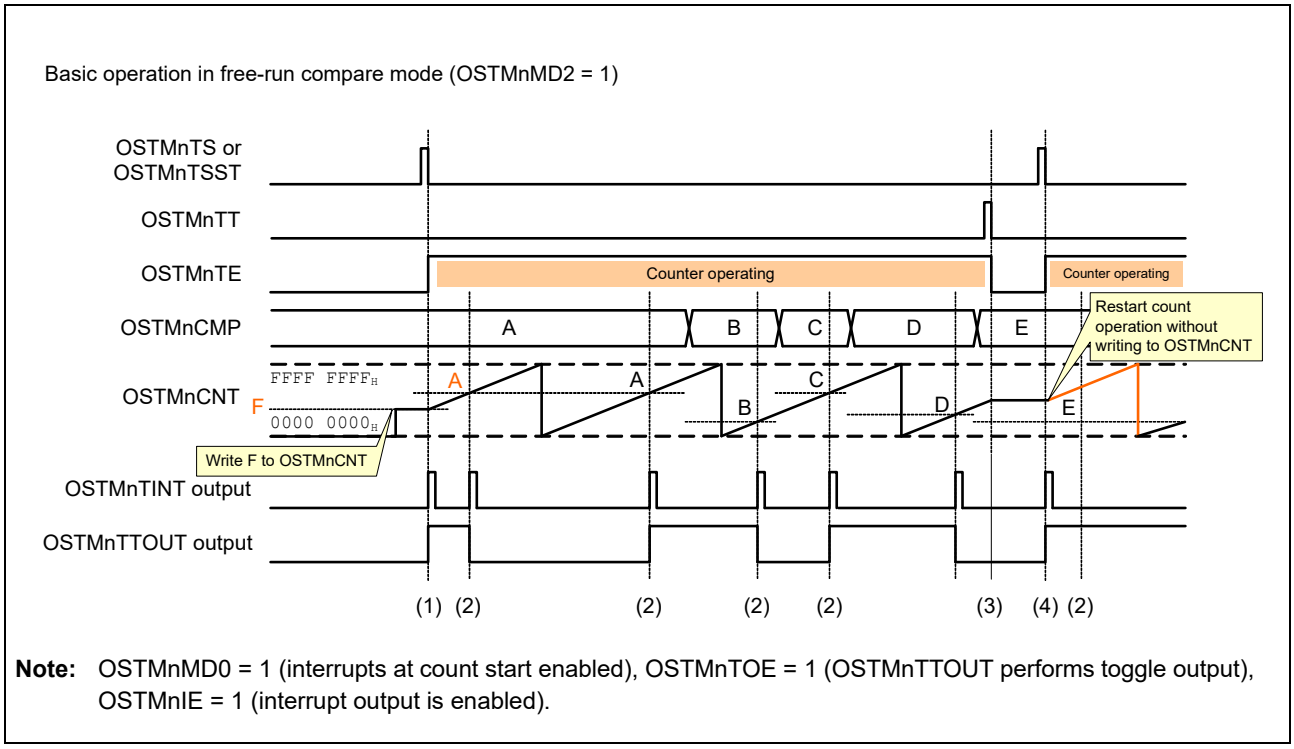
(4) When restarting the counter by setting OSTMnTS.OSTMnTS = 1, the counter starts from 0000 0000_H.

The OSTMnTINT and OSTMnTTOUT occurrence timing in the free-run compare mode may not be a constant period depending on the usage conditions. This occurs at count start and when OSTMnCMP is rewritten during the count operation. This is explained using **Figure 32.13**.

Table 32.21 Timing of OSTMnTINT Generation

Old value for comparison	New value for comparison	Counter value at time of rewriting	Period of OSTMnTINT Generation	Label in timing diagram
Counter starts			(A + 1) × counter clock period	[1]
A	A	No rewriting	(FFFF FFFF$_H$ + 1) × counter clock period	[2]
B	C > B	B < counter value < C	(C - B) × counter clock period	[3]
C	D < C	Counter value > D,C	(FFFF FFFF$_H$ - C + D + 1) × counter clock period	[4]

Even during the count operation, OSTMnCMP rewrite is immediately reflected, and OSTMnCNT match judgment is executed. When "1" is written to OSTMnTT to set OSTMnTE = 0, the counter and OSTMnTO stop operation while holding their values.



Note: OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE = 1 (interrupt output is enabled).

Figure 32.14 Timing Diagram of OSTMn in Free-Run Compare Mode (OSTMnMD2 = 1)

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter starts count up to FFFF FFFF$_H$ from the value written to the OSTMnCNT register. The OSTMnCNT register is the counter, so it contains the current value. When the OSTMnCTL.OSTMnMD0 set to 1, the OSTMnTINT interrupt request is generated when counting starts.

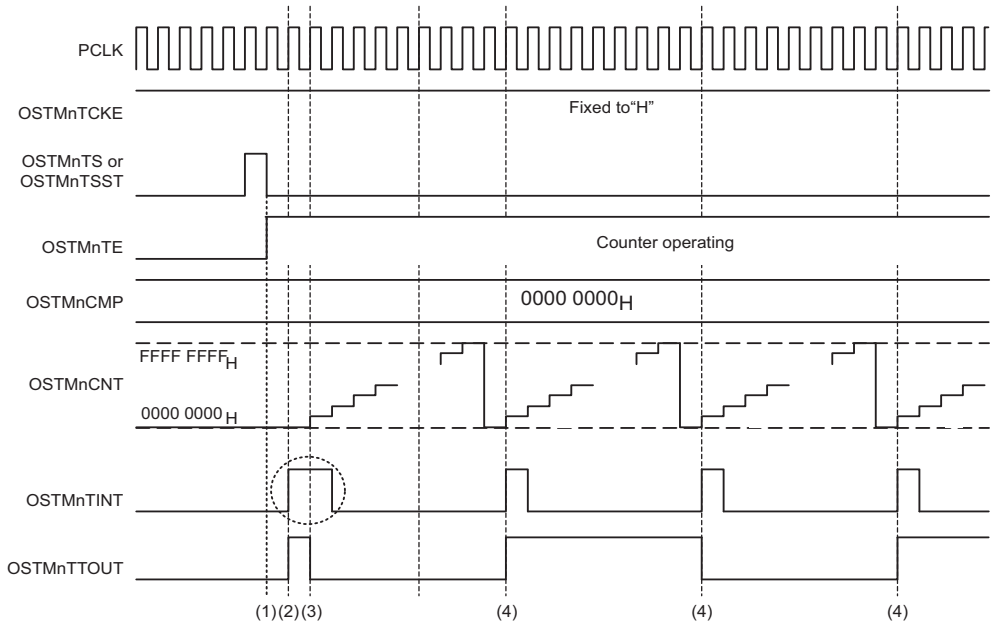
(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT signal is toggled.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from the value at which counting stopped when OSTMnTS.OSTMnTS = 1, or when OSTMnTSST = 1.

32.4.3.2 Operation when OSTMnCMP = 0000 0000$_H$

The following figure shows the operation of OSTMn when PCLK is selected as the source of the clock signal for counting and OSTMnCMP = 0000 0000$_H$, counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1) and the timer is in timer output toggling mode, where OSTMnTTOUT is toggled (OSTMnTOE.OSTMnTOE = 1).



Note: OSTMnMD2 = 0 (load at count start enabled), OSTMnMD0 = 1 (interrupts at count start enabled), OSTMnTOE = 1 (OSTMnTTOUT performs toggle output), OSTMnIE=1 (interrupt output is enabled)

Figure 32.15 Timing Diagram when OSTMnCMP = 0000 0000H in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from 0000 0000$_H$ to FFFF FFFF$_H$.

(2) An OSTMnTINT interrupt request is generated when counting starts and the OSTMnTTOUT signal is toggled.

(3) If the current counter value matches OSTMnCMP, the interrupt request OSTMnTINT is generated.
If OSTMnCMP = 0000 0000$_H$ as shown above, OSTMnTINT is generated over two clock cycles and the OSTMnTTOUT signal is toggled.

(4) For every (FFFF FFFF$_H$ + 1) clock cycles, the OSTMnTINT interrupt request signal is asserted and the OSTMnTTOUT signal is toggled.

When an interrupt is prohibited when the counter starts, 1 clock cycle of the counter clock is not generated at the start timing of the counter.

32.4.3.3 Forced Restart Operation in Free-Run Compare Mode

In the free-run compare mode, forced restart operation is not performed even when “1” is written to OSTMnTS during the count operation (OSTMnTS = 1). The OSTMnCNT counter continues counting at this time.

Forced restart operation is not performed regardless of the value of OSTMnCTL.OSTMnMD2.

Section 33 Timer Array Unit D (TAUD)

This section contains a generic description of the Timer Array Unit D (TAUD).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the TAUD.

33.1 Features TAUD for RH850/U2A-EVA

33.1.1 Number of Units and Channels

This microcontroller has the following number of TAUD units.

Table 33.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)
Name	TAUDn									

TAUDn has the following timers for the quantity of channels of timers.

Table 33.2 TAUDn Unit Configurations and Channels

Unit Name (Channel Name) TAUDn	Channels per Unit	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
TAUD0	16	16	16	16	16	16	16	16	16	16	16
TAUD1	16	16	16	16	16	16	16	16	16	16	16
TAUD2	16	16	16	16	16	16	16	16	16	16	16

Table 33.3 Index

Index	Description
n	Throughout this section, the individual TAUD units are identified by the index “n” (n = 0 to 2)
m	The TAUD has 16 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

33.1.2 Register Base Addresses

TAUDn base addresses are listed in the following table.

TAUDn register addresses are given as offsets from the base addresses.

Table 33.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<TAUD0_base>	FFBF 4000_H	P-Bus Group 5
<TAUD1_base>	FFBF 5000_H	P-Bus Group 5
<TAUD2_base>	FFBF 6000_H	P-Bus Group 5

33.1.3 Clock Supply

Clock supply by and to TAUDn is listed in the following table.

Table 33.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUDn	PCLK	Peripheral high speed clock CLK_HSB

33.1.4 Interrupt Requests and Error Notifications

TAUDn interrupt requests are listed in the following table.

Table 33.6 Interrupt and DMA/DTS Requests (1/2)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
TAUD0				
INTTAUD0I0	Channel 0 interrupt	10	Group 0-192	Group 1-64
INTTAUD0I1	Channel 1 interrupt	376	Group 0-193	Group 1-65
INTTAUD0I2	Channel 2 interrupt	11	Group 0-194	Group 1-66
INTTAUD0I3	Channel 3 interrupt	377	Group 0-195	Group 1-67
INTTAUD0I4	Channel 4 interrupt	12	Group 0-196	Group 1-68
INTTAUD0I5	Channel 5 interrupt	378	Group 0-197	Group 1-69
INTTAUD0I6	Channel 6 interrupt	13	Group 0-198	Group 1-70
INTTAUD0I7	Channel 7 interrupt	379	Group 0-199	Group 1-71
INTTAUD0I8	Channel 8 interrupt	23	Group 0-200	Group 1-72
INTTAUD0I9	Channel 9 interrupt	380	Group 0-201	Group 1-73
INTTAUD0I10	Channel 10 interrupt	24	Group 0-202	Group 1-74
INTTAUD0I11	Channel 11 interrupt	381	Group 0-203	Group 1-75
INTTAUD0I12	Channel 12 interrupt	25	Group 0-204	Group 1-76
INTTAUD0I13	Channel 13 interrupt	382	Group 0-205	Group 1-77
INTTAUD0I14	Channel 14 interrupt	26	Group 0-206	Group 1-78
INTTAUD0I15	Channel 15 interrupt	383	Group 0-207	Group 1-79
TAUD1				
INTTAUD1I0	Channel 0 interrupt	384	Group 0-208	Group 1-80
INTTAUD1I1	Channel 1 interrupt	385	Group 0-209	Group 1-81

Table 33.6 Interrupt and DMA/DTS Requests (2/2)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
INTTAUD1I2	Channel 2 interrupt	386	Group 0-210	Group 1-82
INTTAUD1I3	Channel 3 interrupt	387	Group 0-211	Group 1-83
INTTAUD1I4	Channel 4 interrupt	388	Group 0-212	Group 1-84
INTTAUD1I5	Channel 5 interrupt	389	Group 0-213	Group 1-85
INTTAUD1I6	Channel 6 interrupt	390	Group 0-214	Group 1-86
INTTAUD1I7	Channel 7 interrupt	391	Group 0-215	Group 1-87
INTTAUD1I8	Channel 8 interrupt	392	Group 0-216	Group 1-88
INTTAUD1I9	Channel 9 interrupt	393	Group 0-217	Group 1-89
INTTAUD1I10	Channel 10 interrupt	394	Group 0-218	Group 1-90
INTTAUD1I11	Channel 11 interrupt	395	Group 0-219	Group 1-91
INTTAUD1I12	Channel 12 interrupt	396	Group 0-220	Group 1-92
INTTAUD1I13	Channel 13 interrupt	397	Group 0-221	Group 1-93
INTTAUD1I14	Channel 14 interrupt	398	Group 0-222	Group 1-94
INTTAUD1I15	Channel 15 interrupt	399	Group 0-223	Group 1-95
TAUD2				
INTTAUD2I0	Channel 0 interrupt	400	Group 0-224	Group 1-96
INTTAUD2I1	Channel 1 interrupt	401	Group 0-225	Group 1-97
INTTAUD2I2	Channel 2 interrupt	402	Group 0-226	Group 1-98
INTTAUD2I3	Channel 3 interrupt	403	Group 0-227	Group 1-99
INTTAUD2I4	Channel 4 interrupt	404	Group 0-228	Group 1-100
INTTAUD2I5	Channel 5 interrupt	405	Group 0-229	Group 1-101
INTTAUD2I6	Channel 6 interrupt	406	Group 0-230	Group 1-102
INTTAUD2I7	Channel 7 interrupt	407	Group 0-231	Group 1-103
INTTAUD2I8	Channel 8 interrupt	408	Group 0-232	Group 1-104
INTTAUD2I9	Channel 9 interrupt	409	Group 0-233	Group 1-105
INTTAUD2I10	Channel 10 interrupt	410	Group 0-234	Group 1-106
INTTAUD2I11	Channel 11 interrupt	411	Group 0-235	Group 1-107
INTTAUD2I12	Channel 12 interrupt	412	Group 0-236	Group 1-108
INTTAUD2I13	Channel 13 interrupt	413	Group 0-237	Group 1-109
INTTAUD2I14	Channel 14 interrupt	414	Group 0-238	Group 1-110
INTTAUD2I15	Channel 15 interrupt	415	Group 0-239	Group 1-111

This module has no error notifications.

33.1.5 Reset Sources

TAUDn reset sources are listed in the following table. TAUDn is initialized by these reset sources.

Table 33.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
TAUDn	All registers	√	√	√	√	√	√	—

33.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below.

Table 33.8 External Input/Output Signals (1/3)

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
TAUD0			
TAUD0TTIN0	I	Channel 0 Input*1*2	TAUD0I0
TAUD0TTIN1	I	Channel 1 Input*1*2	TAUD0I1
TAUD0TTIN2	I	Channel 2 Input*1*2	TAUD0I2
TAUD0TTIN3	I	Channel 3 Input*1*2	TAUD0I3
TAUD0TTIN4	I	Channel 4 Input*1*2	TAUD0I4
TAUD0TTIN5	I	Channel 5 Input*1*2	TAUD0I5
TAUD0TTIN6	I	Channel 6 Input*1*2	TAUD0I6
TAUD0TTIN7	I	Channel 7 Input*1*2	TAUD0I7
TAUD0TTIN8	I	Channel 8 Input*1*2	TAUD0I8
TAUD0TTIN9	I	Channel 9 Input*1*2	TAUD0I9
TAUD0TTIN10	I	Channel 10 Input*1*2	TAUD0I10
TAUD0TTIN11	I	Channel 11 Input*1*2	TAUD0I11
TAUD0TTIN12	I	Channel 12 Input*1*2	TAUD0I12
TAUD0TTIN13	I	Channel 13 Input*1*2	TAUD0I13
TAUD0TTIN14	I	Channel 14 Input*1*2	TAUD0I14
TAUD0TTIN15	I	Channel 15 Input*1*2	TAUD0I15
TAUD0TTOUT0	O	Channel 0 output	TAUD0O0
TAUD0TTOUT1	O	Channel 1 output	TAUD0O1
TAUD0TTOUT2	O	Channel 2 output	TAUD0O2
TAUD0TTOUT3	O	Channel 3 output	TAUD0O3
TAUD0TTOUT4	O	Channel 4 output	TAUD0O4
TAUD0TTOUT5	O	Channel 5 output	TAUD0O5
TAUD0TTOUT6	O	Channel 6 output	TAUD0O6
TAUD0TTOUT7	O	Channel 7 output	TAUD0O7
TAUD0TTOUT8	O	Channel 8 output	TAUD0O8
TAUD0TTOUT9	O	Channel 9 output	TAUD0O9
TAUD0TTOUT10	O	Channel 10 output	TAUD0O10
TAUD0TTOUT11	O	Channel 11 output	TAUD0O11
TAUD0TTOUT12	O	Channel 12 output	TAUD0O12
TAUD0TTOUT13	O	Channel 13 output	TAUD0O13
TAUD0TTOUT14	O	Channel 14 output	TAUD0O14
TAUD0TTOUT15	O	Channel 15 output	TAUD0O15
TAUD1			
TAUD1TTIN0	I	Channel 0 Input*1*2	TAUD1I0
TAUD1TTIN1	I	Channel 1 Input*1*2	TAUD1I1
TAUD1TTIN2	I	Channel 2 Input*1*2	TAUD1I2
TAUD1TTIN3	I	Channel 3 Input*1*2	TAUD1I3
TAUD1TTIN4	I	Channel 4 Input*1*2	TAUD1I4
TAUD1TTIN5	I	Channel 5 Input*1*2	TAUD1I5

Table 33.8 External Input/Output Signals (2/3)

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
TAUD1TTIN6	I	Channel 6 Input*1*2	TAUD1I6
TAUD1TTIN7	I	Channel 7 Input*1*2	TAUD1I7
TAUD1TTIN8	I	Channel 8 Input*1*2	TAUD1I8
TAUD1TTIN9	I	Channel 9 Input*1*2	TAUD1I9
TAUD1TTIN10	I	Channel 10 Input*1*2	TAUD1I10
TAUD1TTIN11	I	Channel 11 Input*1*2	TAUD1I11
TAUD1TTIN12	I	Channel 12 Input*1*2	TAUD1I12
TAUD1TTIN13	I	Channel 13 Input*1*2	TAUD1I13
TAUD1TTIN14	I	Channel 14 Input*1*2	TAUD1I14
TAUD1TTIN15	I	Channel 15 Input*1*2	TAUD1I15
TAUD1TTOUT0	O	Channel 0 output	TAUD1O0
TAUD1TTOUT1	O	Channel 1 output	TAUD1O1
TAUD1TTOUT2	O	Channel 2 output	TAUD1O2
TAUD1TTOUT3	O	Channel 3 output	TAUD1O3
TAUD1TTOUT4	O	Channel 4 output	TAUD1O4
TAUD1TTOUT5	O	Channel 5 output	TAUD1O5
TAUD1TTOUT6	O	Channel 6 output	TAUD1O6
TAUD1TTOUT7	O	Channel 7 output	TAUD1O7
TAUD1TTOUT8	O	Channel 8 output	TAUD1O8
TAUD1TTOUT9	O	Channel 9 output	TAUD1O9
TAUD1TTOUT10	O	Channel 10 output	TAUD1O10
TAUD1TTOUT11	O	Channel 11 output	TAUD1O11
TAUD1TTOUT12	O	Channel 12 output	TAUD1O12
TAUD1TTOUT13	O	Channel 13 output	TAUD1O13
TAUD1TTOUT14	O	Channel 14 output	TAUD1O14
TAUD1TTOUT15	O	Channel 15 output	TAUD1O15
TAUD2			
TAUD2TTIN0	I	Channel 0 Input*1*3	TAUD2I0
TAUD2TTIN1	I	Channel 1 Input*1*3	TAUD2I1
TAUD2TTIN2	I	Channel 2 Input*1*3	TAUD2I2
TAUD2TTIN3	I	Channel 3 Input*1*3	TAUD2I3
TAUD2TTIN4	I	Channel 4 Input*1*3	TAUD2I4
TAUD2TTIN5	I	Channel 5 Input*1*3	TAUD2I5
TAUD2TTIN6	I	Channel 6 Input*1*3	TAUD2I6
TAUD2TTIN7	I	Channel 7 Input*1*3	TAUD2I7
TAUD2TTIN8	I	Channel 8 Input*1*3	TAUD2I8
TAUD2TTIN9	I	Channel 9 Input*1*3	TAUD2I9
TAUD2TTIN10	I	Channel 10 Input*1*3	TAUD2I10
TAUD2TTIN11	I	Channel 11 Input*1*3	TAUD2I11
TAUD2TTIN12	I	Channel 12 Input*1*3	TAUD2I12
TAUD2TTIN13	I	Channel 13 Input*1*3	TAUD2I13
TAUD2TTIN14	I	Channel 14 Input*1*3	TAUD2I14
TAUD2TTIN15	I	Channel 15 Input*1*3	TAUD2I15

Table 33.8 External Input/Output Signals (3/3)

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
TAUD2TTOUT0	O	Channel 0 output	TAUD2O0
TAUD2TTOUT1	O	Channel 1 output	TAUD2O1
TAUD2TTOUT2	O	Channel 2 output	TAUD2O2
TAUD2TTOUT3	O	Channel 3 output	TAUD2O3
TAUD2TTOUT4	O	Channel 4 output	TAUD2O4
TAUD2TTOUT5	O	Channel 5 output	TAUD2O5
TAUD2TTOUT6	O	Channel 6 output	TAUD2O6
TAUD2TTOUT7	O	Channel 7 output	TAUD2O7
TAUD2TTOUT8	O	Channel 8 output	TAUD2O8
TAUD2TTOUT9	O	Channel 9 output	TAUD2O9
TAUD2TTOUT10	O	Channel 10 output	TAUD2O10
TAUD2TTOUT11	O	Channel 11 output	TAUD2O11
TAUD2TTOUT12	O	Channel 12 output	TAUD2O12
TAUD2TTOUT13	O	Channel 13 output	TAUD2O13
TAUD2TTOUT14	O	Channel 14 output	TAUD2O14
TAUD2TTOUT15	O	Channel 15 output	TAUD2O15

Note 1. Setting of the noise filter for the port is required when the channel input pin is used. For details, **Section 2.7, Noise Filter & Edge/Level Detector**.

Note 2. The input signals can be switched by PIC. For details, see **Section 41.2.3.12, Timer Input Select Function**.

Note 3. The input signals can be switched by PIC. For details, see **Section 41.2.3.14, Timer Output Monitor Function**.

33.1.7 Internal Input/Output Signals

The internal input/output signals of TAUDn are listed below.

Table 33.9 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
TAUDnTSSTm*1	Simultaneous channel start trigger input	PIC
TAUDnTUDCm*1 (m = 0, 2, 8)	TAUD master up/down signal output	PIC

Note 1. n = 0, 1 only. TAUD2TSSTm, TAUD2TUDCm (m = 0, 2, 8) are not connected to PIC.

33.2 Overview

33.2.1 Functional Overview

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

33.2.2 Terms

In this section, the following terms are used.

Independent / synchronous channel operation

Independent or synchronous channel operation describes the dependency of channels on each other:

- If a channel operates independently of all other channels, this is called independent channel operation.
- If a channel operates depending on other channels, this is called synchronous channel operation.

Channel group

In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Operation mode

An operation mode can be selected for every channel m. The operation mode defines the basic operation and features of a channel.

In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

Channel output mode

The channel output mode defines the operation of TAUDnTTOUTm

- of a single channel (independent output operation) or
- of all channels in a channel group (synchronous output operation).

Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

Channel operation function

The channel operation function defines the complete function and all features

- of a single channel (independent channel operation) or
- of all channels in a channel group (synchronous channel operation).

Upper / lower channel

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

33.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 33.10 Functional List of TAUD Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 33.12**
Interval Timer Function	**Section 33.12.1**
TAUDnTTINm Input Interval Timer Function	**Section 33.12.2**
Clock Divide Function	**Section 33.12.3**
External Event Count Function	**Section 33.12.4**
Delay Count Function	**Section 33.12.5**
One-Pulse Output Function	**Section 33.12.6**
TAUDnTTINm Input Pulse Interval Measurement Function	**Section 33.12.7**
TAUDnTTINm Input Signal Width Measurement Function	**Section 33.12.8**
TAUDnTTINm Input Position Detection Function	**Section 33.12.9**
TAUDnTTINm Input Period Count Detection Function	**Section 33.12.10**
TAUDnTTINm Input Pulse Interval Judgment Function	**Section 33.12.11**
TAUDnTTINm Input Signal Width Judgment Function	**Section 33.12.12**
Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)	**Section 33.12.13**
Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)	**Section 33.12.14**
One-Phase PWM Output Function	**Section 33.12.15**
Independent Channel Real-Time Functions	**Section 33.13**
Real-Time Output Function Type 1	**Section 33.13.1**
Real-Time Output Function Type 2	**Section 33.13.2**
Independent Channel Simultaneous Rewrite Functions	**Section 33.14**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 33.14.1**
Simultaneous Rewrite Trigger Generation Function Type 2	**Section 33.14.2**
Synchronous Channel Operation Functions	**Section 33.15**
PWM Output Function	**Section 33.15.1**
One-Shot Pulse Output Function	**Section 33.15.2**
Trigger Start PWM Output Function	**Section 33.15.3**
Delay Pulse Output Function	**Section 33.15.4**
Offset Trigger Output Function	**Section 33.15.5**
A/D Conversion Trigger Output Function Type 1	**Section 33.15.6**
Triangle PWM Output Function	**Section 33.15.7**
Triangle PWM Output Function with Dead Time	**Section 33.15.8**
A/D Conversion Trigger Output Function Type 2	**Section 33.15.9**
Interrupt Request Signals Culling Function	**Section 33.15.10**
Synchronous Non-Complementary and Complementary Modulation Output Functions	**Section 33.16**
Non-Complementary Modulation Output Function Type 1	**Section 33.16.1**
Non-Complementary Modulation Output Function Type 2	**Section 33.16.2**
Complementary Modulation Output Function	**Section 33.16.3**

33.2.4 TAUD I/O and Interrupt Request Signals

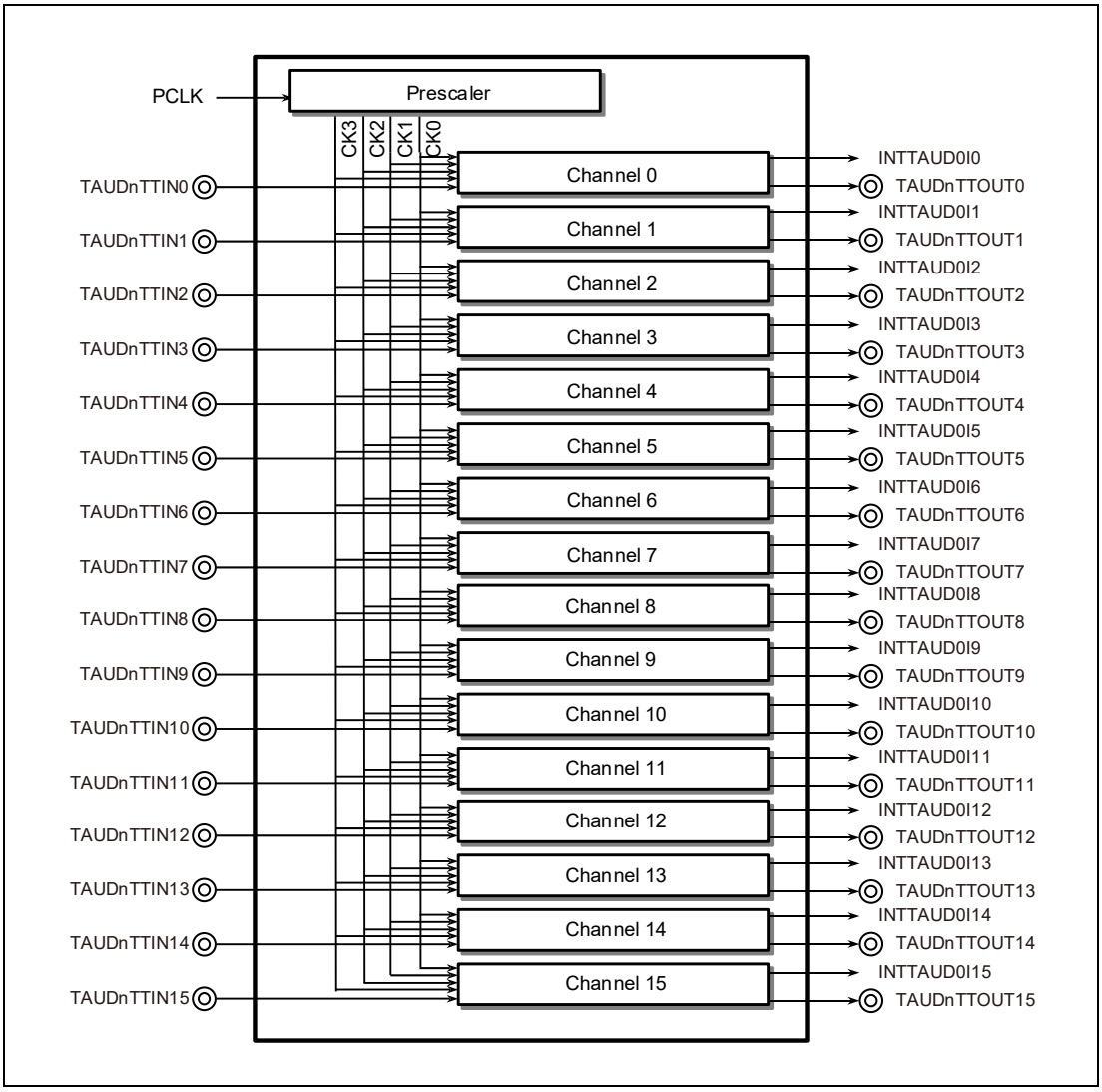


Figure 33.1 TAUD I/O and Interrupt Request Signals

33.2.5 Block Diagram

Figure 33.2, Block Diagram of the TAUD shows the main components of the TAUD.

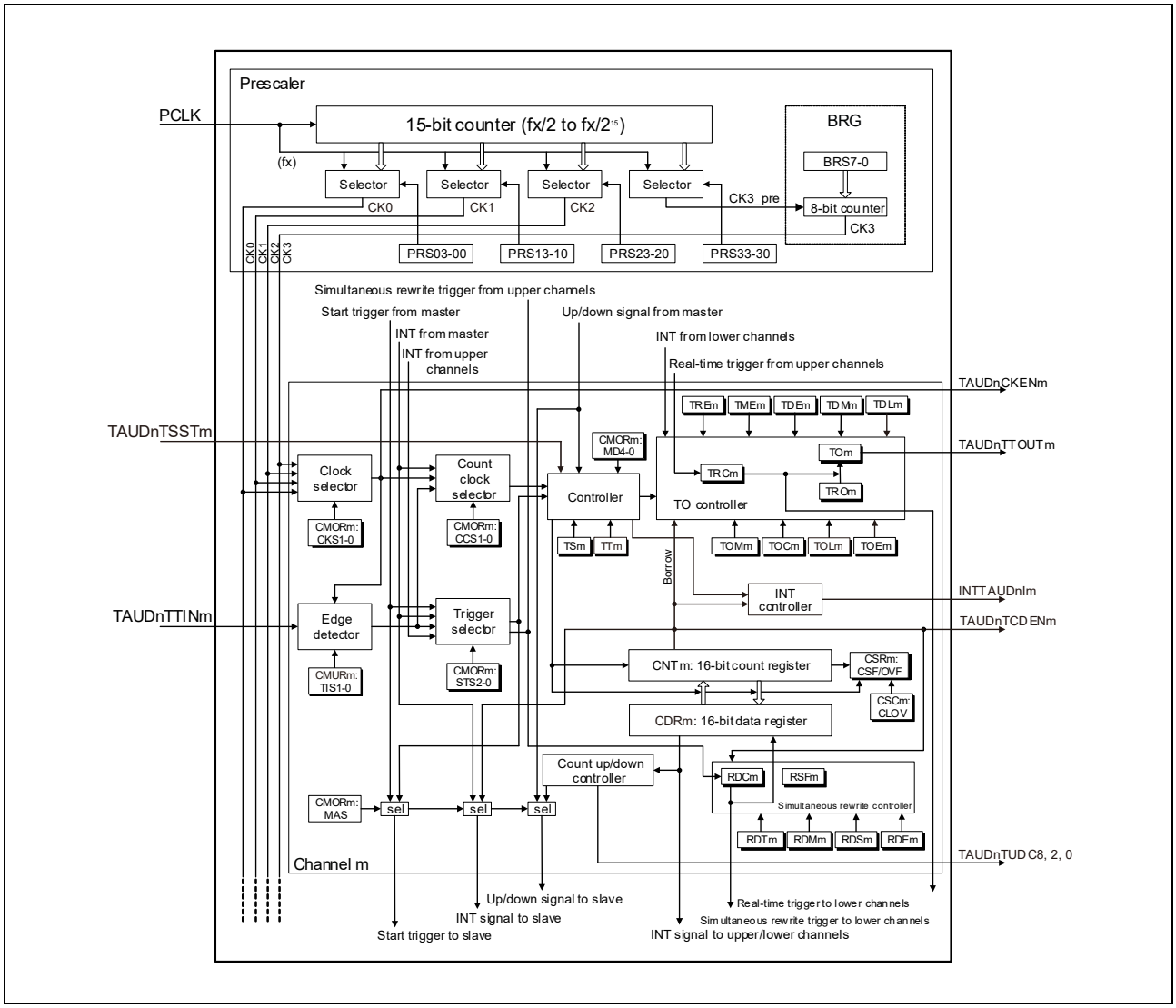


Figure 33.2 Block Diagram of the TAUD

The module name “TAUDn” has been omitted from the register names for the sake of clarity in the above figure.

33.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to four clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth count clock CK3 can be adjusted more precisely by using BRG to set an additional division factor that is not a power of 2.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUDnIm from master channel
- TAUDnTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDnCMORm.TAUDnMD[4:0])
- Counter start enable (TAUDnTS.TAUDnTSm) and counter stop (TAUDnTT.TAUDnTTm)

When counter start is enabled, status flag TAUDnTE.TAUDnTEm is set.

- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUDnTE.TAUDnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUDnTSSTm
- TAUDnTTINm input signal valid edge
- INTTAUDnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUDnTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a function that can be used in synchronous operating modes. The data registers (TAUDnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUDnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

33.3 Registers

33.3.1 List of Registers

TAUD registers are listed in the following table.

For details about <TAUDn_base>, see **Section 33.1.2, Register Base Addresses**.

Table 33.11 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
TAUDn Prescaler Registers					
TAUDn	TAUDn Prescaler Clock Select Register	TAUDnTPS	<TAUDn_base> + 240_H	*1	—
	TAUDn Prescaler Baud Rate Setting Register	TAUDnBRS	<TAUDn_base> + 244_H	*1	—
TAUDn Control Registers					
TAUDn	TAUDn Channel Data Register m	TAUDnCDRm	<TAUDn_base> + m × 4_H	*1	—
	TAUDn Channel Counter Register m	TAUDnCNTm	<TAUDn_base> + 80_H + m × 4_H	*1	—
	TAUDn Channel Mode OS Register m	TAUDnCMORm	<TAUDn_base> + 200_H + m × 4_H	*1	—
	TAUDn Channel Mode User Register m	TAUDnCMURm	<TAUDn_base> + $C0_H$ + m × 4_H	*1	—
	TAUDn Channel Status Register m	TAUDnCSRm	<TAUDn_base> + 140_H + m × 4_H	*1	—
	TAUDn Channel Status Clear Trigger Register m	TAUDnCSCm	<TAUDn_base> + 180_H + m × 4_H	*1	—
	TAUDn Channel Start Trigger Register	TAUDnTS	<TAUDn_base> + $1C4_H$	*1	—
	TAUDn Channel Enable Status Register	TAUDnTE	<TAUDn_base> + $1C0_H$	*1	—
	TAUDn Channel Stop Trigger Register	TAUDnTT	<TAUDn_base> + $1C8_H$	*1	—
TAUDn Output Registers					
TAUDn	TAUDn Channel Output Enable Register	TAUDnTOE	<TAUDn_base> + $5C_H$	*1	—
	TAUDn Channel Output Register	TAUDnTO	<TAUDn_base> + 58_H	*1	—
	TAUDn Channel Output Mode Register	TAUDnTOM	<TAUDn_base> + 248_H	*1	—
	TAUDn Channel Output Configuration Register	TAUDnTOC	<TAUDn_base> + $24C_H$	*1	—
	TAUDn Channel Output Active Level Register	TAUDnTOL	<TAUDn_base> + 040_H	*1	—
	TAUDn Channel Dead Time Output Enable Register	TAUDnTDE	<TAUDn_base> + 250_H	*1	—
	TAUDn Channel Dead Time Output Mode Register	TAUDnTDM	<TAUDn_base> + 254_H	*1	—
	TAUDn Channel Dead Time Output Level Register	TAUDnTDL	<TAUDn_base> + 54_H	*1	—
	TAUDn Channel Real-time Output Register	TAUDnTRO	<TAUDn_base> + $4C_H$	*1	—
	TAUDn Channel Real-time Output Enable Register	TAUDnTRE	<TAUDn_base> + 258_H	*1	—
	TAUDn Channel Real-time Output Control Register	TAUDnTRC	<TAUDn_base> + $25C_H$	*1	—
	TAUDn Channel Modulation Output Enable Register	TAUDnTME	<TAUDn_base> + 50_H	*1	—

Table 33.11 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
TAUDn Reload Data Registers					
TAUDn	TAUDn Channel Reload Data Enable Register	TAUDnRDE	<TAUDn_base> + 260_H	*1	—
	TAUDn Channel Reload Data Mode Register	TAUDnRDM	<TAUDn_base> + 264_H	*1	—
	TAUDn Channel Reload Data Control Channel Select Register	TAUDnRDS	<TAUDn_base> + 268_H	*1	—
	TAUDn Channel Reload Data Control Register	TAUDnRDC	<TAUDn_base> + $26C_H$	*1	—
	TAUDn Channel Reload Data Trigger Register	TAUDnRDT	<TAUDn_base> + 44_H	*1	—
	TAUDn Channel Reload Status Register	TAUDnRSF	<TAUDn_base> + 48_H	*1	—

Note 1. n = 0: PBG51#1
n = 1: PBG51#2
n = 2: PBG51#3

33.3.2 Details of TAUDn Prescaler Registers

33.3.2.1 TAUDnTPS — TAUDn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3_PRE by the factor specified in TAUDnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 240_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnPRS3[3:0]				TAUDnPRS2[3:0]				TAUDnPRS1[3:0]				TAUDnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.12 TAUDnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUDnPRS3 [3:0]	Specifies CK3_PRE clock. CK3_PRE clock is an input clock to BRG unit which supplies the CK3 operation clock to all channels.

TAUDnPRS3[3:0]	CK3_PRE Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 33.12 TAUDnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUDnPRS2 [3:0]	Specifies the CK2 clock. **TAUDnPRS2[3:0]** / **CK2 Clock** 0000_B / $PCLK/2^0$ 0001_B / $PCLK/2^1$ 0010_B / $PCLK/2^2$ 0011_B / $PCLK/2^3$ 0100_B / $PCLK/2^4$ 0101_B / $PCLK/2^5$ 0110_B / $PCLK/2^6$ 0111_B / $PCLK/2^7$ 1000_B / $PCLK/2^8$ 1001_B / $PCLK/2^9$ 1010_B / $PCLK/2^{10}$ 1011_B / $PCLK/2^{11}$ 1100_B / $PCLK/2^{12}$ 1101_B / $PCLK/2^{13}$ 1110_B / $PCLK/2^{14}$ 1111_B / $PCLK/2^{15}$ The above bits are rewritable only when all the counters using CK2 are stopped (TAUDnTE.TAUDnTEm = 0).
7 to 4	TAUDnPRS1 [3:0]	Specifies the CK1 clock. **TAUDnPRS1[3:0]** / **CK1 Clock** 0000_B / $PCLK/2^0$ 0001_B / $PCLK/2^1$ 0010_B / $PCLK/2^2$ 0011_B / $PCLK/2^3$ 0100_B / $PCLK/2^4$ 0101_B / $PCLK/2^5$ 0110_B / $PCLK/2^6$ 0111_B / $PCLK/2^7$ 1000_B / $PCLK/2^8$ 1001_B / $PCLK/2^9$ 1010_B / $PCLK/2^{10}$ 1011_B / $PCLK/2^{11}$ 1100_B / $PCLK/2^{12}$ 1101_B / $PCLK/2^{13}$ 1110_B / $PCLK/2^{14}$ 1111_B / $PCLK/2^{15}$ The above bits are rewritable only when all the counters using CK1 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 33.12 TAUDnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUDnPRS0 [3:0]	Specifies the CK0 clock.

TAUDnPRS0[3:0]	CK0 Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK0 are stopped (TAUDnTE.TAUDnTEm = 0).

NOTE

The TAUDn clock input PCLK is specified in the first part of this section, **Section 33.1.3, Clock Supply**.

33.3.2.2 TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUDnTPS.TAUDnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + 244_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUDnBRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.13 TAUDnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUDnBRS[7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUDnBRS[7:0]	CK3 Clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

33.3.3 Details of TAUDn Control Registers

33.3.3.1 TAUDnCDRm — TAUDn Channel Data Register m

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDnCMORm.TAUDnMD[4:1].

Access: This register can be read or written in 16-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUDn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.14 TAUDnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCDR [15:0]	Data register for capture/compare values

33.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register m

This is a channel m counter register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.15 TAUDnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCNT [15:0]	16-bit counter value

A read value depends on a counter value, a changed operating mode, TAUDnTS.TAUDnTSm or TAUDnTT.TAUDnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUDnTT.TAUDnTTm = 1)

Table 33.16, TAUDnCNTm Read Values after Re-Enabling Counter lists the initial counter read values after the counter is stopped (TAUDnTE.TAUDnTEm = 0) and re-enabled (TAUDnTS.TAUDnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUDnTS.TAUDnTSm = 1) with the counter waiting for a start trigger.

Table 33.16 TAUDnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUDnCNTm Value		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	$FFFF_H$	Stop value	—
Judge mode	Count down	$FFFF_H$	Stop value	—
Capture mode	Count up	0000_H	Stop value	—
Event count mode	Count down	$FFFF_H$	Stop value	—
One-count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000_H	Stop value	Capture value + 1 (TAUDnCDRm)
Judge and one-count mode	Count down	$FFFF_H$	Stop value	TAUDnCNTm value – 1
Count-up/-down mode	Count down/up	$FFFF_H$	Stop value	—
Pulse one-count mode	Count down	$FFFF_H$	Stop value	0000_H
Count capture mode	Count up	0000_H	Stop value	—
Gate count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000_H	Stop value	Stop value

Note 1. The value set for TAUDnCNTm when the operating mode is changed after a reset is deasserted.

33.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register m

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. Writable only when the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.17 TAUDnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Selects an operation clock. An operation clock is used for the TAUDnTTINm input edge detection circuit. Setting of TAUDnCMORm.TAUDnCCS[1:0] bits also allows the operation clock to serve as the TAUDnCNTm counter clock.

TAUDn CKS1	TAUDn CKS0	Selection of Operation Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

Bit Position	Bit Name	Function
13, 12	TAUDnCCS[1:0]	Selects a count clock for TAUDnCNTm counter.

TAUDn CCS1	TAUDn CCS0	Selection of Count Clock
0	0	Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]
0	1	Valid edge of TAUDnTTINm input signal
1	0	Setting prohibited
1	1	INTTAUDnIm signal of master channel

Bit Position	Bit Name	Function
11	TAUDnMAS	Specifies whether the channel is a master channel or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm_odd) are fixed to 0.

Table 33.17 TAUDnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUDnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUDn STS2</th><th>TAUDn STS1</th><th>TAUDn STS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUDnTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUDnTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Triggers simultaneous rewrite.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUDnIm is the start trigger of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUDnIm of upper channel (m - 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead time output signal of TAUDnTTOUTm generating unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of master channel</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUDnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUDnCDRm and overflow flag TAUDnCSRm.TAUDnOVF of channel m.
These bits are valid only when channel m is in capture mode or capture one-count mode.
<table>
<tr><th>TAUDn COS1</th><th>TAUDn COS0</th><th>TAUDnCDRm</th><th>TAUDnCSRm.TAUDnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUDnTTINm input.</td><td>Updated (cleared or set) upon detection of a TAUDnTTINm input valid edge:
• If a counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is set.
• If no counter overflow has occurred since the last valid edge detection, TAUDnCSRm.TAUDnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUDnTTINm input and at the occurrence of counter overflow:
• Detection of valid edge of TAUDnTTINm input: Counter value is written into TAUDnCDRm.
• Occurrence of overflow: $FFFF_H$ is loaded into TAUDnCDRm. The next detection of valid edge of TAUDnTTINm input is ignored.</td><td>Not set</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 33.17 TAUDnCMORm Register Contents (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUDnMD[4:0]</td><td>Specifies an operating mode.
<table>
<tr><th>TAUDn MD4</th><th>TAUDn MD3</th><th>TAUDn MD2</th><th>TAUDn MD1</th><th>TAUDn MD0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Judge mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Event count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Judge and one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Count-up/-down mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Pulse one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr>
<tr><td colspan="5">Other than above</td><td>Setting prohibited</td></tr>
</table>
<table>
<tr><th>Mode</th><th>Role of TAUDnMD0 Bit</th></tr>
<tr><td>Interval timer mode
Capture mode
Count capture mode</td><td>Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not.
0: INTTAUDnIm is not generated.
1: INTTAUDnIm is generated.</td></tr>
<tr><td>Event count mode
Count-up/-down mode</td><td>This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).</td></tr>
<tr><td>One-count mode
Pulse one-count mode</td><td>Enables/disables start trigger detection during counting.
0: Disables detection.
1: Enables detection.
CAUTION
• INTTAUDnIm signal is not output at the beginning of count operation in one-count mode.
• INTTAUDnIm signal is output at the beginning of count operation in pulse one-count mode.</td></tr>
<tr><td>Gate count mode</td><td>This bit should be set to 0 (disables start trigger detection during counting).</td></tr>
<tr><td>Capture and one-count mode
Capture and gate count mode</td><td>This bit should be set to 0.
CAUTION
INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td></tr>
<tr><td>Judge mode
Judge and one-count mode</td><td>Specifies INTTAUDnIm output timing.
0: When TAUDnCNTm ≤ TAUDnCDRm
1: When TAUDnCNTm > TAUDnCDRm</td></tr>
</table></td></tr>
</table>

33.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register m

This register specifies a type of valid edge detection used for TAUDnTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + $C0_H$ + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.18 TAUDnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	Specifies a valid edge of TAUDnTTINm input signal.

TAUDn TIS1	TAUDn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of rising and falling edges (selects low width measurement) • Start trigger: Falling edge • Stop trigger (capture): Rising edge
1	1	Detection of rising and falling edges (selects high width measurement) • Start trigger: Rising edge • Stop trigger (capture): Falling edge

Edge detection of TAUDnTTINm input signal is based on the operation clock selected by TAUDnCMORm.TAUDnCKS[1:0].

33.3.3.5 TAUDnCSRm — TAUDn Channel Status Register m

This register indicates the count direction and overflow status of channel m counter.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUDn_base> + 140_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnCSF	TAUDnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 33.19 TAUDnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUDnCSF	Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in the following mode: • Count-up/-down mode
0	TAUDnOVF	Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bit TAUDnCMORm.TAUDnCOS[1:0].

33.3.3.6 TAUDnCSCm — TAUDn Channel Status Clear Trigger Register m

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUDn_base> + 180_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUDnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 33.20 TAUDnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUDnCLOV	0: No function 1: Clears overflow flag TAUDnCSRm.TAUDnOVF.

33.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + $1C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTS15	TAUDnTS14	TAUDnTS13	TAUDnTS12	TAUDnTS11	TAUDnTS10	TAUDnTS09	TAUDnTS08	TAUDnTS07	TAUDnTS06	TAUDnTS05	TAUDnTS04	TAUDnTS03	TAUDnTS02	TAUDnTS01	TAUDnTS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 33.21 TAUDnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTSm	Enables the counter operation of channel m. 0: No function 1: Enables the counter operation and sets TAUDnTE.TAUDnTEm to 1. The counter operation is only enabled when TAUDnTE.TAUDnTEm is set to 1. Whether counting is started or not depends on a selected operating mode.

33.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register enables/disables a counter operation.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + $1C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTE15	TAUDnTE14	TAUDnTE13	TAUDnTE12	TAUDnTE11	TAUDnTE10	TAUDnTE09	TAUDnTE08	TAUDnTE07	TAUDnTE06	TAUDnTE05	TAUDnTE04	TAUDnTE03	TAUDnTE02	TAUDnTE01	TAUDnTE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.22 TAUDnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTEm	Enables/disables the counter operation of channel m. 0: Disables counter operation. 1: Enables counter operation. This bit is set to 1 when trigger input of TAUDnTSSTm (synchronous channel start trigger signal) is detected or when TAUDnTS.TAUDnTSm is set to 1. This bit is set to 0 when TAUDnTT.TAUDnTTm is set to 1.

33.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + $1C8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTT15	TAUDnTT14	TAUDnTT13	TAUDnTT12	TAUDnTT11	TAUDnTT10	TAUDnTT09	TAUDnTT08	TAUDnTT07	TAUDnTT06	TAUDnTT05	TAUDnTT04	TAUDnTT03	TAUDnTT02	TAUDnTT01	TAUDnTT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 33.23 TAUDnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTTm	Stops the counter operation of channel m. 0: No function 1: Stops the counter operation and resets TAUDnTE.TAUDnTEm. TAUDnCNTm, TAUDnTO.TAUDnTOm, and TAUDnTTOUTm retain the values provided before the counter is stopped.

33.3.4 Details of TAUDn Simultaneous Rewrite Registers

33.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDE15	TAUDn RDE14	TAUDn RDE13	TAUDn RDE12	TAUDn RDE11	TAUDn RDE10	TAUDn RDE09	TAUDn RDE08	TAUDn RDE07	TAUDn RDE06	TAUDn RDE05	TAUDn RDE04	TAUDn RDE03	TAUDn RDE02	TAUDn RDE01	TAUDn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.24 TAUDnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enables simultaneous rewrite

33.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 268_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDS15	TAUDn RDS14	TAUDn RDS13	TAUDn RDS12	TAUDn RDS11	TAUDn RDS10	TAUDn RDS09	TAUDn RDS08	TAUDn RDS07	TAUDn RDS06	TAUDn RDS05	TAUDn RDS04	TAUDn RDS03	TAUDn RDS02	TAUDn RDS01	TAUDn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.25 TAUDnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDSm	Selects a channel that controls a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

33.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 264_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDM15	TAUDn RDM14	TAUDn RDM13	TAUDn RDM12	TAUDn RDM11	TAUDn RDM10	TAUDn RDM09	TAUDn RDM08	TAUDn RDM07	TAUDn RDM06	TAUDn RDM05	TAUDn RDM04	TAUDn RDM03	TAUDn RDM02	TAUDn RDM01	TAUDn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.26 TAUDnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 0.

33.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnIm signal to trigger simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + $26C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDC15	TAUDn RDC14	TAUDn RDC13	TAUDn RDC12	TAUDn RDC11	TAUDn RDC10	TAUDn RDC09	TAUDn RDC08	TAUDn RDC07	TAUDn RDC06	TAUDn RDC05	TAUDn RDC04	TAUDn RDC03	TAUDn RDC02	TAUDn RDC01	TAUDn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.27 TAUDnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 1.

33.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDT15	TAUDn RDT14	TAUDn RDT13	TAUDn RDT12	TAUDn RDT11	TAUDn RDT10	TAUDn RDT09	TAUDn RDT08	TAUDn RDT07	TAUDn RDT06	TAUDn RDT05	TAUDn RDT04	TAUDn RDT03	TAUDn RDT02	TAUDn RDT01	TAUDn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 33.28 TAUDnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDTm	Triggers a simultaneous rewrite enabling state. 0: No function 1: The simultaneous rewrite enabling flag (TAUDnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger. These bits only apply when: • TAUDnRDE.TAUDnRDEm = 1

33.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RSF15	TAUDn RSF14	TAUDn RSF13	TAUDn RSF12	TAUDn RSF11	TAUDn RSF10	TAUDn RSF09	TAUDn RSF08	TAUDn RSF07	TAUDn RSF06	TAUDn RSF05	TAUDn RSF04	TAUDn RSF03	TAUDn RSF02	TAUDn RSF01	TAUDn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.29 TAUDnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRSFm	Indicates simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUDnRDTm = 1).

33.3.5 Details of TAUDn Output Registers

33.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + $5C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTOE15	TAUDnTOE14	TAUDnTOE13	TAUDnTOE12	TAUDnTOE11	TAUDnTOE10	TAUDnTOE09	TAUDnTOE08	TAUDnTOE07	TAUDnTOE06	TAUDnTOE05	TAUDnTOE04	TAUDnTOE03	TAUDnTOE02	TAUDnTOE01	TAUDnTOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.30 TAUDnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOEm	Enables or disables the independent channel output function. 0: Disables the independent timer output function (controlled by software). 1: Enables the independent timer output function.

33.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDnTTOUTm level.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTO15	TAUDnTO14	TAUDnTO13	TAUDnTO12	TAUDnTO11	TAUDnTO10	TAUDnTO09	TAUDnTO08	TAUDnTO07	TAUDnTO06	TAUDnTO05	TAUDnTO04	TAUDnTO03	TAUDnTO02	TAUDnTO01	TAUDnTO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.31 TAUDnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOm	Specifies and reads a TAUDnTTOUTm level. 0: Low level 1: High level Only TAUDnTOm bits for which Independent Channel Output function is disabled (TAUDnTOEm = 0) can be written.

33.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 248_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOM15	TAUDn TOM14	TAUDn TOM13	TAUDn TOM12	TAUDn TOM11	TAUDn TOM10	TAUDn TOM09	TAUDn TOM08	TAUDn TOM07	TAUDn TOM06	TAUDn TOM05	TAUDn TOM04	TAUDn TOM03	TAUDn TOM02	TAUDn TOM01	TAUDn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.32 TAUDnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

33.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOMm.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + $24C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOC15	TAUDn TOC14	TAUDn TOC13	TAUDn TOC12	TAUDn TOC11	TAUDn TOC10	TAUDn TOC09	TAUDn TOC08	TAUDn TOC07	TAUDn TOC06	TAUDn TOC05	TAUDn TOC04	TAUDn TOC03	TAUDn TOC02	TAUDn TOC01	TAUDn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.33 TAUDnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOCm	Specifies an output mode. 0: Operating mode 1 1: Operating mode 2 As listed below, the output mode depends on the setting of TAUDnTOM.TAUDnTOMm.

TAUDn TOMm	TAUDn TOCm	Functional Description
0	0	Toggle mode: Toggle operation is conducted when INTTAUDnIm occurs.
0	1	Set/reset mode: Set when INTTAUDnIm occurs at the beginning of count operation, and reset when INTTAUDnIm is caused by detection of a match between TAUDnCNTm and TAUDnCDRm.
1	0	Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels.
1	1	Synchronous channel operating mode 2: Set when INTTAUDnIm occurs in count-down status, and reset when INTTAUDnIm occurs in count-up status.

33.3.5.5 TAUDnTOL — TAUDn Channel Output Active Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOm).

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOL15	TAUDn TOL14	TAUDn TOL13	TAUDn TOL12	TAUDn TOL11	TAUDn TOL10	TAUDn TOL09	TAUDn TOL08	TAUDn TOL07	TAUDn TOL06	TAUDn TOL05	TAUDn TOL04	TAUDn TOL03	TAUDn TOL02	TAUDn TOL01	TAUDn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.34 TAUDnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOLm	Specifies the output logic of channel m output bit (TAUDnTO.TAUDnTOm). 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1 and independent channel output mode 1 with real-time output.

33.3.6 Details of TAUDn Dead Time Output Registers

33.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDE15	TAUDn TDE14	TAUDn TDE13	TAUDn TDE12	TAUDn TDE11	TAUDn TDE10	TAUDn TDE09	TAUDn TDE08	TAUDn TDE07	TAUDn TDE06	TAUDn TDE05	TAUDn TDE04	TAUDn TDE03	TAUDn TDE02	TAUDn TDE01	TAUDn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.35 TAUDnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDEm	Enables/disables the dead time control operation of channel m. 0: Disables dead time operation 1: Enables dead time operation. The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm = 1

33.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 254_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDM15	TAUDn TDM14	TAUDn TDM13	TAUDn TDM12	TAUDn TDM11	TAUDn TDM10	TAUDn TDM09	TAUDn TDM08	TAUDn TDM07	TAUDn TDM06	TAUDn TDM05	TAUDn TDM04	TAUDn TDM03	TAUDn TDM02	TAUDn TDM01	TAUDn TDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.36 TAUDnTDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDMm	Specifies the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even channel (duty dead time output). 1: When detecting the TIN input edge of a lower odd channel (one-phase dead time output). The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

33.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 54_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDL15	TAUDn TDL14	TAUDn TDL13	TAUDn TDL12	TAUDn TDL11	TAUDn TDL10	TAUDn TDL09	TAUDn TDL08	TAUDn TDL07	TAUDn TDL06	TAUDn TDL05	TAUDn TDL04	TAUDn TDL03	TAUDn TDL02	TAUDn TDL01	TAUDn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.37 TAUDnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDLm	Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

33.3.7 Details of TAUDn Real-time/Modulation Output Registers

33.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 258_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRE15	TAUDn TRE14	TAUDn TRE13	TAUDn TRE12	TAUDn TRE11	TAUDn TRE10	TAUDn TRE09	TAUDn TRE08	TAUDn TRE07	TAUDn TRE06	TAUDn TRE05	TAUDn TRE04	TAUDn TRE03	TAUDn TRE02	TAUDn TRE01	TAUDn TRE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.38 TAUDnTRE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTREm	Enables or disables real-time output of channel m. 0: Disables real-time output 1: Enables real-time output. These bit settings are applied only when TAUDnTOE.TAUDnTOEm = 1. When TAUDnTRE.TAUDnTREm = 0, TAUDnTTOUT.TAUDnTTOUTm is not affected by real-time output. When TAUDnTRE.TAUDnTREm = 1, TAUDnTTOUTm outputs the value of real-time output bit TAUDnTRO.TAUDnTROm in response to a timer operation.

33.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + $25C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TRC15	TAUDn TRC14	TAUDn TRC13	TAUDn TRC12	TAUDn TRC11	TAUDn TRC10	TAUDn TRC09	TAUDn TRC08	TAUDn TRC07	TAUDn TRC06	TAUDn TRC05	TAUDn TRC04	TAUDn TRC03	TAUDn TRC02	TAUDn TRC01	TAUDn TRC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.39 TAUDnTRC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTRCm	Specifies a channel on which the real-time output trigger for channel m is generated. 0: Next upper channel with this bit set to 1 1: Channel m These bit settings are applied only when TAUDnTRE.TAUDnTREm = 1.

33.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDnTTOUTm.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + $04C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTRO15	TAUDnTRO14	TAUDnTRO13	TAUDnTRO12	TAUDnTRO11	TAUDnTRO10	TAUDnTRO09	TAUDnTRO08	TAUDnTRO07	TAUDnTRO06	TAUDnTRO05	TAUDnTRO04	TAUDnTRO03	TAUDnTRO02	TAUDnTRO01	TAUDnTRO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.40 TAUDnTRO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTROm	Sets a value which is output to TAUDnTTOUTm. 0: Low 1: High TAUDnTROm value is not output to TAUDnTTOUTm when TAUDnTRE.TAUDnTREm = 0, even if a real-time output trigger occurs.

33.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 050_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTME15	TAUDnTME14	TAUDnTME13	TAUDnTME12	TAUDnTME11	TAUDnTME10	TAUDnTME09	TAUDnTME08	TAUDnTME07	TAUDnTME06	TAUDnTME05	TAUDnTME04	TAUDnTME03	TAUDnTME02	TAUDnTME01	TAUDnTME00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.41 TAUDnTME Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTMEm	Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation 1: Enables modulation These bit settings are applied only when TAUDnTOE.TAUDnTOEm and TAUDnTRE.TAUDnTREm = 1.

33.4 Operating Procedure

The following lists the general operation procedure for the TAUDn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUDnTTOUTm is also initialized and outputs a low level.

(1) Set the TAUDnTPS and TAUDnBRS registers to specify the clock frequency of CK0 to CK3.

(2) Configure the desired TAUDn function:
- Set the operation mode
- Set the channel output mode
- Set any other control bits

(3) Enable the counter by setting the TAUDnTS.TAUDnTSm bit to 1.
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.

(4) If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUDnTT.TAUDnTTm bit to 1. The counter can be forcibly restarted by setting the TAUDnTS.TAUDnTSm bit to 1.

(5) Stop the function by setting the TAUDnTT.TAUDnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 33.12, Independent Channel Operation Functions** and **Section 33.15, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

33.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). Several rules apply to the settings of channels. These rules are detailed in **Section 33.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 33.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 33.6, Simultaneous Rewrite**

33.5.1 Rules of Synchronous Channel Operation

Number of master and slave channels

- Only even channels (CH0, CH2, CH4, …) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the master channel and the synchronized slave channel. This is achieved by setting the TAUDnCMORm.TAUDnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 33.3, Grouping of Channels and Assignment of Count Clocks**.

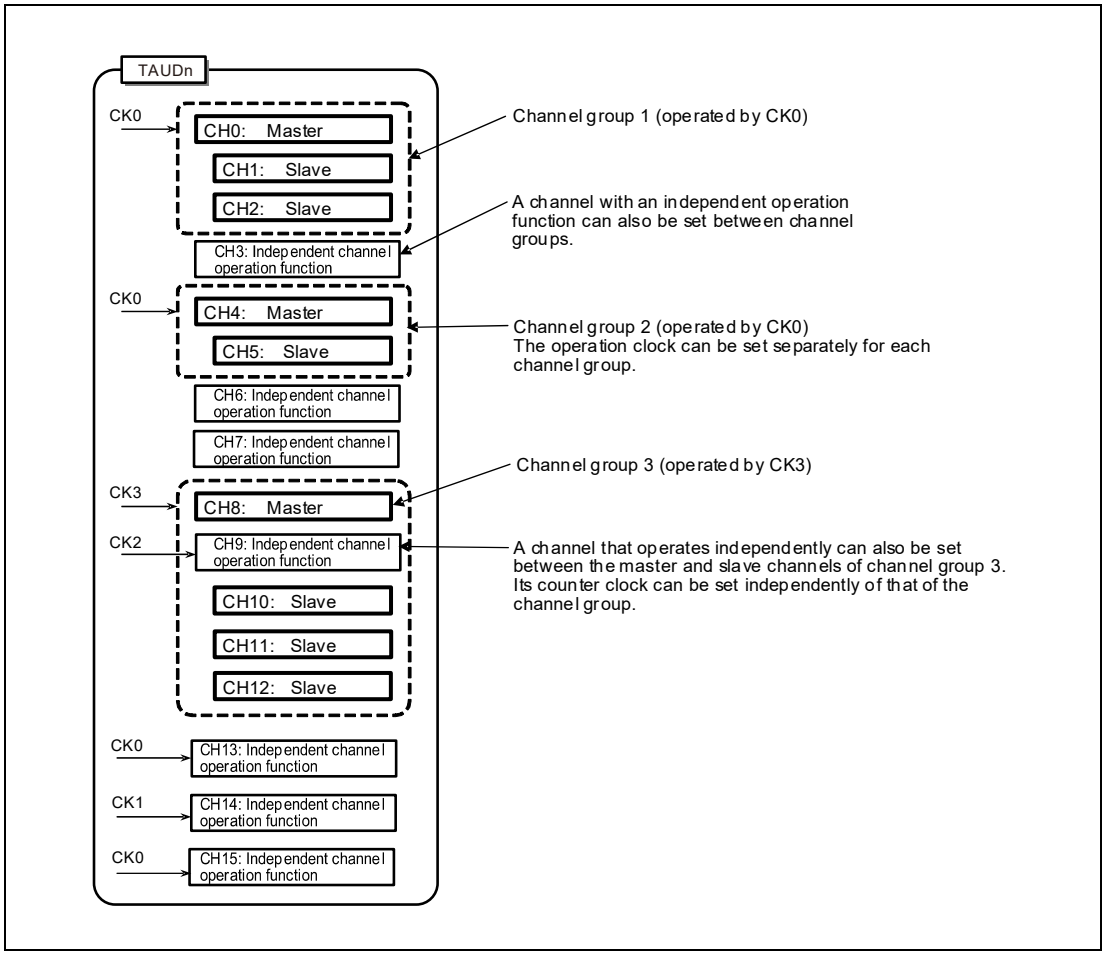


Figure 33.3 Grouping of Channels and Assignment of Count Clocks

Control trigger signal for master/slave channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals for their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

33.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

33.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDnTS.TAUDnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDnTT.TAUDnTTm bits of the channels should be set at the same time.

Setting to the TAUDnTS.TAUDnTSm bits to 1 also sets the corresponding TAUDnTE.TAUDnTEm bits to 1, enabling counting. The count start timing depends on operating mode.

33.5.2.2 Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to perform simultaneous start between the units, **41.2.3.1, Simultaneous Start Trigger Function**.

33.6 Simultaneous Rewrite

33.6.1 Overview of Operations

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUDnCDRm and TAUDnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDnIm being issued on the upper channel specified by TAUDnRDC.TAUDnRDCm

There are four methods for simultaneous rewrite. These are listed in **Table 33.42, Simultaneous Rewrite Methods and when They are Triggered**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 33.42 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUDnRDE. TAUDnRDEm	TAUDnRDS. TAUDnRDSm	TAUDnRDM. TAUDnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular cycle of the corresponding slave channel.	1	0	1
C1	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm	1	1	0/1
C2	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal	1	1	0/1

Table 33.43 lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 33.14, Independent Channel Simultaneous Rewrite Functions**, **Section 33.15, Synchronous Channel Operation Functions**, and **Section 33.16, Synchronous Non-Complementary and Complementary Modulation Output Functions**.

Table 33.43 Channel Functions and the Methods They Use for Simultaneous Rewrite

Function	A	B	C1	C2	TAUDnTOL. TAUDnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			√		
PWM Output Function	√		√		√
One-Shot Pulse Output Function	√				
Trigger Start PWM Output Function	√			√	
Delay Pulse Output Function	√				
Triangle PWM Output Function		√	√		√
Triangle PWM Output Function with Dead Time		√	√		
Interrupt Request Signals Culling Function	√	√	√		
AD Conversion Trigger Output Function Type 1	√		√		
AD Conversion Trigger Output Function Type 2		√	√		
Non-Complementary Modulation Output Function Type 1	√		√		
Non-Complementary Modulation Output Function Type 2		√	√		
Complementary Modulation Output Function		√	√		

Note: √: Available, (Blank): Unavailable

33.6.2 How to Control Simultaneous Rewrite

Figure 33.4, General Procedure for Simultaneous Rewrite shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

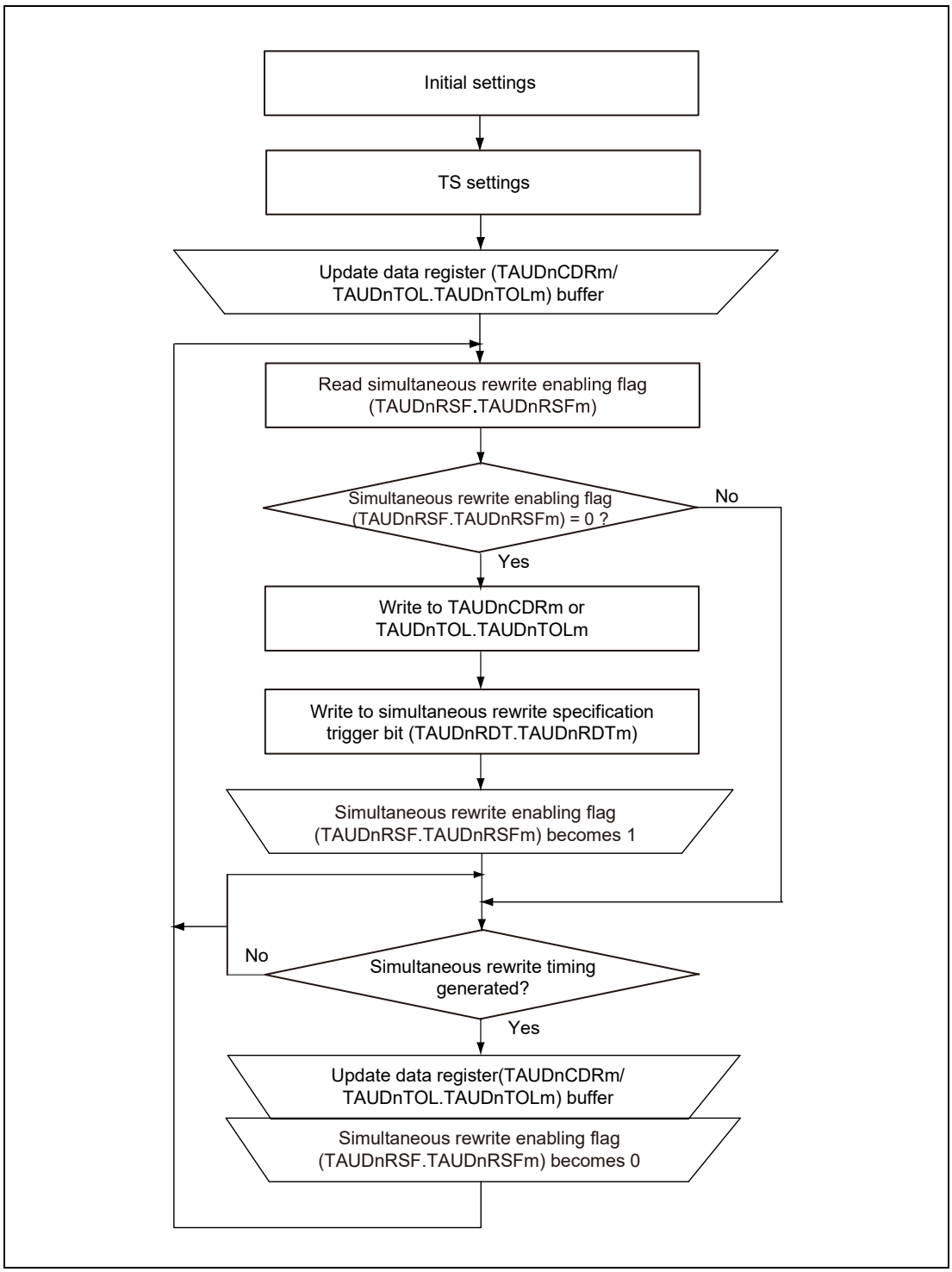


Figure 33.4 General Procedure for Simultaneous Rewrite

33.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUDnRDE.TAUDnRDEm = 1
- To select the type of simultaneous rewrite, set TAUDnRDM.TAUDnRDMm and TAUDnRDS.TAUDnRDSm according to the values listed in **Table 33.42, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using TAUDnRDC.TAUDnRDCm. (Prerequisite: TAUDnRDS.TAUDnRDSm has been set to the upper channel.)

33.6.2.2 Start Counter and Count Operation

- To start all the TAUDnCNTm counters of the channel group, set the corresponding TAUDnTS.TAUDnTSm bits to 1. The values of TAUDnTOL.TAUDnTOLm and the data registers (TAUDnCDRm) are loaded into the corresponding TAUDnTOL.TAUDnTOLm buffer (TAUDnTOL.TAUDnTOLm buf) and data buffer registers (TAUDnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1 sets the reload flag (TAUDnRSF.TAUDnRSFm) to 1, enabling simultaneous rewrite. TAUDnRSF.TAUDnRSFm remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the TAUDnRSF.TAUDnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

33.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- When simultaneous rewrite is complete, the TAUDnRSF.TAUDnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

33.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUDnRDE.TAUDnRDEm, TAUDnRDS.TAUDnRDSm, TAUDnRDM.TAUDnRDMm, and TAUDnRDC.TAUDnRDCm cannot be changed while the counter is in operation (TAUDnTE.TAUDnTEm = 1).
- TAUDnTOL.TAUDnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUDnTOL.TAUDnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUDnTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUDnRDS.TAUDnRDSm = 1), the TAUDnRDC.TAUDnRDCm bit controls all the lower channels. This means that if the TAUDnRDC.TAUDnRDCm bits of CH2 and CH7 are set to 1 and the TAUDnRDC.TAUDnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger (TAUDnRDE.TAUDnRDEm and TAUDnRDS.TAUDnRDSm = 1) but no upper channel is set (TAUDnRDC.TAUDnRDC[15:0] = 0), simultaneous rewrite cannot take place.

33.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

33.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

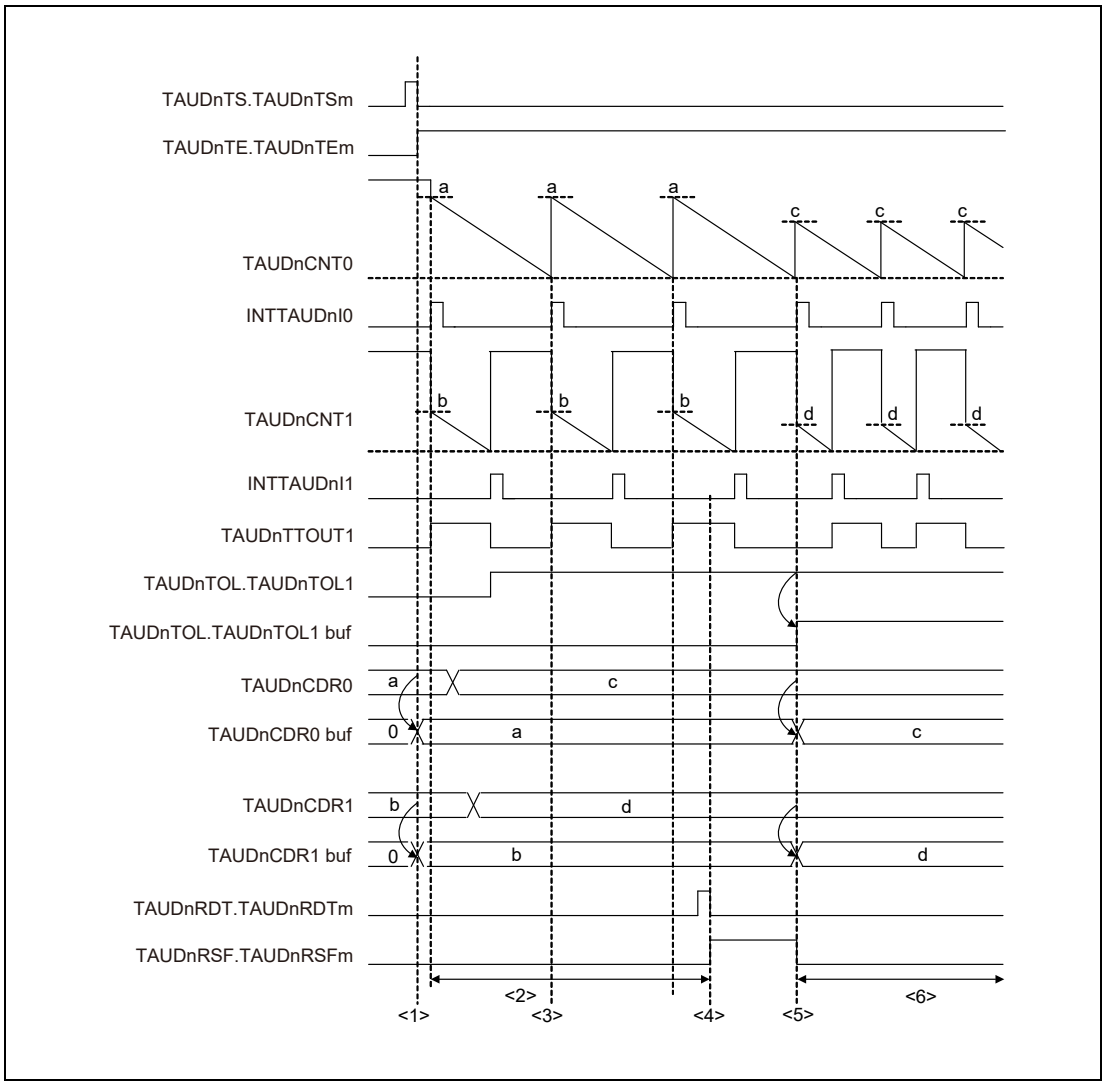


Figure 33.5 Simultaneous Rewrite when the Master Channel (Re)starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer and TAUDnTOL.TAUDnTOLm value is copied to the TAUDnTOL.TAUDnTOLm buffer.

(2) The TAUDnCDRm and TAUDnTOL.TAUDnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0)

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer and the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

33.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

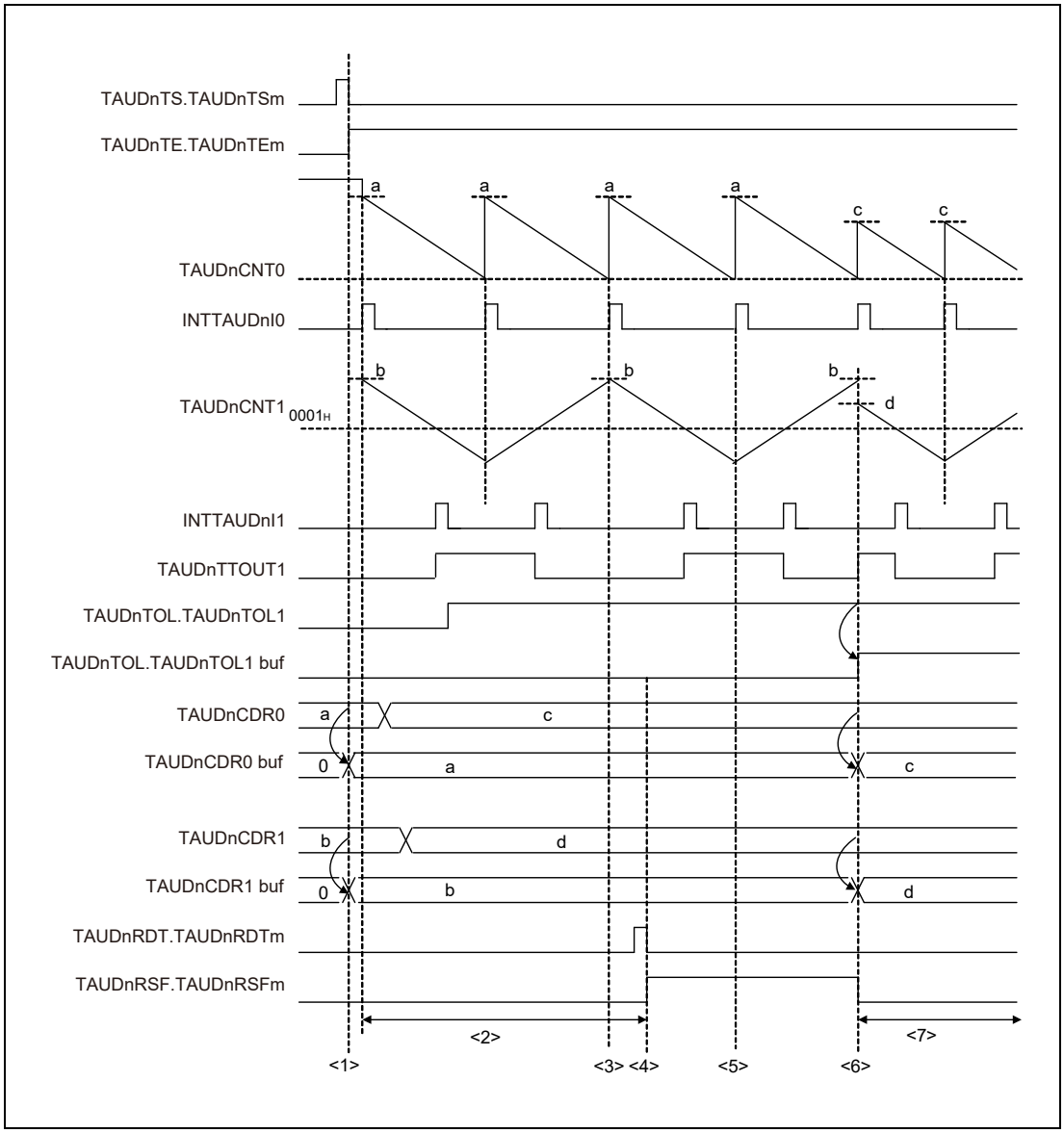


Figure 33.6 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

Setting:

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm and TAUDnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the top of the triangular cycle. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer, the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

33.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

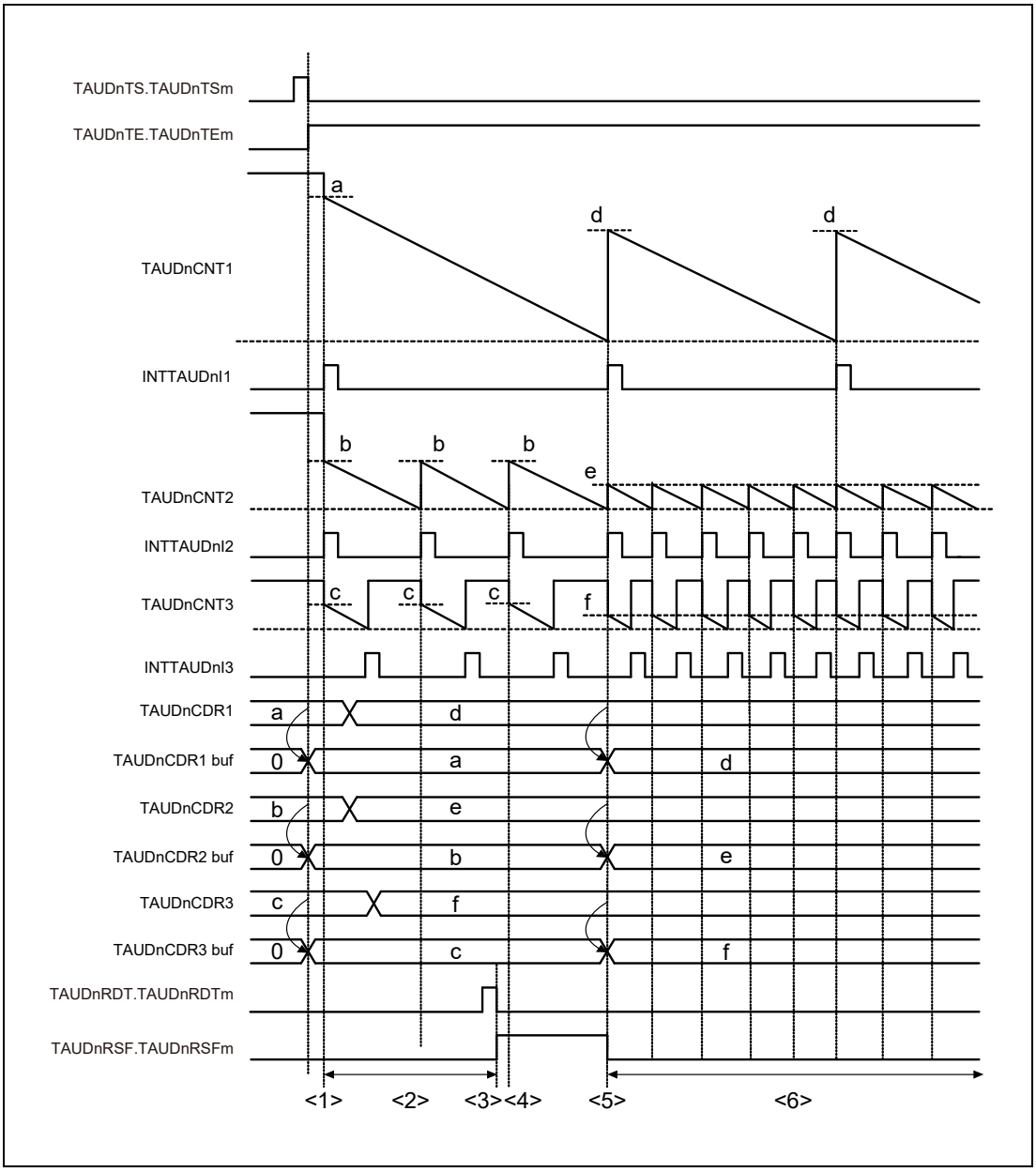


Figure 33.7 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

Setting:

CH1 is an upper channel which performs counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUDnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUDnCDRm values are loaded into the corresponding TAUDnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be rechanged.

33.6.4.4 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2)

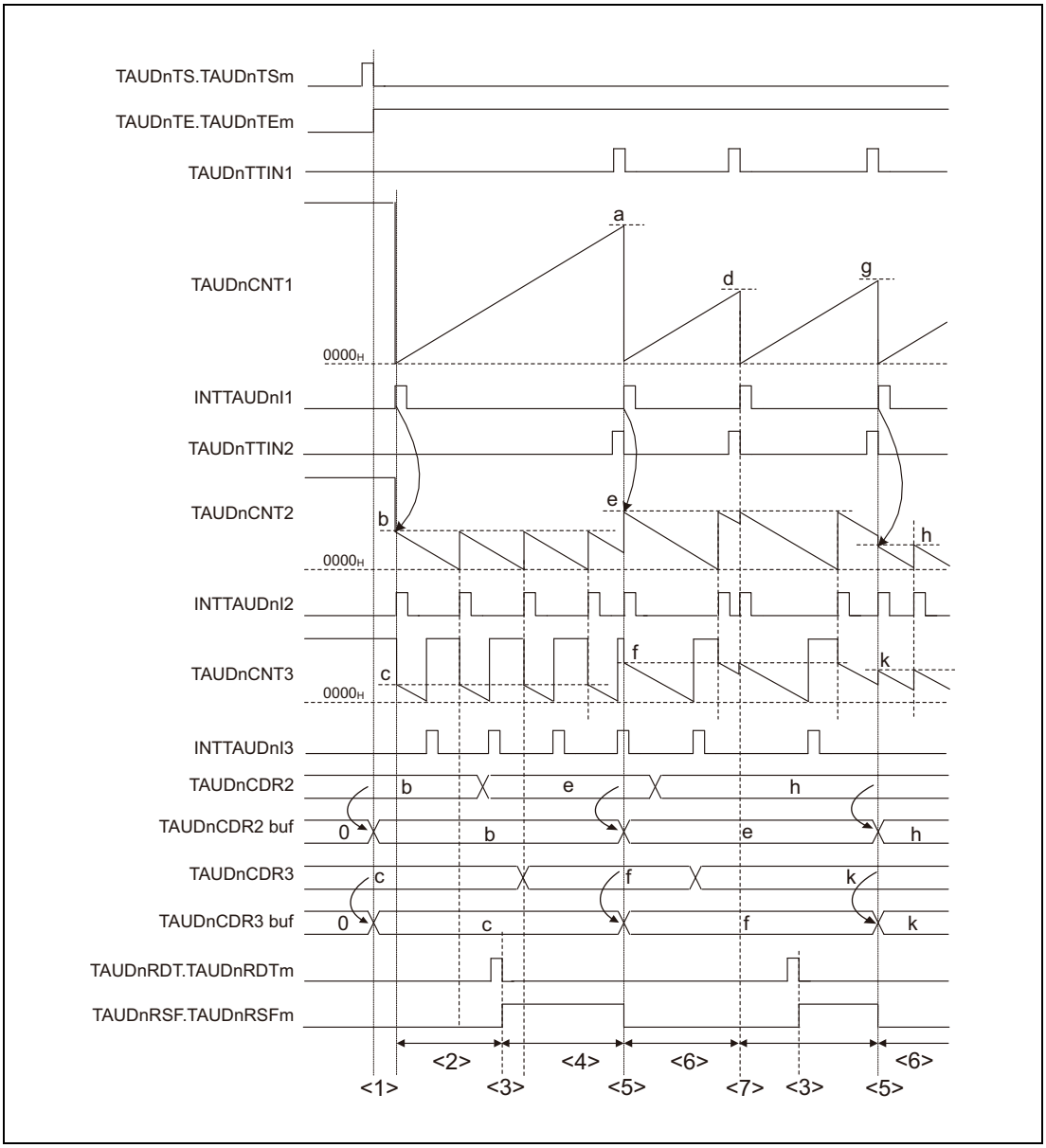


Figure 33.8 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal

Setting:

CH1 is an upper channel which performs counting up, CH2 is a master channel, and CH3 is the slave channel. The synchronous channel operation method C2 is applied. The TAUDnRDC register specifies which upper channel is monitored for an INTTAUDnIm trigger.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer. However, as TAUDnCDR1 operates in capture mode, TAUDnCDR1 value is not copied to the TAUDnCDR1 buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is caused by external signal TIN1. The TAUDnCDRm values are written to the corresponding TAUDnCDRm buffers.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.

(7) An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

33.7 Channel Output Modes

The output of the TAUDnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUDnTOE.TAUDnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUDnTO.TAUDnTOm) is sent to the output pin (TAUDnTTOUTm).
- By TAUD signals (TAUDnTOE.TAUDnTOEm = 1)
  When controlled by TAUD signals, the output level of TAUDnTTOUTm is set or reset or toggled by internal signals. The value of TAUDnTO.TAUDnTOm is updated accordingly to reflect the value of TAUDnTTOUTm.
  - Independently (TAUDnTOM.TAUDnTOMm = 0)
    In case of independent operation, the output of the TAUDnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUDnTOM.TAUDnTOMm = 0).
  - Synchronously (TAUDnTOM.TAUDnTOMm = 1)
    In case of synchronous operation, the output of the TAUDnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUDnTOM.TAUDnTOMm = 1).

The TAUDnTO.TAUDnTOm bit can always be read to determine the current value of TAUDnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 33.44, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 33.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 33.7.3, Channel Output Modes Controlled Synchronously by TAUDn Signals**

Batch operation of TAUDnTOm bit

Whether a set value is reflected to the TAUDnTOm bit or not is controlled by the TAUDnTOE.TAUDnTOEm bit.

The TAUDnTOm setting is written only to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 0 when a write to the TAUDnTO register is attempted. No TAUDnTOm setting is reflected to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 1.

NOTE

TAUDnTO.TAUDnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUDnTOL.TAUDnTOLm.

The value of TAUDnTOL.TAUDnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUDnTOL.TAUDnTOLm result in an undefined TAUDnTTOUTm signal output.

See **Section 33.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 33.44**.

Table 33.44 Channel Output Modes

Channel Output Mode	TAUDnTOE.TAUDnTOEm	TAUDnTOM.TAUDnTOMm	TAUDnTOC.TAUDnTOCm	TAUDnTDE.TAUDnTDEm	TAUDnTRE.TAUDnTREm	TAUDnTME.TAUDnTMEm	TAUDnTDM.TAUDnTDMm
By software							
Independent channel output mode controlled by software	0	X					
By TAUD signals, independently							
Independent channel output mode 1	1	0	0	0	0	0	0
with real-time output					1		
Independent channel output mode 2			1		0		
By TAUD signals, synchronously							
Synchronous channel output mode 1	1	1	0	0	0	0	0
with non-complementary modulation output					1	X	
Synchronous channel output mode 2			1	0	0	0	0
with dead time output				1			
with one-phase PWM output							1
with complementary modulation output					1	1	0
with non-complementary modulation output			1	0			

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTES

1. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1):
   - TAUDnTOM.TAUDnTOMm
   - TAUDnTOC.TAUDnTOCm
   - TAUDnTDE.TAUDnTDEm
   - TAUDnTRE.TAUDnTREm
   - TAUDnTDM.TAUDnTDMm
2. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1) except in channel output modes with modulation output:
   - TAUDnTME.TAUDnTMEm
   - TAUDnTDL.TAUDnTDLm

33.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUDnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUDnTOE.TAUDnTOEm = 0).

(1) Set TAUDnTO.TAUDnTOm to specify the initial level of the TAUDnTTOUTm output.

(2) Set channel output mode according to **Table 33.44, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.

(3) Start the counter (TAUDnTS.TAUDnTSm = 1).

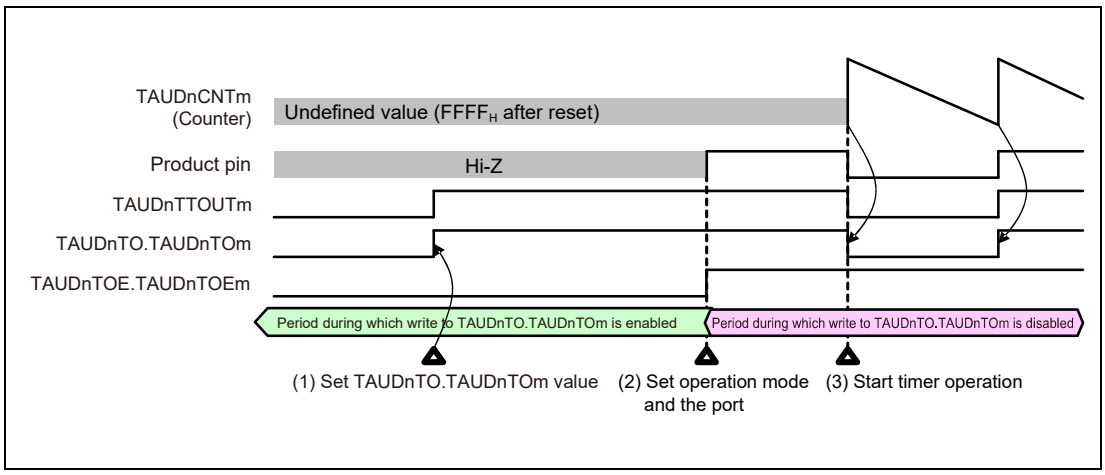


Figure 33.9 General Procedure for Specifying a TAUDnTTOUTm Channel Output Mode

33.7.2 Channel Output Modes Controlled Independently by TAUDn Signals

This section lists the channel output modes that are controlled independently by TAUDn signals. The control bits used to specify a mode are listed in **Table 33.44, Channel Output Modes**.

33.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUDnTTOUTm toggles when INTTAUDnIm is detected. The value of TAUDnTOL.TAUDnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 33.44, Channel Output Modes**.

33.7.2.2 Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDnTRO.TAUDnTROm bit of the trigger channel is output to TAUDnTTOUTm. The trigger channel is specified by setting the corresponding TAUDnTRC.TAUDnTRCm bit to 1. It controls all lower channels for which TAUDnTRC.TAUDnTRCm = 0.

Set/reset conditions

The value of TAUDnTRO.TAUDnTROm bit is sent to TAUDnTTOUTm only when an INTTAUDnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUDnTTINm input edge/counter start

The type of trigger is set using the TAUDnCMORm.TAUDnMD[4:1] bits.

Prerequisites

Both the master and slave channels can be set as a trigger generation channel. A channel for which TAUDnTRC.TAUDnTRCm is set to 1 serves as a trigger generation channel regardless of the value of TAUDnTRE.TAUDnTREm.

If there is no channel for which TAUDnTRC.TAUDnTRCm is set to 1 or if TAUDnTRC.TAUDnTRC0 = 0, real-time output cannot take place.

This can be seen in **Figure 33.10**.

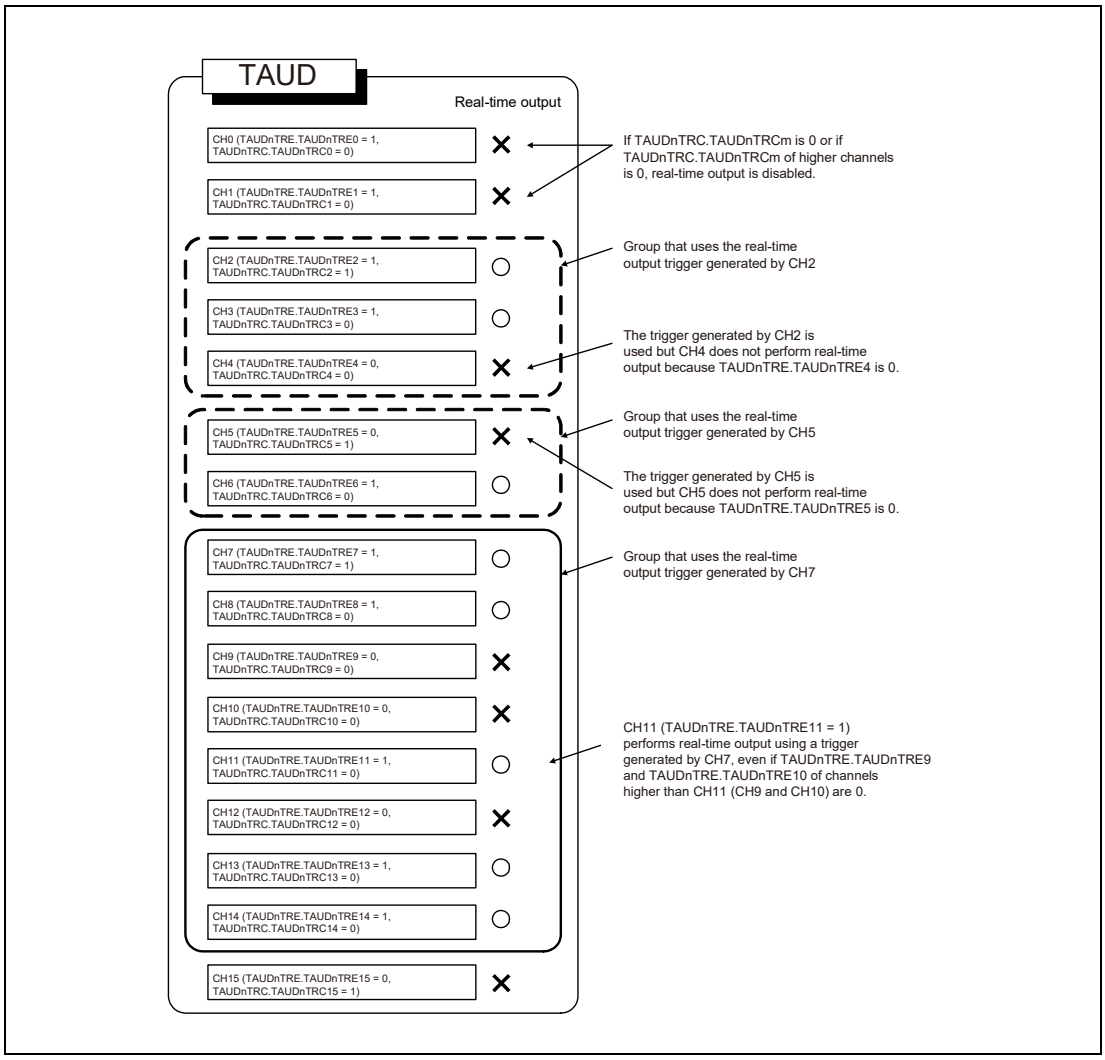


Figure 33.10 Real-Time Output

33.7.2.3 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUDnTTOUTm is set when INTTAUDnIm occurs at the time of count start, and reset when INTTAUDnIm occurs due to a match between TAUDnCNTm and TAUDnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 33.44, Channel Output Modes**.

33.7.3 Channel Output Modes Controlled Synchronously by TAUDn Signals

This section lists the channel output modes that are controlled synchronously by TAUDn signals. The control bits used to specify a mode are listed in **Table 33.44, Channel Output Modes**.

33.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUDnIm of master channel serves as a set signal and INTTAUDnIm of the slave channel as a reset signal. If INTTAUDnIm of master channel and INTTAUDnIm of the slave channel are generated at the same time, INTTAUDnIm of the slave channel (reset signal) has priority over INTTAUDnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 33.44, Channel Output Modes**.

33.7.3.2 Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDnTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUDnTRO.TAUDnTROm) of a channel.

The phase period to which the dead time is added is specified using the TAUDnTDL.TAUDnTDLm bit; for positive phase set TAUDnTDL.TAUDnTDLm = 0 and for negative phase set TAUDnTDL.TAUDnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed, its new value is applied upon detection of INTTAUDnIm on the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on the master channel.

33.7.3.3 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM output at TAUDnTTOUTm. For details, see **Section 33.15.7, Triangle PWM Output Function**.

Set/reset conditions

TAUDnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUDnTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUDnTTOUTm should be set to 0 before the function starts.

33.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDnTTOUTm. The set/reset conditions are shown in **Figure 33.11**.

Set/reset conditions

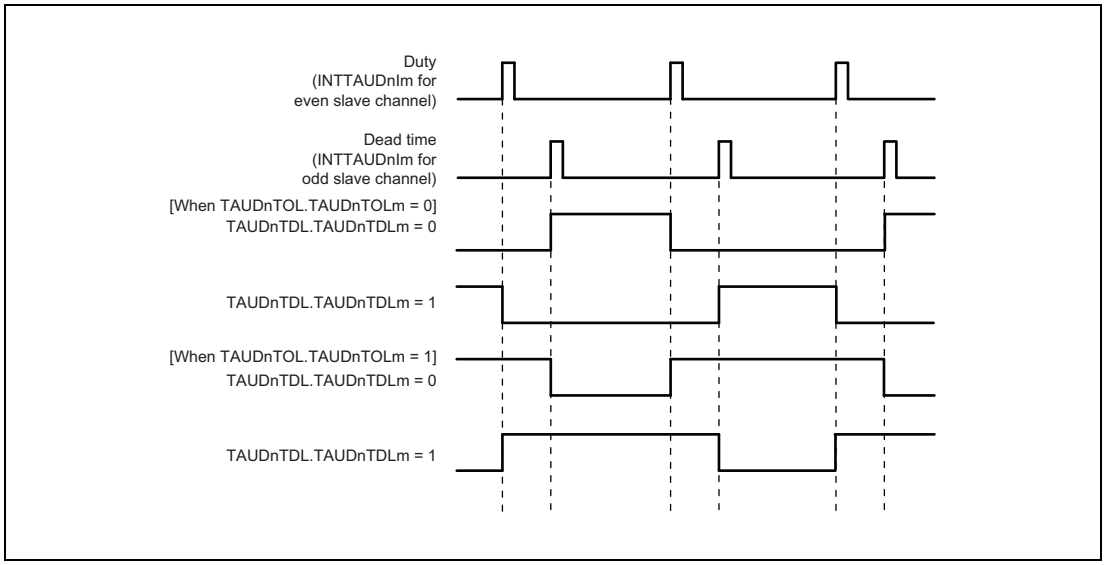


Figure 33.11 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

33.7.3.5 Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUDnTTOUTm. The set/reset conditions are shown in **Figure 33.12**.

Set/reset conditions

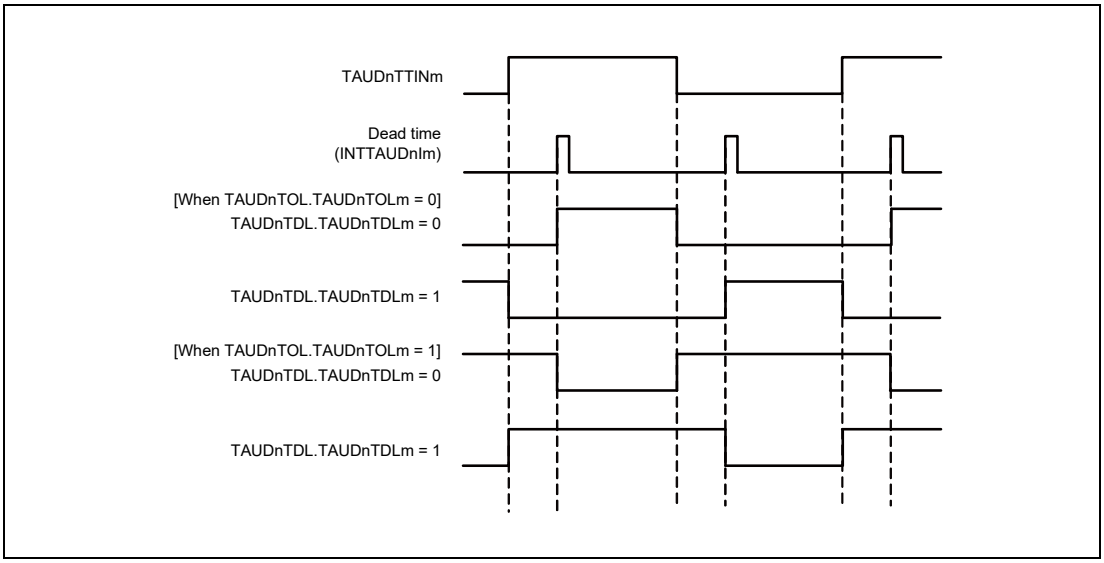


Figure 33.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

One-phase PWM output control requires a set of two channels:

- One even slave channel
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

33.7.3.6 Synchronous Channel Output Mode 2 with Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDnTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUDnTRO.TAUDnTROm), the modulation output bit (TAUDnTME.TAUDnTMEm), and the output level bit (TAUDnTOL.TAUDnTOLm) of a pair of slave channels.

For details, see **Section 33.16.3, Complementary Modulation Output Function**.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, four more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed during operation, its new value is applied upon detection of INTTAUDnIm at the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on an even slave channel.

33.7.3.7 Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a square wave while mode 2 has a triangular wave.

33.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUDnTS.TAUDnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

33.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUDnTS.TAUDnTSm is set to 1. The value of data register is also loaded when the counter starts.

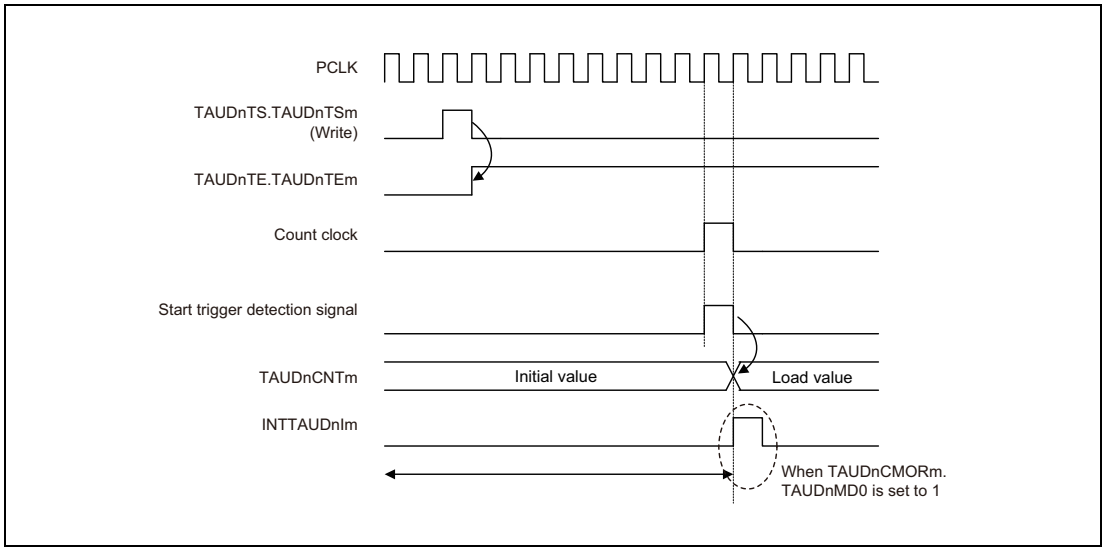


Figure 33.13 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

NOTE

Make sure to set TAUDnCMORm.TAUDnMD0 to 0 when using the count-up/-down mode.

33.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDnTS.TAUDnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

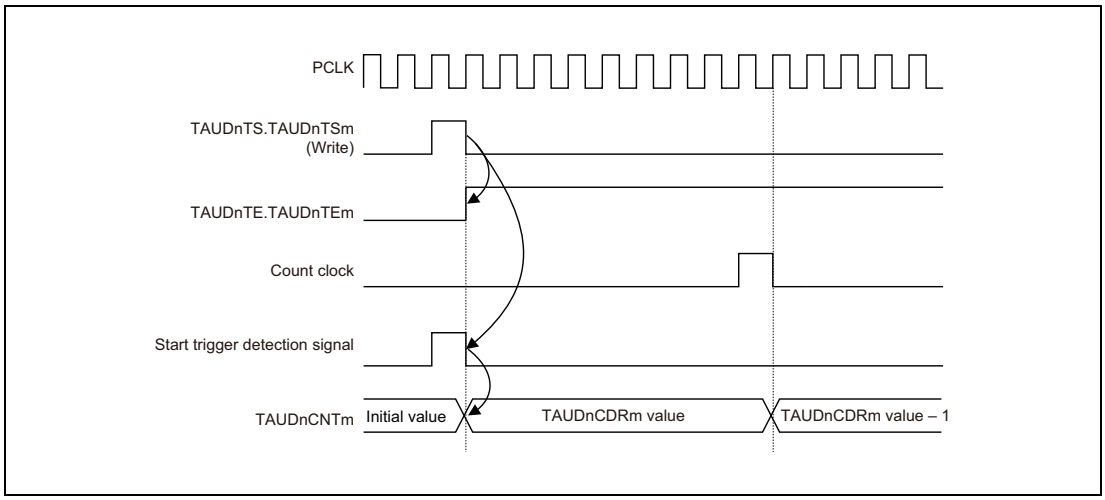


Figure 33.14 Start Timing in Event Count Mode

33.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUDnTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

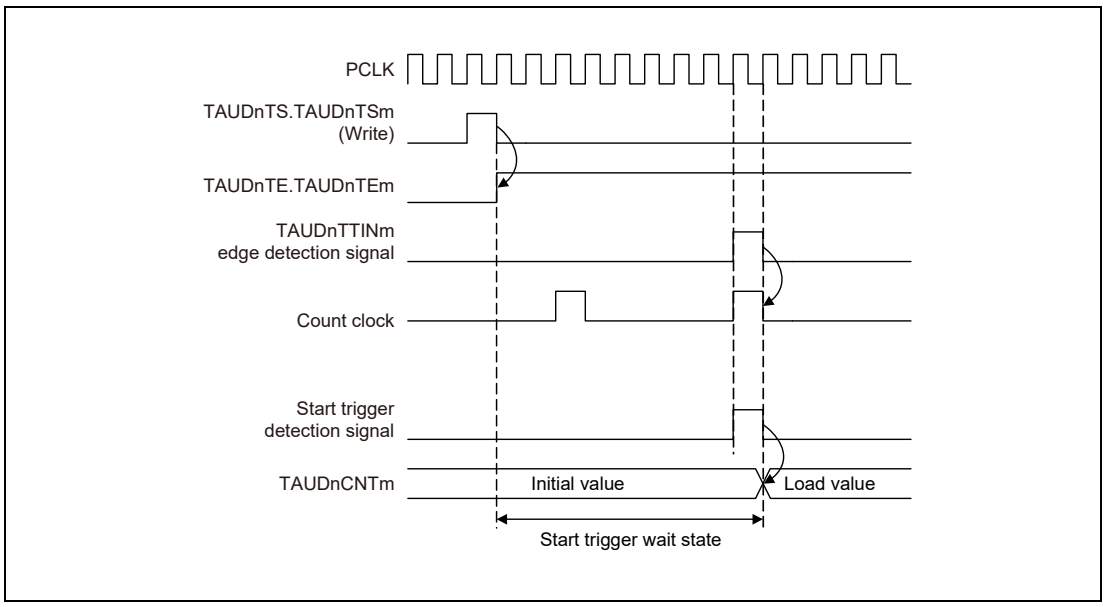


Figure 33.15 Count Start Timing in Other Operating Modes

33.9 TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDnIm is generated using the TAUDnCMORm.TAUDnMD0 bit. The generation of INTTAUDnIm when the TAUDnCMORm.TAUDnMD0 bit starts counting and the effect to TAUDnTTOUTm depend on the selected function. For details, refer to the description of TAUDnCMORm.TAUDnMD0 of each function.

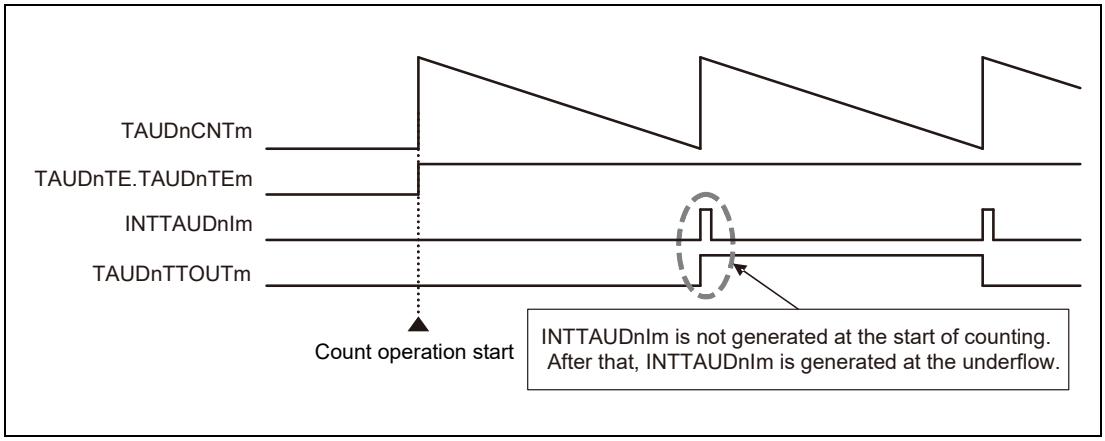


Figure 33.16 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 0)

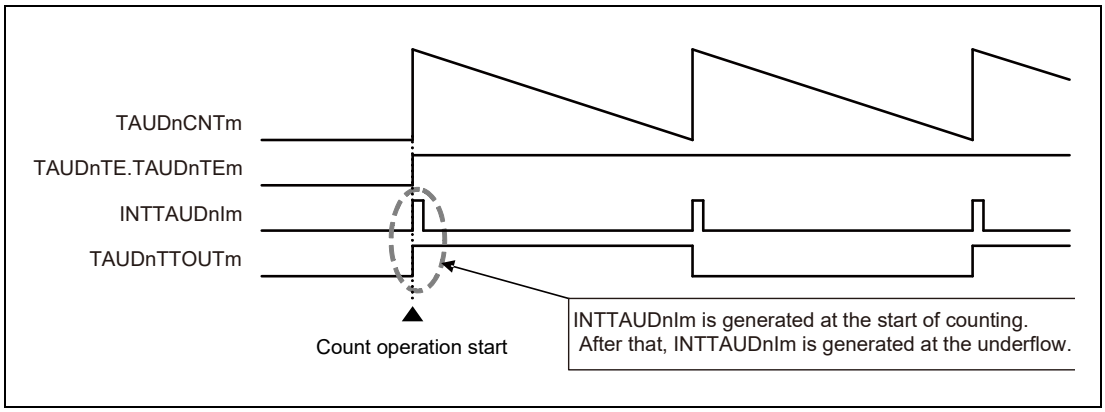


Figure 33.17 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 1)

33.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUDnCNTm = $FFFF_H$).
- Set TAUDnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUDnTTINm input.
- The trigger detection settings (TAUDnCMORm.TAUDnSTS[2:0] and TAUDnCMURm.TAUDnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUDnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

33.10.1 Combination of the TAUDnTTINm Input Pulse Interval Measurement Function and the TAUDnTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUDnTTINm of both channels, INTTAUDnIm of the TAUDnTTINm input interval timer function can detect the overflow when TAUDnCNTm of the TAUDnTTINm input pulse interval measurement function exceeds $FFFF_H$.

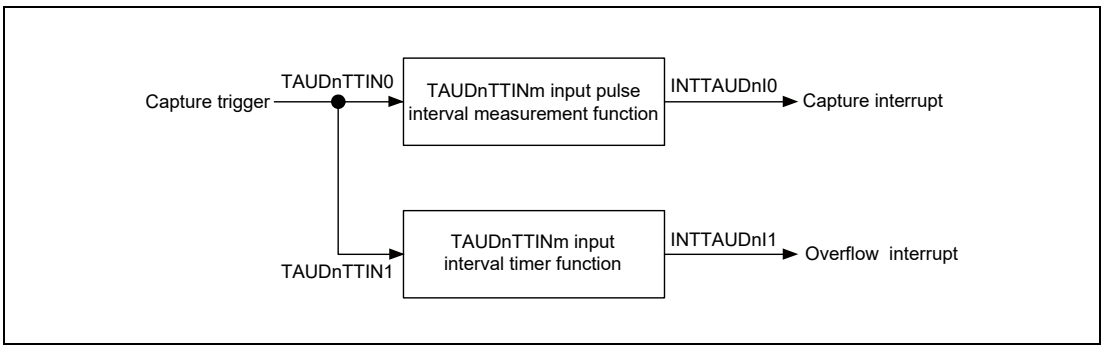


Figure 33.18 Combination of the TAUDnTTINm Input Pulse Interval Measurement Function and the TAUDnTTINm Input Interval Timer Function

Timing diagram

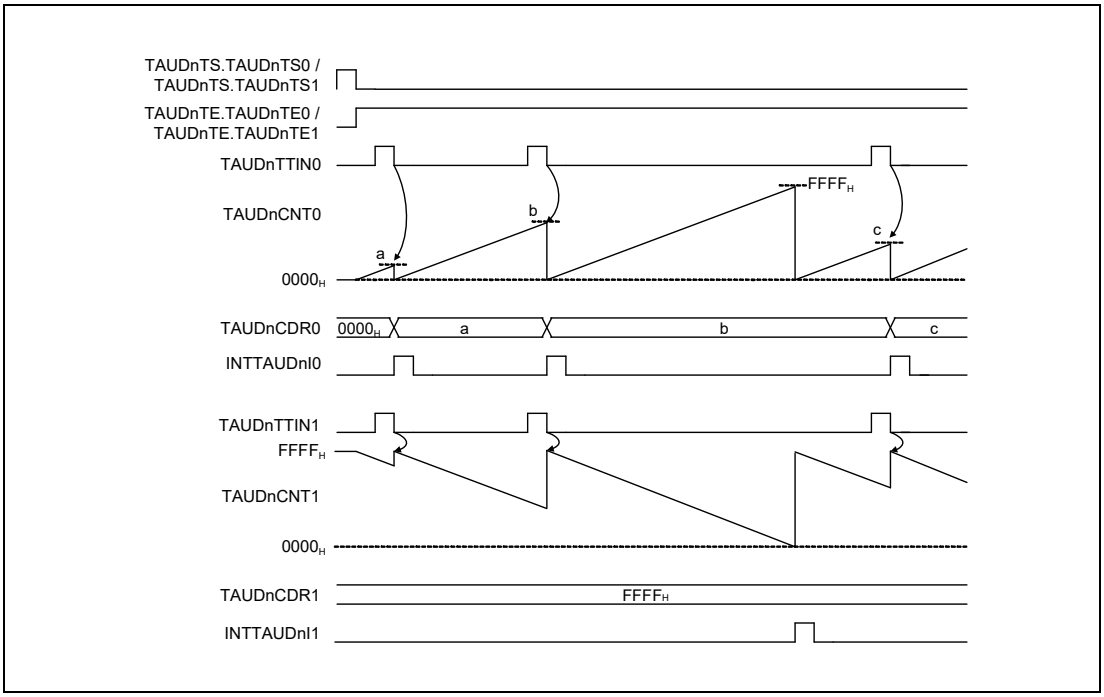


Figure 33.19 Interrupt Generation via Combination of the TAUDnTTINm Input Pulse Interval Measurement Function and the TAUDnTTINm Input Interval Timer Function

33.10.2 Combination of the TAUDnTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDnTTINm Width)

When the capture trigger is input simultaneously to TAUDnTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at measuring the TAUDnTTINm width) can detect the overflow when TAUDnCNTm of the TAUDnTTINm input signal width measurement function exceeds $FFFF_H$.

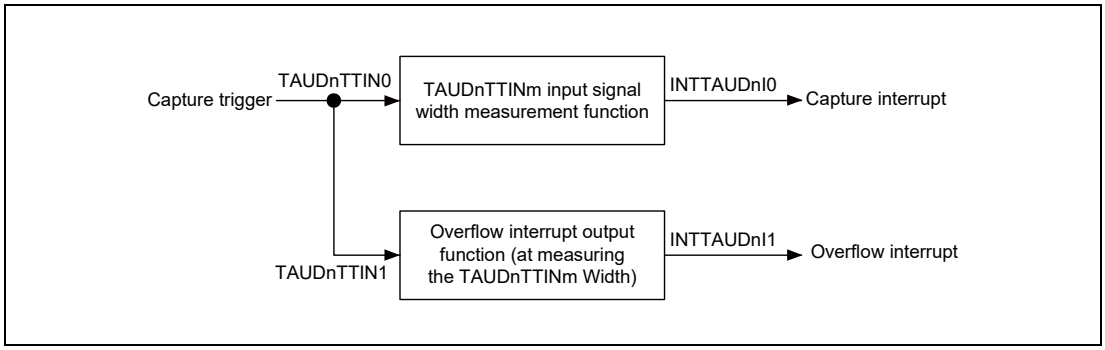


Figure 33.20 Combination of the TAUDnTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDnTTINm Width)

Timing diagram

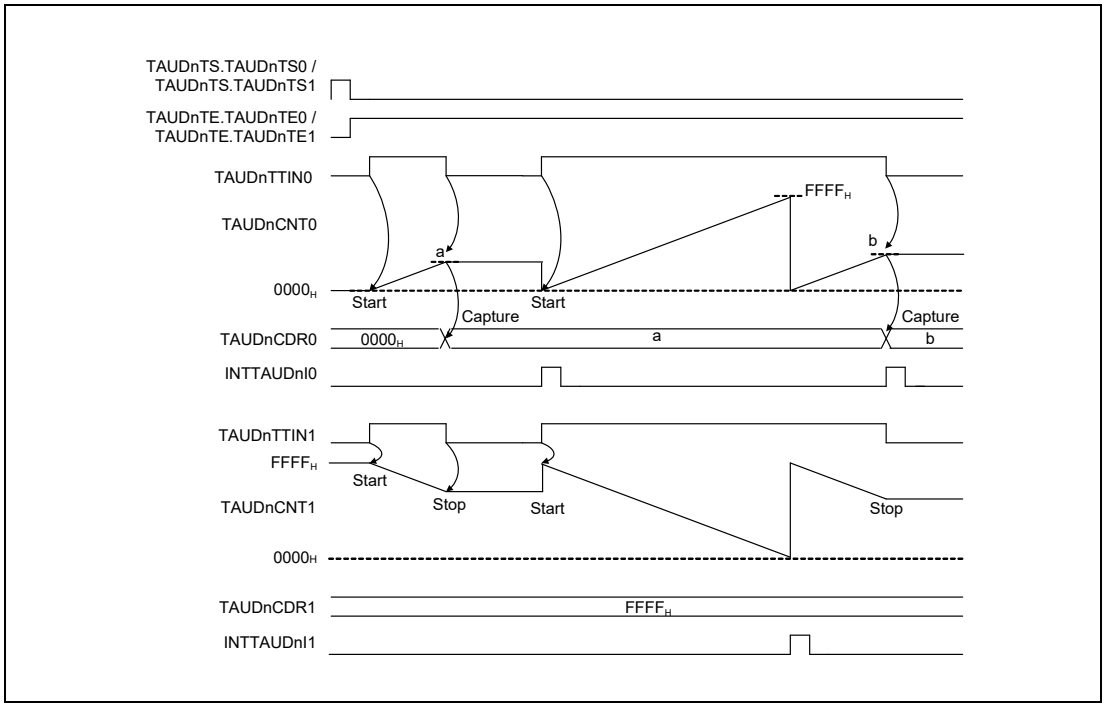


Figure 33.21 Interrupt Generation via Combination of the TAUDnTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDnTTINm Width)

33.10.3 Combination of the TAUDnTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUDnIm of the interval timer function can detect the overflow when TAUDnCNTm of the TAUDnTTINm input position detection function exceeds $FFFF_H$.

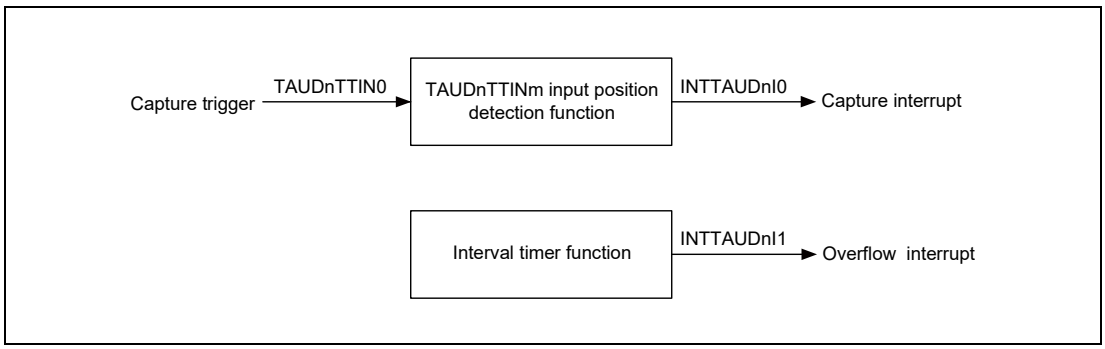


Figure 33.22 Combination of the TAUDnTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

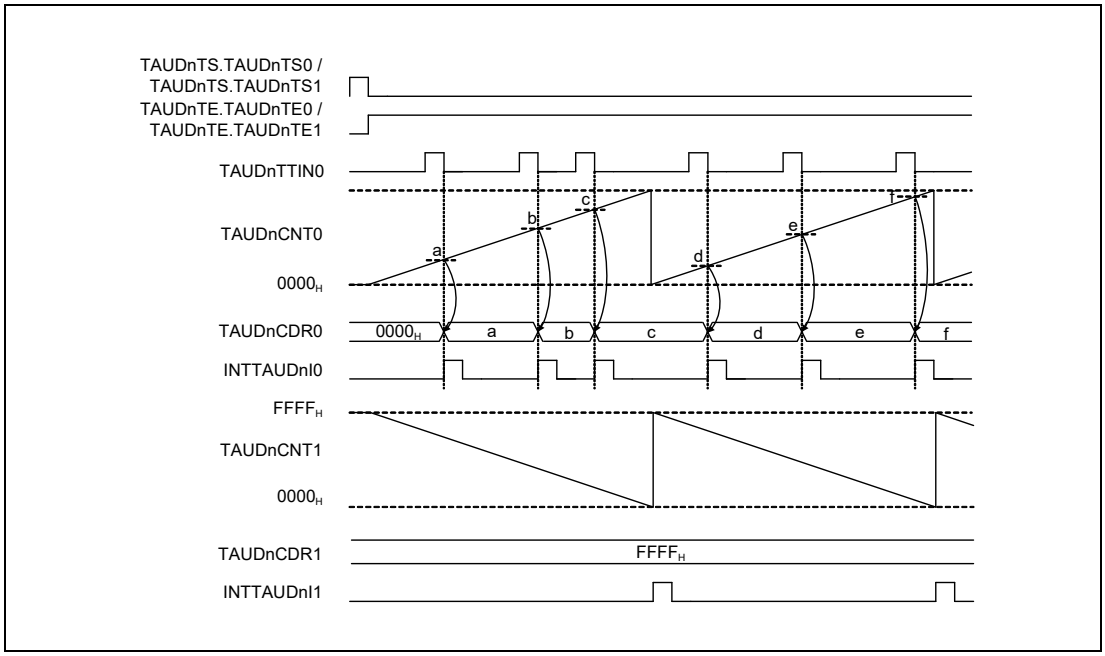


Figure 33.23 Interrupt Generation via Combination of the TAUDnTTINm Input Position Detection Function and the Interval Timer Function

33.10.4 Combination of the TAUDnTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDnTTINm Input Period Count)

When the capture trigger is input simultaneously to TAUDnTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at detecting the TAUDnTTINm input period count) can detect the overflow when TAUDnCNTm of the TAUDnTTINm input period count detection function exceeds $FFFF_H$.

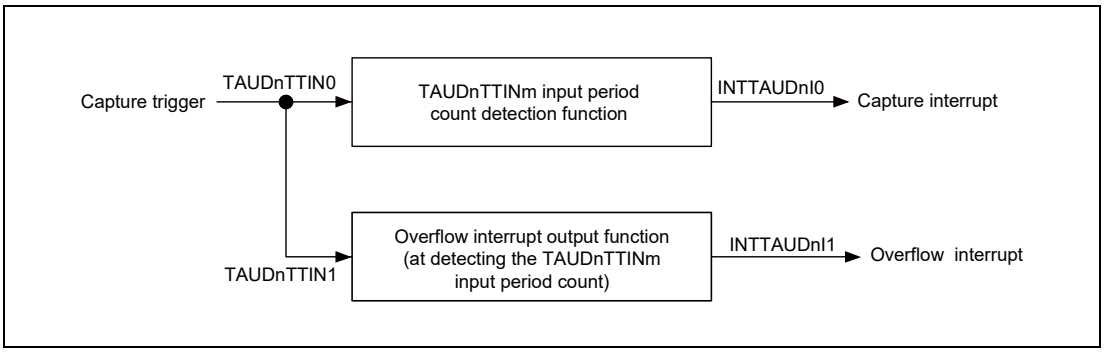


Figure 33.24 Combination of the TAUDnTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDnTTINm Input Period Count)

Timing diagram

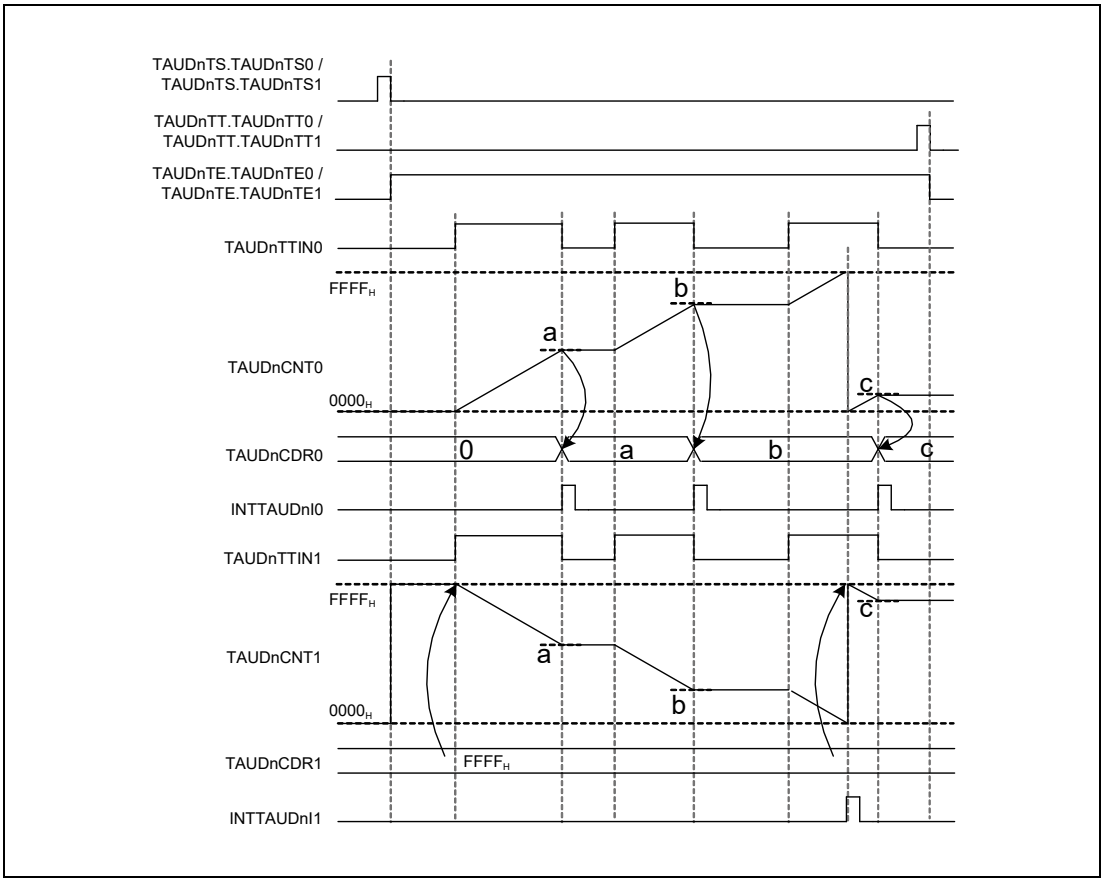


Figure 33.25 Interrupt Generation via Combination of the TAUDnTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDnTTINm Input Period Count)

33.11 TAUDnTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

Figure 33.26 shows when edge detection takes place.

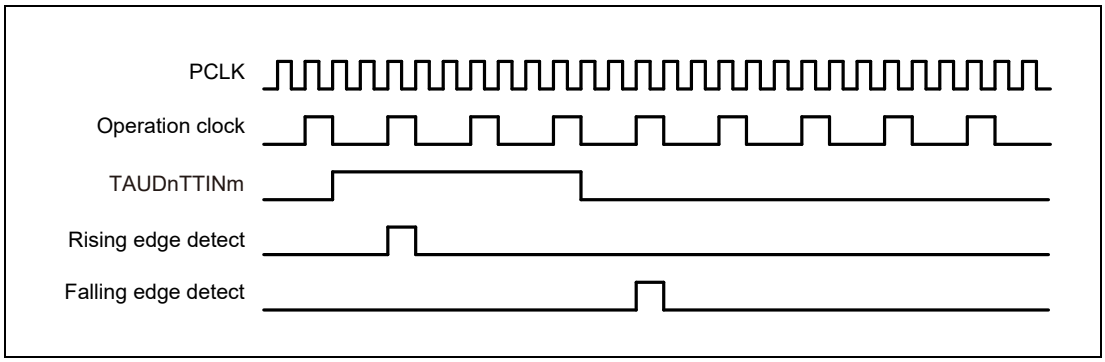


Figure 33.26 Basic Edge Detection Timing

Figure 33.26 shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnIm pin and TAUDn causes a delay time.

33.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the Timer Array Unit D. For a general overview of independent channel operation, see **Section 33.2, Overview**.

This section describes functions that generate interrupts at regular intervals or with a specified delay.

33.12.1 Interval Timer Function

33.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, see **Table 33.45, Contents of the TAUDnCMORm Register for Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, see **Section 33.7, Channel Output Modes**.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is written to TAUDnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUDnIm is generated and the TAUDnTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and subsequently continues operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm stop but retain their values. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDnTTOUTm does not toggle. This results in a negative TAUDnTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.12.1.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDnTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

33.12.1.3 Block Diagram and General Timing Diagram

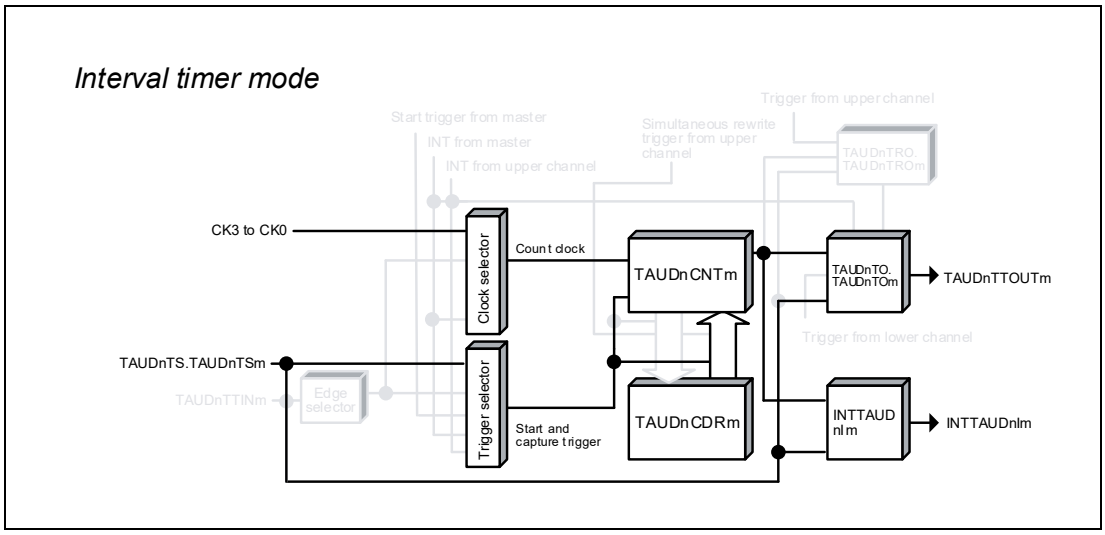


Figure 33.27 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).

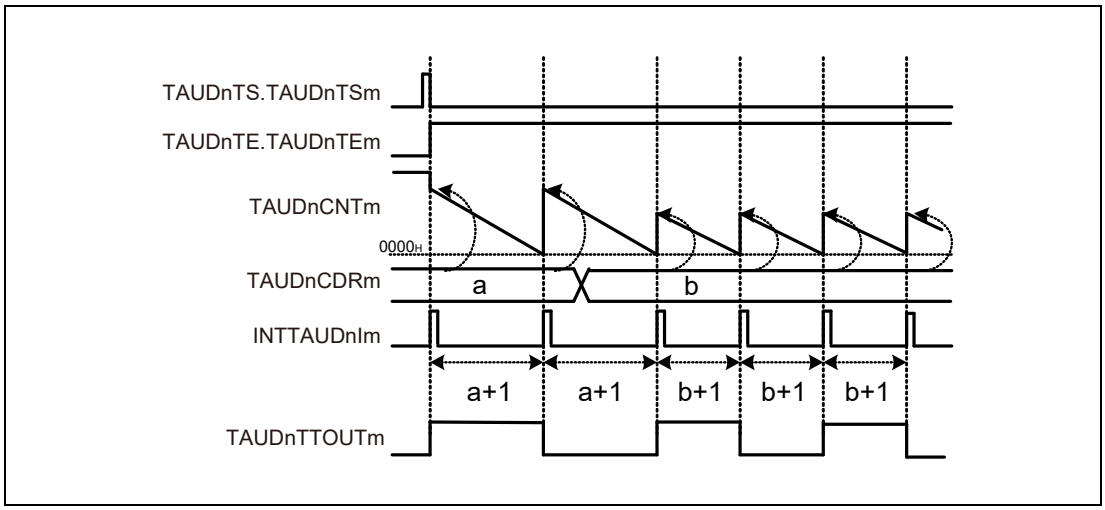


Figure 33.28 General Timing Diagram of Interval Timer Function

33.12.1.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.45 Contents of the TAUDnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Triggers the counter by software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.46 Contents of the TAUDnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

Table 33.47 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 33.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

Table 33.48 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.1.5 Operating Procedure for Interval Timer Function

Table 33.49 Operating Procedure for Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.45, Contents of the TAUDnCMORm Register for Interval Timer Function**, and **Table 33.46, Contents of the TAUDnCMURm Register for Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 33.47, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDnTTOUTm toggles.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated and TAUDnTTOUTm toggles.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.12.1.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H, count clock = PCLK/2

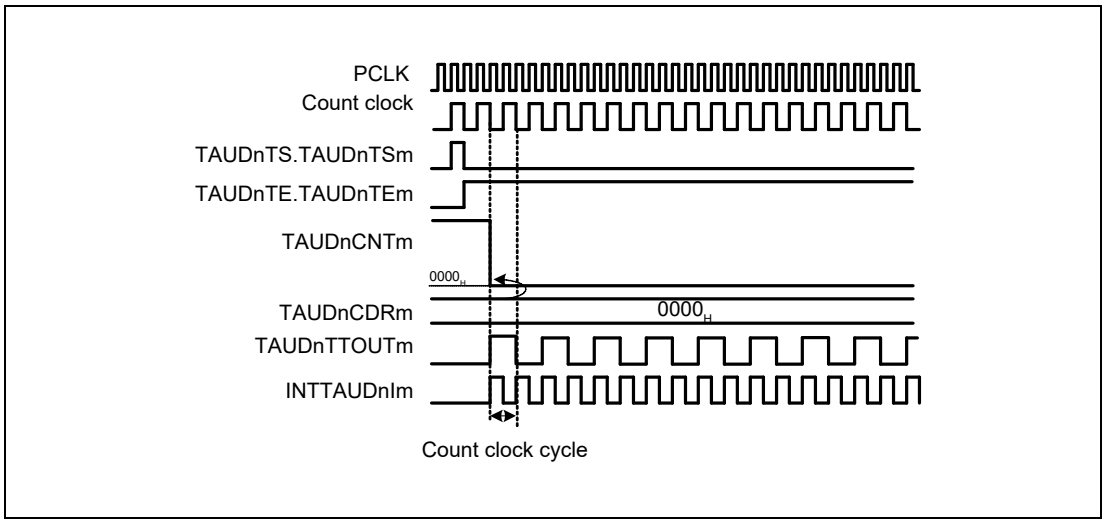


Figure 33.29 TAUDnCDRm = 0000_H, Count Clock = PCLK/2

- If TAUDnCDRm = 0000_H and the count clock = PCLK/2, the TAUDnCDRm value is loaded into TAUDnCNTm every count clock, meaning that TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDnTTOUTm toggling every count clock.

(2) TAUDnCDRm = 0000_H, count clock = PCLK

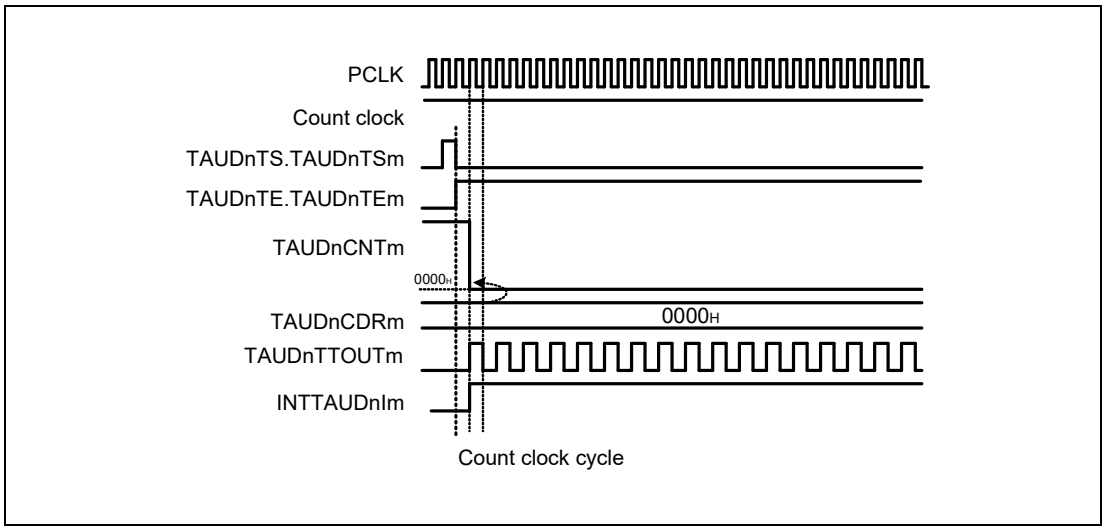


Figure 33.30 TAUDnCDRm = 0000_H, Count Clock = PCLK

- If TAUDnCDRm = 0000_H and the count clock = PCLK, the TAUDnCDRm value is loaded into TAUDnCNTm every PCLK clock, meaning that TAUDnCNTm is always 0000_H.
- INTTAUDnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUDnTTOUTm is toggled every PCLK clock.

(3) Operation Stop and Restart

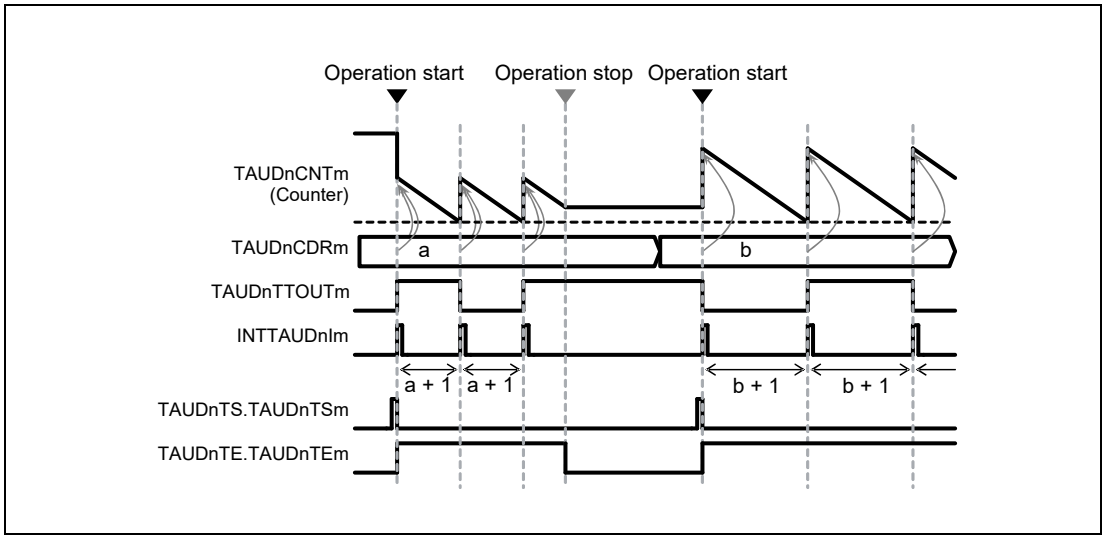


Figure 33.31 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

(4) Forced Restart (TAUDnCMORm.TAUDnMD0 = 1)

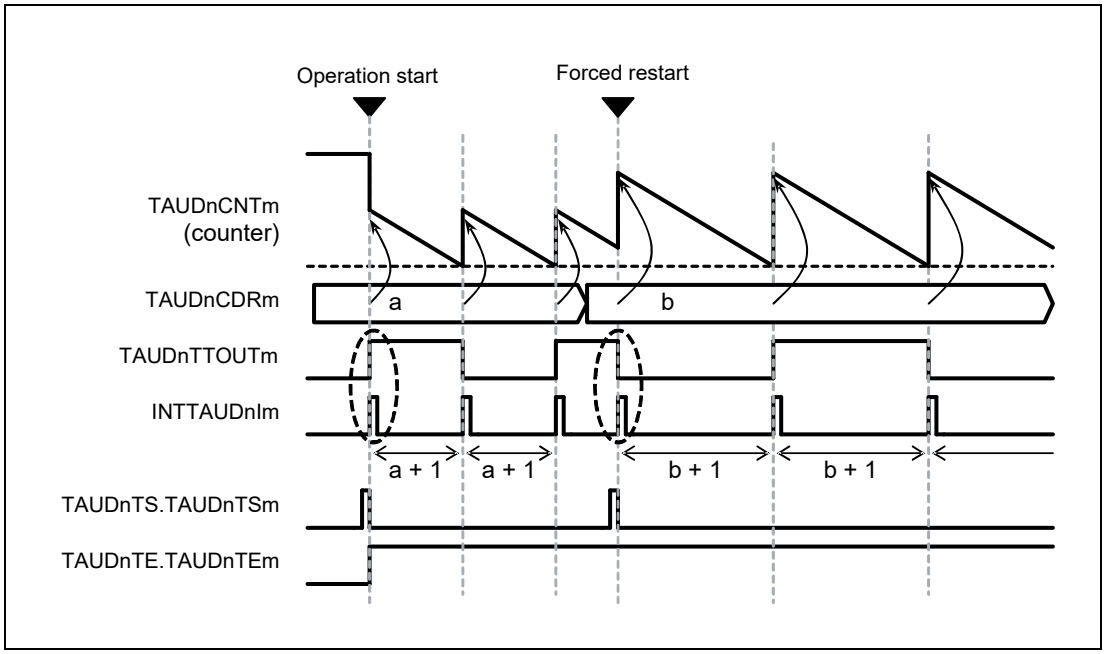


Figure 33.32 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUDnCDRm value is reflected to TAUDnCNTm and counting starts. Execute a forced restart to reflect the changed TAUDnCDRm value immediately.
- When a forced restart is made, an interrupt (INTTAUDnIm) is generated and TAUDnTTOUTm is inverted.

(5) Forced Restart (TAUDnCMORm.TAUDnMD0 = 0)

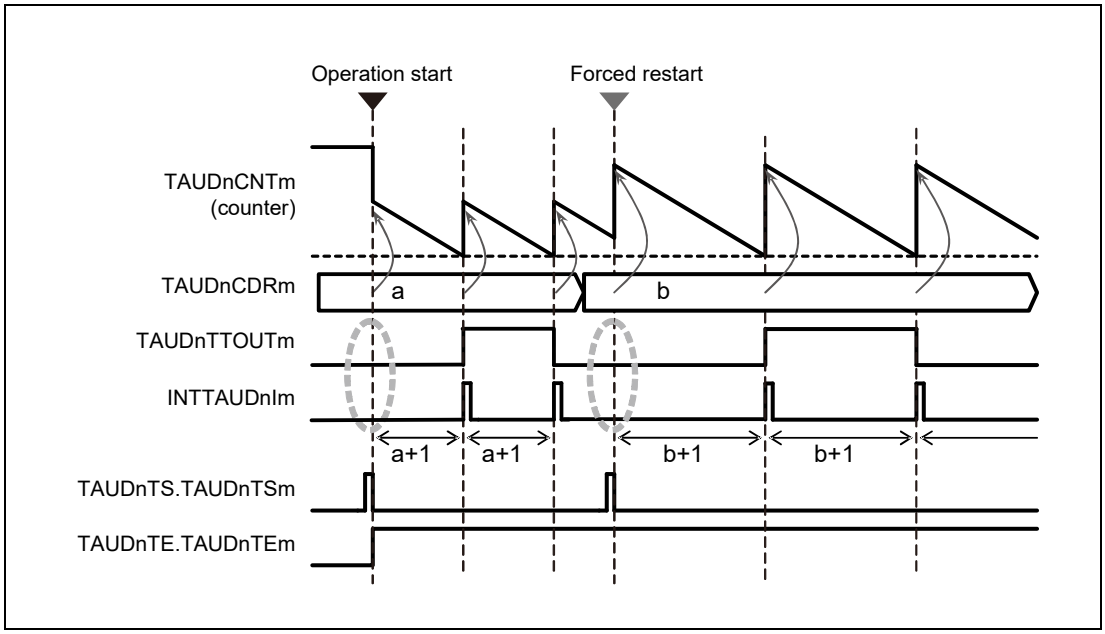


Figure 33.33 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 0)

- When a forced restart is made, an interrupt (INTTAUDnIm) is not generated and TAUDnTTOUTm is not inverted.

33.12.2 TAUDnTTINm Input Interval Timer Function

33.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals or when a valid TAUDnTTINm input edge is detected. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to interval timer mode. See **Table 33.50, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Interval Timer Function**.
- The channel output mode should be set to independent channel output mode 1. See **Section 33.7, Channel Output Modes**.

Functional description

This function operates in an identical manner to the interval timer function (see **Section 33.12.1, Interval Timer Function**) except that this function is restarted by a valid TAUDnTTINm input edge. The type of edge used as a trigger is specified using the TAUDnCMURm.TAUDnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

33.12.2.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDnTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

33.12.2.3 Block Diagram and General Timing Diagram

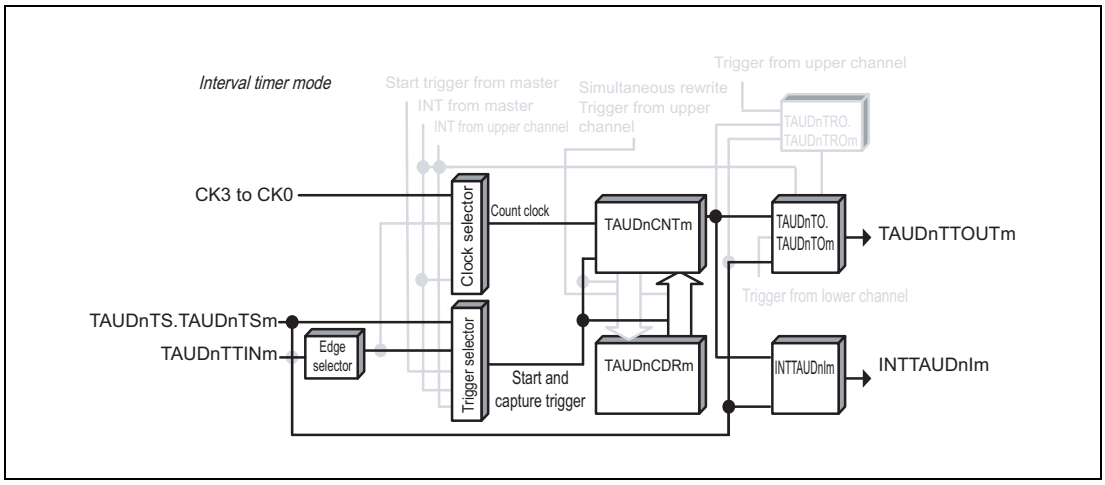


Figure 33.34 Block Diagram of TAUDnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).
- Rising edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

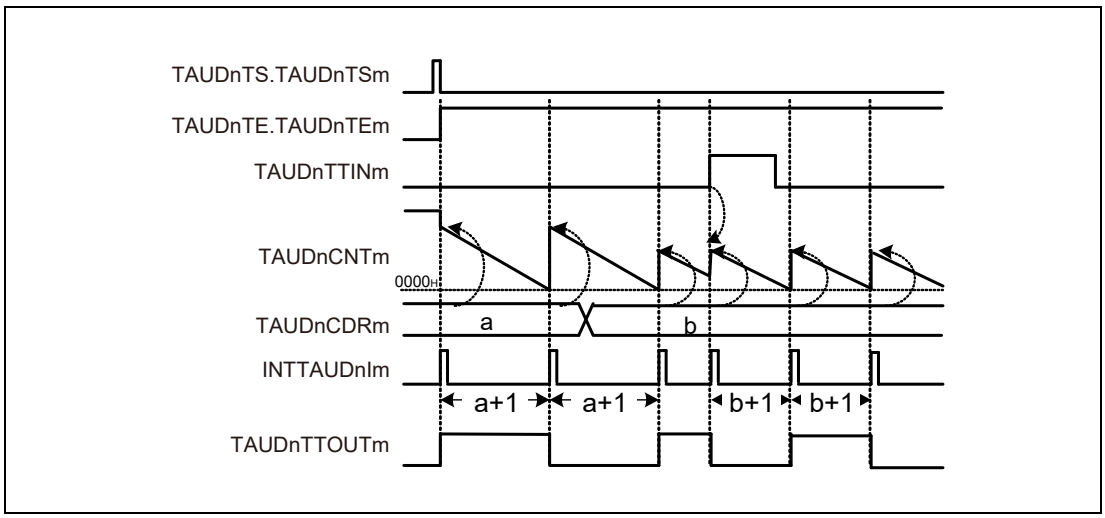


Figure 33.35 General Timing Diagram of TAUDnTTINm Input Interval Timer Function

33.12.2.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.50 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.51 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 33.52 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 33.7, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 33.53 Simultaneous Rewrite Settings for TAUDnTTINm Input Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDS.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.2.5 Operating Procedure for TAUDnTTINm Input Interval Timer Function

Table 33.54 Operating Procedure for TAUDnTTINm Input Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.50, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Interval Timer Function**, and **Table 33.51, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 33.52, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDnTTOUTm toggles.
During Operation	The values of the TAUDnCMURm.TAUDnTIS[1:0] and the TAUDnCDRm register are changeable at any time. The TAUDnCNTm register can be read at all times. Detection of TAUDnTTINm edge	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated and TAUDnTTOUTm toggles. When a TAUDnTTINm input valid edge is detected during count operation, the TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 33.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUDnTTINm input edge.

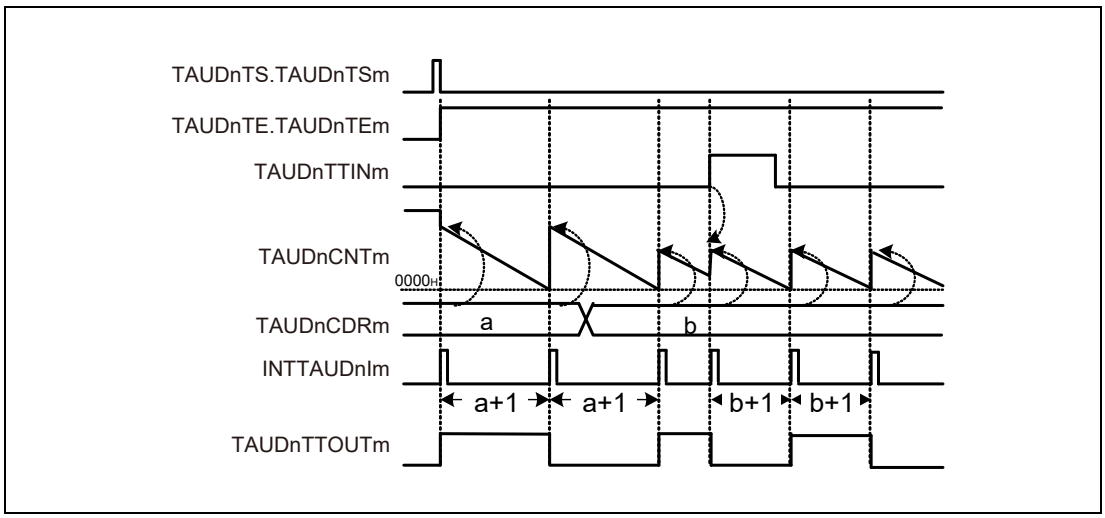


Figure 33.36 Counter Triggered by Rising TAUDnTTINm Input Edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B), TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDnTTINm input edge is detected, an interrupt is generated which causes TAUDnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B).

33.12.3 Clock Divide Function

33.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUDnTTINm is divided by a factor related to TAUDnCDRm, and the resulting signal is output to TAUDnTTOUTm.

Prerequisites

- TAUDnTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See **Table 33.55, Contents of the TAUDnCMORm Register for Clock Divide Function**.)
- The channel output mode should be set to independent channel output mode 1. (See **Section 33.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value, using TAUDnTTINm as a count clock.

When the counter value reaches 0000_H, INTTAUDnIm occurs and TAUDnTTOUTm signal is toggled. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The value of TAUDnCDRm can be rewritten at any time. The changed value of TAUDnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDnTTOUTm stop but retain their values. The function can be restarted by setting TAUDnTS.TAUDnTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUDnTS.TAUDnTSm = 1 during operation (forced restart).

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDnTTOUTm does not toggle. This results in a negative TAUDnTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

TAUDnTTINm input signals are sampled at the frequency of the operation clock set by TAUDnCMORm.TAUDnCKS[1:0] bits. Therefore, the TAUDnTTOUTm output clock cycle has an error of ± 1 operation clock cycle.

33.12.3.2 Equations

- When rising edge detection is selected:
  TAUDnTTOUTm frequency = TAUDnTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUDnTTOUTm frequency = TAUDnTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling and rising edge detection is selected:
  TAUDnTTOUTm frequency = TAUDnTTINm frequency/(TAUDnCDRm + 1)

33.12.3.3 Block Diagram and General Timing Diagram

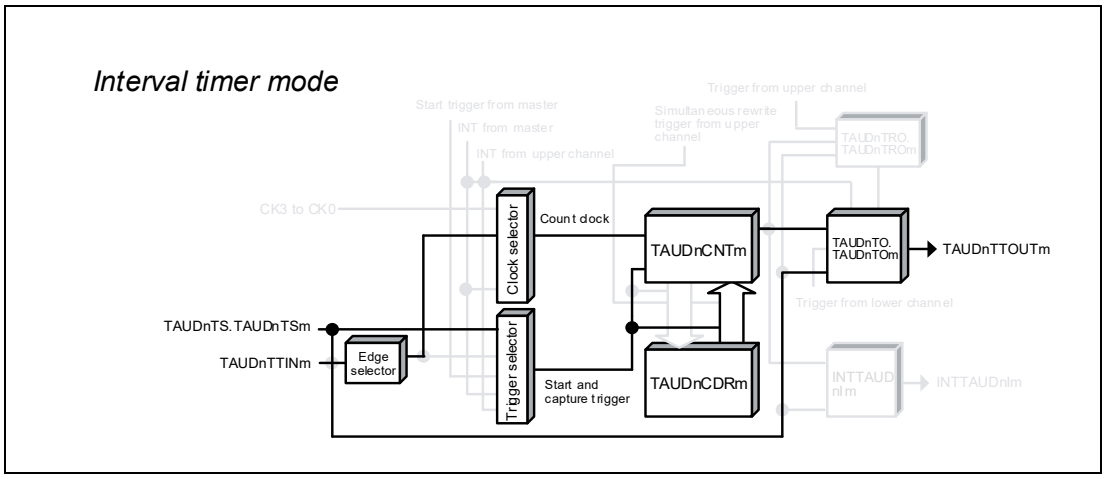


Figure 33.37 Block Diagram of Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

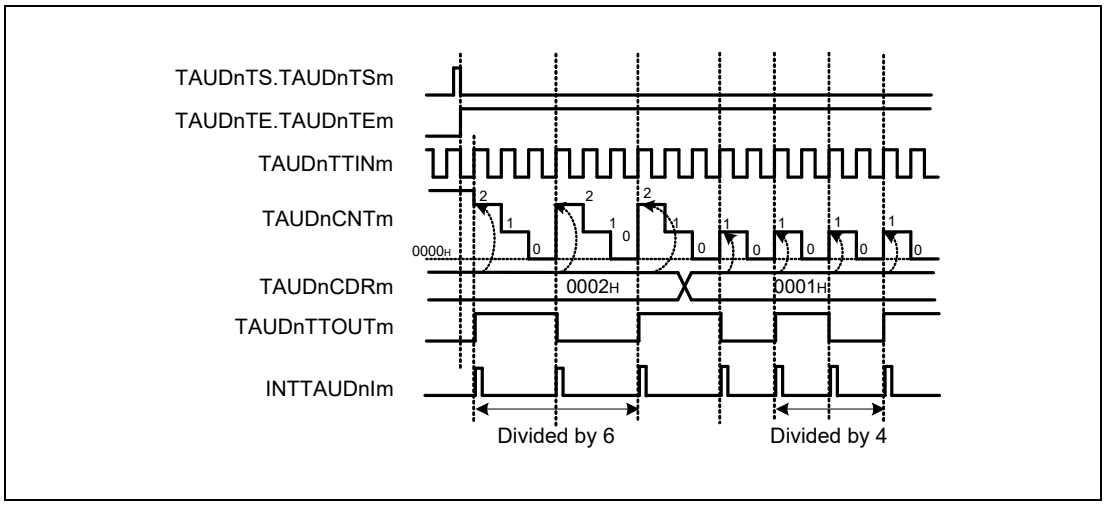


Figure 33.38 General Timing Diagram of Clock Divide Function

33.12.3.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.55 Contents of the TAUDnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDnTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDnTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.56 Contents of the TAUDnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 33.57 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the clock divide function. Therefore, these registers should be set to 0.

Table 33.58 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.3.5 Operating Procedure for Clock Divide Function

Table 33.59 Operating Procedure for Clock Divide Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.55, Contents of the TAUDnCMORm Register for Clock Divide Function**, and **Table 33.56, Contents of the TAUDnCMURm Register for Clock Divide Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 33.57, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 is set to 1, INTTAUDnIm is generated and TAUDnTTOUTm is toggled.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down each time TAUDnTTINm input edge is detected. When the counter reaches 0000_H: • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. • TAUDnTTOUTm is toggled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDnTTOUTm retain their current values.

Restart Operation

33.12.3.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

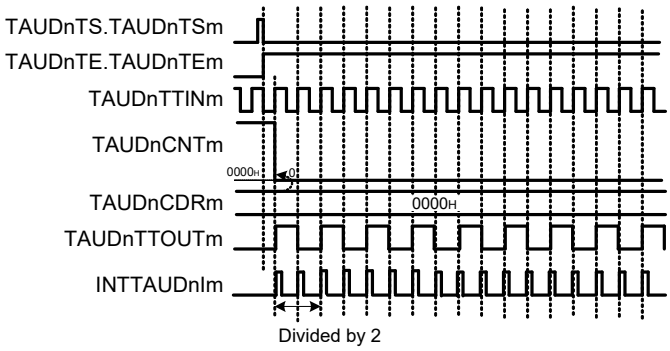


Figure 33.39 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If TAUDnCDRm is 0000_H, TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDnTTOUTm toggling every count clock.

Figure 33.39 shows an operation timing example. Actually, there is a delay from TINm detection until TOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUDnIm pin and TAUDn.

(2) Operation Restart

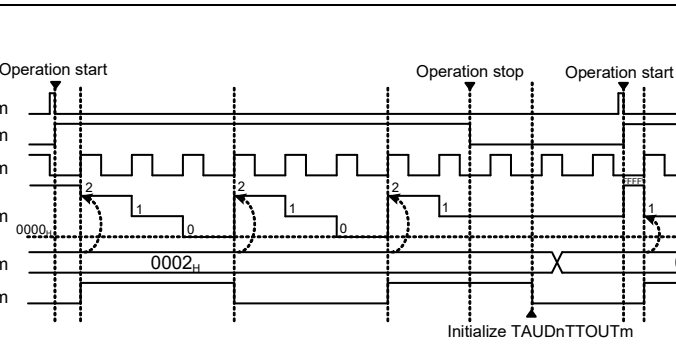


Figure 33.40 Operation Restart (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

To reset the value of TAUDnTTOUTm:

- Set TAUDnTOE.TAUDnTOEm = 0 when the counter is stopped (TAUDnTE.TAUDnTEm = 0).
- Then, write either 0 or 1 to TAUDnTO.TAUDnTOm to set the new start value of TAUDnTTOUTm.

(3) Forced Restart

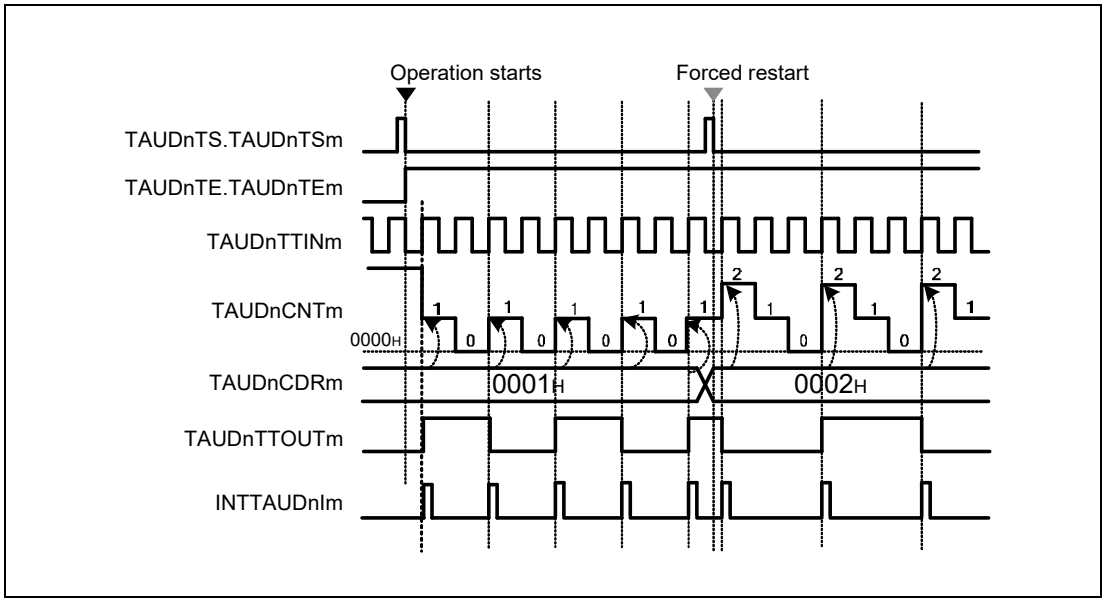


Figure 33.41 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm = 1 during operation.
- The value of TAUDnCDRm is written to TAUDnCNTm and the count operation restarts.
- TAUDnTTOUTm restarts at the same level as before the forced restart.

33.12.4 External Event Count Function

33.12.4.1 Overview

Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of valid TAUDnTTINm input edges are detected.

Prerequisites

- The operating mode should be set to the event count mode. (See **Table 33.60, Contents of the TAUDnCMORm Register for External Event Count Function**.)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When the counter starts, the current value of TAUDnCDRm is loaded into TAUDnCNTm.

When a valid TAUDnTTINm input edge is detected, the value of TAUDnCNTm decrements by 1. TAUDnCNTm retains this value until a valid TAUDnTTINm input edge is detected or the counter is restarted.

When the valid edge is detected for the (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDnTS.TAUDnTSm to 1 during operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

An edge type used as a trigger is specified by TAUDnCMURm.TAUDnTIS[1:0] bits.

- When TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 10_B, both edges are counted.

33.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

33.12.4.3 Block Diagram and General Timing Diagram

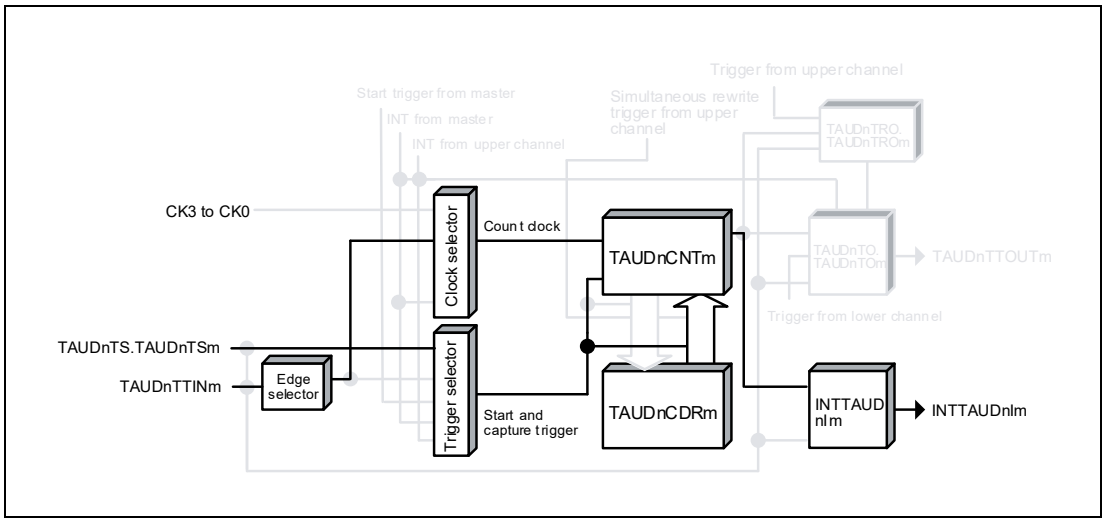


Figure 33.42 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

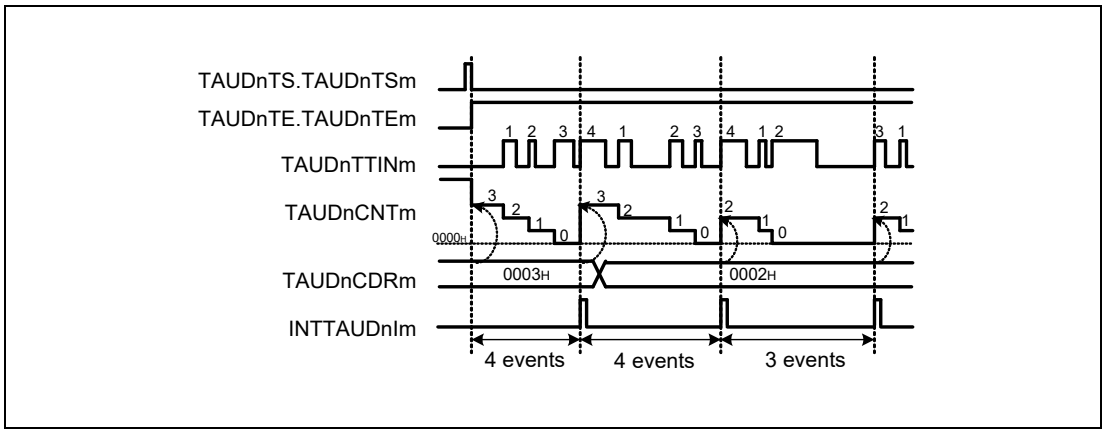


Figure 33.43 General Timing Diagram of External Event Count Function

33.12.4.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.60 Contents of the TAUDnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDnTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.61 Contents of the TAUDnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited.

(3) Channel Output Mode

The channel output mode is not used by this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the external event count function. Therefore, these registers should be set to 0.

Table 33.62 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.4.5 Operating Procedure for External Event Count Function

Table 33.63 Operating Procedure for External Event Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.60, Contents of the TAUDnCMORm Register for External Event Count Function**, and **Table 33.61, Contents of the TAUDnCMURm Register for External Event Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value and waits for TAUDnTTINm input edge detection.
During Operation	Detection of TAUDnTTINm edge The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down each time TAUDnTTINm input edge is detected. When effective edges are detected (TAUDnCDRm + 1) times: • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.4.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

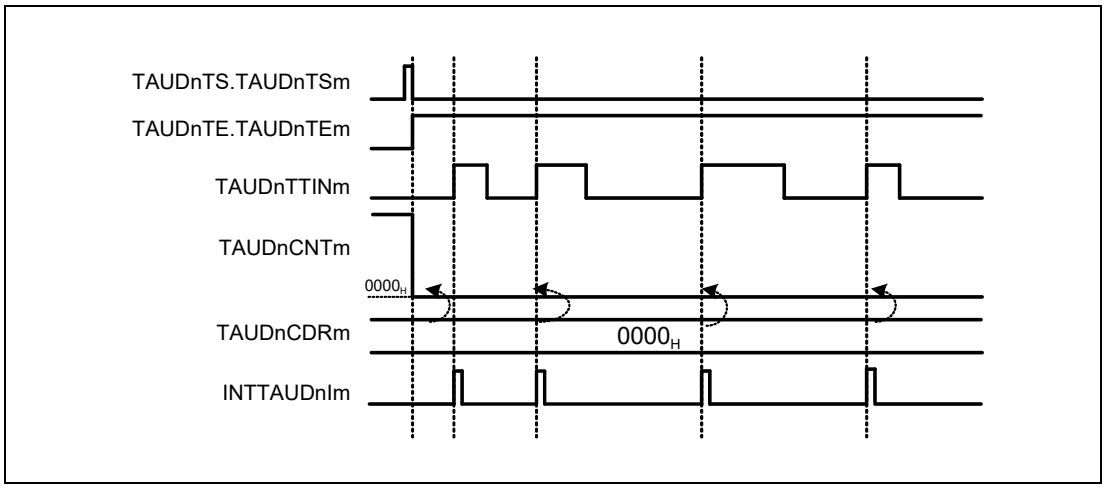


Figure 33.44 TAUDnCDRm = 0000_H, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If 0000_H = TAUDnCDRm, 0000_H is loaded into TAUDnCNTm each time a valid TAUDnTTINm input edge is detected.
  In other words, INTTAUDnIm is generated each time a valid TAUDnTTINm input edge is detected.

(2) Operation Stop and Restart

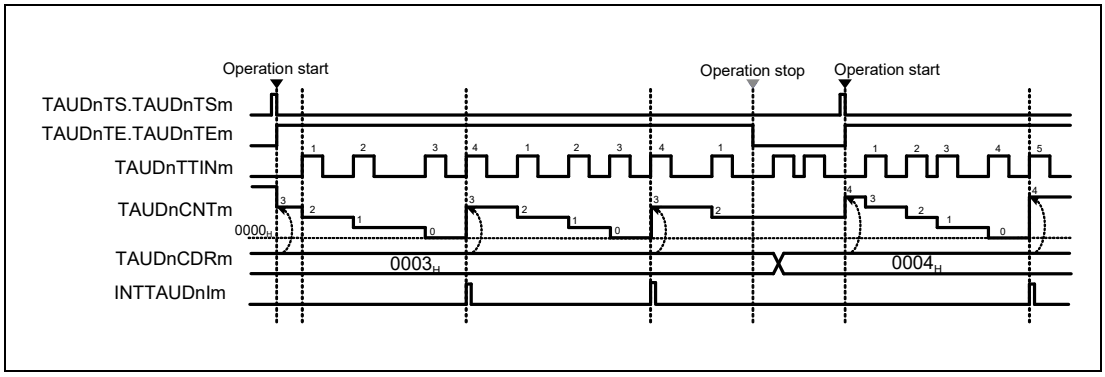


Figure 33.45 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value. TAUDnTTINm continues and TAUDnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm loads the TAUDnCDRm value and restarts count operation.

(3) Forced Restart

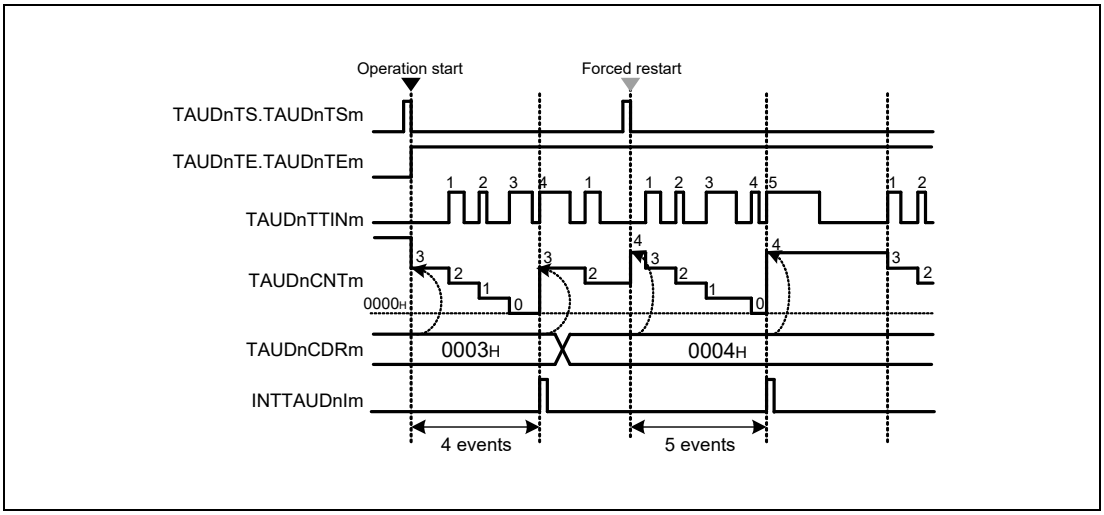


Figure 33.46 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

Once a forced restart is made, the changed TAUDnCDRm value is applied to TAUDnCNTm immediately.

- The counter can be restarted without making a stop by setting TAUDnTS.TAUDnTSm to 1 during operation.
- The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter awaits the next valid TAUDnTTINm input edge.

33.12.5 Delay Count Function

33.12.5.1 Overview

Summary

This function generates interrupts (INTTAUDnIm), which have a defined delay to the TAUDnTTINm input signal. TAUDnTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites

- The operating mode should be set to one-count mode. See **Table 33.64, Contents of the TAUDnCMORm Register for Delay Count Function**.
- TAUDnTTOUTm is not used with this function.
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUDnTTINm input edge.

When the counter is counting down, further TAUDnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

33.12.5.2 Equations

Delay between TAUDnTTINm and INTTAUDnIm = count clock cycle × (TAUDnCDRm + 1)

33.12.5.3 Block Diagram and General Timing Diagram

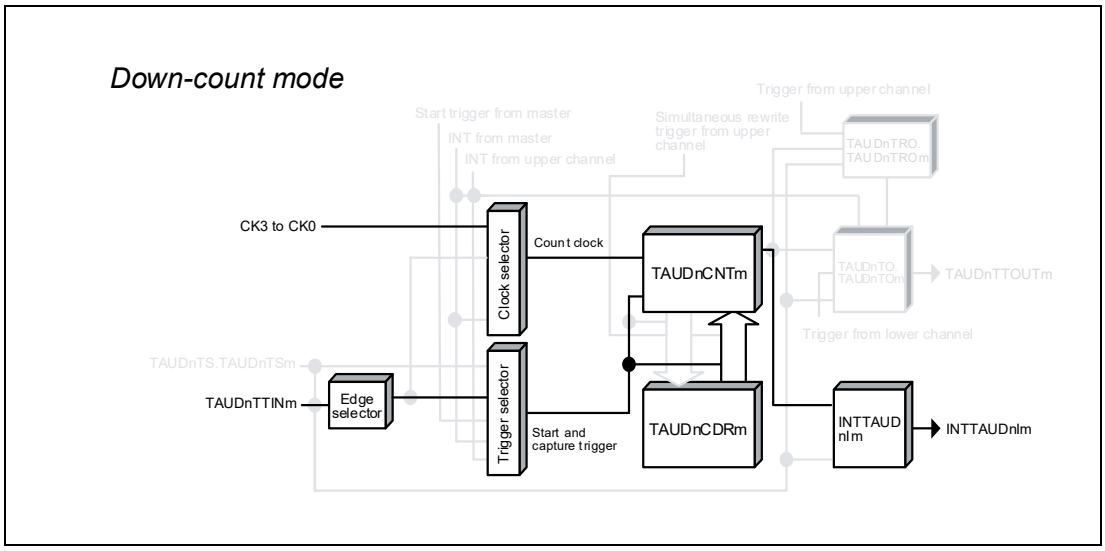


Figure 33.47 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

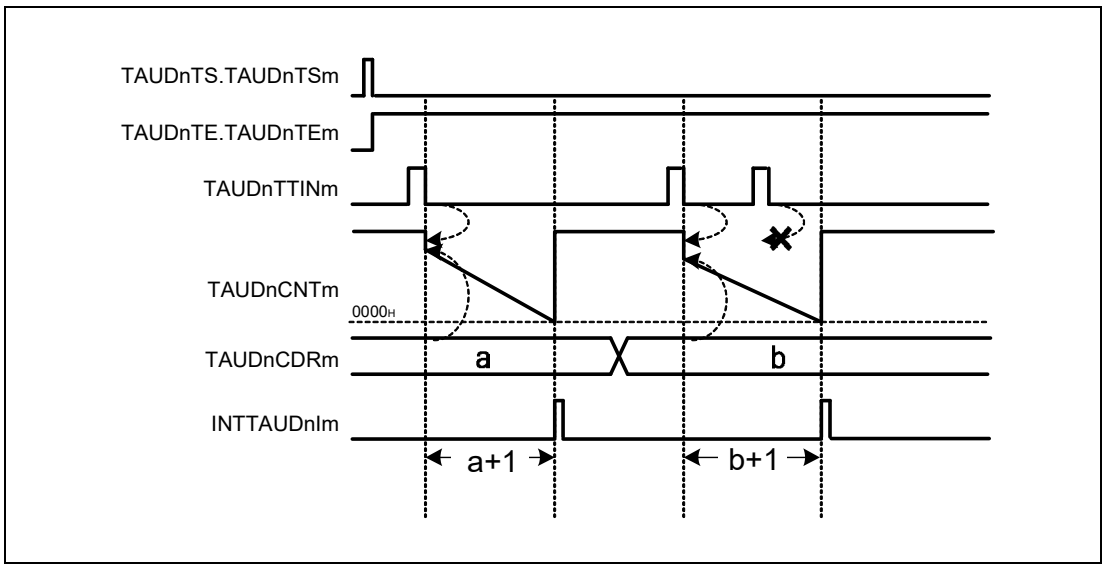


Figure 33.48 General Timing Diagram of Delay Count Function

33.12.5.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.64 Contents of the TAUDnCMORm Register for Delay Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables a start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.65 Contents of the TAUDnCMURm Register for Delay Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the delay count function. Therefore, these registers should be set to 0.

Table 33.66 Simultaneous Rewrite Settings for Delay Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.5.5 Operating Procedure for Delay Count Function

Table 33.67 Operating Procedure for Delay Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.64, Contents of the TAUDnCMORm Register for Delay Count Function**, and **Table 33.65, Contents of the TAUDnCMURm Register for Delay Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When a start edge is detected, the TAUDnCDRm value is loaded in TAUDnCNTm.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H. INTTAUDnIm is generated. TAUDnCNTm stops counting, returns $FFFF_H$, and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its value.

Restart Operation

33.12.6 One-Pulse Output Function

33.12.6.1 Overview

Summary

This function generates an interrupt (INTTAUDnIm) when a valid TAUDnTTINm input edge is detected and at a defined interval afterward. TAUDnTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUDnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode should be set to pulse one-count mode. (See **Table 33.68, Contents of the TAUDnCMORm Register for One-Pulse Output Function**.)
- The channel output mode should be set to independent channel output mode 2. (See **Section 33.7, Channel Output Modes**.)
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. An interrupt is generated and TAUDnTTOUTm is set to active level.

When the counter reaches 0001_H, an interrupt is generated and TAUDnTTOUTm is set to the inactive level. The counter stops at 0000_H and awaits the next valid TAUDnTTINm input edge.

When the counter is counting down, further TAUDnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

33.12.6.2 Equations

Interval between TAUDnTTINm and INTTAUDnIm = TAUDnTTOUTm (timer output) width = count clock cycle × TAUDnCDRm

33.12.6.3 Block Diagram and General Timing Diagram

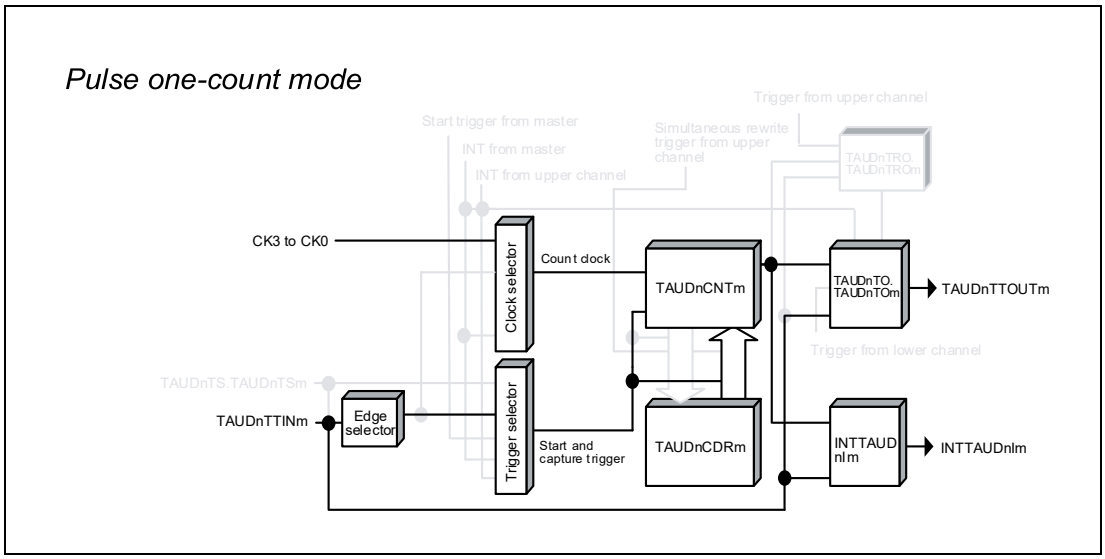


Figure 33.49 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

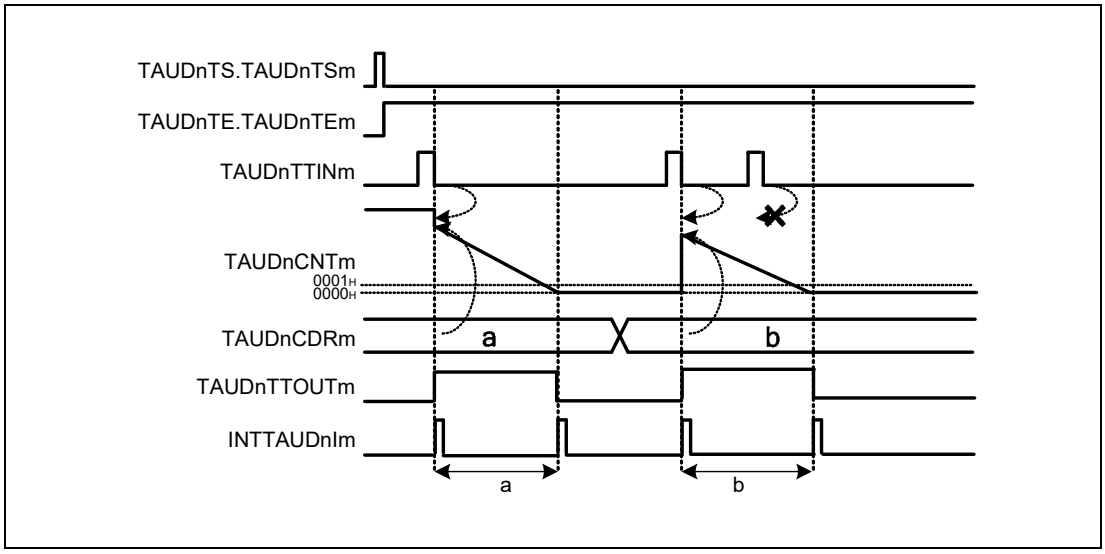


Figure 33.50 General Timing Diagram of One-Pulse Output Function

33.12.6.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.68 Contents of the TAUDnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables a start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.69 Contents of the TAUDnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

Table 33.70 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode controlled by software.
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Table 33.44, Channel Output Modes**.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

Table 33.71 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.6.5 Operating Procedure for One-Pulse Output Function

Table 33.72 Operating Procedure for One-Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.68, Contents of the TAUDnCMORm Register for One-Pulse Output Function**, and **Table 33.69, Contents of the TAUDnCMURm Register for One-Pulse Output Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 33.70, Control Bit Settings in Independent Channel Output Mode 2**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	INTTAUDnIm is generated when TAUDnCNTm starts and TAUDnTTOUTm is set to its active level. TAUDnCNTm counts down. When the counter reaches 0001_H: • INTTAUDnIm is generated. • TAUDnTTOUTm is set to its inactive level. TAUDnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.12.7 TAUDnTTINm Input Pulse Interval Measurement Function

33.12.7.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUDnCSRm.TAUDnOVF to measure the interval of the TAUDnTTINm input signals.

Prerequisites

- The operating mode should be set to capture mode. See **Table 33.74, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Measurement Function**.
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUDnTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 33.73 Effects of Overflow

TAUDnCMORm.TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDnTTINm Input is Detected	
	TAUDnCDRm	TAUDnCSRm.TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm.TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm set to 0, TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, the overflow bit (TAUDnCSRm.TAUDnOVF) can be cleared only by setting TAUDnCSCm.TAUDnCLOV = 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the interval of the TAUDnTTINm signal. However, if an overflow occurs multiple times before a valid TAUDnTTINm input is detected, the overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm stops but retains its value. While the function is stopped, valid TAUDnTTINm input edge detection and TAUDnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUDnCMORm.TAUDnCOS[1:0] = 10_B or 11_B, the value of TAUDnCNTm is not loaded into TAUDnCDRm when the first valid TAUDnTTINm input edge occurs after an overflow. However, an interrupt is generated.

33.12.7.2 Equations

TAUDnTTINm input pulse interval = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

33.12.7.3 Block Diagram and General Timing Diagram

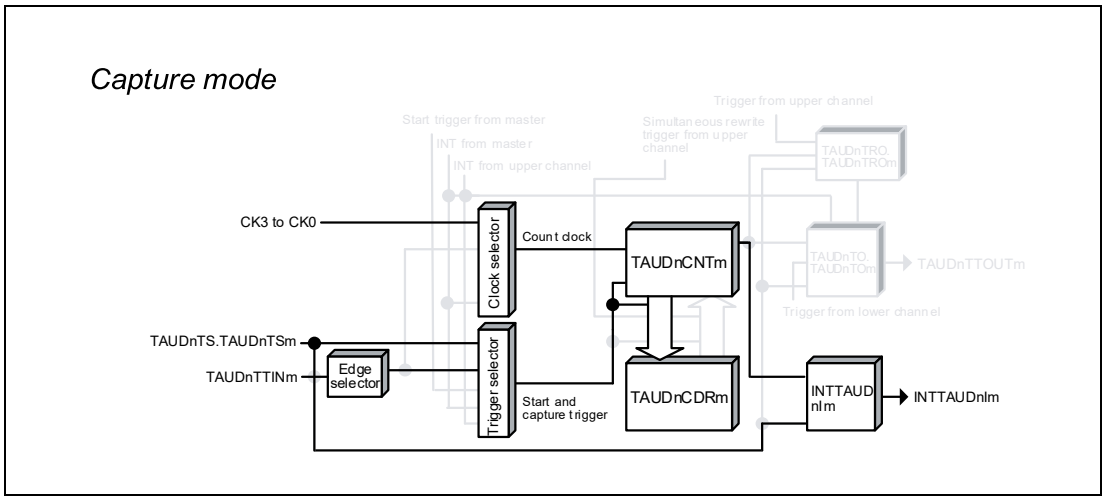


Figure 33.51 Block Diagram of TAUDnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 0).
- Falling edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- When a valid TAUDnTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1 (TAUDnCMORm.TAUDnCOS[1:0] = 00_B).

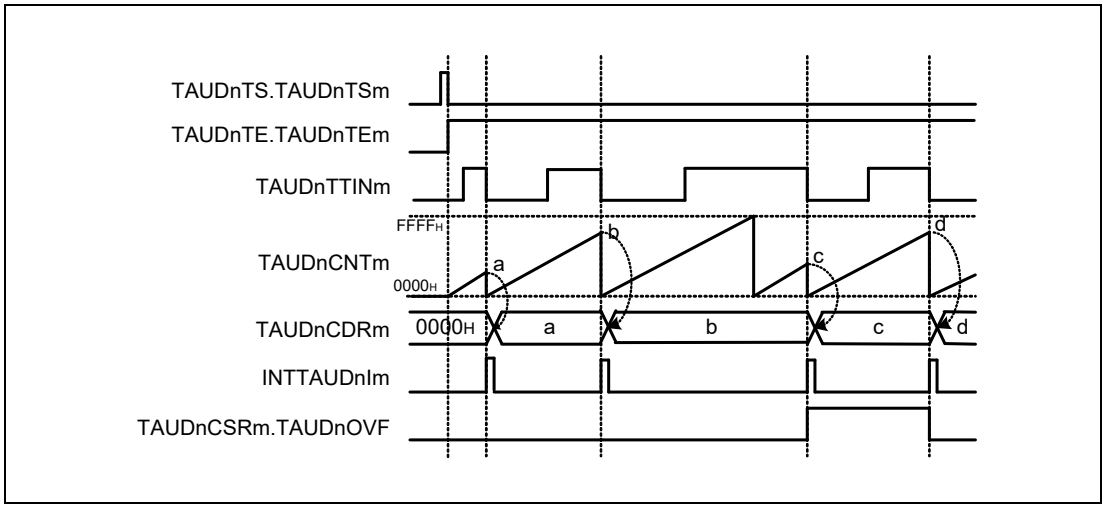


Figure 33.52 General Timing Diagram of TAUDnTTINm Input Pulse Interval Measurement Function

33.12.7.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.74 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDnTTINm input signal is used as the external capture trigger.
7, 6	TAUDnCOS[1:0]	See **Table 33.73, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.75 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

Table 33.76 Simultaneous Rewrite Settings for TAUDnTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.7.5 Operating Procedure for TAUDnTTINm Input Pulse Interval Measurement Function

Table 33.77 Operating Procedure for TAUDnTTINm Input Pulse Interval Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.74, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Measurement Function**, and **Table 33.75, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Pulse Interval Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	Detection of TAUDnTTINm edge The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV can be written to 1. (TAUDnCSRm.TAUDnOVF bit is cleared to 0.)	TAUDnCNTm starts to count up from 0000_H. When a TAUDnTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and returns to 0000_H. • INTTAUDnIm is then generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

Restart Operation

33.12.7.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

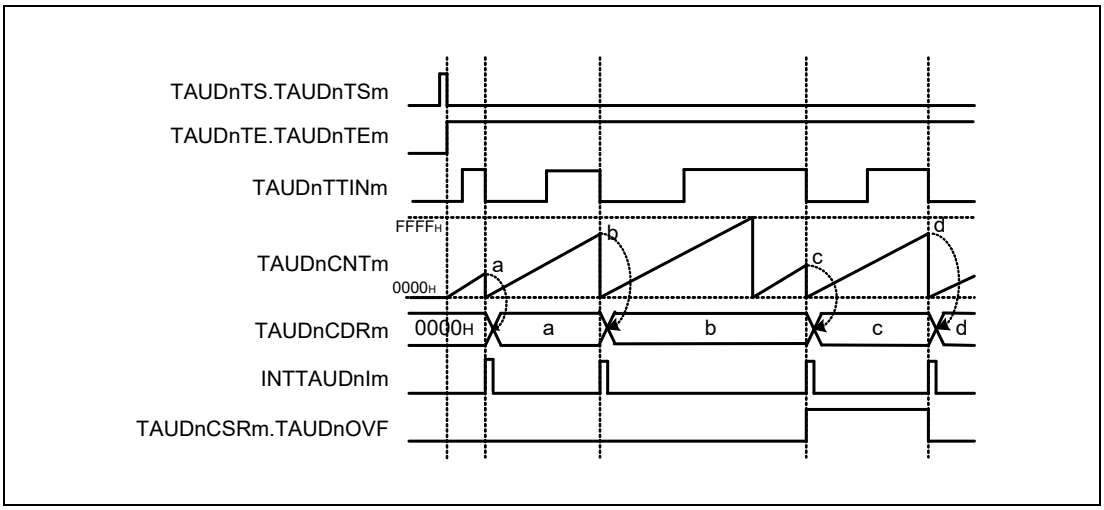


Figure 33.53 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

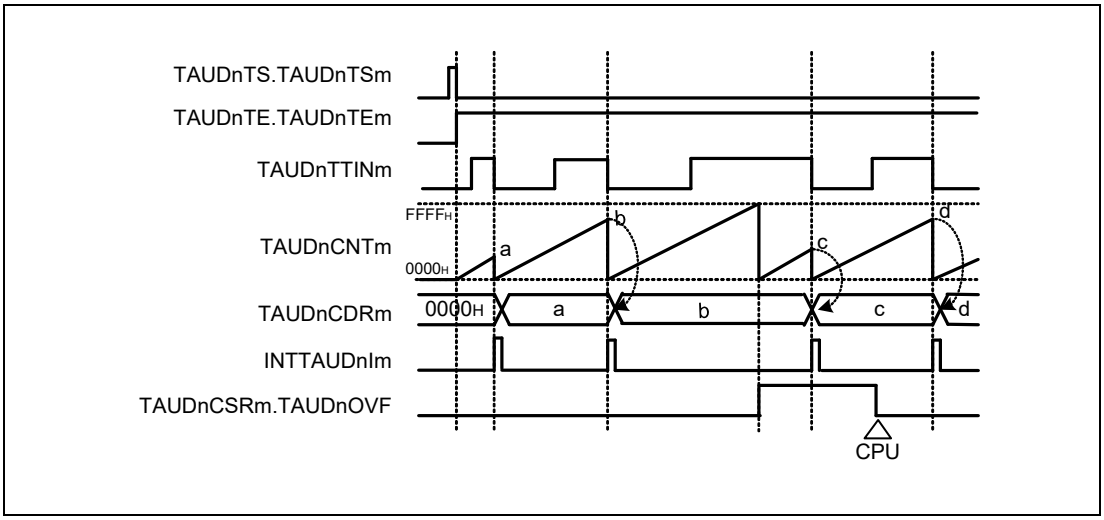


Figure 33.54 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

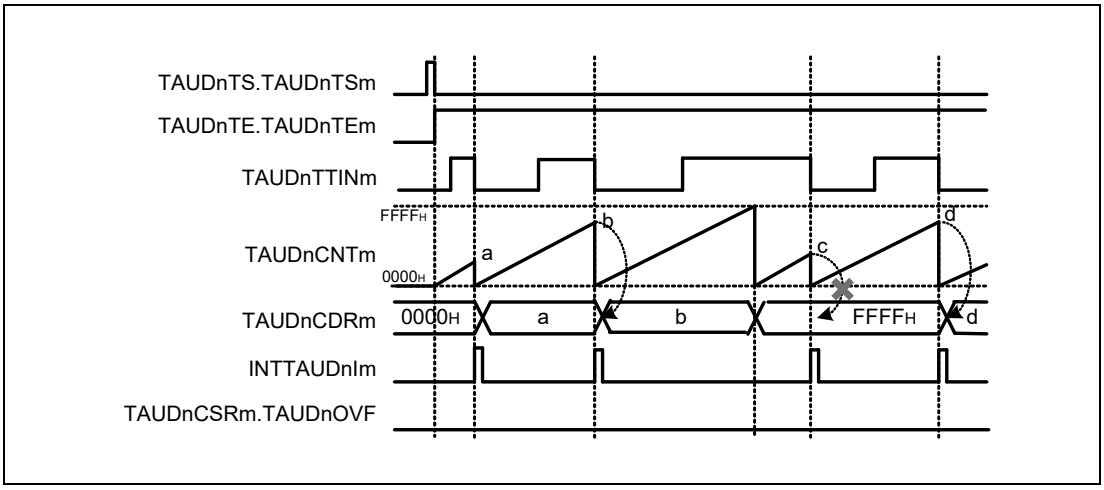


Figure 33.55 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

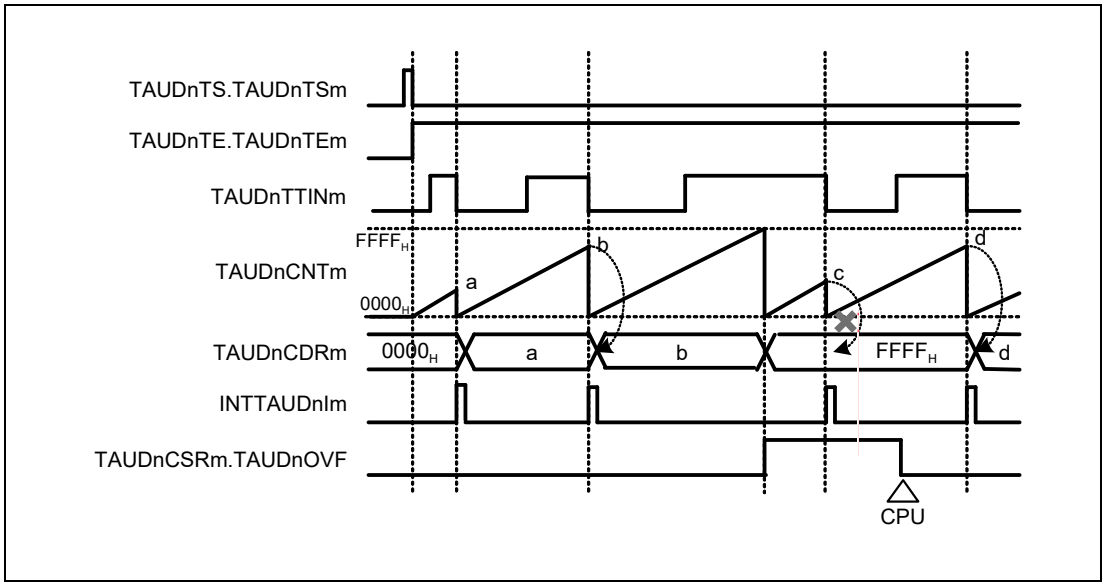


Figure 33.56 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

33.12.8 TAUDnTTINm Input Signal Width Measurement Function

33.12.8.1 Overview

Summary

This function measures the width of a TAUDnTTINm signal, by starting the count at one edge of TAUDnTTINm and capturing the count value at the other edge.

Prerequisites

- The operating mode should be set to capture and one-count mode. See **Table 33.79, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Measurement Function**.
- TAUDnTTOUTm is not used with this function.
- TAUDnCMORm.TAUDnMD0 should be set to 0.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDnTTINm start edge is detected, the counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm stop edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter retains its value (TAUDnCDRm +1) and awaits the next valid TAUDnTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUDnTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 33.78 Effects of Overflow

TAUDnCMORm. TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDnTTINm Input Stop Edge is Detected	
	TAUDnCDRm	TAUDnCSRm. TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm. TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm stops counting TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, overflow bit TAUDnCSRm.TAUDnOVF can be cleared only by setting TAUDnCSCm.TAUDnCLOV to 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the width of the TAUDnTTINm signal. However, if an overflow occurs multiple times before a valid TAUDnTTINm input is detected, overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

This function cannot be forcibly restarted.

NOTE

When TAUDnCMORm.TAUDnCOS[1] = 1, the value of TAUDnCNTm is not loaded to TAUDnCDRm when the first valid TAUDnTTINm input edge occurs after an overflow. However, an interrupt is generated.

33.12.8.2 Equations

TAUDnTTINm input signal width = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

33.12.8.3 Block Diagram and General Timing Diagram

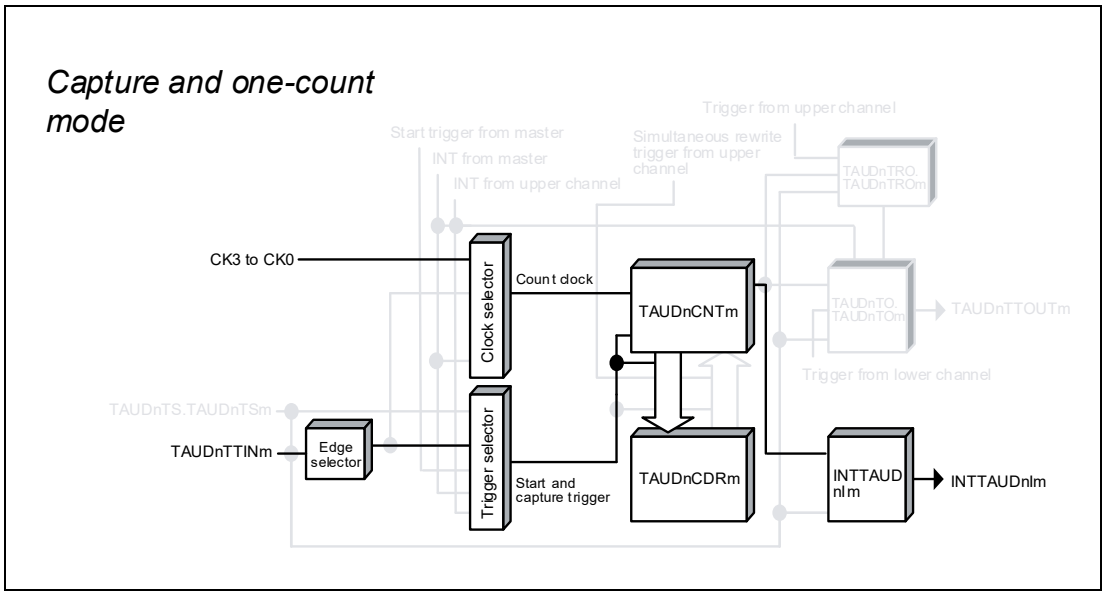


Figure 33.57 Block Diagram of TAUDnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)
- When a valid TAUDnTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1. (TAUDnCMORm.TAUDnCOS[1:0] = 00_B)

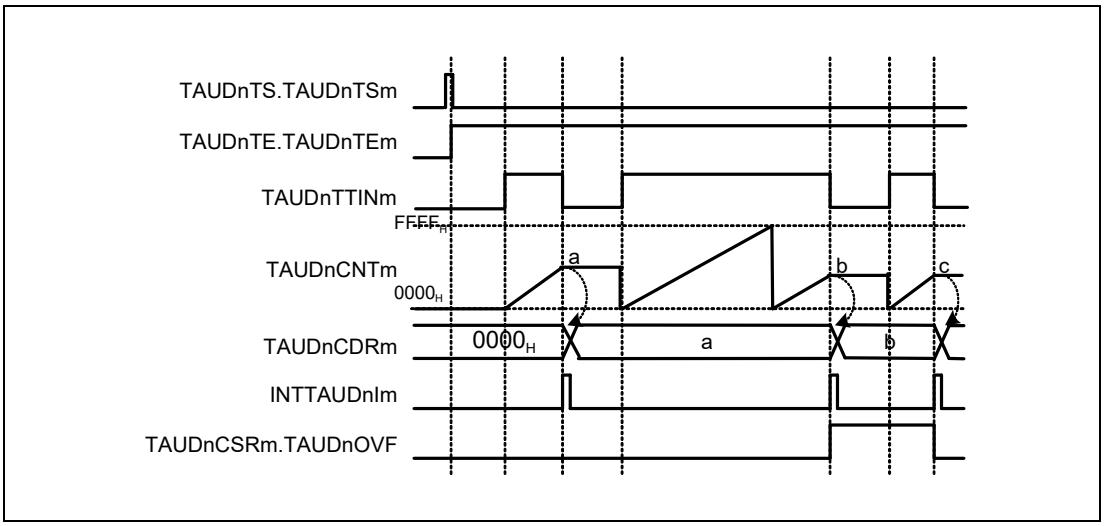


Figure 33.58 General Timing Diagram of TAUDnTTINm Input Signal Width Measurement Function

33.12.8.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.79 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	See **Table 33.78, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0110: Capture and one-count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.80 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input signal width measurement function. Therefore, these registers should be set to 0.

Table 33.81 Simultaneous Rewrite Settings for TAUDnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.8.5 Operating Procedure for TAUDnTTINm Input Signal Width Measurement Function

Table 33.82 Operating Procedure for TAUDnTTINm Input Signal Width Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.79, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Measurement Function**, and **Table 33.80, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Signal Width Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When a TAUDnTTINm start edge is detected, TAUDnCNTm starts to count up.
During Operation	TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV bit can be set to 1.	TAUDnCNTm starts to count up from 0000_H. When TAUDnTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and retains its value. INTTAUDnIm is then generated. Counting stops at the “value that transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

Restart Operation

33.12.8.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

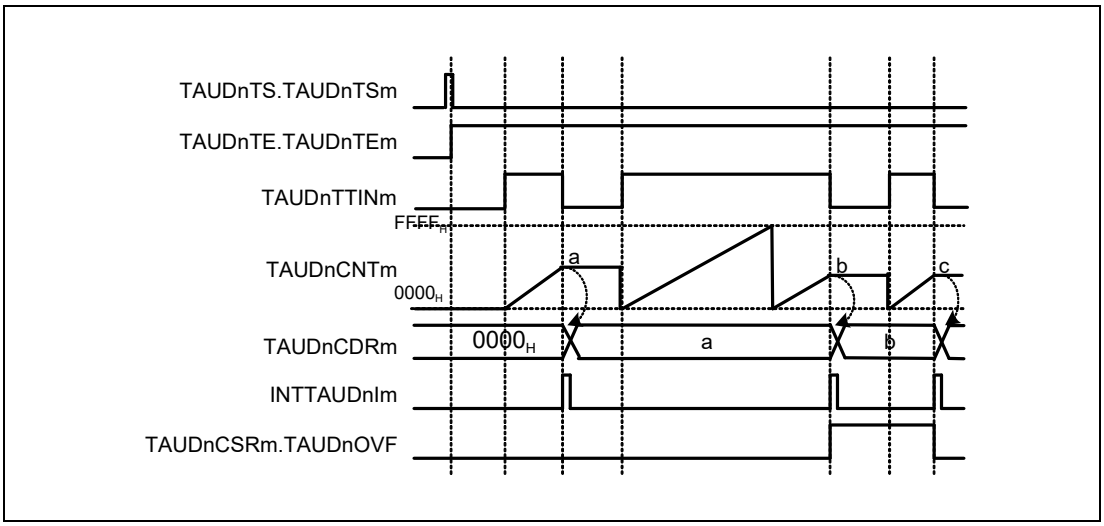


Figure 33.59 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

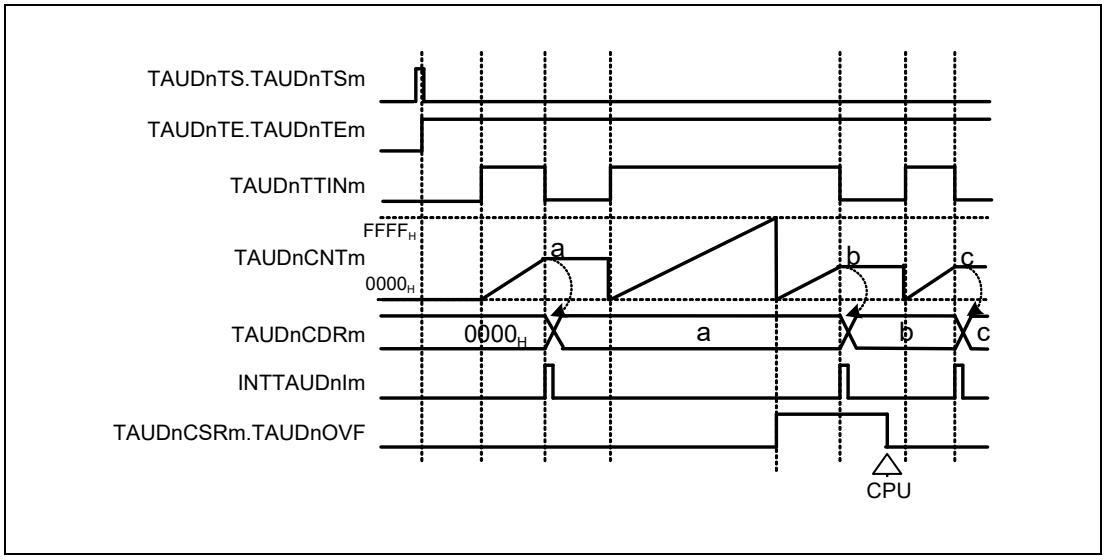


Figure 33.60 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

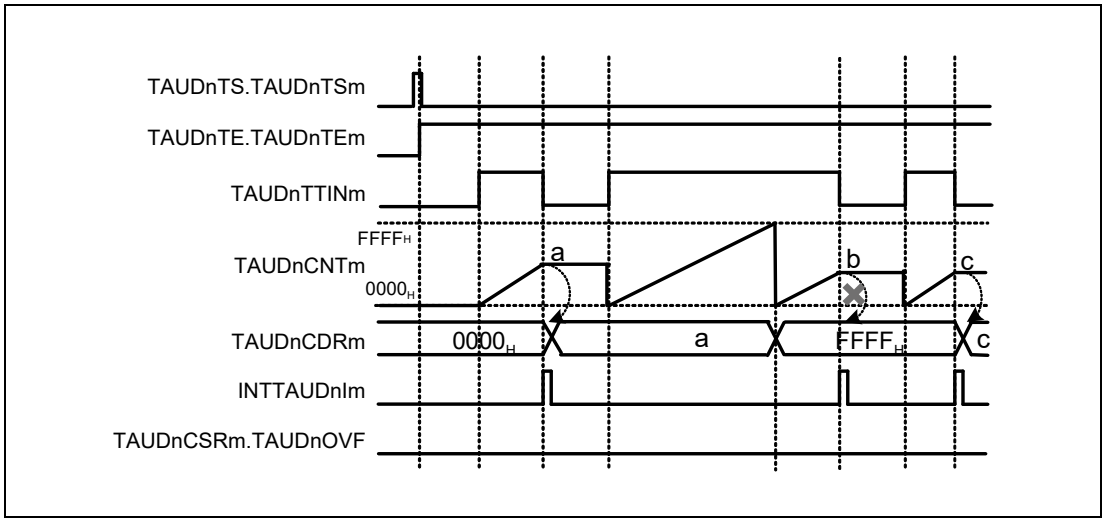


Figure 33.61 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

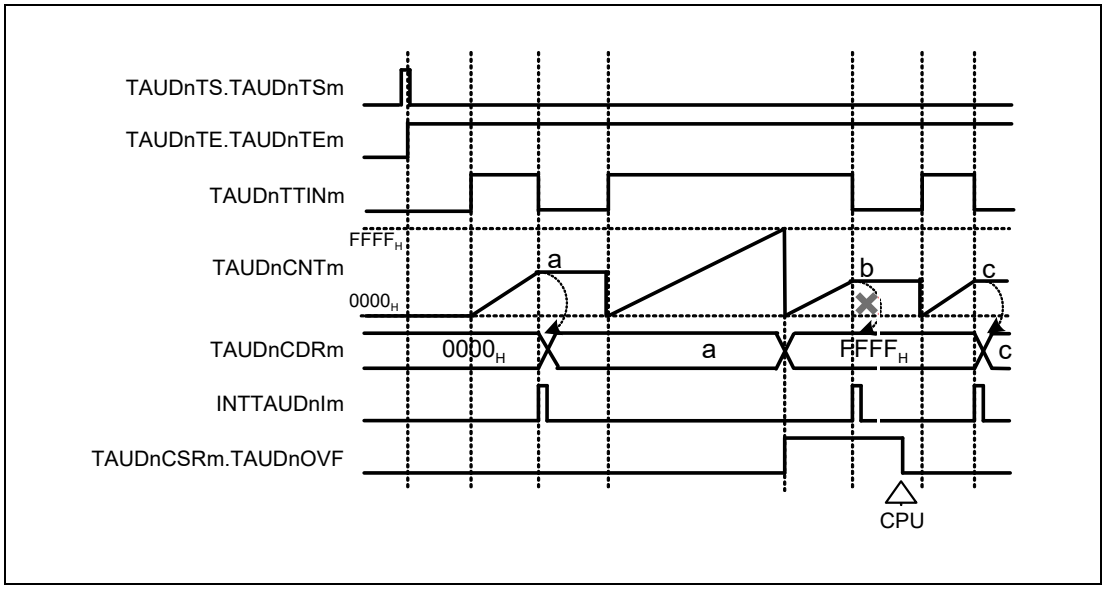


Figure 33.62 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDnTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDnTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

33.12.9 TAUDnTTINm Input Position Detection Function

33.12.9.1 Overview

Summary

This function measures the input signal duration by capturing the count value at the valid edge of TAUDnTTINm.

Prerequisites

- The operating mode should be set to count capture mode. (See **Table 33.83, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Position Detection Function**.)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter starts counting from 0000_H. When a valid TAUDnTTINm input edge is detected, the current value of TAUDnCNTm is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUDnTTINm input signal is sampled at the frequency of the operation clock, specified by TAUDnCMORm.TAUDnCKS[1:0] bits.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.12.9.2 Equations

Functional duration at a TAUDnTTINm input pulse =

count clock cycle × (TAUDnCDRm capture value + 1)

33.12.9.3 Block Diagram and General Timing Diagram

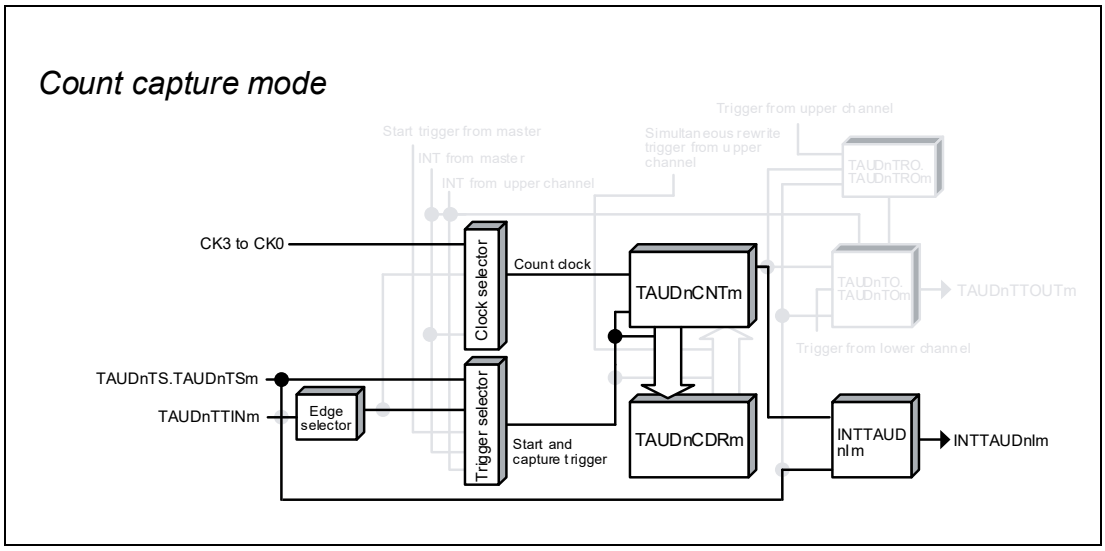


Figure 33.63 Block Diagram of TAUDnTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

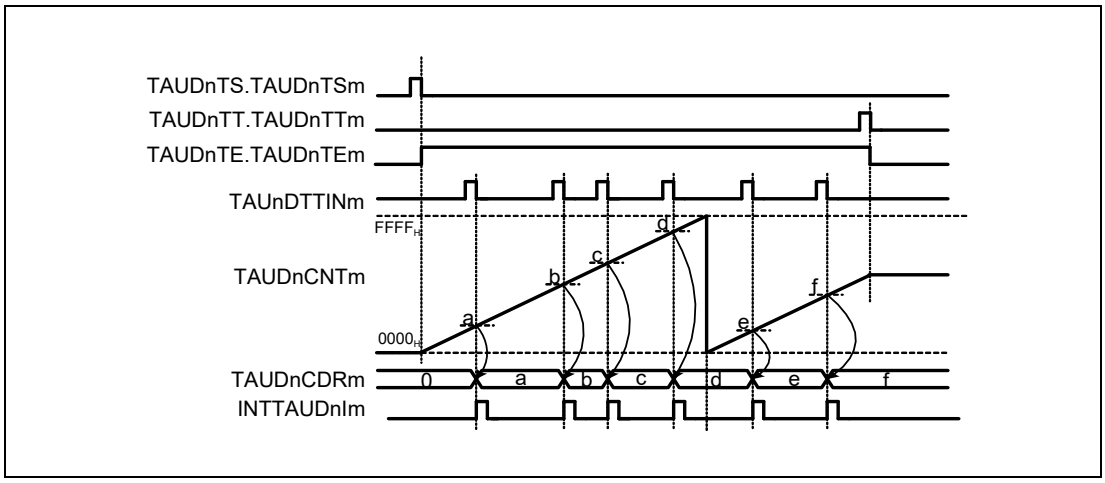


Figure 33.64 General Timing Diagram of TAUDnTTINm Input Position Detection Function

33.12.9.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.83 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as an external capture trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1011: Count capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.84 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

The channel output mode is not used by this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input position detection function. Therefore, these registers should be set to 0.

Table 33.85 Simultaneous Rewrite Settings for TAUDnTTINm Input Position Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.9.5 Operating Procedure for TAUDnTTINm Input Position Detection Function

Table 33.86 Operating Procedure for TAUDnTTINm Input Position Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.83, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Position Detection Function**, and **Table 33.84, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Position Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs.
During Operation	The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm edge is detected: • TAUDnCNTm transfers (captures) its own value to TAUDnCDRm. • Outputs INTTAUDnIm. • The counter is not cleared to 0000_H and TAUDnCNTm continues counting. Afterwards, this procedure is repeated. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.9.6 Specific Timing Diagrams

(1) Operation Stop and Restart

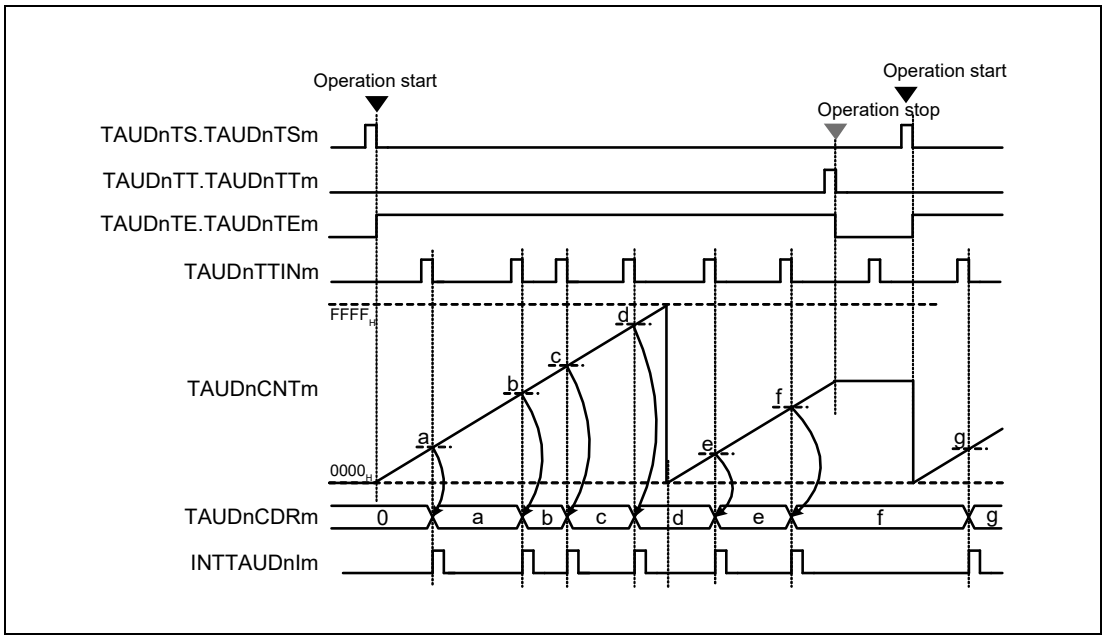


Figure 33.65 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- The counter can stop operating by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUDnTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

33.12.10 TAUDnTTINm Input Period Count Detection Function

33.12.10.1 Overview

Summary

This function measures the cumulative width of a TAUDnTTINm input signal.

Prerequisites

- The operating mode should be set to capture and gate count mode. (See **Table 33.87, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Period Count Detection Function**.)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter awaits a valid TAUDnTTINm input edge.

When a valid TAUDnTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUDnTTINm input stop edge is detected, the current TAUDnCNTm value is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The counter stops and retains its value (TAUDnCDRm + 1) until the next valid TAUDnTTINm input start edge is detected.

When the next valid TAUDnTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. TAUDnTTINm input signal is sampled at the frequency of an operation clock set by the TAUDnCMORm.TAUDnCKS[1:0] bits.
2. As this function is to measure the TAUDnTTINm input signal width, setting TAUDnTS.TAUDnTSm to 1 is disabled while TAUDnTE.TAUDnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDnTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDnTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

33.12.10.2 Equations

Cumulative TAUDnTTINm input width = count clock cycle × (TAUDnCDRm capture value + 1)

33.12.10.3 Block Diagram and General Timing Diagram

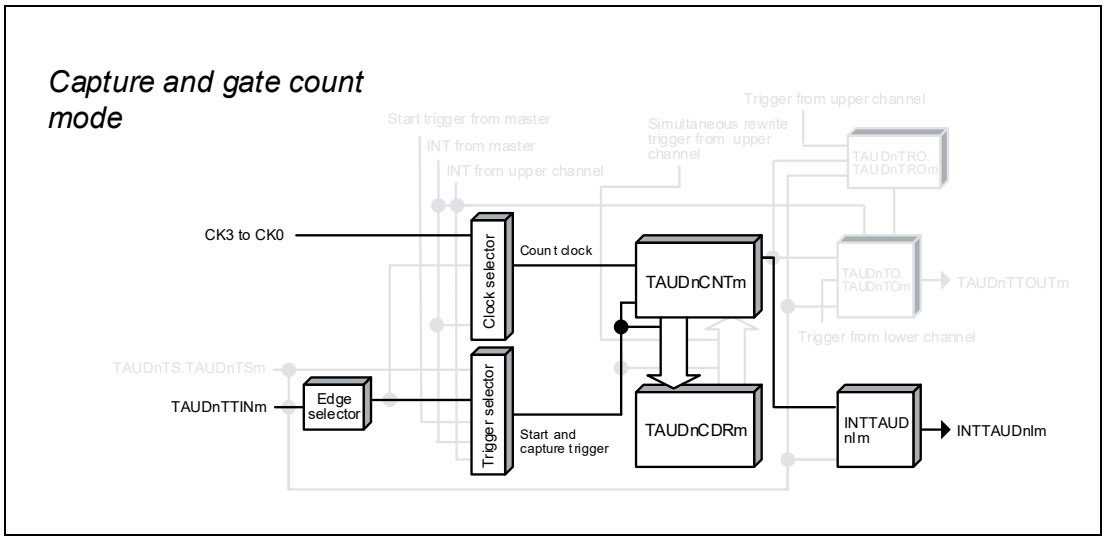


Figure 33.66 Block Diagram of TAUDnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

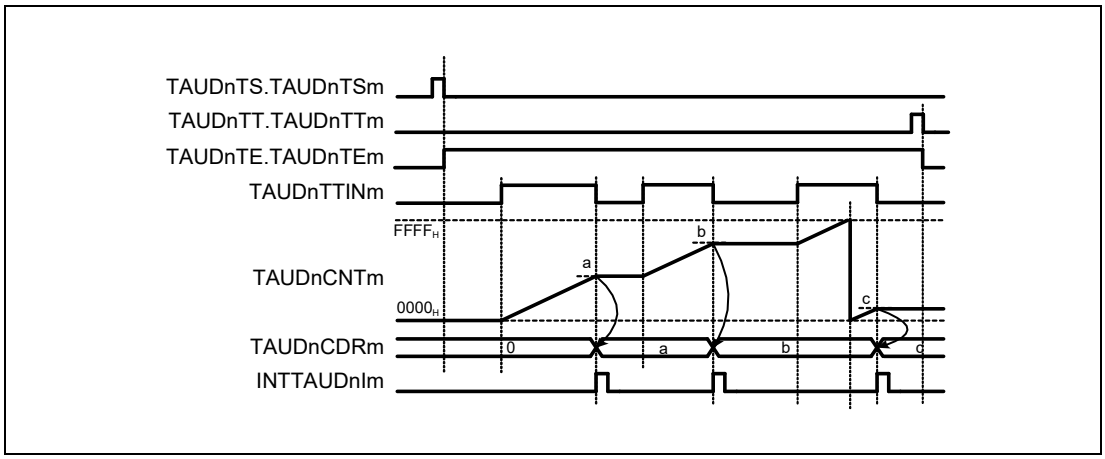


Figure 33.67 General Timing Diagram of TAUDnTTINm Input Period Count Detection Function

33.12.10.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.87 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1101: Capture and gate count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.88 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input period count detection function. Therefore, these registers should be set to 0.

Table 33.89 Simultaneous Rewrite Settings for TAUDnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.10.5 Operating Procedure for TAUDnTTINm Input Period Count Detection Function

Table 33.90 Operating Procedure for TAUDnTTINm Input Period Count Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.87, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Period Count Detection Function**, and **Table 33.88, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Period Count Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge.
During Operation	Detection of TAUDnTTINm edge The TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time.	When a TAUDnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUDnCNTm starts counting up from the stop value. When TAUDnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDnCDRm and INTTAUDnIm is generated. Counting stops at the “value transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDnTTINm start edge. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.10.6 Specific Timing Diagrams

(1) Operation stop and restart

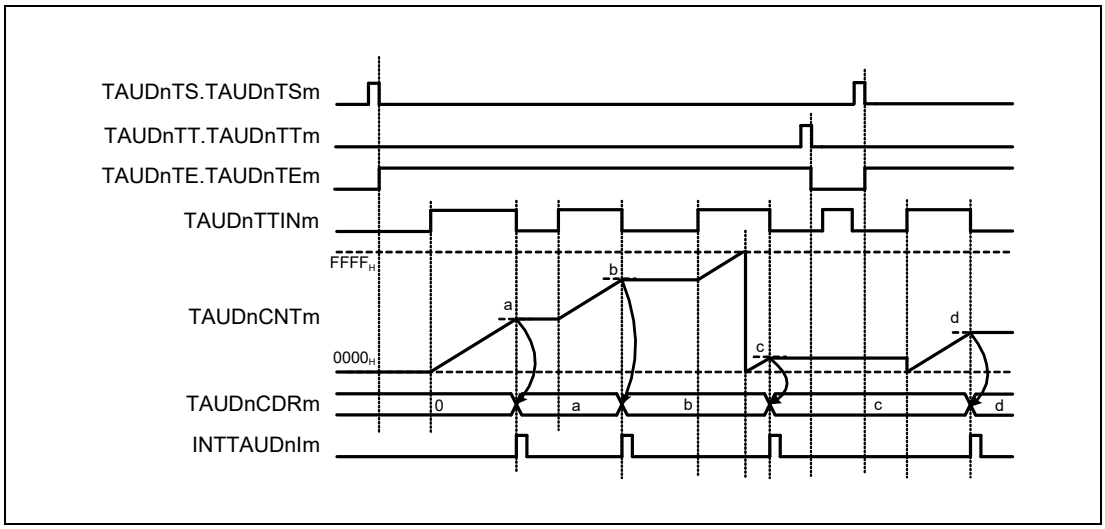


Figure 33.68 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter is stopped, valid TAUDnTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

33.12.11 TAUDnTTINm Input Pulse Interval Judgment Function

33.12.11.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUDnCNTm) and the value in the channel data register (TAUDnCDRm) when a TAUDnTTINm input pulse occurs. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge mode. See **Table 33.91, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Judgment Function**.
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDnTTINm valid edge is detected or TAUDnTS.TAUDnTSm is set to 1, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. TAUDnCNTm reloads the value of TAUDnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUDnTTINm valid edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:

- If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
- If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.

33.12.11.2 Block Diagram and General Timing Diagram

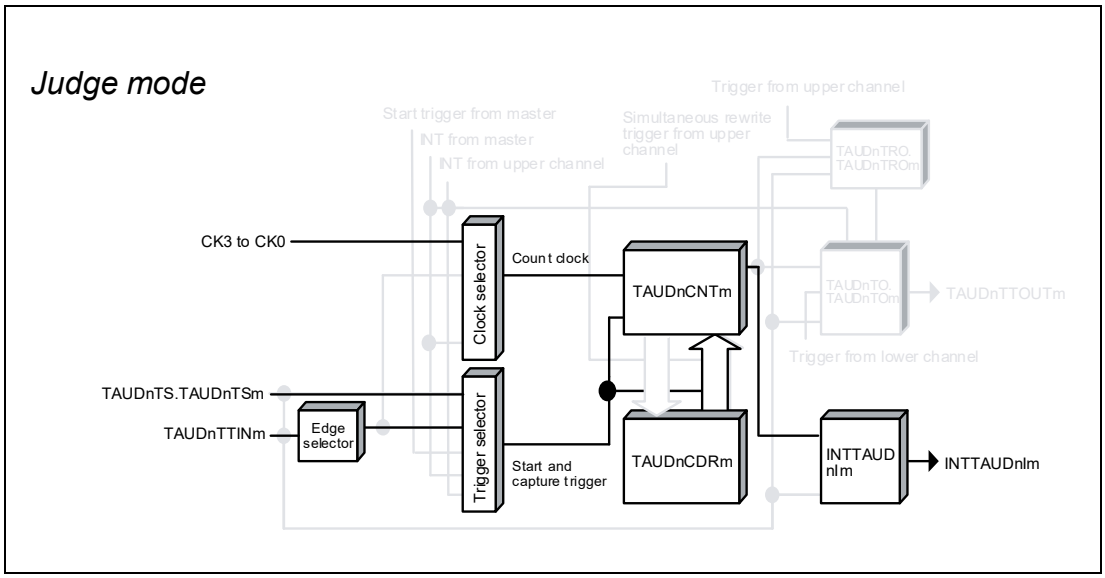


Figure 33.69 Block Diagram of TAUDnTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

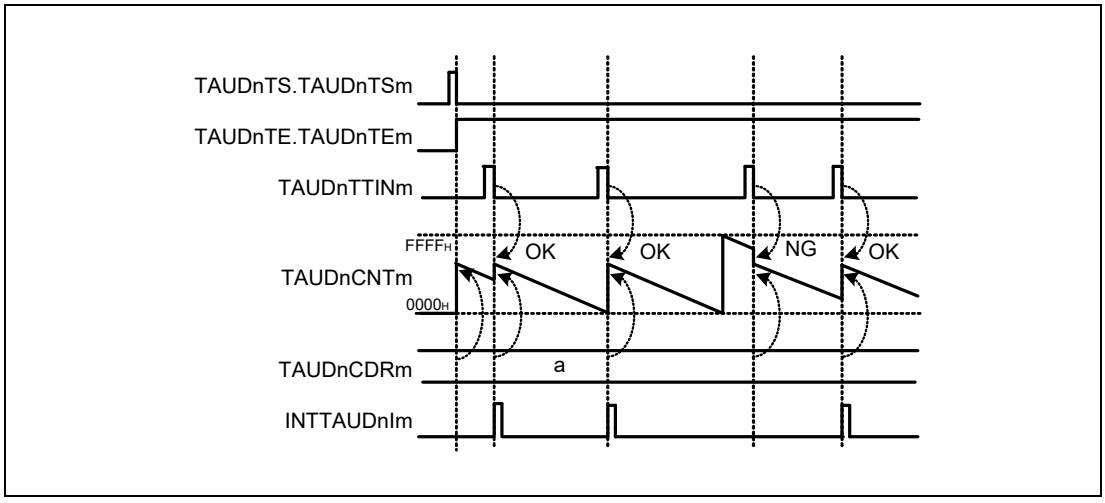


Figure 33.70 General Timing Diagram of TAUDnTTINm Input Pulse Interval Judgment Function

33.12.11.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.91 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDnTTINm m input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0001: Judge mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.92 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input pulse interval judgment function. Therefore, these registers should be set to 0.

Table 33.93 Simultaneous Rewrite Settings for TAUDnTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.11.4 Operating Procedure for TAUDnTTINm Input Pulse Interval Judgment Function

Table 33.94 Operating Procedure for TAUDnTTINm Input Pulse Interval Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.91, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Pulse Interval Judgment Function**, and **Table 33.92, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Pulse Interval Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm.
During Operation	The following register can be changed at any time: • TAUDnCDRm register	When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDnTTINm input edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDnTTINm input edge is detected, INTTAUDnIm is generated. If a TAUDnTTINm input edge is detected, then TAUDnCNTm starts to count down from the value of TAUDnCDRm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.12 TAUDnTTINm Input Signal Width Judgment Function

33.12.12.1 Overview

Summary

This function compares the count value (TAUDnCNTm) for the high or low level width of a TAUDnTTINm input signal and the TAUDnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDnIm.

Prerequisites

- The operating mode should be set to judge and one-count mode. (See **Table 33.95, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Judgment Function**.)
- TAUDnTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDnTTINm input start edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDnTTINm valid stop edge is detected, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. The counter TAUDnCNTm retains its value until the next valid TAUDnTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUDnTTINm stop edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. The counter then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

- The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:
  - If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
  - If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.
- The TAUDnCMURm.TAUDnTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B), TAUDnTTINm rising edge is used as a start edge and TAUDnTTINm falling edge as a stop edge.
  - For low width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 10_B), TAUDnTTINm falling edge is used as a start edge and TAUDnTTINm rising edge as a stop edge.
- Setting TAUDnTS.TAUDnTSm to 1 is prohibited during operation.

33.12.12.2 Block Diagram and General Timing Diagram

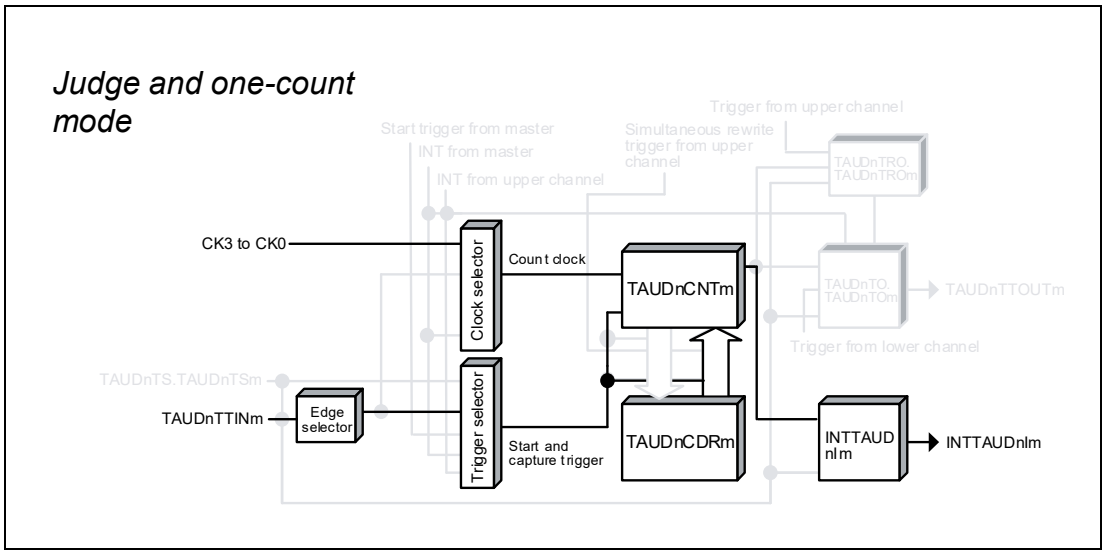


Figure 33.71 Block Diagram of TAUDnTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm (TAUDnCMORm.TAUDnMD0 = 0).
- TAUDnTTINm valid start edge = rising edge, TAUDnTTINm valid stop edge = falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

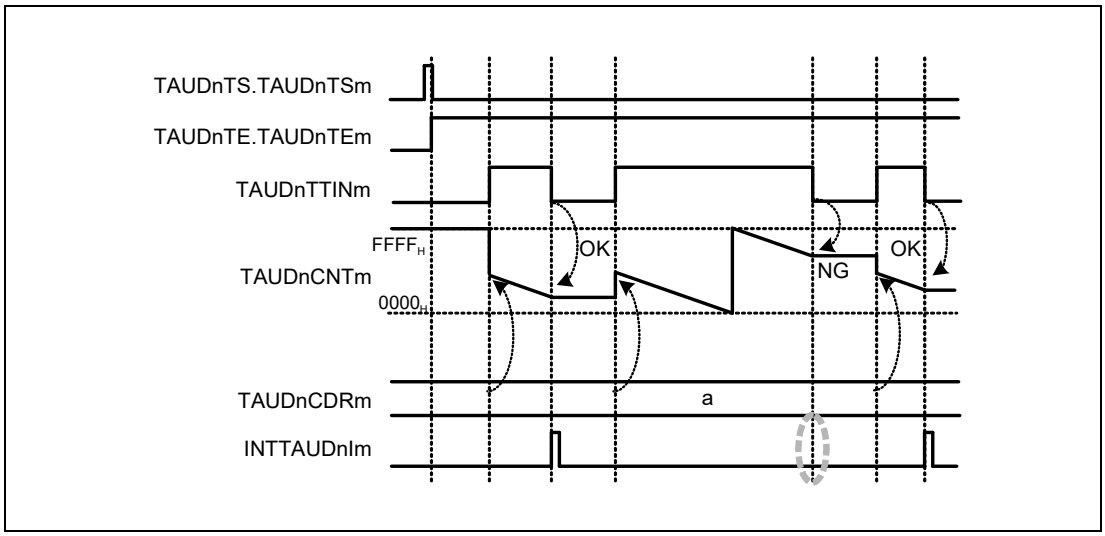


Figure 33.72 General Timing Diagram of TAUDnTTINm Input Signal Width Judgment Function

33.12.12.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.95 Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0111: Judge and one-count mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.96 Contents of the TAUDnCMURm Register for TAUDnTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDnTTINm input signal width judgment function. Therefore, these registers should be set to 0.

Table 33.97 Simultaneous Rewrite Settings for TAUDnTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.12.4 Operating Procedure for TAUDnTTINm Input Signal Width Judgment Function

Table 33.98 Operating Procedure for TAUDnTTINm Input Signal Width Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.95, Contents of the TAUDnCMORm Register for TAUDnTTINm Input Signal Width Judgment Function**, and **Table 33.96, Contents of the TAUDnCMURm Register for TAUDnTTINm Input Signal Width Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDnTTINm start edge.
During Operation	The following register can be changed at any time: • TAUDnCDRm register	Upon detection of a TAUDnTTINm start edge, TAUDnCNTm starts count down from the value of TAUDnCDRm. When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDnTTINm input stop edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDnTTINm input stop edge is detected, INTTAUDnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.13 Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

33.12.13.1 Overview

Summary

This function measures the width of an individual TAUDnTTINm input signal. An interrupt is generated if the TAUDnTTINm input width is longer than $FFFF_H + 1$.

Prerequisites

- The operation mode must be set to One-Count Mode (see **Table 33.99, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)**).
- TAUDnTTOUTm is not used for this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUDnTTINm input start edge is detected, TAUDnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDnTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDnTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

33.12.13.2 Block Diagram and General Timing Diagram

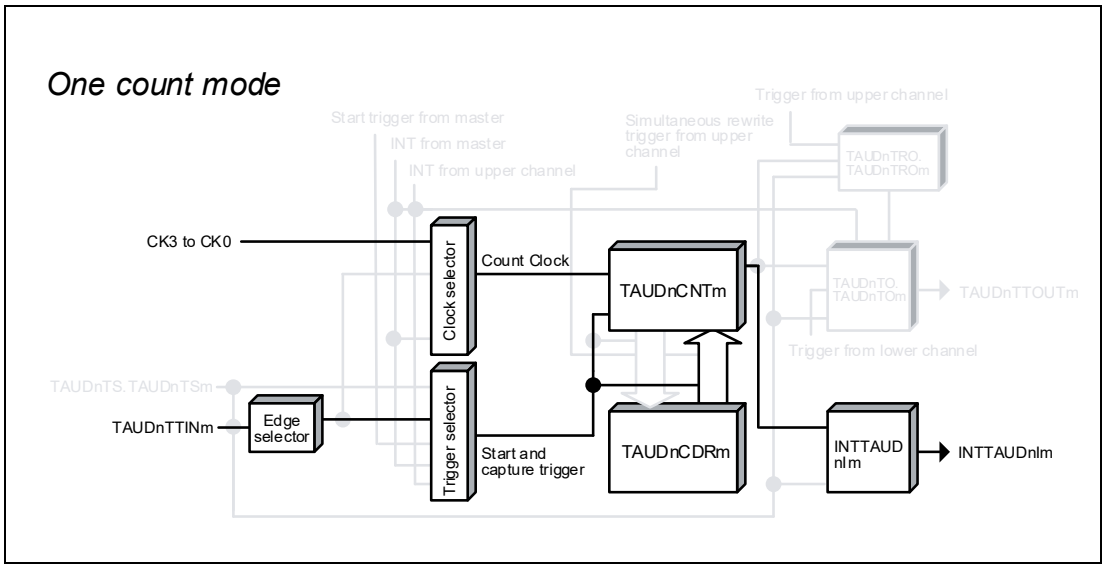


Figure 33.73 Block Diagram for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

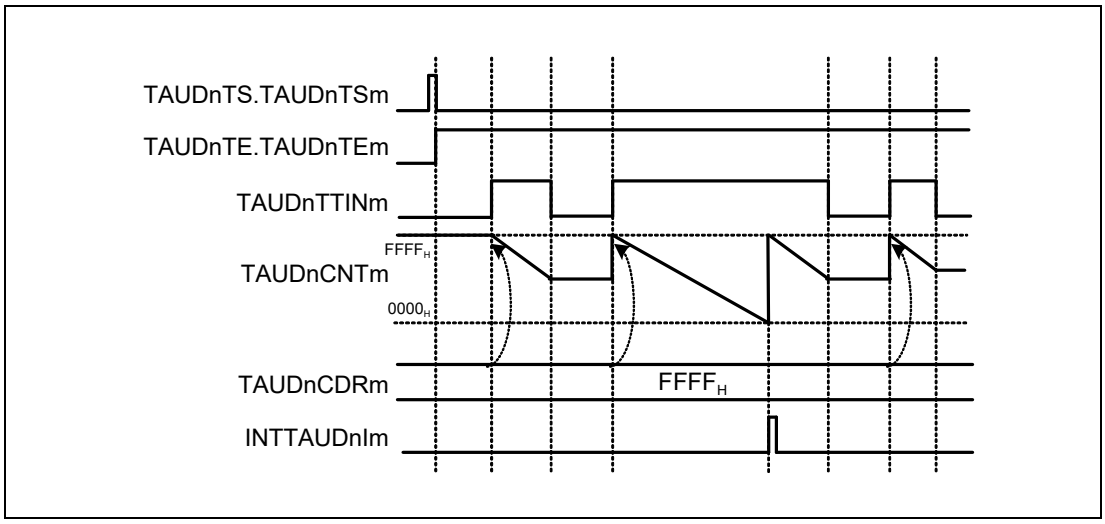


Figure 33.74 General Timing Diagram for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

33.12.13.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.99 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables the start trigger during operation

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.100 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 33.101 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

Table 33.102 Operating Procedure for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.99, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)**, and **Table 33.100, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDnTTINm Width Measurement)**. Set the value of TAUDnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$).
During Operation	The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_H: INTTAUDnIm is generated. When TAUDnTTINm input stop edge is detected during count operation: • TAUDnCNTm stops and retains its current value. When TAUDnTTINm input start edge is detected while the counter is stopped: • TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$) again, and continues to count down. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.14 Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

33.12.14.1 Overview

Summary

This function measures the cumulative width of a TAUDnTTINm input signal. If the cumulative TAUDnTTINm input width is longer than $FFFF_H$, an interrupt is generated and an overflow interrupt can be output.

Prerequisites

- The operation mode must be set to Gate Count Mode, (see **Table 33.103, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)**).
- TAUDnTTOUTm is not used with this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUDnTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is loaded to TAUDnCNTm and the counter continues to count down until a TAUDnTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by theTAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDnTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDnTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

33.12.14.2 Block Diagram and General Timing Diagram

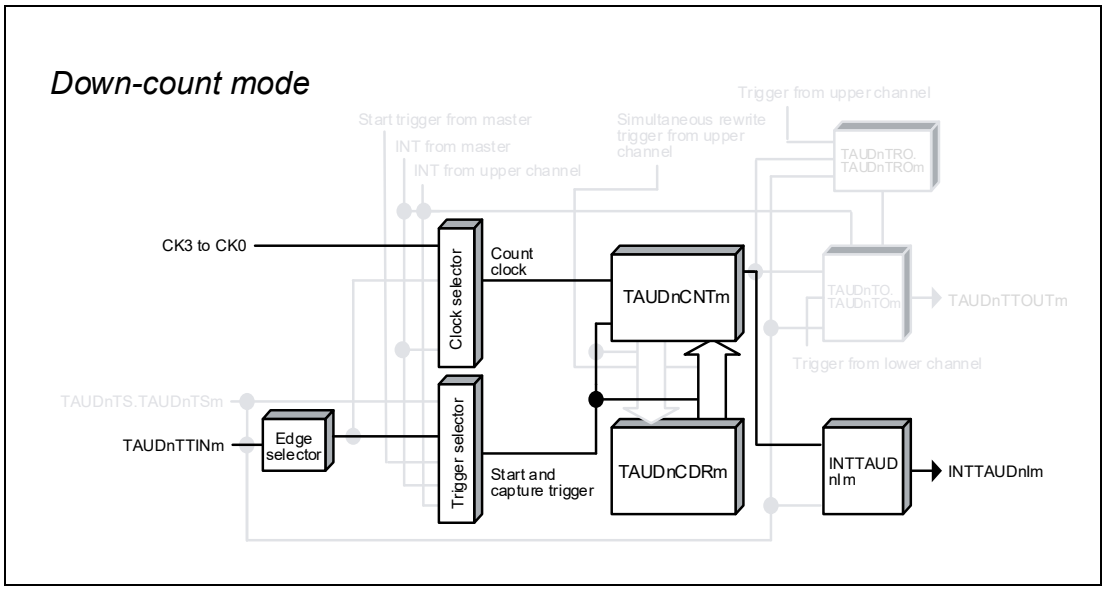


Figure 33.75 Block Diagram for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

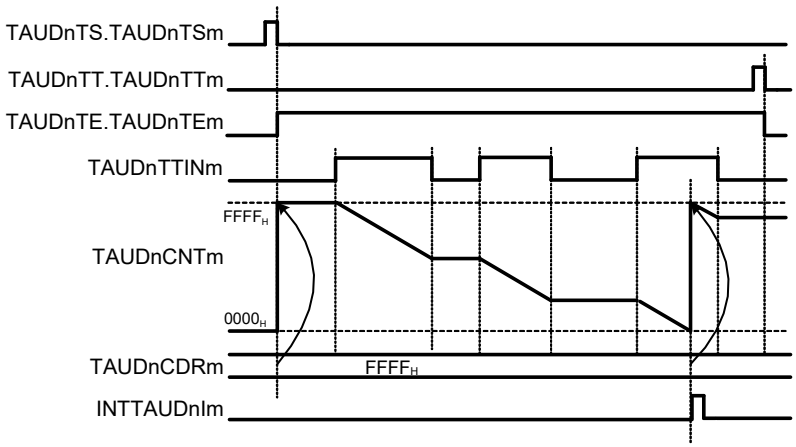


Figure 33.76 General Timing Diagram for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

33.12.14.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.103 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDnTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1100: Gate count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation

(2) TAUDnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.104 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 33.105 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.14.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

Table 33.106 Operating Procedure for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 33.103, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)**, and **Table 33.104, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDnTTINm Input Period Count Detection)**. Set the value of TAUDnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$).
During Operation	The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$) and continues to count down. When TAUDnTTINm input stop edge is detected during count operation: • TAUDnCNTm stops and retains the current value. When TAUDnTTINm input start edge is detected while the counter is stopped: • TAUDnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.12.15 One-Phase PWM Output Function

33.12.15.1 Overview

Summary

This function adds dead time to a TAUDnTTINm input signal. The resulting PWM signal is output via TAUDnTTOUTm of the channel and TAUDnTTOUTm of upper channels.

Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDnTDE.TAUDnTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See **Table 33.108, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**.)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWM output. (See **Section 33.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDnTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt occurs. The counter is reset to $FFFF_H$ and waits for the next valid TAUDnTTINm input start edge.

Table 33.107 TAUDnTTOUTm to which Dead Time is Added and State of TAUDnTTINm

TAUDnCMURm. TAUDnTIS[1:0]	TAUDnTOL. TAUDnTOLm	TAUDnTTOUTm to which Dead Time is Added	TAUDnTDL. TAUDnTDLm	TAUDnTTINm State when Added
10	0	TAUDnTTOUTm low	0	High
			1	Low
	1	TAUDnTTOUTm high	0	High
			1	Low
11	0	TAUDnTTOUTm low	0	Low
			1	High
	1	TAUDnTTOUTm high	0	Low
			1	High

Conditions

- TAUDnCMURm.TAUDnTIS[1:0] bits specify the type of width measurement:
  - TAUDnCMURm.TAUDnTIS[1:0] = 10_B: Uses both edges as valid edges for detection (Low width measurement).
  - TAUDnCMURm.TAUDnTIS[1:0] = 11_B: Uses both edges as valid edges for detection (High width measurement).
- The TAUDnTDL.TAUDnTDLm bit specifies the operation of TAUDnTTOUTm for each channel when an interrupt or valid TAUDnTTINm edge is detected on the lower channel:
  - If TAUDnTDL.TAUDnTDLm = 0, an interrupt is used as a TAUDnTTOUTm set trigger and a valid TAUDnTTINm edge as a TAUDnTTOUTm reset trigger.
  - If TAUDnTDL.TAUDnTDLm = 1, a valid TAUDnTTINm edge is used as a TAUDnTTOUTm set trigger and an interrupt as a TAUDnTTOUTm reset trigger.
- This function cannot make a forced restart.

33.12.15.2 Block Diagram and General Timing Diagram

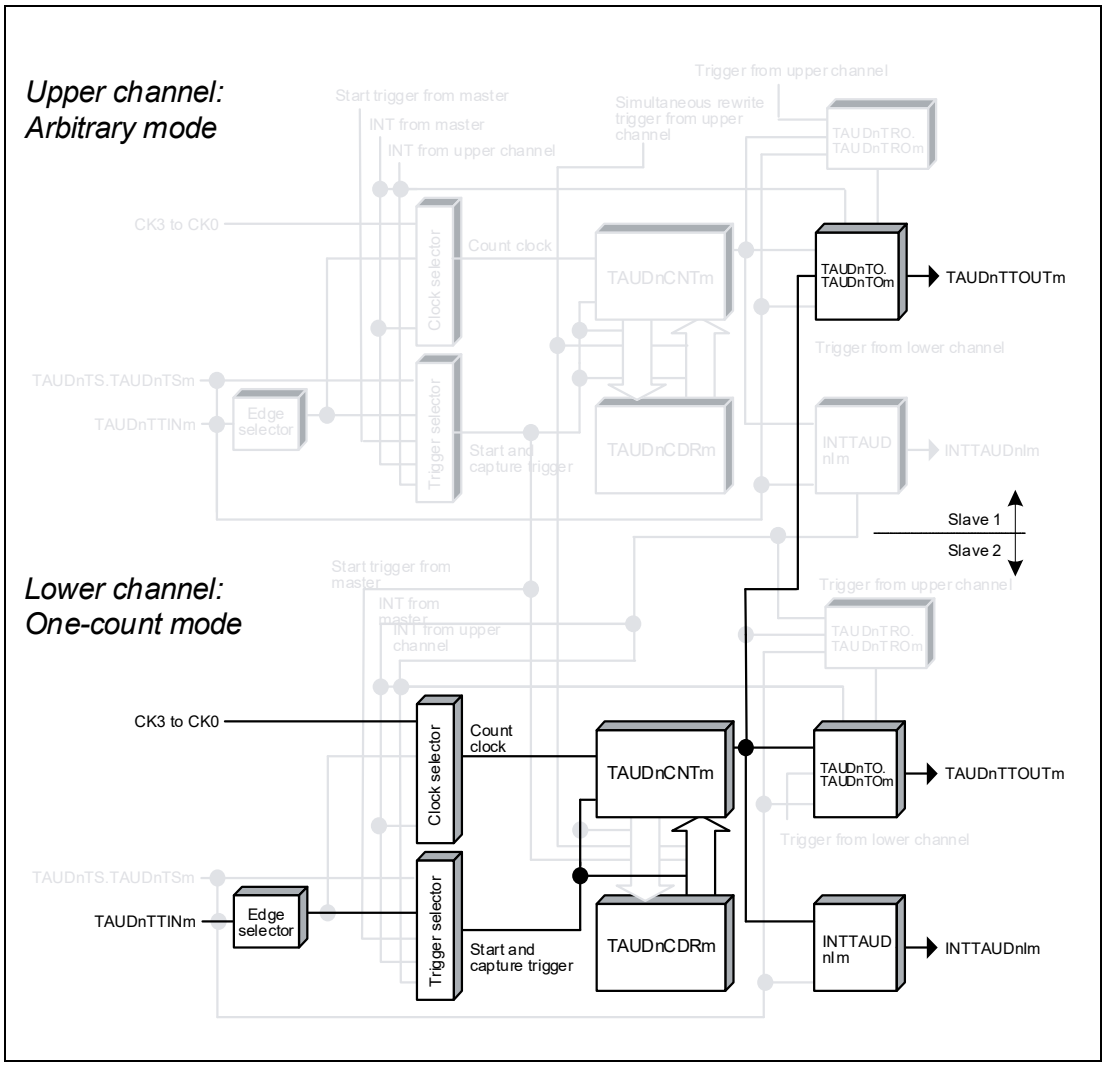


Figure 33.77 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

This setting considers a duty as an active high.

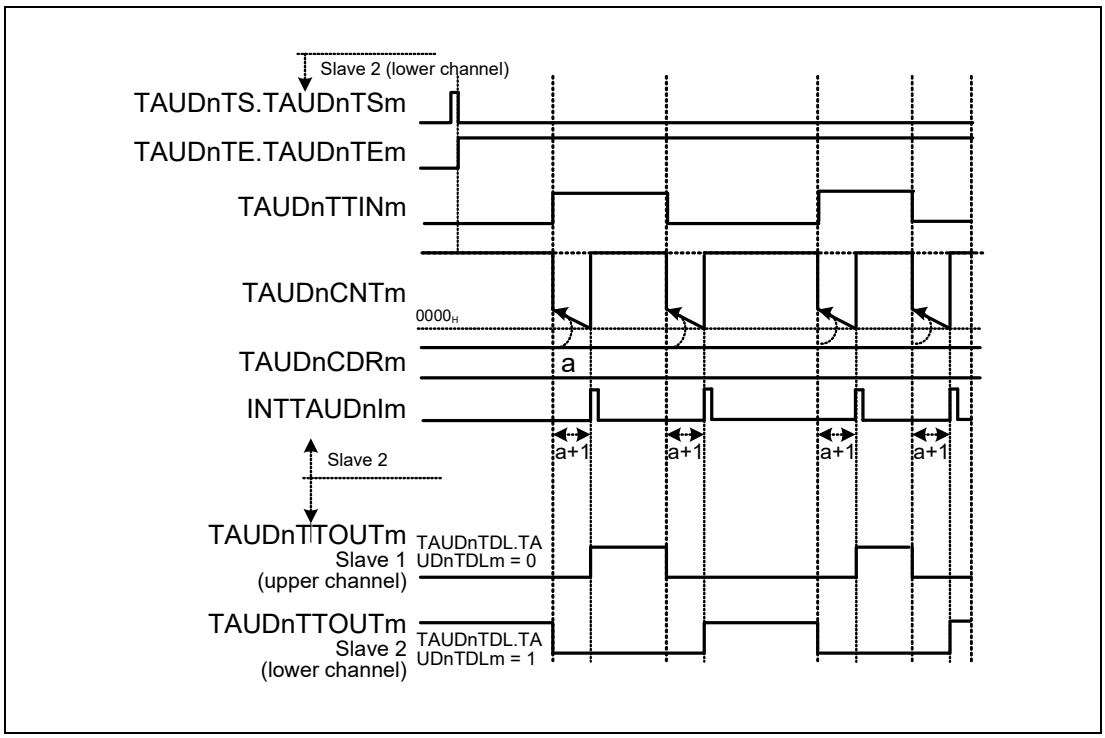


Figure 33.78 General Timing Diagram of One-Phase PWM Output Function

33.12.15.3 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.108 Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDnTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Lower Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.109 Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel Output Mode for Lower Channels

Table 33.110 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDnTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from upper channels.

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the one-phase PWM output function. Therefore, these registers should be set to 0.

Table 33.111 Simultaneous Rewrite Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.12.15.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

TAUDnCMORm register for upper channels can be set arbitrarily.

(2) TAUDnCMURm for Upper Channels

TAUDnCMURm register for upper channels can be set arbitrarily.

(3) Channel Output Mode for Upper Channels

Table 33.112 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDnTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from lower channels.

(4) Simultaneous Rewrite for Upper Channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

33.12.15.5 Operating Procedure for One-phase PWM Output Function

Table 33.113 Operating Procedure for One-phase PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Table 33.108, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**, and **Table 33.109, Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function**. Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Section 33.12.15.4, Register Settings for Upper Channels**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 33.110, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTOE.TAUDnTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUDnTS.TAUDnTSm = 1 for slave channel 2. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDnTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 (slave channel 2) and TAUDnCNTm waits for detection of TAUDnTTINm start edge. TAUDnCNTm loads TAUDnCDRm value.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm of slave channel 2 counts down. When the counter reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCNTm stops counting. TAUDnTTOUTm is changed by a TAUDnTTINm edge detection signal and slave channel 2 INTTAUDnIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm = 1 for slave channel 2. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDnTTOUTm retain their current values.

Restart Operation

33.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDnTRO.TAUDnTROm bit in real time.

33.13.1 Real-Time Output Function Type 1

33.13.1.1 Overview

Summary

This function outputs a value of the TAUDnTRO.TAUDnTROm bit from TAUDnTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDnTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 33.114, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 33.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter of the upper channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of the data register of the upper channel (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUDnIm is generated and TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1). TAUDnCNTm then reloads the TAUDnCDRm value to continue operation subsequently.

The TAUDnTTOUTm signal changes only when an interrupt is generated, and when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.13.1.2 Equations

INTTAUDnIm generation cycle = count clock cycle × (TAUDnCDRm value + 1)

33.13.1.3 Block Diagram and General Timing Diagram

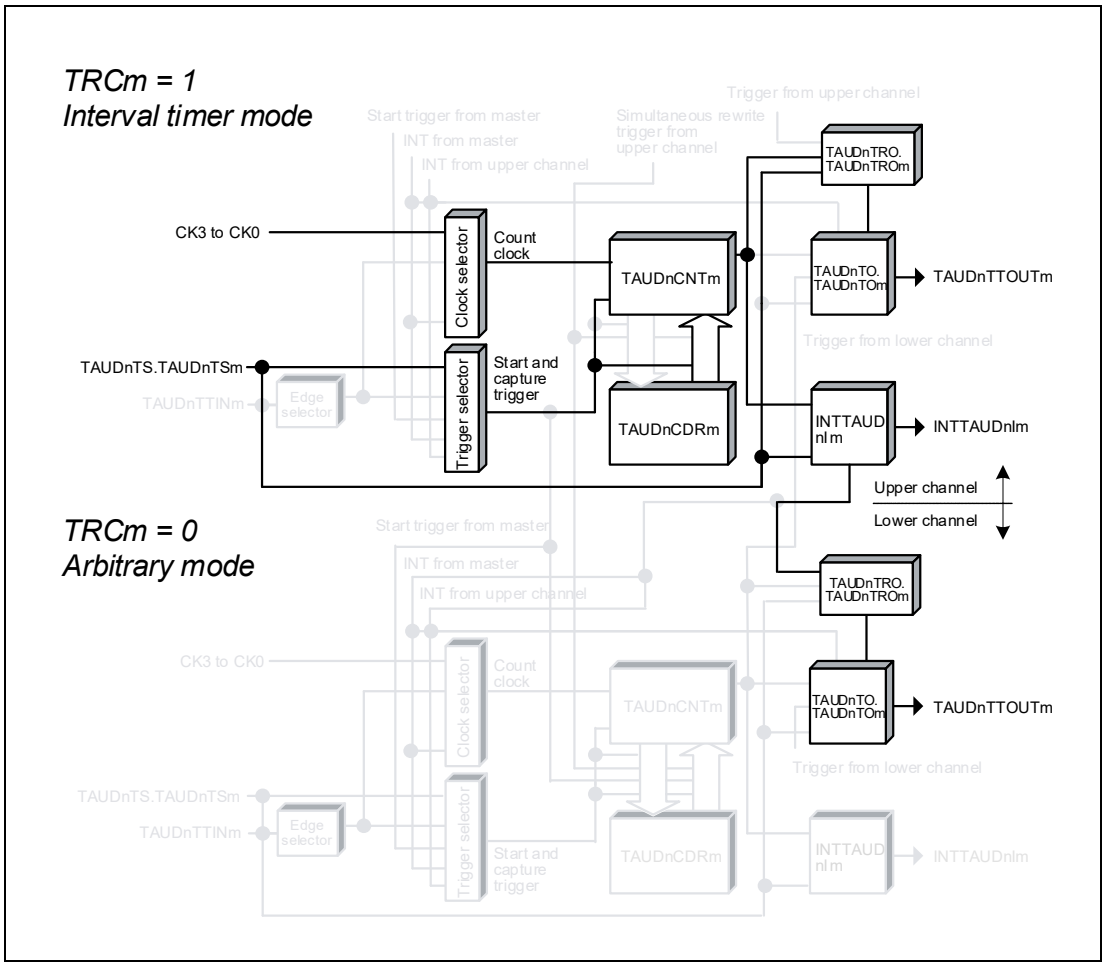


Figure 33.79 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

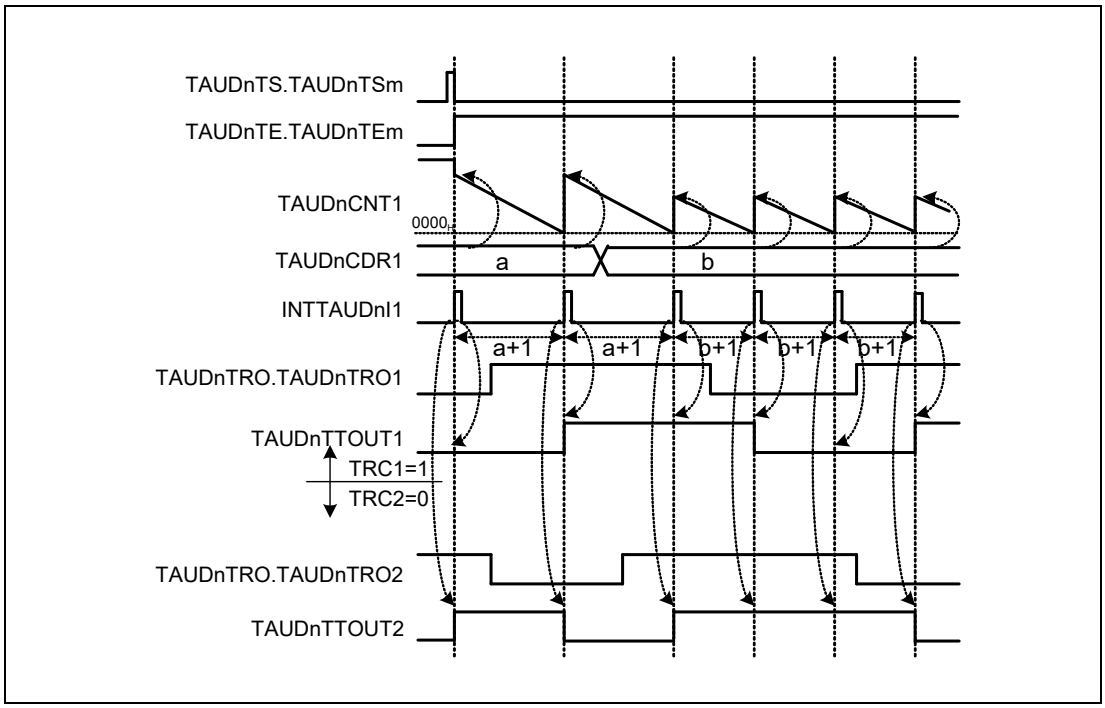


Figure 33.80 General Timing Diagram of Real-Time Output Function Type 1

33.13.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.114 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS [1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.115 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Upper Channels

Table 33.116 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

Table 33.117 Simultaneous Rewrite Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.13.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for Lower Channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel Output Mode for Lower Channels

Table 33.118 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

33.13.1.6 Operating Procedure for Real-Time Output Function Type 1

Table 33.119 Operating Procedure for Real-Time Output Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 33.114, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**, and **Table 33.115, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 33.13.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register (only channels with TAUDnTRC.TAUDnTRCm = 1) Set channel output mode by setting the control bits as described in **Table 33.116, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 33.118, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	TAUDnCDRm and TAUDnTRO.TAUDnTROm can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated. • TAUDnTTOUTm outputs the current value of the real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. Both TAUDnCNTm and TAUDnTTOUTm retain their current values.

Restart Operation

33.13.1.7 Specific Timing Diagrams

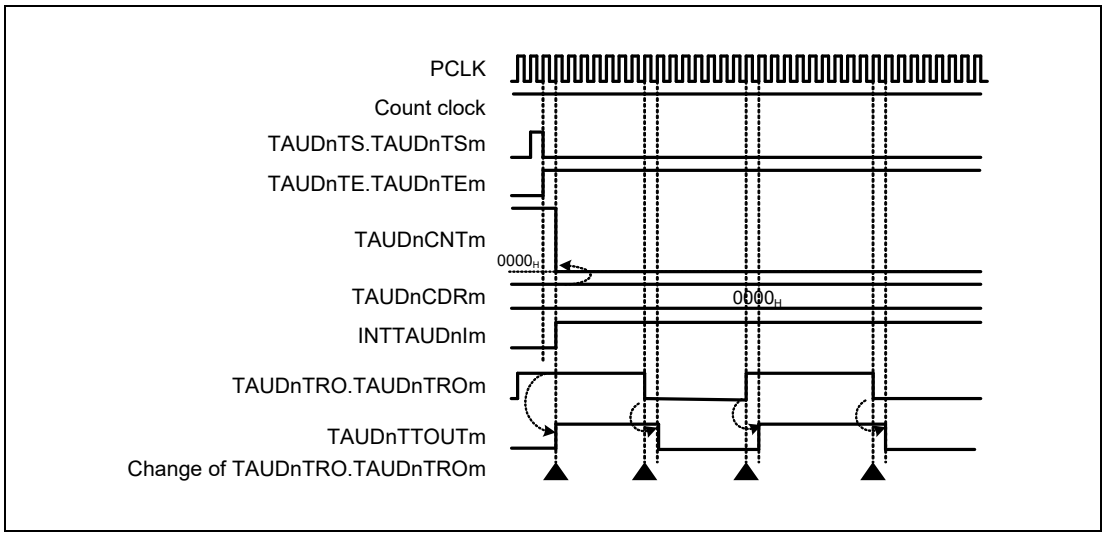


Figure 33.81 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1

- The value of TAUDnTTOUTm changes according to the setting of TAUDnTRO.TAUDnTROm with a delay of one PCLK cycle.

33.13.2 Real-Time Output Function Type 2

33.13.2.1 Overview

Summary

This function outputs the value of TAUDnTRO.TAUDnTROm bit from TAUDnTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated when a valid TAUDnTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDnTTOUTm control of the other channels.
- The operating mode for the upper channel should be set to capture mode. (See **Table 33.120, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 33.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter for upper channels is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The counter starts to count up.

When a valid TAUDnTTINm input edge is generated on one of upper channels, an interrupt occurs and TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1).

The TAUDnTTOUTm signal changes only when an interrupt is generated, and when TAUDnTTOUTm value is different from the current value of TAUDnTRO.TAUDnTROm during the occurrence of the interrupt.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel’s TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.

- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.13.2.2 Block Diagram and General Timing Diagram

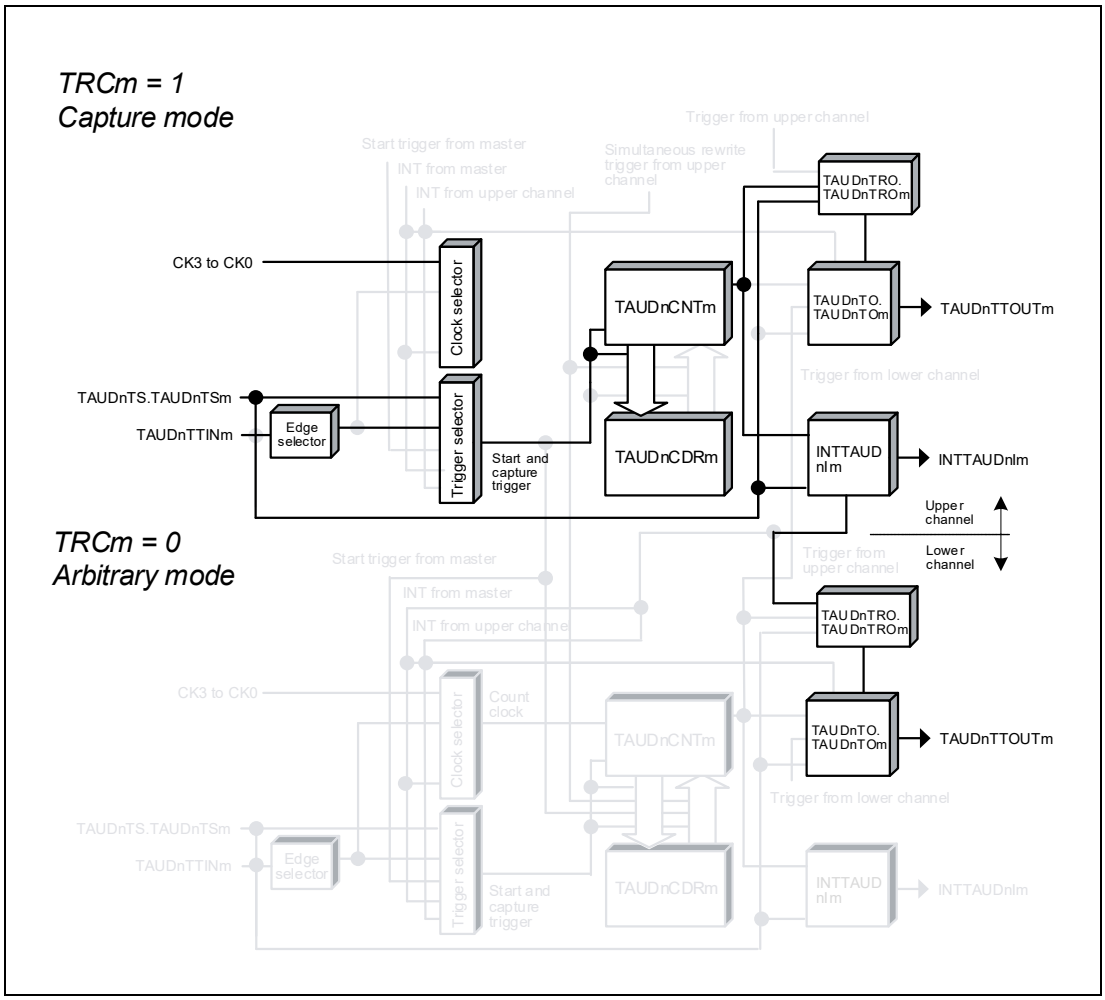


Figure 33.82 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)

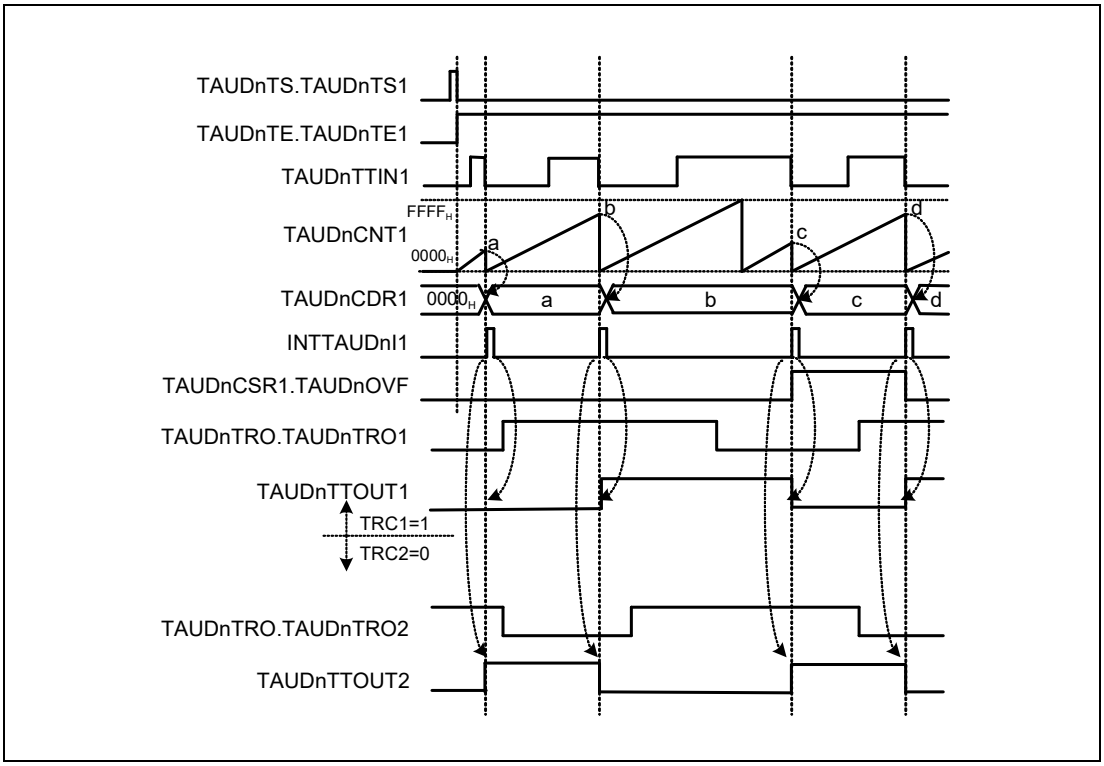


Figure 33.83 General Timing Diagram of Real-Time Output Function Type 2

33.13.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnSOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.120 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDnTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.121 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Upper Channels

Table 33.122 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Upper Channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

Table 33.123 Simultaneous Rewrite Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.13.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for Lower Channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel Output Mode for Lower Channels

Table 33.124 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Lower Channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

33.13.2.5 Operating Procedure for Real-Time Output Function Type 2

Table 33.125 Operating Procedure for Real-Time Output Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 33.120, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**, and **Table 33.121, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2**. Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Section 33.13.2.4, Register Settings for Lower Channels**. The TAUDnCDRm register functions as a capture register (only channels with TAUDnTRC.TAUDnTRCm = 1). Set channel output mode by setting the control bits as described in **Table 33.122, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 33.124, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	TAUDnTRO.TAUDnTROm can be changed at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDnTTINm input edge is detected: • TAUDnCNTm captures the TAUDnCDRm value, and the counter is cleared to 0000_H. • INTTAUDnIm is generated. • When the TAUDnTTINm input valid edge is detected immediately after the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is set to 1. When detected before the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is cleared to 0. TAUDnTTOUTm outputs the current value of real-time output bit TAUDnTRO.TAUDnTOROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm, TAUDnCSRm.TAUDnOVF, and TAUDnTTOUTm retain their current values.

Restart Operation

33.13.2.6 Specific Timing Diagrams

(1) Operation Start and Stop

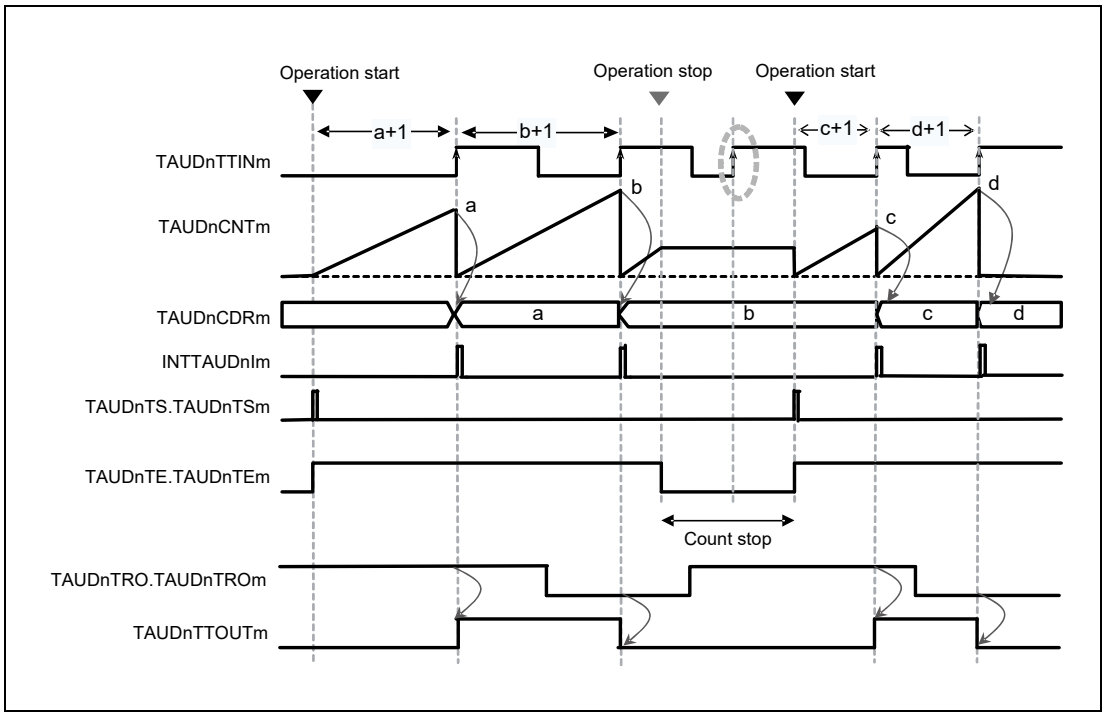


Figure 33.84 Operation Start and Stop (TAUDnCMORm.TAUDnMD0 = 0)

- When TAUDnTS.TAUDnTSm is set to 1, the counter starts counting up.
- When a valid input edge is detected, the current value of the counter is written to the data register (TAUDnCDRm) and an interrupt is generated.
- TAUDnTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) and the counter resets and starts to count up again.
- The TAUDnTTOUTm signal only changes when an interrupt is generated, and then only when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.
- If the counter is stopped (TAUDnTE.TAUDnTEm = 0), valid input edges are ignored and no interrupt is generated.

33.14 Independent Channel Simultaneous Rewrite Functions

This section describes functions that carry out simultaneous rewrite.

33.14.1 Simultaneous Rewrite Trigger Generation Function Type 1

33.14.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 33.126, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**.)
- For the operating mode that can be set for lower channels, see **Table 33.43, Channel Functions and the Methods They Use for Simultaneous Rewrite**.
- TAUDnTTOUTm is not used for any channel in this function.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDnCDRm buf) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000_H, an interrupt occurs on the channel. The current value of the corresponding TAUDnCDRm buffer is loaded into TAUDnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUDnRDC.TAUDnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit should be set to 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.14.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUDnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[Triangle PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUDnCDRm + 1 and value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1 must be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

33.14.1.3 Block Diagram and General Timing Diagram

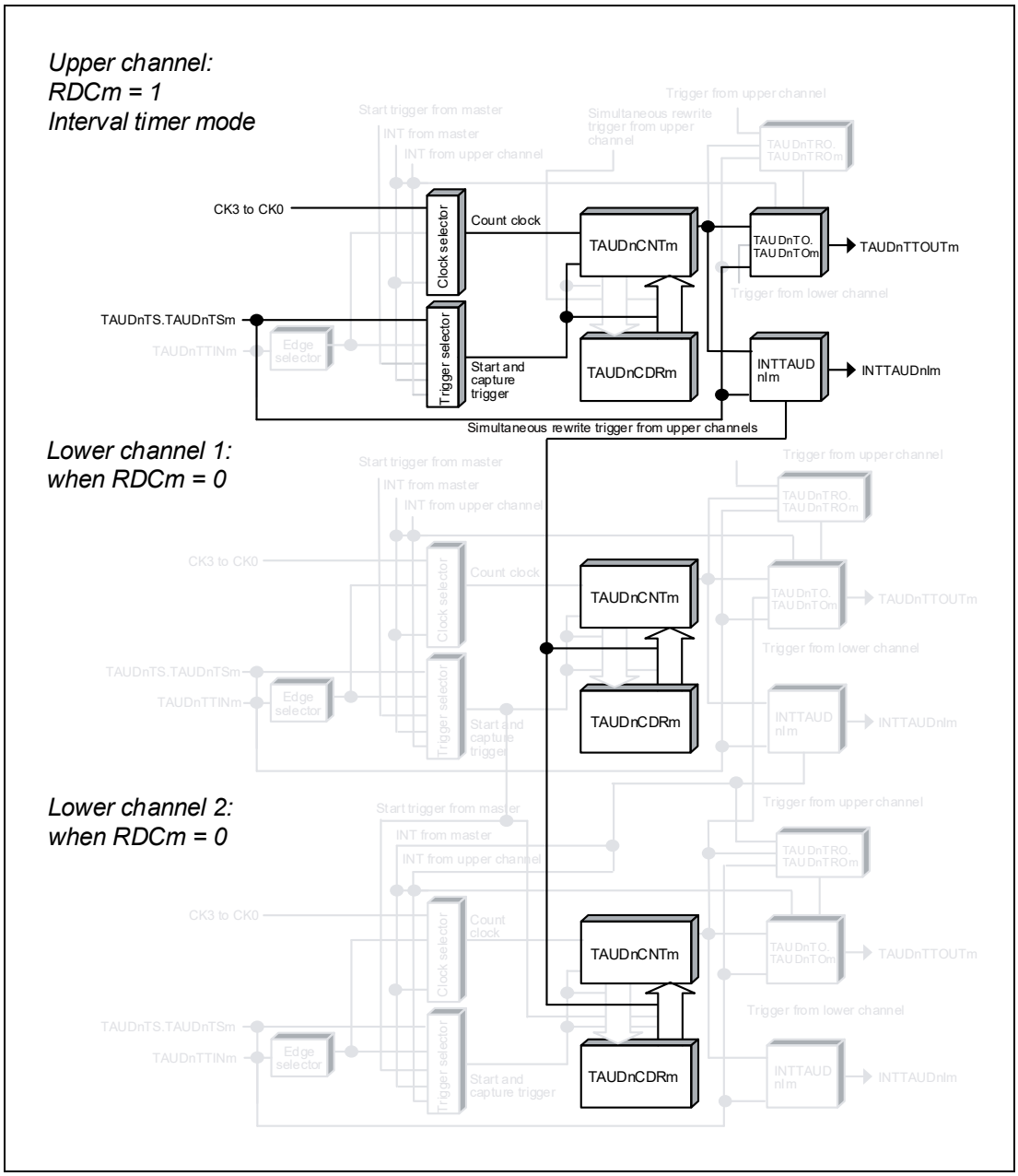


Figure 33.85 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

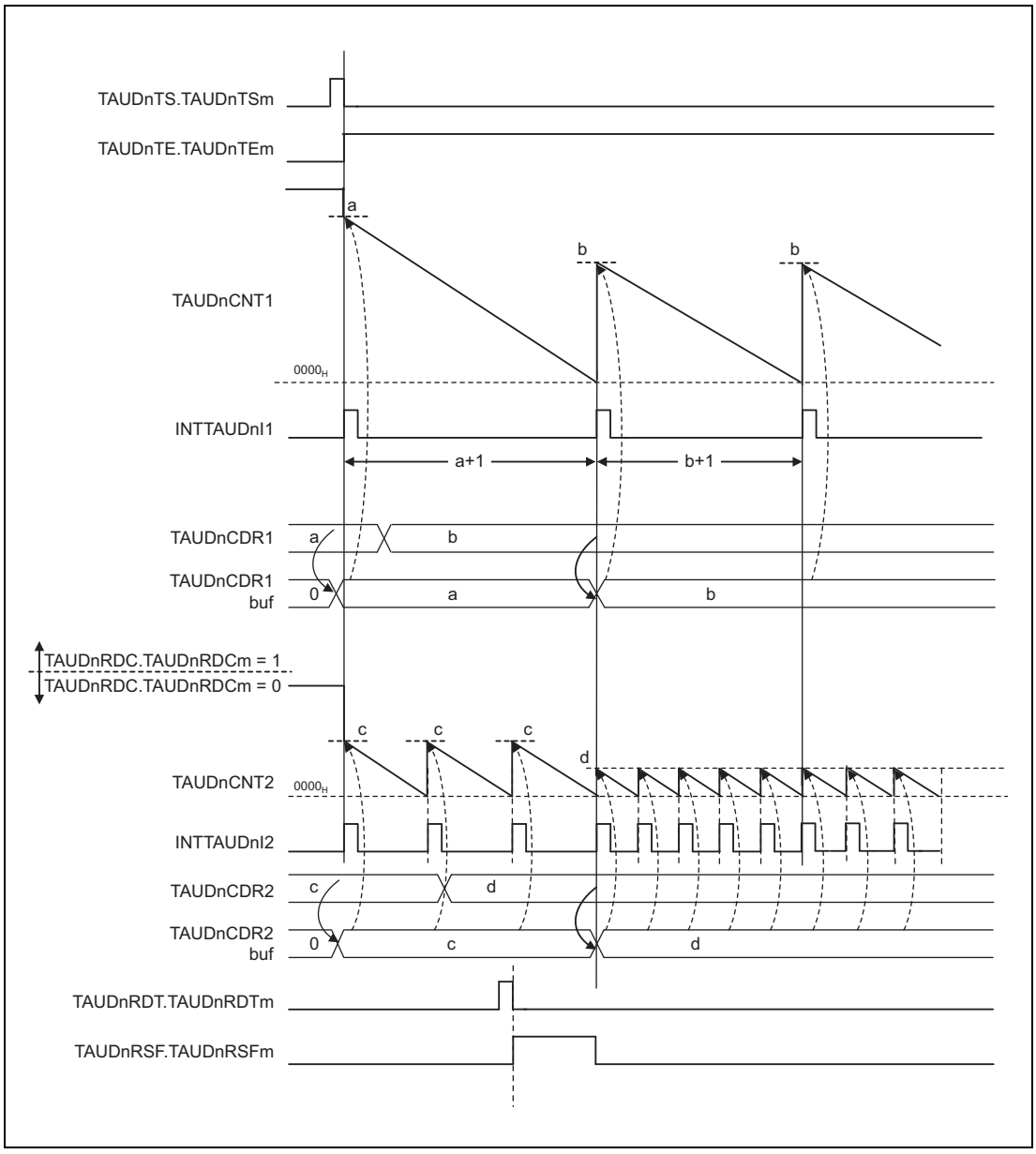


Figure 33.86 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

33.14.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.126 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.127 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Upper Channels

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous Rewrite for Upper Channels

Table 33.128 Simultaneous Rewrite Settings for Upper Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

33.14.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

TAUDnCMORm register for lower channels must follow the TAUDnCMORm register settings in the operating mode which can be set. (See **Table 33.43, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(2) TAUDnCMURm for Lower Channels

TAUDnCMURm register for lower channels must follow the TAUDnCMURm register settings in the operating mode which can be set. (See **Table 33.43, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(3) Channel Output Mode for Lower Channels

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 33.42, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous Rewrite for Lower Channels

Table 33.129 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 33.130 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Table 33.126, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**, and **Table 33.127, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 33.14.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated.
During Operation	TAUDnRDT.TAUDnRDTm and TAUDnCDRm.TAUDnCDR are changeable. TAUDnRSF.TAUDnRSFm can be always read.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated. If INTTAUDnIm is generated on the channel where TAUDnRDC.TAUDnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.14.2 Simultaneous Rewrite Trigger Generation Function Type 2

33.14.2.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. In this function, the interrupt is generated when a valid TAUDnTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operation mode of the upper channel must be set to Capture Mode (see **Table 33.131, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**).
- For the operation mode that can be set for a lower channel, see **Table 33.43, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter for the upper channel starts to count up, and then the counter for lower channels start to count according to the selected operating mode.

When a TAUDnTTINm input edge occurs on the upper channel, an interrupt is generated. The trigger is detected by the lower channel(s), which then also generate an interrupt.

When TAUDnRDC.TAUDnRDCm = 1 on the upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, an interrupt is generated when the function starts. For details see **Section 33.9, TAUDnTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

33.14.2.2 Block Diagram and General Timing Diagram

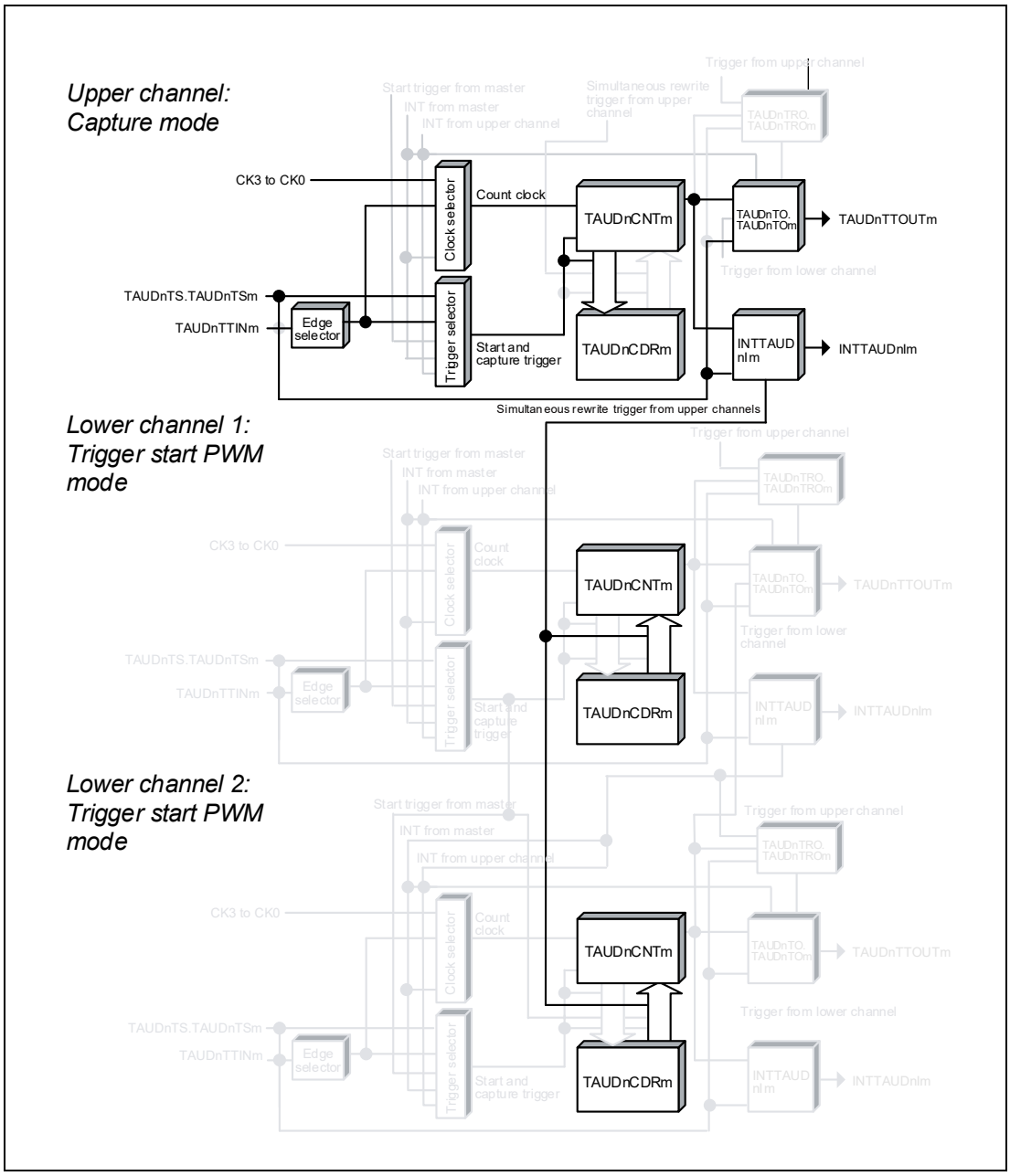


Figure 33.87 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- Upper channel (CH1) generates simultaneous rewrite trigger.

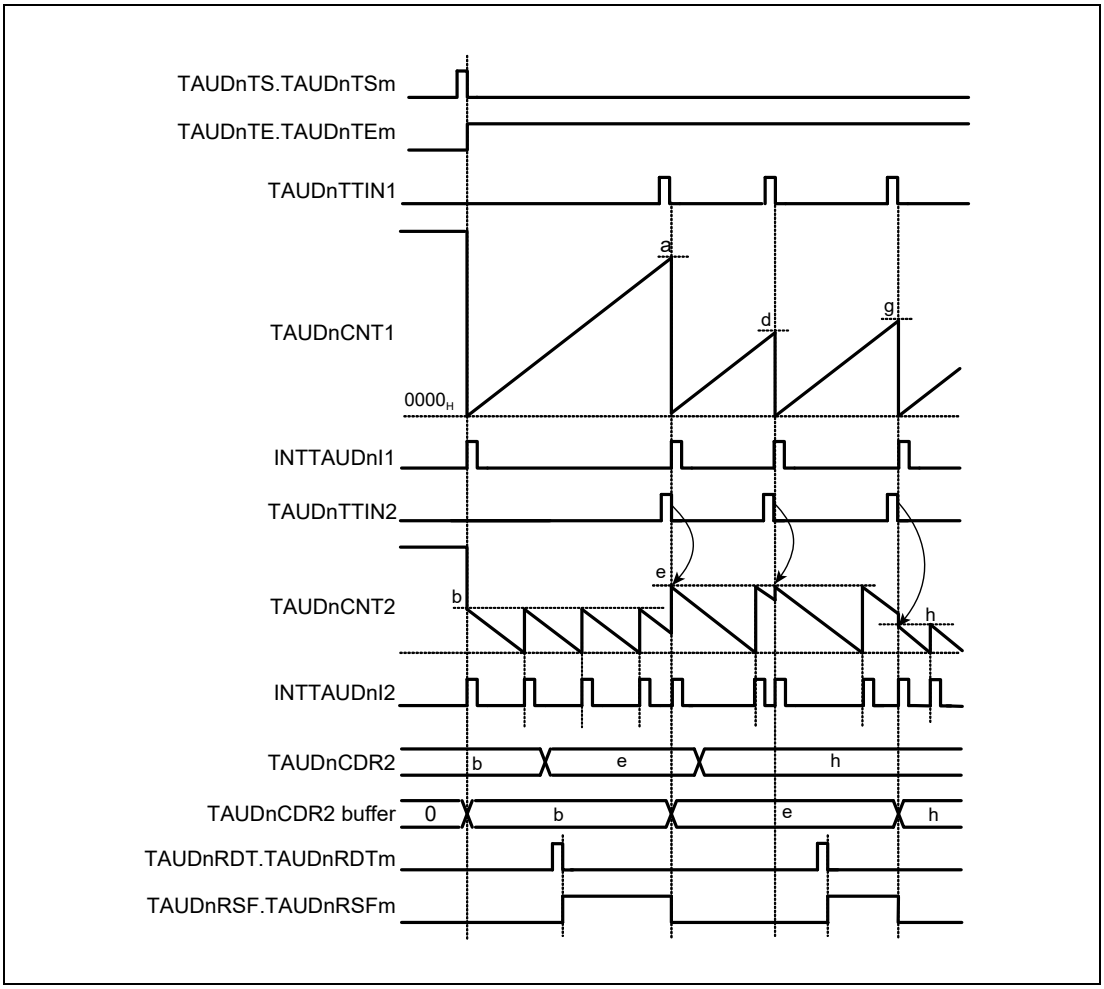


Figure 33.88 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

33.14.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for Upper Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.131 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDnTTINm input signal is used as the external capture trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Upper Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.132 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Upper Channels

The channel output mode is not used by this function.

(4) Simultaneous Rewrite for Upper Channels

Table 33.133 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

33.14.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for Lower Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.134 Contents of the TAUDnCMORm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as the start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for Lower Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.135 Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for Lower Channels

Output can be made according to the trigger start PWM mode setting.

(4) Simultaneous Rewrite for Lower Channels

Table 33.136 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

Table 33.137 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

		Operation	TAUDn Status
Restart Operation	Initial Channel Setting	Set the TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Table 33.131, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 33.132, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. Set the TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Table 33.134, Contents of the TAUDnCMORm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 33.135, Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
	During Operation	TAUDnRDT.TAUDnRDTm can be set at any time. TAUDnRSF.TAUDnRSFm can be read at any time.	TAUDnCNTm counts up from 0000_H. When a TAUDnTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm and returns to 0000_H. • INTTAUDnIm is generated. Simultaneous rewrite is controlled when INTTAUDnIm is generated from the channel where TAUDnRDC.TAUDnRDCm is set to 1. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and it retains its current value.

33.15 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit D. For a general overview of synchronous channel operation, see **Section 33.2, Overview**.

This section describes functions that generate PWM signals at regular intervals.

33.15.1 PWM Output Function

33.15.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUDnTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 33.138, Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operating mode for the slave channels should be set to one-count mode. (See **Table 33.141, Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUDnTTOUTm is not used with the master channel of this function.
- The channel output mode for slave channels should be set to Synchronous Channel Output Mode 1. (See **Section 33.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm, and the counter starts counting down from the TAUDnCDRm value. If an INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:
  When the master channel counter reaches 0000_H and the pulse cycle time has passed, INTTAUDnIm is generated. The counter loads TAUDnCDRm value into TAUDnCNTm and counts down.
- Slave channel:
  When INTTAUDnIm is generated on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts counting down from the TAUDnCDRm value. TAUDnTTOUTm signal is set to the active level.
  When the counter reaches to 0000_H (duty time has elapsed), INTTAUDnIm is generated and a TAUDnTTOUTm signal is set to an inactive level. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm (start of the next pulse cycle) of the master channel.

The counter can stop operating by setting the TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 33.6, Simultaneous Rewrite**.

33.15.1.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)) × 100

– Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

33.15.1.3 Block Diagram and General Timing Diagram

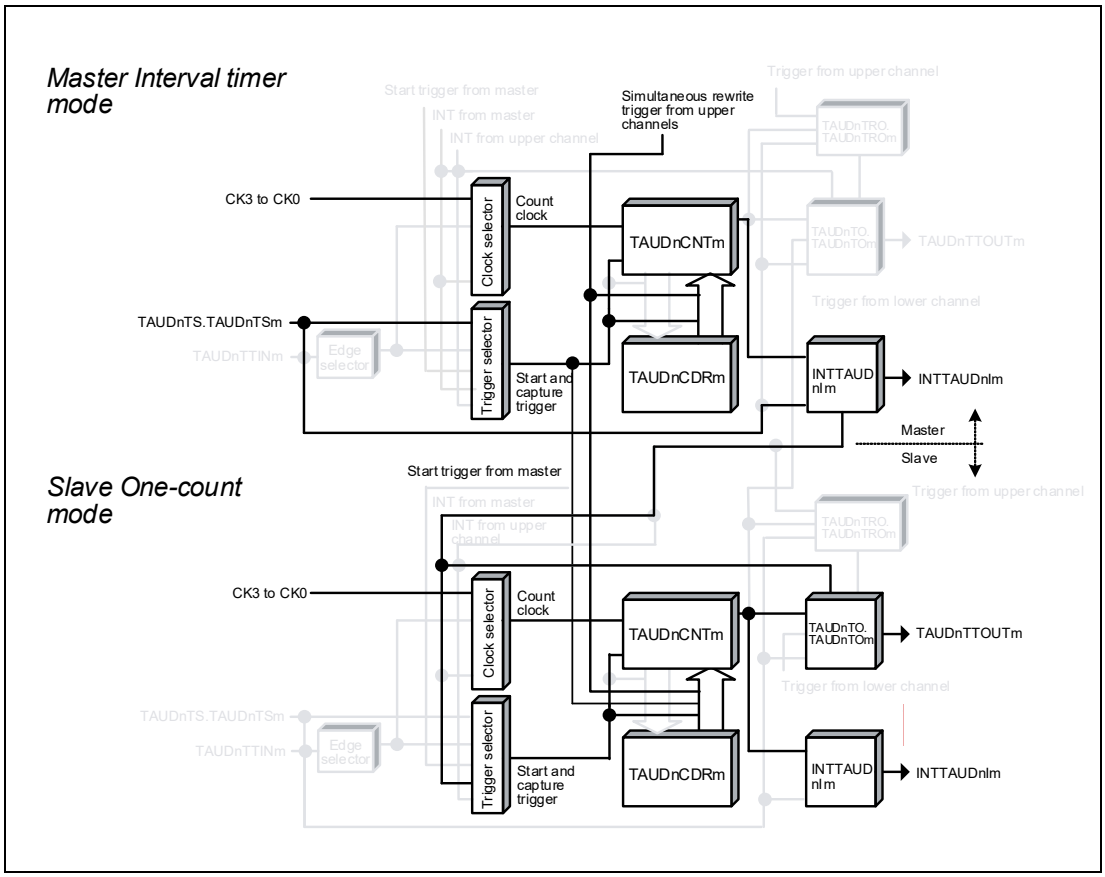


Figure 33.89 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

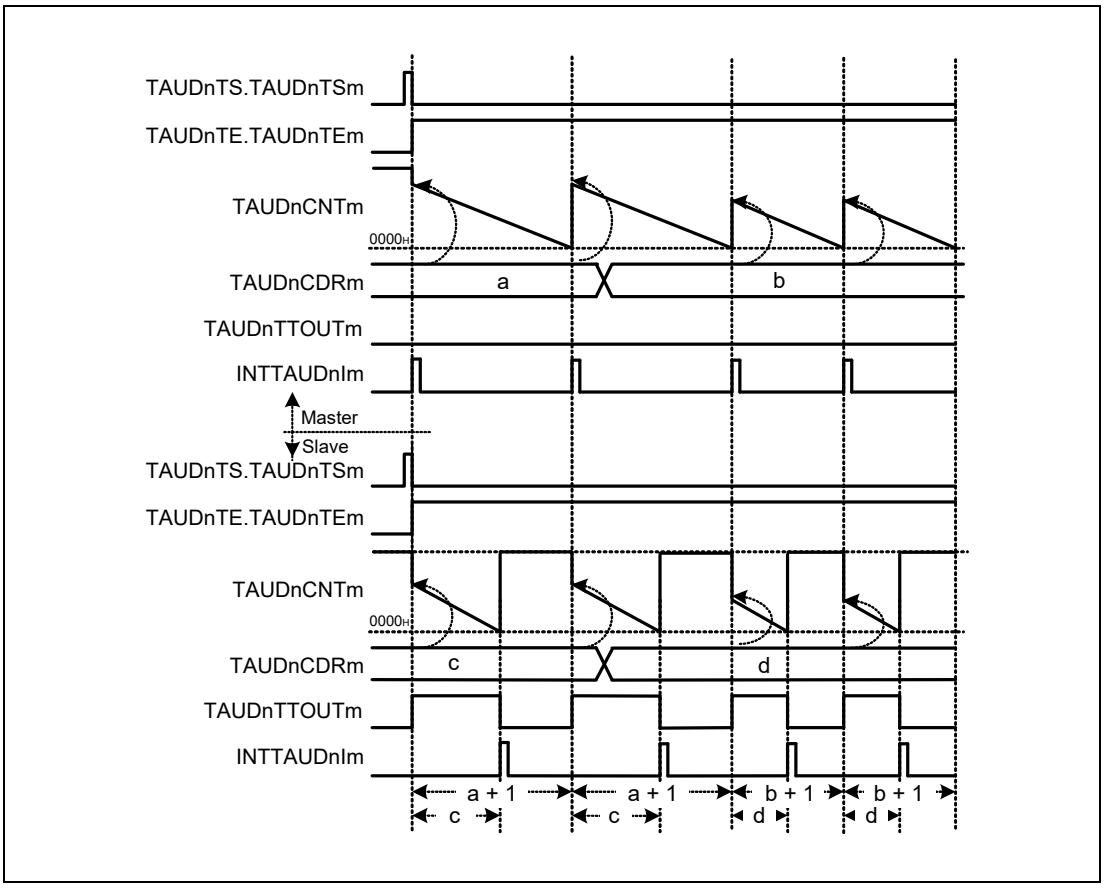


Figure 33.90 General Timing Diagram of PWM Output Function

NOTES

1. The interval between the starting to count and an interrupt being generated is the value of corresponding TAUDnCDRm + 1.
2. TAUDnTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

33.15.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.138 Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.139 Contents of the TAUDnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

The channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.140 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires a channel higher than the master channel that operates with **Section 33.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  TAUDnCDRm settings for this channel are as follows:
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × interrupt count) - 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

If TAUDnCDRm (slave) setting > TAUDnCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

33.15.1.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.141 Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.142 Contents of the TAUDnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 33.143 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.144 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.1.6 Operating Procedure for PWM Output Function

Table 33.145 Operating Procedure for PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.1.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.1.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave) is set.
During operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. • TAUDnTTOUTm (slave) is set to the active level. If TAUDnCNTm (slave) reaches 0000_H: • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

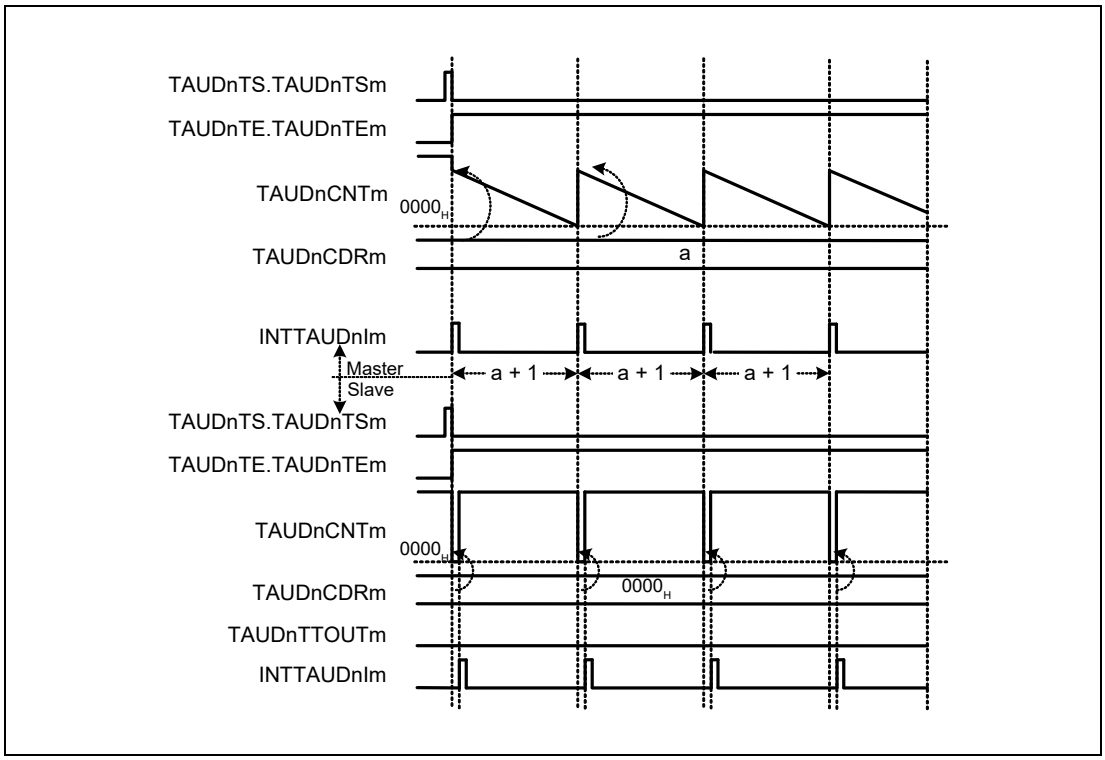


Figure 33.91 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is loaded into TAUDnCNTm (slave). As a result, a slave channel interrupt (INTTAUDnIm) is generated at the same time and TAUDnTTOUTm remains inactive.
- TAUDnCDRm value is loaded into TAUDnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

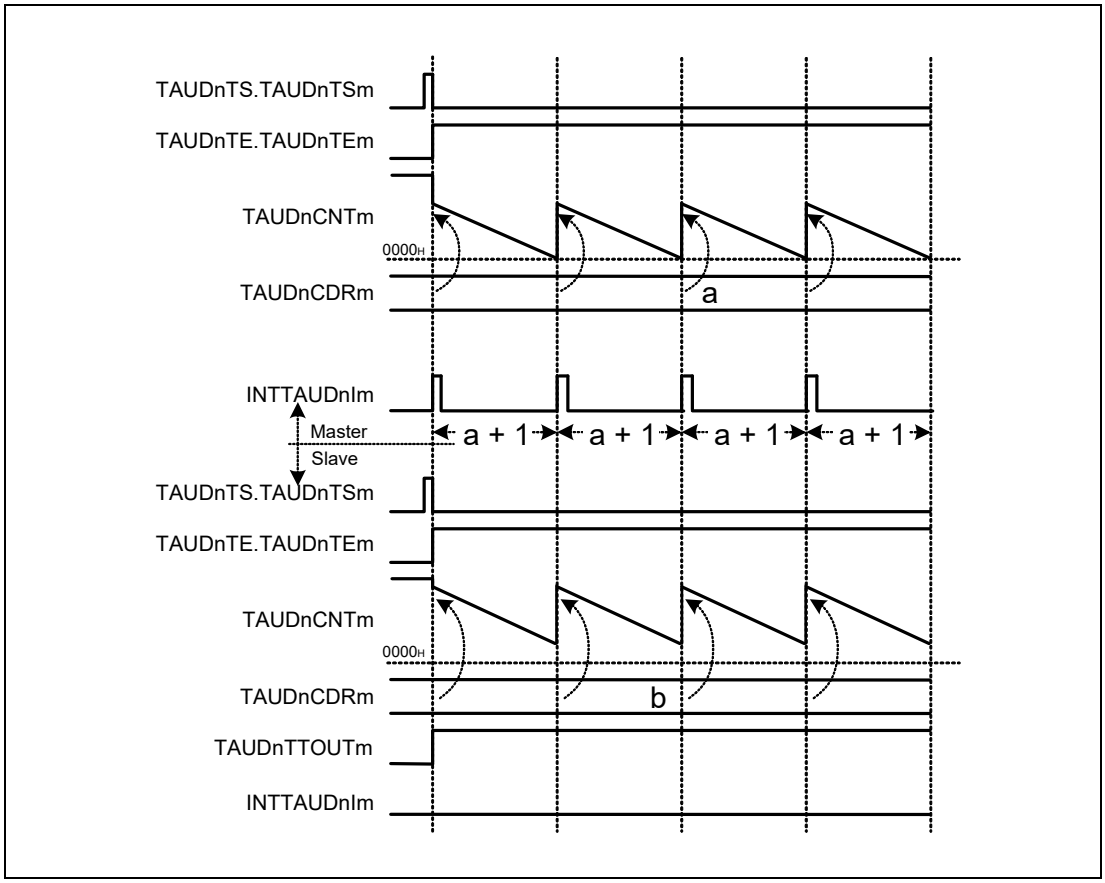


Figure 33.92 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the slave channel counter does not reach 0000_H and consequently, no interrupt occurs. TAUDnTTOUTm remains active.

(3) Operation stop and restart

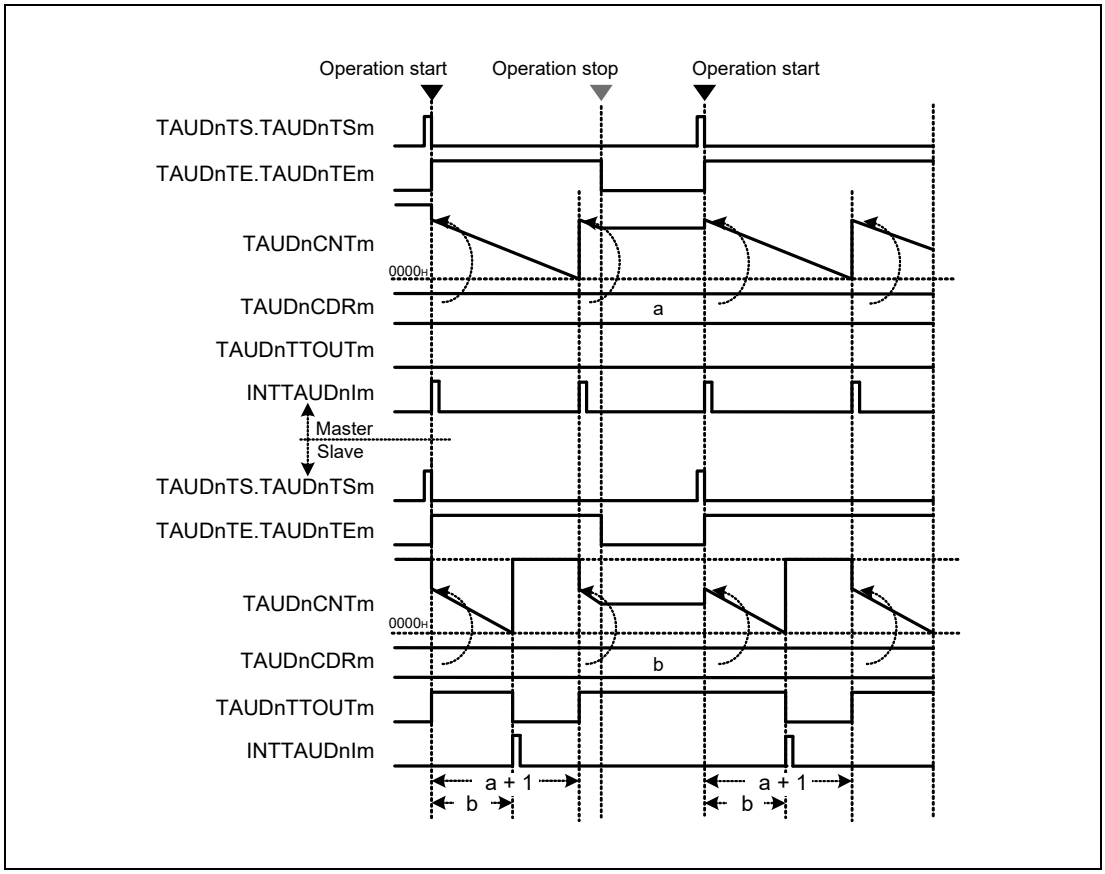


Figure 33.93 Operation Stop and Restart
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDnTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm of master and slave channels to 1. TAUDnCDRm values of the master and slave channels are loaded to TAUDnCNTm and start to count down from these values.

33.15.2 One-Shot Pulse Output Function

33.15.2.1 Overview

Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse or software trigger) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See **Table 33.146, Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**.)
- The operating mode for slave channels should be set to pulse one-count mode. (See **Table 33.149, Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**.)
- TAUDnTTOUTm is not used with the master channel of this function.
- The channel output mode for slave channel should be set to independent channel output mode 2. (See **Section 33.7, Channel Output Modes**.)
- TAUDnTTINm (master) has to be detected while TAUDnCNTm (master) and TAUDnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDnTTINm (slave).
- When using only the software trigger, do not select TAUDnTTINm with the alternative function of the pins.

Functional description

The counters are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1 for the master and slave channels. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  When the next valid TAUDnTTINm input edge or software trigger (when TAUDnTE.TAUDnTEm = 1, TAUDnTS.TAUDnTSm = 1 (m: master channel number)) is detected the current value of TAUDnCDRm is loaded into TAUDnCNTm. The counter starts to count down from this value. If TAUDnCMORm.TAUDnMD0 = 0, a trigger (TAUDnTTINm) which is detected within the delay time is ignored.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next valid TAUDnTTINm input edge or software trigger (when TAUDnTE.TAUDnTEm = 1, TAUDnTS.TAUDnTSm = 1 (m: master channel number)).
- Slave channel:
  INTTAUDnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDnTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUDnIm is generated and TAUDnTTOUTm signal is reset. The counter stops at 0000_H and waits for the next INTTAUDnIm of master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Setting TAUDnTS.TAUDnTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

Conditions

- If TAUDnCMORm.TAUDnMD0 of master channel is set to 0, TAUDnTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See **Section 33.6, Simultaneous Rewrite**.

33.15.2.2 Equations

Delay from trigger input to pulse output = (TAUDnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

33.15.2.3 Block Diagram and General Timing Diagram

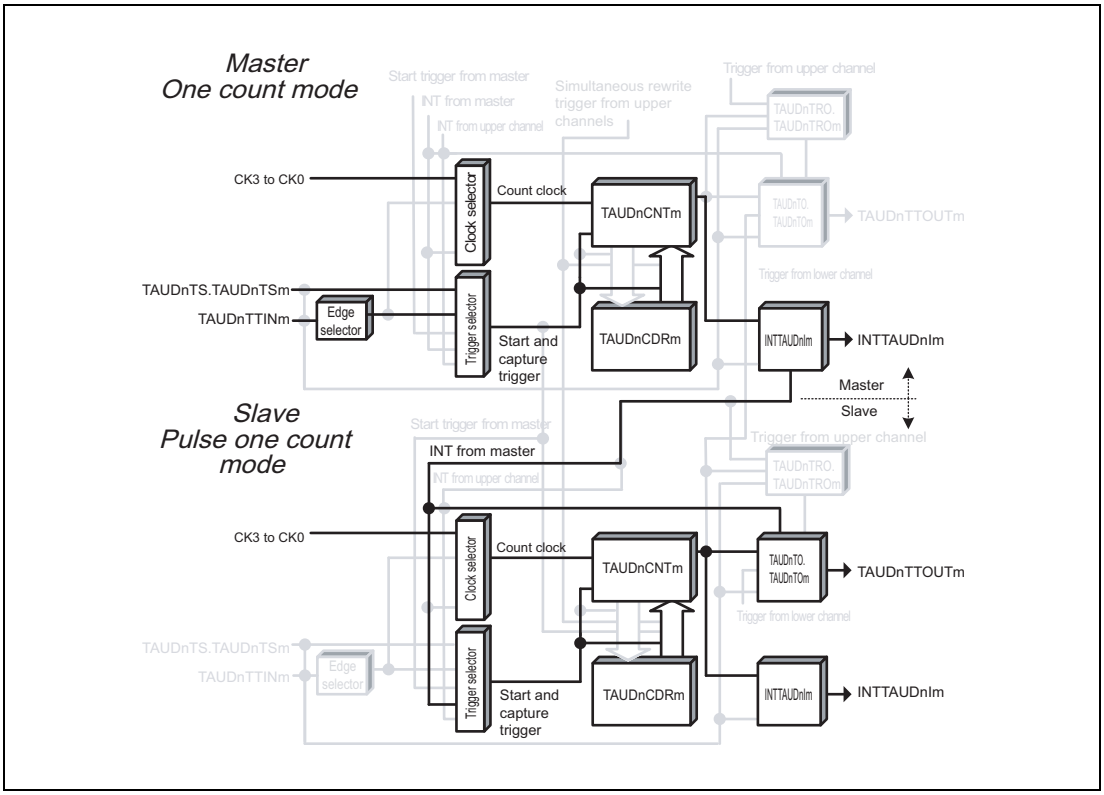


Figure 33.94 Block Diagram of One-Shot Pulse Output Function

The following settings apply to **Figure 33.95, General Timing Diagram of One-Shot Pulse Output Function (External Input Signal)** and **Figure 33.96, General Timing Diagram of One-Shot Pulse Output Function (Software Trigger)**.

- Start trigger detection is disabled during counting (TAUDnCMORm.TAUDnMD0 = 0).
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

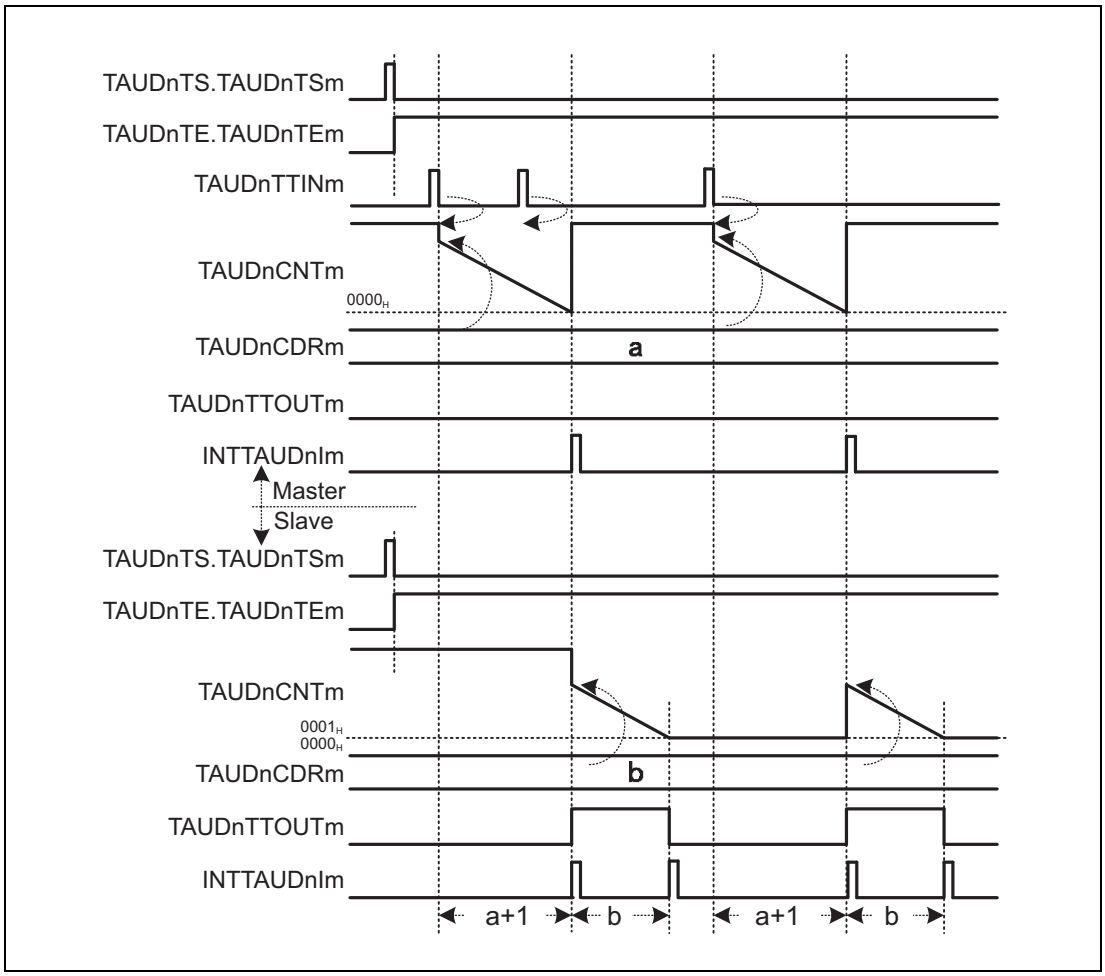


Figure 33.95 General Timing Diagram of One-Shot Pulse Output Function (External Input Signal)

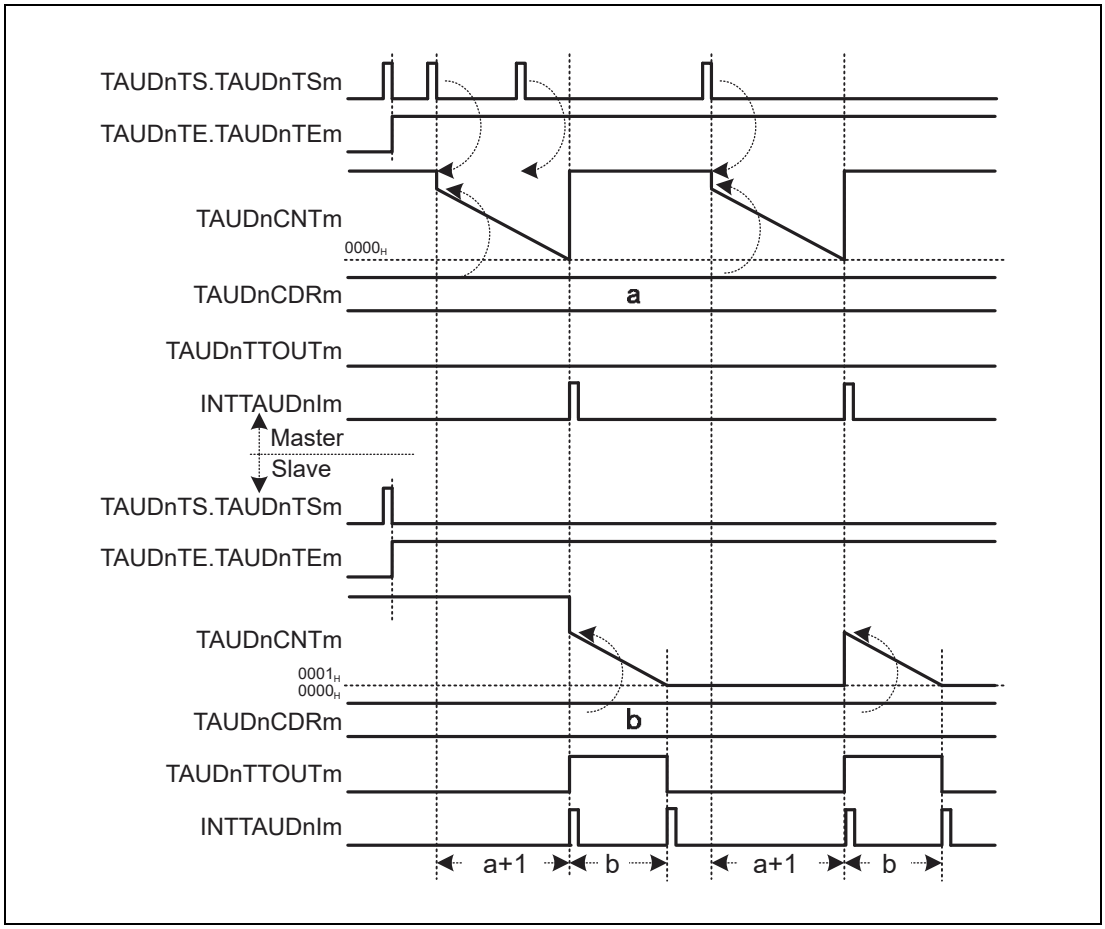


Figure 33.96 General Timing Diagram of One-Shot Pulse Output Function (Software Trigger)

33.15.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.146 Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. The value of the MD0 bits of master and slave channels must be identical.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.147 Contents of the TAUDnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.148 Simultaneous Rewrite Settings for the Master Channel of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.2.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.149 Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables start trigger detection while counting. The MD0 bits of the master and slave channels must be identical.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.150 Contents of the TAUDnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel

Table 33.151 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.152 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 33.153 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.2.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 33.15.2.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUDnTTINm input.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	When valid TAUDnTTINm input edge is detected, TAUDnCDRm value of master channel is loaded into TAUDnCNTm to perform counting down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) is reset to $FFFF_H$ and waits for the next valid TAUDnTTINm input edge. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is set to the active level. When TAUDnCNTm (slave) reaches 0001_H: • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.2.7 Specific Timing Diagrams

(1) TAUDnCDRm (master) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

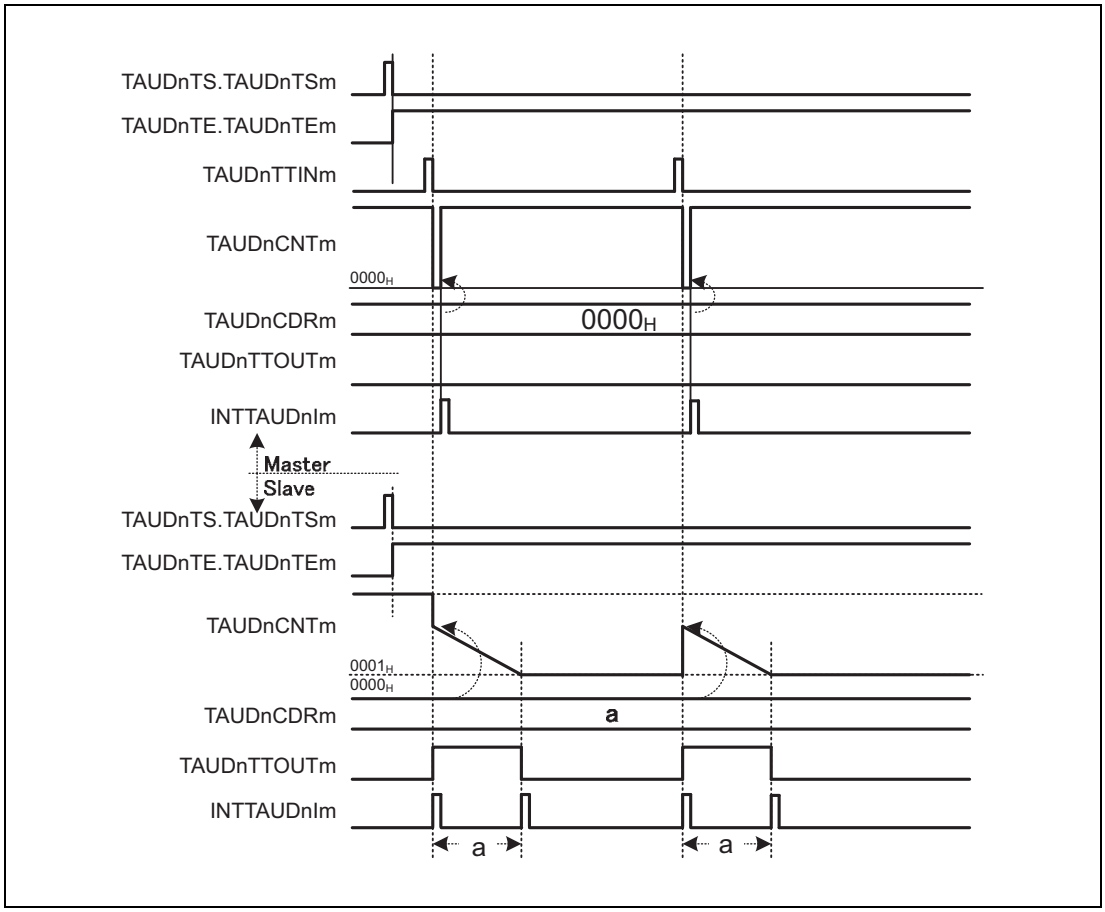


Figure 33.97 TAUDnCDRm (Master) = 0000_H

- When a valid TAUDnTTINm input edge is detected, the value 0000_H is written to TAUDnCNTm (master). The counter is set to 0000_H for one count and returns to $FFFF_H$.
  Thus the slave channel starts to count down one count clock later than TAUDnTTINm (master).

(2) TAUDnCDRm (slave) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

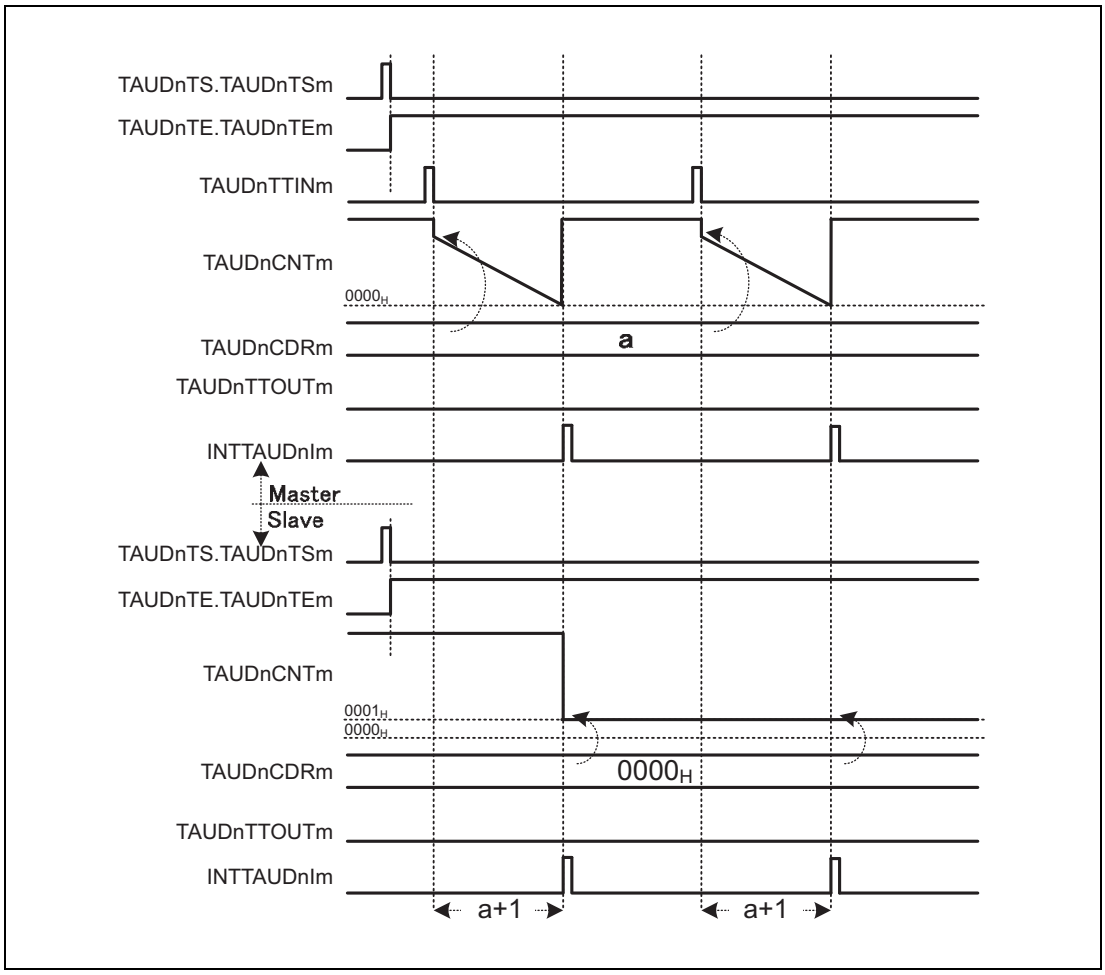


Figure 33.98 TAUDnCDRm (Slave) = 0000_H

- TAUDnTTOUTm remains inactive, because the pulse width is zero.

(3) TAUDnCMORm.TAUDnMD0 = 1

The following settings apply to this diagram.

- Enables start trigger detection while counting. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

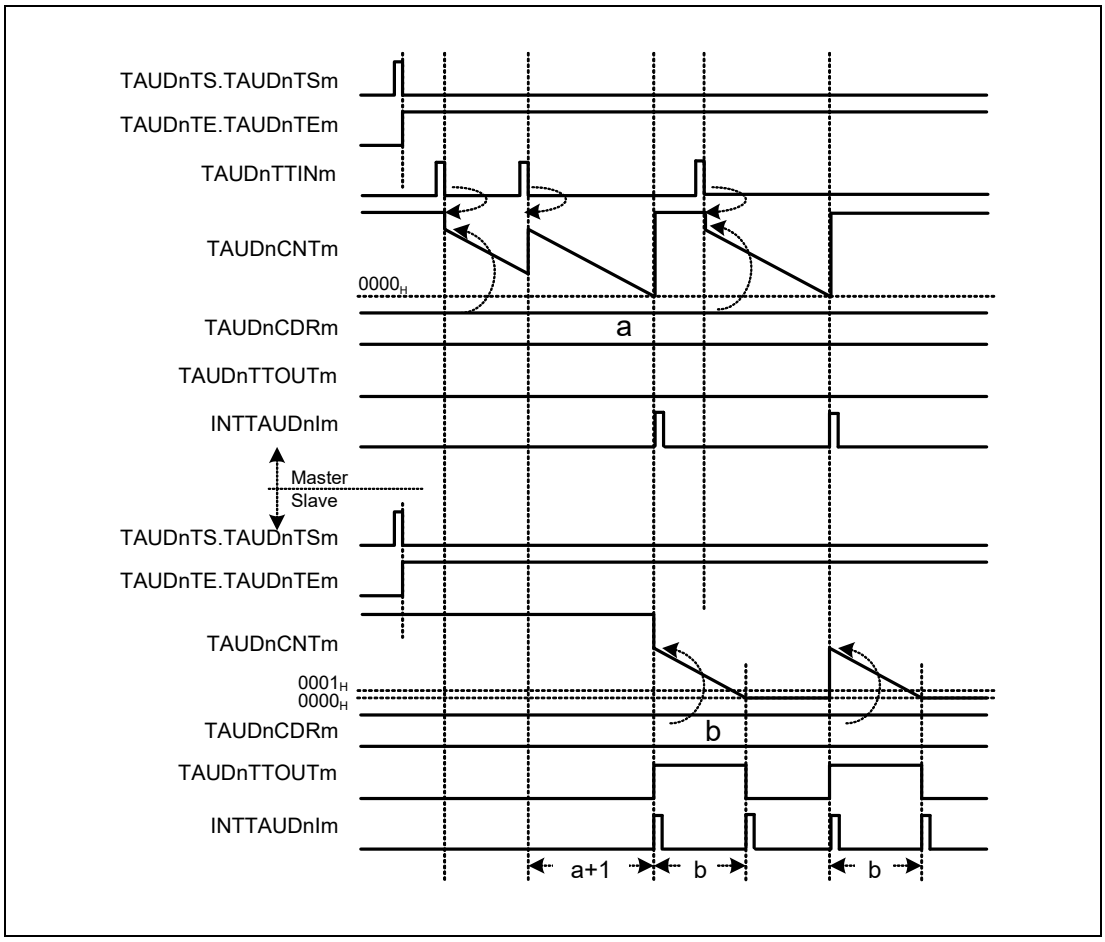


Figure 33.99 TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDnTTINm input edge is detected while the counter of the master channel counts down, TAUDnCNTm reloads the value of TAUDnCDRm. The counter restarts to count down. This means the delay is extended by the value of TAUDnCNTm at the time when a valid TAUDnTTINm input edge is detected.

(4) Restarting the Master Channel while the Slave Channel is Counting

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

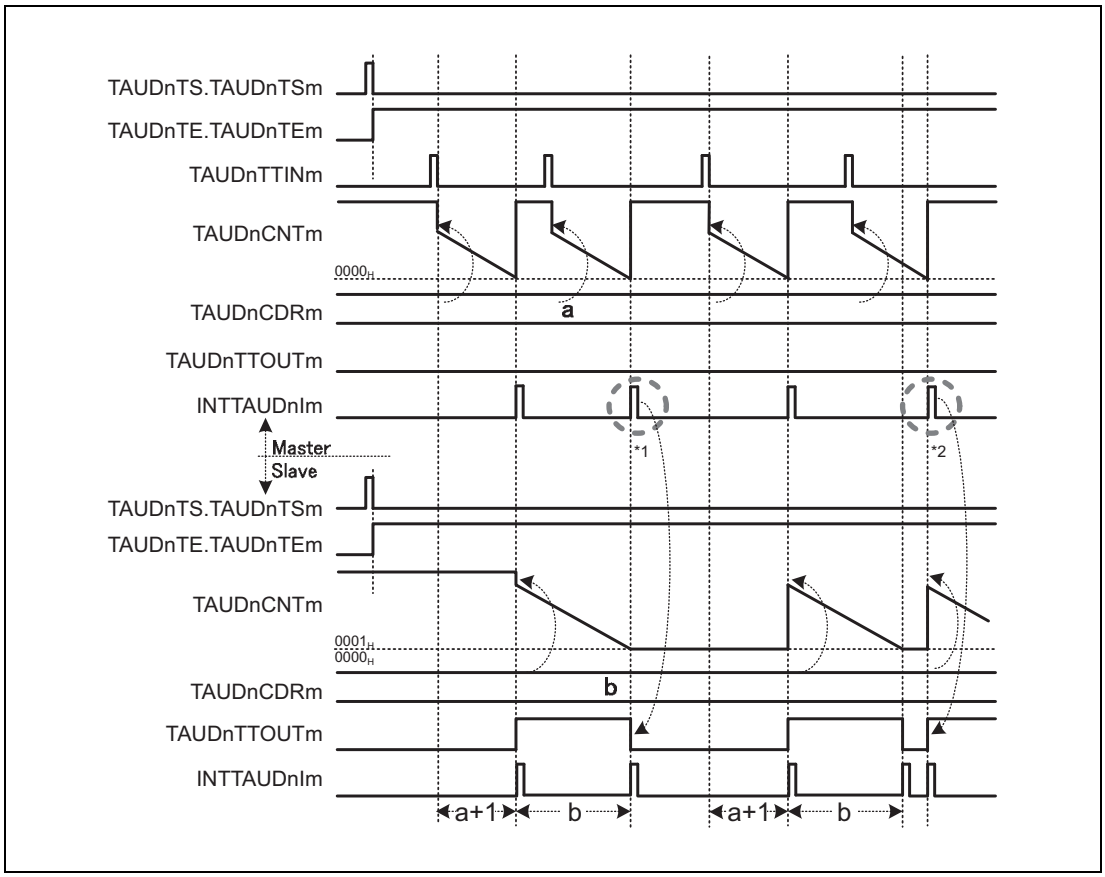


Figure 33.100 TAUDnTTINm input interval ≤ Delay Time + Pulse Width + 1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached[*1], the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUDnCDRm (slave) is reloaded. An interrupt is generated and TAUDnTTOUTm toggles. If TAUDnCNTm (master) has started to count down while the TAUDnCNTm (slave) is still counting[*2], TAUDnTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

33.15.3 Trigger Start PWM Output Function

33.15.3.1 Overview

Summary

This function generates a PWM output using a master and a slave channel. It enables the pulse cycle (frequency) and the duty of the TAUDnTTOUTm to be set. The pulse cycle is specified using the master channel. The duty is specified using the slave channel. The Trigger Start PWM Output Function is identical to PWM Output Function except that the master channel of this function can be reset by a valid TAUDnTTINm input edge.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode (see **Table 33.154, Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function**).
- The operation mode of the slave channel must be set to One-Count Mode (see **Table 33.157, Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function**).
- TAUDnTTOUTm is not used with the master channel of this function.
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1 (see **Section 33.6, Simultaneous Rewrite**).

Functional description

The counters (master and slave) are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of TAUDnCDRm is loaded to TAUDnCNTm, and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel, and a PWM output is realized by setting and resetting TAUDnTTOUTm (slave).

- Master channel:
  The current value of TAUDnCDRm is loaded to the counter (TAUDnCNTm), INTTAUDnIm is generated and the counter starts to count down from this value.
  When the counter reaches 0000_H and the pulse cycle time has elapsed, INTTAUDnIm is generated and the counters (master and slave) load the current TAUDnCDRm values.
  If a valid TAUDnTTINm input edge is detected, the counter of the master channel loads the current TAUDnCDRm value, restarts counting down and generates an interrupt.
- Slave channel:
  When the slave detects an interrupt from the master channel, it starts to count down from the current value of TAUDnCDRm. The TAUDnTTOUTm signal is set to the active level.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and the TAUDnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 33.6, Simultaneous Rewrite**.

33.15.3.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave) / (TAUDnCDRm (master) + 1)] × 100

– Duty cycle = 0%
TAUDnCDRm (slave) = 0000_H

– Duty cycle = 100%
TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

33.15.3.3 Block Diagram and General Timing Diagram

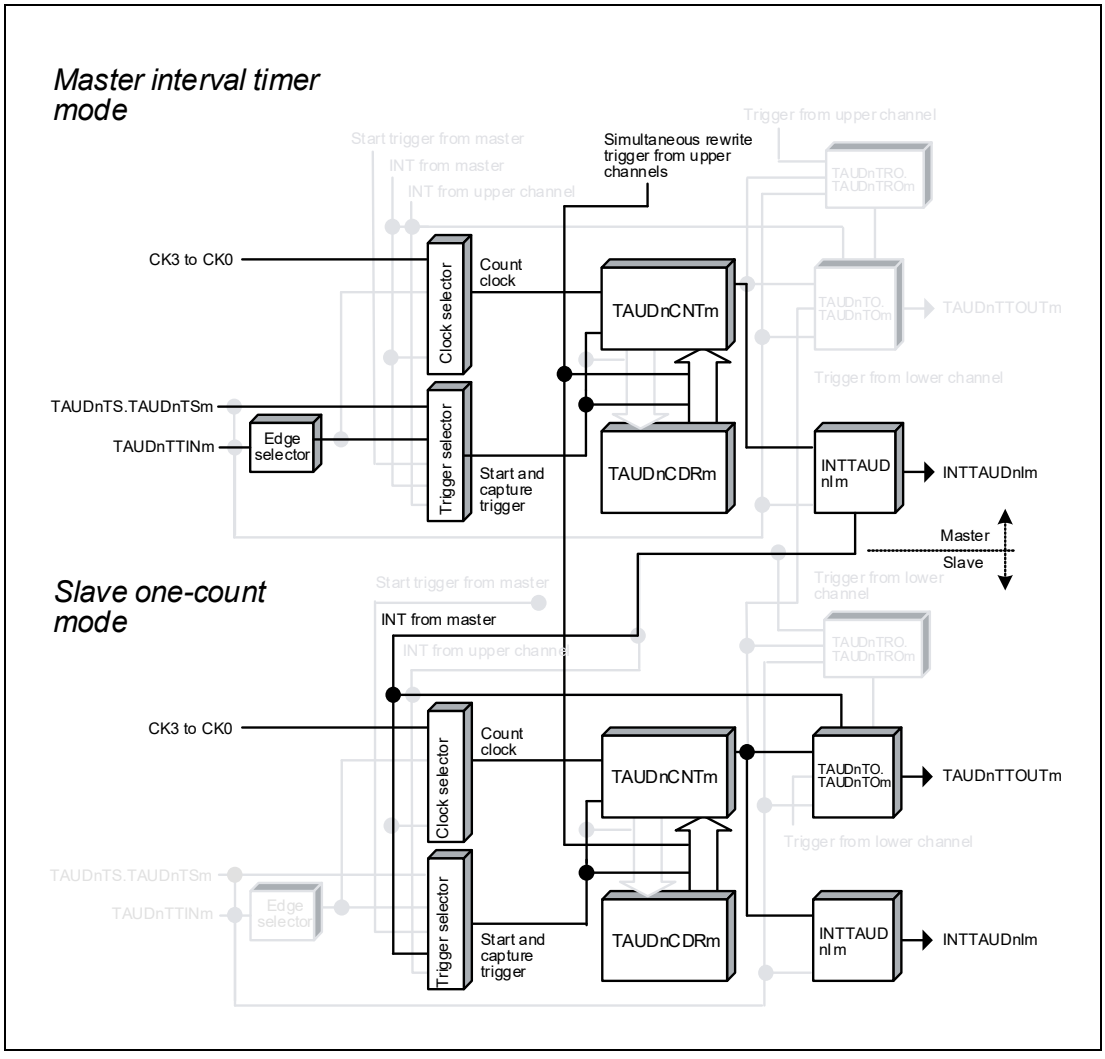


Figure 33.101 Block Diagram for Trigger Start PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)
- Positive logic (TAUDnTOL.TAUDnTOLm (slave) = 0)

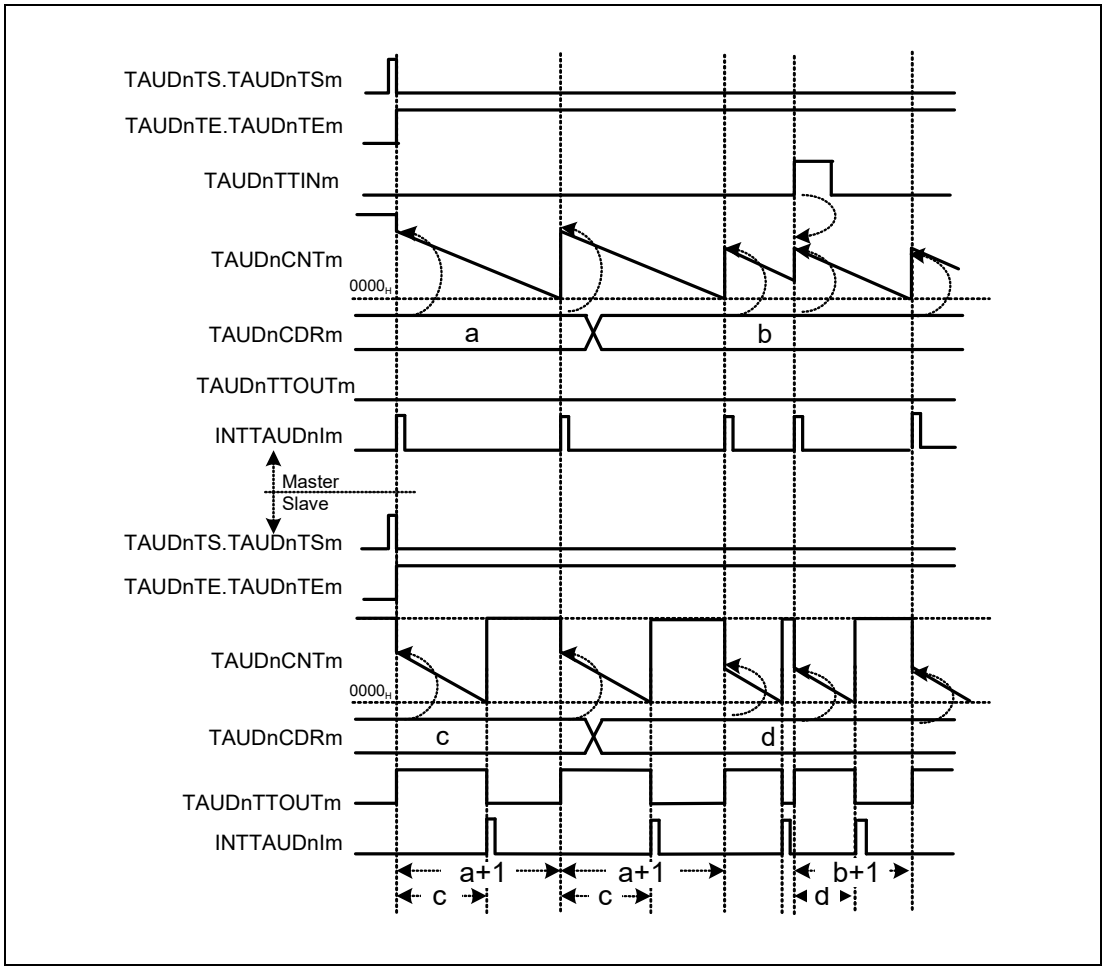


Figure 33.102 General Timing Diagram for Trigger Start PWM Output Function

NOTE

TAUDnTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

33.15.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.154 Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.155 Contents of the TAUDnCMURm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

The channel output mode is not used by this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.156 Simultaneous Rewrite Settings for the Master Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.3.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.157 Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid. The value of the TAUDnMD[0] bit of the master and slave channels must be identical.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.158 Contents of the TAUDnCMURm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel

Table 33.159 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.160 Simultaneous Rewrite Settings for Slave Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.3.6 Operating Procedure for Trigger Start PWM Output Function

Table 33.161 Operating Procedure for Trigger Start PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **33.15.3.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **33.15.3.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. • TAUDnCNTm (slave) loads the TAUDnCDRm value and starts to count down • TAUDnTTOUTm (slave) is set When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops. If a TAUDnTTINm input is detected on the master channel while TAUDnCNTm of the master channel is counting down: • TAUDnCNTm (master and slave) loads the TAUDnCDRm value and counts down • INTTAUDnIm (master) is generated. • TAUDnTTOUTm (slave) is set to the active level.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.3.7 Specific Timing Diagrams

(1) Duty cycle = 0%

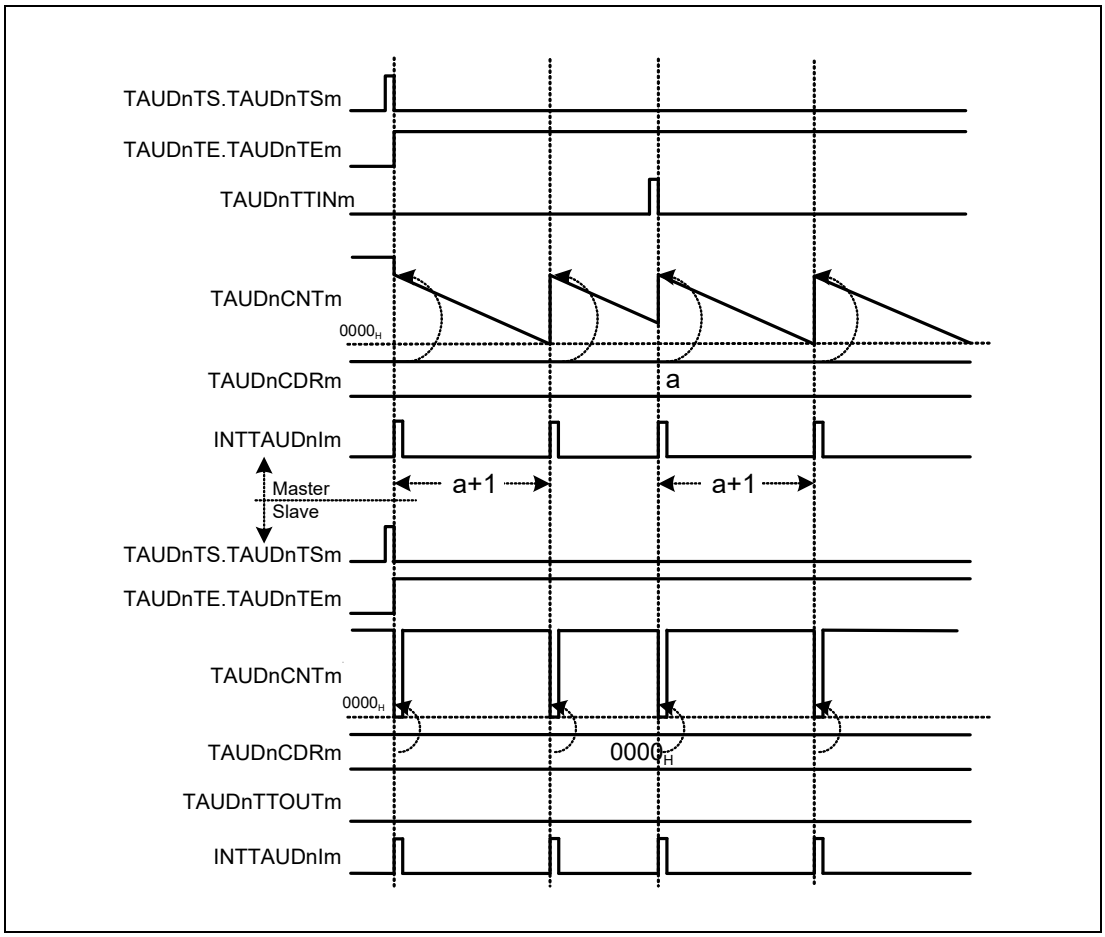


Figure 33.103 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is written to TAUDnCNTm (slave). Therefore, TAUDnCNTm (slave) cannot start to count and TAUDnTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The detection of a valid TAUDnTTINm input edge has no effect on TAUDnTTOUTm (slave).

(2) Duty cycle = 100%

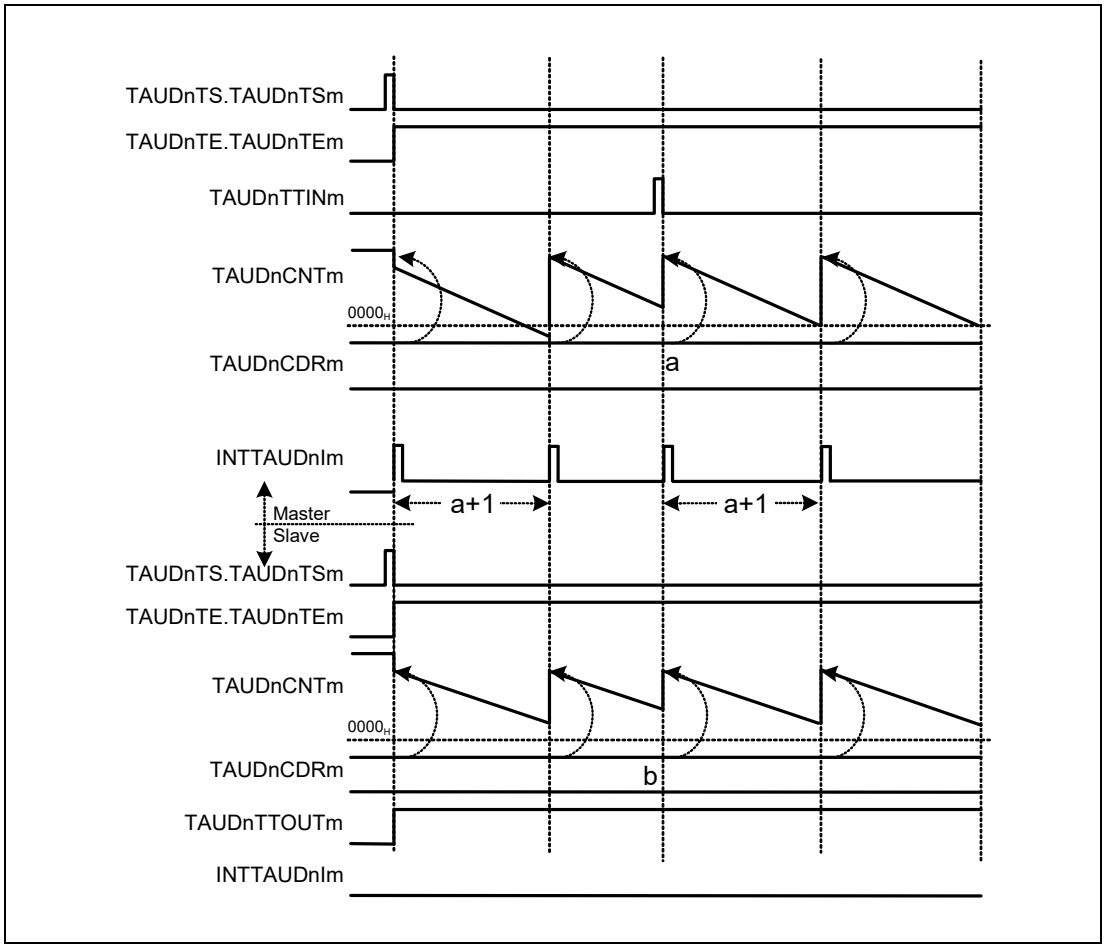


Figure 33.104 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Falling Edge Detection (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- If the value TAUDnCDRm (slave) is higher than the value TAUDnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts.
  The TAUDnTTOUTm remains at active state.
  The detection of a valid TAUDnTTINm input edge has no effect on TAUDnTTOUTm (slave).

(3) TAUDnTTINm Detection and Active Slave Counter

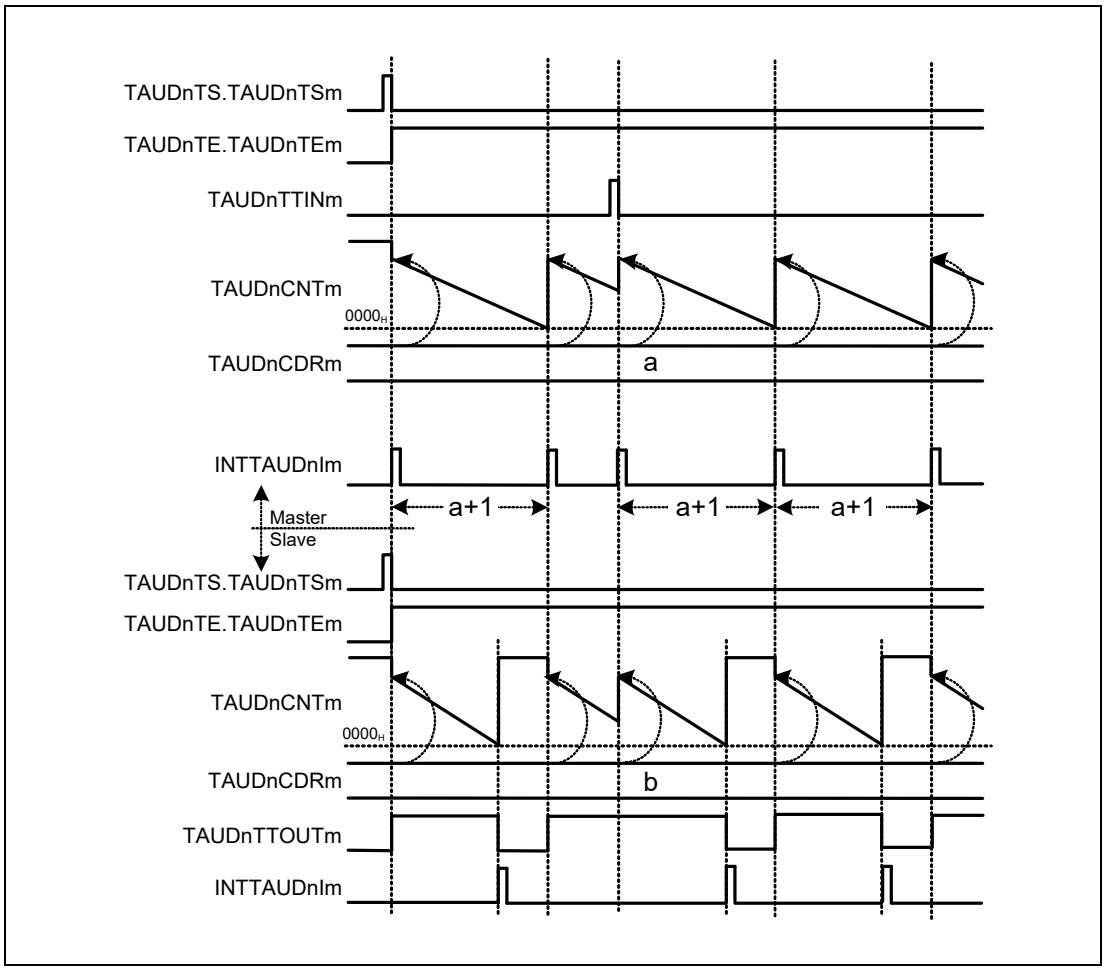


Figure 33.105 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- If TAUDnCNTm (slave) reloads the value TAUDnCDRm (slave) while it is still counting down, TAUDnTTOUTm remains unchanged and extends the duty.
  The duty does not correspond to the value of the slave's data register.

33.15.4 Delay Pulse Output Function

33.15.4.1 Overview

Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified on slave channel 2.

Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 33.162, Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function**.)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See **Table 33.165, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function** and **Table 33.169, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function**.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 33.172, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function**.)
- TAUDnTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See **Section 33.7, Channel Output Modes**.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 33.7, Channel Output Modes**.)

Functional description

The counters of the channel group are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel.
  When the counter value of master channel reaches 0000_H and pulse cycle time has elapsed, INTTAUDnIm is generated. The TAUDnCDRm value is reloaded into the counter to perform counting down.

- Slave channels 1 and 2:
  Slave channels 1 and 2 start to count down from the current TAUDnCDRm value when detecting an interrupt from the master channel. TAUDnTTOUTm signal (slave 1) is set.
  - Slave channel 1:
    When the counter of slave channel 1 reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDnTTOUTm signal is reset. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.
  - Slave channel 2:
    When the counter of slave channel 2 reaches 0000_H and delay time has elapsed, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.
    Generating INTTAUDnIm (slave channel 2) triggers the counter of slave channel 3.
- Slave channel 3:
  When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUDnCDRm. INTTAUDnIm is generated and the TAUDnTTOUTm signal (slave channel 3) is set.
  When the counter of slave channel 3 reaches 0001_H, INTTAUDnIm is generated and the TAUDnTTOUTm signal is reset.
  The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 33.6, Simultaneous Rewrite**.

33.15.4.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDnCDRm (slave 1)) × count clock cycle

Delay width = (TAUDnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:

$0000_H \leq$ TAUDnCDRm (slave 2) < TAUDnCDRm (master)

NOTES

1. The waveform of TAUDnTTOUTm (slave 3) becomes the waveform made by delaying the waveform of TAUDnTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.
2. If INTTAUDnIm of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the waveform of TAUDnTTOUTm (slave 3) is retained on the active level. In this case, TAUDnTTOUTm (slave 3) cannot output the waveform generated by delaying the basic pulse of TAUDnTTOUTm (slave 1).

33.15.4.3 Block Diagram and General Timing Diagram

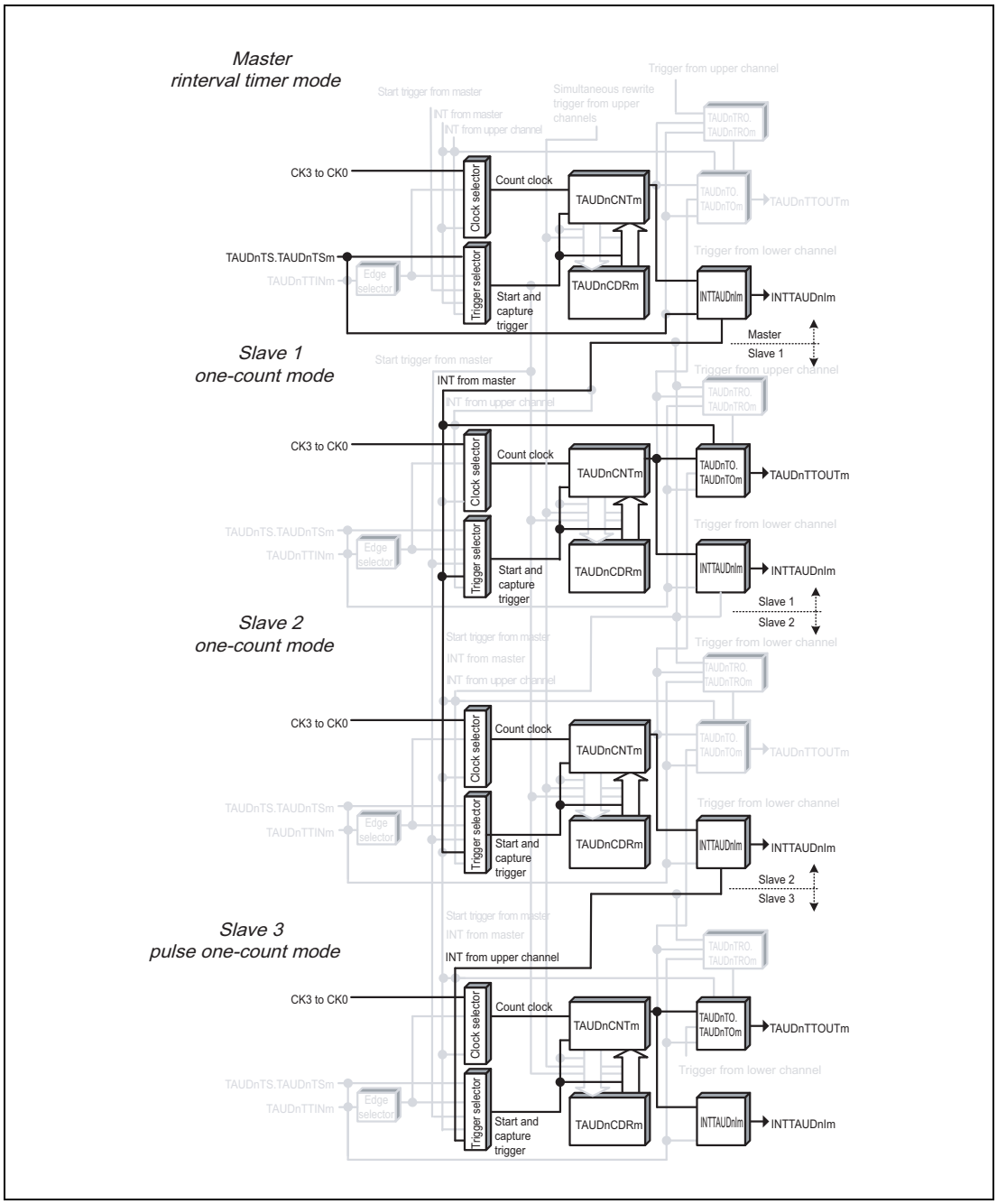


Figure 33.106 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDnTOL.TAUDnTOLm = 0)
- Slave channel 3: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

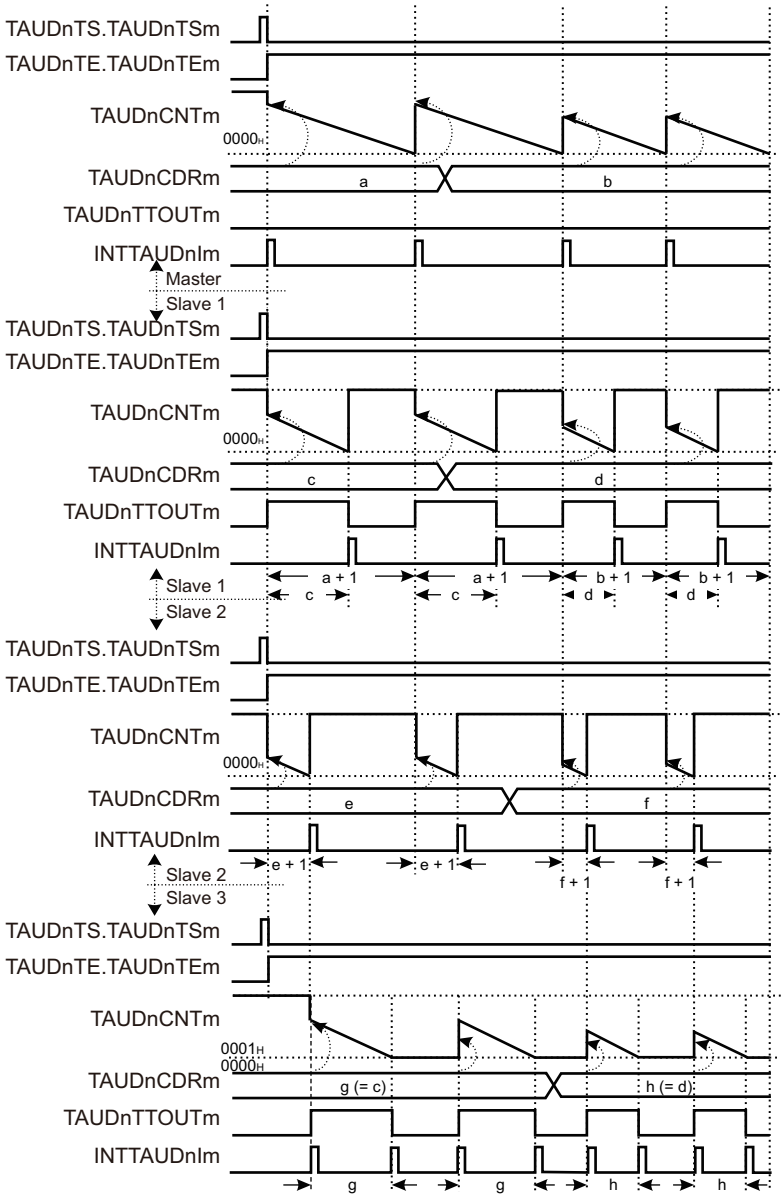


Figure 33.107 General Timing Diagram of Delay Pulse Output Function

NOTE

TAUDnTTOUTm of slave channel 1 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

33.15.4.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.162 Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.163 Contents of the TAUDnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used for the master channel with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.164 Simultaneous Rewrite Settings for the Master Channel of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.4.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.165 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.166 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 1

Table 33.167 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.168 Simultaneous Rewrite Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.4.6 Register Settings for Slave Channel 2

(1) TAUDnCMORm for Slave Channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.169 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.170 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 2

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for Slave Channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.171 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.4.7 Register Settings for Slave Channel 3

(1) TAUDnCMORm for Slave Channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.172 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	101: INTTAUDnIm of upper channel (m - 1) is a start trigger regardless of master setting.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for Slave Channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.173 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 3

Table 33.174 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.175 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.4.8 Operating Procedure for Delay Pulse Output Function

Table 33.176 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.4.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.4.5, Register Settings for Slave Channel 1**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.4.6, Register Settings for Slave Channel 2**. Slave channel 3: Set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.4.7, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.

Table 33.176 Operating Procedure for Delay Pulse Output Function (2/2)

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave channel 1) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 1 and 2 is loaded to TAUDnCNTm and count down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1/2) to count down. • TAUDnTTOUTm (slave 1) is set. When TAUDnCNTm (slave 1) reaches 0000_H: • INTTAUDnIm (slave 1) is generated. • TAUDnTTOUTm (slave 1) is reset. When TAUDnCNTm (slave 2) reaches 0000_H: • INTTAUDnIm (slave 2) is generated. • INTTAUDnIm (slave 3) is generated. • TAUDnTTOUTm (slave 3) is set. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 3) to count down operation. When TAUDnCNTm (slave 3) reaches 0001_H: • INTTAUDnIm (slave 3) is generated. • TAUDnTTOUTm (slave 3) is reset.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.4.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 33.108**:

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = $000B_H$
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = $000B_H$

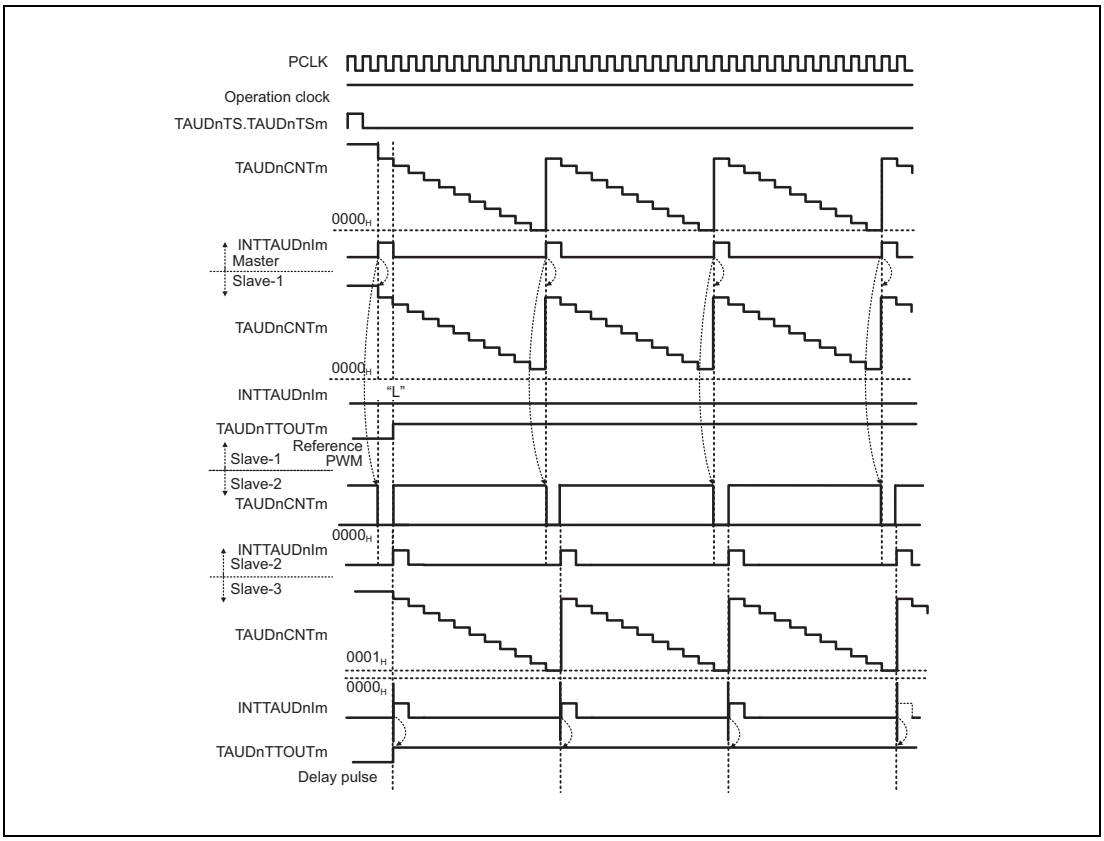


Figure 33.108 Duty Cycle (Slave 3) = 100%

- If the value of TAUDnCDRm (slaves 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of the slave channel 1 cannot reach 0000_H and cannot generate interrupts. TAUDnTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUDnTTOUTm (slave 1) = TAUDnTTOUTm (slave 3)

The following values apply to **Figure 33.109**.

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = 0005_H
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = 0005_H

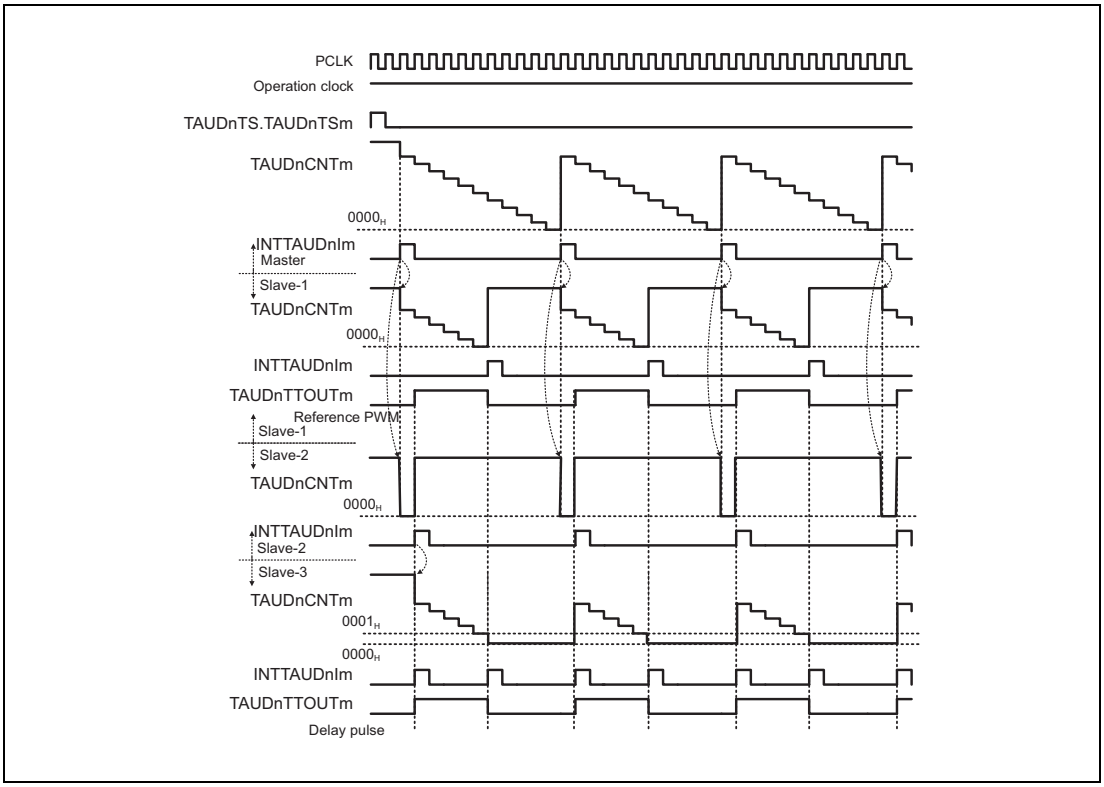


Figure 33.109 TAUDnTTOUTm (Slave 1) = TAUDnTTOUTm (Slave 3)

- If TAUDnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

33.15.5 Offset Trigger Output Function

33.15.5.1 Overview

Summary

This function generates a PWM output using a master channel and a slave channel, enabling the pulse width (duration) of the TAUDnTTOUTm to be set. The pulse cycle is set by detecting a valid input edge of master channel. The pulse width is specified on the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See **Table 33.177, Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function**.)
- The operating mode for slave channels should be set to one-count mode. (See **Table 33.180, Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function**.)
- The output mode for slave channels should be set to synchronous channel output mode 1. (See **Section 33.7, Channel Output Modes**.)
- TAUDnTTOUTm is not used with the master channel of this function.

Functional description

The counter can be enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The master channel counter (TAUDnCNTm) starts to count up from 0000_H.

- Master channel:
  When a valid TAUDnTTINm input edge is detected, the current value of the counter (TAUDnCNTm) is loaded into the data register of master channel (TAUDnCDRm). INTTAUDnIm is generated and the counter restarts to count up from 0000_H.
- Slave channel:
  If INTTAUDnIm is generated on the master channel, the TAUDnTTOUTm (slave) signal is set and the counter of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts to count down from this value.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

33.15.5.2 Equations

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)] × 100

– Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

33.15.5.3 Block Diagram and General Timing Diagram

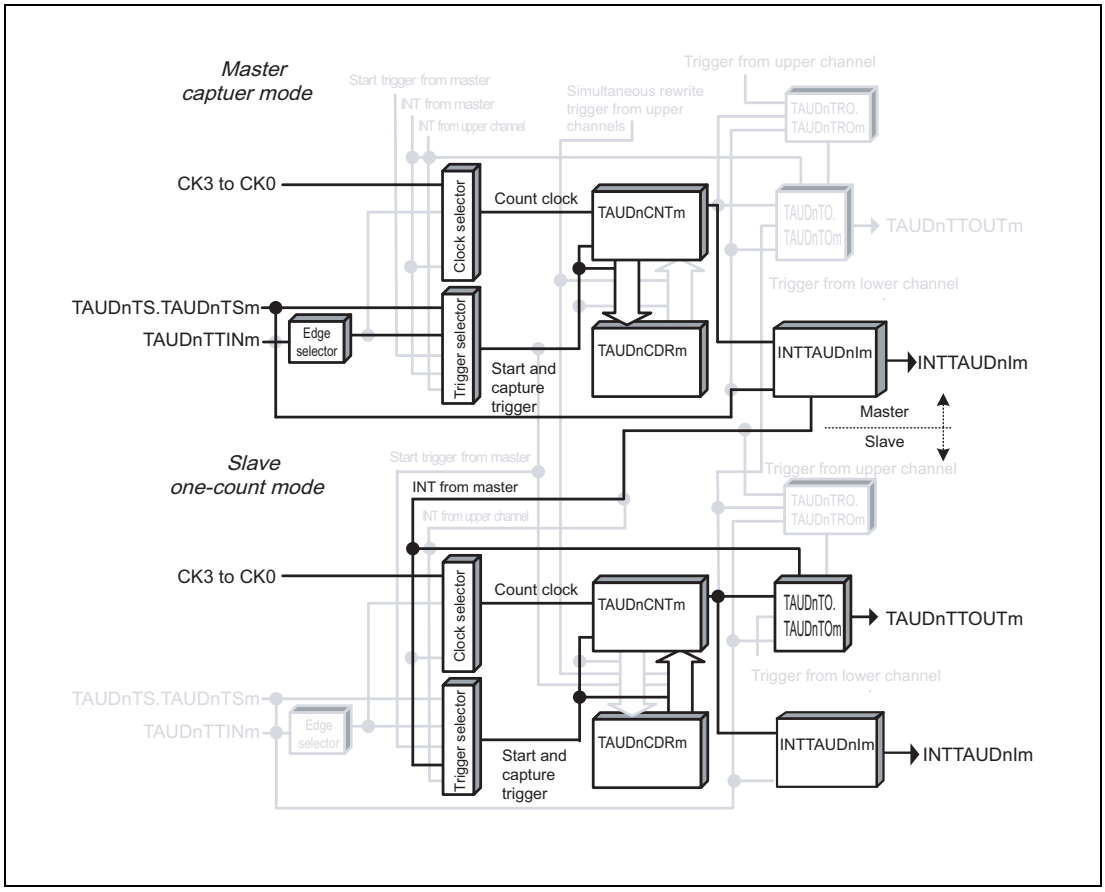


Figure 33.110 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

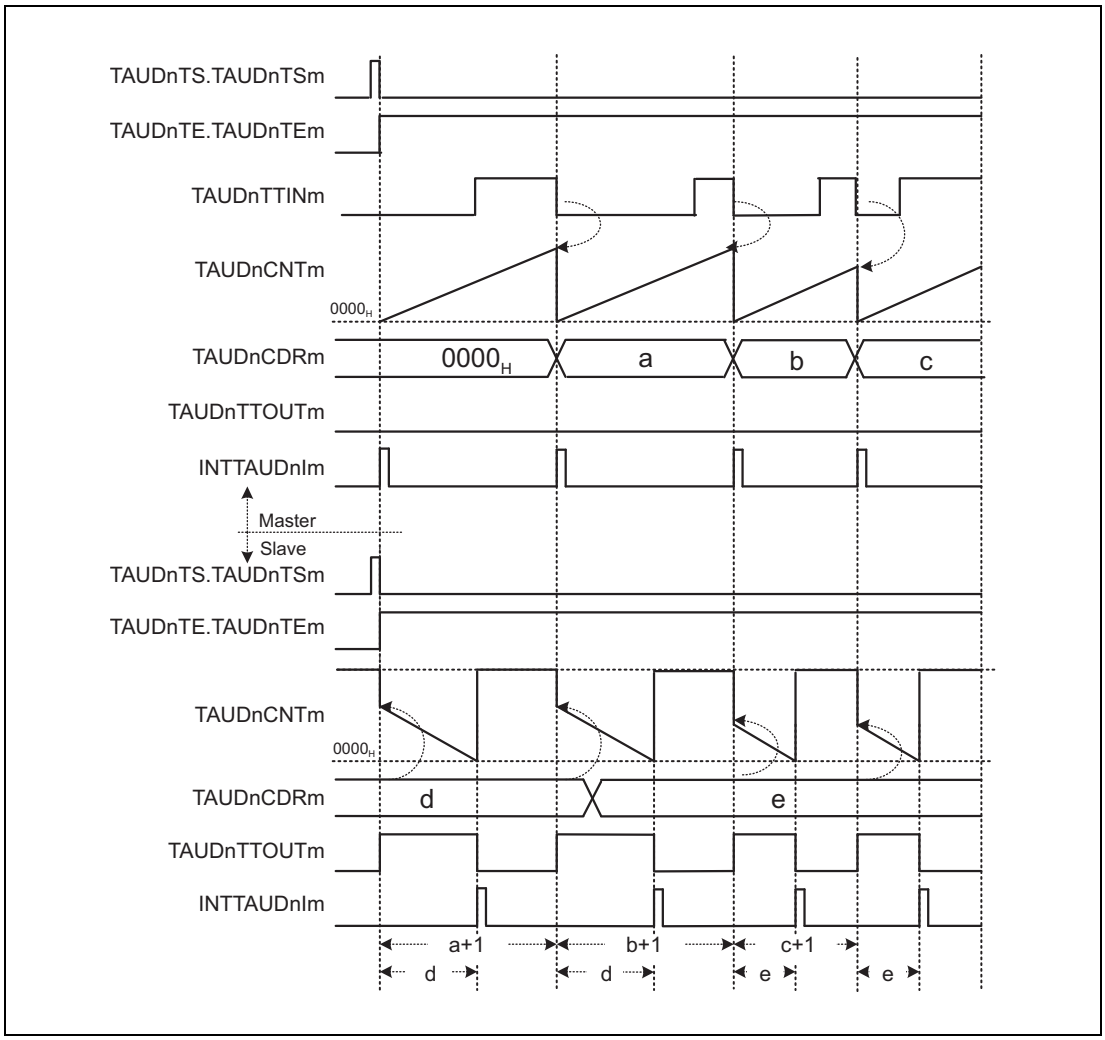


Figure 33.111 General Timing Diagram of Offset Trigger Output Function

NOTE

TAUDnTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

33.15.5.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.177 Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDnTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	11: Capture register is updated upon detection of a valid TAUDnTTINm input edge or when a counter overflow occurs: – Detection of valid TAUDnTTINm input edge: The counter value is written into TAUDnCDRm. – Occurrence of overflow: $FFFF_H$ is written into TAUDnCDRm. A valid TAUDnTTINm input edge to be detected next is ignored. TAUDnCSRm.TAUDnOVF is set when a counter overflow occurs, and cleared by setting TAUDnCSCm.TAUDnCLOV = 1.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.178 Contents of the TAUDnCMURm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used in this function.

(4) Simultaneous Rewrite for the Master Channel

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

Table 33.179 Simultaneous Rewrite Settings for the Master Channel of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.15.5.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.180 Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.181 Contents of the TAUDnCMURm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 33.182 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the offset trigger output function. Therefore, these registers should be set to 0.

Table 33.183 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0.
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

33.15.5.6 Operating Procedure for Offset Trigger Output Function

Table 33.184 Operating Procedure for Offset Trigger Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.5.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 33.15.5.5, Register Settings for Slave Channels**. The TAUDnCDRm register of master channel functions as a capture register. Set the value of TAUDnCDRm register of slave channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start: • TAUDnCNTm (master) counts up. • TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. INTTAUDnIm is generated on the master channel and TAUDnTTOUTm (slave) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCSCm.TAUDnCLOV can be set to 1. TAUDnCDRm of slave channel can be changed after the generation of INTTAUDnIm (master). TAUDnCNT.TAUDnCNTm and TAUDnCSRm can be read at any time.	When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is reset, and the counter of slave channel stops. When TAUDnTTINm input edge is detected on the master channel: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) is reset to 0000_H and then continues count operation subsequently. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. • TAUDnTTOUTm (slave) is set.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

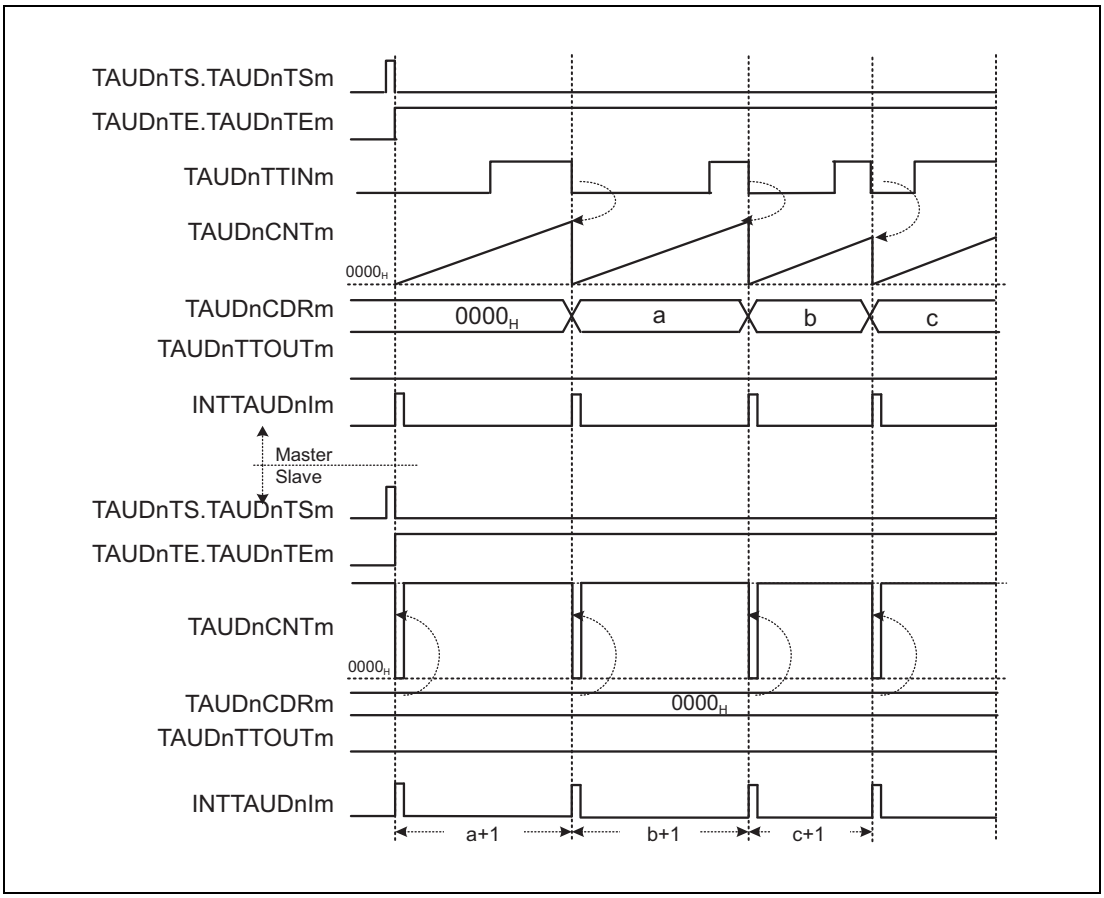


Figure 33.112 TAUDnCDRm (Slave) = 0000_H

- When TAUDnCDRm (slave) = 0000_H, 0000_H is written to TAUDnCNTm every time the master channel generates an interrupt (INTTAUDnIm), and TAUDnCNTm cannot start to count. The TAUDnTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The slave and the master channels generate interrupts in the same cycle.

(2) Duty cycle = 100%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

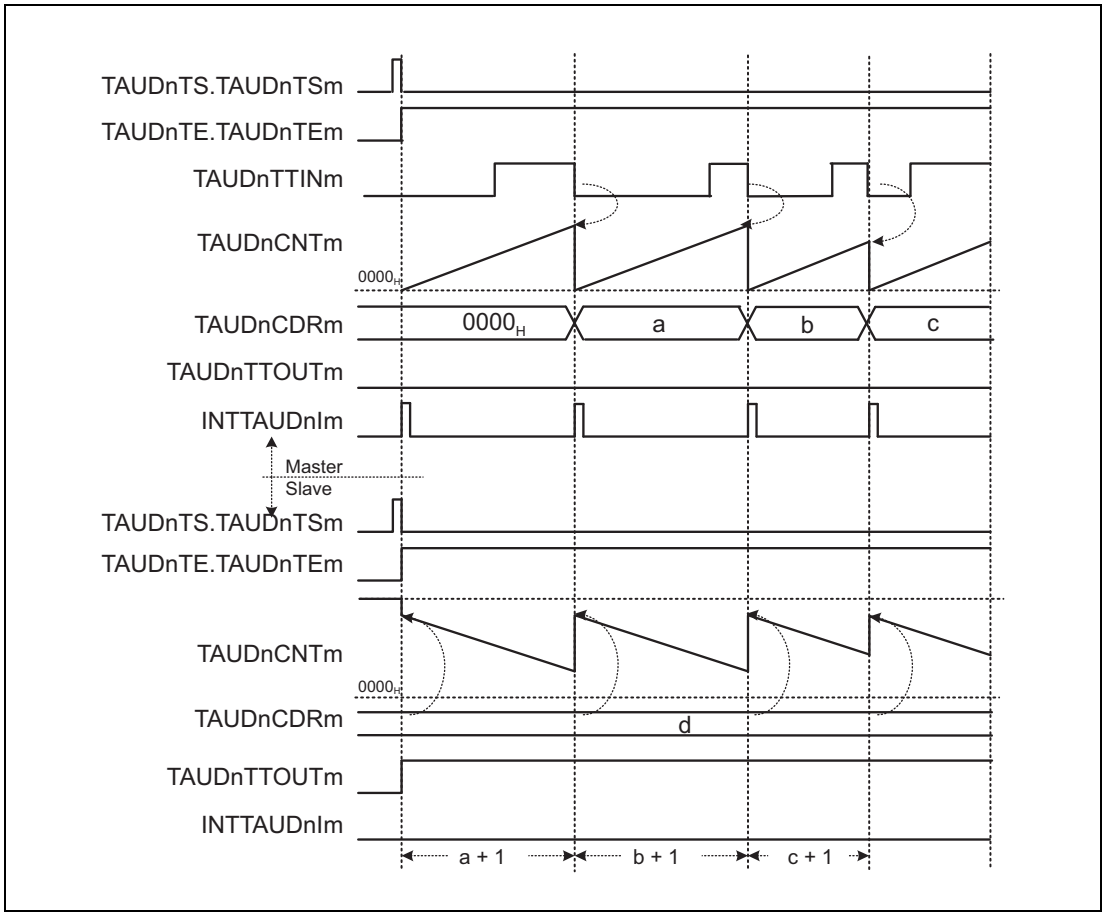


Figure 33.113 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If the value TAUDnCDRm (slave) is higher than the interval of valid input edges, the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUDnTTOUTm remains at active state.

33.15.6 A/D Conversion Trigger Output Function Type 1

33.15.6.1 Overview

Summary

This function is identical to **Section 33.15.1, PWM Output Function**, except that TAUDnTTOUTm is not output.

This function is enabled by setting the channel output mode for slave to independent channel output mode controlled by software.

33.15.6.2 Block Diagram and General Timing Diagram

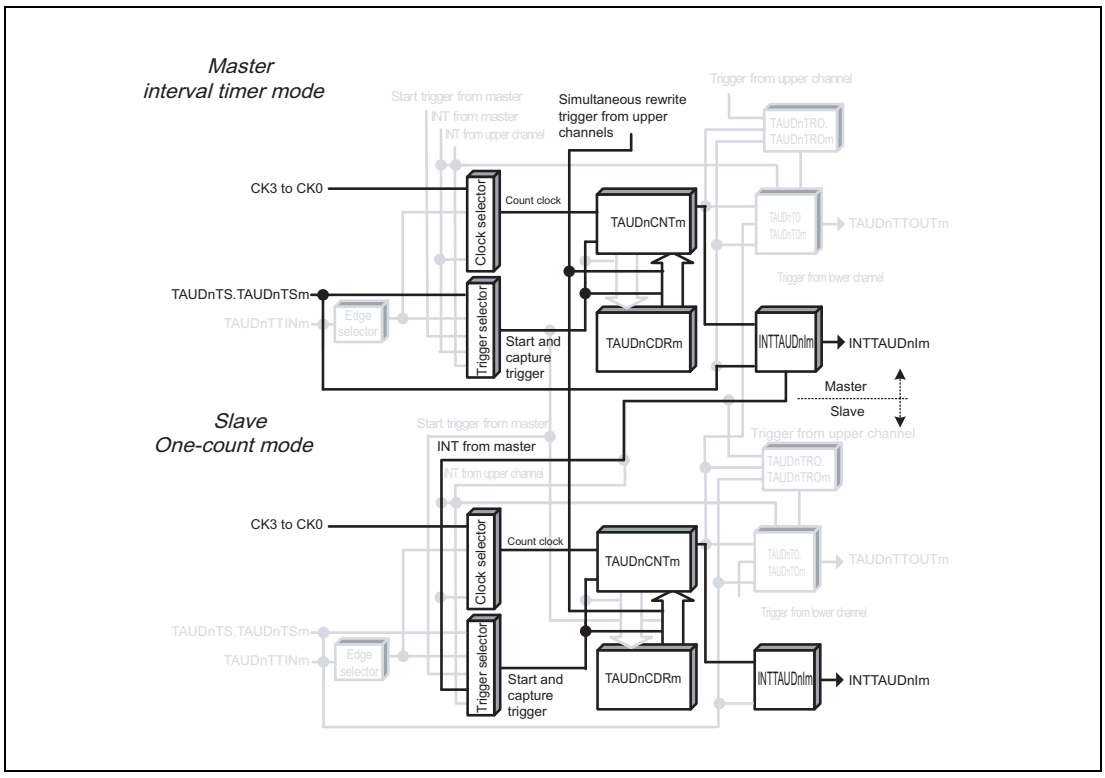


Figure 33.114 Block Diagram of A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

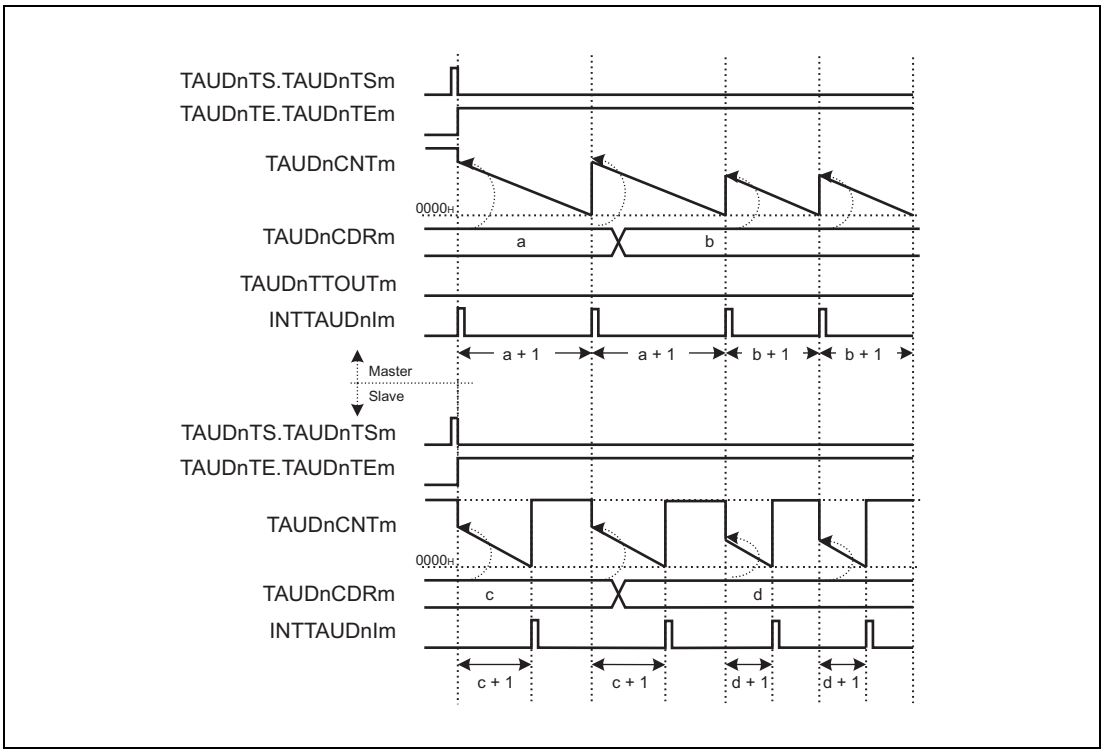


Figure 33.115 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

33.15.7 Triangle PWM Output Function

33.15.7.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUDnTTOUTm to be set using the master and slave channels respectively.

The master channel generates a carrier cycle. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slave counter.

Prerequisites

- Two channels
- The operating mode for the master channels should be set to interval timer mode. (See **Table 33.185, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function**.)
- The operating mode for slave channels should be set to count-up/-down mode. (See **Table 33.189, Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function**.)
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 33.7, Channel Output Modes**.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 33.7, Channel Output Modes**.)
- The following settings allow the TAUDnTTOUTm signal to be at high level during the down status of a carrier cycle.
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

Functional description

The counters are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for every channel. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation.

The current values of TAUDnCDRm (master and slave) are loaded into TAUDnCNTm (master and slave) and the counters start counting down from these values. When the TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and TAUDnTTOUTm signal of master toggles.

- Master channel:
  When the counter of master channel reaches 0000_H (pulse cycle time has elapsed), INTTAUDnIm is generated and the TAUDnTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and counts down.

- Slave channel:
  If INTTAUDnIm is generated on the master channel, the counter of the slave channel is triggered.
  - If the slave counter is counting down, the count direction changes.
  - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts to count down.

When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUDnIm is generated and the TAUDnTTOUTm (slave) signal is set/reset.
The counter continues count-up/-down and waits for the next INTTAUDnIm of the master channel.
Setting TAUDnTOL.TAUDnTOLm allows TAUDnTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values.

Conditions

This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.

33.15.7.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle [%] =

[(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave))/(TAUDnCDRm (master) + 1)] × 100

- Duty cycle = 100%
  TAUDnCDRm (slave) = 0000_H
- Duty cycle = 0%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

33.15.7.3 Block Diagram and General Timing Diagram

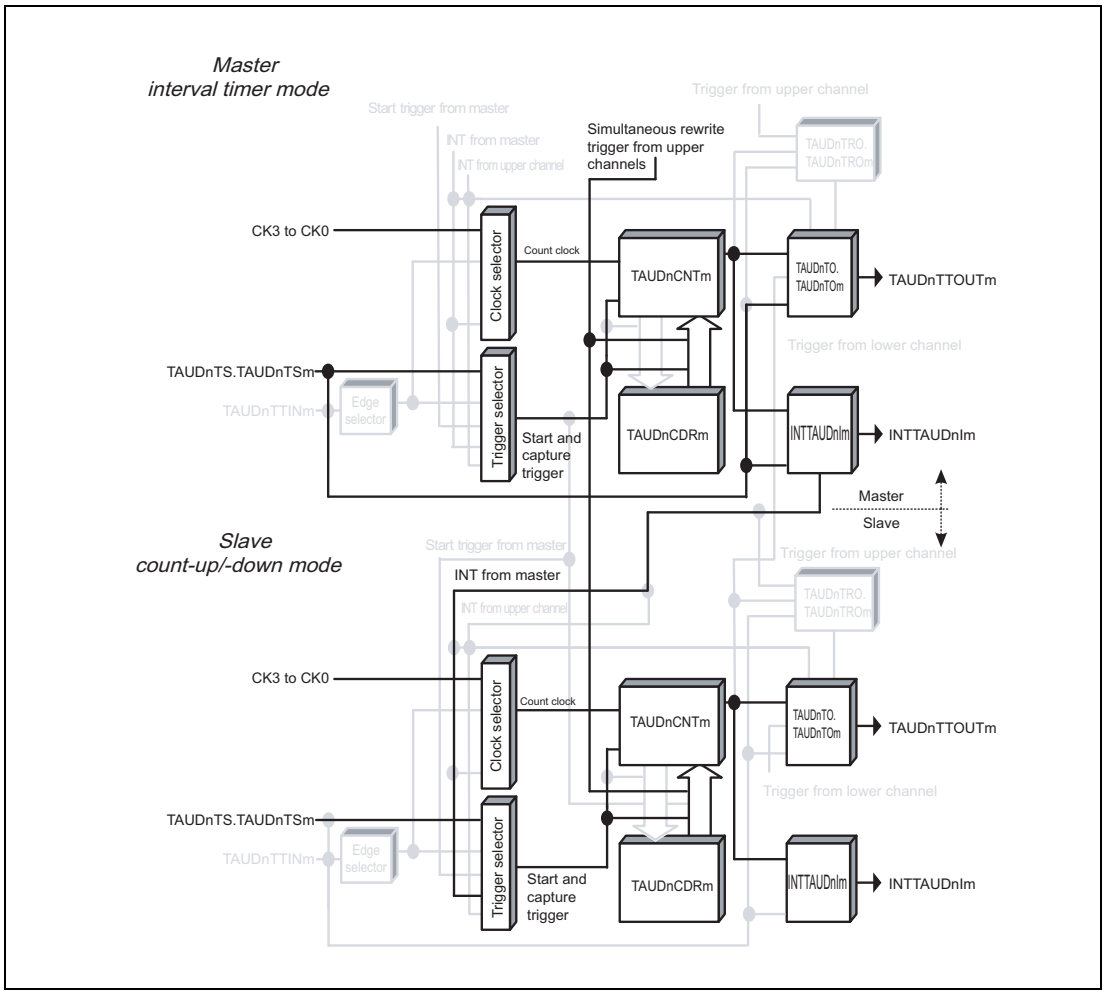


Figure 33.116 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

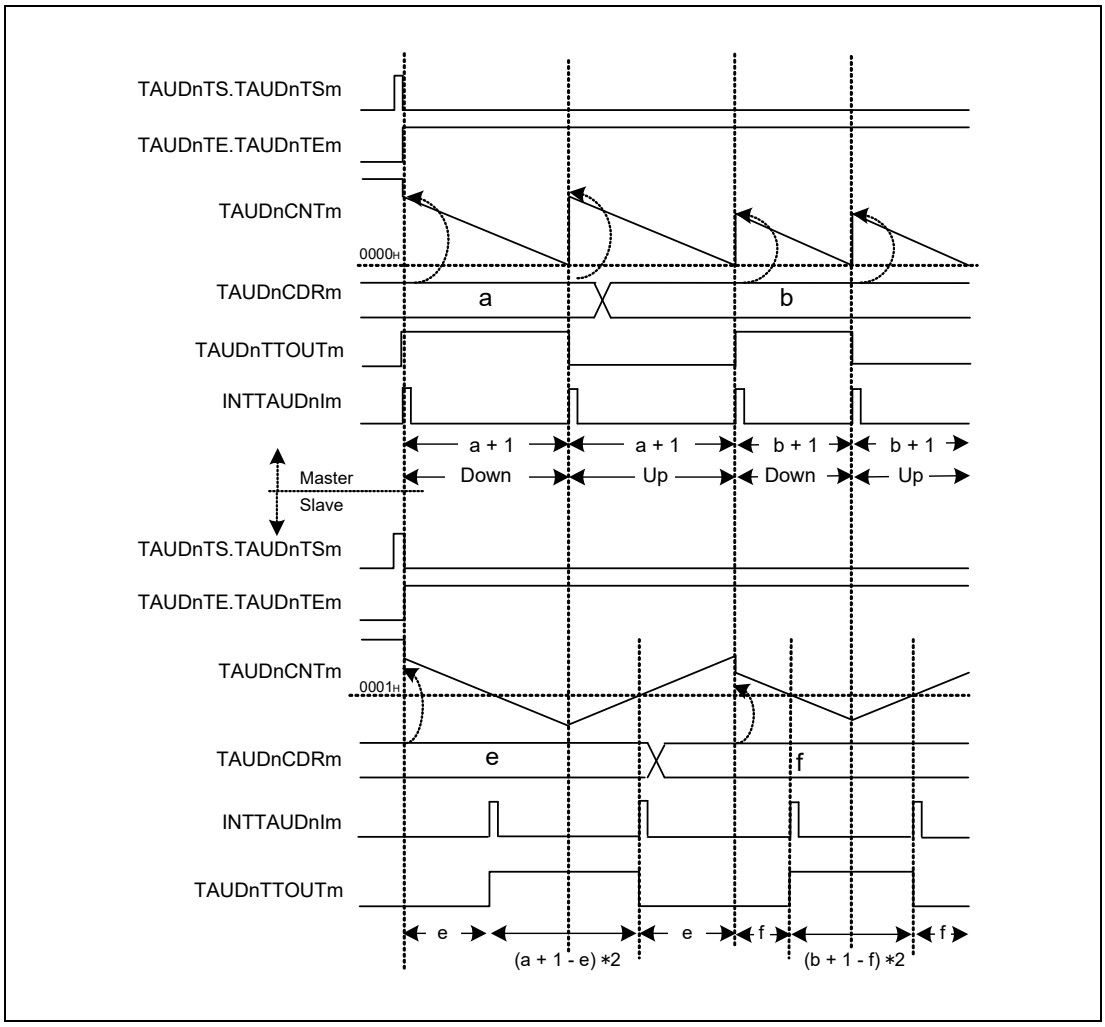


Figure 33.117 General Timing Diagram of Triangle PWM Output Function

33.15.7.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.185 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDnTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDnTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.186 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel

Table 33.187 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.188 Simultaneous Rewrite Settings for the Master Channel of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

33.15.7.5 Register Settings for Slave Channels

(1) TAUDnCMORm for Slave Channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.189 Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for Slave Channels

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.190 Contents of the TAUDnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels

Table 33.191 Control Bit Settings in Synchronous Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for Slave Channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.192 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.7.6 Operating Procedure for Triangle PWM Output Function

Table 33.193 Operating Procedure for Triangle PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.7.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.7.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master and slave channels is loaded into TAUDnCNTm to count down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnTTOUTm (master) is toggled. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) or counting is started in opposite direction. When TAUDnCNTm of slave channel reaches 0001_H: • INTTAUDnIm (slave) is generated. • TAUDnTTOUTm (slave) is set in the count-down status or reset in count-up status.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.7.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 6_H

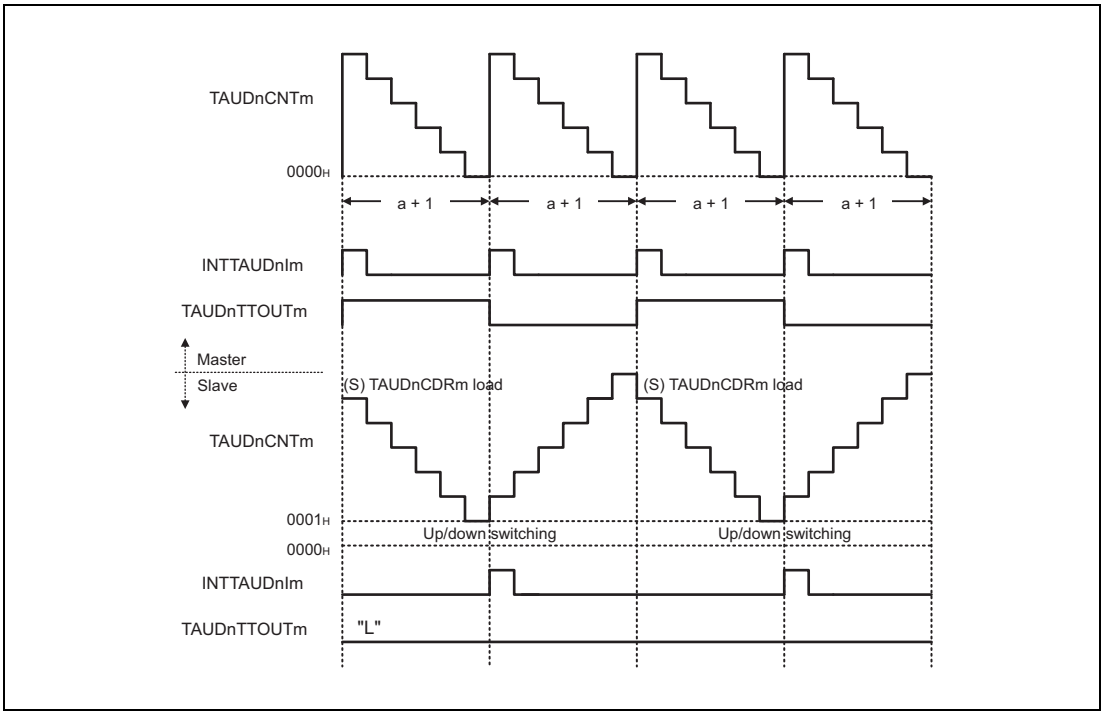


Figure 33.118 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value ≥ TAUDnCDRm (master) value + 1, INTTAUDnIm of the slave channel is not generated while counting down. TAUDnTTOUTm remains low because there is no set signal to be detected.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 0_H

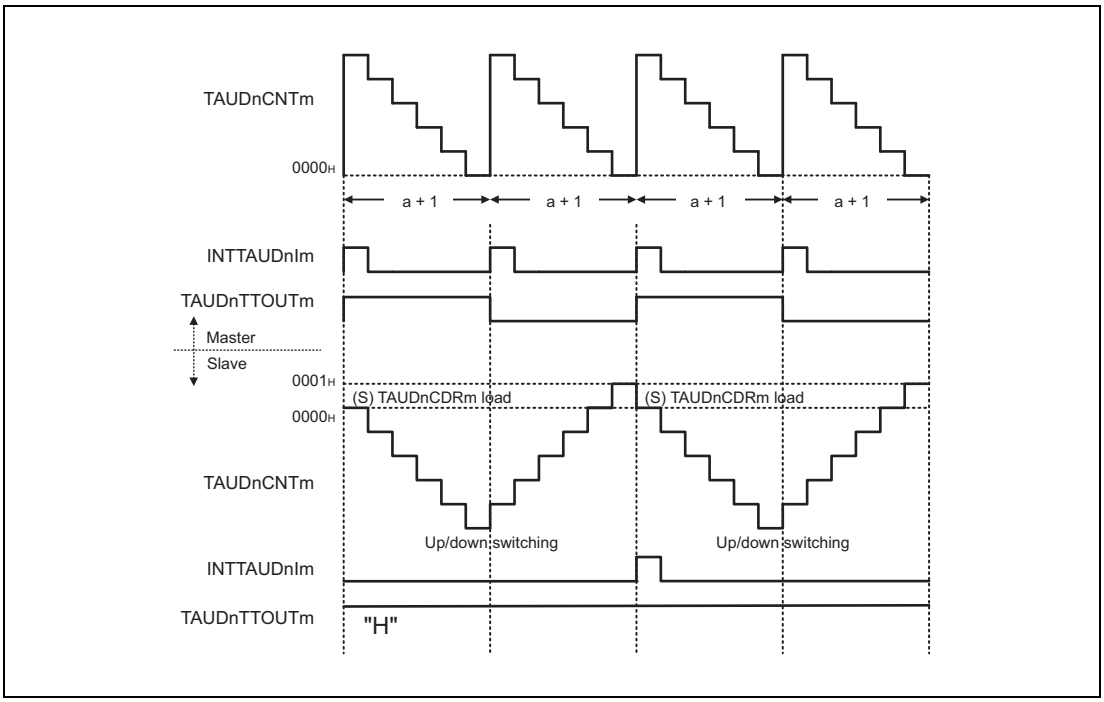


Figure 33.119 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave) = 0000_H, INTTAUDnIm of slave channel is not generated while counting up. TAUDnTTOUTm remains high because there is no reset signal to be detected.

33.15.8 Triangle PWM Output Function with Dead Time

33.15.8.1 Overview

Summary

This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals with dead time are output via TAUDnTTOUTm of the slave channels 2 and 3, enabling the pulse cycle (frequency) and the duty cycle of TAUDnTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on the master channel. The first pulse controls the down status of the slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUDnTTOUTm of slave channels to be set/reset. Depending on the settings of TAUDnTDL.TAUDnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUDnTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select even channel CH (a) and odd channel CH (a + 1).
- The operating mode for the master channel should be set to interval timer mode. (See **Table 33.195, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**)
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even channel (a), and slave channel 3 is an odd channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operating mode for slave channel 2 should be set to `count-up/-down` mode (See **Table 33.199, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time**). Slave channel 2 should be an even channel.
- The operating mode for slave channel 3 should be set to one-count mode (See **Table 33.203, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time**). Slave channel 3 should be an odd channel.
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 33.7, Channel Output Modes**)
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See **Section 33.7, Channel Output Modes**)

- The following settings make a TAUDnTTOUTm signal at high level during the down status of the carrier cycle:
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

NOTE

The triangle PWM output function with dead time does not use slave channel 1. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counter starts by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUDnTTOUTm signal is toggled.

- Master channel:
  When the counter of the master channel reaches 0000_H, an INTTAUDnIm is generated and the TAUDnTTOUTm signal is toggled. The TAUDnCDRm value is reloaded to continue counting down.
- Slave channel 2:
  If INTTAUDnIm is generated on the master channel, the counter of slave channel 2 is triggered.
  - If the slave counter is counting down, the counting direction changes.
  - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts counting down.

  The counter continues to count down/up and waits for the next INTTAUDnIm of the master channel.
  When the counter value of slave channel 2 reaches 0001_H, INTTAUDnIm is generated.
- Slave channel 3:
  If INTTAUDnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDnCDRm (slave 3) is loaded into TAUDnCNTm (slave 3) and the counter starts to count down from the TAUDnCDRm value.
  When the counter reaches 0000_H, INTTAUDnIm occurs. The counter returns to $FFFF_H$ and waits for the next INTTAUDnIm of slave channel 2.

As described in **Table 33.194, Operation of TAUDnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (right after occurrence of an interrupt or after dead time has elapsed) depends on the TAUDnTDL.TAUDnTDLm setting of the corresponding channel.

The setting of TAUDnTOL.TAUDnTOLm also determines whether a high level signal (TAUDnTOL.TAUDnTOLm = 0) or a low level signal (TAUDnTOL.TAUDnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values.

TAUDnTTOUTm can be 100% output by setting the TAUDnCDRm value of slave channel 2 to 0000_H.

Conditions

This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.

TAUDnTOL.TAUDnTOLm and TAUDnTDL.TAUDnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDnTDL.TAUDnTDLm.

Table 33.194 Operation of TAUDnTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

TAUDnTDL. TAUDnTDLm	Count Direction of Slave Channel 2 upon Occurrence of Interrupt	TAUDnTTOUTm Set/Reset Timing
0	Down	Set after elapse of dead time
	Up	Reset right after interrupt occurs
1	Down	Set right after interrupt occurs
	Up	Reset after elapse of dead time

33.15.8.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (normal phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

33.15.8.3 Block Diagram and General Timing Diagram

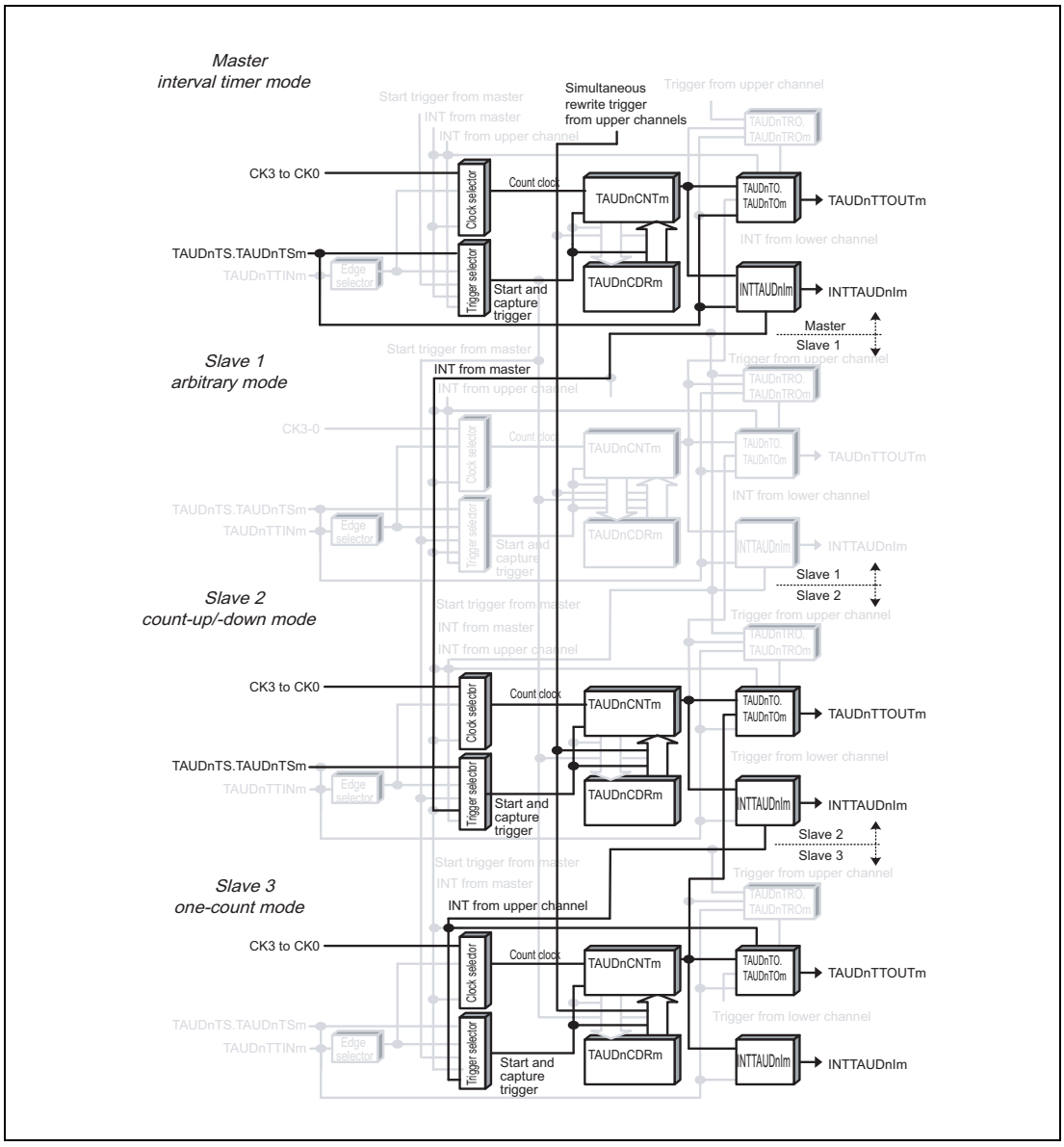


Figure 33.120 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Slave channel 2:
  - INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
  - TAUDnTDL.TAUDnTDLm = 0
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

- Slave channel 3:
  - Enables start trigger detection during counting (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnTDL.TAUDnTDLm = 1
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

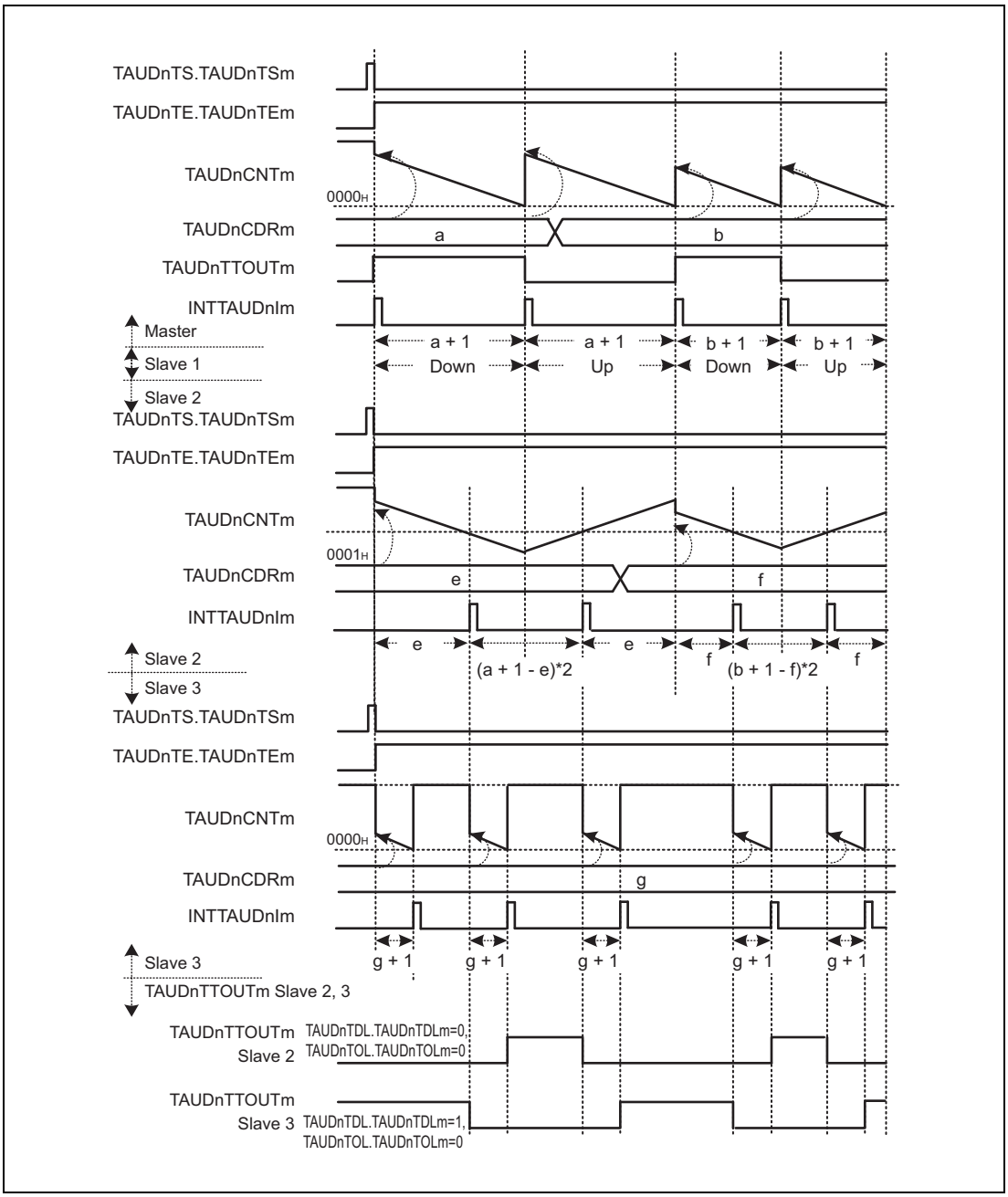


Figure 33.121 General Timing Diagram of Triangle PWM Output Function with Dead Time

33.15.8.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.195 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDnTTOUTm is not toggled at the beginning of operation. 1: INTTAUDnIm is generated and TAUDnTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.196 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 33.197 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.198 Simultaneous Rewrite Setting for the Master Channel of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

33.15.8.5 Register Settings for Slave Channel 2

(1) TAUDnCMORm for Slave Channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.199 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for Slave Channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.200 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 2

Table 33.201 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous Rewrite for Slave Channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.202 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.8.6 Register Settings for Slave Channel 3

(1) TAUDnCMORm for Slave Channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.203 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time output signal of the TAUDnTTOUTm generation unit
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channels 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.204 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel 3

Table 33.205 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous Rewrite for Slave Channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.206 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.8.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 33.207 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.8.4, Register Settings for the Master Channel**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.8.5, Register Settings for Slave Channel 2**. Slave channel 3: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.8.6, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channel 2 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 2) or counting is started in opposite direction. When TAUDnCNTm of slave channel 2 reaches 0001_H: • INTTAUDnIm (slave 2) is generated. • TAUDnCDRm value of slave channel 3 is loaded into TAUDnCNTm perform counting down. When TAUDnCNTm of slave channel 3 reaches 0000_H: • INTTAUDnIm is generated.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.15.8.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram in **Figure 33.122**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

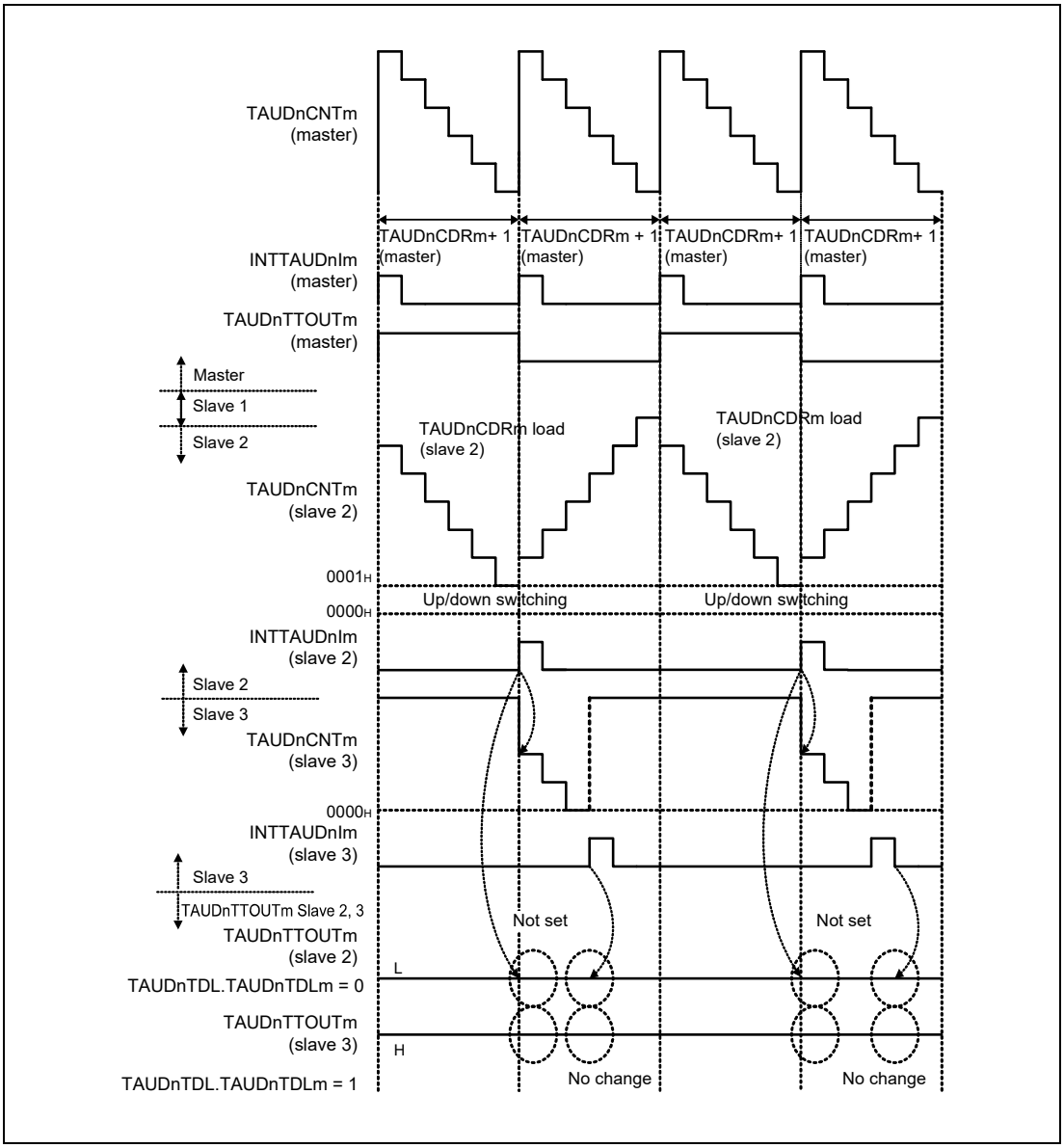


Figure 33.122 TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave 2) is greater than TAUDnCDRm (master), the counter of slave channel does not reach 0000_H while counting down. Therefore, TAUDnTTOUTm signal is not set/reset and remains in the initial state. This signal becomes a reset signal because an interrupt occurs on slave channel 2 during count-up operation.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram in **Figure 33.123**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

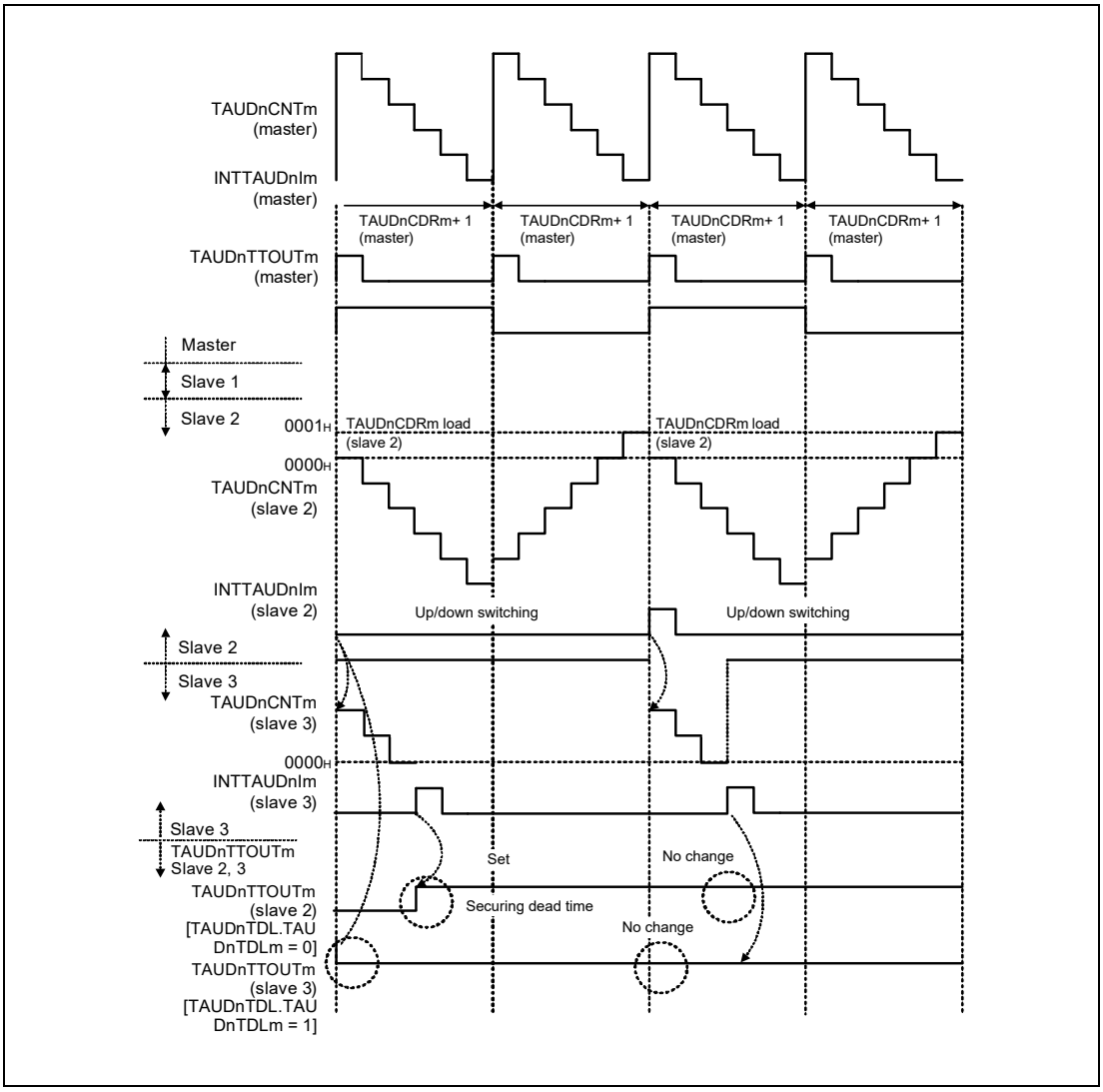


Figure 33.123 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave 2) = 0000_H, the slave channel counter does not reach 0001_H while counting up. Therefore, no INTTAUDnIm is generated during count-up operation.
  - The set conditions for a channel with TAUDnTDL.TAUDnTDLm = 0 are met after elapse of dead time. TAUDnTTOUTm is left in a newly set state even if a set/reset is made because no reset conditions are satisfied on such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts. However, TAUDnTTOUTm is left in an initial state on the slave channel with TAUDnTDL.TAUDnTDLm = 1 because no reset conditions are satisfied on that channel.

33.15.9 A/D Conversion Trigger Output Function Type 2

33.15.9.1 Overview

Summary

This function is identical to **Section 33.15.7, Triangle PWM Output Function**, except that TAUDnTTOUTm is not output.

This function is enabled by setting channel output mode for slave to independent channel output mode controlled by software.

33.15.9.2 Block Diagram and General Timing Diagram

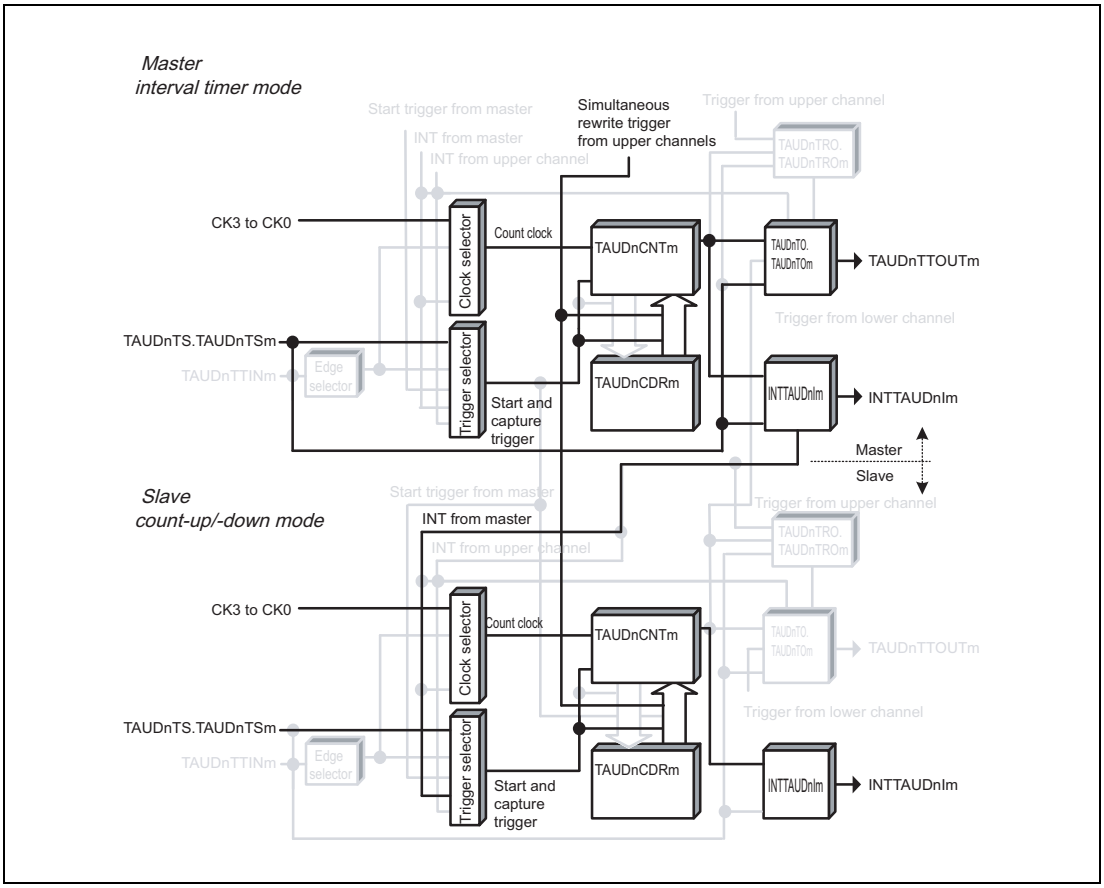


Figure 33.124 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

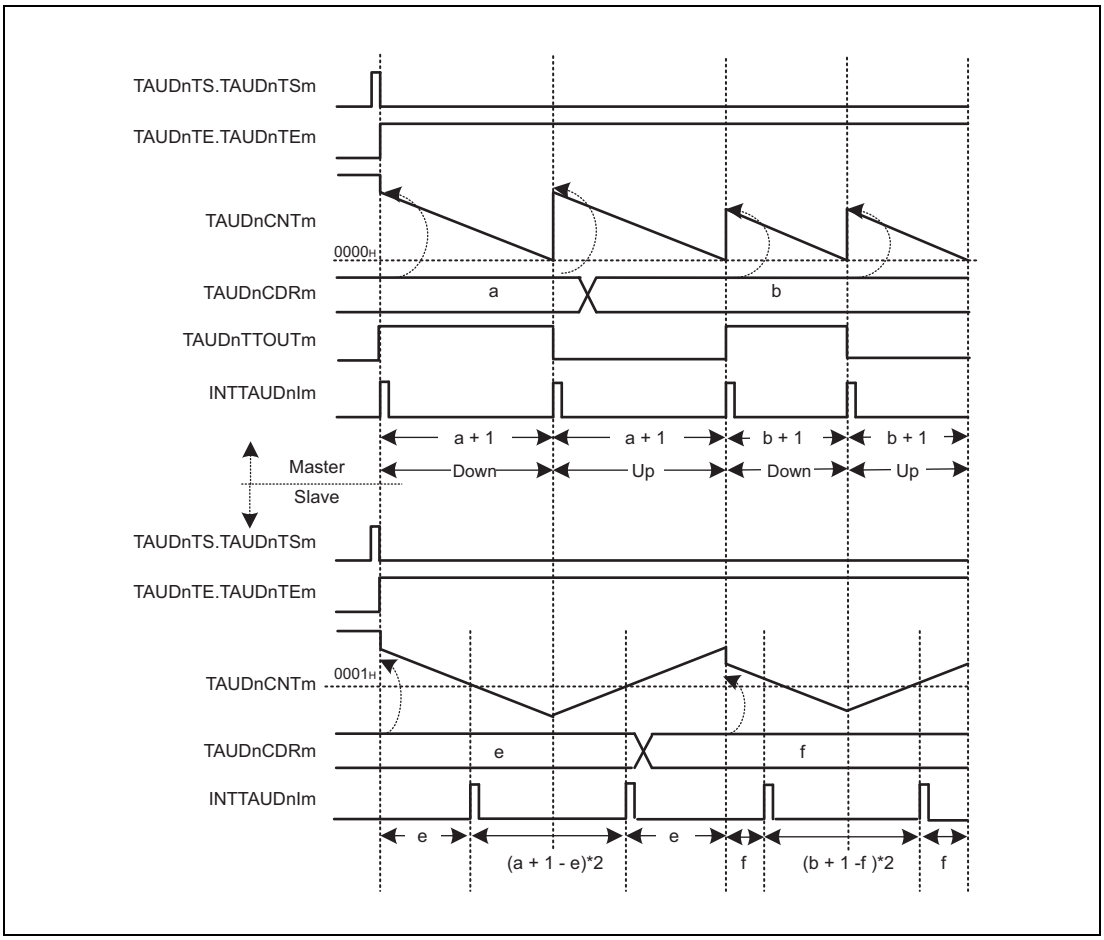


Figure 33.125 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

33.15.10 Interrupt Request Signals Culling Function

33.15.10.1 Overview

Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The interrupt request signals culling function is a sub function of the following functions:

- PWM Output Function (See **Section 33.15.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 33.15.7, Triangle PWM Output Function**)
- Triangle PWM Output Function with Dead Time
  (See **Section 33.15.8, Triangle PWM Output Function with Dead Time**)

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 33.208, Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function**)
- The operation mode of the slave channel must be set to Event Count Mode. (See **Table 33.211, Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function**)
- This function does not use TAUDnTTOUTm.

Functional description

The counters (master and slave) are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for both channels. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDnCDRm) are written to the counter (TAUDnCNTm).

- Master channel:
  When the counter of the master channel reaches 0000_H, INTTAUDnIm is generated and TAUDnCDRm value is reloaded to TAUDnCNTm.
- Slave channel:
  Every time the master channel generates an INTTAUDnIm, the counter of the slave channel decrements by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUDnCNTm (slave) to reload the value of TAUDnCDRm, and an INTTAUDnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channels, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm of master and slave channels stops but retains its value.

Conditions

This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.

33.15.10.2 Equations

Interrupt division operator = TAUDnCDRm (slave channel)

– One INTTAUDnIm is generated for the INTTAUDnIm count of the master channel defined by TAUDnCDRm (slave channel) + 1.

33.15.10.3 Block Diagram and General Timing Diagram

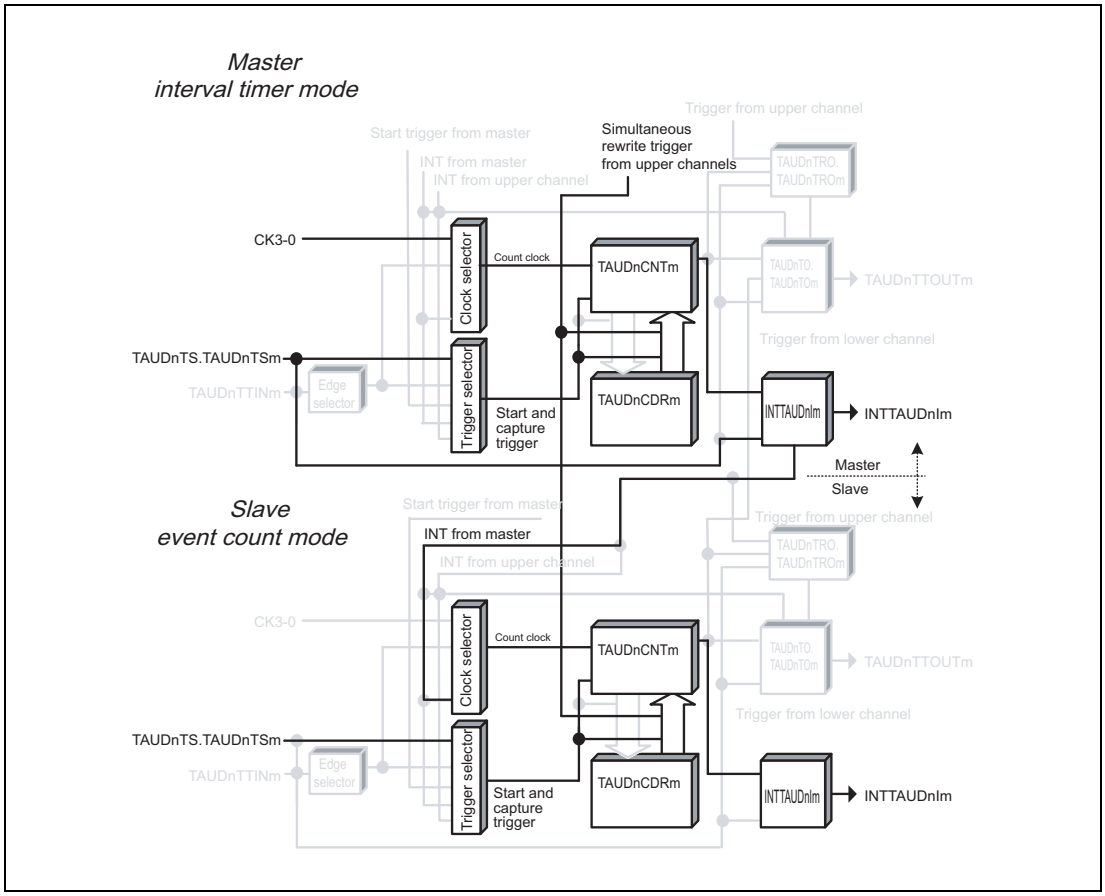


Figure 33.126 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

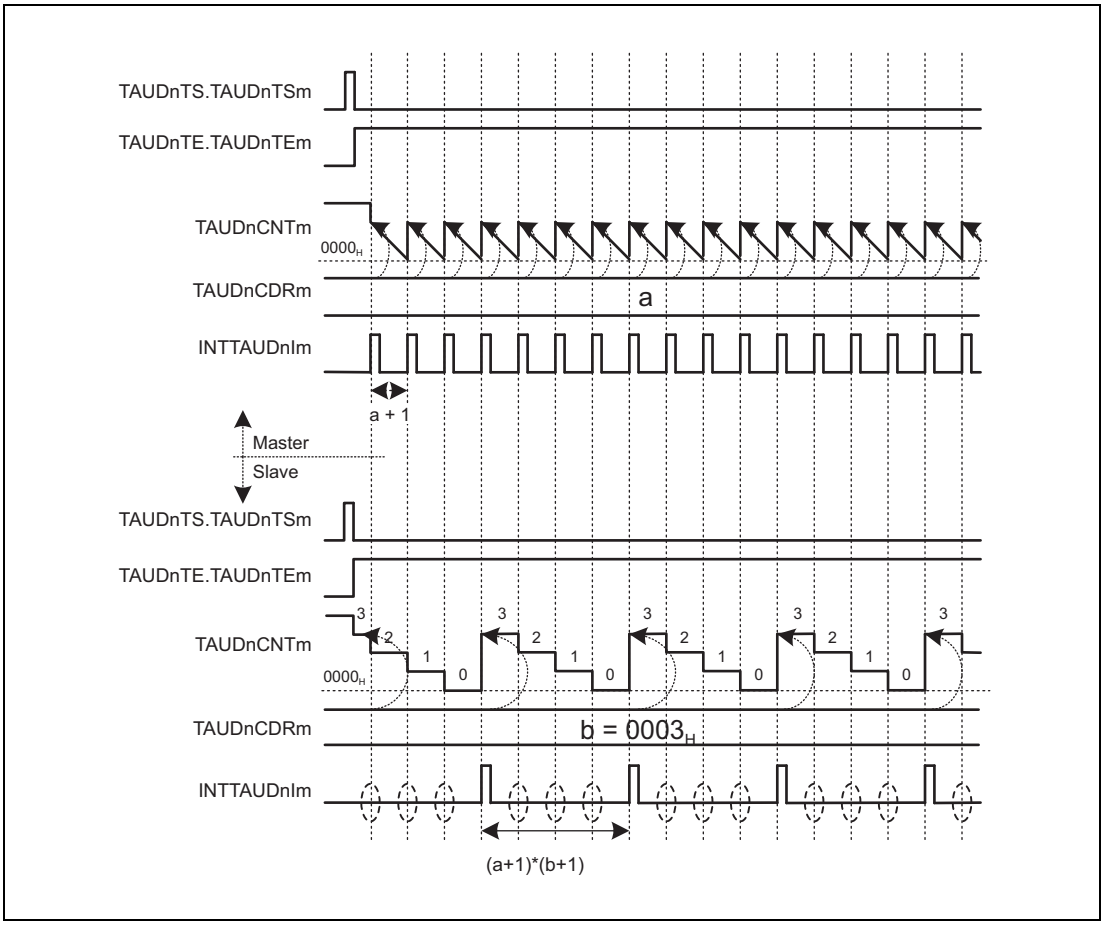


Figure 33.127 General Timing Diagram of Interrupt Request Signals Culling Function

33.15.10.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.208 Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.209 Contents of the TAUDnCMURm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.210 Simultaneous Rewrite Settings for the Master Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.10.5 Register Settings for Slave Channel

(1) TAUDnCMORm for Slave Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.211 Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for Slave Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.212 Contents of the TAUDnCMURm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous Rewrite for Slave Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.213 Simultaneous Rewrite Settings for Slave Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects a master channel for simultaneous rewrite triggers. 1: Selects an upper channel outside the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

33.15.10.6 Operating Procedure for Interrupt Request Signals Culling Function

Table 33.214 Operating Procedure for Interrupt Request Signals Culling Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.10.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.15.10.5, Register Settings for Slave Channel**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation. • TAUDnCNTm of slave channels counts down each time INTTAUDnIm of master channel is detected. When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • The TAUDnCDRm value is loaded in TAUDnCNTm (slave) and count operation continues.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

33.15.10.7 Specific Timing Diagrams

(1) Interrupt count (master) = interrupt count (slave)

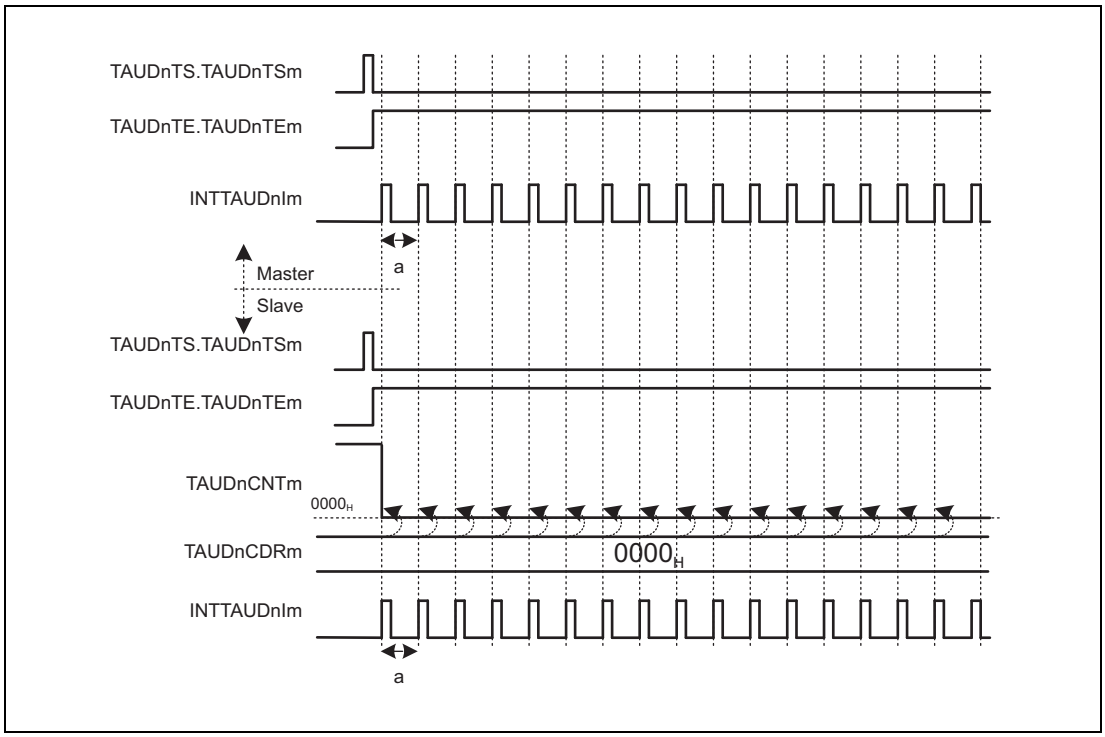


Figure 33.128 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm = 0000_H, the TAUDnCDRm value of slave channel is loaded into TAUDnCNTm each time INTTAUDnIm of master channel is detected. In other words, TAUDnCNTm is always 0000_H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

33.16 Synchronous Non-Complementary and Complementary Modulation Output Functions

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and seven slave channels.

33.16.1 Non-Complementary Modulation Output Function Type 1

33.16.1.1 Overview

Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUDnTTOUTm depending on the values of the real-time output bits (TAUDnTRO.TAUDnTROm) and the modulation output enable bits (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 33.216, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1**).
- The operating mode for slave channels 1 to 7 should be set to one-count mode (See **Table 33.219, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1** and **Table 33.222, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1**).
- TAUDnTTOUTm is not used with the master channel of this function.
- TAUDnTTOUTm of slave channel 1 is not used with this function, but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 33.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output (See **Section 33.7, Channel Output Modes**).
- TAUDnCDRm of slave channel 1 should be set to 0000_H.

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.

- Slave channel 1:
  Slave channel 1 is set as a channel that triggers real-time output (TAUDnTRC.TAUDnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDnCDRm is fixed to 0000_H), the value of real-time output bit (TAUDnTRO.TAUDnTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to the TAUDnTTOUTm output. After that, the counter returns to $FFFF_H$ and waits for the next interrupt of master channel.

- Slave channel 2:
  Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in **Table 33.215, TAUDnTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**, a signal output from TAUDnTTOUTm depends on the value of the real-time output bit (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs PWM corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 33.215 TAUDnTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDnTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.
- TAUDnCDRm value of slave channel 1 should be set to 0000_H so that a real-time output is triggered at the same time with PWM generation on slave channels 2 to 7.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

33.16.1.2 Equations

Slave channels 2 to 7:

Pulse period = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

33.16.1.3 Block Diagram and General Timing Diagram

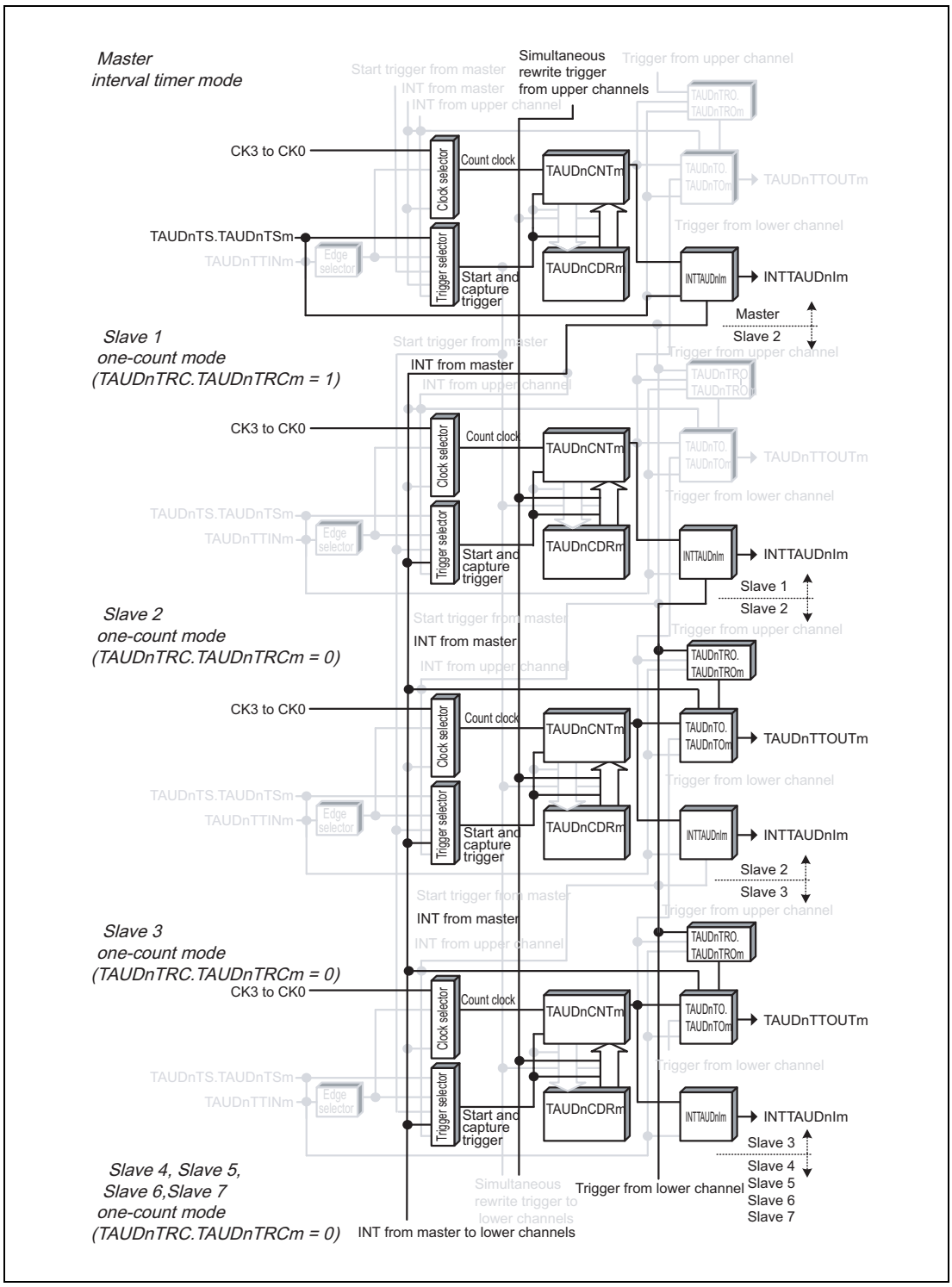


Figure 33.129 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

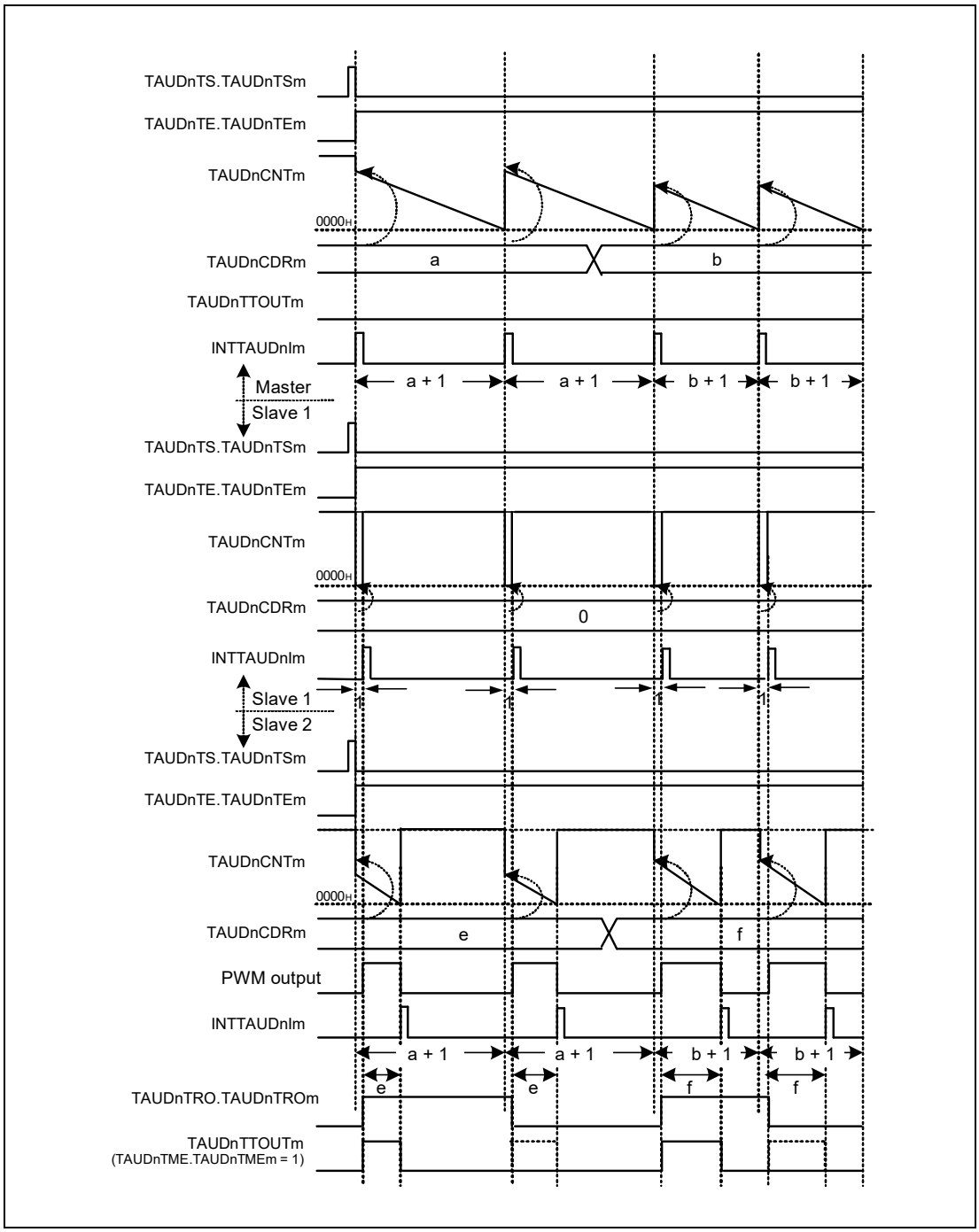


Figure 33.130 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

NOTE

TAUDnTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

33.16.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.216 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.217 Contents of the TAUDnCMURm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.218 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires an upper channel higher than the master channel that operates with **Section 33.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions.

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  In addition, TAUDnCDRm settings for this channel are as follows.
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × Interrupt count) – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

33.16.1.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.219 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.220 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.221 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.1.6 Register Settings for Slave Channels 2 to 7

(1) TAUDnCMORm for Slave Channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.222 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for Slave Channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.223 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for Slave Channels 2 to 7

Table 33.224 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous Rewrite of Slave Channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.225 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 33.226 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.1.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.1.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.1.6, Register Settings for Slave Channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle with TAUDnCDRm of master channel, 0000_H in TAUDnCDRm of slave channel 1, and duty width with TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 33.226 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (2/2)

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value of master channel is reloaded into TAUDnCNTm to continue counting down. • PWM output signals of slave channels 2 to 7 are set. • TAUDnCDRm value of slave channel 1 is reloaded into TAUDnCNTm to perform counting down. • TAUDnCDRm value of slave channels 2 to 7 is reloaded into TAUDnCNTm to perform counting down. • When the counter of slave channel 1 reaches 0000_H: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDnTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set. TAUDnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of a pair of slave channels.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.16.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

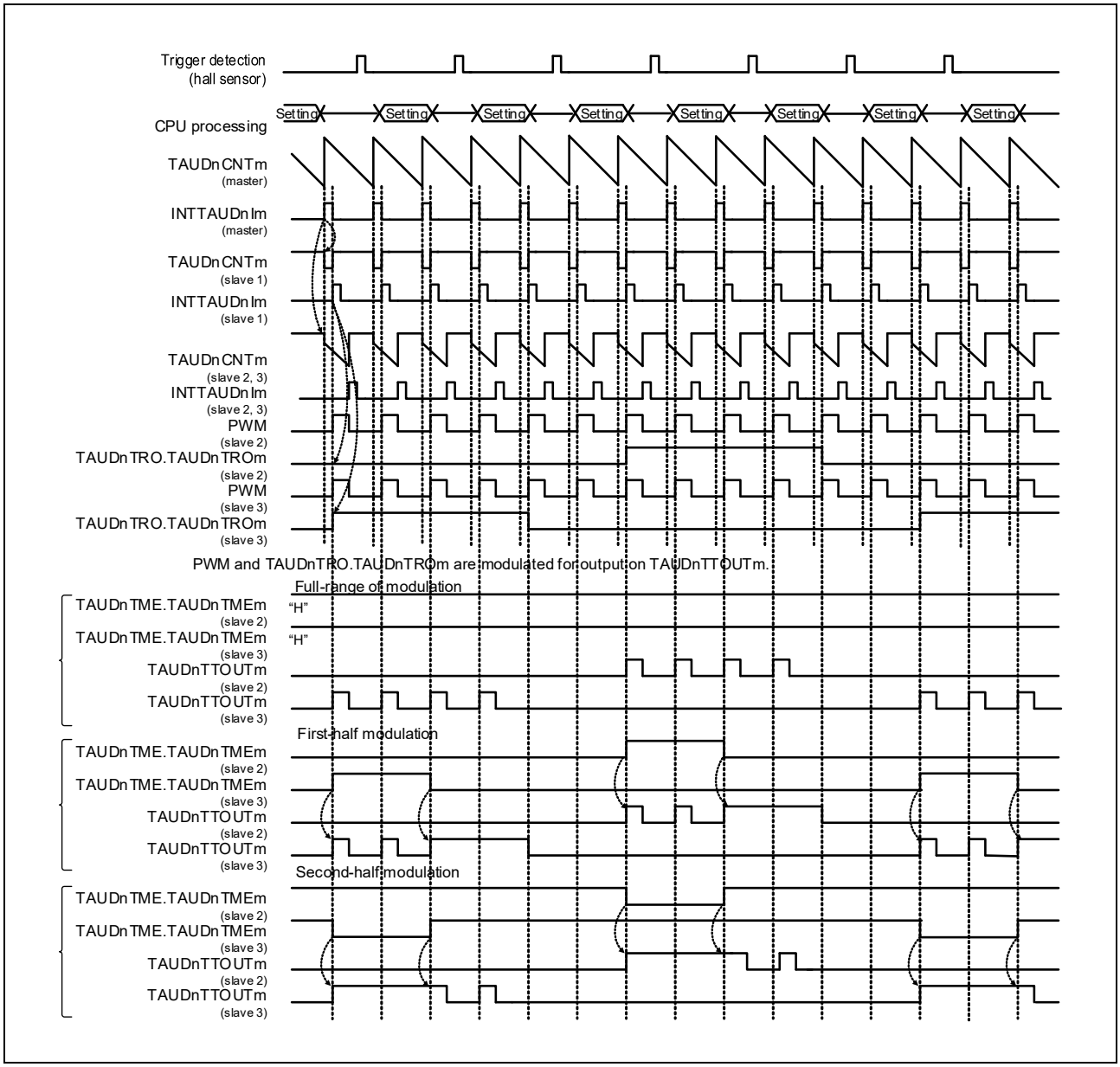


Figure 33.131 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.
The "Setting" symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.
TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUDnTTOUTm.
A TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

33.16.2 Non-Complementary Modulation Output Function Type 2

33.16.2.1 Overview

Summary

This function outputs a triangular PWM output signal, a high-level signal, or low-level signal from TAUDnTTOUTm depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output enable bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 33.228, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 33.232, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channels 2 to 7 should be set to count-up/-down mode (See **Table 33.235, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2**).
- The output mode for the master channel should be set to independent channel output mode 1 (See **Section 33.7, Channel Output Modes**).
- This function does not use TAUDnTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 33.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output (See **Section 33.7, Channel Output Modes**).

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm).

- Master channel:
  The counter of master channel starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.
- Slave channel 1:
  When slave channel 1 detects an interrupt from the master channel, the TAUDnCNTm value is decremented. When an interrupt from the master channel is detected (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, the TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.
  Since slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1), if an interrupt occurs on slave channel 1, the real-time output bit (TAUDnTRO.TAUDnTROm) of the channel which monitors an interrupt on the corresponding channel is reflected to the TAUDnTTOUTm output.

- Slave channel 2:
  Once detecting an interrupt from the master channel, TAUDnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDnCDRm value is reloaded and then the counter starts to count down.
  If TAUDnCNTm = 0001_H, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operate like slave channel 2.

A signal that is output from TAUDnTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm) and a modulation output bit value (TAUDnTME.TAUDnTMEm) of the slave channel, as described in **Table 33.227, TAUDnTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**.

This function cannot make a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 33.227 TAUDnTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDnTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

33.16.2.2 Equations

Slave channels 2 to 7:

Carrier cycle (down/up)

= [TAUDnCDRm (master) + 1] × 2 × count clock cycle

Duty time

= [TAUDnCDRm (master) + 1 – TAUDnCDRm (slave)] × 2 × count clock cycle

33.16.2.3 Block Diagram and General Timing Diagram

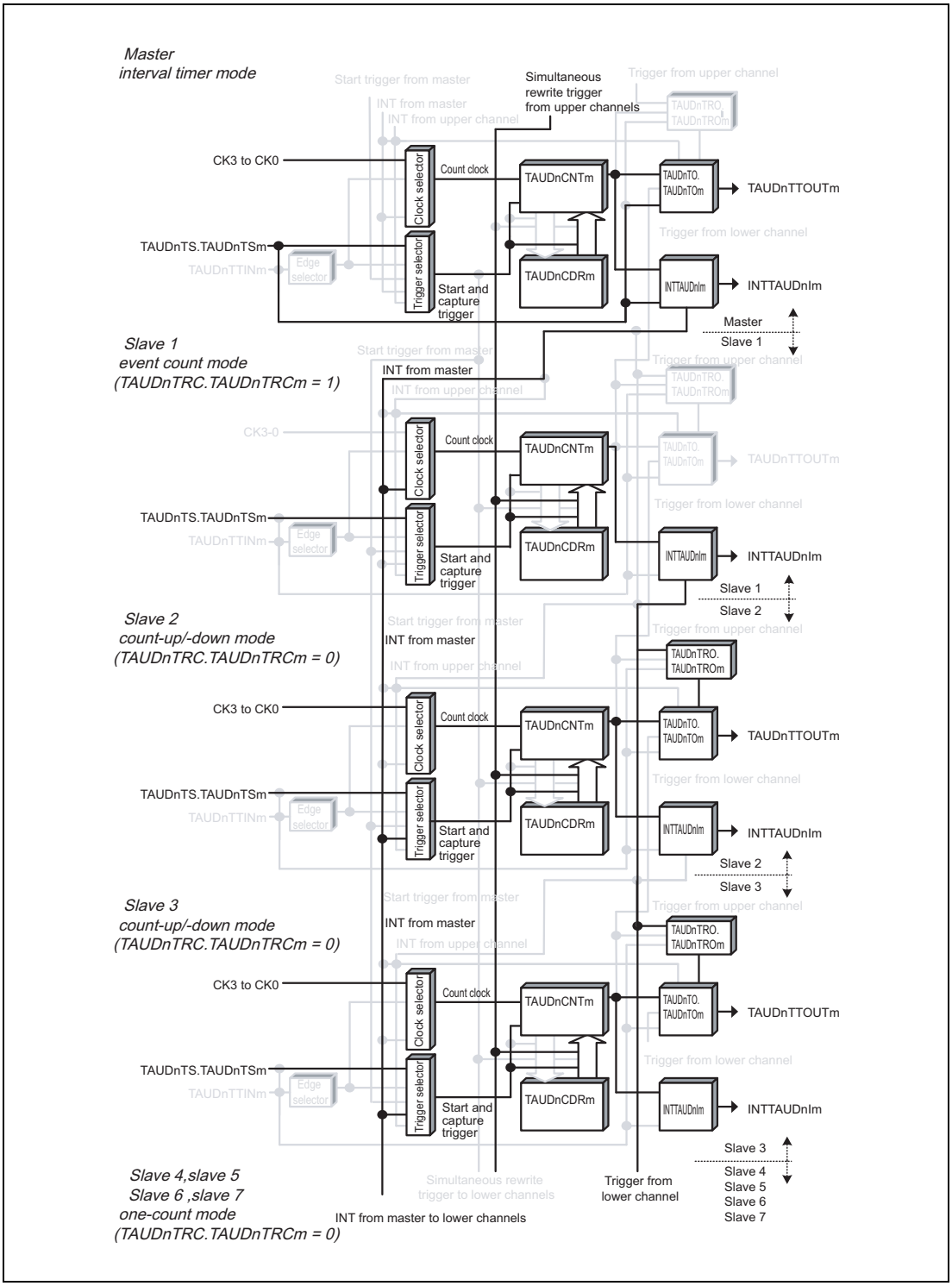


Figure 33.132 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

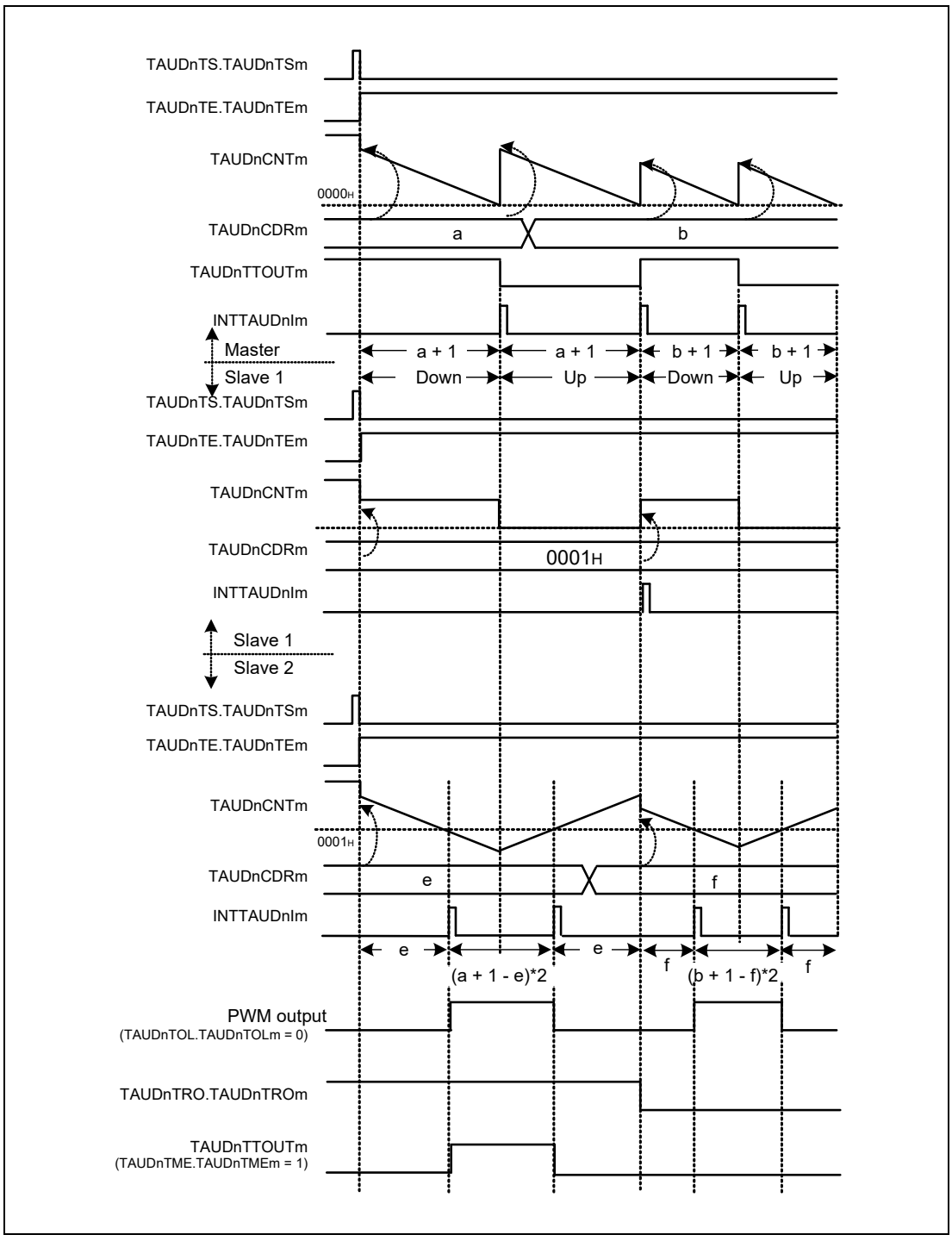


Figure 33.133 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

33.16.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.228 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.229 Contents of the TAUDnCMURm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for the Master Channel

Table 33.230 Control Bit Settings for the Master Channel in Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (toggle mode with TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0.
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.231 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

33.16.2.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.232 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.233 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.234 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.2.6 Register settings for slave channels 2 to 7

(1) TAUDnCMORm for Slave Channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.235 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: The up/down output trigger signal of the master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channels 2 to 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.236 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output Mode for Slave Channels 2 to 7

Table 33.237 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: The upper channel generates the real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous Rewrite for Slave Channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.238 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 33.239 Operating Procedure for Non-Complementary Modulation Output Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.2.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.2.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.2.6, Register settings for slave channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set pulse cycle in TAUDnCDRm of master channel, and in TAUDnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates a real-time output trigger. Set duty width in TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 33.239 Operating Procedure for Non-Complementary Modulation Output Function Type 2

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	The TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. The TAUDnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • The TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. • TAUDnCNTm of slave channels 2 to 7 reloads the TAUDnCDRm value, or performs counting in opposite direction. • At the same timing when the TAUDnCDRm value is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDnTTOUTm output. • When slave channel 1 detects an interrupt from the master channel for the (TAUDnCDRm + 1) times: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDnTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0001_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set/reset.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.16.2.8 Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

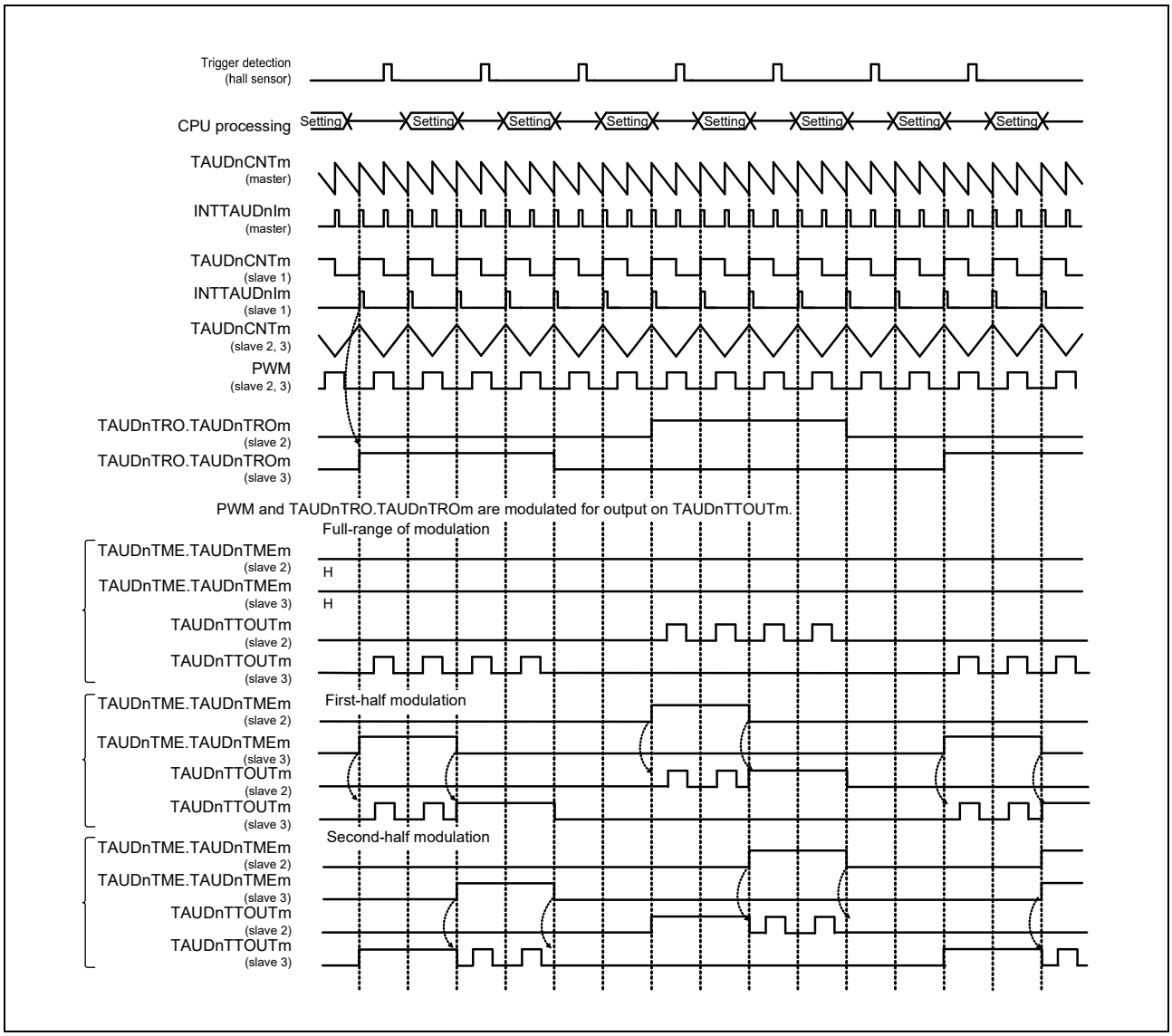


Figure 33.134 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.
The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.
TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).
TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

33.16.3 Complementary Modulation Output Function

33.16.3.1 Overview

Summary

This function outputs a triangle PWM output signal, a high-level signal, or low-level signal from TAUDnTTOUTm with dead time added, depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels, and an output level bit value (TAUDnTDL.TAUDnTDLm). Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 33.241, Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 33.245, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See **Table 33.248, Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See **Table 33.252, Contents of the TAUDnCMORm Register for Slave Channels 3, 5, and 7 of the Complementary Modulation Output Function**). In addition, as the number of occurrences of an interrupt for slave channels 3, 5 and 7 within the carrier cycle is not uniquely determined, do not use the interrupt as an interrupt source.
- The output mode for the master channels should be set to independent channel output mode 1 (See **33.7, Channel Output Modes**).
- This function does not use TAUDnTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 33.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with complementary modulation output (See **Section 33.7, Channel Output Modes**).

Functional description

- Master channel:
  The counter of the master channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) of the master channel is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.

- Slave channel 1:
  When the counter reaches 0000_H, slave channel 1 waits for the next interrupt from the master channel. And the TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1) and INTTAUDnIm is generated.
  Slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1). The value of real-time output bit (TAUDnTRO.TAUDnTROm) of each channel is applied to the channel that detects the occurrence of an interrupt on slave channel 1 by an interrupt. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.
- Slave channel 2:
  When the slave channel 2 counter reaches 0001_H, the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000_H, an interrupt occurs.
- Slave channels 2 and 3:
  The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.
- Slave channels 4 to 7:
  Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

A signal that is output from TAUDnTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm), a modulation output bit value (TAUDnTME.TAUDnTMEm), and an output level bit value (TAUDnTDL.TAUDnTDLm) of the slave channel, as described in **Table 33.240, TAUDnTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function**.

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit).

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 1 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm of one channel is set to 1, TAUDnTTOUTm outputs the corresponding PWM of the channel.
  - If TAUDnTRO.TAUDnTROm of both channels is set to 0, TAUDnTTOUTm of a pair outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 0 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm is set to 1, TAUDnTTOUTm of the channel outputs a high-level signal.
  - If TAUDnTRO.TAUDnTROm is set to 0, TAUDnTTOUTm of the channel outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDnTTOUTm are inverted. The PWM signal is negative logic.

Table 33.240 TAUDnTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

TAUDnTME. TAUDnTME02	TAUDnTME. TAUDnTME03	TAUDnTRO. TAUDnTRO02	TAUDnTRO. TAUDnTRO03	TAUDnTDL. TAUDnTDL02	TAUDnTDL. TAUDnTDL03	TAUDnTTOUT2 Output	TAUDnTTOUT3 Output
0	0	0	0	X	X	Low level	Low level
		0	1	1	0	Low level	High level
		1	0	0	1	High level	Low level
		1	1	X	X	Setting prohibited	Setting prohibited
1	1	0	0	X	X	Low level	Low level
		0	1	1	0	~PWM	PWM
		1	0	0	1	PWM	~PWM
		1	1	X	X	Setting prohibited	Setting prohibited

NOTES

1. In the above table, PWM indicates a positive PWM signal and ~PWM indicates an inverted PWM signal (positive logic). PWM and ~PWM are set by TAUDnTDL.TAUDnTDLm.
2. Any settings not listed above are prohibited.

- If TAUDnTME.TAUDnTMEm is continuously set to 1 while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the first half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the second half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, second-half modulation is applied.

- Whether dead time is added to a normal or reverse phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUDnTDL.TAUDnTDLm bit value.
  - If TAUDnTDL.TAUDnTDLm = 0, dead time is added to a normal phase PWM signal.
  - If TAUDnTDL.TAUDnTDLm = 1, dead time is added to a reverse phase PWM signal.
  - The operation defined by a TAUDnTDL.TAUDnTDLm bit value should be conducted by application software during operation. To modify TAUDnTDL.TAUDnTDLm, rewrite it during the period when TAUDnTRO.TAUDnTROm is 00_B.
- The TAUDnCDRm value of slave channel 1 should be set to the value to generate INTTAUDnIm of slave channel 1 at a carrier cycle (peak interrupt timing).
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- This function enables simultaneous rewrite. See **Section 33.6, Simultaneous Rewrite**.

33.16.3.2 Equations

Pulse period = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

For slave channels 2 and 3:

PWM signal width (positive phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

For slave channels 4 to 7:

Slave channels 4 and 6 are calculated in the same way as slave channel 2, whereas slave channels 5 and 7 are calculated as slave channel 3.

33.16.3.3 Block Diagram and General Timing Diagram

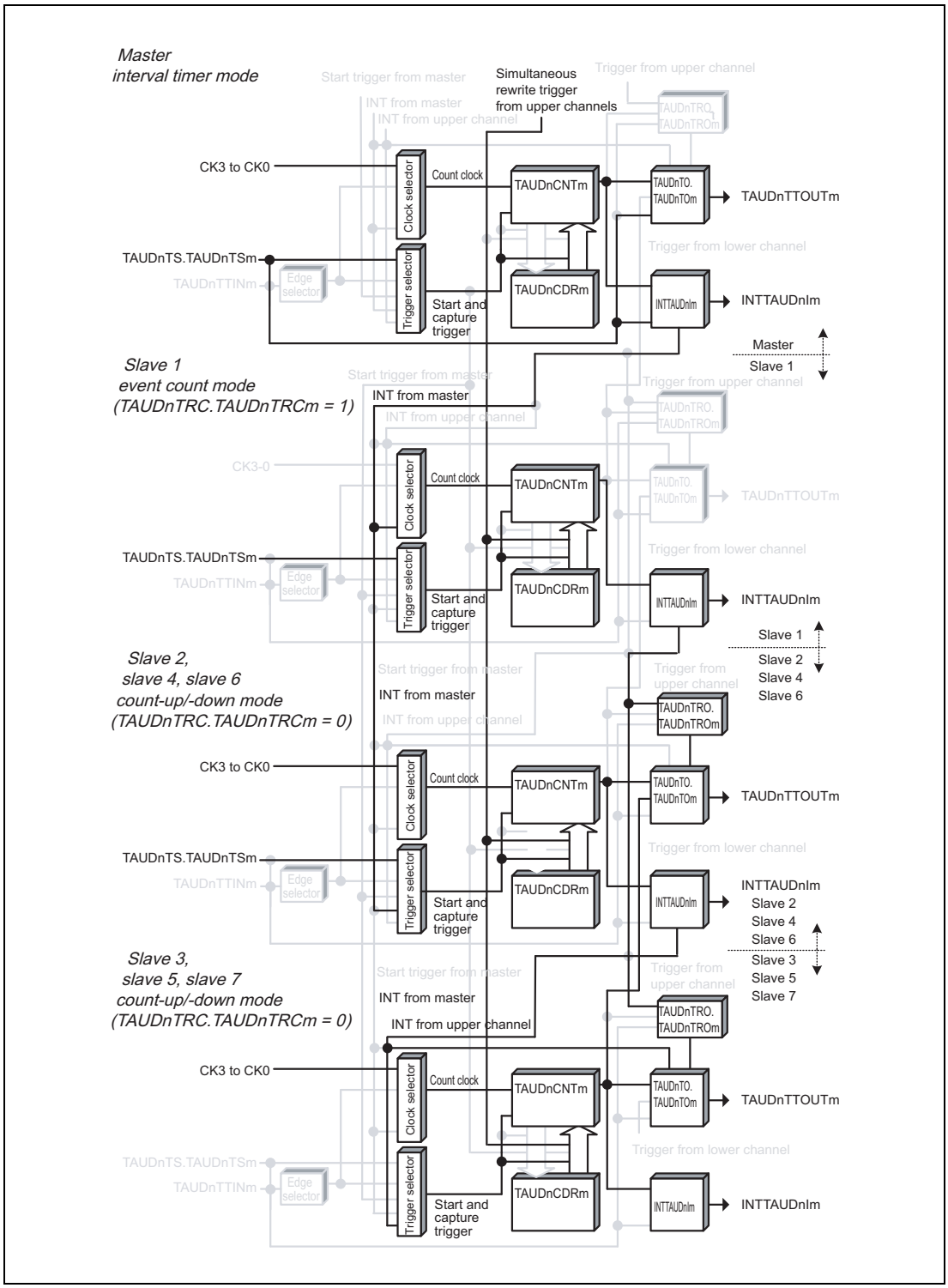


Figure 33.135 Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001_H

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

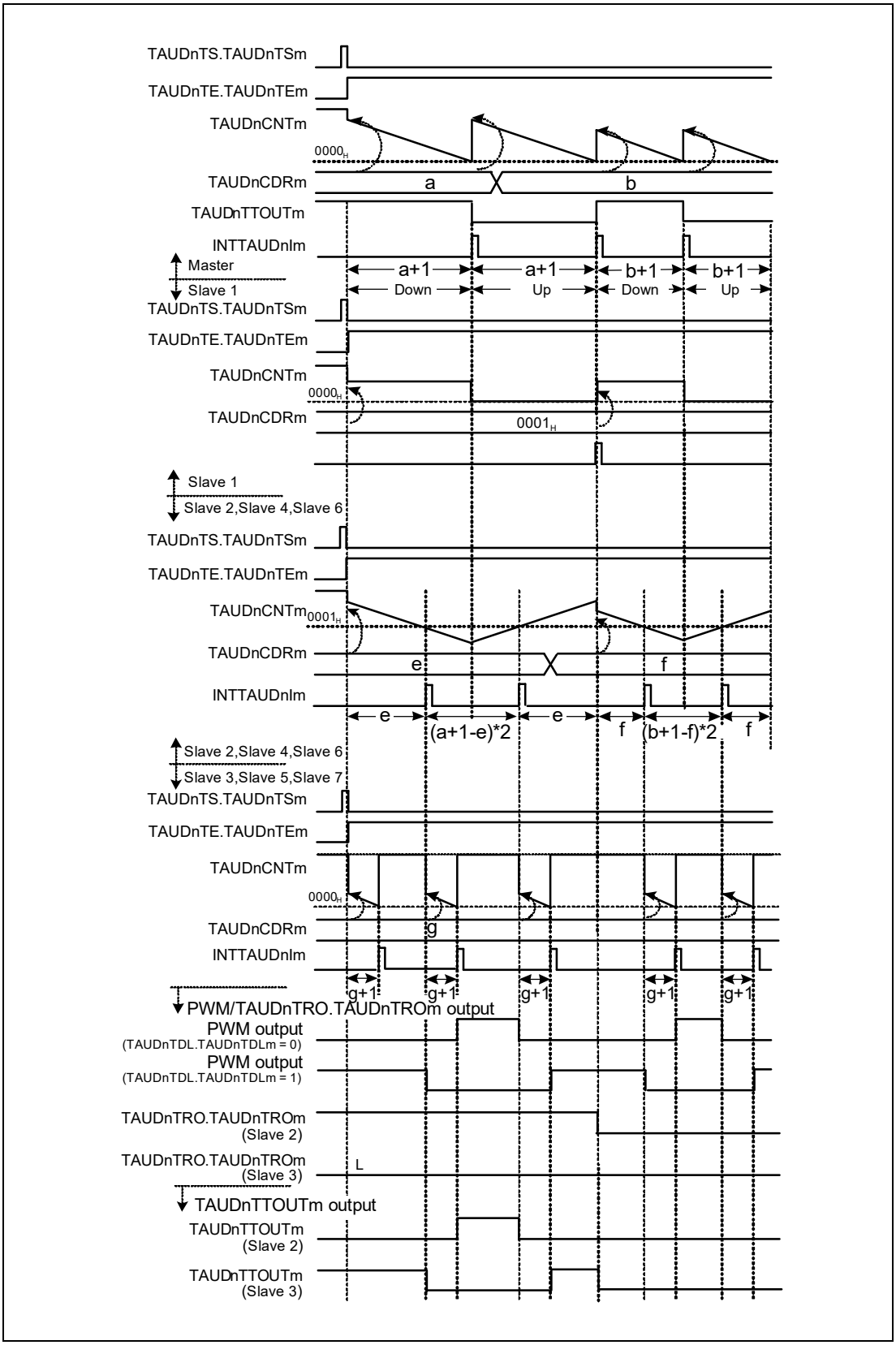


Figure 33.136 General Timing Diagram of Complementary Modulation Output Function

33.16.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the Master Channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.241 Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDnTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated to toggle TAUDnTTOUTm at the beginning of an operation.

(2) TAUDnCMURm for the Master Channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.242 Contents of the TAUDnCMURm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode for The Master Channel

Table 33.243 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous Rewrite for the Master Channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.244 Simultaneous Rewrite Settings for the Master Channel of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

33.16.3.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for Slave Channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.245 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.246 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel Output Mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous Rewrite for Slave Channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.247 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.3.6 Register Settings for Slave Channels 2, 4, and 6

(1) TAUDnCMORm for Slave Channels 2, 4, and 6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.248 Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for Slave Channels 2, 4, and 6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.249 Contents of the TAUDnCMURm Register for Slave Channel 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for Slave Channels 2, 4, and 6

Table 33.250 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous Rewrite for Slave Channels 2, 4, and 6

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.251 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.3.7 Register settings for slave channels 3, 5, and 7

(1) TAUDnCMORm for Slave Channels 3, 5, and 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 33.252 Contents of the TAUDnCMORm Register for Slave Channels 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for Slave Channels 3, 5, and 7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUDnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 33.253 Contents of the TAUDnCMURm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output Mode for Slave Channels 3, 5, and 7

Table 33.254 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous Rewrite for Slave Channels 3, 5, and 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 33.255 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

33.16.3.8 Operating Procedure for Complementary Modulation Output Function

Table 33.256 Operating Procedure for Complementary Modulation Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.3.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.3.5, Register Settings for Slave Channel 1**. Slave channels 2, 4, and 6: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.3.6, Register Settings for Slave Channels 2, 4, and 6**. Slave channels 3, 5, and 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 33.16.3.7, Register settings for slave channels 3, 5, and 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle using TAUDnCDRm of master channel, and an interrupt count of master channel to be ignored using TAUDnCDRm of slave channel 1. Also set a duty width in TAUDnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 33.256 Operating Procedure for Complementary Modulation Output Function (2/2)

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. TAUDnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for the next master channel interrupt. • TAUDnCNTm of slave channels 2, 4, and 6 reloads the TAUDnCDRm value, or performs counting in opposite direction. • At the same timing when the TAUDnCDRm value of slave channels 2, 4, and 6 is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDnTTOUTm output. • The counter of slave channel 1 waits for the next interrupt from the master channel when reaching 0000_H. When the interrupt is detected: – TAUDnCDRm value is reloaded into TAUDnCNTm and the counter waits for the next master channel interrupt. – INTTAUDnIm is generated. – TAUDnTRO.TAUDnTROm is changeable. • When the counter of slave channels 2, 4, and 6 reaches 0001_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched). – TAUDnCDRm value of slave channels 3, 5, and 7 is loaded into TAUDnCNTm to perform counting down. • When the counter of slave channels 3, 5, and 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched).
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDnTTOUTm stop and retain their current values.

Restart Operation

33.16.3.9 Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0).
- Slave channel 1: TAUDnCDRm = 0001_H
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

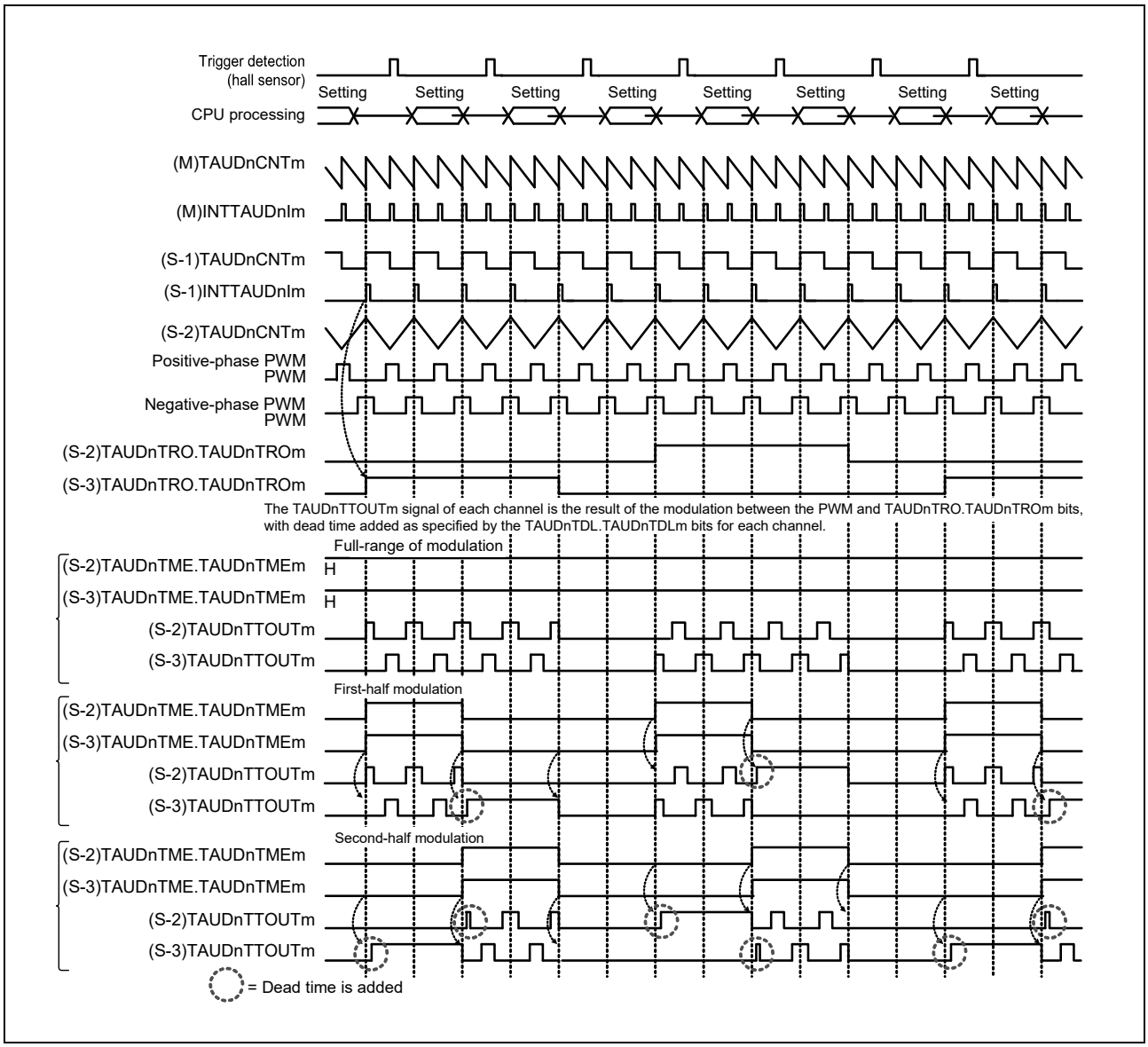


Figure 33.137 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDnTRO.TAUDnTROm bit value are output from slave channels 2 and 3.

TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm settings are reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

NOTE

Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

The “Setting” symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, TAUDnTRO.TAUDnTROm, and TAUDnTDL.TAUDnTDLm can be changed.

Section 34 Timer Array Unit J (TAUJ)

This section contains a generic description of the timer array unit J (TAUJ).

The first part of this section describes the RH850/U2A-EVA specific features, such as the number of units and the register base addresses.

The remainder of the section describes the functions and registers of the TAUJ.

34.1 Features TAUJ for RH850/U2A-EVA

34.1.1 Units and Channels

This microcontroller has the following number of TAUJ units.

Table 34.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)
Name	TAUJn									

TAUJn has the following number of channels of timers.

Table 34.2 TAUJn Unit Configurations and Channels

Unit Name (Channel Name) TAUJn	Number of Channels per Unit	RH850/ U2A-EVA (516 pins) (16 ch)	RH850/ U2A16 (516 pins) (16 ch)	RH850/ U2A16 (373 pins) (16 ch)	RH850/ U2A16 (292 pins) (16 ch)	RH850/ U2A8 (373 pins) (16 ch)	RH850/ U2A8 (292 pins) (16 ch)	RH850/ U2A6 (292 pins) (16 ch)	RH850/ U2A6 (176 pins) (16 ch)	RH850/ U2A6 (156 pins) (16 ch)	RH850/ U2A6 (144 pins) (16 ch)
TAUJ0	4	√	√	√	√	√	√	√	√	√	√
TAUJ1	4	√	√	√	√	√	√	√	√	√	√
TAUJ2	4	√	√	√	√	√	√	√	√	√	√
TAUJ3	4	√	√	√	√	√	√	√	√	√	√

Table 34.3 Index

Index	Description
n	Throughout this section, the individual TAUJ units are identified by the index “n” (n = 0 to 3); for example, TAUJnTOM is the TAUJn channel output mode register.
m	The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 3), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2) are denoted as CHm_even. The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

34.1.2 Register Base Addresses

TAUJn base addresses are listed in the following table.

TAUJn register addresses are given as offsets from the base addresses.

Table 34.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<TAUJ0_base>	FFBF 7000_H	P-Bus Group 5
<TAUJ1_base>	FFBF 7200_H	P-Bus Group 5
<TAUJ2_base>	FFE8 0000_H	P-Bus Group 2H
<TAUJ3_base>	FFE8 1000_H	P-Bus Group 2H

34.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

Table 34.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
TAUJ0	PCLK	CLK_HSB	Peripheral high speed clock
TAUJ1	PCLK	CLK_HSB	Peripheral high speed clock
TAUJ2	PCLK	CLKA_TAUJ	TAUJ2 ope. clock
TAUJ3	PCLK	CLKA_TAUJ	TAUJ3 ope. clock

34.1.4 Interrupt Requests and Error Notifications

TAUJn interrupt requests are listed in the following table.

Table 34.6 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
TAUJ0				
INTTAUJ0I0	Channel 0 interrupt	360	Group0-240	Group1-112
INTTAUJ0I1	Channel 1 interrupt	361	Group0-241	Group1-113
INTTAUJ0I2	Channel 2 interrupt	362	Group0-242	Group1-114
INTTAUJ0I3	Channel 3 interrupt	363	Group0-243	Group1-115
TAUJ1				
INTTAUJ1I0	Channel 0 interrupt	364	Group0-244	Group1-116
INTTAUJ1I1	Channel 1 interrupt	365	Group0-245	Group1-117
INTTAUJ1I2	Channel 2 interrupt	366	Group0-246	Group1-118
INTTAUJ1I3	Channel 3 interrupt	367	Group0-247	Group1-119
TAUJ2				
INTTAUJ2I0	Channel 0 interrupt	368	Group0-248	Group1-120
INTTAUJ2I1	Channel 1 interrupt	369	Group0-249	Group1-121
INTTAUJ2I2	Channel 2 interrupt	370	Group0-250	Group1-122
INTTAUJ2I3	Channel 3 interrupt	371	Group0-251	Group1-123
TAUJ3				
INTTAUJ3I0	Channel 0 interrupt	372	Group0-252	Group1-124
INTTAUJ3I1	Channel 1 interrupt	373	Group0-253	Group1-125
INTTAUJ3I2	Channel 2 interrupt	374	Group0-254	Group1-126
INTTAUJ3I3	Channel 3 interrupt	375	Group0-255	Group1-127

This module has no error notifications.

34.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

Table 34.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
TAUJ0	All registers	√	√	√	√	√	√	—
TAUJ1	All registers	√	√	√	√	√	√	—
TAUJ2	All registers	√	√	√	√	—	—	—
TAUJ3	All registers	√	√	√	√	—	—	—

34.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 34.8 External Input/Output Signals (RH850/U2A-EVA) (1/2)

Unit Signal Name	Description	Alternative Port Pin Signal Name
TAUJ0		
TAUJ0TTIN0	Input for channel 0	TAUJ0I0
TAUJ0TTIN1	Input for channel 1	TAUJ0I1
TAUJ0TTIN2	Input for channel 2	TAUJ0I2
TAUJ0TTIN3	Input for channel 3	TAUJ0I3
TAUJ0TTOUT0	Output for channel 0	TAUJ0O0
TAUJ0TTOUT1	Output for channel 1	TAUJ0O1
TAUJ0TTOUT2	Output for channel 2	TAUJ0O2
TAUJ0TTOUT3	Output for channel 3	TAUJ0O3
TAUJ1		
TAUJ1TTIN0	Input for channel 0	TAUJ1I0
TAUJ1TTIN1	Input for channel 1	TAUJ1I1
TAUJ1TTIN2	Input for channel 2	TAUJ1I2
TAUJ1TTIN3	Input for channel 3	TAUJ1I3
TAUJ1TTOUT0	Output for channel 0	TAUJ1O0
TAUJ1TTOUT1	Output for channel 1	TAUJ1O1
TAUJ1TTOUT2	Output for channel 2	TAUJ1O2
TAUJ1TTOUT3	Output for channel 3	TAUJ1O3
TAUJ2		
TAUJ2TTIN0	Input for channel 0	TAUJ2I0*2
TAUJ2TTIN1	Input for channel 1	TAUJ2I1*2
TAUJ2TTIN2	Input for channel 2	TAUJ2I2*2 or TAUJ1TTOUT0*1
TAUJ2TTIN3	Input for channel 3	TAUJ2I3*2 or TAUJ1TTOUT0*1
TAUJ2TTOUT0	Output for channel 0	TAUJ2O0
TAUJ2TTOUT1	Output for channel 1	TAUJ2O1
TAUJ2TTOUT2	Output for channel 2	TAUJ2O2
TAUJ2TTOUT3	Output for channel 3	TAUJ2O3

Table 34.8 External Input/Output Signals (RH850/U2A-EVA) (2/2)

Unit Signal Name	Description	Alternative Port Pin Signal Name
TAUJ3		
TAUJ3TTIN0	Input for channel 0	TAUJ3I0
TAUJ3TTIN1	Input for channel 1	TAUJ3I1
TAUJ3TTIN2	Input for channel 2	TAUJ3I2 or RTCA0OUT*1 or TAUJ0TTOUT0*1
TAUJ3TTIN3	Input for channel 3	TAUJ3I3 or RTCA0OUT*1 or TAUJ0TTOUT0*1
TAUJ3TTOUT0	Output for channel 0	TAUJ3O0
TAUJ3TTOUT1	Output for channel 1	TAUJ3O1
TAUJ3TTOUT2	Output for channel 2	TAUJ3O2
TAUJ3TTOUT3	Output for channel 3	TAUJ3O3

Note 1. For details, see **Section 41.2.3.12, Timer Input Select Function**.

Note 2. External input from pins (TAUJ2I0-3) are not supported in DeepSTOP mode.

34.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

Table 34.9 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
TAUJnTSSTm	Simultaneous channel start trigger input	PIC

34.2 Overview

34.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The timer array unit J is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

34.2.2 Terms

In this section, the following terms are used.

Independent channel operation function/synchronous operation channel operation function

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented by using a combination of channels.

- The independent channel operation function can be used any channel independently of all other channels.
- The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Upper/lower channel

Based on the channel number m, a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

34.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 34.10 Functional List of TAUJ Operations

Operation Function		Example
Independent Channel Operation Functions		**Section 34.4.9**
	Interval timer function	**Section 34.4.9.1**
	TAUJnTTINm input interval timer function	**Section 34.4.9.2**
	TAUJnTTINm input pulse interval measurement function	**Section 34.4.9.3**
	TAUJnTTINm input signal width measurement function	**Section 34.4.9.4**
	TAUJnTTINm input position detection function	**Section 34.4.9.5**
	TAUJnTTINm input period count detection function	**Section 34.4.9.6**
	Overflow interrupt output function (during TAUJnTTINm width measurement)	**Section 34.4.9.7**
	Overflow interrupt output function (during TAUJnTTINm input period count detection)	**Section 34.4.9.8**
Synchronous Channel Operation Functions		**Section 34.4.10**
	PWM output function	**Section 34.4.10.1**

34.2.4 TAUJ I/O and Interrupt Request Signals

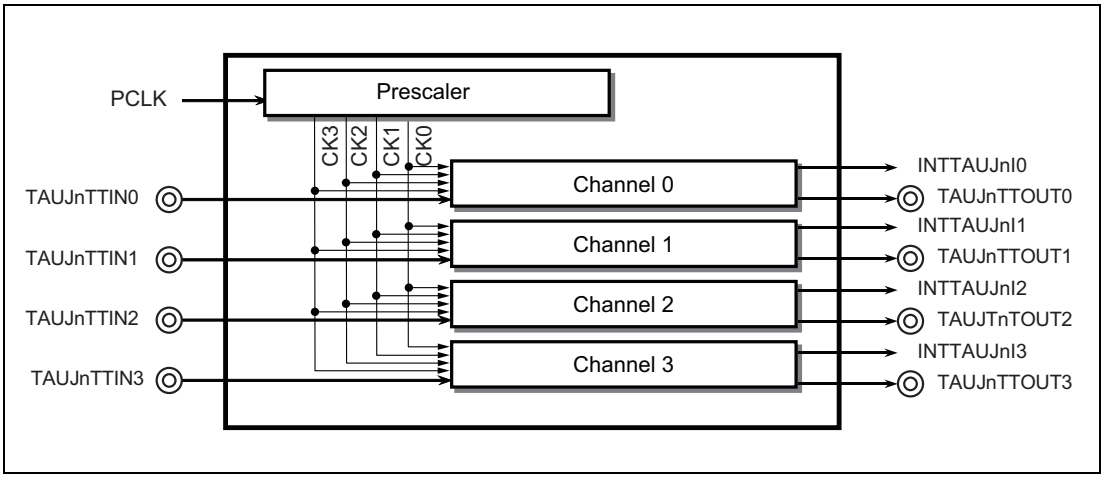


Figure 34.1 TAUJ I/O and Interrupt Request Signals

34.2.5 Block Diagram

The following figure shows the main components of the TAUJ:

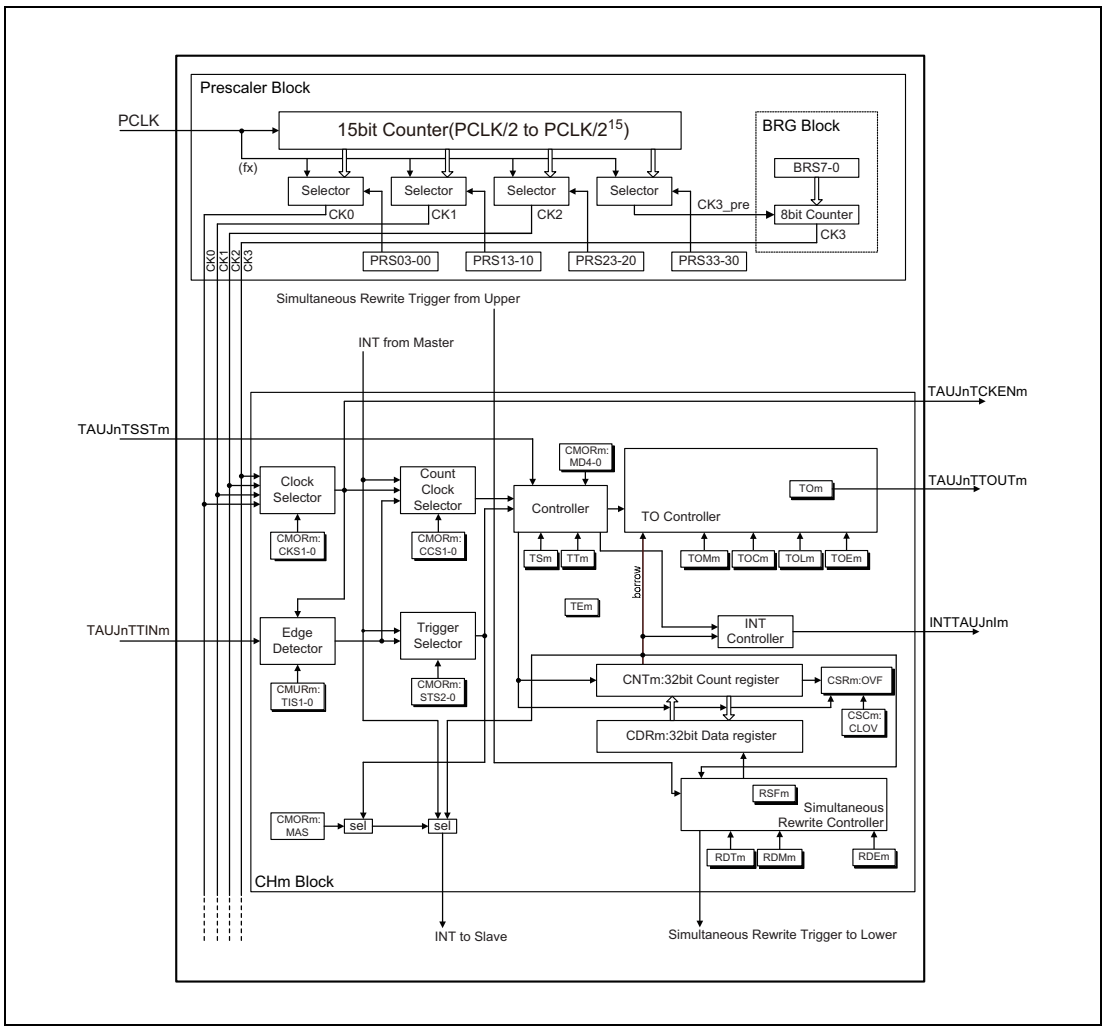


Figure 34.2 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

34.2.6 Description of Block Diagram

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived by dividing PCLK in the prescaler division factor of 2^0 to 2^{15}. The fourth count clock, CK3, is derived by dividing PCLK by a division factor that is not a power of 2 by using the baud rate generator.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of CK0 to CK3 clocks (selected by the clock selector)

Controller

The controller controls the main operations of the counter.

- Operating mode (selected with the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
- Valid edge of the TAUJnTTINm input signal
- INTTAUJnIm from the master channel

Simultaneous rewrite controller

Simultaneous rewrite control is enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

34.3 Registers

34.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see **Section 34.1.2, Register Base Addresses**.

Table 34.11 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
TAUJn prescaler registers						
TAUJn	TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90_H	16	*1	—
	TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94_H	8	*1	—
TAUJn control registers						
TAUJn	TAUJn channel data register m	TAUJnCDRm	<TAUJn_base> + m × 4_H	32	*1	—
	TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base> + 10_H + m × 4_H	32	*1	—
	TAUJn channel mode OS register m	TAUJnCMORm	<TAUJn_base> + 80_H + m × 4_H	16	*1	—
	TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base> + 20_H + m × 4_H	8	*1	—
	TAUJn channel status register m	TAUJnCSRm	<TAUJn_base> + 30_H + m × 4_H	8	*1	—
	TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base> + 40_H + m × 4_H	8	*1	—
	TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54_H	8	*1	—
	TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50_H	8	*1	—
	TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58_H	8	*1	—
TAUJn output registers						
TAUJn	TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60_H	8	*1	—
	TAUJn channel output register	TAUJnTO	<TAUJn_base> + $5C_H$	8	*1	—
	TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98_H	8	*1	—
	TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + $9C_H$	8	*1	—
	TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64_H	8	*1	—
TAUJn reload data registers						
TAUJn	TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + $A0_H$	8	*1	—
	TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + $A4_H$	8	*1	—
	TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68_H	8	*1	—
	TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + $6C_H$	8	*1	—

Note 1. n = 0: PBG51#4
n = 1: PBG51#5
n = 2: PBG20#5
n = 3: PBG20#6

34.3.2 Details of TAUJn Prescaler Registers

34.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUJn_base> + 90_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.12 TAUJnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUJnPRS3 [3:0]	Specifies a CK3_PRE clock. The CK3_PRE clock is an input clock of the BRG unit which supplies CK3 operation clocks to all channels.

TAUJnPRS3[3:0]	CK3_PRE Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 34.12 TAUJnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUJnPRS2 [3:0]	Specifies a CK2 clock. **TAUJnPRS2[3:0]** / **CK2 Clock** 0000_B PCLK/2^0 0001_B PCLK/2^1 0010_B PCLK/2^2 0011_B PCLK/2^3 0100_B PCLK/2^4 0101_B PCLK/2^5 0110_B PCLK/2^6 0111_B PCLK/2^7 1000_B PCLK/2^8 1001_B PCLK/2^9 1010_B PCLK/2^{10} 1011_B PCLK/2^{11} 1100_B PCLK/2^{12} 1101_B PCLK/2^{13} 1110_B PCLK/2^{14} 1111_B PCLK/2^{15} The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).
7 to 4	TAUJnPRS1 [3:0]	Specifies a CK1 clock. **TAUJnPRS1[3:0]** / **CK1 Clock** 0000_B PCLK/2^0 0001_B PCLK/2^1 0010_B PCLK/2^2 0011_B PCLK/2^3 0100_B PCLK/2^4 0101_B PCLK/2^5 0110_B PCLK/2^6 0111_B PCLK/2^7 1000_B PCLK/2^8 1001_B PCLK/2^9 1010_B PCLK/2^{10} 1011_B PCLK/2^{11} 1100_B PCLK/2^{12} 1101_B PCLK/2^{13} 1110_B PCLK/2^{14} 1111_B PCLK/2^{15} The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 34.12 TAUJnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUJnPRS0 [3:0]	Specifies a CK0 clock.

TAUJnPRS0[3:0]	CK0 Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 34.1.3, Clock Supply**.

34.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 94_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUJnBRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.13 TAUJnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUJnBRS [7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUJnBRS[7:0]	CK3 Clock
$0000\ 0000_B$	CK3_PRE/1
$0000\ 0001_B$	CK3_PRE/2
$0000\ 0010_B$	CK3_PRE/3
$0000\ 0011_B$	CK3_PRE/4
...	...
$1111\ 1110_B$	CK3_PRE/255
$1111\ 1111_B$	CK3_PRE/256

34.3.3 Details of TAUJn Control Registers

34.3.3.1 TAUJnCDRm — TAUJn Channel Data Register m

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

Access: This register can be read or written in 32-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUJn_base> + m × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.14 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

34.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register m

This is a channel m counter register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.15 TAUJnCNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT [31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/ TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 34.16 TAUJnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUJnCNTm		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF FFFF$_H$	Stop value	—
Capture mode	Count up	0000 0000$_H$	Stop value	—
One-count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000 0000$_H$	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count up	0000 0000$_H$	Stop value	—
Gate count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000 0000$_H$	Stop value	Stop value

Note 1. The value set for the TAUJnCNTm when operating mode is changed after a reset is deasserted.

34.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register m

This register controls channel m operation.

Access: This register can be read or written in 16-bit units.
Writable only when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 80_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.17 TAUJnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Selects an operation clock, which is used with the TAUJnTTINm input edge detection circuit. Setting of TAUJnCMORm.TAUJnCCS[1:0] bits also allows the operation clock to serve as the TAUJnCNTm count clock.
13, 12	TAUJnCCS[1:0]	Selects a count clock for TAUJnCNTm counter.
11	TAUJnMAS	Specifies whether the channel is a master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm-odd) are fixed to 0.

TAUJn CKS1	TAUJn CKS0	Selection of Operating Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

TAUJn CCS1	TAUJn CCS0	Selection of Count Clock
0	0	Operation clock specified by TAUJnCMORm.TAUJnCKS[1:0].
0	1	Setting prohibited
1	0	
1	1	

Table 34.17 TAUJnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUJnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUJnSTS2</th><th>TAUJnSTS1</th><th>TAUJnSTS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJnTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJnTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUJnIm of master channel is used as a start trigger.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUJnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m.
These bits are only valid if channel m is for capture function (capture mode and capture & one-count mode).
<table>
<tr><th>TAUJnCOS1</th><th>TAUJnCOS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJnTTINm input is detected.</td><td>Updated (cleared or set) when valid edge of TAUJnTTINm input is detected:
• Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.
• Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJnTTINm input is detected and when a counter overflow occurs.
• Detection of valid edge of TAUJnTTINm input: The counter value is written into TAUJnCDRm.
• Occurrence of overflow: FFFF FFFF$_H$ is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJnTTINm is ignored.</td><td>No setting</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 34.17 TAUJnCMORm Register Contents (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUJnMD[4:0]</td><td>Specifies an operating mode.
<table>
<tr><th>TAUJn MD4</th><th>TAUJn MD3</th><th>TAUJn MD2</th><th>TAUJn MD1</th><th>TAUJn MD0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr>
</table>
<table>
<tr><th>Mode</th><th>Role of TAUJnMD0 Bit</th></tr>
<tr><td>Interval timer mode
Capture mode
Count capture mode</td><td>Specifies whether INTTAUJnIm is generated at the beginning of count operation (when a start trigger is entered) or not.
0: INTTAUJnIm is not generated.
1: INTTAUJnIm is generated.</td></tr>
<tr><td>One-count mode</td><td>Enables/disables start trigger detection during counting.
0: Disables detection.
1: Enables detection.
CAUTION
In one-count mode, INTTAUJnIm signal is not output at the beginning of count operation.</td></tr>
<tr><td>Capture and one-count mode
Gate count mode
Capture and gate count mode</td><td>This bit should be set to 0.
CAUTION
INTTAUJnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td></tr>
</table>
</td></tr>
</table>

34.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register m

This register specifies a type of valid edge detection used for TAUJnTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 20_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.18 TAUJnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	Specifies a valid edge of TAUJnTTINm input signal. **TAUJn TIS1 / TAUJn TIS0 / Functional Description** 0 / 0 / Falling edge 0 / 1 / Rising edge 1 / 0 / Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge 1 / 1 / Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge Edge detection of TAUJnTTINm input signal is based on the operation clock selected by TAUJnCMORm.TAUJnCKS[1:0].

34.3.3.5 TAUJnCSRm — TAUJn Channel Status Register m

This register indicates the overflow status of channel m.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnOVF
Value after reset	0	0	0	0	0	0	—	0
R/W	R	R	R	R	R	R	R	R

Table 34.19 TAUJnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	Reserved	When read, an undefined value is returned.
0	TAUJnOVF	Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

34.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register m

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units.
The read value is always 00_H.

Address: <TAUJn_base> + 40_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 34.20 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

34.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: This register is a write-only register that can be written in 8-bit units.
The read value is always 00_H.

Address: <TAUJn_base> + 54_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 34.21 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: No function 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1 Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1. Whether the counter is started or not depends on the selected operating mode.

34.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.22 TAUJnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnTEm	Indicates whether channel m's counter operation is enabled. 0: Counter operation is disabled 1: Counter operation is enabled This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is reset to 0 when TAUJnTT.TAUJnTTm is set to 1.

34.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 8-bit units.
The read value is always 00_H.

Address: <TAUJn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 34.23 TAUJnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTTm	Stops channel m's counter operation. 0: No function 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJnTTOUTm retain the values provided before the counter is stopped.

34.3.4 Details of TAUJn Simultaneous Rewrite Register

34.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOLm for the PWM output function.

Access: This register can be read or written in 8-bit units.
It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.24 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

34.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read or written in 8-bit units.
It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.25 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDMm	Specifies when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: No function These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

34.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 68_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 34.26 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnRDTm	Triggers the simultaneous rewrite enabling state: 0: No function 1: Simultaneous rewrite enabling state is triggered. The simultaneous rewrite enabling flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

34.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + $6C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 34.27 TAUJnRSF Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to generation of the simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUJnRDTm = 1).

34.3.5 Details of TAUJn Output Registers

34.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 60_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.28 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOEm	Enables/disables Independent Channel Output Function: 0: Disables independent timer output function (controlled by software) 1: Enables independent timer output function

34.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJnTTOUTm.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + $5C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.29 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOm	Specifies and reads the level of TAUJnTTOUTm: 0: Low 1: High Only TAUJnTOm bits for which independent channel output function is disabled (TAUJnTOEm = 0) can be written.

34.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 8-bit units.
It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 98_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.30 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on the settings of channel output control (TAUJnTOE.TAUJnTOEm) bits.

34.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

Access: This register can be read or written in 8-bit units.
It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + $9C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.31 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOCm	Specifies the output mode: 0: Operation mode 1 (= Toggle mode) 1: No function This bit must be set to 0 for all output modes except independent channel output mode controlled by software. The output mode also depends on TAUJnTOM.TAUJnTOMm, as shown in the following table.

TAUJn TOMm	TAUJn TOCm	Functional Description
0	0	Toggle mode: Toggling proceeds when INTTAUJnIm is generated.
0	1	No function
1	0	Synchronous channel operation mode 1: Set when INT generates on the master channel and reset when INT generates on the slave channel.
1	1	No function

34.3.5.5 TAUJnTOL — TAUJn Channel Output Active Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.32 TAUJnTOL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOm): 0: Positive logic (active high) 1: Negative logic (active low) These bits apply in all channel output modes except independent channel output mode controlled by software and independent channel output mode 1.

34.4 Operation

34.4.1 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After a reset is released, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJnTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   - Set the operation mode
   - Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSm bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 34.4.9, Independent Channel Operation Functions** and **Section 34.4.10, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

34.4.2 Concepts of Synchronous Channel Operation Function

The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 34.4.2.1, Rules of Synchronous Channel Operation Function**.

The synchronous channel operation function are detailed in the following section.

- **Section 34.4.10, Synchronous Channel Operation Functions**

34.4.2.1 Rules of Synchronous Channel Operation Function

Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel other than CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
  Example: If CH0 and CH2 are master channels, CH1 can be set as slave channel for CH0, but CH3 cannot.

Operation clock

- The same operation clock should be set for the master channel and the synchronized slave channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 34.3**.

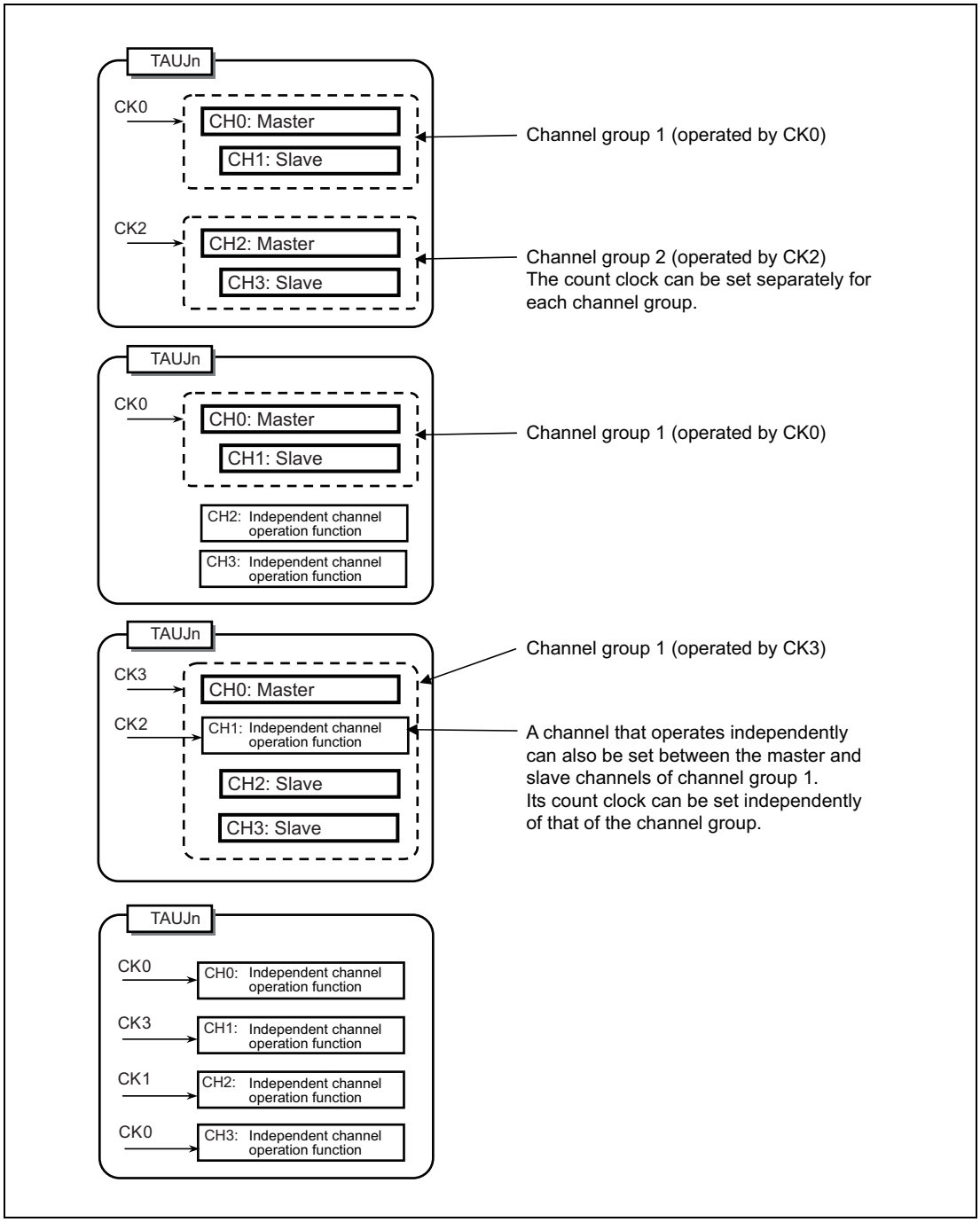


Figure 34.3 Grouping of Channels and Assignment of Operation Clocks

34.4.2.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

(1) Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEm bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

(2) Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

34.4.3 Simultaneous Rewrite

34.4.3.1 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

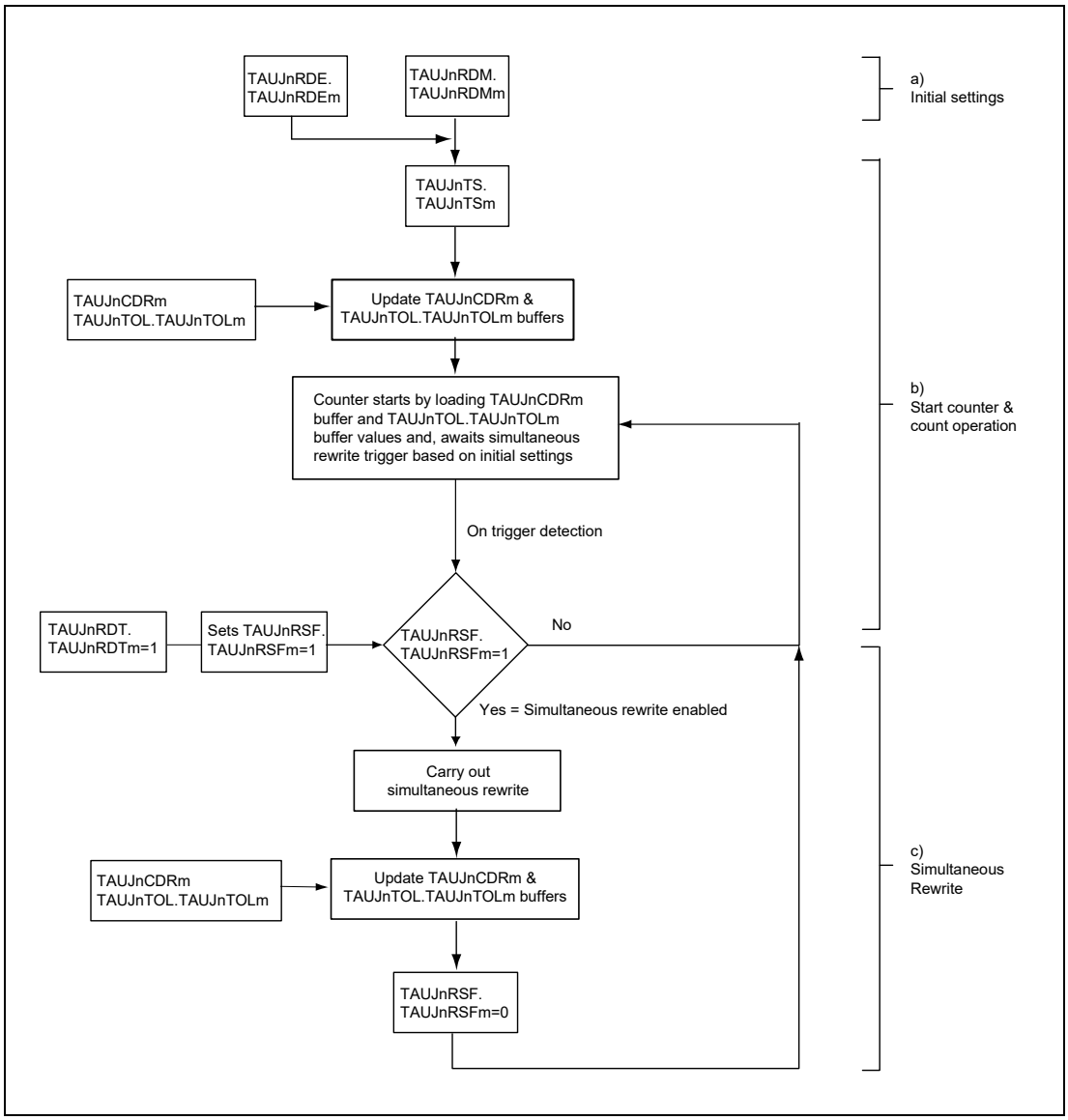


Figure 34.4 General Procedure for Simultaneous Rewrite

(1) Initial Settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1
- To select simultaneous rewrite when the master channel starts counting, set TAUJnRDM.TAUJnRDMm.

(2) Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. The values of TAUJnTOL.TAUJnTOLm and the values in the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

(3) Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is complete, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

34.4.3.2 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJnTTOUTm outputs an invalid waveform.

34.4.3.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

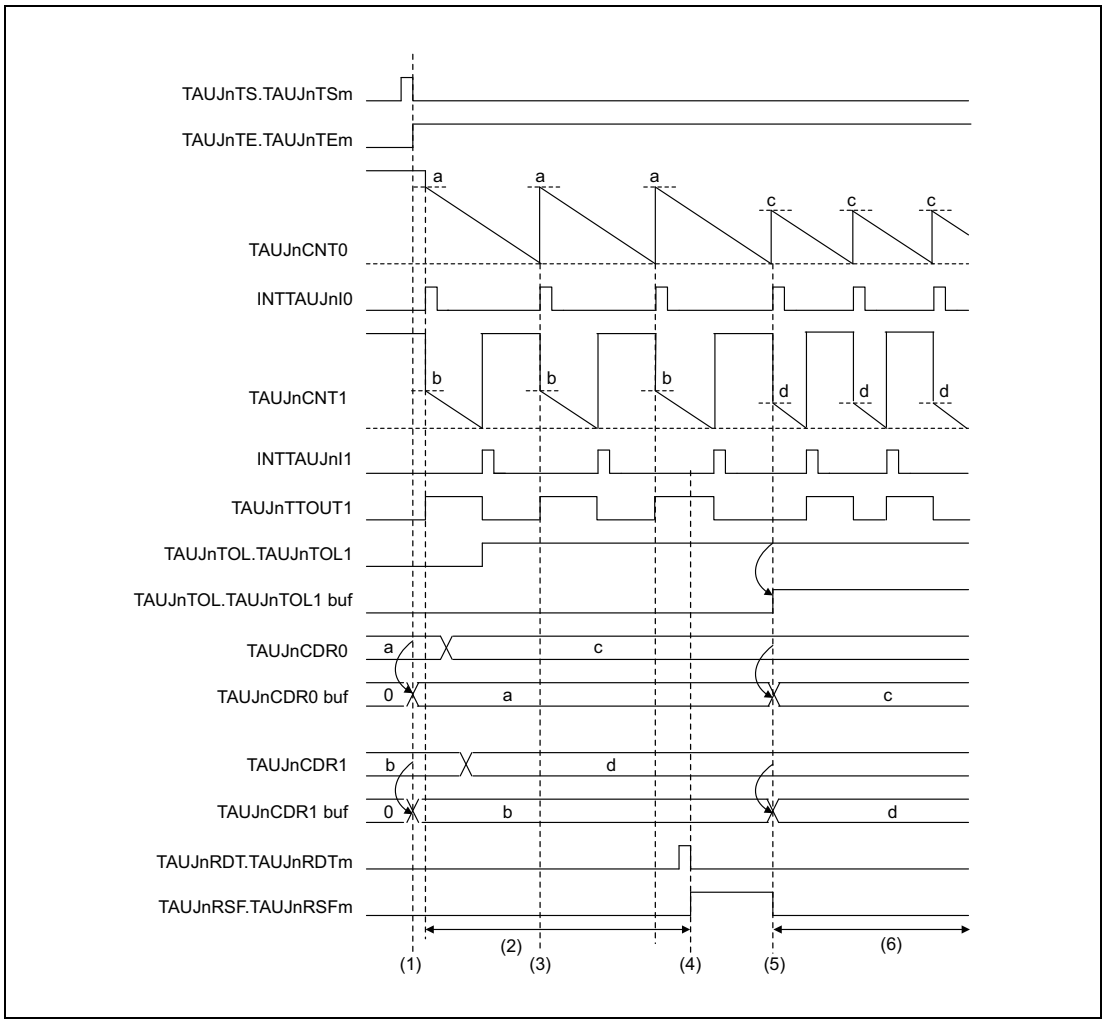


Figure 34.5 Simultaneous Rewrite with PWM Output Function

Setting:

CH0 is a master channel of PWM output function, and CH1 is a slave channel of PWM output function. Simultaneous rewrite is applied when the master channel starts counting.

Description:

(1) When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.

(2) The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).

(4) The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

34.4.4 Channel Output Modes

The output of the TAUJnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is sent to the output pin (TAUJnTTOUTm).
- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)
  When controlled by TAUJ signals, the output level of TAUJnTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJnTTOUTm.
  - Independently (TAUJnTOM.TAUJnTOMm = 0)
    In case of independent operation, the output of the TAUJnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).
  - Synchronously (TAUJnTOM.TAUJnTOMm = 1)
    In case of synchronous operation, the output of the TAUJnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 34.33, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 34.4.4.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 34.4.4.3, Channel Output Modes Controlled Synchronously by TAUJn Signals**

Batch operation of TAUJnTOm bit

Whether a set value is reflected to the TAUJnTO.TAUJnTOm bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTO.TAUJnTOm setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted.

No TAUJnTO.TAUJnTOm setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

NOTE

TAUJnTO.TAUJnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an undefined TAUJnTTOUTm signal output.

See **Section 34.4.3, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 34.33**.

Table 34.33 Channel Output Modes

Channel Output Mode	TAUJnTOE. TAUJnTOEm	TAUJnTOM. TAUJnTOMm
By software		
Independent channel output mode controlled by software	0	x
By TAUJ signals, independently		
Independent channel output mode 1	1	0
By TAUJ signals, synchronously		
Synchronous channel output mode 1	1	1

- All combinations not listed is this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUJnTE.TAUJnTEm = 1):

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

34.4.4.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

(1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJnTTOUTm output.

(2) Set channel output mode according to Table 34.33, Channel Output Modes, and the output logic using the TAUJnTOL.TAUJnTOLm bit.

(3) Start the counter (TAUJnTS.TAUJnTSm = 1).

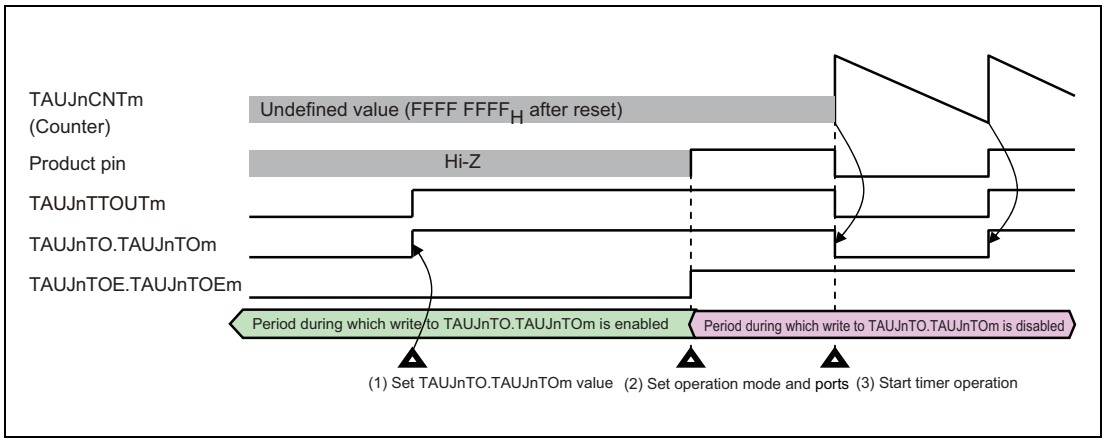


Figure 34.6 General Procedure for Specifying a TAUJnTTOUTm Channel Output Mode

34.4.4.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in **Table 34.33, Channel Output Modes**.

(1) Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUJnTTOUTm toggles when INTTAUJnIm is detected.

The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 34.33, Channel Output Modes**.

34.4.4.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in **Table 34.33, Channel Output Modes**.

(1) Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 34.33, Channel Output Modes**.

34.4.5 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

34.4.5.1 Interval Timer Mode, Capture Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.

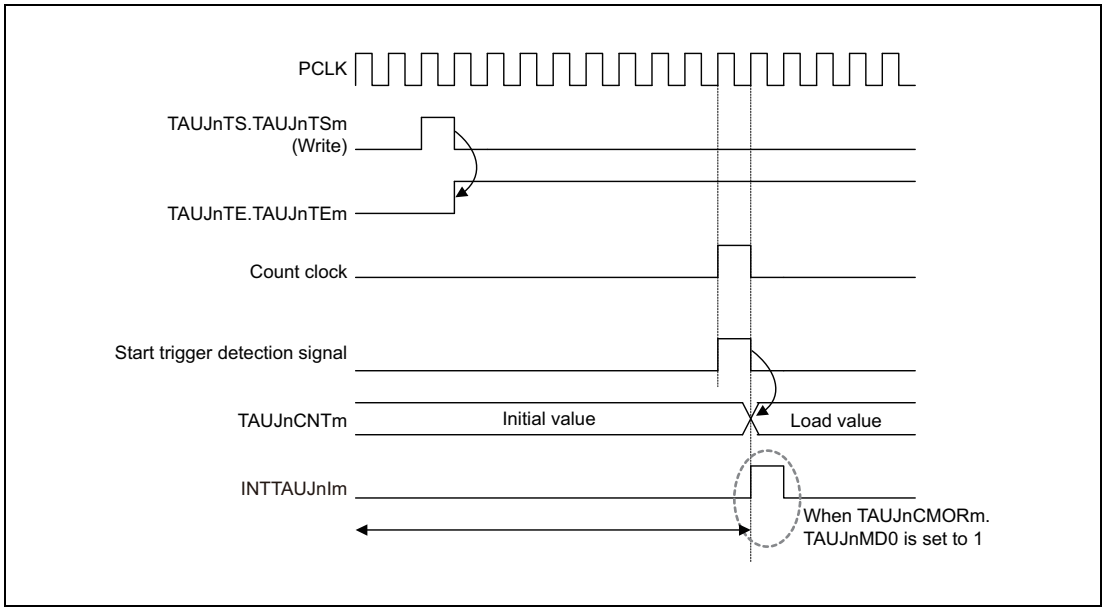


Figure 34.7 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

34.4.5.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter operation start timing is triggered only upon detection of a valid edge of TAUJnTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

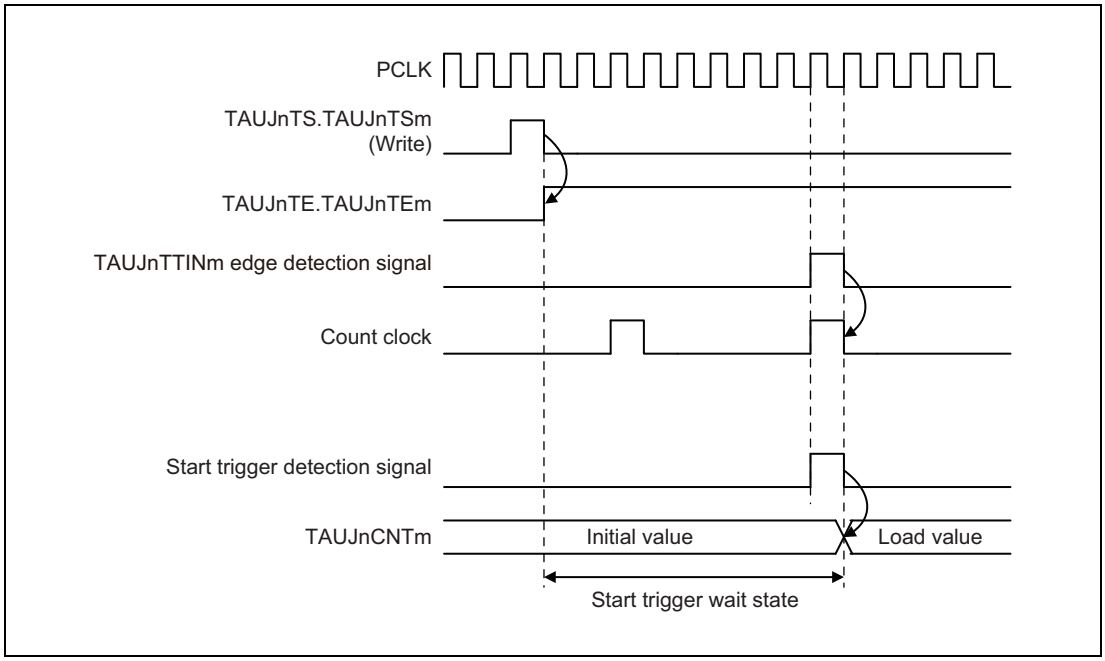


Figure 34.8 Count Start Timing in Other Operating Modes

34.4.6 TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The generation of INTTAUJnIm when the TAUJnCMORm.TAUJnMD0 bit starts counting and the effect to TAUJnTTOUTm depend on the selected function. For details, refer to the description of TAUJnCMORm.TAUJnMD0 of each function.

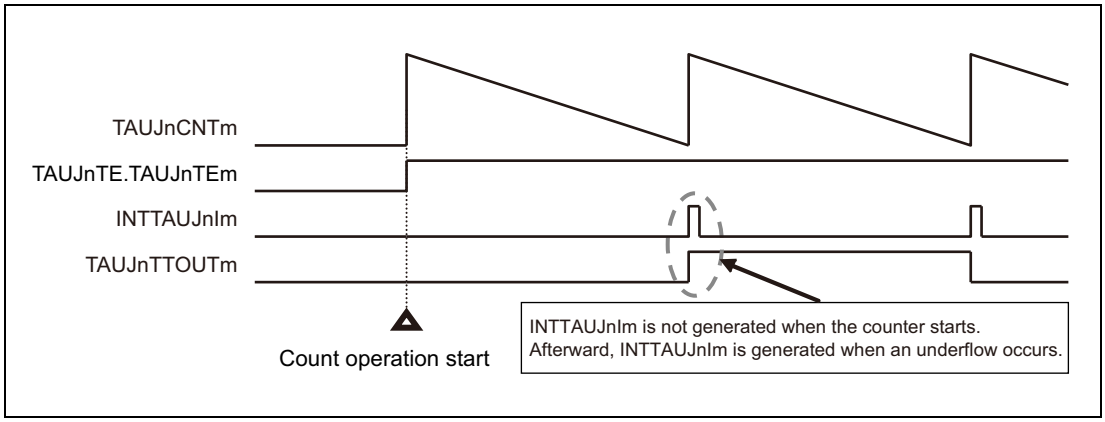


Figure 34.9 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 0)

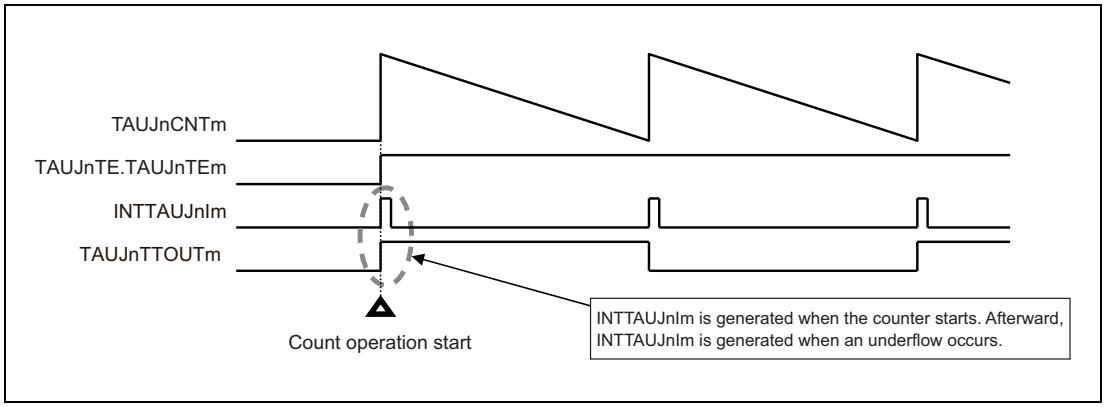


Figure 34.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 1)

34.4.7 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches FFFF FFFF$_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operations in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000 0000$_H$ at the same time as the first channel overflows (TAUJnCNTm = FFFF FFFF$_H$).
- Set TAUJnCDRm of the second channel to FFFF FFFF$_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUJnTTINm input.
- The trigger detection settings (TAUJnCMORm.TAUJnSTS[2:0] and TAUJnCMURm.TAUJnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000 0000$_H$ at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF FFFF$_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

34.4.7.1 Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function

When the capture trigger is input simultaneously to TAUJnTTINm of both channels, INTTAUJnIm of the interval timer function can detect the overflow when TAUJnCNTm of the TAUJnTTINm input position detection function exceeds FFFF FFFF$_H$.

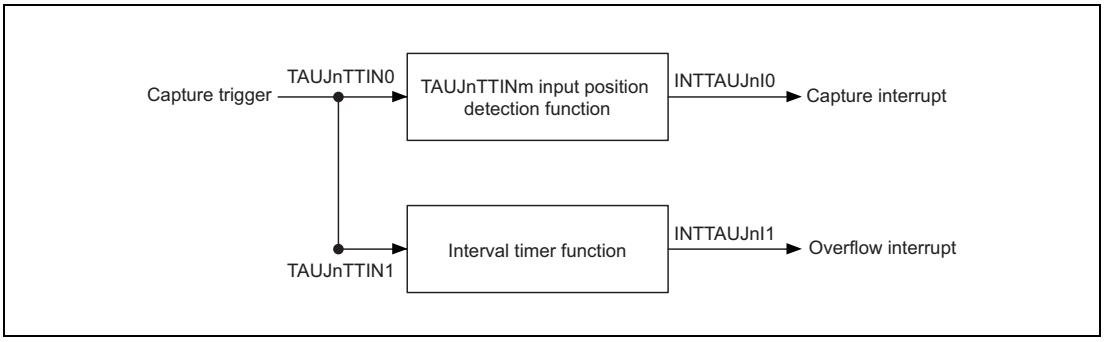


Figure 34.11 Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

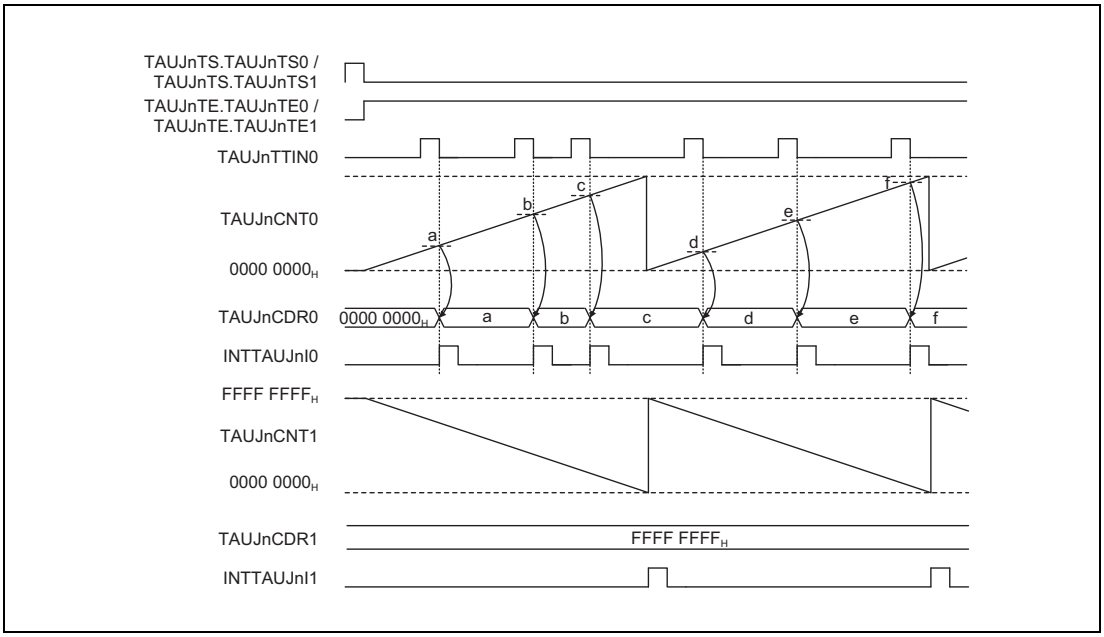


Figure 34.12 Interrupt Generation via Combination of the TAUJnTTINm Input Position Detection Function and the Interval Timer Function

34.4.8 TAUJnTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

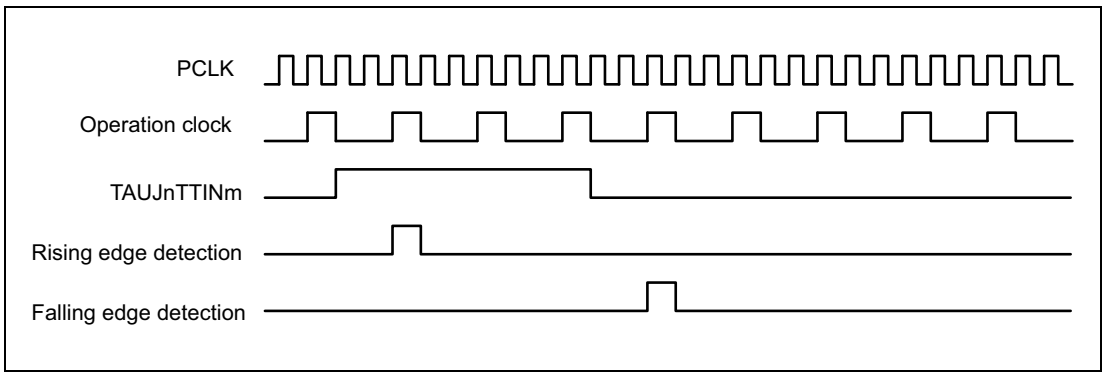


Figure 34.13 Basic Edge Detection Timing

Figure 34.13 shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

- Delay time by a noise filter + delay time for edge detection (maximum one operation clock cycle)

34.4.9 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see **Section 34.2, Overview**.

34.4.9.1 Interval Timer Function

(1) Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a square wave.

Functional Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJnTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJnTTOUTm does not toggle. This results in an reverted TAUJnTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

(2) Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJnTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

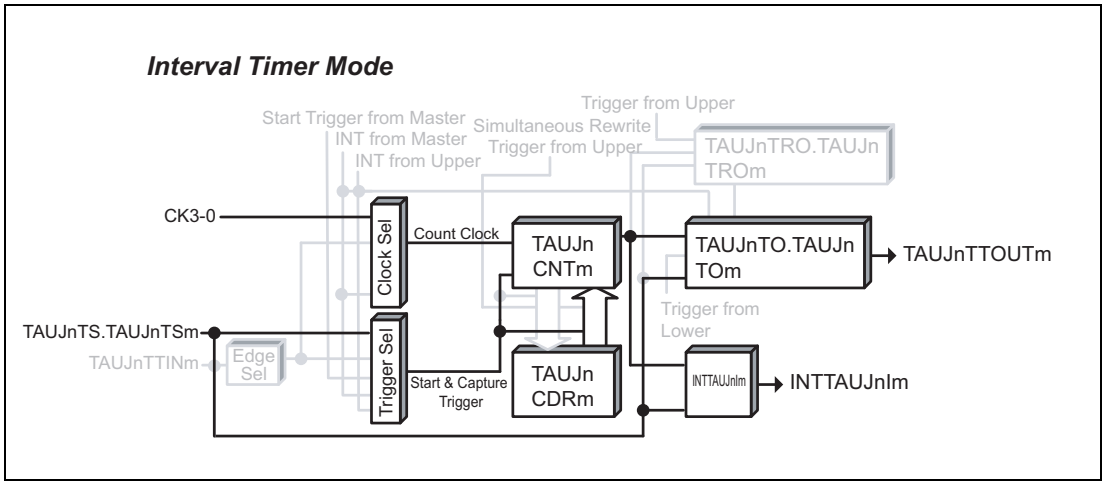


Figure 34.14 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1)

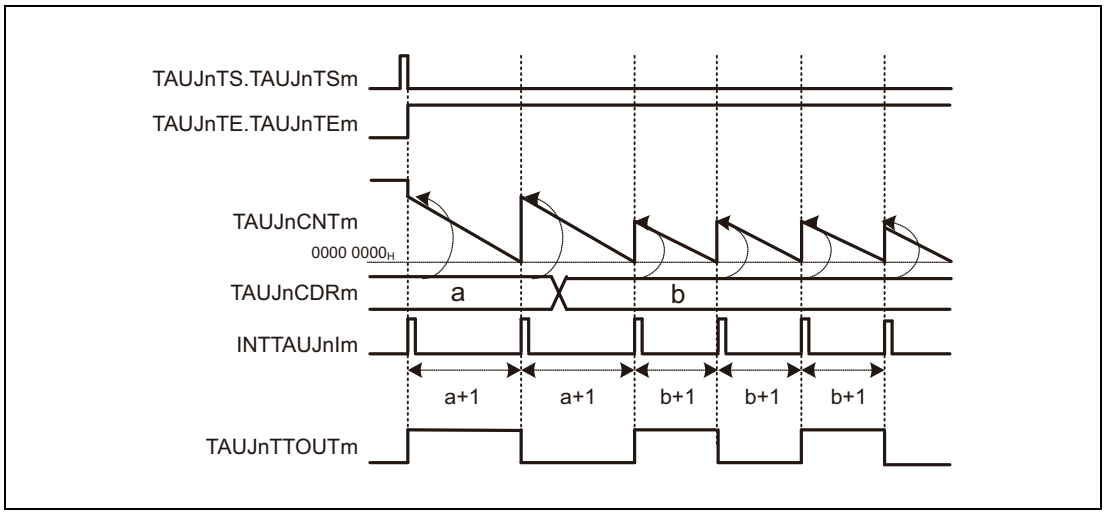


Figure 34.15 General Timing Diagram for Interval Timer Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.34 Contents of the TAUJnCMORm register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJnTTOUTm does not toggle when operation starts or restarts 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm when operation starts or restarts

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.35 Contents of the TAUJnCMURm register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Table 34.36 Control Bit Settings in Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 34.4.4, Channel Output Modes**.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the interval timer function. Therefore, these registers must be set to 0.

Table 34.37 Simultaneous Rewrite Settings for Interval Timer Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for Interval Timer Function

Table 34.38 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 34.34, Contents of the TAUJnCMORm register for Interval Timer Function** and **Table 34.35, Contents of the TAUJnCMURm register for Interval Timer Function**. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in **Table 34.36, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJnTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart operation

(6) Specific Timing Diagrams

(a) TAUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

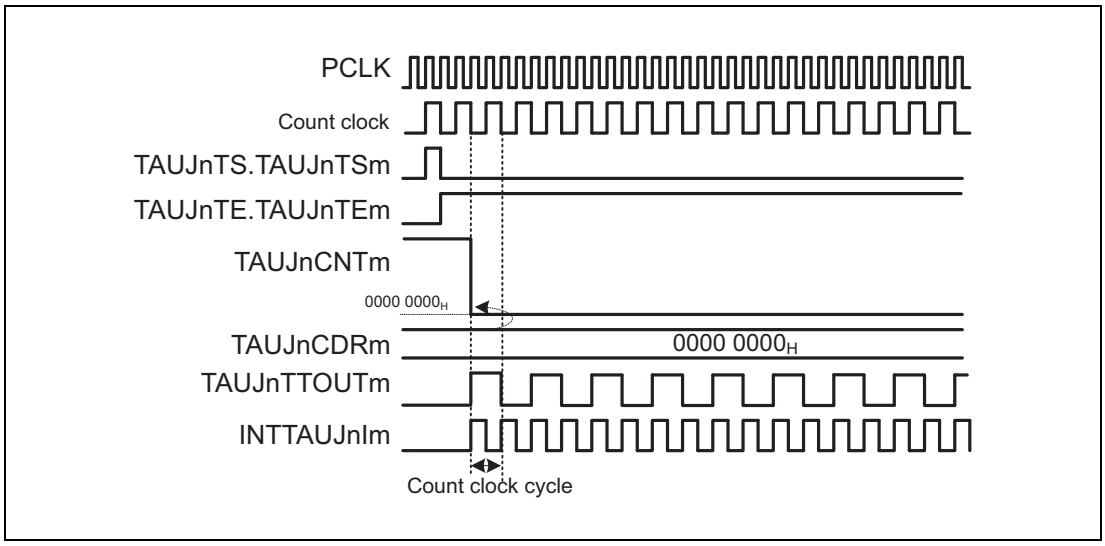


Figure 34.16 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK/2, the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is generated every count clock, resulting in TAUJnTTOUTm toggling every count clock.

(b) TAUJnCDRm = 0000 0000_H, Count Clock = PCLK

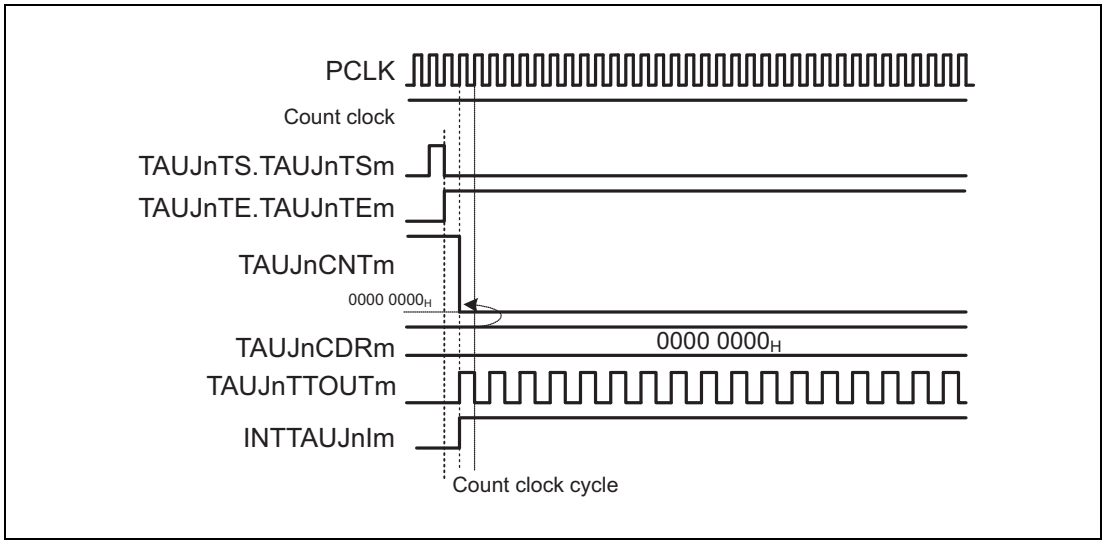


Figure 34.17 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated. TAUJnTTOUTm is toggled every PCLK clock.

(c) Operation stop and restart

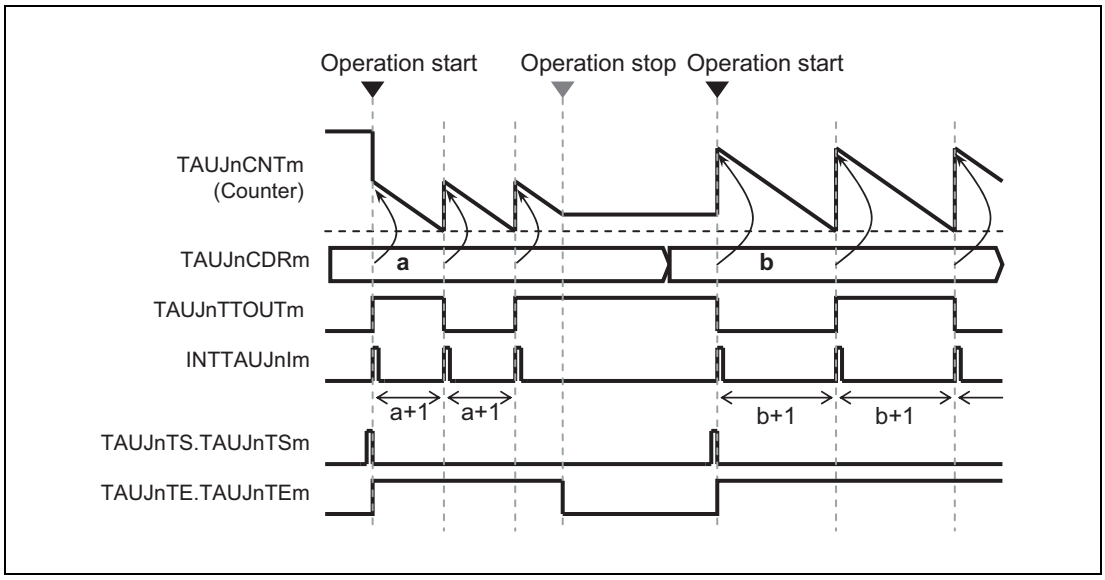


Figure 34.18 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(d) Forced restart

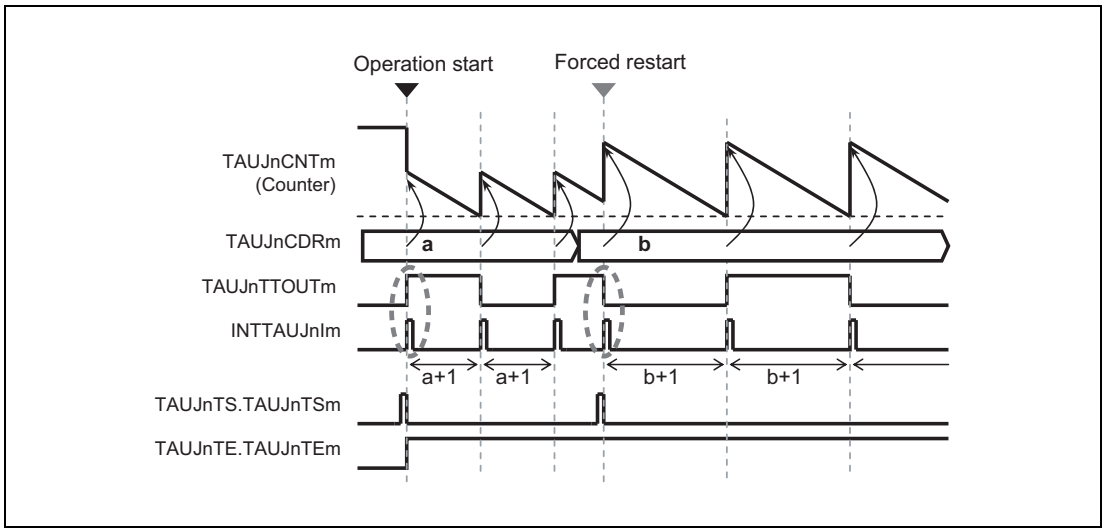


Figure 34.19 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUJnCDRm value is reflected to TAUJnCNTm and counting starts. Execute a forced restart to reflect the changed TAUJnCDRm value immediately.

34.4.9.2 TAUJnTTINm Input Interval Timer Function

(1) Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJnTTINm input edge is detected. When an interrupt is generated, the TAUJnTTOUTm signal toggles*1, resulting in a square wave.

Note 1. Following products do not support output of square waves.
TAUJ0: RH850/U2A16(292 pins), RH850/U2A8(292 pins), RH850/U2A6(292 pins), RH850/U2A6(176 pins), RH850/U2A6(156 pins) and RH850/U2A6(144 pins).
TAUJ1: RH850/U2A16(292 pins), RH850/U2A8(292 pins), RH850/U2A6(292 pins), RH850/U2A6(176 pins), RH850/U2A6(156 pins) and RH850/U2A6(144 pins).

Description

This function operates in an identical manner to the interval timer function (see **Section 34.4.9.1, Interval Timer Function**), except that this function is restarted by a valid TAUJnTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

(2) Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJnTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

(3) Block Diagram and General Timing Diagram

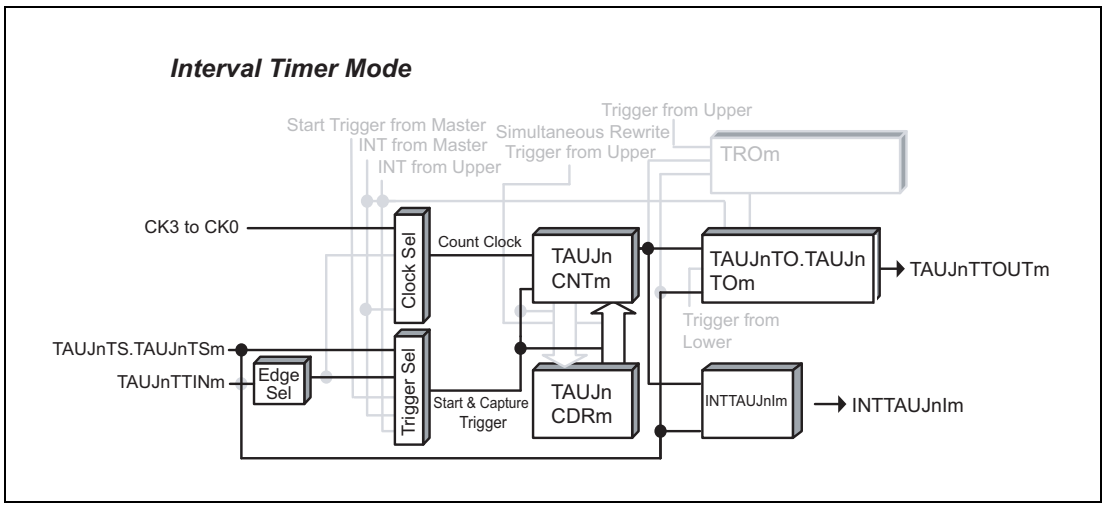


Figure 34.20 Block Diagram for TAUJnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

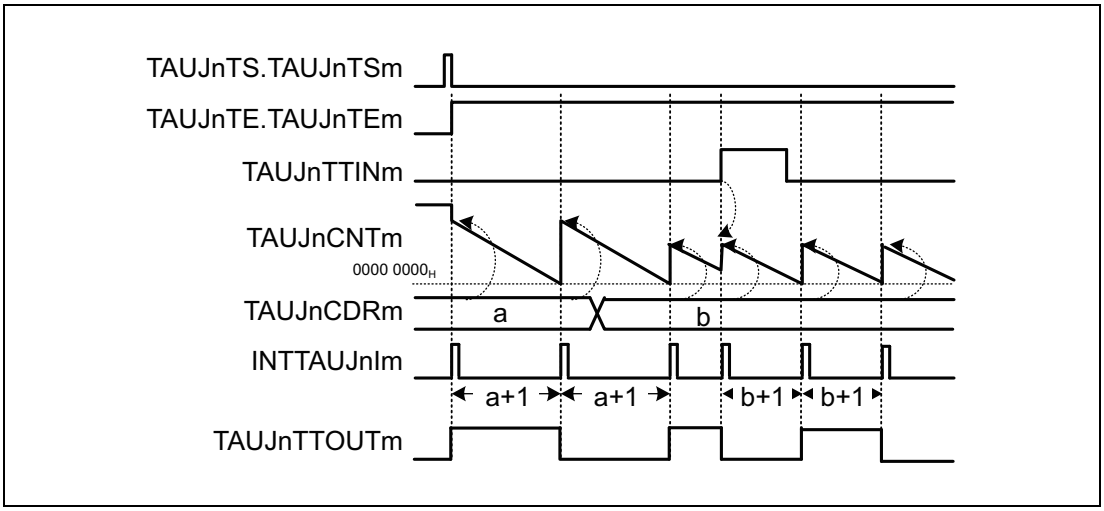


Figure 34.21 General Timing Diagram for TAUJnTTINm Input Interval Timer Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.39 Contents of the TAUJnCMORm register for TAUJnTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJnTTOUTm does not toggle when operation starts 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm when operation starts

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.40 Contents of the TAUJnCMURm register for TAUJnTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode

Table 34.41 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 34.4.4, Channel Output Modes**.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input interval timer function. Therefore, these registers must be set to 0.

Table 34.42 Simultaneous Rewrite Settings for TAUJnTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Interval Timer Function

Table 34.43 Operating Procedure for TAUJnTTINm Input Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 34.39, Contents of the TAUJnCMORm register for TAUJnTTINm Input Interval Timer Function** and **Table 34.40, Contents of the TAUJnCMURm register for TAUJnTTINm Input Interval Timer Function**. Set the value of the TAUJnCDRm register Set the channel output mode by setting the control bits as described in **Table 34.41, Control Bit Settings for Independent Channel Output Mode 1**.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJnTTINm edge	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation • INTTAUJnIm is generated and TAUJnTTOUTm toggles When a TAUJnTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart operation

(6) Specific Timing Diagrams

The timing diagrams in **Section 34.4.9.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUJnTTINm input edge.

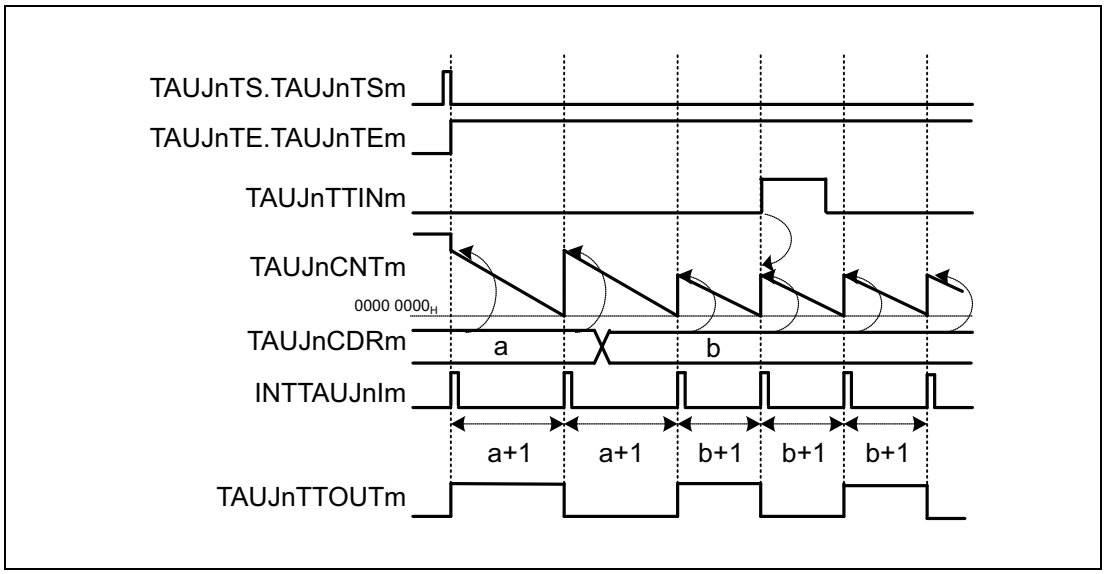


Figure 34.22 Counter Triggered by Rising TAUJnTTINm Input Edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

If a valid TAUJnTTINm input edge is detected, an interrupt is generated which causes TAUJnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

34.4.9.3 TAUJnTTINm Input Pulse Interval Measurement Function

(1) Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJnTTINm input signals.

Prerequisites

TAUJnTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm edge is detected, it overflows to 0000 0000_H. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 34.44 Effects of an Overflow

TAUJnCMORm. TAUJnCOS[1:0]	When Overflow Occurs		When a Valid TAUJnTTINm Input is then detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm written to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the value of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJnTTINm input valid edge detection and TAUJnCNTm capture are not performed.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 34.4.6, TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input pulse interval = count clock cycle × [(TAUJnCSRm.TAUJnOVF × (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

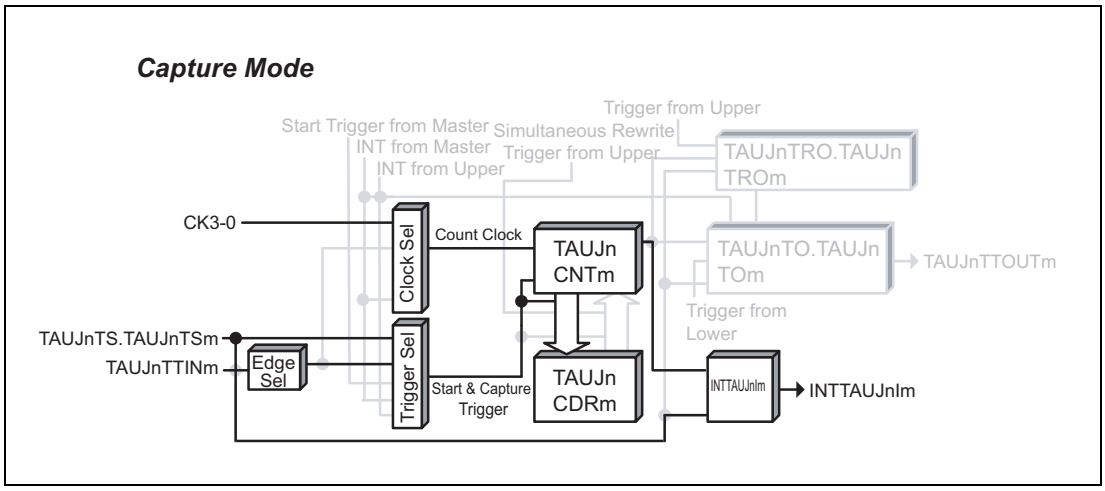


Figure 34.23 Block Diagram for TAUJnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

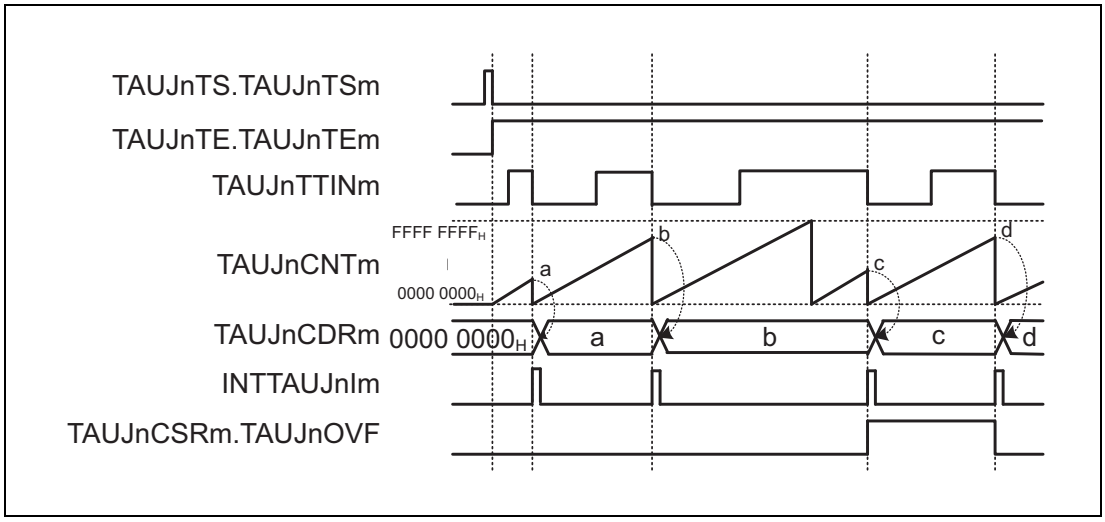


Figure 34.24 General Timing Diagram for TAUJnTTINm Input Pulse Interval Measurement Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.45 Contents of the TAUJnCMORm Register for TAUJnTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	See **Table 34.44, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0010_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts 1: Generates INTTAUJnIm when operation starts

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.46 Contents of the TAUJnCMURm Register for TAUJnTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input pulse interval measurement function. Therefore, these registers must be set to 0.

Table 34.47 Simultaneous Rewrite Settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

Table 34.48 Operating Procedure for TAUJnTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 34.45, Contents of the TAUJnCMORm Register for TAUJnTTINm Input Pulse Interval Measurement Function** and **Table 34.46, Contents of the TAUJnCMURm Register for TAUJnTTINm Input Pulse Interval Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJnTTINm edges. The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

(6) Specific Timing Diagrams: Overflow Behavior

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

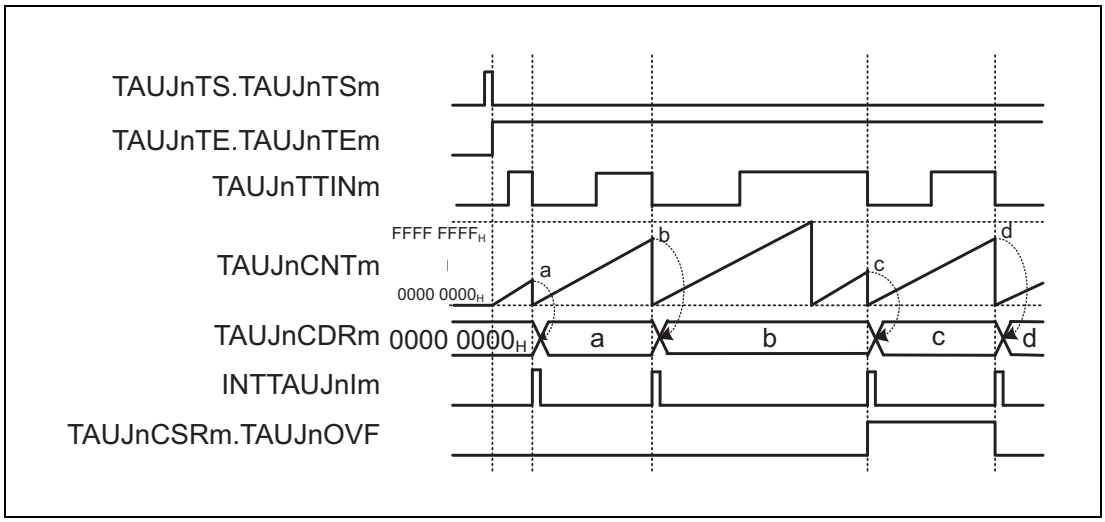


Figure 34.25 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJnTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

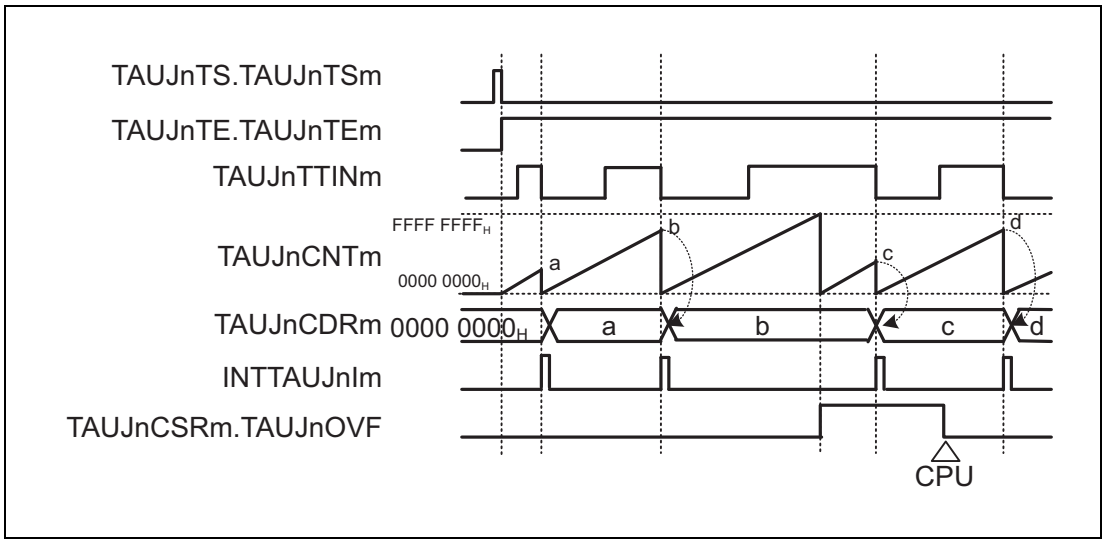


Figure 34.26 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

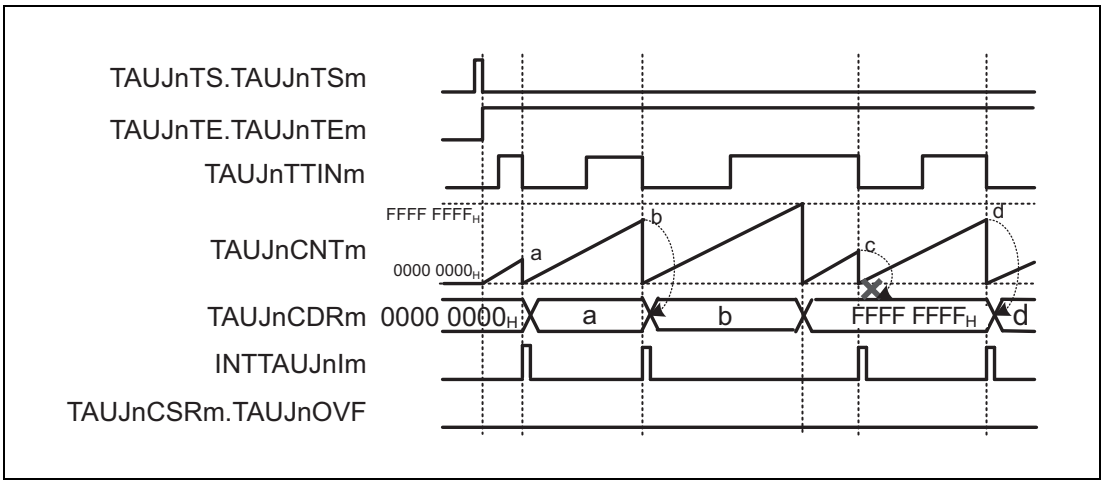


Figure 34.27 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

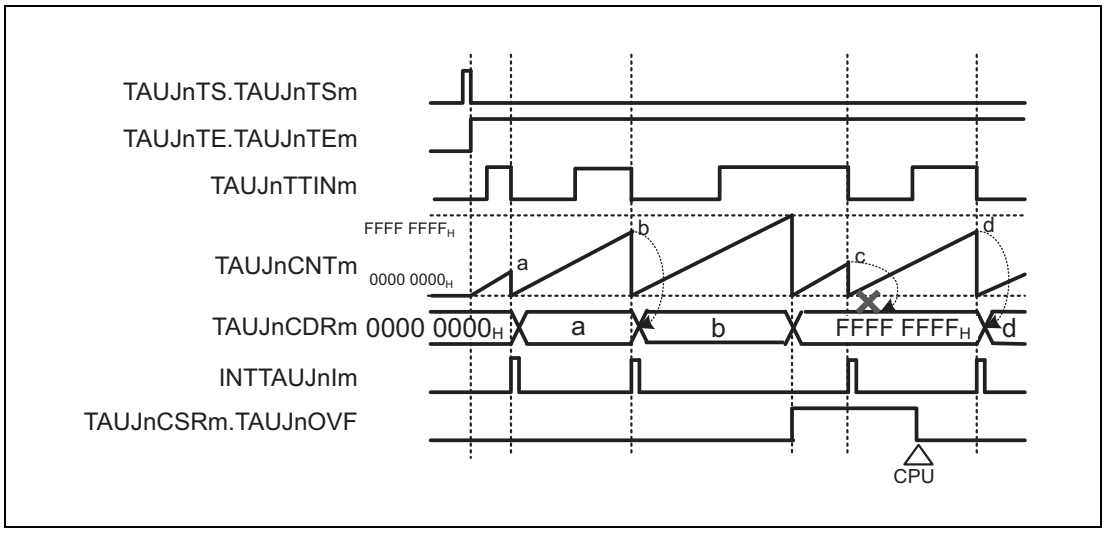


Figure 34.28 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

34.4.9.4 TAUJnTTINm Input Signal Width Measurement Function

(1) Overview

Summary

This function measures the width of a TAUJnTTINm signal by starting counting on one edge of the TAUJnTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

- The operation mode must be set to capture and one-count mode, refer to **Table 34.50, Contents of the TAUJnCMORm register for TAUJnTTINm Input Signal Width Measurement Function**.
- TAUJnTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJnTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJnTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 34.49 Effects of an Overflow

TAUJnCMORm. TAUJnCOS[1:0]	When overflow occurs		When a valid TAUJnTTINm input stop edge is detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input signal width = count clock cycle × [(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block Diagram and General Timing Diagram

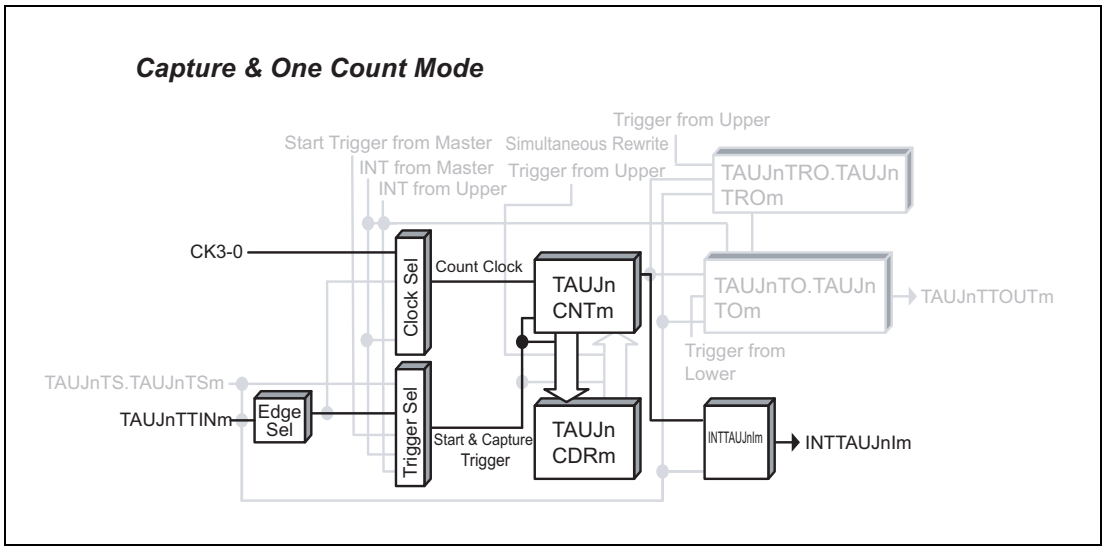


Figure 34.29 Block Diagram for TAUJnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJnTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

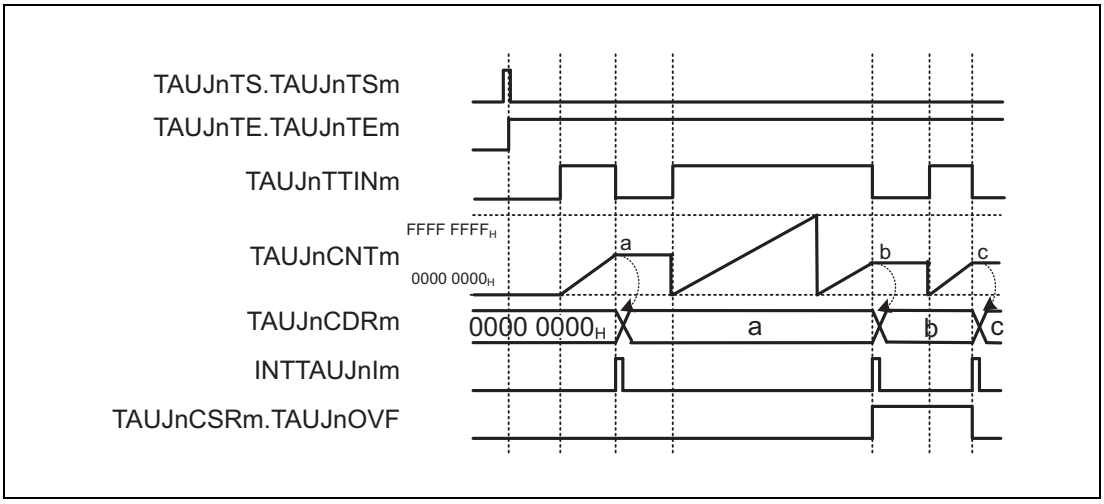


Figure 34.30 General Timing Diagram for TAUJnTTINm Input Signal Width Measurement Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.50 Contents of the TAUJnCMORm register for TAUJnTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	See **Table 34.49, Effects of an Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0110_B.
0	TAUJnMD0	Write 0_B.

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.51 Contents of the TAUJnCMURm Register for TAUJnTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input signal width measurement function. Therefore, these registers must be set to 0.

Table 34.52 Simultaneous Rewrite Settings for TAUJnTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

Table 34.53 Operating Procedure for TAUJnTTINm Input Signal Width Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 34.50, Contents of the TAUJnCMORm register for TAUJnTTINm Input Signal Width Measurement Function** and **Table 34.51, Contents of the TAUJnCMURm Register for TAUJnTTINm Input Signal Width Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. When a TAUJnTTINm start edge is detected, TAUJnCNTm starts to count up.
During operation	The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value • INTTAUJnIm is then generated. • Counting stops at the “value that transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

(6) Specific Timing Diagrams: Overflow Behavior

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

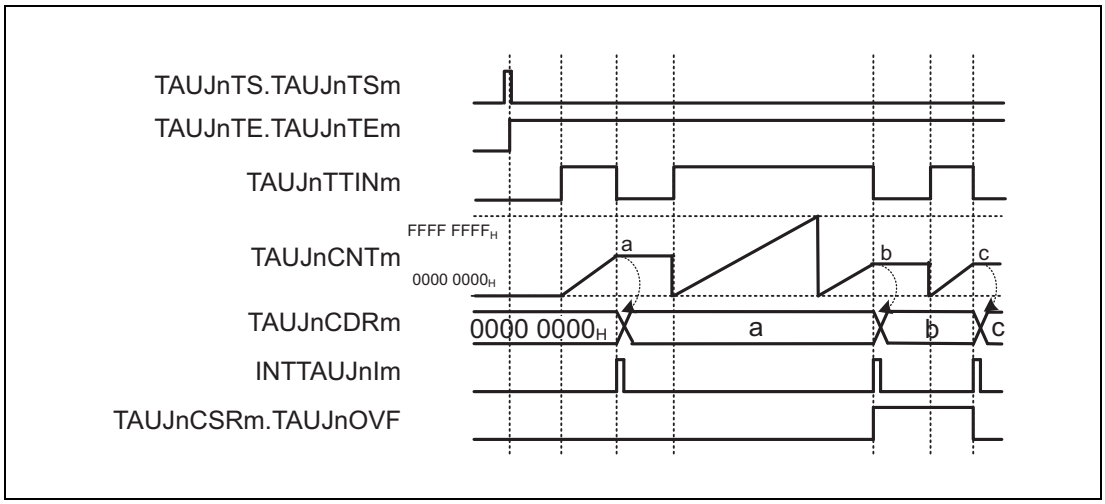


Figure 34.31 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

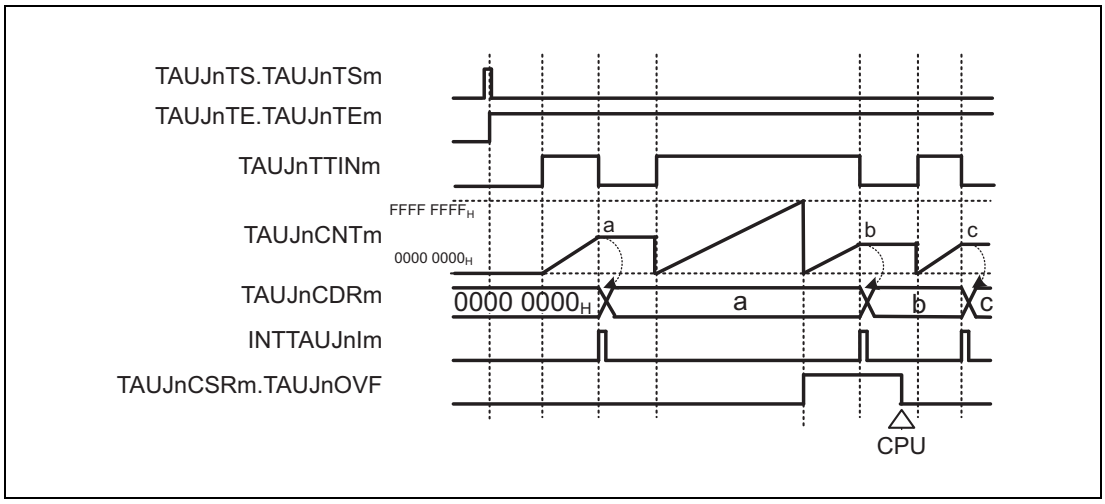


Figure 34.32 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

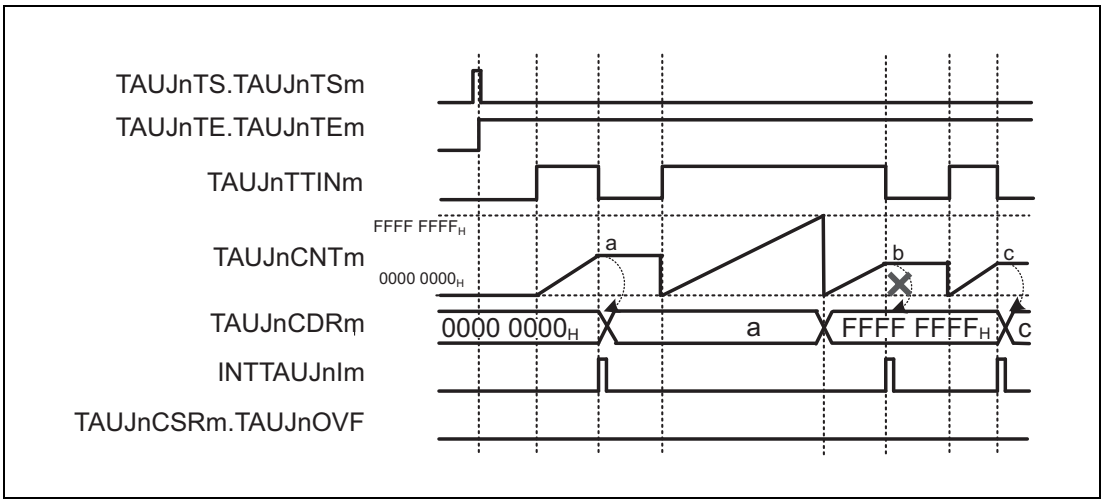


Figure 34.33 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and the value of TAUJnCSRm.TAUJnOVF remains = 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

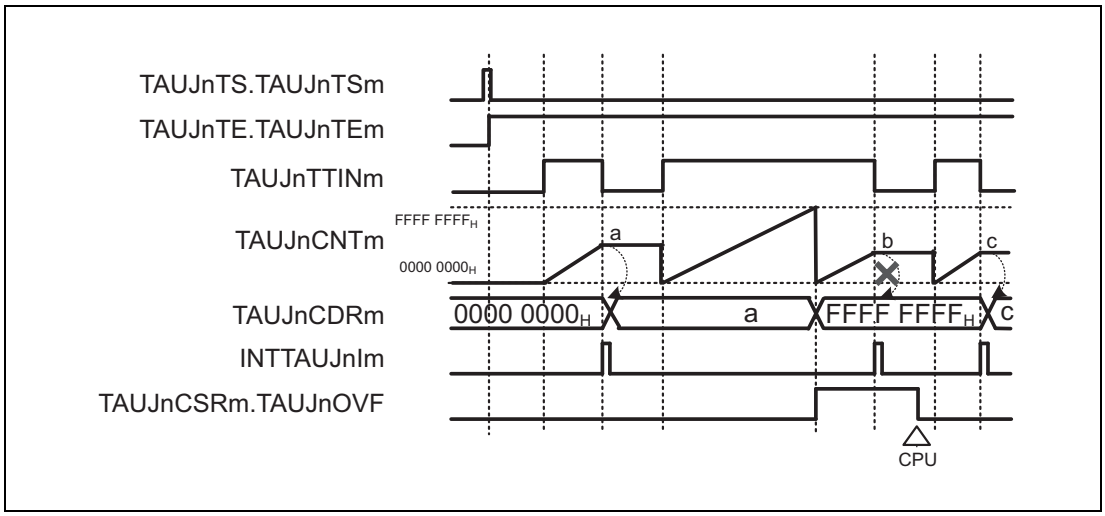


Figure 34.34 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

34.4.9.5 TAUJnTTINm Input Position Detection Function

(1) Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUJnTTINm signal.

Prerequisites

TAUJnTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 0000_H. When a valid TAUJnTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

NOTE

The input TAUJnTTINm is sampled at the frequency of the operation clock, specified by TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

If the TAUJnCMORm. TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 34.4.6, TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

(2) Equations

Function duration at a TAUJnTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

(3) Block Diagram and General Timing Diagram

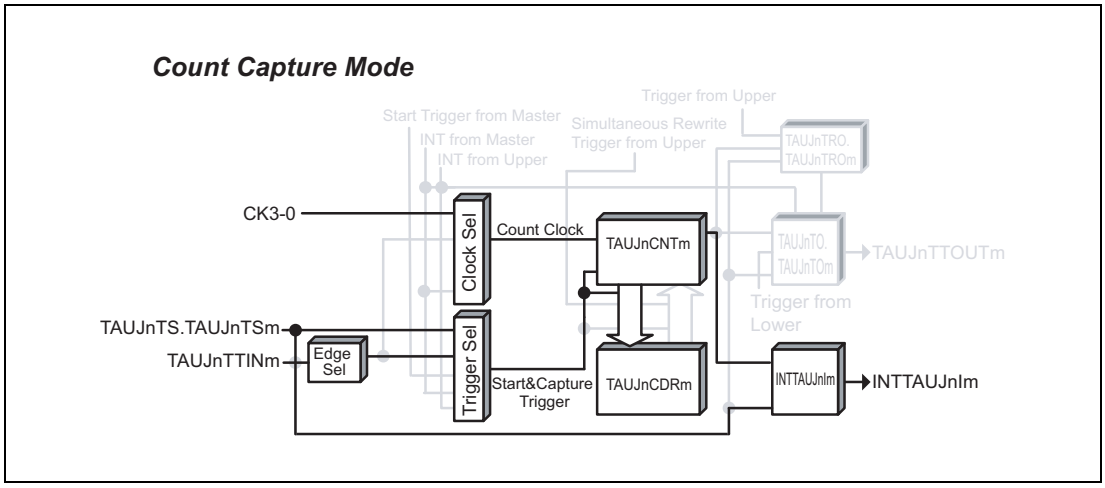


Figure 34.35 Block Diagram of TAUJnTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

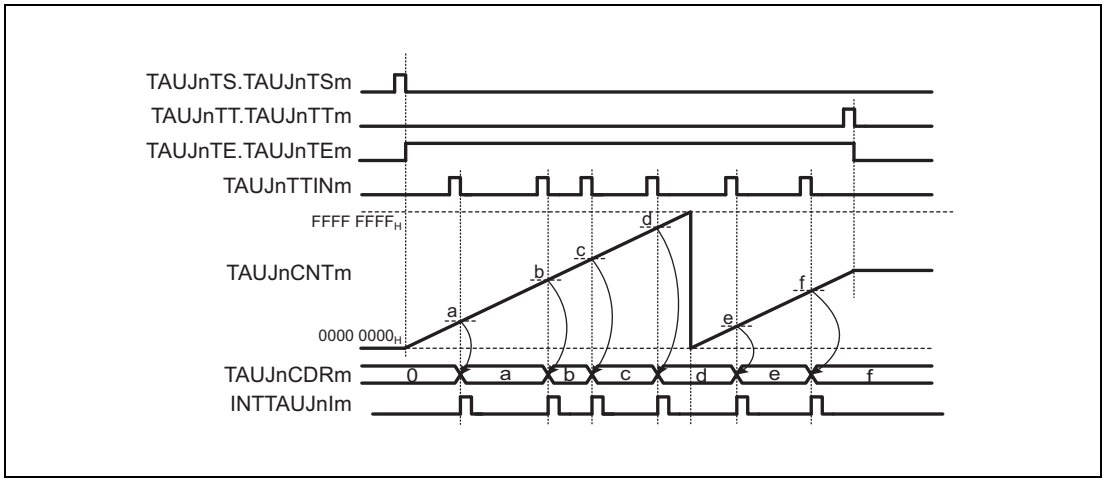


Figure 34.36 General Timing Diagram for TAUJnTTINm Input Position Detection Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.54 Contents of the TAUJnCMORm Register for TAUJnTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1011_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts 1: Generates INTTAUJnIm when operation starts

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.55 Contents of the TAUJnCMURm Register for TAUJnTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode

The channel output mode is not used by this function.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input position detection function. Therefore, these registers must be set to 0.

Table 34.56 Simultaneous Rewrite Settings for TAUJnTTINm Input Position Detection Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Position Detection Function

Table 34.57 Operating Procedure for TAUJnTTINm Input Position Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 34.54, Contents of the TAUJnCMORm Register for TAUJnTTINm Input Position Detection Function** and **Table 34.55, Contents of the TAUJnCMURm Register for TAUJnTTINm Input Position Detection Function**. The TAUJnCDRm register operates as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	The TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm • INTTAUJnIm is output. • The counter value is not cleared to 0000 0000_H and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated. When TAUJnCNTm reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

(6) Specific Timing Diagrams

(a) Operation stop and restart

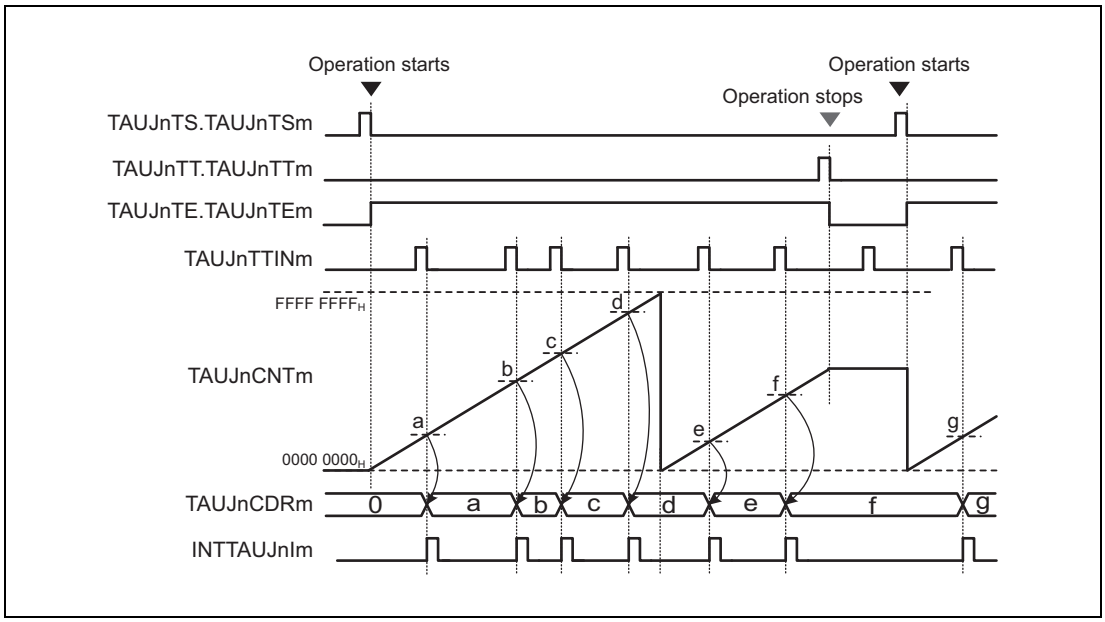


Figure 34.37 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

34.4.9.6 TAUJnTTINm Input Period Count Detection Function

(1) Overview

Summary

This function measures the cumulative width of a TAUJnTTINm input signal.

Prerequisites

TAUJnTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJnTTINm input edge.

When a valid TAUJnTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJnTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJnTTINm input start edge is detected.

When the next valid TAUJnTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF $FFFF_H$ the counter restarts from 0000 0000_H.

This function cannot be forcibly restarted.

NOTE

The input TAUJnTTINm signal is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits:

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TIS[1:0] = 11_B, the TAUJnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

(2) Equations

Cumulative TAUJnTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

(3) Block Diagram and General Timing Diagram

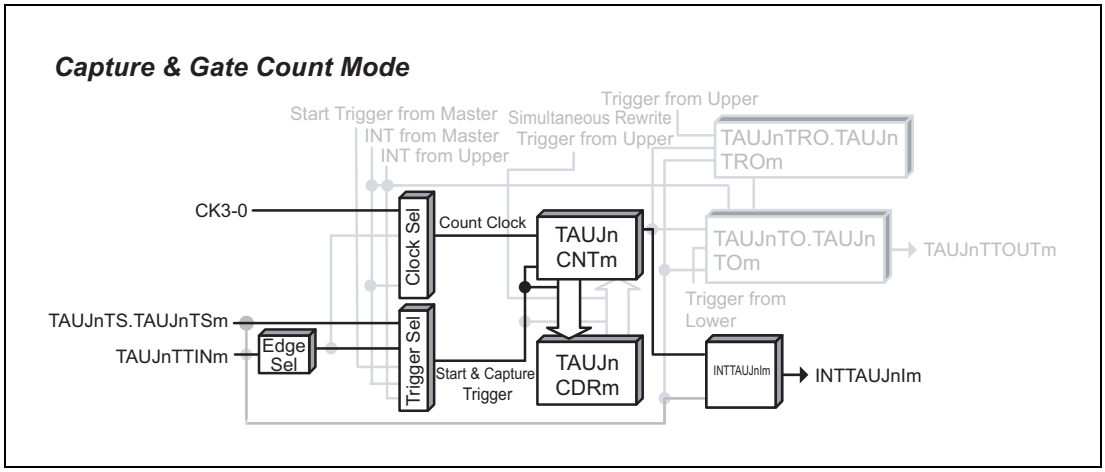


Figure 34.38 Block Diagram for TAUJnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

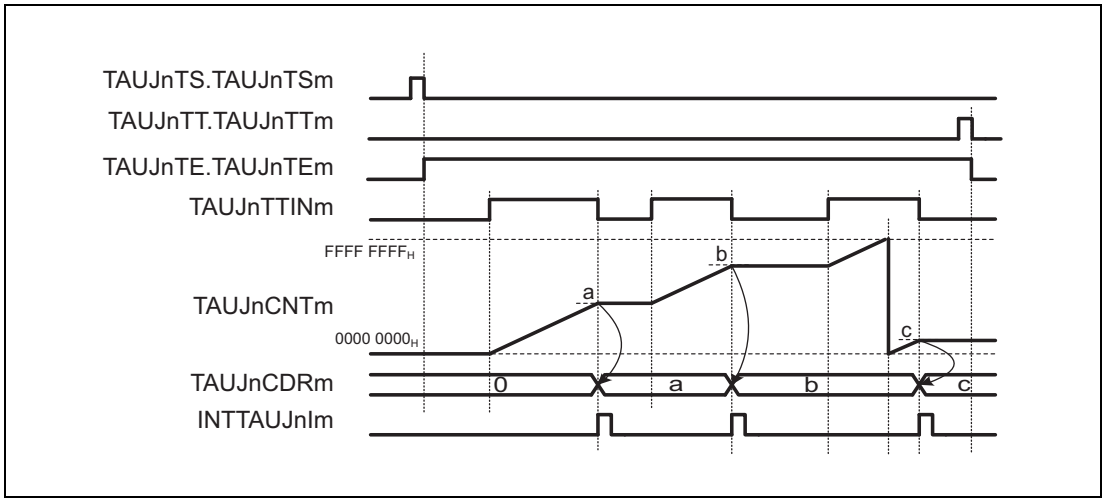


Figure 34.39 General Timing Diagram for TAUJnTTINm Input Period Count Detection Function

(4) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.58 Contents of the TAUJnCMORm Register for TAUJnTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1101_B.
0	TAUJnMD0	Write 0_B.

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.59 Contents of the TAUJnCMURm Register for TAUJnTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input period count detection function. Therefore, these registers must be set to 0.

Table 34.60 Simultaneous Rewrite Settings for TAUJnTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

(5) Operating Procedure for TAUJnTTINm Input Period Count Detection Function

Table 34.61 Operating Procedure for TAUJnTTINm Input Period Count Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 34.58, Contents of the TAUJnCMORm Register for TAUJnTTINm Input Period Count Detection Function** and **Table 34.59, Contents of the TAUJnCMURm Register for TAUJnTTINm Input Period Count Detection Function**. The TAUJnCDRm register operates as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge.
During operation	TAUJnTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time.	When a TAUJnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. If the TAUJnCNTm reaches FFFF FFFF$_H$, it restarts counting from 0000 0000$_H$. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

(6) Specific Timing Diagrams

(a) Operation stop and restart

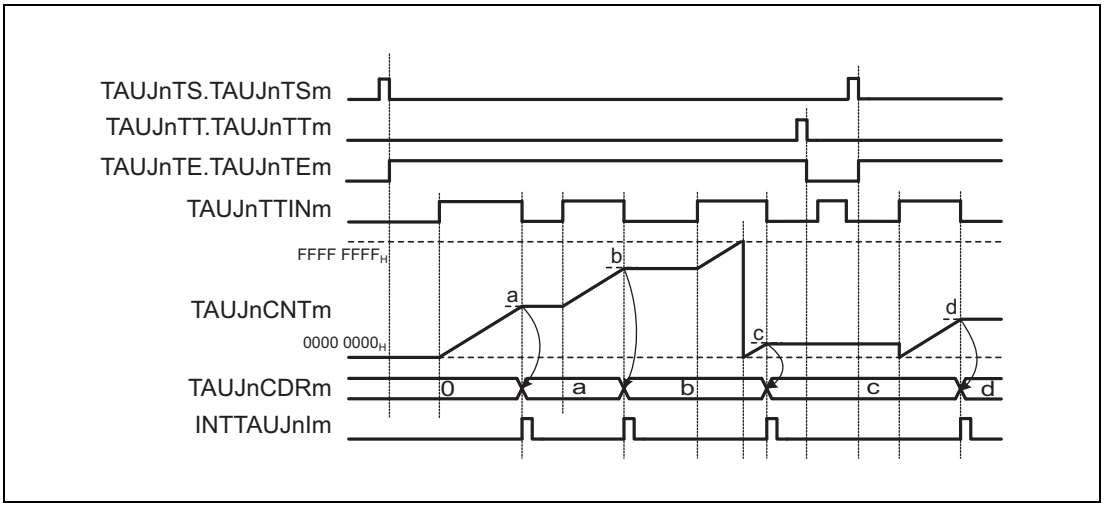


Figure 34.40 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS. TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

34.4.9.7 Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

(1) Overview

Summary

This function measures the width of an individual TAUJnTTINm input signal. An interrupt is generated if the TAUJnTTINm input width is longer than FFFF $FFFF_H$ + 1.

Prerequisites

- TAUJnTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF $FFFF_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJnTTINm input start edge is detected. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJnTTINm input start edge is detected, TAUJnCNTm loads FFFF $FFFF_H$ and starts to count down.

If the counter reaches 0000 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJnTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

(2) Block Diagram and General Timing Diagram

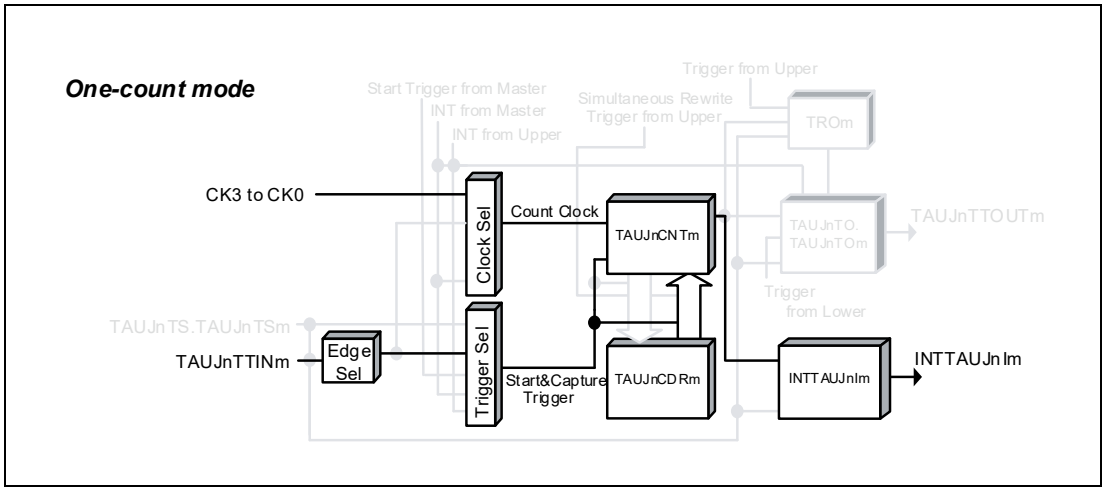


Figure 34.41 Block Diagram for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

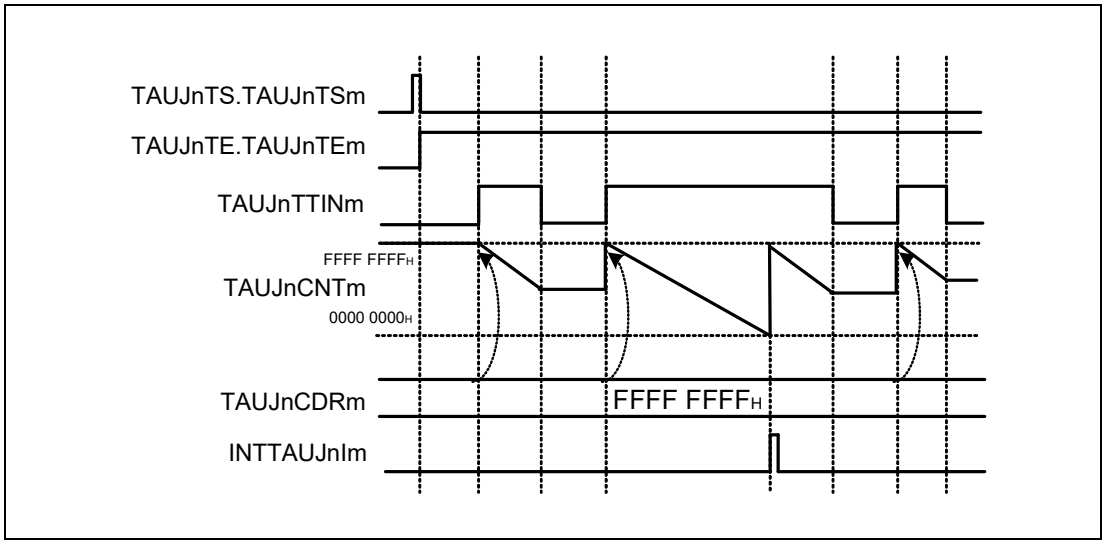


Figure 34.42 General Timing Diagram for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

(3) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.62 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 0_B.

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.63 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous Rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the overflow interrupt output function (during TAUJnTTINm width measurement). Therefore, these registers must be set to 0.

Table 34.64 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(4) Operating Procedure for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

Table 34.65 Operating Procedure for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 34.62, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)** and **Table 34.63, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJnTTINm Width Measurement)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJnTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at any time.	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. When a reverse edge of TAUJnTTINm is detected during count operation: • TAUJnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

34.4.9.8 Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

(1) Overview

Summary

This function measures the cumulative width of a TAUJnTTINm input signal. An interrupt is generated if the cumulative TAUJnTTINm input width is longer than FFFF $FFFF_H$, and an overflow interrupt can be output.

Prerequisites

- TAUJnTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF $FFFF_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJnTTINm input start edge is detected. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJnTTINm input start edge and then continues to count down from the current value.

When the counter reaches $0000\ 0000_H$ an interrupt is generated. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter continues to count down until a TAUJnTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

(2) Block Diagram and General Timing Diagram

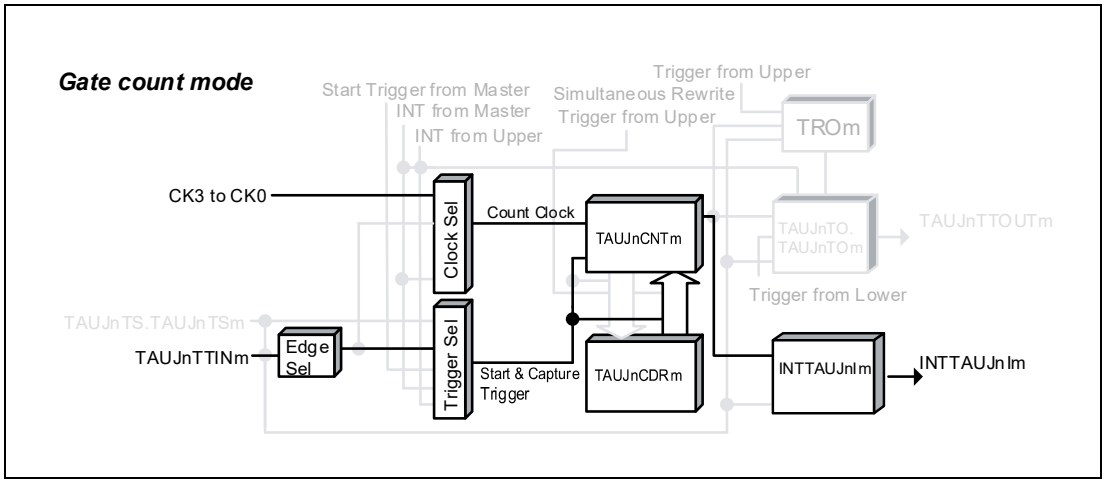


Figure 34.43 Block Diagram for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

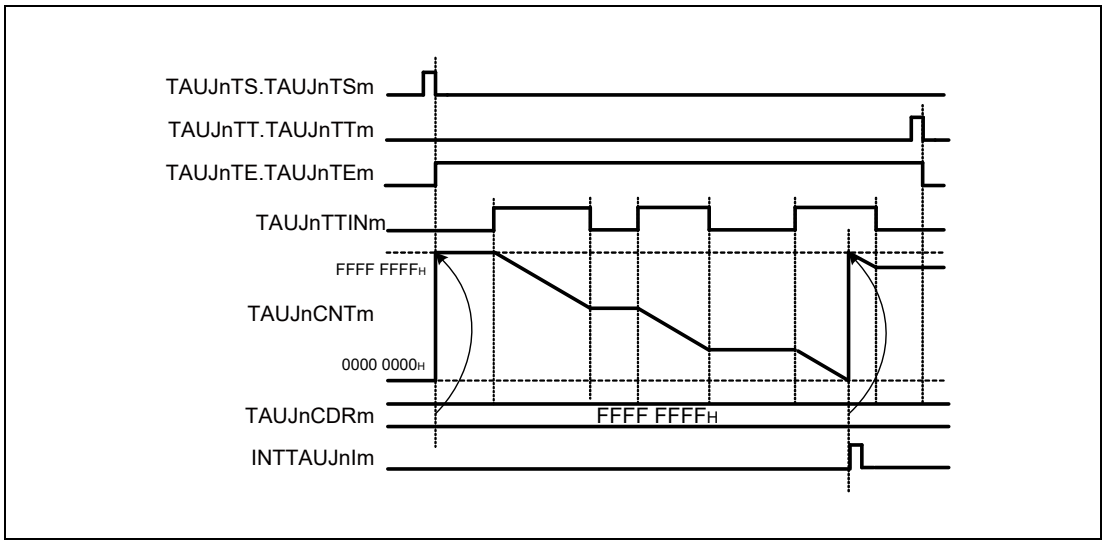


Figure 34.44 General Timing Diagram For Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

(3) Register Settings

(a) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.66 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1100_B.
0	TAUJnMD0	Write 0_B.

(b) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.67 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 34.68 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(4) Operating Procedure for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

Table 34.69 Operating Procedure for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 34.66, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)** and **Table 34.67, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJnTTINm Input Period Count Detection)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJnTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, the value of TAUJnCDRm (FFFF FFFF$_H$) is loaded to TAUJnCNTm.
During operation	The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000H: ●INTTAUJnIm is generated. ●TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$) and continues to count down. When a reverse edge of TAUJnTTINm is detected during count operation: ●TAUJnCNTm stops and retains the stop value. When a TAUJnTTINm valid edge is detected while the counter is stopped: ●TAUJnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

34.4.10 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see **Section 34.2, Overview**.

34.4.10.1 PWM output function

(1) Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJnTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operation mode for the master channel should be set to interval timer mode (see **Table 34.70, Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operation mode for the slave channel should be set to the one-count mode (see **Table 34.73, Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUJnTTOUTm is not used for the master channel of this function.
- The channel output mode for the slave channels should be set to synchronous channel output mode 1 (see **Section 34.4.4, Channel Output Modes**.)

Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

  When the counter of the master channel reaches $0000\ 0000_H$ and pulse cycle time has elapsed, INTTAUJnIm is generated. The TAUJnCDRm value is loaded to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

  INTTAUJnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJnTTOUTm signal is set to the active level.
  When the counter reaches $0000\ 0000_H$, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJnTTOUTm signal is set to the inactive level. The counter returns to $FFFF\ FFFF_H$ and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Refer to **Section 34.4.3, Simultaneous Rewrite**.

(2) Equations

Pulse cycle = (TAUJnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave)/(TAUJnCDRm (master) + 1)) × 100

– Duty cycle = 0%
  TAUJnCDRm (slave) = 0000 0000_H
– Duty cycle = 100%
  TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

(3) Block Diagram and General Timing Diagram

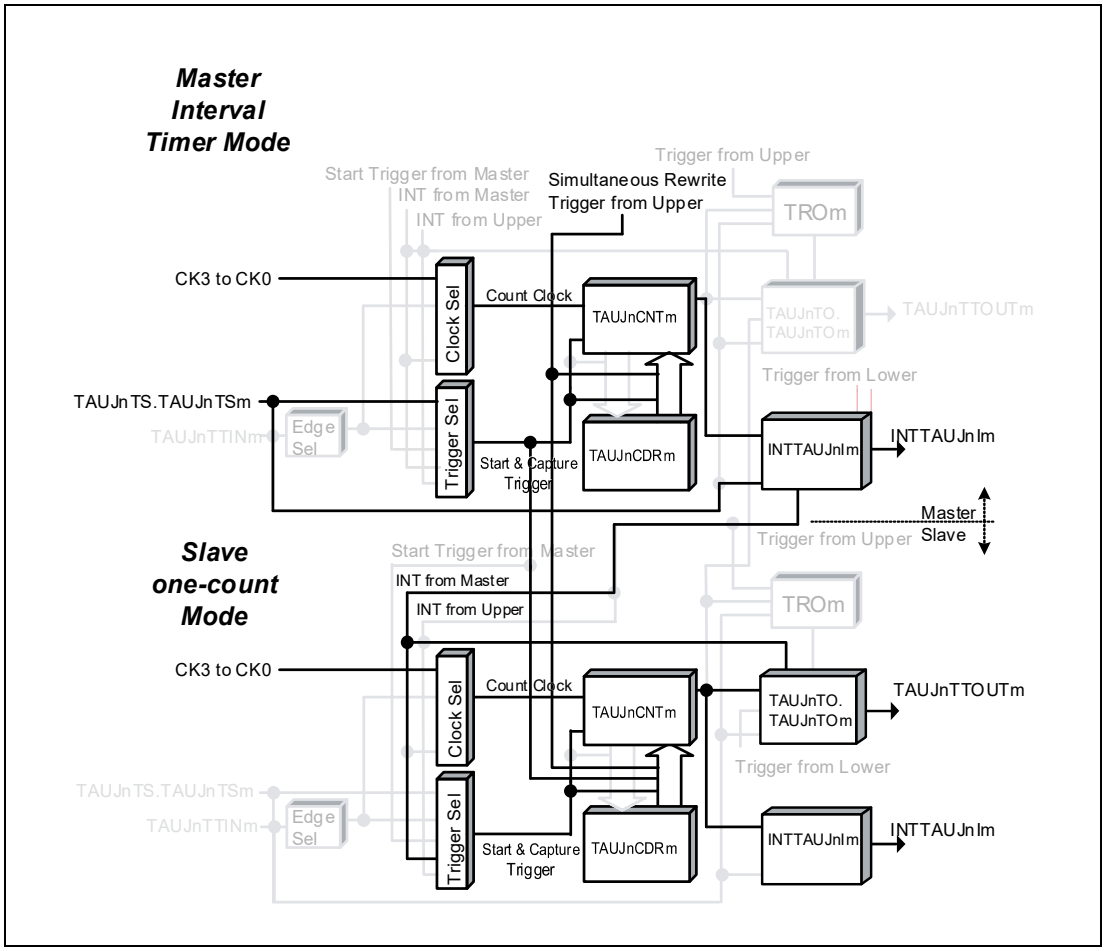


Figure 34.45 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

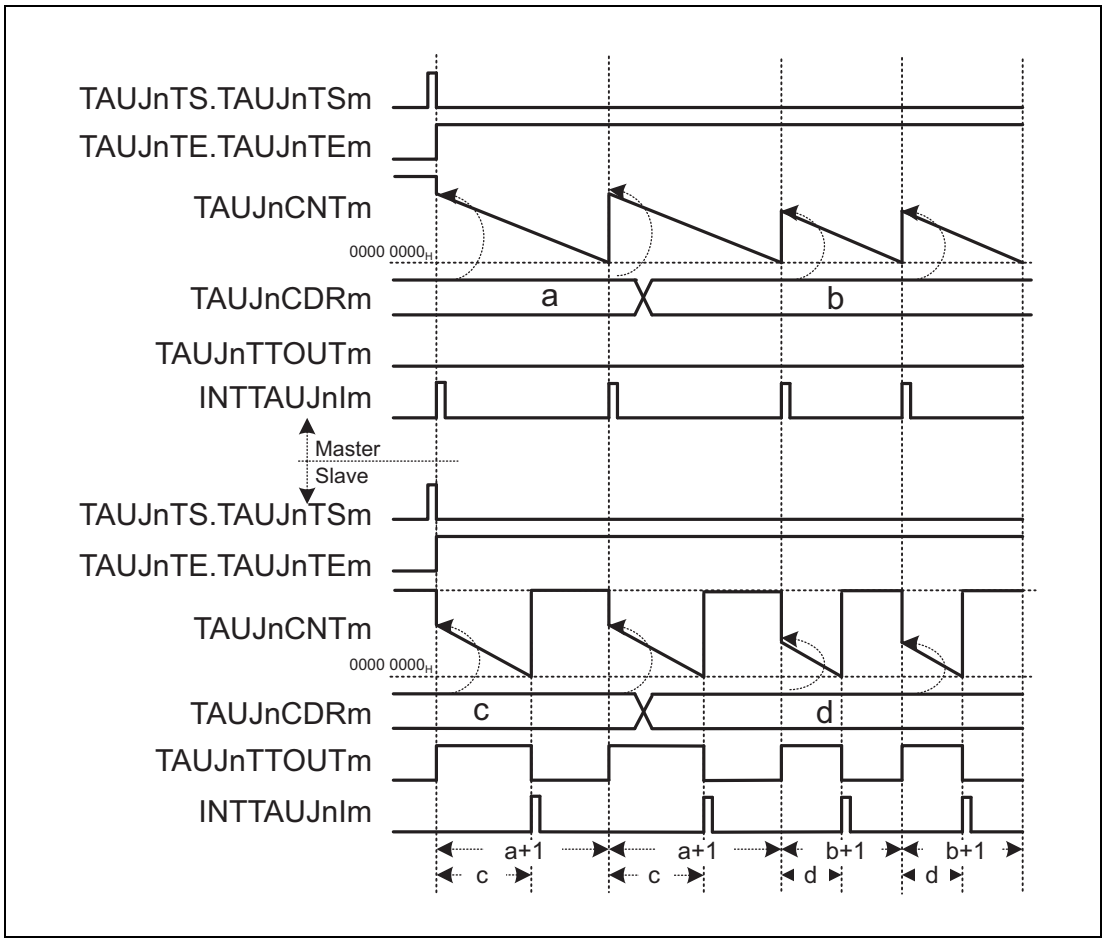


Figure 34.46 General Timing Diagram for PWM Output Function

NOTE

- The interval between the starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm +1.
- TAUJnTTOUTm of the slave channel will rise with a delay of one count clock after the rising of INTTAUJnIm of the master channel.

(4) Register Settings for the Master Channel

(a) TAUJnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.70 Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bit of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 1_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	Write 1_B.

(b) TAUJnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.71 Contents of the TAUJnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode for the master channel

The channel output mode is not used by this function.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 34.72 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting

(5) Register Settings for the Slave Channel(s)

(a) TAUJnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJnMAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 34.73 Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 100_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 1_B.

(b) TAUJnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 34.74 Contents of the TAUJnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode for the slave channel(s)

Table 34.75 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 1_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Negative logic

(d) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 34.76 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting

(6) Operating Procedure for PWM Output Function

Table 34.77 Operating Procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	Master channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section (4), Register Settings for the Master Channel**. Slave channel: set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section (5), Register Settings for the Slave Channel(s)**. Set the values of the TAUJnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set.
During operation	TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation.	TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000_H: • INTTAUJnIm (master) is generated • TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation • TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down • TAUJnTTOUTm (slave) is set to the active level. When TAUJnCNTm (slave) reaches 0000 0000_H: • INTTAUJnIm (slave) is generated • TAUJnTTOUTm (slave) is set to the inactive level.
Stop operation	Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart operation

(7) Specific Timing Diagrams

(a) Duty cycle = 0%

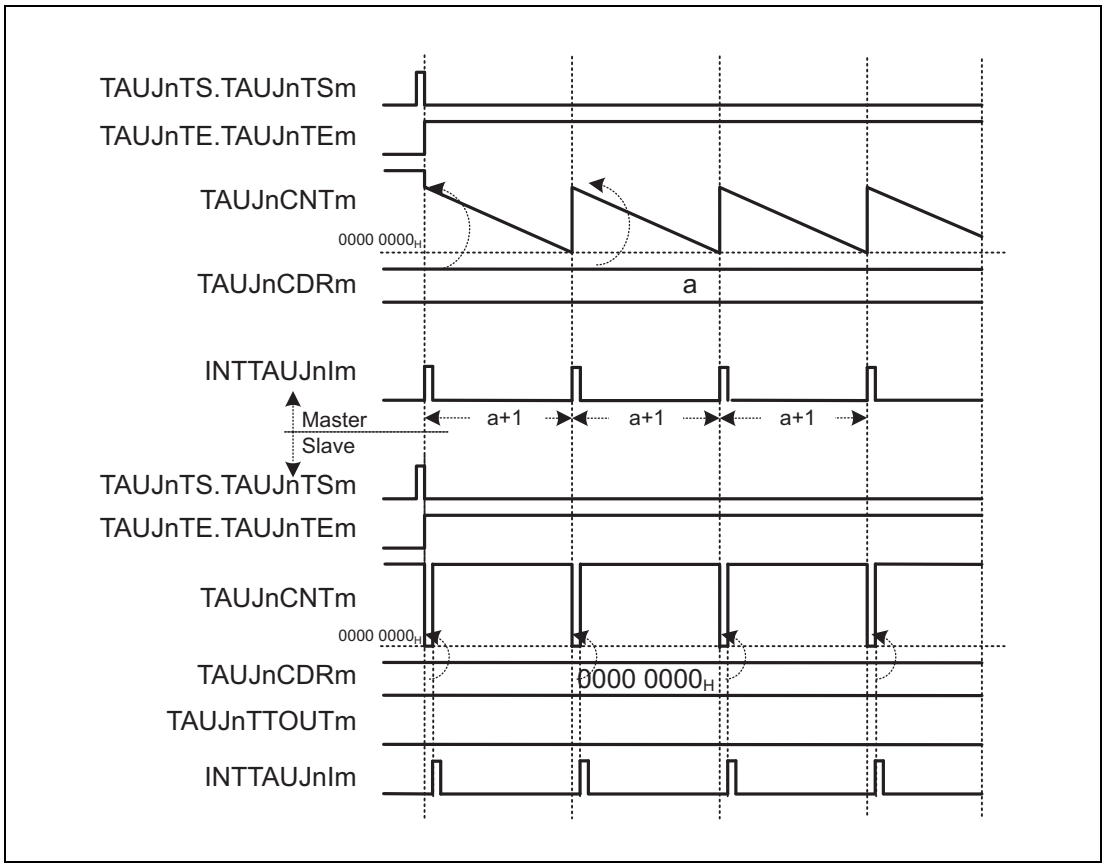


Figure 34.47 TAUJnCDRm (Slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded to TAUJnCNTm (slave). As a result, a slave channel interrupt (INTTAUJnIm) is generated at the same time and TAUJnTTOUTm remains inactive.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(b) Duty cycle = 100%

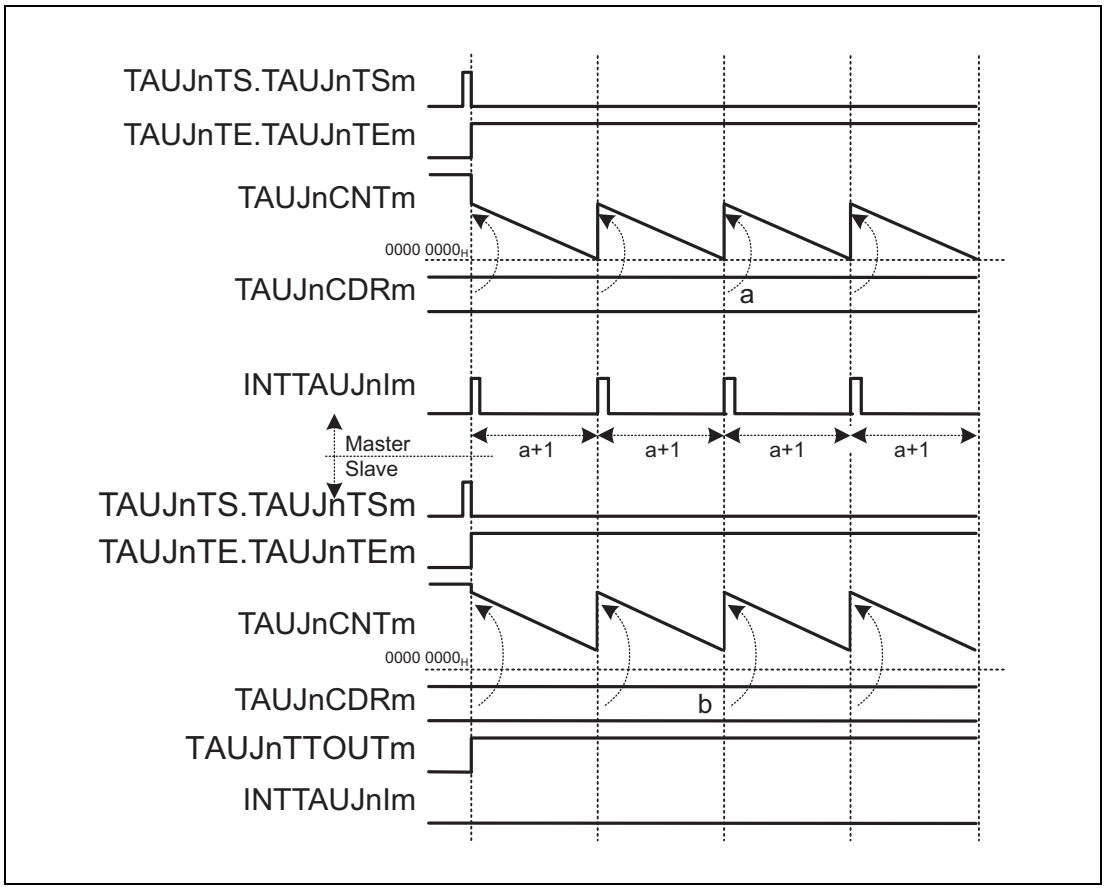


Figure 34.48 TAUJnCDRm (Slave) ≥ TAUJnCDRm (Master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000_H. TAUJnTTOUTm remains active.

(c) Operation stop and restart

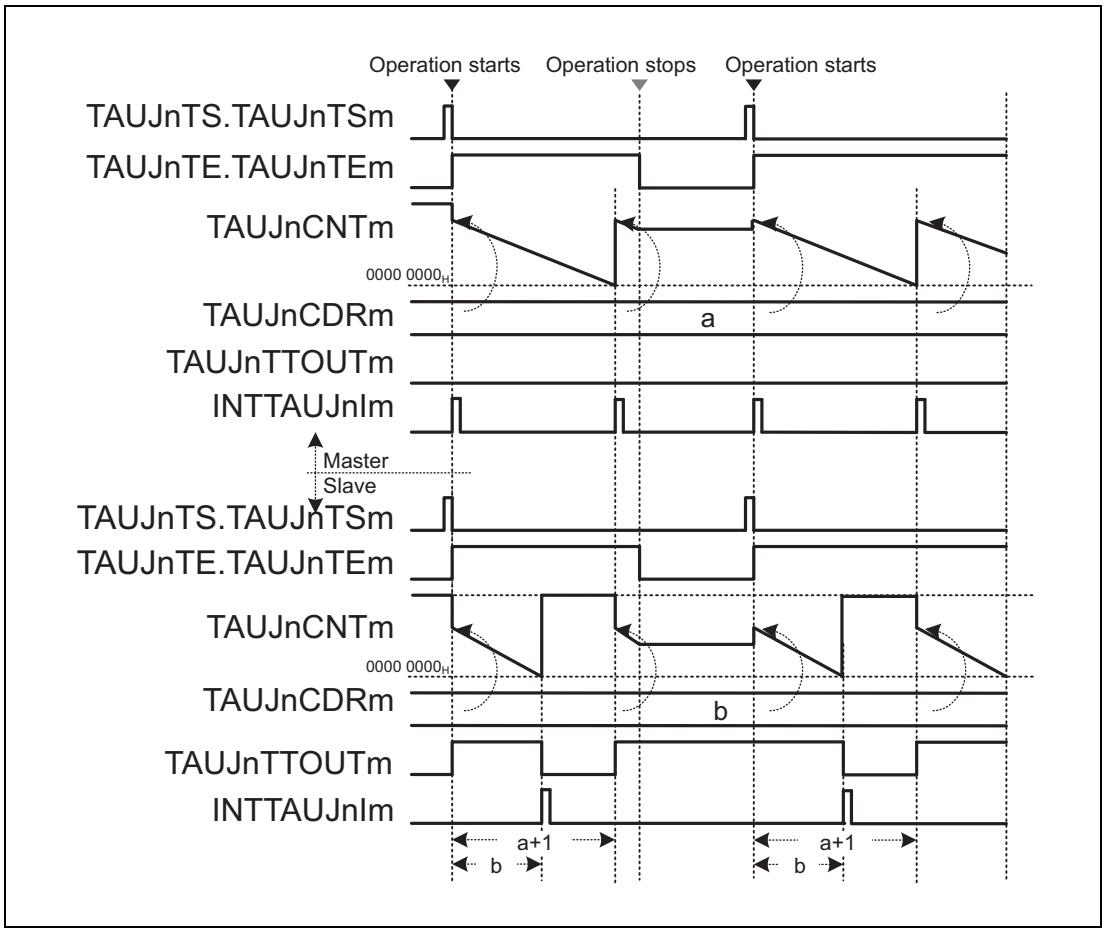


Figure 34.49 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. The TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

Section 35 Motor Control Timer (TSG3)

The Motor Control Timer (TSG3) is 18-bit counter which can generate the output corresponding to various motor control systems. TSG3 is implemented 2 units.

35.1 Features TSG3 for RH850/U2A-EVA

35.1.1 Number of Units

This microcontroller has the following number of units of TSG3 units.

Table 35.1 Number of Units

Product	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	1 (n = 1)	1 (n = 1)
Name	TSG3n									

Table 35.2 Index

Index	Description
n	Throughout this section, the individual TSG3 units are identified by the index “n” (n = 0, 1), for example, TSG3nCTL0 refers to the TSG3n control register 0.
m, k	Throughout this section, the variables used for description are indicated by the index “m” or “k”, for example, TSG3nCMPmE refers to a specified compare register.

35.1.2 Register Base Addresses

TSG3 base addresses are listed in the following table.

TSG3 register addresses are given as offsets from the base addresses.

Table 35.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<TSG30_base>	FFBF 8000_H	P-Bus Group 5
<TSG31_base>	FFBF 8800_H	P-Bus Group 5

35.1.3 Clock Supply

Clock supply by and to TSG3 is listed in the following table.

Table 35.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TSG3n	PCLK	High-speed peripheral clock CLK_HSB

35.1.4 Interrupt Requests and Error Notifications

TSG3 interrupt requests are listed in the following table.

Table 35.5 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
TSG30				
INTTSG30I0	TSG30 compare match interrupt 0/ TSG30 period interrupt (in HT-PWM mode only)	165	—	—
INTTSG30I1	TSG30 compare match interrupt 1	166	—	—
INTTSG30I2	TSG30 compare match interrupt 2	167	—	—
INTTSG30I3	TSG30 compare match interrupt 3	168	—	—
INTTSG30I4	TSG30 compare match interrupt 4	169	—	—
INTTSG30I5	TSG30 compare match interrupt 5	170	—	—
INTTSG30I6	TSG30 compare match interrupt 6	171	—	—
INTTSG30I7	TSG30 compare match interrupt 7	172	—	—
INTTSG30I8	TSG30 compare match interrupt 8	173	—	—
INTTSG30I9	TSG30 compare match interrupt 9	174	—	—
INTTSG30I10	TSG30 compare match interrupt 10	175	—	—
INTTSG30I11	TSG30 compare match interrupt 11	176	Group0-173	Group1-46
INTTSG30I12	TSG30 compare match interrupt 12	177	Group0-174	Group1-47
INTTSG30IPEK	TSG30 peak interrupt	178	Group0-175	Group1-48
INTTSG30IVLY	TSG30 valley interrupt	179	Group0-176	Group1-49
INTTSG30IER	TSG30 error interrupt	180	—	—
INTTSG30IWN	TSG30 warning interrupt	181	—	—
TSG31				
INTTSG31I0	TSG31 compare match interrupt 0/ TSG31 period interrupt (in HT-PWM mode only)	182	—	—
INTTSG31I1	TSG31 compare match interrupt 1	183	—	—
INTTSG31I2	TSG31 compare match interrupt 2	184	—	—
INTTSG31I3	TSG31 compare match interrupt 3	185	—	—
INTTSG31I4	TSG31 compare match interrupt 4	186	—	—
INTTSG31I5	TSG31 compare match interrupt 5	187	—	—
INTTSG31I6	TSG31 compare match interrupt 6	188	—	—
INTTSG31I7	TSG31 compare match interrupt 7	189	—	—
INTTSG31I8	TSG31 compare match interrupt 8	190	—	—
INTTSG31I9	TSG31 compare match interrupt 9	191	—	—
INTTSG31I10	TSG31 compare match interrupt 10	192	—	—
INTTSG31I11	TSG31 compare match interrupt 11	193	Group0-177	Group1-50
INTTSG31I12	TSG31 compare match interrupt 12	194	Group0-178	Group1-51
INTTSG31IPEK	TSG31 peak interrupt	195	Group0-179	Group1-52
INTTSG31IVLY	TSG31 valley interrupt	196	Group0-180	Group1-53
INTTSG31IER	TSG31 error interrupt	197	—	—
INTTSG31IWN	TSG31 warning interrupt	198	—	—

This module has no error notifications.

35.1.5 Reset Sources

TSG3 reset sources are listed in the following table. TSG3n is initialized by these reset sources.

Table 35.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
TSG3n	All registers	√	√	√	√	√	√	—

35.1.6 External Input/Output Signals

External input/output signals of TSG3 are listed in the following table.

Table 35.7 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
TSG30			
TSG30CLKI	I	Clock enable input	TSG30CLKI
TSG30PTSI0 to TSG30PTSI2	I	External pattern input	ENCA0E0, ENCA0E1, ENCA0EC*1
TSG30O0 to TSG30O7	O	Timer output	TSG30O0 to TSG30O7
TSG31			
TSG31CLKI	I	Clock enable input	TSG31CLKI
TSG31PTSI0 to TSG31PTSI2	I	External pattern input	ENCA1E0, ENCA1E1, ENCA1EC*1
TSG31O0 to TSG31O7	O	Timer output	TSG31O0 to TSG31O7

Note 1. External pattern input is shared with ENCAn input.

35.2 Overview

35.2.1 Functional Overview

The TSG3n is an 18-bit timer counter with various motor control functions.

- Count clock resolution: Minimum 12.5 ns (count clock = 80 MHz)
- Operating mode corresponding to various motor control methods
- Compare registers with reload buffer
- 10-bit dead time counter
  - Dead time counter with reload buffer
  - Independent dead time can be set for positive to negative phase change and negative to positive phase change.
- A/D conversion trigger signal generation
  - Three A/D conversion trigger signals can be generated by the compare registers TSG3nDCMP0E, TSG3nDCMP1E, and TSG3nDCMP2E.
  - Skipping function of A/D conversion trigger signals TSG3nADTRG0 and TSG3nADTRG1 can be set independently. The skipping ratio can be selected among 1/1, 1/2, 1/4, and 1/8.
  - The dedicated pin (TSG3nO7) can be used to output the toggle or diagnostic signal set by the TSG3nADTRG0 signal and reset by the TSG3nADTRG1 signal.
- Interrupt skipping
  - Skipping rate: 1/1 to 1/32
- Forced output stop function
  - Using the timer option (TAPA) function allows high-impedance control of the TSG3nO1 to TSG3nO6 pin outputs.
- Compare value setting
  - Reload (simultaneous rewrite) or anytime rewrite can be selected.
- Reload mode
  - Writing to TSG3nCMP1E (setting the reload request flag (TSG3nRSF)) enables reload, and allows simultaneous transfer of values for multiple registers.
  - Data can be transferred at peak/valley/peak or valley reload timing
  - Reload request flag (TSG3nRSF)
  - Register address assignment allowing DMA transfer
  - Reload skipping
- HT-PWM mode
  - 0 to 100% PWM duty cycle output is possible (with possible dead time reduction).
  - The LSB in the compare register can be used to append an additional pulse to the PWM output during count up, thus improving the output resolution without extra load on software.

- 120-DC control
  - Semi-automatic cruise function (trigger signal can be generated by an offset in conjunction with two-phase encoder, three-phase encoder, or ENCA).
- Three-phase encoder function (hall sensor signals can be input).
- Active level of the output pins TSG3nO1 to TSG3nO6 can be set individually.
- Fail-safe function (warning interrupt or error interrupt can be generated)
  - Simultaneous active output detect function for positive and inverse phases.
  - Abnormal input detection function of the three-phase encoder

NOTE

In this section, active level is indicated as high level.

35.2.2 Block Diagram

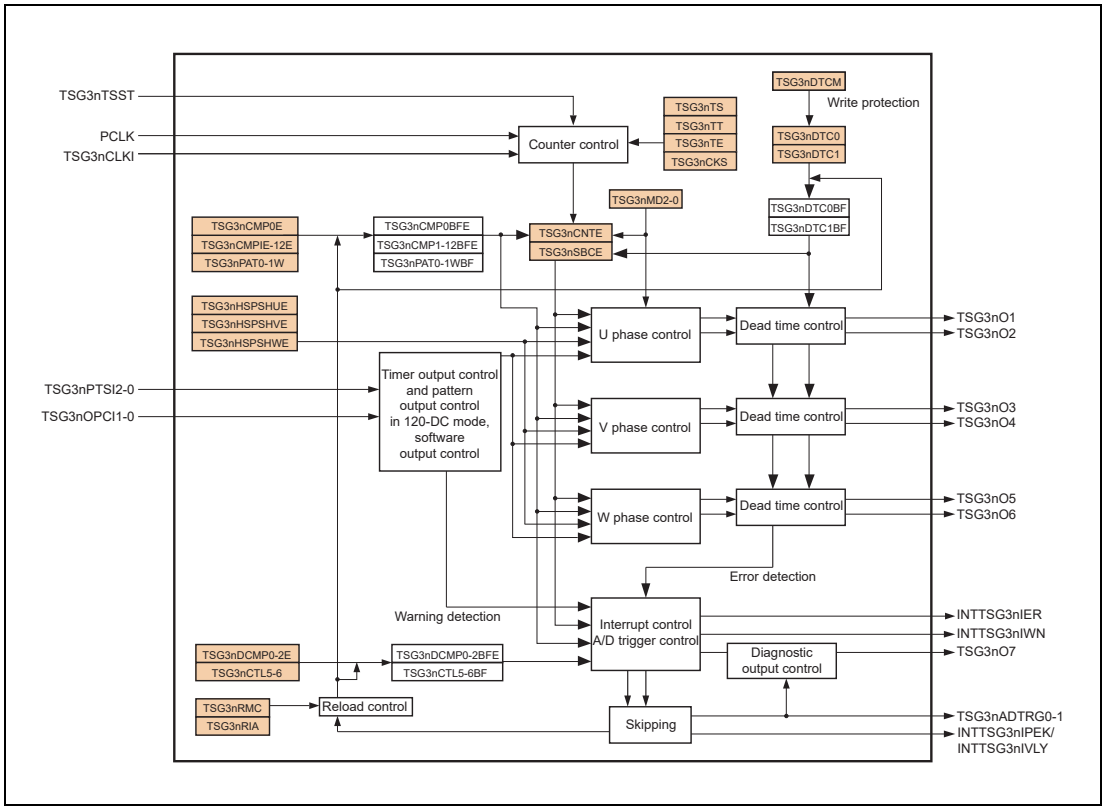


Figure 35.1 TSG3n Block Diagram

- TSG3nTSST: Simultaneous start trigger (input from PIC1)

35.3 Registers

35.3.1 List of Registers

TSG3n registers are listed in the following table.

For <TSG3n_base>, see **Section 35.1.2, Register Base Addresses**.

Table 35.8 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Reload	Access Protection	
					PBG	Other
TSG3n	TSG3n control register 0	TSG3nCTL0	<TSG3n_base> + 208_H	Disabled	*1	—
TSG3n	TSG3n control register 1	TSG3nCTL1	<TSG3n_base> + $20C_H$	Disabled	*1	—
TSG3n	TSG3n control register 2	TSG3nCTL2	<TSG3n_base> + 078_H	Enabled	*1	—
TSG3n	TSG3n control register 3	TSG3nCTL3	<TSG3n_base> + 004_H	Disabled	*1	—
TSG3n	TSG3n control register 4	TSG3nCTL4	<TSG3n_base> + $07C_H$	Enabled	*1	—
TSG3n	TSG3n control register 5	TSG3nCTL5	<TSG3n_base> + 008_H	Disabled	*1	—
TSG3n	TSG3n control register 6	TSG3nCTL6	<TSG3n_base> + $00C_H$	Disabled	*1	—
TSG3n	TSG3n control register 7	TSG3nCTL7	<TSG3n_base> + 218_H	Disabled	*1	—
TSG3n	TSG3n control register 8	TSG3nCTL8	<TSG3n_base> + $21C_H$	Disabled	*1	—
TSG3n	TSG3n I/O control register 0	TSG3nIOC0	<TSG3n_base> + 200_H	Disabled	*1	—
TSG3n	TSG3n I/O control register 1	TSG3nIOC1	<TSG3n_base> + 204_H	Disabled	*1	—
TSG3n	TSG3n I/O control register 2	TSG3nIOC2	<TSG3n_base> + 000_H	Disabled	*1	—
TSG3n	TSG3n I/O control register 3	TSG3nIOC3	<TSG3n_base> + 074_H	Enabled	*1	—
TSG3n	TSG3n status register 0	TSG3nSTR0	<TSG3n_base> + 010_H	Disabled	*1	—
TSG3n	TSG3n status register 1	TSG3nSTR1	<TSG3n_base> + 014_H	Disabled	*1	—
TSG3n	TSG3n status register 2	TSG3nSTR2	<TSG3n_base> + 018_H	Disabled	*1	—
TSG3n	TSG3n status clear trigger register	TSG3nSTC	<TSG3n_base> + $01C_H$	Disabled	*1	—
TSG3n	TSG3n option register 0	TSG3nOPT0	<TSG3n_base> + 020_H	Disabled	*1	—
TSG3n	TSG3n option register 1	TSG3nOPT1	<TSG3n_base> + 024_H	Disabled	*1	—
TSG3n	TSG3n trigger register 0	TSG3nTRG0	<TSG3n_base> + 030_H	Disabled	*1	—
TSG3n	TSG3n trigger register 1	TSG3nTRG1	<TSG3n_base> + 034_H	Disabled	*1	—
TSG3n	TSG3n trigger register 2	TSG3nTRG2	<TSG3n_base> + 038_H	Disabled	*1	—
TSG3n	TSG3n counter read buffer register	TSG3nCNT	<TSG3n_base> + 028_H	Disabled	*1	—
TSG3n	TSG3n bit extended counter read buffer register	TSG3nCNTE	<TSG3n_base> + $1A0_H$	Disabled	*1	—
TSG3n	TSG3n sub-counter read buffer register	TSG3nSBC	<TSG3n_base> + $02C_H$	Disabled	*1	—
TSG3n	TSG3n bit extended sub-counter read buffer register	TSG3nSBCE	<TSG3n_base> + $1A4_H$	Disabled	*1	—
TSG3n	TSG3n compare register 0	TSG3nCMP0	<TSG3n_base> + 058_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 0	TSG3nCMP0E	<TSG3n_base> + $14C_H$	Enabled	*1	—
TSG3n	TSG3n compare register 1, 2	TSG3nCMP1W	<TSG3n_base> + 040_H	Enabled	*1	—
TSG3n	TSG3n compare register 5, 6	TSG3nCMP5W	<TSG3n_base> + 044_H	Enabled	*1	—
TSG3n	TSG3n compare register 9, 10	TSG3nCMP9W	<TSG3n_base> + 048_H	Enabled	*1	—
TSG3n	TSG3n compare register 3, 4	TSG3nCMP3W	<TSG3n_base> + $04C_H$	Enabled	*1	—
TSG3n	TSG3n compare register 7, 8	TSG3nCMP7W	<TSG3n_base> + 050_H	Enabled	*1	—
TSG3n	TSG3n compare register 11, 12	TSG3nCMP11W	<TSG3n_base> + 054_H	Enabled	*1	—
TSG3n	TSG3n compare register 1	TSG3nCMP1	<TSG3n_base> + 080_H	Enabled	*1	—
TSG3n	TSG3n compare register 2	TSG3nCMP2	<TSG3n_base> + 084_H	Enabled	*1	—

Table 35.8 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Reload	Access Protection	
					PBG	Other
TSG3n	TSG3n compare register 3	TSG3nCMP3	<TSG3n_base> + 098_H	Enabled	*1	—
TSG3n	TSG3n compare register 4	TSG3nCMP4	<TSG3n_base> + $09C_H$	Enabled	*1	—
TSG3n	TSG3n compare register 5	TSG3nCMP5	<TSG3n_base> + 088_H	Enabled	*1	—
TSG3n	TSG3n compare register 6	TSG3nCMP6	<TSG3n_base> + $08C_H$	Enabled	*1	—
TSG3n	TSG3n compare register 7	TSG3nCMP7	<TSG3n_base> + $0A0_H$	Enabled	*1	—
TSG3n	TSG3n compare register 8	TSG3nCMP8	<TSG3n_base> + $0A4_H$	Enabled	*1	—
TSG3n	TSG3n compare register 9	TSG3nCMP9	<TSG3n_base> + 090_H	Enabled	*1	—
TSG3n	TSG3n compare register 10	TSG3nCMP10	<TSG3n_base> + 094_H	Enabled	*1	—
TSG3n	TSG3n compare register 11	TSG3nCMP11	<TSG3n_base> + $0A8_H$	Enabled	*1	—
TSG3n	TSG3n compare register 12	TSG3nCMP12	<TSG3n_base> + $0AC_H$	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 1	TSG3nCMP1E	<TSG3n_base> + $17C_H$	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 2	TSG3nCMP2E	<TSG3n_base> + 178_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 3	TSG3nCMP3E	<TSG3n_base> + 164_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 4	TSG3nCMP4E	<TSG3n_base> + 160_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 5	TSG3nCMP5E	<TSG3n_base> + 174_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 6	TSG3nCMP6E	<TSG3n_base> +170_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 7	TSG3nCMP7E	<TSG3n_base> + $15C_H$	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 8	TSG3nCMP8E	<TSG3n_base> + 158_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 9	TSG3nCMP9E	<TSG3n_base> + $16C_H$	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 10	TSG3nCMP10E	<TSG3n_base> + 168_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 11	TSG3nCMP11E	<TSG3n_base> + 154_H	Enabled	*1	—
TSG3n	TSG3n bit extended compare register 12	TSG3nCMP12E	<TSG3n_base> + 150_H	Enabled	*1	—
TSG3n	TSG3n diagnostic output compare register 0, 1	TSG3nDCMP0W	<TSG3n_base> + $05C_H$	Enabled	*1	—
TSG3n	TSG3n diagnostic output compare register 2	TSG3nDCMP2	<TSG3n_base> + 060_H	Enabled	*1	—
TSG3n	TSG3n bit extended diagnostic output compare register 0	TSG3nDCMP0E	<TSG3n_base> + 148_H	Enabled	*1	—
TSG3n	TSG3n bit extended diagnostic output compare register 1	TSG3nDCMP1E	<TSG3n_base> + 144_H	Enabled	*1	—
TSG3n	TSG3n bit extended diagnostic output compare register 2	TSG3nDCMP2E	<TSG3n_base> + 140_H	Enabled	*1	—
TSG3n	TSG3n pattern register 0	TSG3nPAT0W	<TSG3n_base> + 064_H	Enabled	*1	—
TSG3n	TSG3n pattern register 1	TSG3nPAT1W	<TSG3n_base> + 068_H	Enabled	*1	—
TSG3n	TSG3n dead time control register 0	TSG3nDTC0W	<TSG3n_base> + $06C_H$	Enabled	*1	—
TSG3n	TSG3n dead time control register 1	TSG3nDTC1W	<TSG3n_base> + 070_H	Enabled	*1	—
TSG3n	TSG3n HT-PWM U phase compare register	TSG3nCMPU	<TSG3n_base> + $0B0_H$	Enabled	*1	—
TSG3n	TSG3n HT-PWM V phase compare register	TSG3nCMPV	<TSG3n_base> + $0B4_H$	Enabled	*1	—
TSG3n	TSG3n HT-PWM W phase compare register	TSG3nCMPW	<TSG3n_base> + $0B8_H$	Enabled	*1	—
TSG3n	TSG3n bit extended HT-PWM U phase compare register	TSG3nCMPUE	<TSG3n_base> + 188_H	Enabled	*1	—
TSG3n	TSG3n bit extended HT-PWM V phase compare register	TSG3nCMPVE	<TSG3n_base> + 184_H	Enabled	*1	—
TSG3n	TSG3n bit extended HT-PWM W phase compare register	TSG3nCMPWE	<TSG3n_base> + 180_H	Enabled	*1	—
TSG3n	TSG3n SP-PWM U phase active width register	TSG3nUPW	<TSG3n_base> + $0BC_H$	Enabled	*1	—

Table 35.8 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Reload	Access Protection	
					PBG	Other
TSG3n	TSG3n SP-PWM V phase active width register	TSG3nVPW	<TSG3n_base> + $0C0_H$	Enabled	*1	—
TSG3n	TSG3n SP-PWM W phase active width register	TSG3nWPW	<TSG3n_base> + $0C4_H$	Enabled	*1	—
TSG3n	TSG3n bit extended SP-PWM U phase active width register	TSG3nUPWE	<TSG3n_base> + 198_H	Enabled	*1	—
TSG3n	TSG3n bit extended SP-PWM V phase active width register	TSG3nVPWE	<TSG3n_base> + 194_H	Enabled	*1	—
TSG3n	TSG3n bit extended SP-PWM W phase active width register	TSG3nWPWE	<TSG3n_base> + 190_H	Enabled	*1	—
TSG3n	TSG3n HSP-PWM W phase shift register	TSG3nHSPSHWE	<TSG3n_base> + 120_H	Enabled	*1	—
TSG3n	TSG3n HSP-PWM V phase shift register	TSG3nHSPSHVE	<TSG3n_base> + 124_H	Enabled	*1	—
TSG3n	TSG3n HSP-PWM U phase shift register	TSG3nHSPSHUE	<TSG3n_base> + 128_H	Enabled	*1	—
TSG3n	TSG3n HSP-PWM W phase compare register	TSG3nHSPCMWE	<TSG3n_base> + $12C_H$	Enabled	*1	—
TSG3n	TSG3n HSP-PWM V phase compare register	TSG3nHSPCMVE	<TSG3n_base> + 130_H	Enabled	*1	—
TSG3n	TSG3n HSP-PWM U phase compare register	TSG3nHSPCMUE	<TSG3n_base> + 134_H	Enabled	*1	—
TSG3n	TSG3n dead time protection register	TSG3nDTPR	<TSG3n_base> + 210_H	Disabled	*1	—

Note 1. n = 0: PBG51#6
n = 1: PBG51#7

35.3.2 TSG3nCTL0 — TSG3n Control Register 0

This register specifies the pulse width for the diagnostic output and operating mode of the TSG3n.

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> + 208_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TSG3nDWD	—	TSG3nMD[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W

Table 35.9 TSG3nCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TSG3nDWD	Selects the pulse width for the diagnostic output. 0: The output pulse width is set to 8 clocks. 1: The output pulse width is set to 16 clocks. The setting of this bit is valid when diagnostic output is enabled (TSG3nIOC1.TSG3nTGS = 1).
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	TSG3nMD[2:0]	Selects timer mode

TSG3n MD2	TSG3n MD1	TSG3n MD0	Timer Mode
0	0	0	PWM mode
0	0	1	HT-PWM mode (HT-PWM)
0	1	0	Shift pulse PWM mode (SP-PWM)
0	1	1	120-DC mode
1	0	0	High-accuracy shift pulse PWM mode (HSP-PWM)
Other than above			Setting prohibited

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer

35.3.3 TSG3nCTL1 — TSG3n Control Register 1

This register controls the flags of TSG3n.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $20C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3nTBA2	TSG3nTBA1	TSG3nTBA0	TSG3nPPC	TSG3nPEC	TSG3nTDC	TSG3nNDC	TSG3nPRC	TSG3nPTC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.10 TSG3nCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TSG3nTBA2	Enables or disables detection of the simultaneous active states of the TSG3nO5 and TSG3nO6 pins. 0: Disables detection of simultaneous active states of the TSG3nO5 and TSG3nO6 pins. 1: Enables detection of simultaneous active states of the TSG3nO5 and TSG3nO6 pins. If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA2 bits are 1, the positive phase and inverse phase simultaneous active state detection flag 2 (TSG3nTBF2) is set to 1, and an error interrupt (INTTSG3nIER) is generated.
8	TSG3nTBA1	Enables or disables detection of the simultaneous active states of the TSG3nO3 and TSG3nO4 pins. 0: Disables detection of simultaneous active states of the TSG3nO3 and TSG3nO4 pins. 1: Enables detection of simultaneous active states of the TSG3nO3 and TSG3nO4 pins. If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA1bits are 1, the positive phase and inverse phase simultaneous active state detection flag 1 (TSG3nTBF1) is set to 1, and an error interrupt (INTTSG3nIER) is generated.
7	TSG3nTBA0	Enables or disables detection of the simultaneous active states of the TSG3nO1 and TSG3nO2 pins. 0: Disables detection of simultaneous active states of the TSG3nO1 and TSG3nO2 pins. 1: Enables detection of simultaneous active states of the TSG3nO1 and TSG3nO2 pins. If the simultaneous active state is detected when the TSG3nIOC1.TSG3nEOC and TSG3nTBA0 bits are 1, the positive phase and inverse phase simultaneous active state detection flag 0 (TSG3nTBF0) is set to 1, and an error interrupt (INTTSG3nIER) is generated.
6	TSG3nPPC	Enables or disables detection of the pattern phase difference (TSG3nSTR2.TSG3nPPF) between the TSG3nPTSI2-0 and TSG3nOPF2-0. 0: Disables detection of I/O pattern difference. 1: Enables detection of I/O pattern difference
5	TSG3nPEC	Enables or disables detection of the pattern error (TSG3nSTR2.TSG3nPEF) of the TSG3nPTSI2-0 pins. 0: Disables detection of the pattern error of the TSG3nPTSI2-0 pins. 1: Enables detection of the pattern error of the TSG3nPTSI2-0 pins.

Table 35.10 TSG3nCTL1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4</td><td>TSG3nTDC</td><td>Enables or disables detection of the simultaneous trigger (TSG3nSTR2.TSG3nTDF) of the TSG3nOPCI0 and TSG3nOPCI1.
0: Disables detection of the simultaneous trigger of the TSG3nOPCI0 and TSG3nOPCI1.
1: Enables detection of the simultaneous trigger of the TSG3nOPCI0 and TSG3nOPCI1.</td></tr>
<tr><td>3</td><td>TSG3nNDC</td><td>Enables or disables detection of the noise generation (two or more pins change simultaneously) (TSG3nSTR2.TSG3nNDF) on the TSG3nPTSI2-0 pins.
0: Disables detection of the noise generation on the TSG3nPTSI2-0 pins.
1: Enables detection of the noise generation on the TSG3nPTSI2-0 pins.</td></tr>
<tr><td>2</td><td>TSG3nPRC</td><td>Enables or disables detection of the reversal of the pattern (TSG3nSTR2.TSG3nPRF) of the TSG3nPTSI2-0 pins.
0: Disables detection of the reversal of the pattern of the TSG3nPTSI2-0 pins.
1: Enables detection of the reversal of the pattern of the TSG3nPTSI2-0 pins.</td></tr>
<tr><td>1, 0</td><td>TSG3nPTC[1:0]</td><td>Enables or disables detection of an abnormal toggle (TSG3nSTR2.TSG3nPTF) of the TSG3nPTSI2-0 pins between TSG3nOPCI1 and TSG3nOPCI0 triggers
<table>
<tr><th>TSG3n PTC1</th><th>TSG3n PTC0</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Disables detection of an abnormal toggle of the TSG3nPTSI2-0 pins.</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins.</td></tr>
<tr><td>1</td><td>1</td><td>Enables detection of an abnormal toggle of the TSG3nPTSI2-0 pins. When an abnormal toggle is detected, the pattern output switch trigger is automatically switched from trigger switch to pattern switch (TSG3nPOT is switched from 1 to 0.)</td></tr>
</table>
</td></tr>
</table>

CAUTIONS

1. If TSG3nDTC0 or TSG3nDTC1 is set to 0000_H (without dead time), the TSG3nTBA2-0 bits should be set to 0.
2. This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer

35.3.4 TSG3nCTL2 — TSG3n Control Register 2

This register selects a count clock for TSG3n.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 078_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCKS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 35.11 TSG3nCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSG3nCKS	Selects a count clock 0: Selects PCLK as a count clock. 1: Counted by PCLK when clock enable input (TSG3nCLKI) is high level.

CAUTION

Set the TSG3nCTL2.TSG3nCKS bit to 0 in HT-PWM mode and HSP-PWM mode.

35.3.5 TSG3nCTL3 — TSG3n Control Register 3

This register selects the rewrite method of the compare registers.

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> + 004_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3nRIA	TSG3nRMC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 35.12 TSG3nCTL3 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TSG3nRIA	Selects the reload timing of the compare register values. 0: The reload timing is set to peak reload timing (set by TSG3nCTL4.TSG3nPRE) and valley reload timing (set by TSG3nCTL4.TSG3nVRE). 1: The reload timing is set to peak interrupt timing and valley interrupt timing. The setting of this bit is valid in reload mode (TSG3nRMC = 0).
0	TSG3nRMC	Selects the transfer timing of the compare register values. 0: Reload mode (simultaneous rewrite) Writing to registers to be reloaded enables reloading and the register values are rewritten simultaneously at the next reload timing. Writing to any register other than registers to be reloaded does not enable reloading. For the register to be reloaded, see **Section 35.3.1, List of Registers**. 1: Anytime rewrite mode The compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected immediately. TSG3nRSF is cleared. Do not set TSG3nRMC to 1 when operated in 120-DC mode or in HSP-PWM mode.

35.3.6 TSG3nCTL4 — TSG3n Control Register 4

This register enables or disables generation of a peak interrupt and a valley interrupt, and the reload timing.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + $07C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSG3n PRE	TSG3n VRE	TSG3n PIE	TSG3n VIE	TSG3nRCC[04:00]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.13 TSG3nCTL4 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TSG3nPRE	Enables or disables the peak reload timing. 0: Disables reload operation at the peak timing of the 18-bit counter. 1: Enables reload operation at the peak timing of the 18-bit counter. • The peak reload timing means the peak timing of the 18-bit counter in HT-PWM mode and the clear timing of the 18-bit counter by compare match in any mode other than HT-PWM mode. • When the reload operation at the peak timing of the 18-bit counter is disabled (TSG3nPRE = 0), reload is not executed in any mode other than HT-PWM mode.
7	TSG3nVRE	Enables or disables the valley reload timing. 0: Disables reload operation at the valley timing of the 18-bit counter. 1: Enables reload operation at the valley timing of the 18-bit counter. The setting of this bit is valid only in HT-PWM mode.
6	TSG3nPIE	Enables or disables generation of a peak interrupt (INTTSG3nIPEK). 0: Disables generation of a peak interrupt (INTTSG3nIPEK) at the peak timing of the 18-bit counter. Interrupts are not skipped. 1: Enables generation of a peak interrupt (INTTSG3nIPEK) at the peak timing of the 18-bit counter. Interrupts are skipped.
5	TSG3nVIE	Enables or disables generation of a valley interrupt (INTTSG3nIVLY). 0: Disables generation of a valley interrupt (INTTSG3nIVLY) at the valley timing of the 18-bit counter. Interrupts are not skipped. 1: Enables generation of a valley interrupt (INTTSG3nIVLY) at the valley timing of the 18-bit counter. Interrupts are skipped. The setting of this bit is valid only in HT-PWM mode.

Table 35.13 TSG3nCTL4 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TSG3nRCC [04:00]</td><td>Specifies the skipping rate of the interrupts (INTTSG3nIPEK and INTTSG3nIVLY) and reload.
<table>
<tr><th>TSG3n RCC04</th><th>TSG3n RCC03</th><th>TSG3n RCC02</th><th>TSG3n RCC01</th><th>TSG3n RCC00</th><th>Skipping Rate</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Skipping Disabled</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1/2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1/3</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1/4</td></tr>
<tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1/30</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1/31</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1/32</td></tr>
</table>
When a write access is made (including a write of the same value to TSG3nRCC04-TSG3nRCC00) to TSG3nCTL4 during timer operation (TSG3nSTR0.TSG3nTE = 1), the interrupt skipping counter is cleared</td></tr>
</table>

35.3.7 TSG3nCTL5 — TSG3n Control Register5

This register controls A/D conversion trigger output (TSG3nADTRG0).

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> +008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	TSG3nACC [01:00]		TSG3n AT09	TSG3n AT08	TSG3n AT07	TSG3n AT06	TSG3n AT05	TSG3n AT04	TSG3n AT03	TSG3n AT02	TSG3n AT01	TSG3n AT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.14 TSG3nCTL5 Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11, 10	TSG3nACC [01:00]	Specifies the skipping rate of the A/D conversion trigger (TSG3nADTRG0). **TSG3nACC01 / TSG3nACC00 / Skipping Rate** 0 / 0 / Skipping Disabled 0 / 1 / 1/2 1 / 0 / 1/4 1 / 1 / 1/8 When a write access is made (including a write of the same value to TSG3nACC01 and TSG3nACC00) to TSG3nCTL5 during timer operation (TSG3nSTR0.TSG3nTE = 1), the interrupt skipping counter is cleared.
9	TSG3nAT09	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the (peak) timing when the 18-bit sub-counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. • The TSG3nAT09 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT09 bit should be set to 0. • Do not set the TSG3nAT09 bit to 1 when TSG3nDTC0W is not 0000_H and TSG3nDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 18-bit sub-counter even if set so.
8	TSG3nAT08	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the (valley) timing when the 18-bit sub-counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. • The TSG3nAT08 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT08 bit should be set to 0. • Do not set the TSG3nAT08 bit to 1 when TSG3nDTC0W is 0000_H and TSG3nDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 18-bit sub-counter even if set so.

Table 35.14 TSG3nCTL5 Register Contents (2/3)

Bit Position	Bit Name	Function
7	TSG3nAT07	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
6	TSG3nAT06	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value
5	TSG3nAT05	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
4	TSG3nAT04	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1Evalue. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.
3	TSG3nAT03	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.

Table 35.14 TSG3nCTL5 Register Contents (3/3)

Bit Position	Bit Name	Function
2	TSG3nAT02	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value. 1: Enables generation of the A/D conversion trigger at the timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E.
1	TSG3nAT01	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the timing (peak interrupt) when the 18-bit counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.
0	TSG3nAT00	Specifies generation of A/D conversion trigger (TSG3nADTRG0) at the timing (valley interrupt) when the 18-bit counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.

CAUTION

TSG3nATxx register bits should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0).
Only the same value with current setting can be written during timer operation (TSG3nSTR0.TSG3nTE = 1).
If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed.

35.3.8 TSG3nCTL6 — TSG3n Control Register 6

This register controls the A/D conversion trigger output (TSG3nADTRG1).

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $00C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	TSG3nACC [11:10]		TSG3n AT19	TSG3n AT18	TSG3n AT17	TSG3n AT16	TSG3n AT15	TSG3n AT14	TSG3n AT13	TSG3n AT12	TSG3n AT11	TSG3n AT10
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.15 TSG3nCTL6 Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11, 10	TSG3nACC [11:10]	Specifies the skipping rate of the A/D conversion trigger (TSG3nADTRG1). **TSG3nACC11 / TSG3nACC10 / Skipping Rate** 0 / 0 / Skipping Disabled 0 / 1 / 1/2 1 / 0 / 1/4 1 / 1 / 1/8 When a write access is made (including a write of the same value to TSG3nACC11 and TSG3nACC10) to TSG3nCTL6 during timer operation (TSG3nSTR0.TSG3nTE = 1), the skipping counter is cleared.
9	TSG3nAT19	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the (peak) timing when the 18-bit sub-counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the peak timing of the 18-bit sub-counter. • The TSG3nAT19 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT19 bit should be set to 0. • Do not set the TSG3nAT19 bit to 1 when TSG3nDTC0W is not 0000_H and TSG3nDTC1W is 0000_H. A/D conversion trigger is not generated at the peak timing of the 18-bit sub-counter even if set so.
8	TSG3nAT18	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the (valley) timing when the 18-bit sub-counter switches from decrementing to incrementing. 0: Disables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. 1: Enables generation of the A/D conversion trigger at the valley timing of the 18-bit sub-counter. • The TSG3nAT18 bit can be set to 1 only in HT-PWM mode. In other modes, the TSG3nAT18 bit should be set to 0. • Do not set the TSG3nAT18 bit to 1 when TSG3nDTC0W is 0000_H and TSG3nDTC1W is not 0000_H. A/D conversion trigger is not generated at the valley timing of the 18-bit sub-counter even if set so.

Table 35.15 TSG3nCTL6 Register Contents (2/3)

Bit Position	Bit Name	Function
7	TSG3nAT17	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP2E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
6	TSG3nAT16	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP2E value.
5	TSG3nAT15	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP1E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.
4	TSG3nAT14	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP1E value.
3	TSG3nAT13	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during decrementation with the TSG3nDCMP0E value. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit should be set to 0.

Table 35.15 TSG3nCTL6 Register Contents (3/3)

Bit Position	Bit Name	Function
2	TSG3nAT12	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E. 0: Disables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value. 1: Enables generation of the A/D conversion trigger at the match timing of the 18-bit counter value during incrementation with the TSG3nDCMP0E value.
1	TSG3nAT11	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the timing (peak interrupt) when the 18-bit counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a peak interrupt (INTTSG3nIPEK) after being skipped.
0	TSG3nAT10	Specifies generation of A/D conversion trigger (TSG3nADTRG1) at the timing (valley interrupt) when the 18-bit counter switches from incrementing to decrementing. 0: Disables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped. 1: Enables generation of the A/D conversion trigger at the timing of a valley interrupt (INTTSG3nIVLY) after being skipped. This bit can be set to 1 only in HT-PWM mode. In other modes, this bit must be set to 0.

CAUTION

TSG3nATxx register bits should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0).
Only the same value with current setting can be written during timer operation (TSG3nSTR0.TSG3nTE = 1).
If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed.

35.3.9 TSG3nCTL7 — TSG3n Control Register 7

This register sets the level of PWM output from the TSG3O1 to TSG3O6 pins at operation start (TSG3nTE is changed from 0 to 1) and at operation restart in SP-PWM mode.

This register can be written only when SP-PWM mode is selected (TSG3nMD2 to TSG3nMD0 = 010) and the timer is stopped (TSG3nTE = 0). Do not rewrite this register in other modes (PWM mode, HT-PWM mode, 120-DC mode, HSP-PWM mode) or while the timer is operating (TSG3nTE = 1).

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> + 218_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TSG3n SPSTL2	TSG3n SPSTL1	TSG3n SPSTL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 35.16 TSG3nCTL7 Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TSG3nSPSTL2	SP-PWM Mode Start Level Control Bit 2 0: TSG3nO5 (W phase) is cleared and TSG3nO6 (WB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO5 (W phase) is set and TSG3nO6 (WB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode.
1	TSG3nSPSTL1	SP-PWM Mode Start Level Control Bit 1 0: TSG3nO3 (V phase) is cleared and TSG3nO4 (VB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO3 (V phase) is set and TSG3nO4 (VB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode.
0	TSG3nSPSTL0	SP-PWM Mode Start Level Control Bit 0 0: TSG3nO1 (U phase) is cleared and TSG3nO2 (UB phase) is set at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode. 1: TSG3nO3 (U phase) is set and TSG3nO1 (UB phase) is cleared at operation start (TSG3nTE changed from 0 to 1) or restart in SP-PWM mode.

NOTE

The settings of bits TSG3nSPSTL2 to TSG3nSPSTL0 affect output on the TSG3nO1 to TSG3nO6 pins when operation starts or is restarted. The set dead time is always inserted at these times.

35.3.10 TSG3nCTL8 — TSG3n Control Register 8

This register specifies timer output timing when input patterns are changed in 120-DC mode.

This register can be written only when 120-DC mode is selected (TSG3nMD2-0 = 011) and the timer is stopped (TSG3nTE = 0).

Do not rewrite this register in other modes (PWM mode, SP-PMW mode, HT-PWM mode, HSP-PWM mode) or while the timer is operating (TSG3nTE = 1).

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> +$21C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSG3n S120DCO
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 35.17 TSG3nCTL8 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TSG3n S120DCO	120-DC Mode Control Bit 0 0: When the input patterns are changed while 120-DC mode is selected, the main counter (TSG3nCNTE) is cleared and the change of input patterns is immediately reflected to timer output. 1: When the input patterns are changed while 120-DC mode is selected, the change of input patterns is reflected to timer output after a match of the main counter (TSG3nCNTE) with TSG3nCMP0E (from the next timer period).

CAUTION

When TSG3nS120DCO is set to 1 in 120DC mode, set the TSG3nOPT0.TSG3nSOC bit to 0.

The settings of the TSG3nOPT0.TSG3nSTE and TSG3nOPT0.TSG3nPOT bits must not be changed while the timer is operating (TSG3nSTR0.TSG3nTE = 1).

35.3.11 TSG3nIOC0 — TSG3n I/O Control Register0

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> +200_H

Value after reset: $7E_H$

Bit	7	6	5	4	3	2	1	0
	—	TSG3nTOE6	TSG3nTOE5	TSG3nTOE4	TSG3nTOE3	TSG3nTOE2	TSG3nTOE1	—
Value after reset	0	1	1	1	1	1	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 35.18 TSG3nIOC0 Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 1	TSG3nTOE6 to TSG3nTOE1	Enables or disables controlling of TSG3nO6 to TSG3nO1 by TSG3nIOC2. When these bits are 1, rewriting of TSG3nIOC2 is ignored. 0: Controlling is enabled. 1: Controlling is disabled.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

35.3.12 TSG3nIOC1 — TSG3n I/O Control Register1

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> + 204_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TSG3nPTS	TSG3nEOC	TSG3nWOC	TSG3nTGS	TSG3nTOS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 35.19 TSG3nIOC1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TSG3nPTS	Enables or disables output of the edge detection signal (TSG3nPTE) of TSG3nPTSI0 to TSG3nPTSI2 and two-phase encoder count signal (TSG3nPEC). 0: Disables output of the toggle signal by edge detection of TSG3nPTSI0 to TSG3nPTSI2. 1: Enables output of the toggle signal by edge detection of TSG3nPTSI0 to TSG3nPTSI2.
3	TSG3nEOC	Enables or disables detection of the error condition at the motor control. 0: Disables generation of an error interrupt (INTTSG3nIER). 1: Enables generation of an error interrupt (INTTSG3nIER). For details on controlling the error interrupt, see **Section 35.4.6.1, Error Interrupt Function**.
2	TSG3nWOC	Enables or disables detection of the warning condition at the motor control. 0: Disables generation of a warning interrupt (INTTSG3nIWN). 1: Enables generation of a warning interrupt (INTTSG3nIWN). For details on the controlling generation of warning interrupt, see **Section 35.4.6.2, Warning Interrupt Function**.
1	TSG3nTGS	Selects the A/D conversion trigger diagnostic output (TSG3nO7) signal. 0: Selects A/D conversion trigger output. 1: Selects diagnostic output.
0	TSG3nTOS	Selects the timer counter increment/decrement status output (TSG3nO0) signal. 0: Outputs the up/down count flag of the 18-bit counter. 1: Outputs the up/down count flag of the 18-bit sub-counter. • When TSG3nTOS is 0, the status of TSG3nSTR0.TSG3nCUF is output to TSG3nO0. When TSG3nTOS is 1, the status of TSG3nSTR0.TSG3nSUF is output to TSG3nO0. • The setting of this bit is valid only in HT-PWM mode.

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

35.3.13 TSG3nIOC2 — TSG3n I/O Control Register2

This register controls the timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + 000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TSG3nOL6	TSG3nOL5	TSG3nOL4	TSG3nOL3	TSG3nOL2	TSG3nOL1	—	—	TSG3nTO6	TSG3nTO5	TSG3nTO4	TSG3nTO3	TSG3nTO2	TSG3nTO1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 35.20 TSG3nIOC2 Register Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 9	TSG3nOL6 to TSG3nOL1	Specifies the active level of TSG3nO6 to TSG3nO1 outputs. 0: Active level is high level 1: Active level is low level
8 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 1	TSG3nTO6 to TSG3nTO1	Specifies the latch level of the output buffer of the TSG3nO6 to TSG3nO1. 0: Latch level of output buffer is low level 1: Latch level of output buffer is high level
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When the counting is stopped (TSG3nSTR0.TSG3nTE = 0), the TSG3nO1-6 pins maintain their previous output states. The output level should be changed by setting the TSG3nIOC0.TSG3nTOEm bit to 0, using the TSG3nTOm bit. This register can be rewritten when TSG3nIOC0.TSG3nTOEm = 0 (m = 1 to 6).

NOTE

While the timer is stopped (TSG3nSTR0.TSG3nTE = 0) and control of TSG3nOm by rewriting TSG3nIOC2 is enabled (TSG3nIOC0.TSG3nTOEm = 0), TSG3nOLm and TSG3nTOm of TSG3nIOC2 select the output level for the corresponding TSG3nOm output as listed in the table below.

TSG3nOLm	TSG3nTOm	Output level of TSG3nOm
0	0	Low level
0	1	High level
1	0	High level
1	1	Low level

35.3.14 TSG3nIOC3 — TSG3n I/O Control Register3

This register controls timer output pins (TSG3nO1 to TSG3nO6).

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 074_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TSG3n TOL6	TSG3n TOL5	TSG3n TOL4	TSG3n TOL3	TSG3n TOL2	TSG3n TOL1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 35.21 TSG3nIOC3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 1	TSG3nTOL6 to TSG3nTOL1	Controls the set/clear level of output. 0: Outputs the normal level. 1: Outputs the reversed level. Setting of this bit is reflected at the start of output. The change of the output level is reflected at the next compare match timing after the change.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

TSG3nTOL6 to TSG3nTOL1 should be set to 0 in HT-PWM mode and HSP-PWM mode.

35.3.15 TSG3nSTR0 — TSG3n Status Register 0

This register controls the flags.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TSG3n_base> + 010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TSG3nCUF	TSG3nSUF	TSG3nRSF	TSG3nTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.22 TSG3nSTR0 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	TSG3nCUF	Indicates the count direction of the 18-bit counter. 0: The 18-bit counter is incremented. 1: The 18-bit counter is decremented. TSG3nCUF is valid only in HT-PWM mode. In other modes, it is invalid (TSG3nCUF = 0).
2	TSG3nSUF	Indicates the count direction of the 18-bit sub-counter. 0: The 18-bit sub-counter is incremented. 1: The 18-bit sub-counter is decremented. • TSG3nSUF detects counting of the 18-bit sub-counter from 0000_H to (TSG3nCMP0E value - 0002_H) as up-counting, and counting from the TSG3nCMP0E value to 0002_H as down-counting. • This bit is valid only in HT-PWM mode.
1	TSG3nRSF	Indicates whether there is a reload request. 0: No reload request or reload has completed. 1: There is a reload request. • This bit is valid only in TSG3nRMC = 0. • This bit indicates that the data to be transferred next is held. • This bit is set to 1 by writing to registers to be reloaded, and cleared to 0 when reload has completed. • When TSG3nRMC is changed from 0 to 1 in HT-PWM mode, TSG3nRSF is cleared to 0. For registers to be reloaded, see **Section 35.3.1, List of Registers**.
0	TSG3nTE	Indicates the TSG3n operation status. 0: TSG3n is stopped. 1: TSG3n is operating. This bit is set when TSG3nTRG0.TSG3nTS = 1, and cleared when TSG3nTRG1.TSG3nTT = 1.

35.3.16 TSG3nSTR1 — TSG3n Status Register 1

This register controls the flags.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TSG3n_base> + 014_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TSG3nTSF	TSG3nOPF[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 35.23 TSG3nSTR1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	TSG3nTSF	Indicates the pattern change order of TSG3nPTSI0 to TSG3nPTSI2. 0: Indicates that patterns are input to TSG3nPTSI0 to TSG3nPTSI2 in the normal rotation pattern order 1: Indicates that patterns are input to TSG3nPTSI0 to TSG3nPTSI2 in the reverse rotation pattern order. Normal Rotation → Reverse Rotation ← TSG3nPTSI2-TSG3nPTSI0: [1,0,1] [1,0,0] [1,1,0] [0,1,0] [0,1,1] [0,0,1] Normal or reverse rotation can be detected from the first change of TSG3nPTSI0 to TSG3nPTSI2 after TSG3nTRG0.TSG3nTS has been set to 1. For details, see **Section 35.4.3.5, (b) Detection of Change in Input Pattern Order**.
2 to 0	TSG3nOPF [2:0]	Indicates the output pattern of the timer output pins (TSG3nO1 to TSG3nO6).

35.3.17 TSG3nSTR2 — TSG3n Status Register 2

This register controls the flags.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TSG3n_base> + 018_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3n TBF2	TSG3n TBF1	TSG3n TBF0	TSG3n PPF	TSG3n PEF	TSG3n TDF	TSG3n NDF	TSG3n PRF	TSG3n PTF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.24 TSG3nSTR2 Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned.
9	TSG3nTBF2	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA2 is 1. 0: Positive phase (TSG3nO5) and inverse phase (TSG3nO6) are not active simultaneously. 1: Positive phase (TSG3nO5) and inverse phase (TSG3nO6) are active simultaneously. • TSG3nTBF2 is set to 1 when the simultaneous active state of the positive phase (TSG3nO5) and inverse phase (TSG3nO6) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF2 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR2. • The simultaneous active state is not detected when TSG3nTBA2 = 0.
8	TSG3nTBF1	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA1 is 1. 0: Positive phase (TSG3nO3) and inverse phase (TSG3nO4) are not active simultaneously. 1: Positive phase (TSG3nO3) and inverse phase (TSG3nO4) are active simultaneously. • TSG3nTBF1 is set to 1 when the simultaneous active state of the positive phase (TSG3nO3) and inverse phase (TSG3nO4) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF1 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR1. • The simultaneous active state is not detected when TSG3nTBA1 = 0.
7	TSG3nTBF0	Indicates whether the simultaneous active state of the positive phase and inverse phase is detected when TSG3nCTL1.TSG3nTBA0 is 1. 0: Positive phase (TSG3nO1) and inverse phase (TSG3nO2) are not active simultaneously. 1: Positive phase (TSG3nO1) and inverse phase (TSG3nO2) are active simultaneously. • TSG3nTBF0 is set to 1 when the simultaneous active state of the positive phase (TSG3nO1) and inverse phase (TSG3nO2) is detected, and an error interrupt (INTTSG3nIER) is generated. TSG3nTBF0 can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start) or by writing 1 to TSG3nSTC.TSG3nTBR0. • The simultaneous active state is not detected when TSG3nTBA0 = 0.

Table 35.24 TSG3nSTR2 Register Contents (2/3)

Bit Position	Bit Name	Function
6	TSG3nPPF	Indicates detection of the difference between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6) after they are compared. 0: No phase difference detected between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6). 1: A phase difference detected between the input patterns (TSG3nPTSI0 to TSG3nPTSI2) and the output patterns (TSG3nO1 to TSG3nO6). • TSG3nPPF is set to 1 when a difference between input and output patterns is detected, and a warning interrupt (INTTSG3nIWN) is generated. This bit can be cleared either by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), setting TSG3nTRG0.TSG3nTS to 1 (timer start), input to TSG3nTSST (timer restart), or writing 1 to TSG3nSTC.TSG3nPPR.
5	TSG3nPEF	Indicates whether an abnormal input (000_B or 111_B is input to TSG3nPTSI0 to TSG3nPTSI2) is detected. 0: No abnormal input (000_B or 111_B) to TSG3nPTSI0 to TSG3nPTSI2 detected. 1: Abnormal input (000_B or 111_B) to TSG3nPTSI0 to TSG3nPTSI2 detected. TSG3nPEF is set to 1 when an input of 000_B or 111_B to TSG3nPTSI0 to TSG3nPTSI2 is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPEF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nPER. TSG3nPEF is valid when TSG3nCTL1.TSG3nPEC = 1.
4	TSG3nTDF	Indicates whether simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers is detected. 0: Simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers not detected. 1: Simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers detected. TSG3nTDF is set to 1 when simultaneous generation of the TSG3nOPCI0 and TSG3nOPCI1 triggers is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nTDF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nTDR. TSG3nTDF is valid when TSG3nCTL1.TSG3nTDC = 1.
3	TSG3nNDF	Indicates whether noise on TSG3nPTSI0 to TSG3nPTSI2 is detected. 0: Noise on TSG3nPTSI0 to TSG3nPTSI2 due to simultaneous change of two or more pins not detected. 1: Noise on TSG3nPTSI0 to TSG3nPTSI2 due to simultaneous change of two or more pins detected. TSG3nNDF is set to 1 when simultaneous change of two or more pins in TSG3nPTSI0 to TSG3nPTSI2 is detected, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nNDF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1(starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nNDR. TSG3nNDF is valid when TSG3nCTL1.TSG3nNDC = 1.

Table 35.24 TSG3nSTR2 Register Contents (3/3)

Bit Position	Bit Name	Function
2	TSG3nPRF	Indicates whether reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order is detected. 0: The reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order not detected. 1: The reversal of the TSG3nPTSI0 to TSG3nPTSI2 input order detected. TSG3nPRF is set to 1 when TSG3nSTR1.TSG3nTSF changes, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPRF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (timer operation start), by input to TSG3nTSST(timer restart), or by writing 1 to TSG3nSTC.TSG3nPRR. Detection is possible from the second TSG3nPTSI0 to TSG3nPTSI2 change timing after setting TSG3nTRG0.TSG3nTS = 1. TSG3nPRF is valid when TSG3nCTL1.TSG3nPRC = 1.
1	TSG3nPTF	Indicates whether an abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 is detected. 0: No abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 detected. 1: An abnormal toggle of TSG3nPTSI0 to TSG3nPTSI2 detected. TSG3nPTF is set to 1 when TSG3nPTSI0 to TSG3nPTSI2 (TSG3nPTE signal toggle) are changed three times or more during TSG3nOPCI0 trigger or TSG3nPTSI0 to TSG3nPTSI2 (TSG3nPTE signal toggle) are changed three times or more during TSG3nOPCI1 trigger, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPTF can be cleared by changing TSG3nSTR0.TSG3nTE from 0 to 1 (operation start), by setting TSG3nTRG0.TSG3nTS to 1 (starting timer operation), by input to TSG3nTSST (timer restart), or by writing 1 to TSG3nSTC.TSG3nPTR. TSG3nPTF is valid when TSG3nCTL1.TSG3nPTC[1:0] = 10_B or 11_B.
0	Reserved	When read, the value after reset is returned.

35.3.18 TSG3nSTC — TSG3n Status Clear Trigger Register

This register controls the flags.

Access: This register is a write-only register that can be written in 16-bit units.

Address: <TSG3n_base> + $01C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3n TBR2	TSG3n TBR1	TSG3n TBR0	TSG3n PPR	TSG3n PER	TSG3n TDR	TSG3n NDR	TSG3n PRR	TSG3n PTR	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W	W	W	W	W	W	W	W	R

Table 35.25 TSG3nSTC Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 10	Reserved	When writing, write the value after reset.
9	TSG3nTBR2	This is a trigger bit that clears TSG3nSTR2.TSG3nTBF2. 0: Does not clear TSG3nTBF2. 1: Clears TSG3nTBF2. When TSG3nTBR2 writing and TSG3nSTR2.TSG3nTBF2 setting occur simultaneously, TSG3nSTR2.TSG3nTBF2 setting has a priority, and the flag is not cleared.
8	TSG3nTBR1	This is a trigger bit that clears TSG3nSTR2.TSG3nTBF1. 0: Does not clear TSG3nTBF1. 1: Clears TSG3nTBF1. When TSG3nTBR1 writing and TSG3nSTR2.TSG3nTBF1 setting occur simultaneously, TSG3nSTR2.TSG3nTBF1 setting has a priority, and the flag is not cleared.
7	TSG3nTBR0	This is a trigger bit that clears TSG3nSTR2.TSG3nTBF0. 0: Does not clear TSG3nTBF0. 1: Clears TSG3nTBF0. When TSG3nTBR0 writing and TSG3nSTR2.TSG3nTBF0 setting occur simultaneously, TSG3nSTR2.TSG3nTBF0 setting has a priority, and the flag is not cleared.
6	TSG3nPPR	This is a trigger bit that clears TSG3nSTR2.TSG3nPPF. 0: Does not clear TSG3nPPF. 1: Clears TSG3nPPF. When TSG3nPPR writing and TSG3nSTR2.TSG3nPPF setting occur simultaneously, TSG3nSTR2.TSG3nPPF setting has a priority, and the flag is not cleared.
5	TSG3nPER	This is a trigger bit that clears TSG3nSTR2.TSG3nPEF. 0: Does not clear TSG3nPEF. 1: Clears TSG3nPEF. When TSG3nPER writing and TSG3nSTR2.TSG3nPEF setting occur simultaneously, TSG3nSTR2.TSG3nPEF setting has a priority, and the flag is not cleared.
4	TSG3nTDR	This is a trigger bit that clears TSG3nSTR2.TSG3nTDF. 0: Does not clear TSG3nTDF. 1: Clears TSG3nTDF. When TSG3nTDR writing and TSG3nSTR2.TSG3nTDF setting occur simultaneously, TSG3nSTR2.TSG3nTDF setting has a priority, and the flag is not cleared.

Table 35.25 TSG3nSTC Register Contents (2/2)

Bit Position	Bit Name	Function
3	TSG3nNDR	This is a trigger bit that clears TSG3nSTR2.TSG3nNDF. 0: Does not clear TSG3nNDF. 1: Clears TSG3nNDF. When TSG3nNDR writing and TSG3nSTR2.TSG3nNDF setting occur simultaneously, TSG3nSTR2.TSG3nNDF setting has a priority, and the flag is not cleared.
2	TSG3nPRR	This is a trigger bit that clears TSG3nSTR2.TSG3nPRF. 0: Does not clear TSG3nPRF. 1: Clears TSG3nPRF. When TSG3nPRR writing and TSG3nSTR2.TSG3nPRF setting occur simultaneously, TSG3nSTR2.TSG3nPRF setting has a priority, and the flag is not cleared.
1	TSG3nPTR	This is a trigger bit that clears TSG3nSTR2.TSG3nPTF. 0: Does not clear TSG3nPTF. 1: Clears TSG3nPTF. When TSG3nPTR writing and TSG3nSTR2.TSG3nPTF setting occur simultaneously, TSG3nSTR2.TSG3nPTF setting has a priority, and the flag is not cleared.
0	Reserved	When writing, write the value after reset.

35.3.19 TSG3nOPT0 — TSG3n Option Register 0

This register sets the optional functions.

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> + 020_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	TSG3nSOC	TSG3nSTE	TSG3nPOT	TSG3nPSS	TSG3nIDC	TSG3nPSC	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 35.26 TSG3nOPT0 Register Contents (1/2)

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TSG3nSOC	Enables or disables control of the timer output (TSG3nO1 to TSG3nO6 pins) by software. 0: Disables control by software. 1: Enables control by software. When TSG3nSOC is set to 1, timer output is switched to the software control/ trigger control output pattern specified by TSG3nSPC2 to TSG3nSPC0. The dead time is secured by the dead time counter.
5	TSG3nSTE	Enables or disables control by the pattern output trigger. 0: Disables the TSG3nPTSI0 to TSG3nPTSI2 and TSG3nOPCI0, and TSG3nOPCI1 inputs. 1: Enables TSG3nPTSI0 to TSG3nPTSI2 and TSG3nOPCI0, and TSG3nOPCI1 inputs. • The pattern output trigger is selected by TSG3nPOT. • TSG3nSTE is valid in 120-DC mode and when software output control function is enabled.
4	TSG3nPOT	Selects the pattern output trigger. 0: Switches the output pattern by the external pattern input pins (TSG3nPTSI0 to TSG3nPTSI2) (pattern switch method). 1: Switches the output pattern by the rising edge of the TSG3nOPCI0 and TSG3nOPCI1 (trigger switch method).
3	TSG3nPSS	Selects the pattern output order switch factor. 0: The pattern output order is not switched by TSG3nPSC. 1: The pattern output order is switched by TSG3nPSC.
2	TSG3nIDC	Determines the output pattern from the TSG3nO1 to TSG3nO6 pins in combination with the TSG3nIDC and TSG3nSTR1.TSG3nTSF and TSG3nPSC signals. For the timer output order and patterns to be output, see **Figure 35.80** to **Figure 35.83**, Example of Operation in 120-DC Mode, in **Section 35.4.7.4, (5) Operation in 120-DC Mode**.

Table 35.26 TSG3nOPT0 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TSG3nPSC	Selects the pattern output order when the semi-automatic cruise function is enabled. 0: Switches the timer output (TSG3nO1 to TSG3nO6) in the normal rotation. 1: Switches the timer output (TSG3nO1 to TSG3nO6) in the reverse rotation. • TSG3nPSC specifies the timer output pattern order assuming the output pattern specified by TSG3nSPC2 to TSG3nSPC0 as the initial pattern. TSG3nPSC is valid when TSG3nPOT = 1 and TSG3nPSS = 1. • It is recommended to rewrite TSG3nPSC when TSG3nSTR0.TSG3nTE = 0 or TSG3nPOT = 0. If TSG3nPSC is rewritten when TSG3nPOT = 1, unexpected timer output pattern might be caused. • If the signal input to TSG3nPTSI0 to TSG3nPTSI2 changes with TSG3n operation being stopped (TSG3nSTR0.TSG3nTE = 0), the TSG3nTRG0.TSG3nTS bit should be set to 1 after matching the input signal change logic with the TSG3nPSC order. • For output order in normal or reverse rotation, see **Section 35.4.7.4, 120-DC Mode**. Here, normal rotation and reverse rotation refer to the change of output, and they are different from normal rotation and reverse rotation of a motor.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

35.3.20 TSG3nOPT1 — TSG3n Option Register 1

This register sets the optional functions.

Access: This register can be read or written in 8-bit units.

Address: <TSG3n_base> +024_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TSG3nSPC[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 35.27 TSG3nOPT1 Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	TSG3nSPC [2:0]	Specifies the timer output pattern when software output function is enabled and in 120-DC mode. For the output pattern, see **Section 35.4.7.8, Software Output Control Function**, and **Section 35.4.7.4, 120-DC Mode**.

35.3.21 TSG3nTRG0 — TSG3n Trigger Register 0

This register controls the start of the timer.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <TSG3n_base> + 030_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSG3nTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.28 TSG3nTRG0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset (or the fixed value).
0	TSG3nTS	This bit is a trigger bit that controls the start of the timer. 0: The timer is not started. 1: The timer is started (restarted if TSG3nSTR0.TSG3nTE = 1). When restarted, the 18-bit counter is initialized. This bit is always read as 0.

35.3.22 TSG3nTRG1 — TSG3n Trigger Register 1

This register controls the stop of the timer.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <TSG3n_base> +034_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSG3nTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.29 TSG3nTRG1 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TSG3nTT	This is a trigger bit that controls the stop of the timer. 0: The timer is not stopped. 1: The timer is stopped (TSG3nSTR0.TSG3nTE = 0). This bit is always read as 0.

35.3.23 TSG3nTRG2 — TSG3n Trigger Register 2

TSG3nTRG2 is a trigger bit to reflect the PWM duty setting to TSG3nO1-6 in anytime rewrite mode of HT-PWM mode.

This register can be set to 1 only in HT-PWM mode and when anytime rewrite mode is selected (TSG3nRMC = 1). Do not rewrite this register in other modes (PWM mode, SP-PMW mode, 120-DC mode, HSP-PWM mode) or when reload mode is selected (TSG3nRMC = 0)

Access: This register is a write-only register that can be written in 8-bit units.

Address: <TSG3n_base> + 038_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TSG3nIMT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 35.30 TSG3nTRG2 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TSG3nIMT	Anytime Rewrite Trigger 0: Disabled 1: Changes in duty settings for U, V, and W phases are reflected to timer output in HT-PWM mode and when anytime rewrite mode is selected.

35.3.24 TSG3nCNT — TSG3n Counter Read Buffer Register

This register can access the 16 lower bits of the 18-bit register TSG3nCNTE.

For the operation of this register, see **Section 35.3.25, TSG3nCNTE — TSG3n Bit Extended Counter Read Buffer Register**.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TSG3n_base> + 028_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	16-bit counter															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

35.3.25 TSG3nCNTE — TSG3n Bit Extended Counter Read Buffer Register

The counter values can be read from this register. This register mirrors the contents of TSG3nCNT from which the 16 lower bits of this register can be accessed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <TSG3n_base> + $1A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCNTE (18-bit counter)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCNTE (18-bit counter)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

18-bit counter

This register is a timer read buffer register from which the 18-bit counter value can be read. In HT-PWM mode, the 18-bit counter provides the triangular waveform control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0.

In other modes, the 18-bit counter provides the sawtooth waveform control in which the counter value is incremented by 1.

Table 35.31 TSG3nCNTE Register Count Value

Operating mode	At the Beginning	Minimum Value	Maximum value
HT-PWM mode	TSG3nDTC0	TSG3nDTC0	TSG3nDTC0+TSG3nCMP0E[*1]
Other modes	00000_H	00000_H	TSG3nCMP0E

Note 1. Set the value as TSG3nDTC0+TSG3nCMP0E < $3FFFF_H$.

35.3.26 TSG3nSBC — TSG3n Sub-Counter Read Buffer Register

This register can access the 16 lower bits of the 18-bit register TSG3nSBCE.

For the operation of this register, see **Section 35.3.27, TSG3nSBCE — TSG3n Bit Extended Sub-Counter Read Buffer Register**.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TSG3n_base> + $02C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	16-bit sub-counter															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

35.3.27 TSG3nSBCE — TSG3n Bit Extended Sub-Counter Read Buffer Register

The sub-counter values can be read from this register. This register mirrors the contents of TSG3nSBC from which the 16 lower bits of this register can be accessed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <TSG3n_base> + $1A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nSBCE (18-bit sub-counter)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nSBCE (18-bit sub-counter)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

18-bit counter

This register is a timer read buffer register from which the 18-bit sub-counter value can be read. In HT-PWM mode, the 18-bit counter provides the triangular waveform control in which the counter value is incremented and decremented by 2. Bit 0 is always read as 0. (Available only in HT-PWM mode.).

Table 35.32 TSG3nSBCE Register Count Value

Operating mode	At the Beginning	Minimum Value	Maximum Value
HT-PWM mode	TSG3nDTC0	00000_H	TSG3nDTC0 + TSG3nDTC1 + TSG3nCMP0E*1
Other modes	00000_H	00000_H	00000_H

Note 1. Set the value as TSG3nDTC0 + TSG3nDTC1 + TSG3nCMP0E < $3FFFF_H$.

35.3.28 TSG3nCMP0 — TSG3n Compare Register 0

This register can access the 16 lower bits of the 18-bit register TSG3nCMP0E.

For the operation of this register, see **Section 35.3.29, TSG3nCMP0E — TSG3n Bit Extended Compare Register 0**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 058_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	16-bit compare register															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.29 TSG3nCMP0E — TSG3n Bit Extended Compare Register 0

This register is an 18-bit compare register that specifies the PWM period in all modes. This register mirrors the contents of TSG3nCMP0 from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> +$14C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCMP0E (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP0E (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.33 TSG3nCMP0E Register Setting

Operating mode	At the Beginning	Minimum Value	Maximum Value
HT-PWM mode	TSG3nCMP0E*1	00002_H	$3FFFE_H$
Other modes	TSG3nCMP0E + 1	1 (TSG3nCMP0E = 00000_H)	40000_H (TSG3nCMP0E = $3FFFF_H$)

Note 1. In HT-PWM mode, the lowest bit is ignored.

35.3.30 TSG3nCMP1W — TSG3n Compare Register 1, 2

This register can access the 16 lower bits of the 18-bit register TSG3nCMP1E and TSG3nCMP2E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP2 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP1 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.31 TSG3nCMP3W — TSG3n Compare Register 3, 4

This register can access the 16 lower bits of the 18-bit register TSG3nCMP3E and TSG3nCMP4E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + $04C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP4 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP3 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.32 TSG3nCMP5W — TSG3n Compare Register 5, 6

This register can access the 16 lower bits of the 18-bit register TSG3nCMP5E and TSG3nCMP6E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP6 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP5 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.33 TSG3nCMP7W — TSG3n Compare Registers 7, 8

This register can access the 16 lower bits of the 18-bit register TSG3nCMP7E and TSG3nCMP8E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP8 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP7 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.34 TSG3nCMP9W — TSG3n Compare Registers 9, 10

This register can access the 16 lower bits of the 18-bit register TSG3nCMP9E and TSG3nCMP10E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 048_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP10 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP9 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.35 TSG3nCMP11W — TSG3n Compare Registers 11, 12

This register can access the 16 lower bits of the 18-bit register TSG3nCMP11E and TSG3nCMP12E.

For the operation of this register, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 054_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nCMP12 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP11 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.36 TSG3nCMP1 to TSG3nCMP12 — TSG3n Compare Registers 1 to 12

These registers can access the 16 lower bits of the 18-bit registers TSG3nCMP1E-12E.

For the operation of these registers, see **Section 35.3.37, TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12**.

Access: This register can be read or written in 16-bit units.

Address:
TSG3nCMP1 <TSG3n_base> + 080_H
TSG3nCMP2 <TSG3n_base> + 084_H
TSG3nCMP3 <TSG3n_base> + 098_H
TSG3nCMP4 <TSG3n_base> + $09C_H$
TSG3nCMP5 <TSG3n_base> + 088_H
TSG3nCMP6 <TSG3n_base> + $08C_H$
TSG3nCMP7 <TSG3n_base> + $0A0_H$
TSG3nCMP8 <TSG3n_base> + $0A4_H$
TSG3nCMP9 <TSG3n_base> + 090_H
TSG3nCMP10 <TSG3n_base> + 094_H
TSG3nCMP11 <TSG3n_base> + $0A8_H$
TSG3nCMP12 <TSG3n_base> + $0AC_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	16-bit compare register															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.37 TSG3nCMP1E to TSG3nCMP12E — TSG3n Bit Extended Compare Registers 1 to 12

The compare value is set by these registers. These registers mirror the contents of TSG3nCMP1-12, TSG3nCMP1W, 3W, 5W, 7W, 9W, and 11W from which the 16 lower bits of these registers can be accessed.

Access: This register can be read or written in 32-bit units.

Address: TSG3nCMP1E <TSG3n_base> + $17C_H$
TSG3nCMP2E <TSG3n_base> + 178_H
TSG3nCMP3E <TSG3n_base> + 164_H
TSG3nCMP4E <TSG3n_base> + 160_H
TSG3nCMP5E <TSG3n_base> + 174_H
TSG3nCMP6E <TSG3n_base> + 170_H
TSG3nCMP7E <TSG3n_base> + $15C_H$
TSG3nCMP8E <TSG3n_base> + 158_H
TSG3nCMP9E <TSG3n_base> + $16C_H$
TSG3nCMP10E <TSG3n_base> + 168_H
TSG3nCMP11E <TSG3n_base> + 154_H
TSG3nCMP12E <TSG3n_base> + 150_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCMP1E - TSG3nCMP12E (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMP1E - TSG3nCMP12E (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.34 TSG3nCMP1E-TSG3nCMP12E Register Setting (1/2)

Register	PWM Mode	HT-PWM Mode	SP-PWM Mode	120-DC Mode	HSP-PWM Mode
TSG3nCMP1E	TSG3nO1 clear timing	TSG3nO1 clear timing /TSG3nO2 set timing		Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0.	TSG3nO1 clear timing
TSG3nCMP2E	TSG3nO1 set timing	TSG3nO1 set timing/TSG3nO2 clear timing			TSG3nO1 set timing
TSG3nCMP3E	TSG3nO2 clear timing	—		Duty whenTSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1.	TSG3nO2 clear timing
TSG3nCMP4E	TSG3nO2 set timing	—			TSG3nO2 set timing
TSG3nCMP5E	TSG3nO3 clear timing	TSG3nO3 clear timing/ TSG3nO4 set timing		Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0.	TSG3nO3 clear timing
TSG3nCMP6E	TSG3nO3 set timing	TSG3nO3 set timing/TSG3nO4 clear timing			TSG3nO3 set timing
TSG3nCMP7E	TSG3nO4 clear timing	—		Duty when TSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1.	TSG3nO4 clear timing
TSG3nCMP8E	TSG3nO4 set timing	—			TSG3nO4 set timing
TSG3nCMP9E	TSG3nO5 clear timing	TSG3nO5 clear timing/ TSG3nO6 set timing		Duty when TSG3nO1, 3, or 5 output pattern is selected by TSG3nPAT0.	TSG3nO5 clear timing
TSG3nCMP10E	TSG3nO5 set timing	TSG3nO5 set timing/TSG3nO6 clear timing			TSG3nO5 set timing

Table 35.34 TSG3nCMP1E-TSG3nCMP12E Register Setting (2/2)

Register	PWM Mode	HT-PWM Mode	SP-PWM Mode	120-DC Mode	HSP-PWM Mode
TSG3nCMP11E	TSG3nO6 clear timing	—		Duty when TSG3nO2, 4, or 6 output pattern is selected by TSG3nPAT1.	TSG3nO6 clear timing
TSG3nCMP12E	TSG3nO6 set timing	—			TSG3nO6 set timing

NOTE

The dead time function is used in all operating modes.

In HT-PWM mode, the compare match signal is generated when a compare match occurs not only with the value of the TSG3nCNTE register but also with the value of the TSG3nSBCE register.

In 120-DC mode, the output from TSG3nO1-6 is controlled by the TSG3nCMPmE, TSG3nPAT0, and TSG3nPAT1 registers.

35.3.38 TSG3nDCMP0W — TSG3n Diagnostic Output Compare Register 0, 1

This register can access the 16 lower bits of the 18-bit register TSG3nDCMP0E and TSG3nDCMP1E.

For the operation of this register, see **Section 35.3.40, TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + $05C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSG3nDCMP1(16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nDCMP0(16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.39 TSG3nDCMP2 — TSG3n Diagnostic Output Compare Register 2

This register can access the 16 lower bits of the 18-bit register TSG3nDCMP2E.

For the operation of this register, see **Section 35.3.40, TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nDCMP2 (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.40 TSG3nDCMP0E to 2E — TSG3n Bit Extended Diagnostic Output Compare Register 0 to 2

The compare value is set by these registers. These registers mirror the contents of TSG3nDCMP0W and TSG3nDCMP2 from which the 16 lower bits of these registers can be accessed.

Access: This register can be read or written in 32-bit units.

Address: TSG3nDCMP0E <TSG3n_base> + 148_H
TSG3nDCMP1E <TSG3n_base> + 144_H
TSG3nDCMP2E <TSG3n_base> + 140_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nDCMP0E - TSG3nDCMP2E (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nDCMP0E - TSG3nDCMP2E (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These registers control the diagnostic output timing or AD conversion trigger timing in all modes. A pulse is generated at the match timing of the 18-bit counter value with this register.

35.3.41 TSG3nPAT0W — TSG3n Pattern Register 0

This register specifies the output pattern.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 064_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PAT5T	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAT5T	PAT4T			PAT3T			PAT2T			PAT1T			PAT0T		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Pattern

This register controls UT/VT/WT output in 120-DC mode.

Table 35.35 TSG3nPAT0W Register Setting value and Output Control

PATmT Value	Output Control
000	Fixed Low level
001	PWM output set by TSG3nCMP1E
010	PWM output set by TSG3nCMP2E
011	PWM output set by TSG3nCMP5E
100	PWM output set by TSG3nCMP6E
101	PWM output set by TSG3nCMP9E
110	PWM output set by TSG3nCMP10E
111	Fixed High level

Note: m = 0, 1, 2, 3, 4, 5

35.3.42 TSG3nPAT1W — TSG3n Pattern Register 1

This register specifies the output pattern.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 068_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PAT5B	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAT5B	PAT4B			PAT3B			PAT2B			PAT1B			PAT0B		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Pattern

This register controls UB/VB/WB output in 120-DC mode.

Table 35.36 TSG3nPAT1W Register Setting value and Output Control

PATmB Value	Output Control
000	Fixed Low level
001	PWM output set by TSG3nCMP3E
010	PWM output set by TSG3nCMP4E
011	PWM output set by TSG3nCMP7E
100	PWM output set by TSG3nCMP8E
101	PWM output set by TSG3nCMP11E
110	PWM output set by TSG3nCMP12E
111	Fixed High level

Note: m = 0, 1, 2, 3, 4, 5

35.3.43 TSG3nDTC0W — TSG3n Dead Time Control Register 0

This register sets the dead time value (the period from inverse phase inactivation to positive phase activation).

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + $06C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	Write Protection Code Check														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3nDTC0(10-bit dead time compare)									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To rewrite TSG3nDTC0W[0:9], set bit 14 to bit 0 and TSG3nDTCM to 0 in TSG3nDTPR, and rewrite the TSG3nDTC0W. At this time, when the rewritten value of TSG3nDTC0W[30:16] and the TSG3nDTPR value match, TSG3nDTC0W is rewritten.

During timer operation (TSG3nSTR0.TSG3nTE = 1), rewriting should be performed in reload mode (TSG3nCTL3.TSG3nRMC = 0).

35.3.44 TSG3nDTC1W — TSG3n Dead Time Control Register 1

This register sets the dead time (the period from positive phase inactivation to inverse phase activation).

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 070_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	Write Protection Code Check														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG3nDTC1(10-bit dead time compare)									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

To rewrite TSG3nDTC1W[0:9], set bit 14 to bit 0 and TSG3nDTCM to 0 in TSG3nDTPR, and rewrite the TSG3nDTC1W. At this time, when the rewritten value of TSG3nDTC1W[30:16] and the TSG3nDTPR value match, TSG3nDTC1W is rewritten.

During timer operation (TSG3nSTR0.TSG3nTE = 1), rewriting should be performed in reload mode (TSG3nCTL3.TSG3nRMC = 0).

35.3.45 TSG3nCMPU — TSG3n HT-PWM U Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPUE.

For the operation of this register, see **Section 35.3.48, TSG3nCMPUE — TSG3n Bit Extended HT-PWM U Phase Compare Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0B0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPU(16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.46 TSG3nCMPV — TSG3n HT-PWM V Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPVE.

For the operation of this register, see **Section 35.3.49, TSG3nCMPVE — TSG3n Bit Extended HT-PWM V Phase Compare Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0B4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPV (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.47 TSG3nCMPW — TSG3n HT-PWM W Phase Compare Register

This register can access the 16 lower bits of the 18-bit register TSG3nCMPWE.

For the operation of this register, see **Section 35.3.50, TSG3nCMPWE — TSG3n Bit Extended HT-PWM W Phase Compare Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0B8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPW (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.48 TSG3nCMPUE — TSG3n Bit Extended HT-PWM U Phase Compare Register

This register sets the compare value for U phase in HT-PWM. In addition to the functions of TSG3nCMP1E and TSG3nCMP2E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP1E and TSG3nCMP2E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 35.2**). When this register is read, the same value as TSG3nCMP1E is returned. This register mirrors the contents of TSG3nCMPU from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> +188_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCMPUE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPUE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

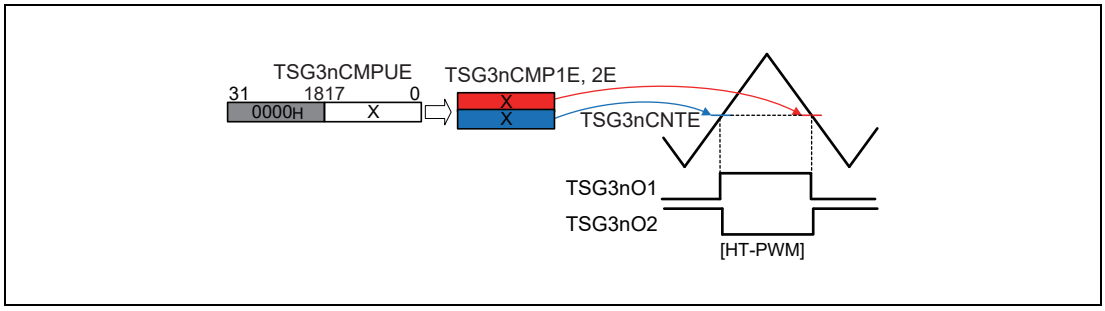


Figure 35.2 TSG3nCMPUE Register Accesses

35.3.49 TSG3nCMPVE — TSG3n Bit Extended HT-PWM V Phase Compare Register

This register sets the compare value for V phase in HT-PWM. In addition to the functions of TSG3nCMP5E and TSG3nCMP6E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP5E and TSG3nCMP6E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 35.3**). When this register is read, the same value as TSG3nCMP5E is returned. This register mirrors the contents of TSG3nCMPV from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> +184_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCMPVE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPVE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

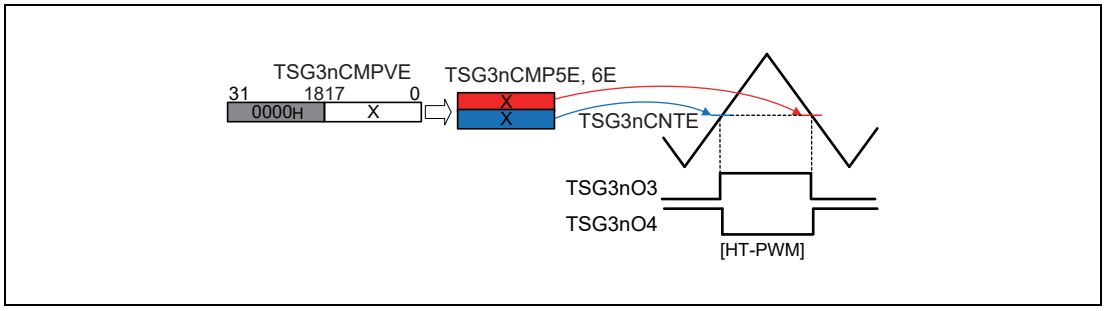


Figure 35.3 TSG3nCMPVE Register Accesses

35.3.50 TSG3nCMPWE — TSG3n Bit Extended HT-PWM W Phase Compare Register

This register sets the compare value for W phase in HT-PWM. In addition to the functions of TSG3nCMP9E and TSG3nCMP10E, this register can access specific registers.

The data written to this register is stored in the TSG3nCMP9E and TSG3nCMP10E registers which allows a generation of symmetry triangular waveform of PWM by one write access (see **Figure 35.4**). When this register is read, the same value as TSG3nCMP9E is returned. This register mirrors the contents of TSG3nCMPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> +180_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nCMPWE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nCMPWE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

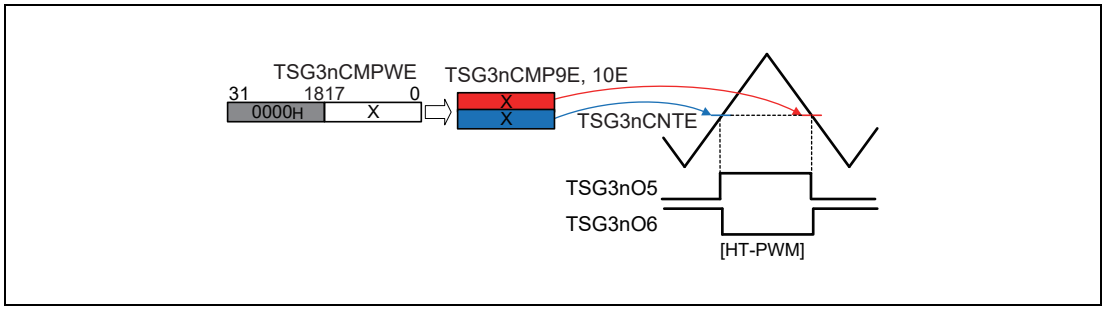


Figure 35.4 TSG3nCMPWE Register Accesses

35.3.51 TSG3nUPW — TSG3n SP-PWM U Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nUPWE.

For the operation of this register, see **Section 35.3.54, TSG3nUPWE — TSG3n Bit Extended SP-PWM U Phase Active Width Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0BC_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nUPW (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.52 TSG3nVPW — TSG3n SP-PWM V Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nVPWE.

For the operation of this register, see **Section 35.3.55, TSG3nVPWE — TSG3n Bit Extended SP-PWM V Phase Active Width Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nVPW (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.53 TSG3nWPW — TSG3n SP-PWM W Phase Active Width Register

This register can access the 16 lower bits of the 18-bit register TSG3nWPWE.

For the operation of this register, see **Section 35.3.56, TSG3nWPWE — TSG3n Bit Extended SP-PWM W Phase Active Width Register**.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + $0C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nWPW (16-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.54 TSG3nUPWE — TSG3n Bit Extended SP-PWM U Phase Active Width Register

This register sets the active width for U phase in SP-PWM mode. The sum of the TSG3nUPWE write data and the TSG3nCMP2E value is stored in TSG3nCMP1E (see **Figure 35.5**). When this register is read, the same value as TSG3nCMP1E is returned. This register mirrors the contents of TSG3nUPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 198_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nUPWE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nUPWE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

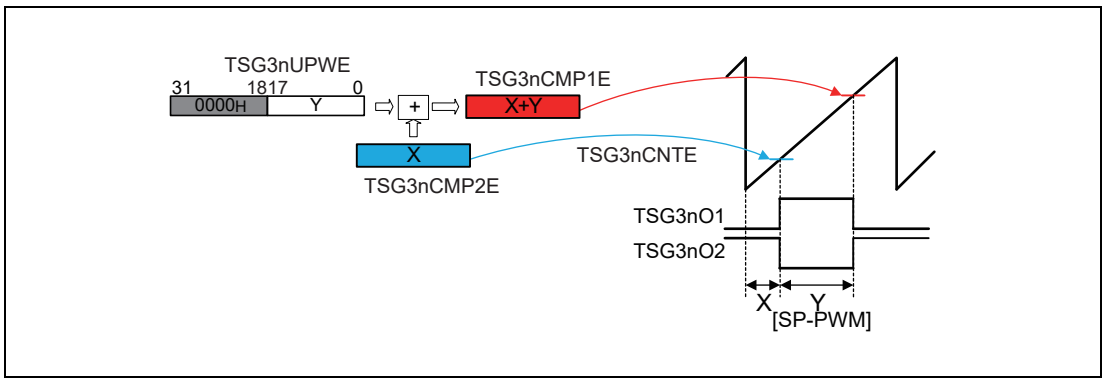


Figure 35.5 TSG3nUPWE Register Accesses

35.3.55 TSG3nVPWE — TSG3n Bit Extended SP-PWM V Phase Active Width Register

This register sets the active width for V phase in SP-PWM mode. The sum of the TSG3nVPWE write data and the TSG3nCMP6E value is stored in TSG3nCMP5E (see **Figure 35.6**). When this register is read, the same value as TSG3nCMP5E is returned. This register mirrors the contents of TSG3nVPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 194_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nVPWE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nVPWE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

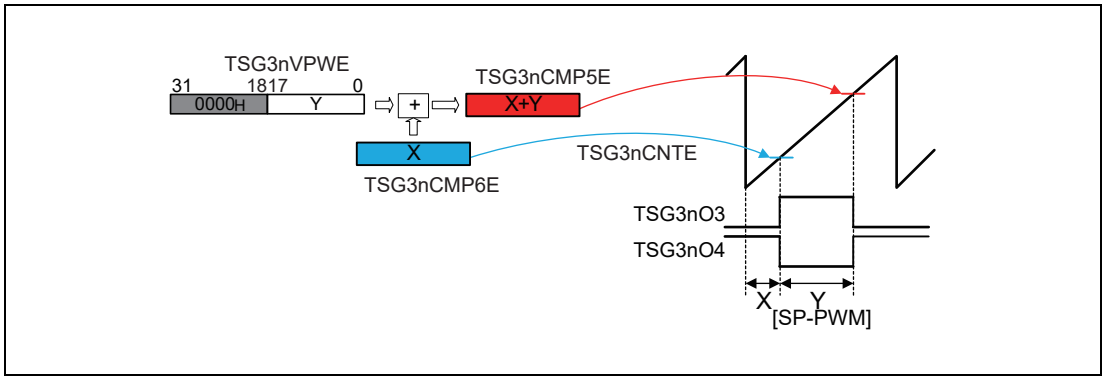


Figure 35.6 TSG3nVPWE Register Accesses

35.3.56 TSG3nWPWE — TSG3n Bit Extended SP-PWM W Phase Active Width Register

This register sets the active width for W phase in SP-PWM mode. The sum of the TSG3nWPWE write data and the TSG3nCMP10E value is stored in TSG3nCMP9E (see **Figure 35.7**). When this register is read, the same value as TSG3nCMP9E is returned. This register mirrors the contents of TSG3nWPW from which the 16 lower bits of this register can be accessed.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 190_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nWPWE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nWPWE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

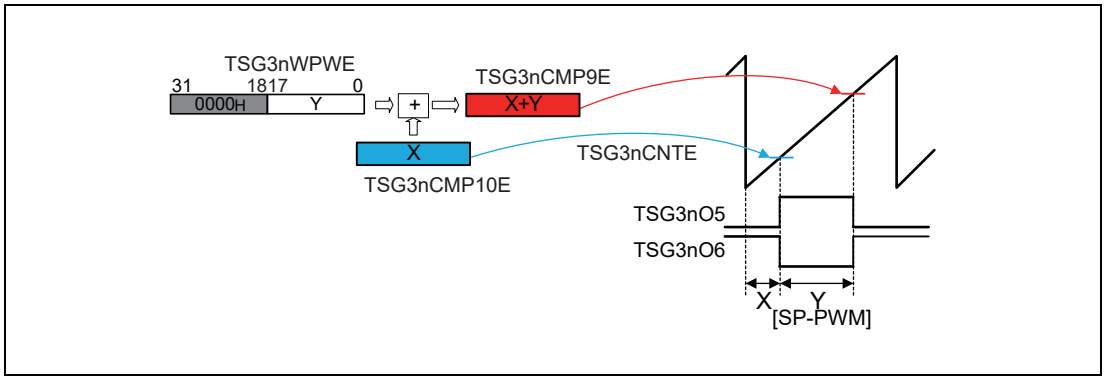


Figure 35.7 TSG3nWPWE Register Accesses

35.3.57 TSG3nHSPCMUE — TSG3n HSP-PWM Mode U Phase Compare Register

This register sets the PWM output width for U phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP1E to TSG3nCMP4E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

When this register is read, the same value as TSG3nCMP1E is returned.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 134_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPCMUE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPCMUE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.58 TSG3nHSPCMVE — TSG3n HSP-PWM Mode V Phase Compare Register

This register sets the PWM output width for V phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP5E to TSG3nCMP8E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

When this register is read, the same value as TSG3nCMP5E is returned.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 130_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPCMVE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPCMVE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.59 TSG3nHSPCMWE — TSG3n HSP-PWM Mode W Phase Compare Register

This register sets the PWM output width for W phase in HSP-PWM mode.

When a write access is made to this register, the values are set to TSG3nCMP9E to TSG3nCMP12E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

When this register is read, the same value as TSG3nCMP9E is returned.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + $12C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPCMWE (18-bit compare register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPCMWE (18-bit compare register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.60 TSG3nHSPSHUE — TSG3n HSP-PWM Mode U Phase Shift Register

This register sets the PWM shift width for U phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMUE after setting the TSG3nHSPSHUE register, the values are set to TSG3nCMP1E to TSG3nCMP4E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 128_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPSHUE (18-bit shift register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPSHUE (18-bit shift register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.61 TSG3nHSPSHVE — TSG3n HSP-PWM Mode V Phase Shift Register

This register sets the PWM shift width for V phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMVE after setting the TSG3nHSPSHVE register, the values are set to TSG3nCMP5E to TSG3nCMP8E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 124_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPSHVE (18-bit shift register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPSHVE (18-bit shift register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.62 TSG3nHSPSHWE — TSG3n HSP-PWM Mode W Phase Shift Register

This register sets the PWM shift width for W phase in HSP-PWM mode.

When a write access is made to TSG3nHSPCMWE after setting the TSG3nHSPSHWE register, the values are set to TSG3nCMP9E to TSG3nCMP12E registers according to the formulas described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode**.

Access: This register can be read or written in 32-bit units.

Address: <TSG3n_base> + 120_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSG3nHSPSHWE (18-bit shift register)	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3nHSPSHWE (18-bit shift register)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

35.3.63 TSG3nDTPR — TSG3n Dead Time Protection Register

This register controls protection of the write access to the dead time register.

Access: This register can be read or written in 16-bit units.

Address: <TSG3n_base> + 210_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG3n DTCM	TSG3nDTPR (Write Protection Code)														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 35.37 TSG3nDTPR Register Contents

Bit Position	Bit Name	Function
15	TSG3nDTCM	Enables or disables rewriting of TSG3nDTC0 and TSG3nDTC1. 0: Enables rewriting of TSG3nDTC0 and TSG3nDTC1. 1: Disables rewriting of TSG3nDTC0 and TSG3nDTC1.
14 to 0	TSG3nDTPR [14:0]	Sets the write protection code (any value from 0000_H to $7FFF_H$).

This register protects TSG3nDTC0 and TSG3nDTC1 from illegal rewriting.

Functions are described below.

- TSG3nDTCM enables or disables rewriting of TSG3nDTC0 and TSG3nDTC1.
- Rewriting of TSG3nDTC0 and TSG3nDTC1 is enabled or disabled by double-checking a match of the write protection code (bits 30 to 16) of TSG3nDTC0 and TSG3nDTC1 with the write protection code of TSG3nDTPR, and the TSG3nDTCM setting.

CAUTION

This register should be set when the timer is stopped (TSG3nSTR0.TSG3nTE = 0). Only the same value can be written during timer operation (TSG3nSTR0.TSG3nTE = 1). If the different value is written to this register when TSG3nSTR0.TSG3nTE = 1, timer operation cannot be guaranteed. If this register is erroneously rewritten, set this register again after stopping the timer.

35.4 Operation

Table 35.38 List of Modes

TSG3nCTL0 Register			
TSG3nMD2	TSG3nMD1	TSG3nMD0	Timer Mode
0	0	0	PWM mode
0	0	1	HT-PWM mode (HT-PWM)
0	1	0	Shift pulse PWM mode (SP-PWM)
0	1	1	120-DC mode
1	0	0	High-accuracy shift pulse PWM mode (HSP-PWM)
Other than above			Setting prohibited

35.4.1 Basic Operation

35.4.1.1 Basic Operation of 18-Bit Counter

The basic operation of the 18-bit counter is described. For details, see **Section 35.4.7, Operating Modes**.

Counting start

The 18-bit counter of TSG3n starts counting in HT-PWM mode when the initial value is 00000_H and after the TSG3nDTC0 value is loaded. The counter starts counting from the initial value 00000_H in all modes except for HT-PWM mode.

In HT-PWM mode, the counter value is incremented by 2 from the value of TSG3nDTC0 and decremented by 2 down to the value of TSG3nDTC0 after the counter value matches with the value of TSG3nCMP0E + TSG3nDTC0. The counter increments from 00000_H, 00001_H, 00002_H, 00003_H, … in all modes except for HT-PWM mode.

Counter clear

The 18-bit counter is cleared by the match of the counter value and the value of TSG3nCMP0E in all modes except for HT-PWM mode. (Clearing operation is not available in HT-PWM mode.)

Counter read during counting

In the TSG3n, the 18-bit counter value during counting can be read through TSG3nCNTE.

Count stop operation

When count operation is stopped (when TSG3nSTR0.TSG3nTE is changed from 1 to 0), TSG3nCNTE and TSG3nSBCE retain the counter value when stopped.

Interrupt operation

In the TSG3n, the following interrupts are generated.

- INTTSG3nI0: A period interrupt by a match of the 18-bit counter value with the TSG3nDTC0 value in HT-PWM mode. A compare match interrupt of the 18-bit counter value with the TSG3nCMP0E buffer register in any mode other than HT-PWM mode.
- INTTSG3nI1: A compare match interrupt of the 18-bit counter value with the TSG3nCMP1E buffer register.
- INTTSG3nI2: A compare match interrupt of the 18-bit counter value with the TSG3nCMP2E buffer register.
- INTTSG3nI3: A compare match interrupt of the 18-bit counter value with the TSG3nCMP3E buffer register.
- INTTSG3nI4: A compare match interrupt of the 18-bit counter value with the TSG3nCMP4E buffer register.
- INTTSG3nI5: A compare match interrupt of the 18-bit counter value with the TSG3nCMP5E buffer register.
- INTTSG3nI6: A compare match interrupt of the 18-bit counter value with the TSG3nCMP6E buffer register.
- INTTSG3nI7: A compare match interrupt of the 18-bit counter value with the TSG3nCMP7E buffer register.
- INTTSG3nI8: A compare match interrupt of the 18-bit counter value with the TSG3nCMP8E buffer register.
- INTTSG3nI9: A compare match interrupt of the 18-bit counter value with the TSG3nCMP9E buffer register.
- INTTSG3nI10: A compare match interrupt of the 18-bit counter value with the TSG3nCMP10E buffer register.
- INTTSG3nI11: A compare match interrupt of the 18-bit counter value with the TSG3nCMP11E buffer register.
- INTTSG3nI12: A compare match interrupt of the 18-bit counter value with the TSG3nCMP12E buffer register.
- INTTSG3nIPEK: A peak interrupt when the 18-bit counter switches from incrementing to decrementing.
- INTTSG3nIVLY: A valley interrupt when the 18-bit counter switches from decrementing to incrementing.
- INTTSG3nIER: A simultaneous active state detection interrupt of the positive phase and inverse phase
- INTTSG3nIWN: A warning detection interrupt

35.4.1.2 Function of Compare Registers

The functions of the compare registers in each operating mode are shown in the following tables.

Table 35.39 Compare Register Functions in Each Mode (1/7)

Operating Mode	TSG3nCMP0E	TSG3nCMP1E	TSG3nCMP2E
PWM mode	PWM period	TSG3nO1 clear timing	TSG3nO1 set timing
HT-PWM mode	PWM period	TSG3nO1 clear timing TSG3nO2 set timing	TSG3nO1 set timing TSG3nO2 clear timing
SP-PWM mode	PWM period	TSG3nO1 clear timing TSG3nO2 set timing	TSG3nO1 set timing TSG3nO2 clear timing
120-DC mode	PWM period	Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0	Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0
HSP-PWM mode	PWM period	TSG3nO1 clear timing	TSG3nO1 set timing

Table 35.39 Compare Register Functions in Each Mode (2/7)

Operating Mode	TSG3nCMP3E	TSG3nCMP4E	TSG3nCMP5E	TSG3nCMP6E
PWM mode	TSG3nO2 clear timing	TSG3nO2 set timing	TSG3nO3 clear timing	TSG3nO3 set timing
HT-PWM mode	Compare match interrupt	Compare match interrupt	TSG3nO3 clear timing TSG3nO4 set timing	TSG3nO3 set timing TSG3nO4 clear timing
SP-PWM mode	—	—	TSG3nO3 clear timing TSG3nO4 set timing	TSG3nO3 set timing TSG3nO4 clear timing
120-DC mode	Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W	Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W	Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W	Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W
HSP-PWM mode	TSG3nO2 clear timing	TSG3nO2 set timing	TSG3nO3 clear timing	TSG3nO3 set timing

Table 35.39 Compare Register Functions in Each Mode (3/7)

Operating Mode	TSG3nCMP7E	TSG3nCMP8E	TSG3nCMP9E	TSG3nCMP10E
PWM mode	TSG3nO4 clear timing	TSG3nO4 set timing	TSG3nO5 clear timing	TSG3nO5 set timing
HT-PWM mode	Compare match interrupt	Compare match interrupt	TSG3nO5 clear timing TSG3nO6 set timing	TSG3nO5 set timing TSG3nO6 clear timing
SP-PWM mode	—	—	TSG3nO5 clear timing TSG3nO6 set timing	TSG3nO5 set timing TSG3nO6 clear timing
120-DC mode	Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W	Select TSG3nO2, TSG3nO4, or TSG3nO6 output by TSG3nPAT1W	Select TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W	TSG3nO1, TSG3nO3, or TSG3nO5 output by TSG3nPAT0W
HSP-PWM mode	TSG3nO4 clear timing	TSG3nO4 set timing	TSG3nO5 clear timing	TSG3nO5 set timing

Table 35.39 Compare Register Functions in Each Mode (4/7)

Operating Mode	TSG3nCMP11E	TSG3nCMP12E	TSG3nDCMP0E	TSG3nDCMP1E
PWM mode	TSG3nO6 clear timing	TSG3nO6 set timing	Diagnostic output or A/D conversion trigger timing	Diagnostic output or A/D conversion trigger timing
HT-PWM mode	Compare match interrupt	Compare match interrupt	Diagnostic output or A/D conversion trigger timing	Diagnostic output or A/D conversion trigger timing
SP-PWM mode	—	—	Diagnostic output or A/D conversion trigger timing	Diagnostic output or A/D conversion trigger timing
120-DC mode	Select TSG3nO2, TSG3nO4, TSG3nO6 by TSG3nPAT1W	Select TSG3nO2, TSG3nO4, TSG3nO6 by TSG3nPAT1W	Diagnostic output or A/D conversion trigger timing	Diagnostic output or A/D conversion trigger timing
HSP-PWM mode	TSG3nO6 clear timing	TSG3nO6 set timing	Diagnostic output or A/D conversion trigger timing	Diagnostic output or A/D conversion trigger timing

Table 35.39 Compare Register Functions in Each Mode (5/7)

Operating Mode	TSG3nDCMP2E	TSG3nCMPUE	TSG3nCMPVE	TSG3nCMPWE
PWM mode	Diagnostic output or A/D conversion trigger timing	—	—	—
HT-PWM mode	Diagnostic output or A/D conversion trigger timing	The TSG3nCMPUE set value is used as the set value of TSG3nCMP1E and TSG3nCMP2E.	The TSG3nCMPVE set value is used as the set value of TSG3nCMP5E and TSG3nCMP6E.	The TSG3nCMPWE set value is used as the set value of TSG3nCMP9E and TSG3nCMP10E.
SP-PWM mode	Diagnostic output or A/D conversion trigger timing	—	—	—
120-DC mode	Diagnostic output or A/D conversion trigger timing	—	—	—
HSP-PWM mode	Diagnostic output or A/D conversion trigger timing	—	—	—

Table 35.39 Compare Register Functions in Each Mode (6/7)

Operating Mode	TSG3nUPWE	TSG3nVPWE	TSG3nWPWE
PWM mode	—	—	—
HT-PWM mode	—	—	—
SP-PWM mode	The sum of the TSG3nUPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value.	The sum of the TSG3nVPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value.	The sum of the TSG3nWPWE set value and the TSG3nCMP2E set value is used as the TSG3nCMP1E set value.
120-DC mode	—	—	—
HSP-PWM mode	—	—	—

Table 35.39 Compare Register Functions in Each Mode (7/7)

Operating Mode	TSG3nHSPCMUE, TSG3nHSPSHUE	TSG3nHSPCMVE, TSG3nHSPSHVE	TSG3nHSPCMWE, TSG3nHSPSHWE
PWM mode	—	—	—
HT-PWM mode	—	—	—
SP-PWM mode	—	—	—
120-DC mode	—	—	—
HSP-PWM mode	TSG3nCMP1E-4E is set based on the TSG3nHSPCMUE set value and the set values in TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHUE.	TSG3nCMP5E-8E is set based on the TSG3nHSPCMVE set value and the set values of TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHVE.	TSG3nCMP9E-12E is set based on the TSG3nHSPCMWE set value and the set values of TSG3nCMP0E, TSG3nDTC0, TSG3nDTC1, and TSG3nHSPSHWE.

35.4.1.3 Compare Register Rewrite Operation

TSG3 can be set to reload mode or anytime rewrite mode using theTSG3nRMC bit.

Reload mode is enabled when TSG3nRMC = 0. The registers listed as “Enable” on the “Reload” column in **Section 35.3.1, List of Registers** are simultaneously updated at the reload timing.

Anytime rewrite mode is enabled when TSG3nRMC = 1. The registers are updated independently every time the value is written to the relevant register.

The update timing of the registers to be reloaded in reload mode and anytime rewrite mode in each mode are listed in the following table.

Table 35.40 Updating Timing of Compare Registers by Mode

Mode	Anytime Rewrite TSG3nRMC = 1	Reload TSG3nRMC = 0
PWM mode	TSG3nCMP0E: At the next counter clear timing of the 18-bit counter	At reload timing
	Registers other than TSG3nCMP0E: At a write access to the register	
HT-PWM mode	TSG3nCMP0E: At the next peak or valley timing of TSG3nCNTE	At reload timing
	TSG3nCMP1E, 2E, 5E, 6E, 9E, 10E: At writing 1 to the TSG3nIMT bit	
	Registers other than TSG3nCMP0E,1E, 2E, 5E, 6E, 9E, 10E: At a write access to the register	
SP-PWM mode	TSG3nCMP0E: At the next counter clear timing of the 18-bit counter	At reload timing
	Registers other than TSG3nCMP0E: At a write access to the register	
120-DC mode	Setting prohibited	At reload timing
HSP-PWM mode	Setting prohibited	At reload timing

Anytime Rewrite Mode

In this mode, the compare registers are rewritten independently. Whenever a value is written to the compare register, the written value is reflected at the timing of **Table 35.40**.

Reload mode (Simultaneous Rewrite Function)

Writing to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) enables reload (sets the reload request flag (TSG3nSTR0.TSG3nRSF)), and the values of all the pertinent registers are updated simultaneously at the next reload timing (reload).

The reload timing is the peak or valley timing of the 18-bit counter when the TSG3nTRG0.TSG3nTS bit is changed from 0 to 1. Reloading is controlled by TSG3nCTL4.TSG3nPRE and TSG3nVRE.

Writing to any register other than TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) does not enable reloading.

Do not write to the registers to be reloaded until the next reload timing after reloading is enabled by writing to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE). The pertinent registers should be rewritten when the reload request flag (TSG3nSTR0.TSG3nRSF) is 0.

Rewriting registers to be reloaded by DMA transfer

Some of the registers to be reloaded can be rewritten by DMA transfer. DMA transfer is performed as follows.

Table 35.41 Example of DMA Transfer Order of Registers to be Reloaded

Address	Register Name	DMA Transfer Order (Example)
<TSG3n_base> + 040_H	TSG3nCMP1W	
<TSG3n_base> + 044_H	TSG3nCMP5W	
<TSG3n_base> + 048_H	TSG3nCMP9W	
<TSG3n_base> + $04C_H$	TSG3nCMP3W	
<TSG3n_base> + 050_H	TSG3nCMP7W	
<TSG3n_base> + 054_H	TSG3nCMP11W	
<TSG3n_base> + 058_H	TSG3nCMP0	
<TSG3n_base> + $05C_H$	TSG3nDCMP0W	
<TSG3n_base> + 060_H	TSG3nDCMP2	
<TSG3n_base> + 064_H	TSG3nPAT0W	
<TSG3n_base> + 068_H	TSG3nPAT1W	
<TSG3n_base> + $06C_H$	TSG3nDTC0W	
<TSG3n_base> + 070_H	TSG3nDTC1W	

Table 35.42 Example of DMA Transfer Order of Registers to be Reloaded

Address	Register Name	DMA Transfer Order (Example)
<TSG3n_base> + 140_H	TSG3nDCMP2E	
<TSG3n_base> + 144_H	TSG3nDCMP1E	
<TSG3n_base> + 148_H	TSG3nDCMP0E	
<TSG3n_base> + $14C_H$	TSG3nCMP0E	
<TSG3n_base> + 150_H	TSG3nCMP12E	
<TSG3n_base> + 154_H	TSG3nCMP11E	
<TSG3n_base> + 158_H	TSG3nCMP8E	
<TSG3n_base> + $15C_H$	TSG3nCMP7E	
<TSG3n_base> + 160_H	TSG3nCMP4E	
<TSG3n_base> + 164_H	TSG3nCMP3E	
<TSG3n_base> + 168_H	TSG3nCMP10E	
<TSG3n_base> + $16C_H$	TSG3nCMP9E	
<TSG3n_base> + 170_H	TSG3nCMP6E	
<TSG3n_base> + 174_H	TSG3nCMP5E	
<TSG3n_base> + 178_H	TSG3nCMP2E	
<TSG3n_base> + $17C_H$	TSG3nCMP1E	

Table 35.43 Duty Setting in HT-PWM Mode

Address	Register Name	DMA Transfer Order (Example)
<TSG3n_base> + 180_H	TSG3nCMPWE	
<TSG3n_base> + 184_H	TSG3nCMPVE	
<TSG3n_base> + 188_H	TSG3nCMPUE	

Table 35.44 Active Width Setting in SP-PWM Mode

Address	Register Name	DMA Transfer Order (Example)
<TSG3n_base> + 190_H	TSG3nWPWE	
<TSG3n_base> + 194_H	TSG3nVPWE	
<TSG3n_base> + 198_H	TSG3nUPWE	

Table 35.45 Shift Width and Duty Setting in HSP-PWM Mode

Address	Register Name	DMA Transfer Order (Example)
<TSG3n_base> + 120_H	TSG3nHSPSHWE	
<TSG3n_base> + 124_H	TSG3nHSPSHVE	
<TSG3n_base> + 128_H	TSG3nHSPSHUE	
<TSG3n_base> + $12C_H$	TSG3nHSPCMWE	
<TSG3n_base> + 130_H	TSG3nHSPCMVE	
<TSG3n_base> + 134_H	TSG3nHSPCMUE	

NOTES

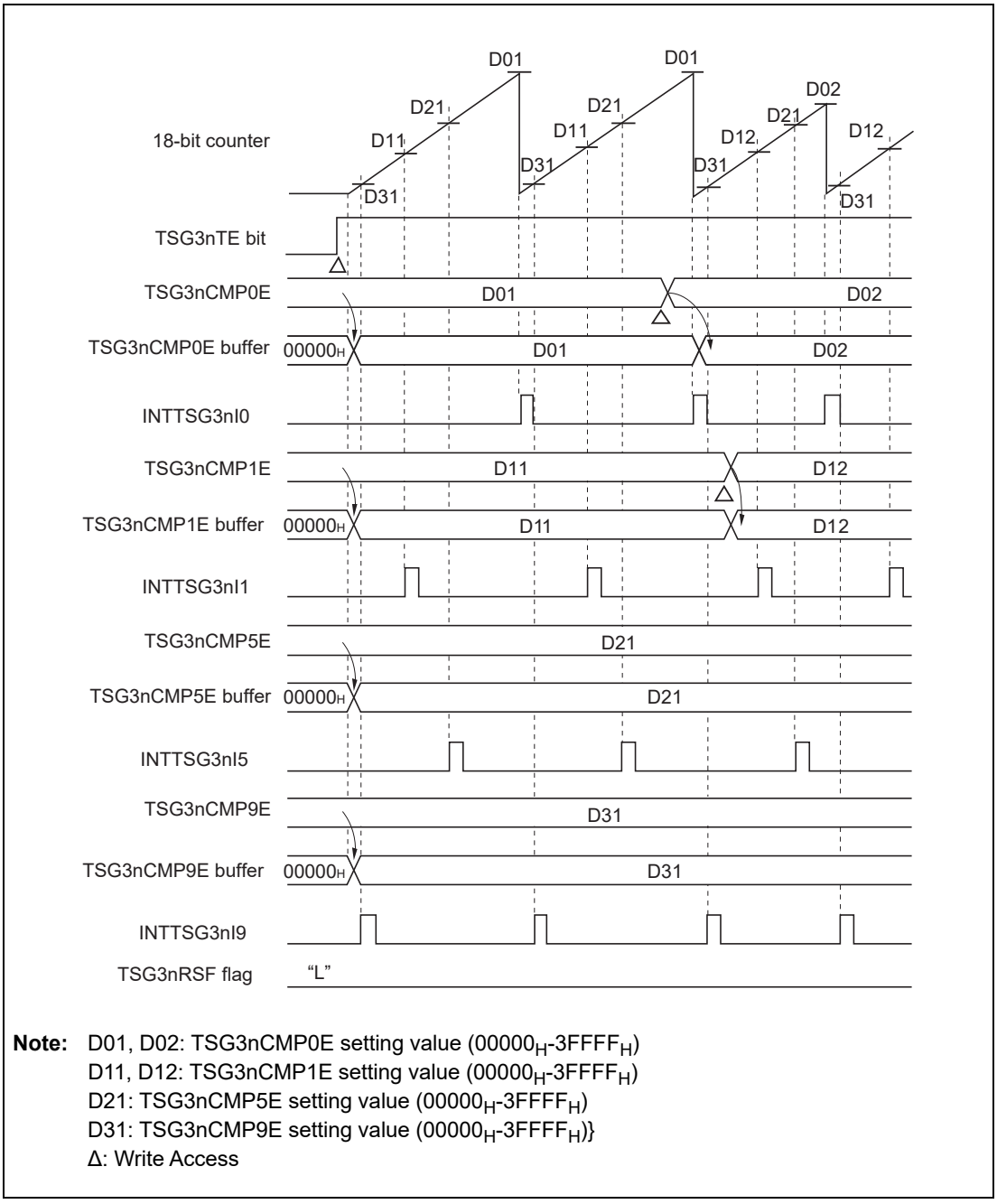
1. TSG3nCTL4 and TSG3nIOC3 should be rewritten individually.
2. Since writing to TSG3nCMP1E (including TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE) enables reloading, it should be rewritten after all the other registers to be reloaded have been rewritten (ready to be reloaded)

(1) Example of Operation in Anytime Rewrite Mode

In this mode, the values written to the compare registers (TSG3nCMP1E to TSG3nCMP12E) are transferred to the internal buffer registers immediately, and are compared with the counter value.

The values are transferred to the internal compare buffer registers one clock cycle (PCLK) after being written to the compare registers (TSG3nCMP1E to TSG3nCMP12E).

The transfer timing of the TSG3nCMP0E is the peak or valley timing (only in HT-PWM mode) of the 18-bit counter after being written to the compare registers, or at the match timing of the TSG3nCMP0E value with the 18-bit counter value (in any mode other than HT-PWM mode).



Note: D01, D02: TSG3nCMP0E setting value (00000_H-$3FFFF_H$)
D11, D12: TSG3nCMP1E setting value (00000_H-$3FFFF_H$)
D21: TSG3nCMP5E setting value (00000_H-$3FFFF_H$)
D31: TSG3nCMP9E setting value (00000_H-$3FFFF_H$)}
Δ: Write Access

Figure 35.8 Anytime Rewrite Timing (Example in PWM Mode)

(a) Data reflection on PWM in Anytime Rewrite in HT-PWM

In anytime rewrite operation in HT-PWM mode, the values are transferred to the buffer at the timing when 1 is written to the TSG3nIMT bit after the settings of TSG3nCMP1E, 2E, 5E, 6E, 9E and 10E registers are modified and PWM output is forcibly set/cleared depending on the modified set value.

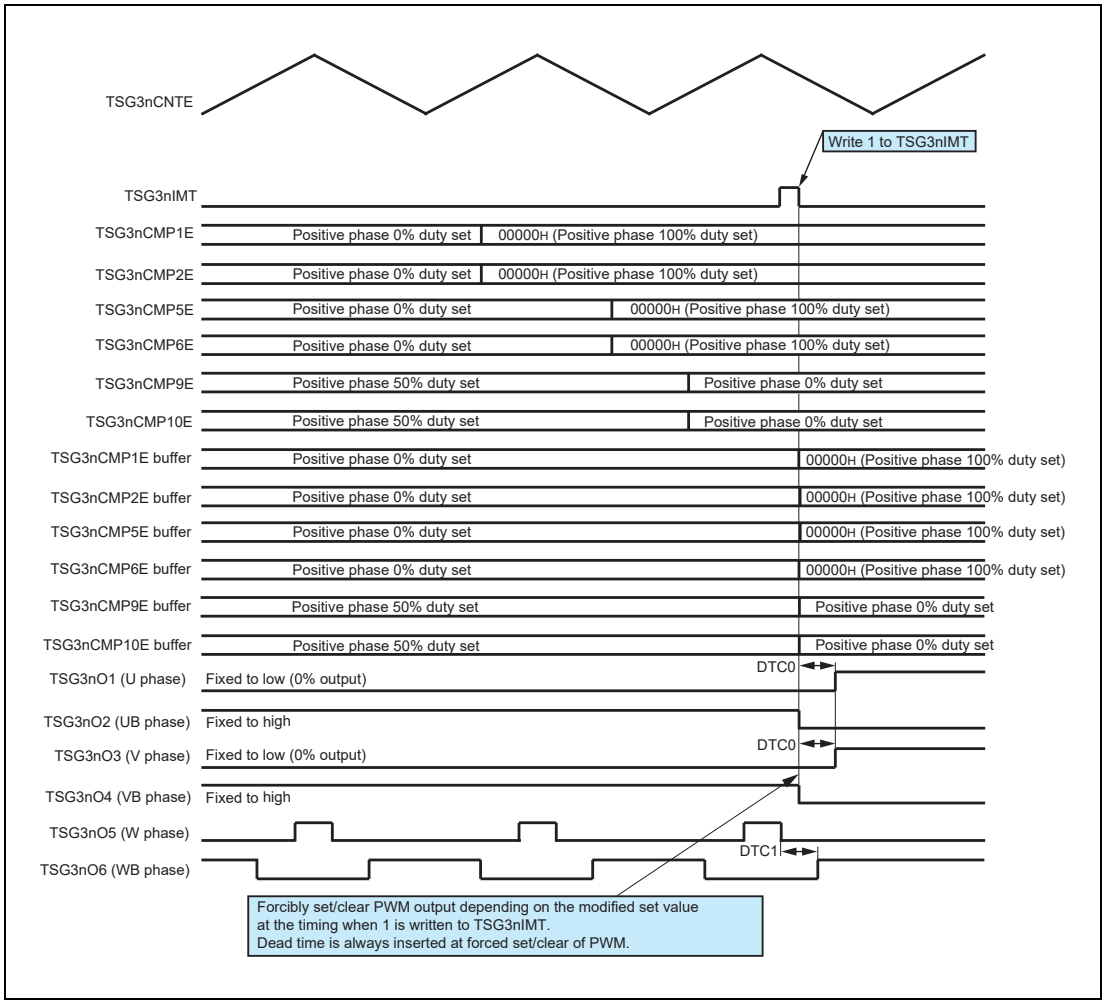


Figure 35.9 Update Timing of TSG3nCMP1E, 2E, 5E, 6E, 9E,and 10E at Anytime Rewrite Operation in HT-PWM Mode

(2) Example of Operation in Reload Mode (Simultaneous Rewrite Function)

The rewritten values of the registers to be reloaded (the registers listed in the **Section 35.3.1, List of Registers** with "Enabled" in the column of "Reload") can be transferred to the corresponding buffer registers simultaneously at the reload timing.

The registers should be rewritten when the pertinent reload request flag (TSG3nSTR0.TSG3nRSF) is 0.

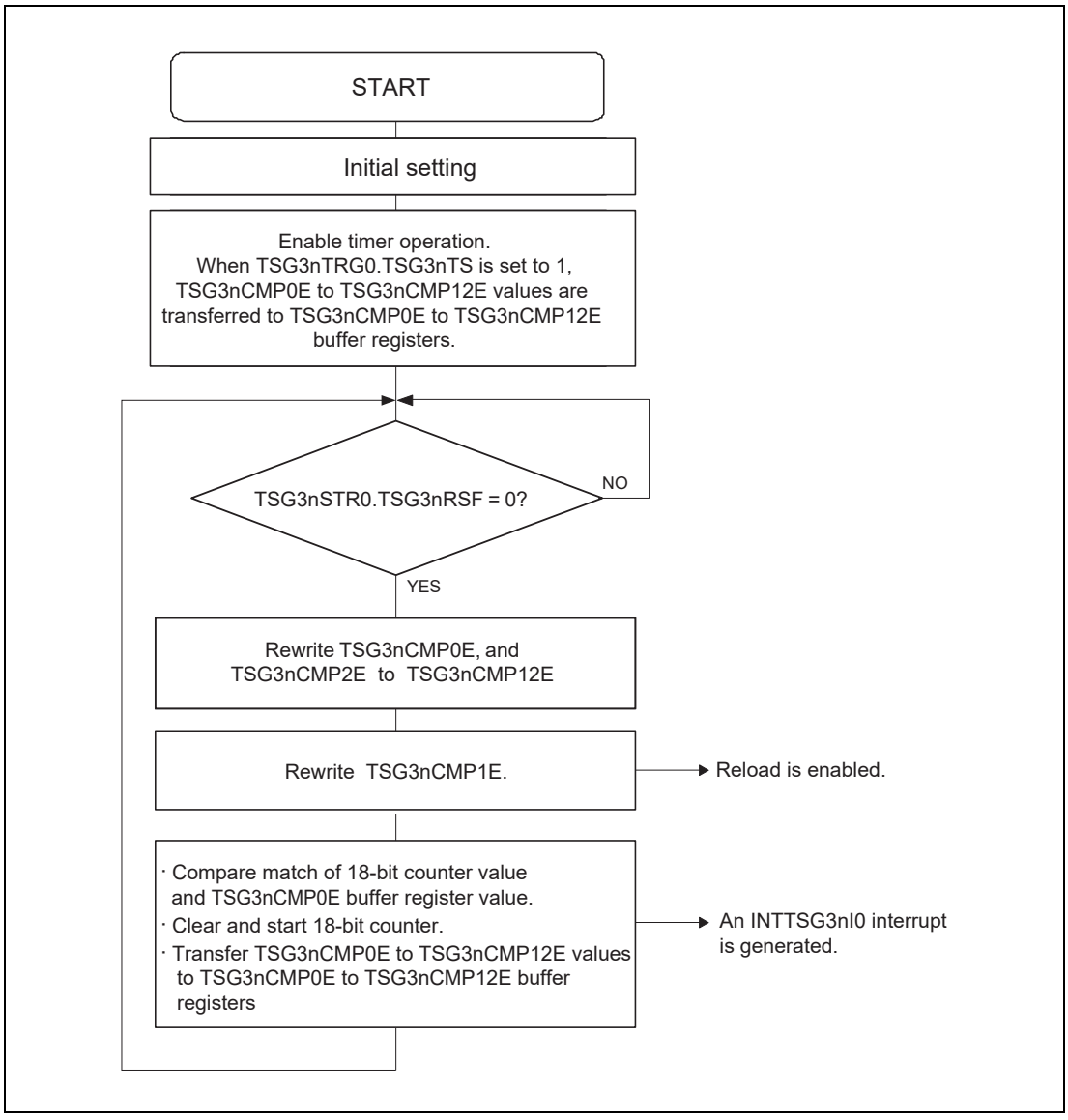


Figure 35.10 Basic Operation Flow in Reload Mode (Simultaneous Rewrite Function) (Example of PWM Mode)

CAUTION

Writing to TSG3nCMP1E also enables reloading. Therefore, TSG3nCMP1E should be rewritten after TSG3nCMP0E and TSG3nCMP2E to TSG3nCMP12E registers have been rewritten.

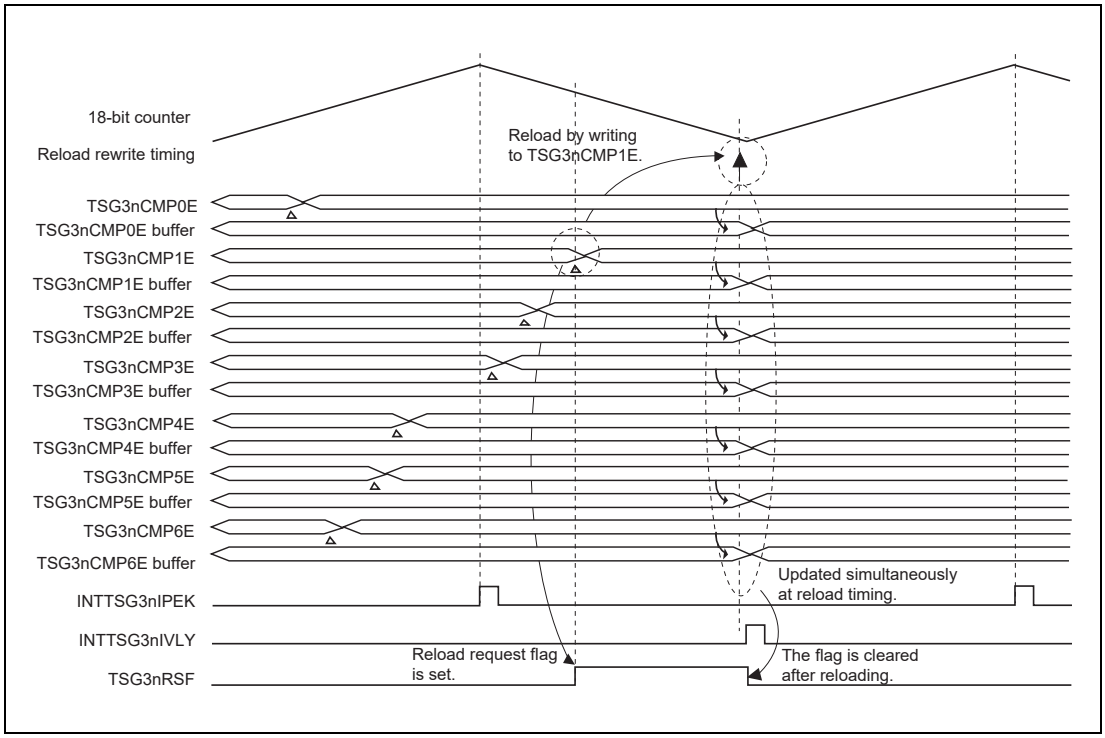


Figure 35.11 Simultaneous Rewrite Timing (Example of HT-PWM Mode) (1/2)

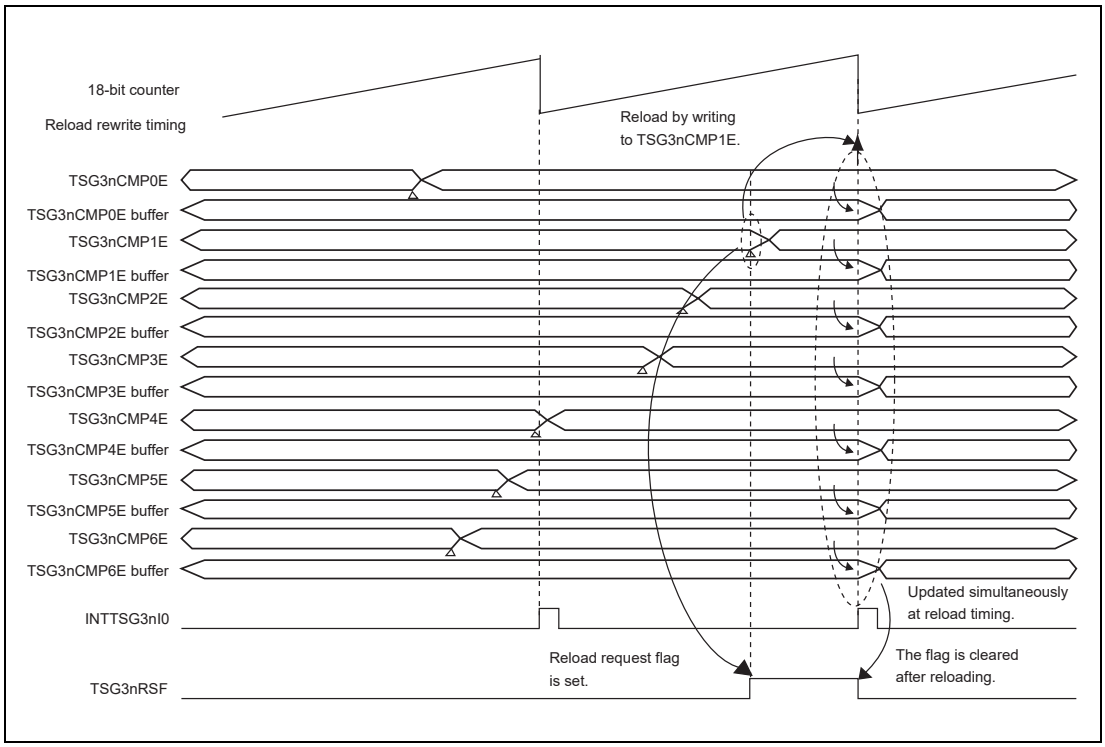


Figure 35.11 Simultaneous Rewrite Timing (Example of PWM Mode) (2/2)

(a) Reload Rewrite Setting Example in Each Mode

Reloading conditions and setting examples are shown in the following tables.

Table 35.46 List of Reload Settings (when TSG3nCTL3.TSG3nRIA = 0)

Mode	TSG3nCTL4. TSG3nPRE	TSG3nCTL4. TSG3nVRE	TSG3nCTL4. TSG3nPIE	TSG3nCTL4. TSG3nVIE	TSG3nCTL4. TSG3nRCC04- TSG3nRCC00	Reload
PWM mode SP-PWM mode 120-DC mode HSP-PWM mode	0	0/1	0/1	0/1	Any value	Setting prohibited
	1	0	0/1	0/1	Any value	When INTTSG3nI0 is generated.
	1	1	0/1	0/1	Any value	When INTTSG3nI0 is generated.
HT-PWM mode	0	0	0/1	0/1	Any value	Setting prohibited
	0	1	0/1	0/1	Any value	When INTTSG3nIVLY is generated.
	1	0	0/1	0/1	Any value	When INTTSG3nIPEK is generated
	1	1	0/1	0/1	Any value	When INTTSG3nIPEK or INTTSG3nIVLY is generated.

Table 35.47 List of Reload Settings (when TSG3nCTL3.TSG3nRIA = 1)

Mode	TSG3nCTL4. TSG3nPRE	TSG3nCTL4. TSG3nVRE	TSG3nCTL4. TSG3nPIE	TSG3nCTL4. TSG3nVIE	TSG3nCTL4. TSG3nRCC04- TSG3nRCC00	Reload
PWM mode SP-PWM mode 120-DC mode HSP-PWM mode	0	0/1	0/1	0/1	Any value	Setting prohibited
	1	0	0	0/1	Any value	Setting prohibited
	1	0	1	0/1	Any value	When INTTSG3nI0 is generated
	1	1	0	0/1	Any value	Setting prohibited
	1	1	1	0/1	Any value	When INTTSG3nI0 is generated
HT-PWM mode	0	0	0/1	0/1	Any value	Setting prohibited
	0	1	0	0	Any value	Setting prohibited
	0	1	0	1	Any value	When INTTSG3nIVLY is generated
	0	1	1	0	Any value	Setting prohibited
	0	1	1	1	Any value	When INTTSG3nIVLY is generated
	1	0	0	0/1	Any value	Setting prohibited
	1	0	1	0/1	Any value	When INTTSG3nIPEK is generated
	1	1	0	0	Any value	Setting prohibited
	1	1	0	1	Any value	When INTTSG3nIVLY is generated
	1	1	1	0	Any value	When INTTSG3nIPEK is generated
	1	1	1	1	Any value	When INTTSG3nIPEK or INTTSG3nIVLY is generated

35.4.1.4 List of Outputs in Each Mode

The list of timer outputs (TSG3nO0-7 pins) in each mode is shown in the following tables.

Table 35.48 List of Timer Outputs in Each mode (1/3)

Operating Mode	TSG3nO0 Pin	TSG3nO1 Pin	TSG3nO2 Pin
PWM mode	— (Fixed to low.)	Outputs a PWM signal by compare match of TSG3nCMP1E and TSG3nCMP2E.	Outputs a PWM signal by compare match of TSG3nCMP3E and TSG3nCMP4E.
HT-PWM mode	Outputs the status indicating whether the 18-bit counter or 18-bit sub-counter is incremented or decremented.	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP1E and TSG3nCMP2E.	Outputs an inverse phase PWM signal (with dead time) to TSG3nO1 pin.
SP-PWM mode	— (Fixed to low.)	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP1E and TSG3nCMP2E.	Outputs an inverse phase PWM signal (with dead time) to TSG3nO1.
120-DC mode	— (Fixed to low.)	Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E.	Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E.
HSP-PWM mode	— (Fixed to low.)	Outputs a PWM signal by compare match of TSG3nCMP1E and TSG3nCMP2E.	Outputs a PWM signal by compare match of TSG3nCMP3E and TSG3nCMP4E.

Table 35.48 List of Timer Outputs in Each mode (2/3)

Operating Mode	TSG3nO3 Pin	TSG3nO4 Pin	TSG3nO5 Pin
PWM mode	Outputs a PWM signal by compare match of TSG3nCMP5E and TSG3nCMP6E.	Outputs a PWM signal by compare match of TSG3nCMP7E and TSG3nCMP8E.	Outputs a PWM signal by compare match of TSG3nCMP9E and TSG3nCMP10E.
HT-PWM mode	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP5E and TSG3nCMP6E.	Outputs an inverse phase PWM signal (with dead time) to TSG3nO3.	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP9E and TSG3nCMP10E.
SP-PWM mode	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP5E and TSG3nCMP6E.	Outputs an inverse phase PWM signal (with dead time) to TSG3nO3.	Outputs a positive phase PWM signal (with dead time) by compare match of TSG3nCMP9E and TSG3nCMP10E
120-DC mode	Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E.	Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E.	Outputs a PWM signal by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E
HSP-PWM mode	Outputs a PWM signal by compare match of TSG3nCMP5E and TSG3nCMP6E	Outputs a PWM signal by compare match of TSG3nCMP7E and TSG3nCMP8E	Outputs a PWM signal by compare match of TSG3nCMP9E and TSG3nCMP10E

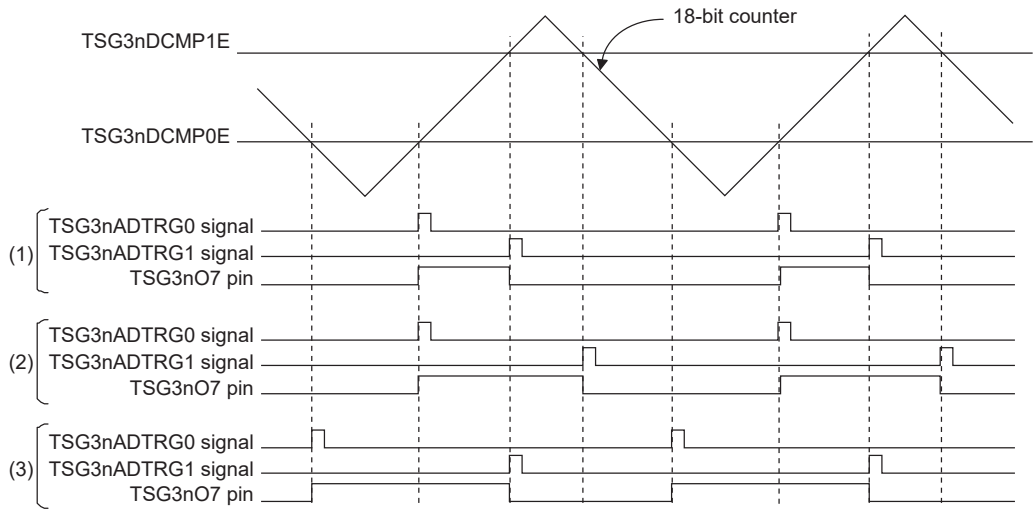
Table 35.48 List of Timer Outputs in Each mode (3/3)

Operating Mode	TSG3nO6 Pin	TSG3nO7 Pin
PWM mode	Outputs a PWM signal by compare match of TSG3nCMP11E and TSG3nCMP12E	Outputs a diagnostic signal or A/D conversion trigger.*1
HT-PWM mode	Outputs an inverse phase PWM signal (with dead time) to TSG3nO5	Outputs a diagnostic signal or A/D conversion trigger.*1
SP-PWM mode	Outputs an inverse phase PWM signal (with dead time) to TSG3nO5	Outputs a diagnostic signal or A/D conversion trigger*1
120-DC mode	Outputs a PWM signal by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E	Outputs a diagnostic signal or A/D conversion trigger.*1
HSP-PWM mode	Outputs a PWM signal by compare match of TSG3nCMP11E and TSG3nCMP12E	Outputs a diagnostic signal or A/D conversion trigger*1

Note 1. For TSG3nO7, See **Section 35.4.1.4, (a) TSG3nO7 Pin Output Control**

(a) TSG3nO7 Pin Output Control

The TSG3nO7 pin can output a pulse of A/D conversion trigger (TSG3nIOC1.TSG3nTGS = 0) or diagnostic output (TSG3nIOC1.TSG3nTGS = 1). When outputting a pulse of A/D conversion trigger, the TSG3nO7 pin is activated at the rising edge of the TSG3nADTRG0 signal, and inactivated at the rising edge of the TSG3nADTRG1 signal. When the TSG3nADTRG0 signal is detected while the TSG3nO7 pin is active, the TSG3nO7 pin remains active. When the TSG3nADTRG1 signal is detected while the TSG3nO7 pin is inactive, the TSG3nO7 pin remains inactive. If TSG3nADTRG0 and TSG3nADTRG1 signal triggers occur simultaneously, the TSG3nO7 pin is inactivated.



Note: (1) When TSG3nDCMP0E register<TSG3nDCMP1E register, TSG3nCTL5 register = 0004_H, and TSG3nCTL6 register = 0010_H

(2) When TSG3nDCMP0E register<TSG3nDCMP1E register, TSG3nCTL5 register = 0004_H, and TSG3nCTL6 register = 0020_H

(3) When TSG3nDCMP0E register<TSG3nDCMP1E register, TSG3nCTL5 register = 0008_H, and TSG3nCTL6 register = 0010_H

Figure 35.12 Example of A/D Trigger Output Timing of TSG3nO7 Pin (TSG3nIOC1.TSG3nTGS = 0)

During diagnostic output, the active level is output on the TSG3nO7 pin with the width specified by the TSG3nCTL0.TSG3nDWD bit when the values of the TSG3nDCMP0E to TSG3nDCMP2E bits match that of the 18-bit counter. If a TSG3nDCMP0E to TSG3nDCMP2E value again matches the value of the 18-bit counter value while the diagnostic output on the TSG3nO7 pin is already at the active level, pulse output on the TSG3nO7 pin continues for the number of cycles of PCLK (8 or 16) set by the TSG3nDWD bit from the point of the later match.

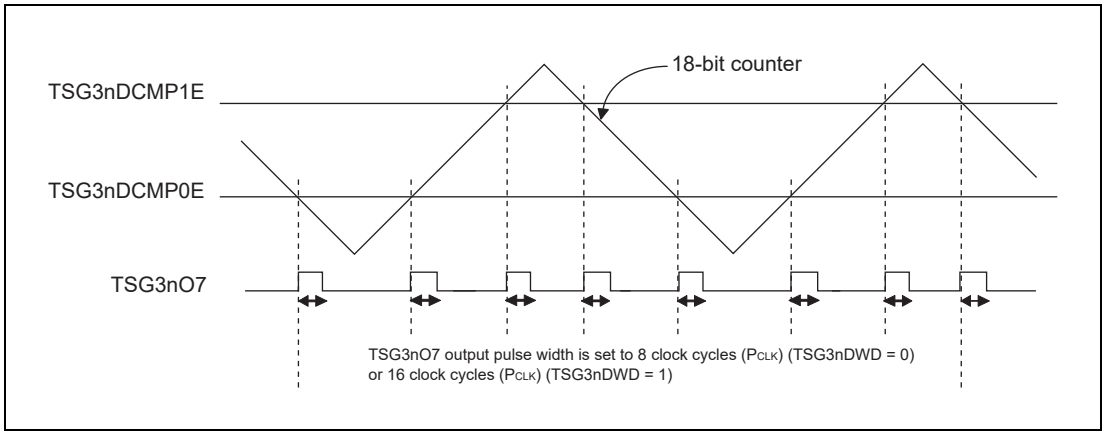


Figure 35.13 Example of TSG3nO7 Pin Diagnostic Pulse Output Timing (1) (TSG3nIOC1.TSG3nTGS = 1)

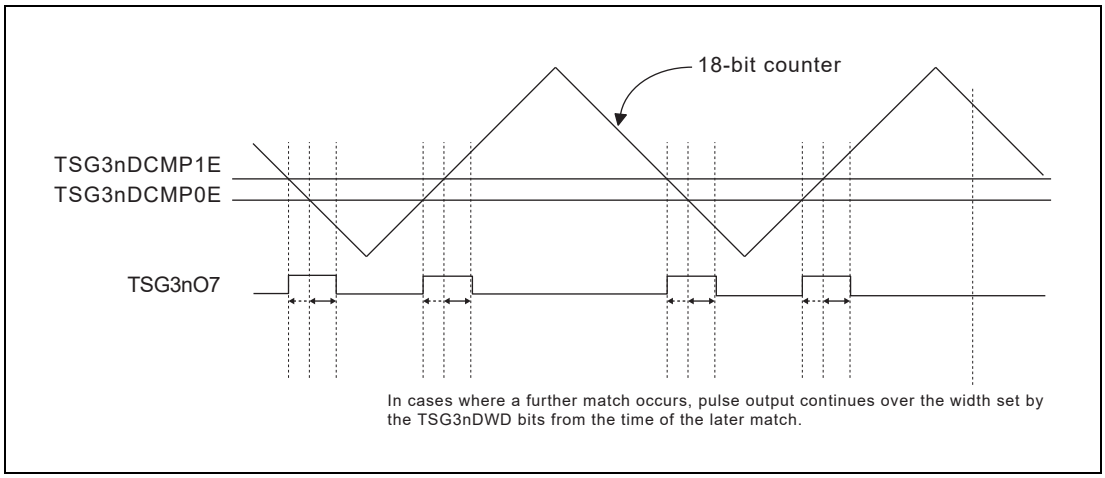


Figure 35.14 Example of TSG3nO7 Pin Diagnostic Pulse Output Timing (2) (with Pulse Output Width Overlapped)

35.4.1.5 Restart

Setting TSG3nTRG0.TSG3nTS to 1 or input to TSG3nTSST while the timer is operating (TSG3nSTR0.TSG3nTE = 1) leads to the following operations.

- Initializing the following counters;

  18-bit main counter, 18-bit sub-counter, interrupt skipping counter

- Initializing the following flags;

  TSG3nCUF, TSG3nSUF, TSG3nRSF, and TSG3nTE of the TSG3nSTR0 register

  TSG3nOPF2 to TSG3nOPF0 of the TSG3nSTR1 register (TSG3nTSF is not initialized)

  TSG3nPPF, TSG3nPEF, TSG3nTDF, TSG3nNDF, TSG3nPRF, and TSG3nPTF of the TSG3nSTR2 register (TSG3nTBF2 to TSG3nTBF20 are not initialized)

- Reloading the following registers;

  TSG3nCTL2, TSG3nCTL4, TSG3nIOC3, TSG3nCMP0, TSG3nCMP0E, TSG3nCMPmW (m = 1, 3, 5, 7, 9, or 11)

  TSG3nCMP1 to TSG3nCMP12, TSG3nCMP1E to TSG3nCMP12E

  TSG3nDCMP0W, TSG3nDCMP2, TSG3nDCMP0E to TSG3nDCMP2E

  TSG3nPAT0W, TSG3nPAT1W

  TSG3nCMPU, TSG3nCMPV, TSG3nCMPW

  TSG3nCMPUE, TSG3nCMPVE, TSG3nCMPWE

  TSG3nUPW, TSG3nVPW, TSG3nWPW

  TSG3nUPWE, TSG3nVPWE, TSG3nWPWE

  TSG3nHSPSHUE, TSG3nHSPSHVE, TSG3nHSPSHWE

  TSG3nHSPCMUE, TSG3nHSPCMVE, TSG3nHSPCMWE

- Reflecting the settings in the following register;

  TSG3nOPT0*1

The dead time is always inserted because the dead time counter continues counting while the dead time control registers TSG3nDTC0W and TSG3nDTC1W are reloaded on restart.

If the counter is restarted while diagnostic output is in progress, the output is initialized.

Note 1. The settings in the TSG3nOPT0 register bits are reflected at the timings described below:
For TSG3nSOC
Switching from 0 to 1: immediately after written
Switching from 1 to 0: when the register is reloaded
For TSG3nSTE, TSG3nPOT, TSG3nPSS, TSG3nIDC, and TSG3nPSC
• At the next timer cycle
• When the output patterns are switched

35.4.2 Match Interrupt

The TSG3n can generate interrupts such as a compare match interrupt (INTTSG3nIm), a peak interrupt (INTTSG3nIPEK), and a valley interrupt (INTTSG3nIVLY). For an error interrupt and warning interrupt (INTTSG3nIER and INTTSG3nIWN), see **Section 35.4.6, Error/Warning Interrupt**.

A period interrupt (INTTSG3nI0) is generated for each timer period. In HT-PWM mode, it is generated when the TSG3nDTC0 buffer register value matches with the 18-bit counter value. When the 18-bit counter performs sawtooth waveform operation (PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode), it is generated after the 18-bit counter value has matched with the TSG3nCMP0E buffer register value.

A compare-match interrupt (INTTSG3nIm) is generated by a match of the TSG3nCMPmE buffer register value with the 18-bit counter value depending on the compare register to be used in each operating mode (m = 1 to 12).

A peak interrupt (INTTSG3nIPEK) is generated in all the modes. In HT-PWM mode, it is generated when the 18-bit counter switches from incrementing to decrementing. When the 18-bit counter performs sawtooth waveform operation (PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode), it is generated after the 18-bit counter value has matched with the TSG3nCMP0E buffer register value (the same timing as an INTTSG3nI0 interrupt).

A valley interrupt (INTTSG3nIVLY) is generated when the 18-bit counter switches from decrementing to incrementing in HT-PWM mode.

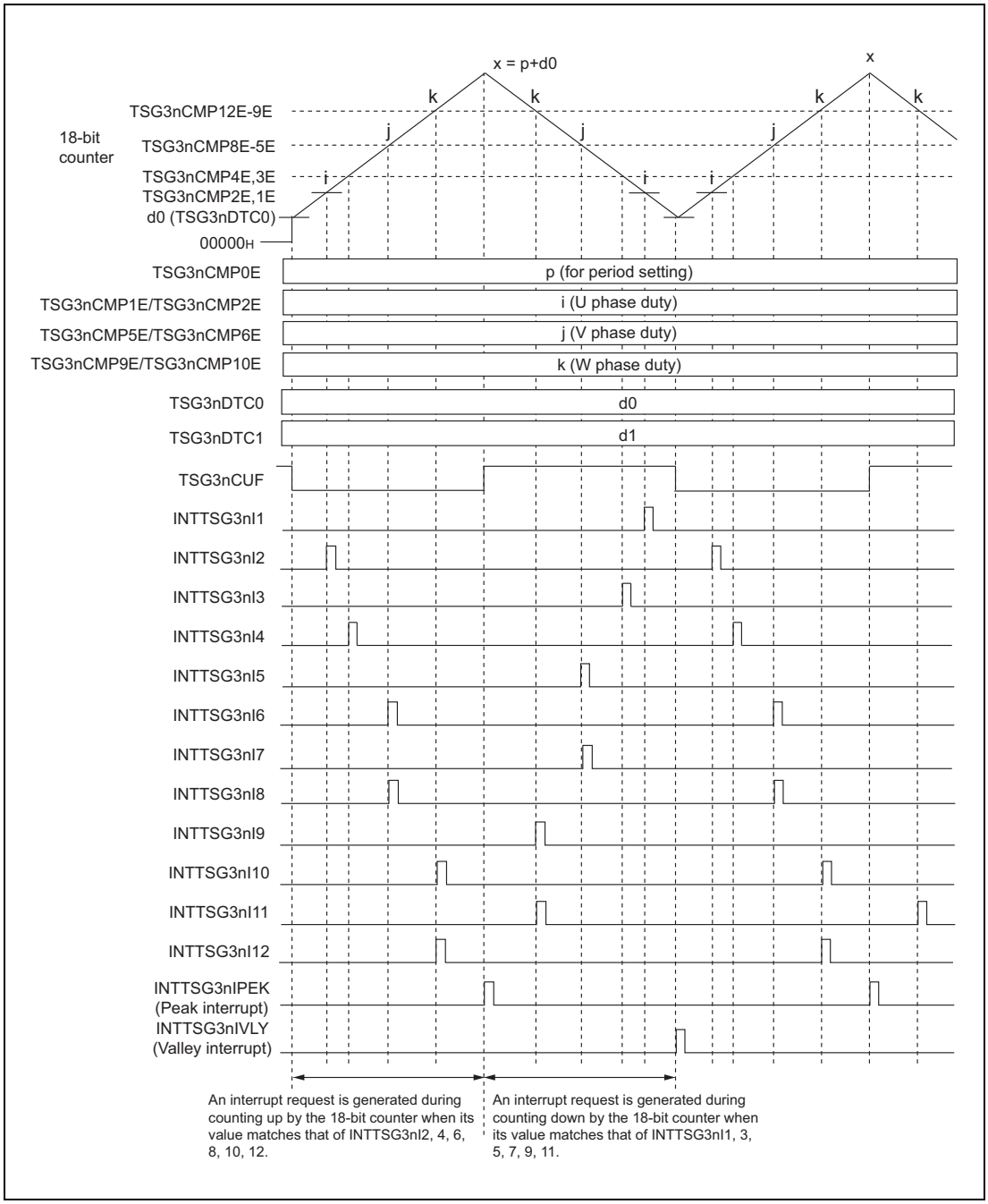


Figure 35.15 Interrupt Generation Example (Example of HT-PWM Mode)

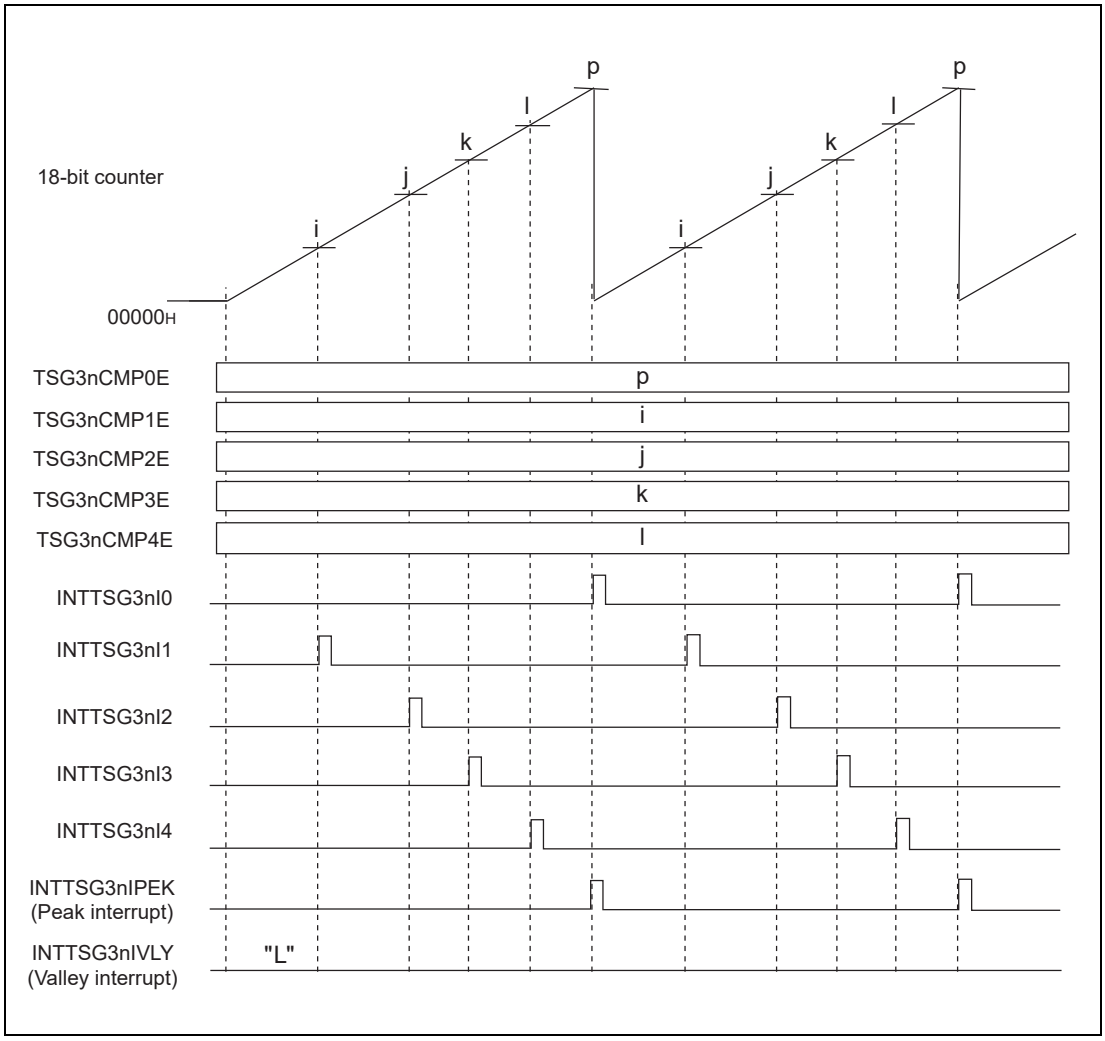


Figure 35.16 Interrupt Generation Example (Example of PWM Mode)

Interrupt in each mode (INTTSG3nI0-INTTSG3nI12, INTTSG3nIPEK, INTTSG3nIVLY, INTTSG3nIER, and INTTSG3nIWN) are listed in **Table 35.49**.

Table 35.49 List of Interrupts in Each Mode (1/5)

Operating Mode	INTTSG3nI0	INTTSG3nI1	INTTSG3nI2	INTTSG3nI3
PWM mode	TSG3nCMP0E compare match interrupt	TSG3nCMP1E compare match interrupt*1	TSG3nCMP2E compare match interrupt*1	TSG3nCMP3E compare match interrupt*1
HT-PWM mode	Period interrupt	TSG3nCMP1E compare match interrupt*2 when decrementing (TSG3nCUF = 1)	TSG3nCMP2E compare match interrupt*2 when incrementing (TSG3nCUF = 0)	TSG3nCMP3E compare match interrupt*2 when decrementing (TSG3nCUF = 1)
SP-PWM mode	TSG3nCMP0E compare match interrupt	TSG3nCMP1E compare match interrupt*1	TSG3nCMP2E compare match interrupt*1	—
120-DC mode	TSG3nCMP0E compare match interrupt	TSG3nCMP1E compare match interrupt*1	TSG3nCMP2E compare match interrupt*1	TSG3nCMP3E compare match interrupt*1
HSP-PWM mode	TSG3nCMP0E compare match interrupt	TSG3nCMP1E compare match interrupt*1	TSG3nCMP2E compare match interrupt*1	TSG3nCMP3E compare match interrupt*1

Note 1. A compare match interrupt is not generated when TSG3nCMP0E < TSG3nCMPmE (m = 1 to 12)

Note 2. A compare match interrupt is not generated when $00000_H \leq$ TSG3nCMPmE < TSG3nDTC0, (TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE.

Table 35.49 List of Interrupts in Each Mode (2/5)

Operating Mode	INTTSG3nI4	INTTSG3nI5	INTTSG3nI6	INTTSG3nI7
PWM mode	TSG3nCMP4E compare match interrupt*1	TSG3nCMP5E compare match interrupt*1	TSG3nCMP6E compare match interrupt*1	TSG3nCMP7E compare match interrupt*1
HT-PWM mode	TSG3nCMP4E compare match interrupt*2 when incrementing (TSG3nCUF = 0)	TSG3nCMP5E compare match interrupt*2 when decrementing (TSG3nCUF = 1)	TSG3nCMP6E compare match interrupt*2 when incrementing (TSG3nCUF = 0)	TSG3nCMP7E compare match interrupt*2 when decrementing (TSG3nCUF = 1)
SP-PWM mode	—	TSG3nCMP5E compare match interrupt*1	TSG3nCMP6E compare match interrupt*1	—
120-DC mode	TSG3nCMP4E compare match interrupt*1	TSG3nCMP5E compare match interrupt*1	TSG3nCMP6E compare match interrupt*1	TSG3nCMP7E compare match interrupt*1
HSP-PWM mode	TSG3nCMP4E compare match interrupt*1	TSG3nCMP5E compare match interrupt*1	TSG3nCMP6E compare match interrupt*1	TSG3nCMP7E compare match interrupt*1

Note 1. A compare match interrupt is not generated when TSG3nCMP0E < TSG3nCMPmE (m = 1 to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq$ TSG3nCMPmE < TSG3nDTC0, (TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE.

Table 35.49 List of Interrupts in Each Mode (3/5)

Operating Mode	INTTSG3nI8	INTTSG3nI9	INTTSG3nI10	INTTSG3nI11
PWM mode	TSG3nCMP8E compare match interrupt*1	TSG3nCMP9E compare match interrupt*1	TSG3nCMP10E compare match interrupt*1	TSG3nCMP11E compare match interrupt*1
HT-PWM mode	TSG3nCMP8E compare match interrupt*2 when incrementing (TSG3nCUF = 0)	TSG3nCMP9E compare match interrupt*2 when decrementing (TSG3nCUF = 1)	TSG3nCMP10E compare match interrupt*2 when incrementing (TSG3nCUF = 0)	TSG3nCMP11E compare match interrupt*2 when decrementing (TSG3nCUF = 1)
SP-PWM mode	—	TSG3nCMP9E compare match interrupt*1	TSG3nCMP10E compare match interrupt*1	—
120-DC mode	TSG3nCMP8E compare match interrupt*1	TSG3nCMP9E compare match interrupt*1	TSG3nCMP10E compare match interrupt*1	TSG3nCMP11E compare match interrupt*1
HSP-PWM mode	TSG3nCMP8E compare match interrupt*1	TSG3nCMP9E compare match interrupt*1	TSG3nCMP10E compare match interrupt*1	TSG3nCMP11E compare match interrupt*1

Note 1. A compare match interrupt is not generated when TSG3nCMP0E < TSG3nCMPmE (m = 1 to 12).

Note 2. A compare match interrupt is not generated when $00000_H \leq$ TSG3nCMPmE < TSG3nDTC0, (TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE.

Table 35.49 List of Interrupts in Each Mode (4/5)

Operating Mode	INTTSG3nI12	INTTSG3nIPEK	INTTSG3nIVLY
PWM mode	TSG3nCMP12E compare match interrupt*1	Peak interrupt at the same timing with INTTSG3nI0	—
HT-PWM mode	TSG3nCMP12E compare match interrupt*2 when incrementing (TSG3nCUF=0)	Peak interrupt	Valley interrupt
SP-PWM mode	—	Peak interrupt at the same timing with INTTSG3nI0	—
120-DC mode	TSG3nCMP12E compare match interrupt*1	Peak interrupt at the same timing with INTTSG3nI0	—
HSP-PWM mode	TSG3nCMP12E compare match interrupt*1	Peak interrupt at the same timing with INTTSG3nI0	—

Note 1. A compare match interrupt is not generated when TSG3nCMP0E < TSG3nCMPmE (m = 1 to 12).

Note 2. A compare match interrupt is not generated when 00000_H ≤ TSG3nCMPmE < TSG3nDTC0, (TSG3nCMP0E + TSG3nDTC0) < TSG3nCMPmE.

Table 35.49 List of Interrupts in Each Mode (5/5)

Operating Mode	INTTSG3nIER	INTTSG3nIWN
PWM mode	Error interrupt	Warning interrupt
HT-PWM mode	Error interrupt	Warning interrupt
SP-PWM mode	Error interrupt	Warning interrupt
120-DC mode	Error interrupt	Warning interrupt
HSP-PWM mode	Error interrupt	Warning interrupt

35.4.3 Flags

Table 35.50 List of Flags

Number	Flag Name	Symbol	Registers	Operating Mode
(1)	Up count flag	TSG3nCUF	TSG3nSTR0	HT-PWM mode
		TSG3nSUF	TSG3nSTR0	
(2)	Positive phase and inverse phase simultaneous active state detection flag	TSG3nTBF0-TSG3nTBF2	TSG3nSTR2	All operating modes
(3)	Reload request flag	TSG3nRSF	TSG3nSTR0	All operating modes
(4)	Noise Detection Flag	TSG3nNDF	TSG3nSTR2	All operating modes
(5)	Pattern order detection flag	TSG3nTSF	TSG3nSTR1	All operating modes
(6)	Pattern error detection flag	TSG3nPEF	TSG3nSTR2	All operating modes
(7)	Pattern reversal detection flag	TSG3nPRF	TSG3nSTR2	All operating modes
(8)	TSG3nPTSI2 to TSG3nPTSI0 pin abnormal toggle detection flag	TSG3nPTF	TSG3nSTR2	All operating modes
(9)	TSG3nOPCI0 and TSG3nOPCI1 signal simultaneous trigger detection flag	TSG3nTDF	TSG3nSTR2	All operating modes
(10)	Pattern phase difference detection flag	TSG3nPPF	TSG3nSTR2	All operating modes
(11)	Timer output pattern flag	TSG3nOPF0 to TSG3nOPF2	TSG3nSTR1	All operating modes
(12)	Pattern switch detection signal (internal signal)	TSG3nPTE	—	All operating modes

35.4.3.1 Up Count Flag (TSG3nCUF and TSG3nSUF)

Name

Up count flag (TSG3nSTR0.TSG3nCUF and TSG3nSUF)

Description

There are following two up count flags.

TSG3nCUF is an up/down count flag of the 18-bit counter.

TSG3nSUF is an up/down count flag of the 18-bit sub-counter.

For both TSG3nCUF and TSG3nSUF, 0 means increment, and 1 means decrement.

TSG3nCUF and TSG3nSUF can be used only in HT-PWM mode.

Example of operation

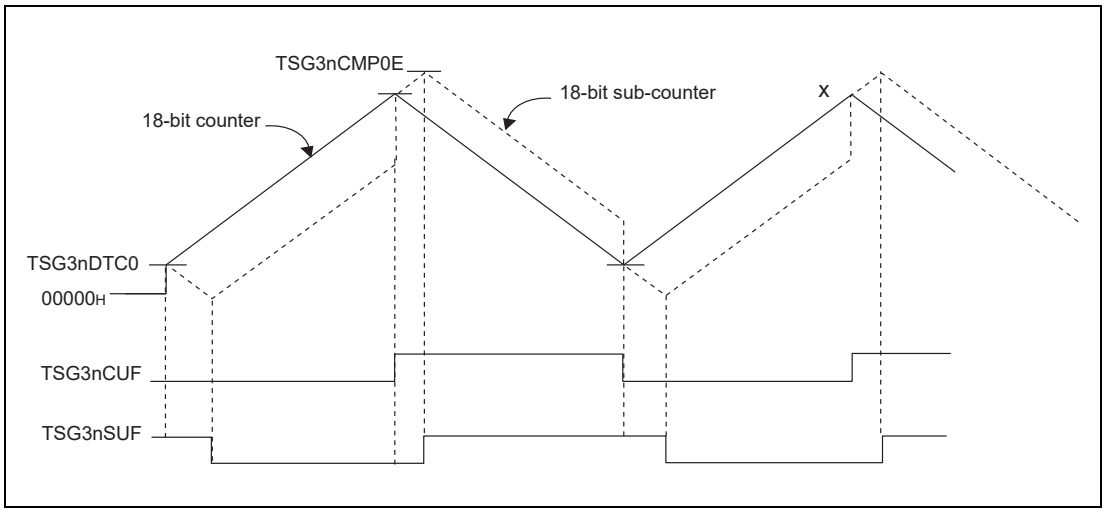


Figure 35.17 Example of Up count flag operation

NOTES

1. TSG3nCUF value is:
   0 (up count) when TSG3nDTC0 ≤ 18-bit counter ≤ (TSG3nCMP0E + TSG3nDTC0-2)
   1 (down count) when (TSG3nCMP0E + TSG3nDTC0) ≥ 18-bit counter ≥ TSG3nDTC0 + 2
2. TSG3nSUF value is:
   0 (up count) when 0 ≤ 18-bit sub-counter ≤ (TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1-2)
   1 (down count) when (TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1) ≥ 18-bit sub-counter ≥ 2

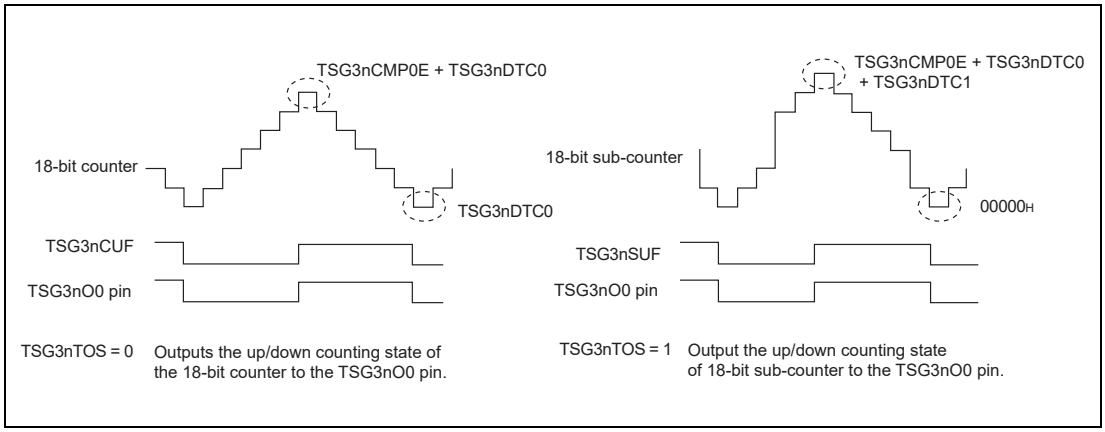


Figure 35.18 TSG3nO0 Pin Output depending onTSG3nIOC1.TSG3nTOS Setting

Operating Mode

TSG3nCUF and TSG3nSUF can be used only in HT-PWM mode.

35.4.3.2 Positive Phase and Inverse Phase Simultaneous Active State Detection Flag (TSG3nTBF0 to TSG3nTBF2)

Name

Positive phase and inverse phase simultaneous active state detection flag (TSG3nSTR2.TSG3nTBF0 to TSG3nTBF2 flags)

Description

When any of TSG3nCTL1.TSG3nTBA2 to TSG3nTBA0 is 1, TSG3nTBF0 to TSG3nTBF2 can detect the simultaneous active state of the positive phase and inverse phase of TSG3n.

When the simultaneous active state of the positive phase and inverse phase of the TSG3n is detected, the corresponding TSG3nTBF0 to TSG3nTBF2 flags are set to 1, and an error interrupt (INTTSG3nIER) is generated. The flags are cleared when 1 is written to TSG3nSTC.TSG3nTBR0 to TSG3nTBR2, respectively.

Example of operation

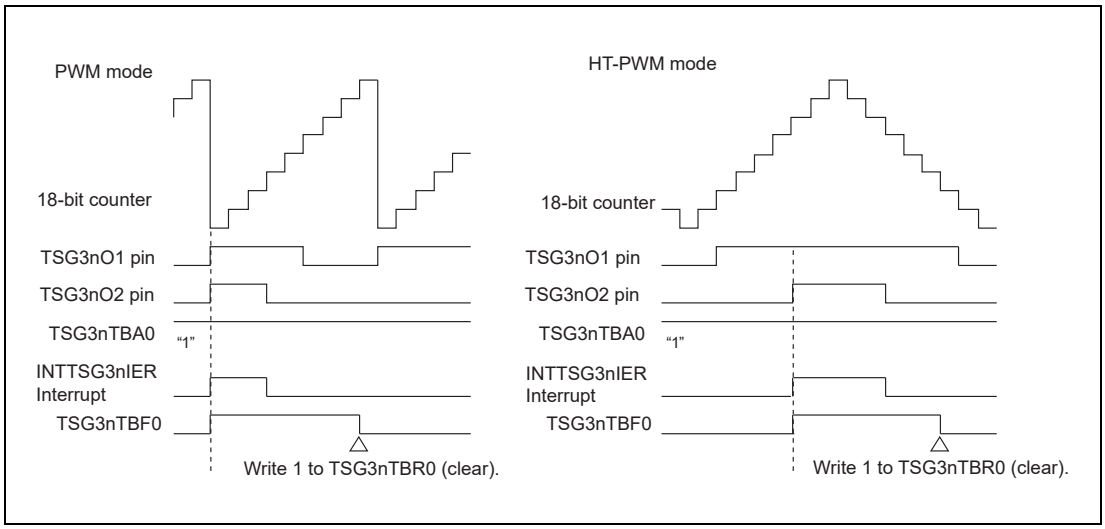


Figure 35.19 Example of Positive Phase and Inverse Phase Simultaneous Active State Detection Flag Operation

Operating Mode

Available in all operating modes.

CAUTION

TSG3nTBF0 to TSG3nTBF2 are valid only when TSG3nCTL1.TSG3nTBA0 to TSG3nTBA2 = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.3.3 Reload Request Flag (TSG3nRSF)

Name

Reload request flag (TSG3nSTR0.TSG3nRSF)

Description

TSG3nRSF is set to 1 when a reload request is generated (when a value is written to TSG3nCMP1E (TSG3nCMP1, TSG3nCMP1W, TSG3nCMPUE, TSG3nCMPU, TSG3nUPWE, TSG3nUPW, and TSG3nHSPCMUE)), and cleared to 0 when reload occurs and the value is transferred to all the buffer registers.

Example of operation

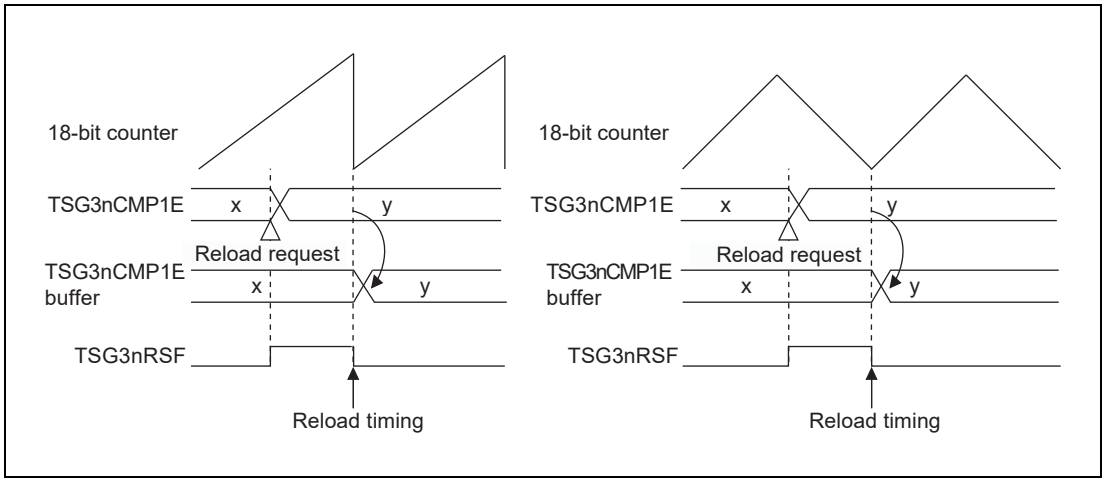


Figure 35.20 Example of Reload Request Flag Operation

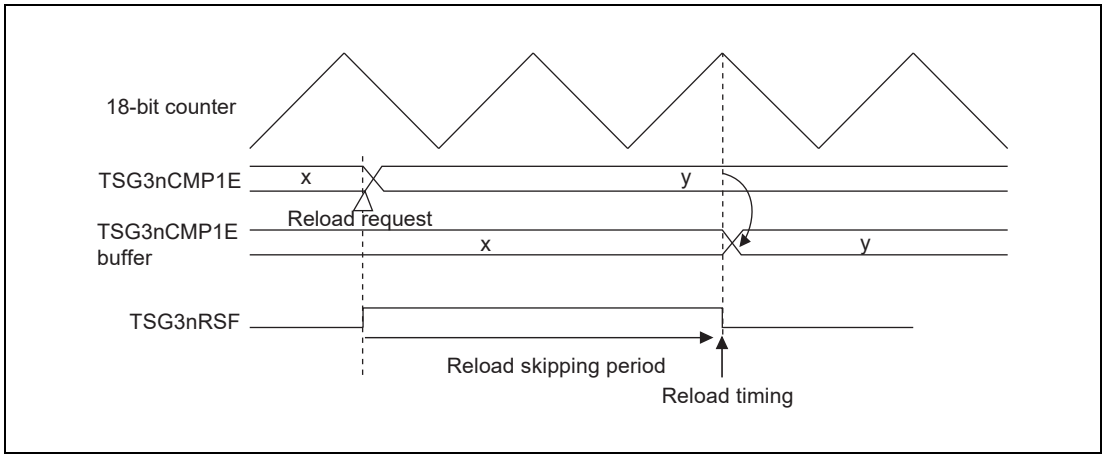


Figure 35.21 Reload Request Flag and Reload Skipping Period

Operating Mode

Available in all operating modes.

35.4.3.4 Noise Detection Flag (TSG3nNDF)

Name

Noise detection flag (TSG3nSTR2.TSG3nNDF)

Description

TSG3nNDF can detect that two or more pins of TSG3nPTSI2 to TSG3nPTSI0 have changed simultaneously (a noise is generated).

TSG3nNDF is set to 1 when two or more pins of TSG3nPTSI2 to TSG3nPTSI0 have changed simultaneously (a noise is generated), and a warning interrupt (INTTSG3nIWN) is generated. The TSG3nNDF flag is cleared to 0 when 1 is written to the TSG3nSTC.TSG3nNDR bit.

Example of operation

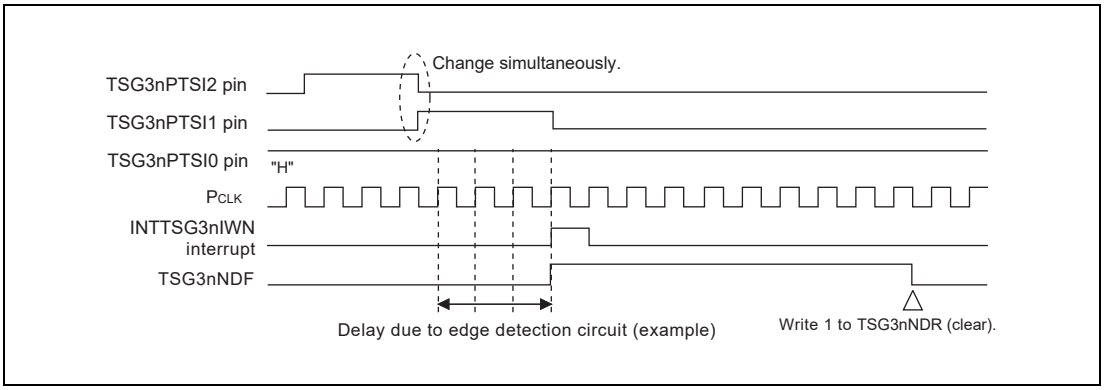


Figure 35.22 Example of Noise Detection Flag Operation

Operation mode

Available in all operating modes.

CAUTION

TSG3nNDF is valid only when TSG3nCTL1.TSG3nNDC = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.3.5 Pattern Order Detection Flag (TSG3nTSF)

Name

Pattern order detection flag (TSG3nSTR1.TSG3nTSF)

Description

TSG3nTSF can detect the order of patterns input to the TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nTSF is set depending on the values input to the TSG3nPTSI2 to TSG3nPTSI0 pins as shown in the table below

Table 35.51 Pattern Order Detection Flag and Pattern Input Order

TSG3nTSF	Values input to TSG3nPTSI2 to TSG3nPTSI0 Pins
0	[1,0,1] → [1,0,0] → [1,1,0] → [0,1,0] → [0,1,1] → [0,0,1]
1	[1,0,1] ← [1,0,0] ← [1,1,0] ← [0,1,0] ← [0,1,1] ← [0,0,1]

Example of operation

(a) When Normal Input to TSG3nPTSI2 to TSG3nPTSI0 Pins is Detected

As shown in **Figure 35.23**, if the TSG3nPTSI2 to TSG3nPTSI0 pins change in the normal order, 0 or 1 is set according to the change order at the change timing.

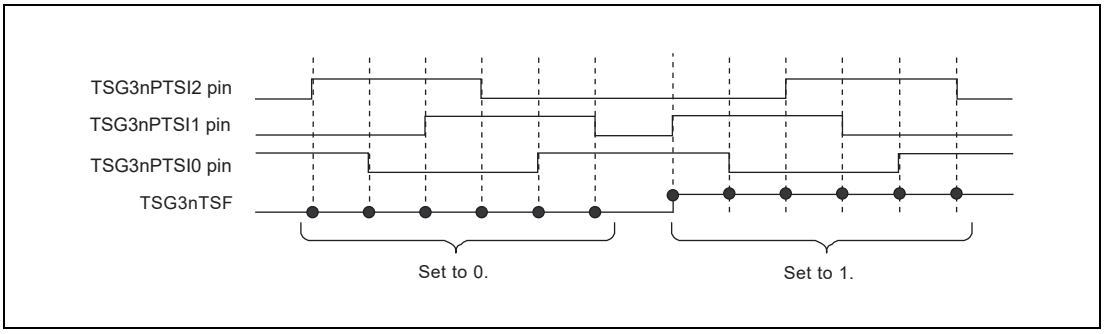


Figure 35.23 Example of Pattern Order Detection Flag Operation (Normal Operation)

(b) Detection of Change in Input Pattern Order

Immediately after TSG3n starts operation, the rotation direction cannot be determined. Therefore, TSG3nTSF cannot detect the change (normal or reverse rotation) in the patterns input to the TSG3nPTSI2 to TSG3nPTSI0 pins. To enable detection of change immediately after the beginning of operation, TSG3nPSC should be set before operation starts (when TSG3nTE = 0, the TSG3nPSC value is reflected).

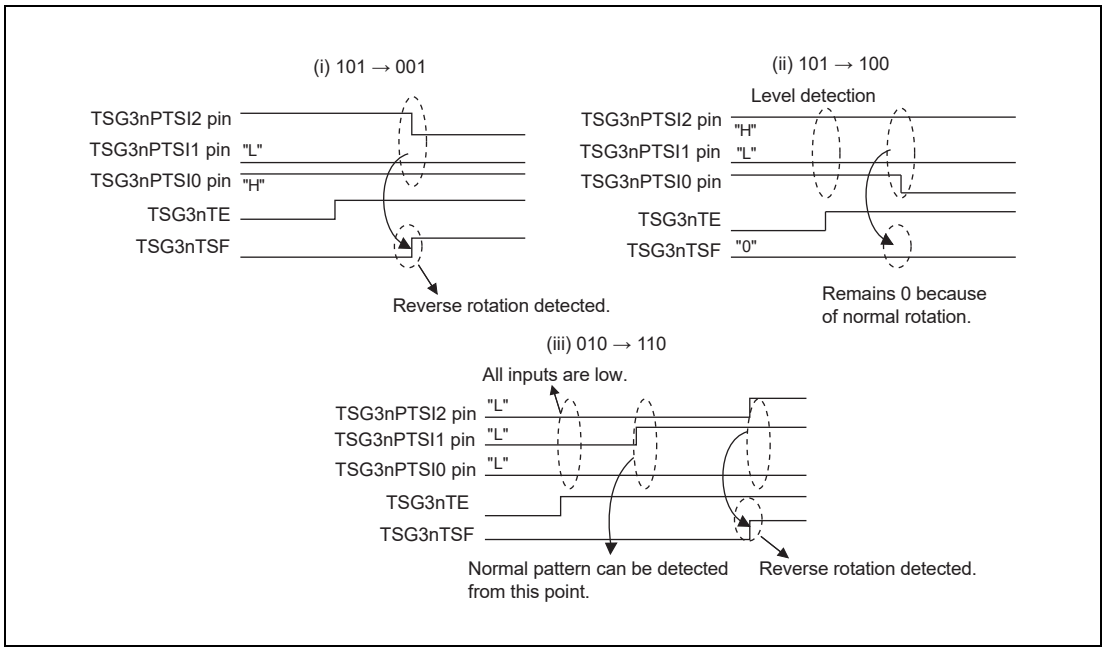


Figure 35.24 Example of Detecting Change (Normal/Reverse Rotation) in Pattern Input to TSG3nPTSI2 to TSG3nPTSI0 Pins

(c) When Abnormal Input to TSG3nPTSI2 to TSG3nPTSI0 Pins is Detected

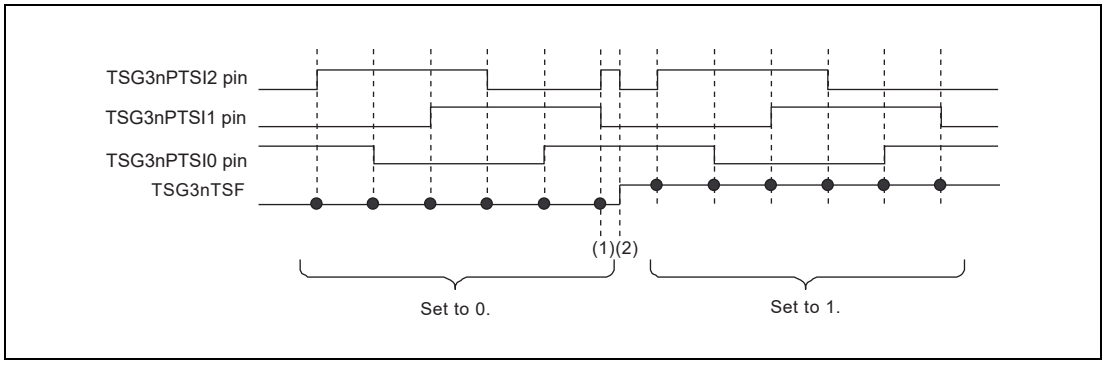


Figure 35.25 Example of Operation when Values Input to Two Pins of TSG3nPTSI2 to TSG3nPTSI0 Change (Abnormal Operation)

(1) TSG3nTSF does not change at this point because it expects the input pattern change to {0, 1, 0} or {0, 0, 1} (if values of two pins change, TSG3nTSF does not change).

(2) TSG3nPTSI2 to TSG3nPTSI0 pins are determined to have been changed from {1, 0, 1} to {0, 0,1}, and TSG3nTSF is set to 1.

Operation mode

Available in all operating modes.

35.4.3.6 Pattern Error Detection Flag (TSG3nPEF)

Name

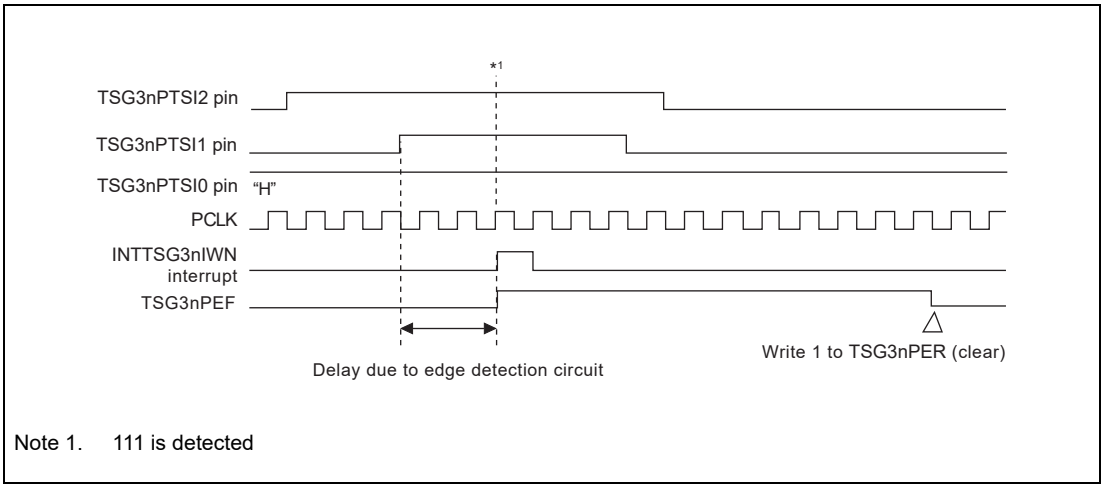
Pattern error detection flag (TSG3nSTR2.TSG3nPEF)

Description

TSG3nPEF can detect that 000 or 111 is input to the TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nPEF is set to 1 when the levels of the TSG3nPTSI2 to TSG3nPTSI0 pins are 111 or 000, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPEF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPER.

Example of operation



Note 1. 111 is detected

Figure 35.26 Example of Pattern Error Detection Flag Operation (TSG3nPTSI2 to TSG3nPTSI0 Pins = 111)

Operation mode

Available in all operating modes.

CAUTIONS

TSG3nPEF is valid only when TSG3nCTL1.TSG3nPEC = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.3.7 Pattern Reversal Detection Flag (TSG3nPRF)

Name

Pattern reversal detection flag (TSG3nSTR2.TSG3nPRF)

Description

TSG3nPRF can detect that the pattern change order of the TSG3nPTSI2 to TSG3nPTSI0 pins have been reversed.

TSG3nPRF is set to 1 when the pattern order detection flag (TSG3nTSF) changes, and a warning interrupt (INTTSG3nIWN) is generated. However, immediately after TSG3nSTR0.TSG3nTE is set to 1, TSG3nPRF becomes valid at the timing of the second and subsequent change in TSG3nPTSI2 to TSG3nPTSI0 pins.

TSG3nPRF is cleared to 0 when 1 is written to the TSG3nSTC.TSG3nPRR bit.

Example of operation

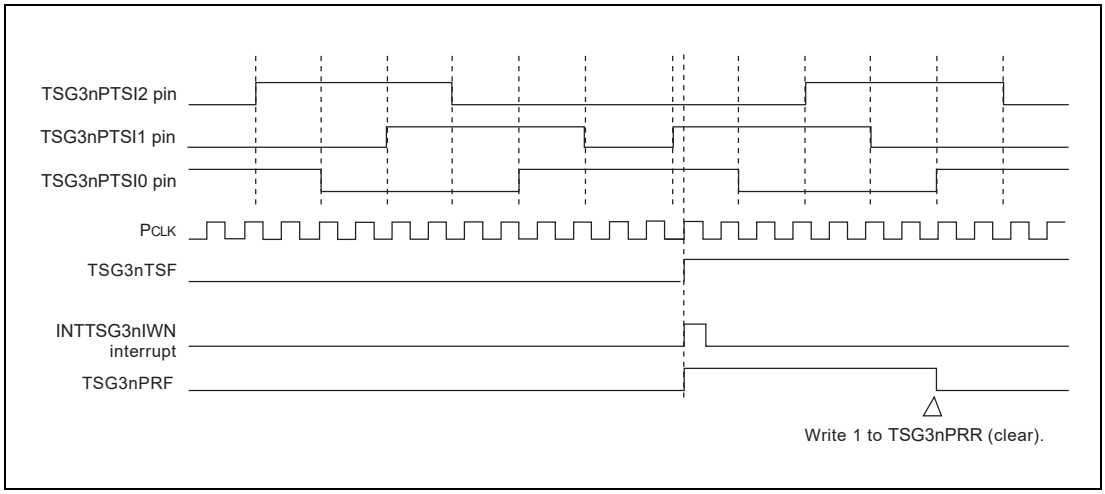
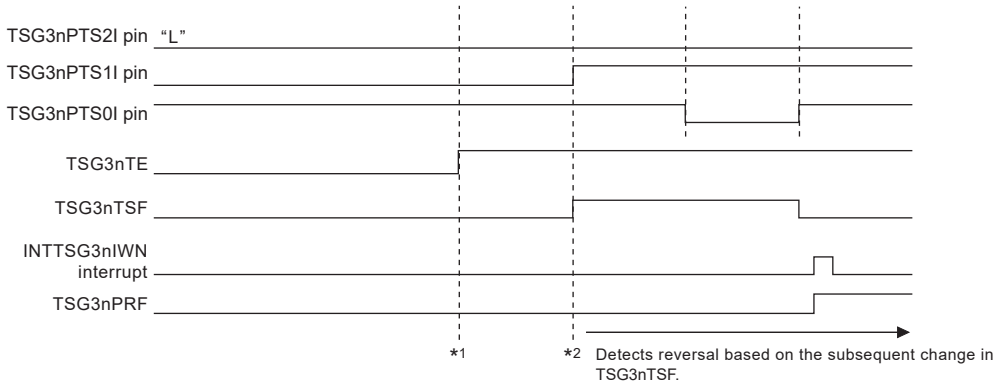


Figure 35.27 Example of Pattern Reversal Detection Flag Operation



Note 1. Operation starts.

Note 2. If TSG3nTSF is set to 1 by the first change in the TSG3nPTSI2 to TSG3nPTSI0 pins immediately after TSG3nTE is set to 1, reversal is not detected.

Figure 35.28 Example of Operation immediately after TSG3nTE Flag in TSG3nSTR0 is Set to 1

Operation mode

Available in all operating modes.

CAUTION

TSG3nPRF is valid only when TSG3nCTL1.TSG3nPRC = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.3.8 TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag (TSG3nPTF)

Name

TSG3nPTSI2 to TSG3nPTSI0 pin abnormal toggle detection flag (TSG3nSTR2.TSG3nPTF)

Description

TSG3nPTF can detect that the values of the TSG3nPTSI2 to TSG3nPTSI0 pins change three or more times during the TSG3nOPCI0 or TSG3nOPCI1 signal trigger.

TSG3nPTF is set to 1 when the third trigger of TSG3nOPCI0 or TSG3nOPCI1 signal occurs simultaneously with the change in TSG3nPTSI2 to TSG3nPTSI0, and a warning interrupt (INTTSG3nIWN) is generated.

TSG3nPTF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPTR.

Example of operation

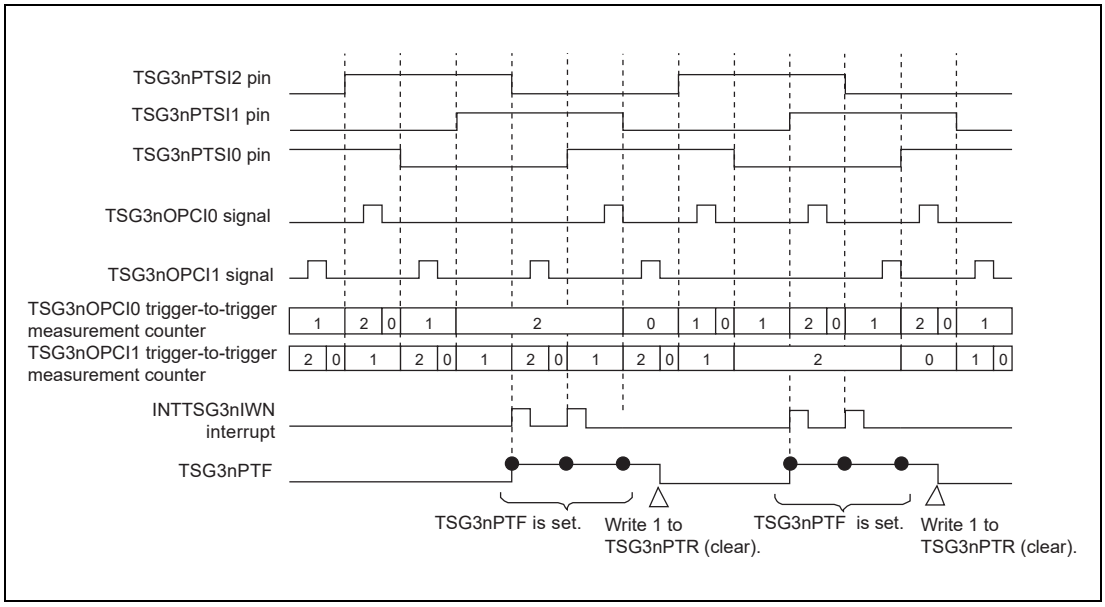


Figure 35.29 Example of TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag Operation

Operating mode

Available in all operating modes.

NOTES

1. TSG3nPTF is valid only when TSG3nCTL1.TSG3nPTC1 bit = 1 and TSG3nSTR0.TSG3nTE = 1.
2. When TSG3nPTC0 bit = 1 and TSG3nPTC1 bit = 1, TSG3nO1 to TSG3nO6 pin output switch control is automatically switched to pattern switch method (TSG3nOPT0.TSG3nPOT bit = 0) if an abnormal toggle is detected.

35.4.3.9 TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag (TSG3nTDF)

Name

TSG3nOPCI0 and TSG3nOPCI1 signal simultaneous trigger detection flag (TSG3nSTR2.TSG3nTDF)

Description

TSG3nTDF can detect that TSG3nOPCI0 and TSG3nOPCI1 signals are generated simultaneously.

TSG3nTDF is set to 1 when the TSG3nOPCI0 and TSG3nOPCI1 signals are generated simultaneously, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nTDF is cleared to 0 when 1 is written to TSG3nSTC.TSG3nTDR.

Example of operation

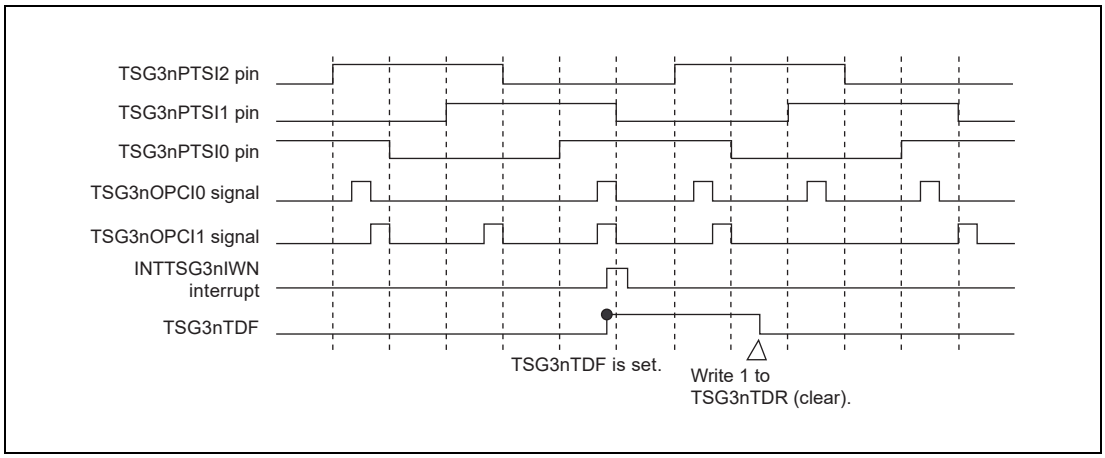


Figure 35.30 Example of TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag Operation

Operating mode

Available in all operating modes.

CAUTION

TSG3nTDF is valid only when TSG3nCTL1.TSG3nTDC = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.3.10 Pattern Phase Difference Detection Flag (TSG3nPPF)

Name

Pattern phase difference detection flag (TSG3nSTR2.TSG3nPPF)

Description

TSG3nPPF can detect the phase difference between the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) and the output pattern (TSG3nSTR1.TSG3nOPF2 to TSG3nOPF0 flags).

TSG3nPPF is set to 1 when the pattern phase difference is detected when the TSG3nOPCI0 and TSG3nOPCI1 signal triggers are input, and a warning interrupt (INTTSG3nIWN) is generated. TSG3nPPF remains 1 until it is cleared to 0 when 1 is written to TSG3nSTC.TSG3nPPR by software. When the phase difference is detected, TSG3nPPF is set at each operation clock cycle (PCLK). TSG3nPPF should be cleared to 0 when no phase difference occurs.

Table 35.52 Correspondence between Normal Input Patterns and Output Patterns

TSG3nPTSI2 to TSG3nPTSI0 pins (input)	“1,0,1”	“1,0,0”	“1,1,0”	“0,1,0”	“0,1,1”	“0,0,1”
TSG3nOPF2 to TSG3nOPF0 flags (output)	“0,0,1” “1,0,1” “1,0,0”	“1,0,1” “1,0,0” “1,1,0”	“1,0,0” “1,1,0” “0,1,0”	“1,1,0” “0,1,0” “0,1,1”	“0,1,0” “0,1,1” “0,0,1”	“0,1,1” “0,0,1” “1,0,1”

Example of operation

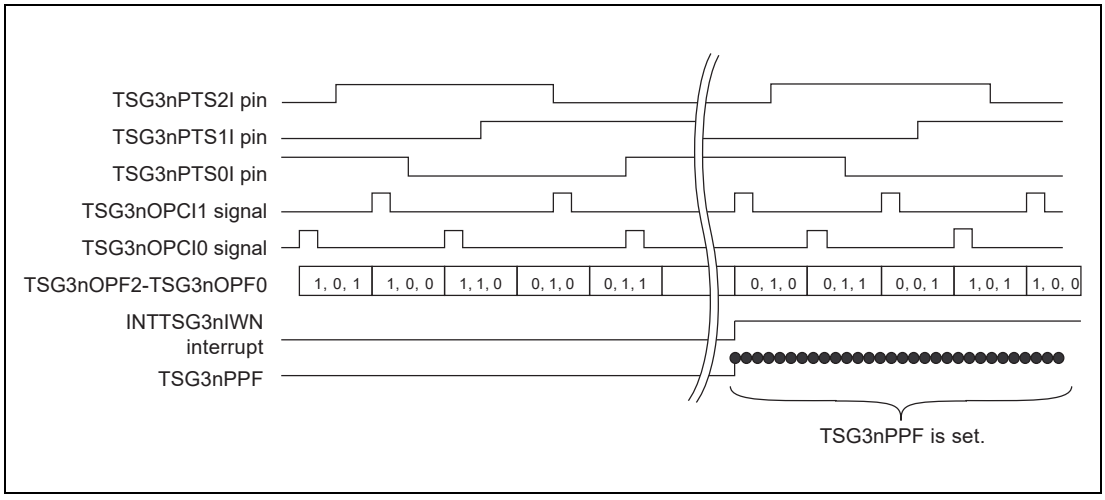


Figure 35.31 Example of Pattern Difference Detection Flag Operation

Operating mode

Available in all operating modes.

CAUTIONS

1. **TSG3nPPF is valid only when TSG3nCTL1.TSG3nPPC = 1 and TSG3nSTR0.TSG3nTE = 1.**
2. **When 000 or 111 is input to the TSG3nPTSI2 to TSG3nPTSI0 pins, or when TSG3nOPF2 to TSG3nOPF0 are set to 000 or 111, TSG3nPPF is not set.**

35.4.3.11 Timer Output Pattern Flag (TSG3nOPF2-TSG3nOPF0)

Name

Timer output pattern flag (TSG3nSTR1.TSG3nOPF2 to TSG3nOPF0)

Description

TSG3nOPF2 to TSG3nOPF0 flags indicate the timer output patterns.

For details, see **Section 35.4.7.4, 120-DC Mode**, and **Section 35.4.7.8, Software Output Control Function**.

Operating mode

Available in all operating modes.

35.4.3.12 Pattern Switch Detection Signal (TSG3nPTE)

Name

Pattern switch detection signal (TSG3nPTE signal)

Description

The TSG3nPTE signal toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes.

The toggle pattern is determined by the TSG3nPSC bit (TSG3nOPT0.TSG3nPSS = 1).

Table 35.53 Change Timing of Pattern Switch Detection Signal (1/2)

- TSG3nPSC = 0

		TSG3nPTSI2-TSG3nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	Toggle	—	—	—	—
	100	—	—	—	—	Toggle	—	—	—
	110	—	—	—	—	—	Toggle	—	—
	010	—	—	—	—	—	—	Toggle	—
	011	—	—	—	—	—	—	—	Toggle
	001	—	—	Toggle	—	—	—	—	—

Table 35.53 Change Timing of Pattern Switch Detection Signal (2/2)

- TSG3nPSC = 1

		TSG3nPTSI2-TSG3nPTSI0 Pins after Change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	—	—	—	—	Toggle
	100	—	—	Toggle	—	—	—	—	—
	110	—	—	—	Toggle	—	—	—	—
	010	—	—	—	—	Toggle	—	—	—
	011	—	—	—	—	—	Toggle	—	—
	001	—	—	—	—	—	—	Toggle	—

Example of operation

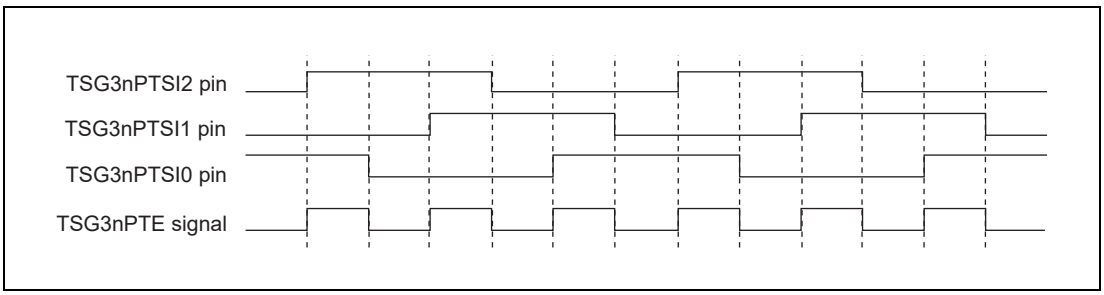


Figure 35.32 Example of Pattern Switch Detection Signal Operation

Operating mode

Available in all operating modes.

CAUTION

The TSG3nPTE signal is valid only when TSG3nIOC1.TSG3nPTS = 1 and TSG3nSTR0.TSG3nTE = 1.

35.4.4 Interrupt Skipping Function

Operation related to the interrupt skipping function is described below.

- Peak interrupts (INTTSG3nIPEK) and valley interrupts (INTTSG3nIVLY) can be skipped.
- TSG3nCTL4.TSG3nPIE enables outputting of the INTTSG3nIPEK interrupt and specifies whether to skip the interrupts.
- TSG3nCTL4.TSG3nVIE enables outputting of the INTTSG3nIVLY interrupt and specifies whether to skip the interrupts.

When TSG3nCTL3.TSG3nRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

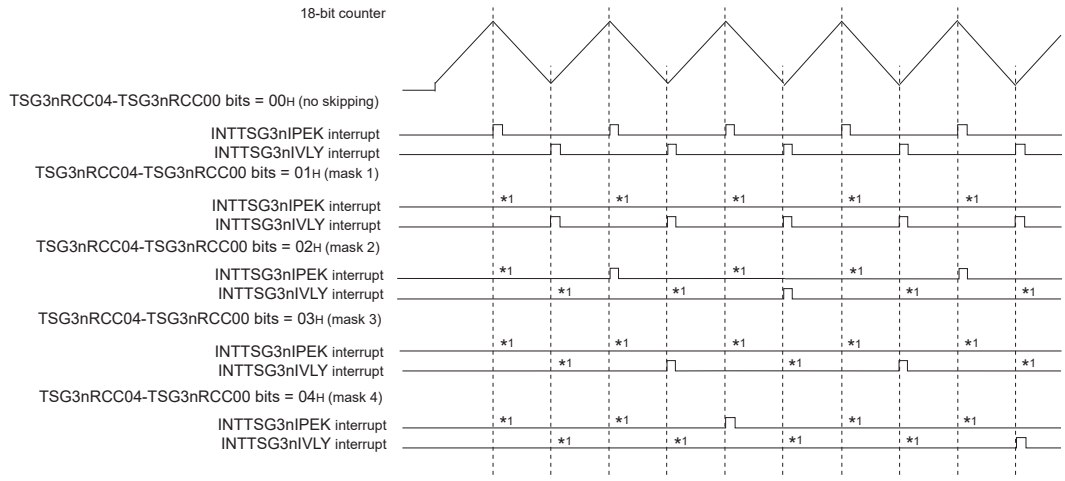
When TSG3nCTL3.TSG3nRIA is set to 0 (without reload skipping), reload is executed at the specified reload timing regardless of interrupt skipping.

CAUTION

When a value is written to TSG3nCTL4, and TSG3nRCC04 to TSG3nRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may temporarily become longer. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSG3nCTL3.TSG3nRIA = 1).

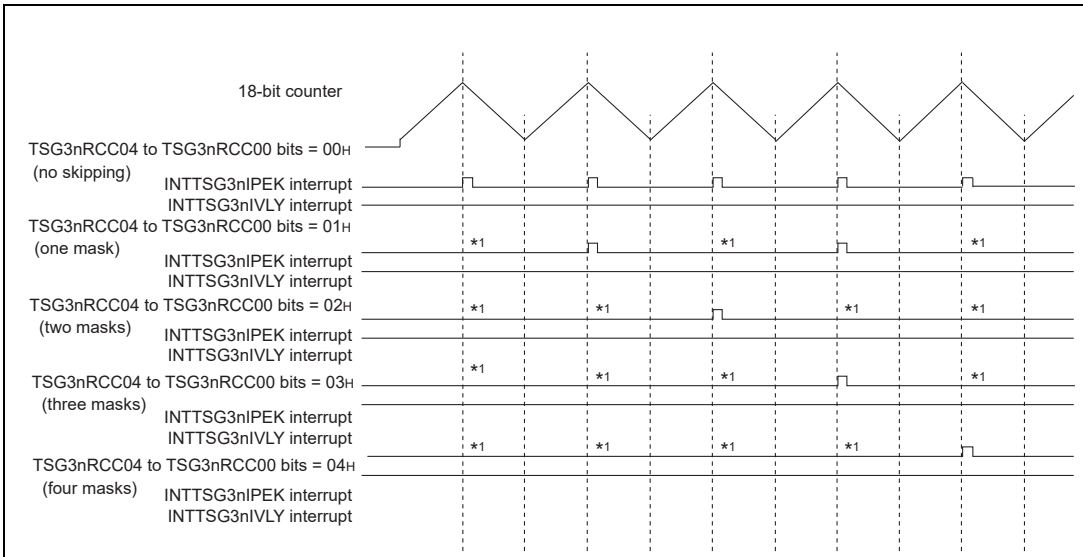
35.4.4.1 Operation of Interrupt Skipping Function

Timing diagram of interrupt skipping function in various conditions are shown in the following figures.



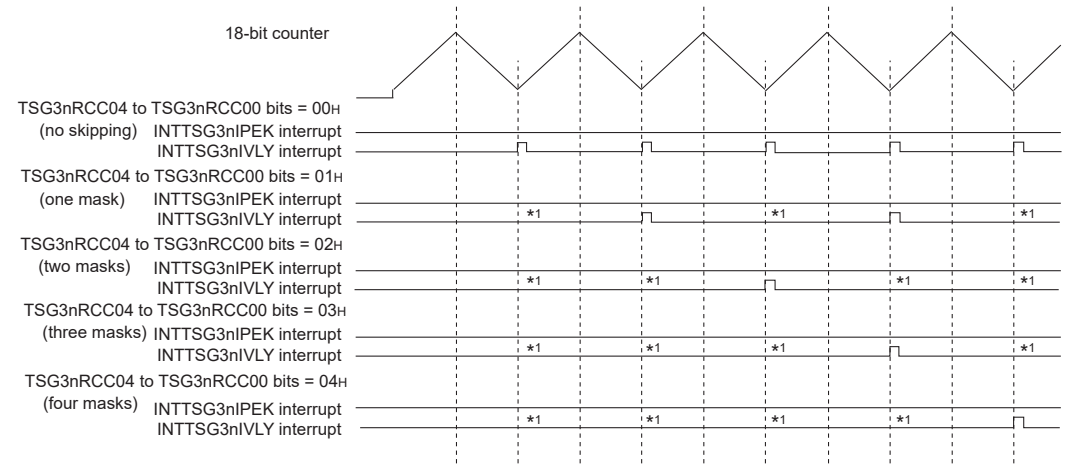
Note 1. Skipped interrupt request

Figure 35.33 Interrupt Skipping Operation when TSG3nPIE = 1 and TSG3nVIE = 1 in TSG3nCTL4 Register (Peak and Valley Interrupt Generation in HT-PWM Mode)



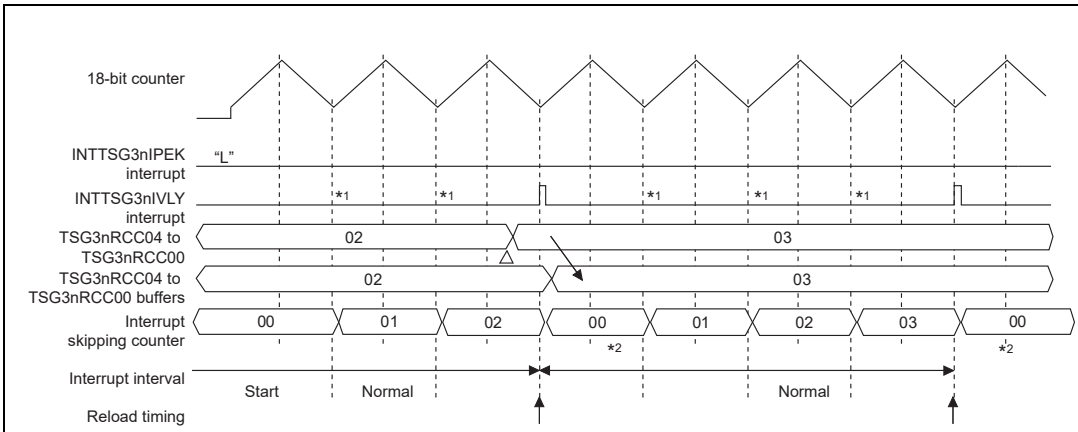
Note 1. Skipped interrupt request

Figure 35.34 Interrupt Skipping Operation when TSG3nPIE = 1 and TSG3nVIE = 0 in TSG3nCTL4 Register (only Peak Interrupt Generation in HT-PWM Mode)



Note 1. Skipped interrupt request

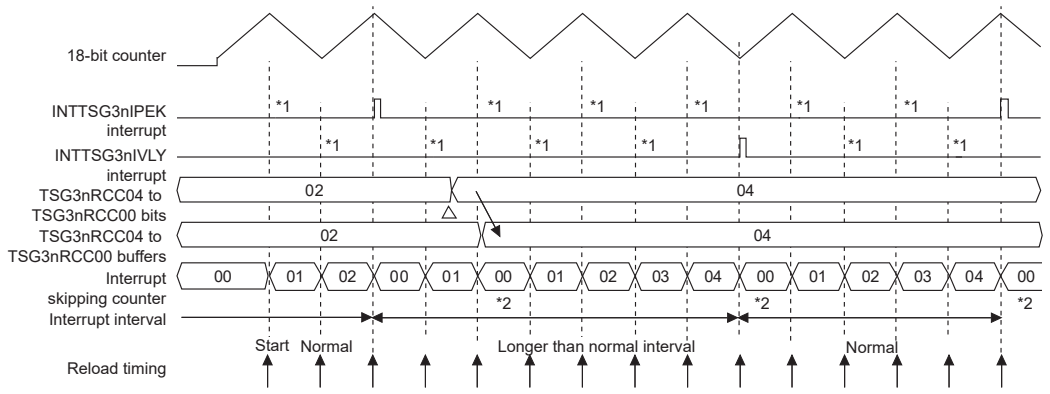
Figure 35.35 Interrupt Skipping Operation when TSG3nPIE = 0 and TSG3nVIE = 1 in TSG3nCTL4 Register (only Valley Interrupt Generation in HT-PWM Mode)



Note 1. Skipped interrupt request

Note 2. Interrupt skipping counter is cleared

Figure 35.36 When TSG3nRMC = 0, TSG3nRIA = 1 in TSG3nCTL3 Register (with Reload Skipping)

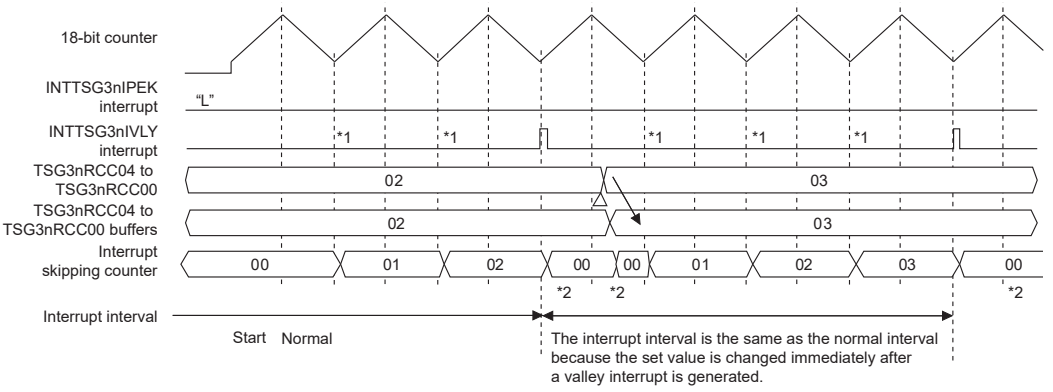


Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared

Figure 35.37 When TSG3nRMC = 0, TSG3nRIA = 0 in TSG3nCTL3 Register (without Reload Skipping)

CAUTION

Interrupt interval might be longer.



Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared

Figure 35.38 When TSG3nRMC = 1 in TSG3nCTL3 Register (Anytime Rewrite Mode)

NOTE

After rewriting, the value is reflected immediately regardless of the reload timing. The interrupt skipping counter is cleared when the value is transferred to the TSG3nRCC04 to TSG3nRCC00 buffers, not when the pertinent register is rewritten.

35.4.4.2 Example of Operation when Peak Interrupt is Generated (in PWM Mode)

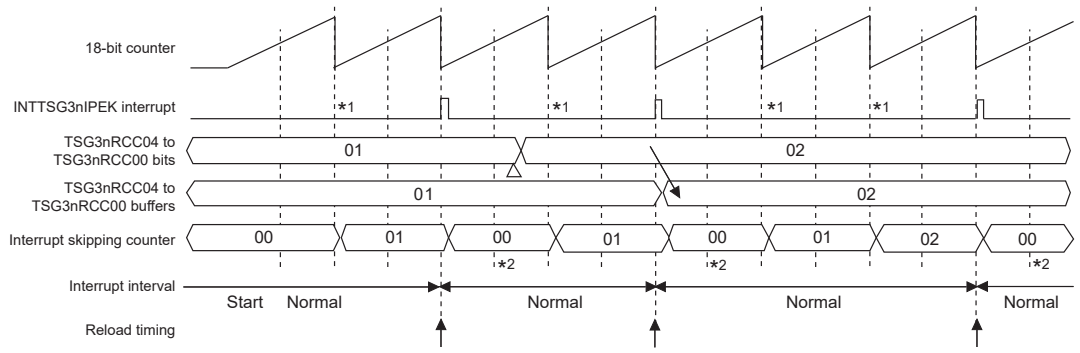
Operation related to the interrupt skipping function in PWM mode is described below.

- Peak interrupts (INTTSG3nIPEK) can be skipped. In PWM mode, it is generated by compare match of TSG3nCMP0E buffer register and 18-bit counter.
- TSG3nCTL4.TSG3nPIE enables outputting of the INTTSG3nIPEK interrupt and specifies whether to skip the interrupts.
- The setting of TSG3nCTL4.TSG3nVIE is disabled. At this time, the INTTSG3nIVLY interrupt is not generated.
  When TSG3nCTL3.TSG3nRIA is set to 1 (with reload skipping), reload is executed at the same timing as the interrupt after being skipped.

CAUTION

When a value is written to TSG3nCTL4, and TSG3nRCC04 to TSG3nRCC00 are transferred to the buffer register, the interrupt skipping counter is cleared. Therefore, when the interrupt skipping function is used, interrupt interval may be long temporarily. To avoid this, the interrupt skipping count should be changed with the reload timing being set to the interrupts skipped (TSG3nCTL3.TSG3nRIA = 1).

(1) Example of operation



Note 1. Skipped interrupt request
Note 2. Interrupt skipping counter is cleared.

Figure 35.39 When TSG3nCTL3.TSG3nRMC = 0, TSG3nRIA = 1, TSG3nCTL4.TSG3nPRE = 1 (Recommended Setting)

NOTE

When TSG3nCTL3.TSG3nRIA = 1, reload is executed at the same timing as the interrupt after being skipped.

35.4.5 A/D Conversion Trigger Function

A/D conversion trigger operation is described below.

The TSG3nDCMP0E, TSG3nDCMP1E, and TSG3nDCMP2E registers are used as compare registers of the A/D conversion trigger function.

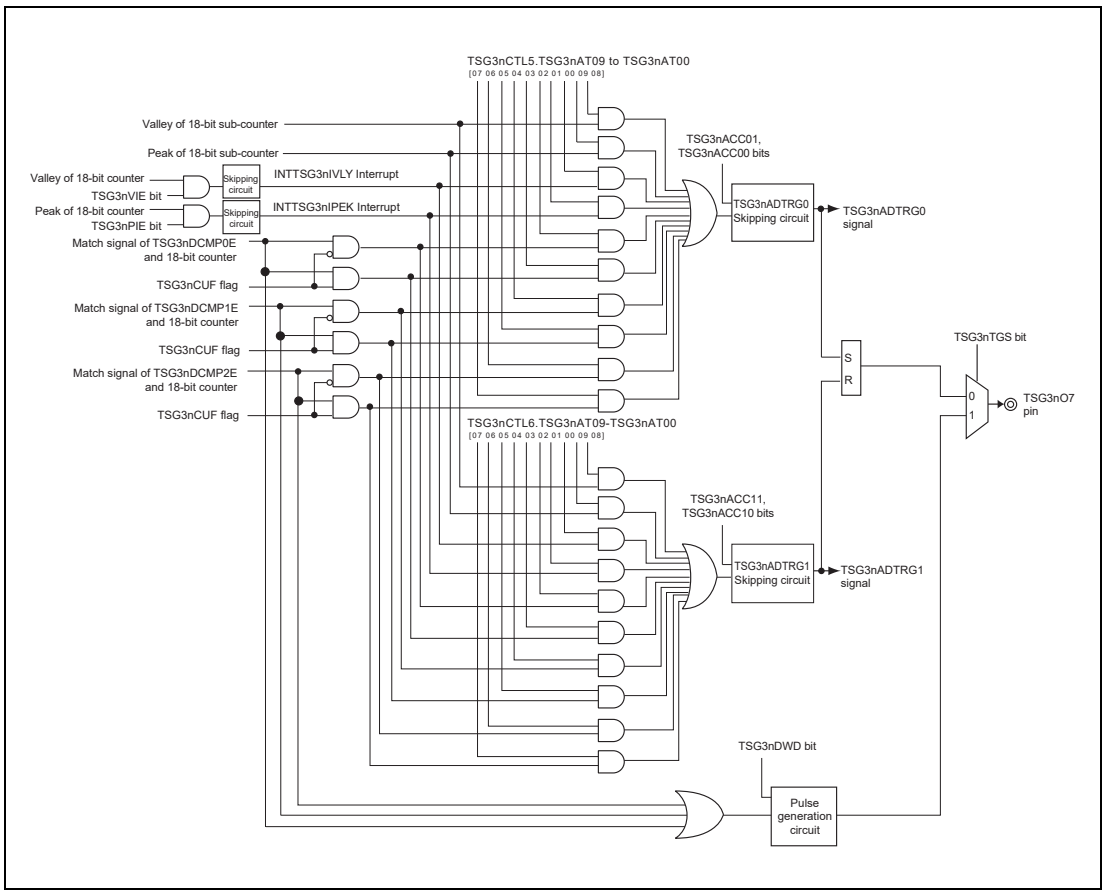


Figure 35.40 A/D Conversion Trigger and Diagnostic Output Control Circuit

As shown in **Figure 35.40**, a logical ORed signal can be generated by selecting the compare match of TSG3nDCMP0E to TSG3nDCMP2E with the 18-bit counter, a peak interrupt (INTTSG3nIPEK) and a valley interrupt (INTTSG3nIVLY) of the 18-bit counter, the peak timing of 18-bit sub-counter, and the valley timing of 18-bit sub-counter.

TSG3n has two channels of the identical A/D conversion trigger control circuits, which can be controlled independently. TSG3n also provides the A/D conversion trigger skipping function with the skipping rate of 1/1 (no skipping), 1/2, 1/4, or 1/8.

35.4.5.1 Operation of A/D Conversion Trigger

TSG3n has a function to generate A/D conversion start triggers (TSG3nADTRG0 and TSG3nADTRG1 signals) by selecting any of ten trigger sources as required. The trigger sources are selected by TSG3nAT09 to TSG3nAT00 in TSG3nCTL5 and TSG3nAT19 to TSG3nAT10 in TSG3nCTL6.

(1) TSG3nADTRG0/TSG3nADTRG1 Signal Output Control (TSG3nCTL5 and TSG3nCTL6)

[Trigger sources]

- TSG3nAT00/TSG3nAT10 = 1 : A valley interrupt (INTTSG3nIVLY) causes an A/D conversion trigger pulse to be generated.
- TSG3nAT01/TSG3nAT11 = 1 : A peak interrupt (INTTSG3nIPEK) causes an A/D conversion trigger pulse to be generated.
- TSG3nAT02/TSG3nAT12 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP0E compare match enables A/D conversion trigger to be generated.
- TSG3nAT03/TSG3nAT13 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP0E compare match enables A/D conversion trigger to be generated.
- TSG3nAT04/TSG3nAT14 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP1E compare match enables A/D conversion trigger to be generated.
- TSG3nAT05/TSG3nAT15 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP1E compare match enables A/D conversion trigger to be generated.
- TSG3nAT06/TSG3nAT16 = 1 : While the 18-bit counter is counting up, a TSG3nDCMP2E compare match enables A/D conversion trigger to be generated.
- TSG3nAT07/TSG3nAT17 = 1 : While the 18-bit counter is counting down, a TSG3nDCMP2E compare match enables A/D conversion trigger to be generated.
- TSG3nAT08/TSG3nAT18 = 1 : A valley timing of the 18-bit sub-counter (at a switch from decrementing to incrementing) enables A/D conversion trigger to be generated.
- TSG3nAT09/TSG3nAT19 = 1 : A peak timing of 18-bit sub-counter (at a switch from incrementing to decrementing) enables A/D conversion trigger to be generated.

[Skipping setting]

- TSG3nACC01, TSG3nACC00 and TSG3nACC11, TSG3nACC10 :
  Set the skipping rate of TSG3nADTRG0/TSG3nADTRG1

All A/D conversion triggers selected by TSG3nAT09 to TSG3nAT00 and TSG3nAT19 to TSG3nAT10 are logically ORed, and the resultant signals are subjected to skipping control specified by TSG3nACC01 and TSG3nACC00, and TSG3nACC11 and TSG3nACC10, and then the TSG3nADTRG0 and TSG3nADTRG1 signals are generated.

A peak interrupt (INTTSG3nIPEK) and a valley interrupt (INTTSG3nIVLY) selected by TSG3nAT00 and TSG3nAT01, and TSG3nAT10 and TSG3nAT11 are interrupt signals obtained after skipping. Therefore, they are output at the timing according to interrupt skipping control. If the interrupt output is not enabled by TSG3nCTL4.TSG3nPIE and TSG3nVIE, A/D conversion trigger is not output.

TSG3nACC01, TSG3nACC00, and TSG3nAT09 to TSG3nAT00, and TSG3nACC11, TSG3nACC10, and TSG3nAT19 to TSG3nAT10 can be rewritten during timer operation.

If A/D conversion trigger setting bits are rewritten during operation, the rewritten values are reflected on the A/D conversion trigger output status immediately. Such control bits are rewritten anytime regardless of operating modes. If a write access is made to TSG3nCTL5 and TSG3nCTL6 (including a rewrite of the same value), the A/D conversion trigger skipping counter is cleared and starts counting from zero.

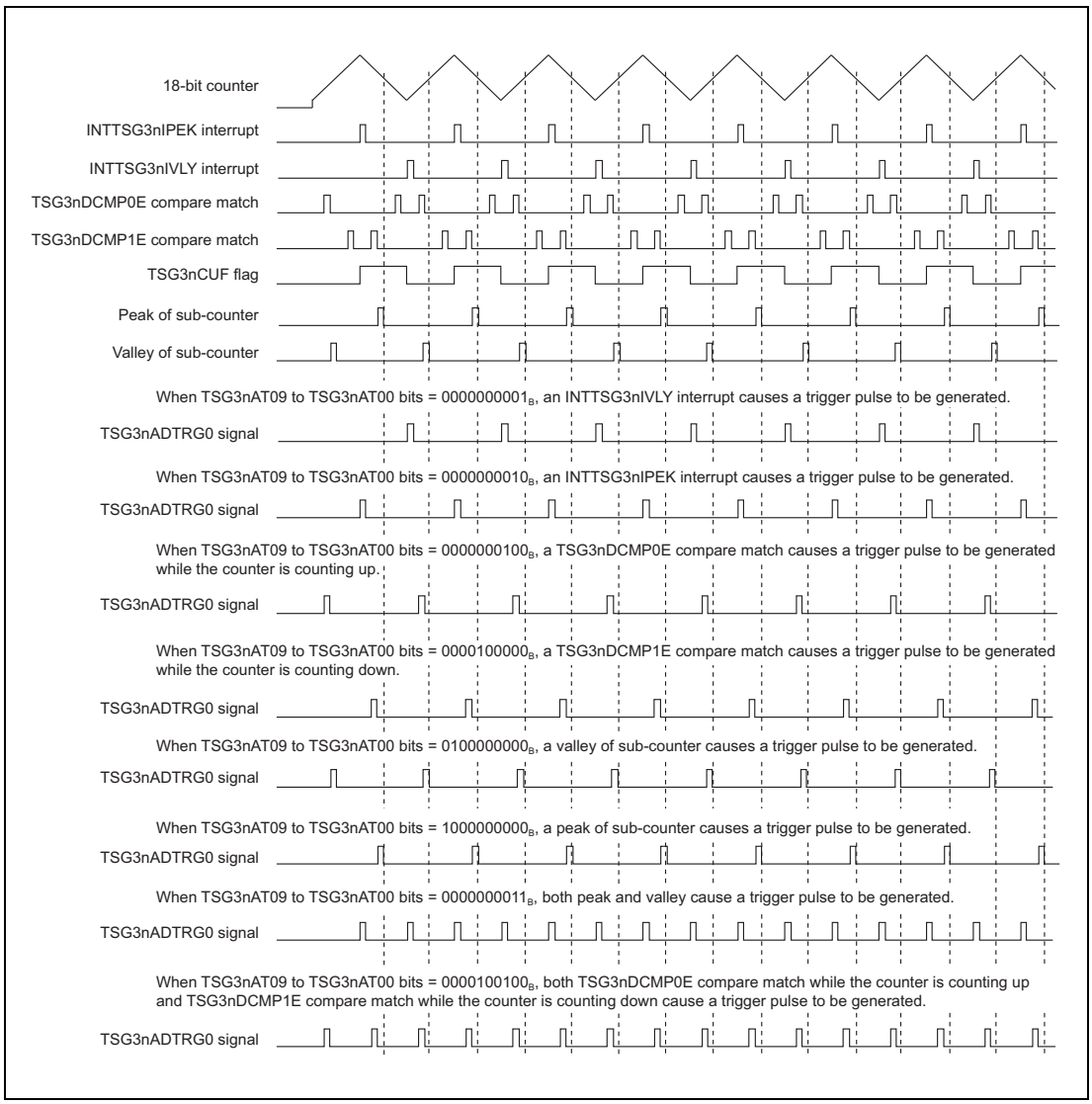
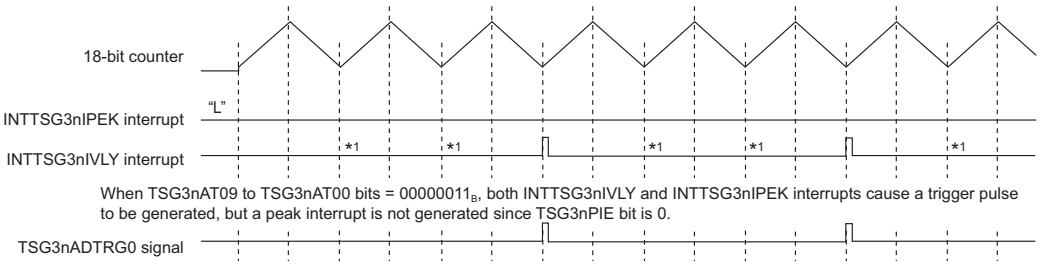
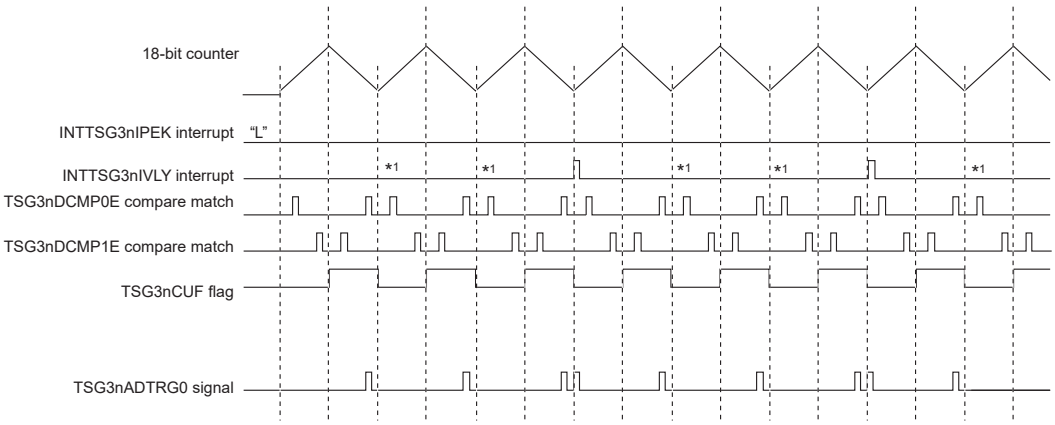


Figure 35.41 When TSG3nPIE = 1, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 00_H in TSG3nCTL4, and TSG3nACC01 and TSG3nACC00 = 00_B in TSG3nCTL5 (HT-PWM Mode)



Note 1. Skipped interrupt request

Figure 35.42 When TSG3nPIE = 0, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 02_H in TSG3nCTL4 and TSG3nACC01 and TSG3nACC00 = 00_B in TSG3nCTL5 (HT-PWM Mode)



Note 1. Skipped interrupt request

Figure 35.43 When TSG3nPIE = 0, TSG3nVIE = 1, and TSG3nRCC04 to TSG3nRCC00 = 02_H in TSG3nCTL4 and TSG3nACC01 and TSG3nACC00 = 00_B, and TSG3nAT09 to TSG3nAT00 = 0000 1001_B in TSG3nCTL5 (HT-PWM Mode)

(2) A/D Conversion Trigger Skipping Function

Example of operation of the A/D conversion trigger skipping function is shown in **Figure 35.44**.

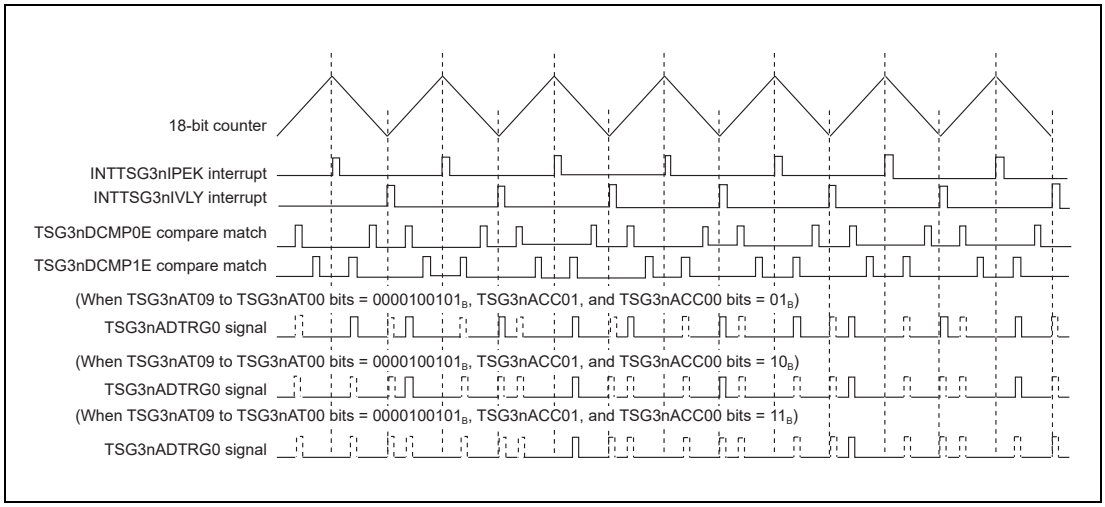


Figure 35.44 Example of Operation of A/D Conversion Trigger Skipping Function

NOTE

Broken-lined pulses indicate A/D conversion trigger pulses skipped by the A/D conversion trigger skipping function.

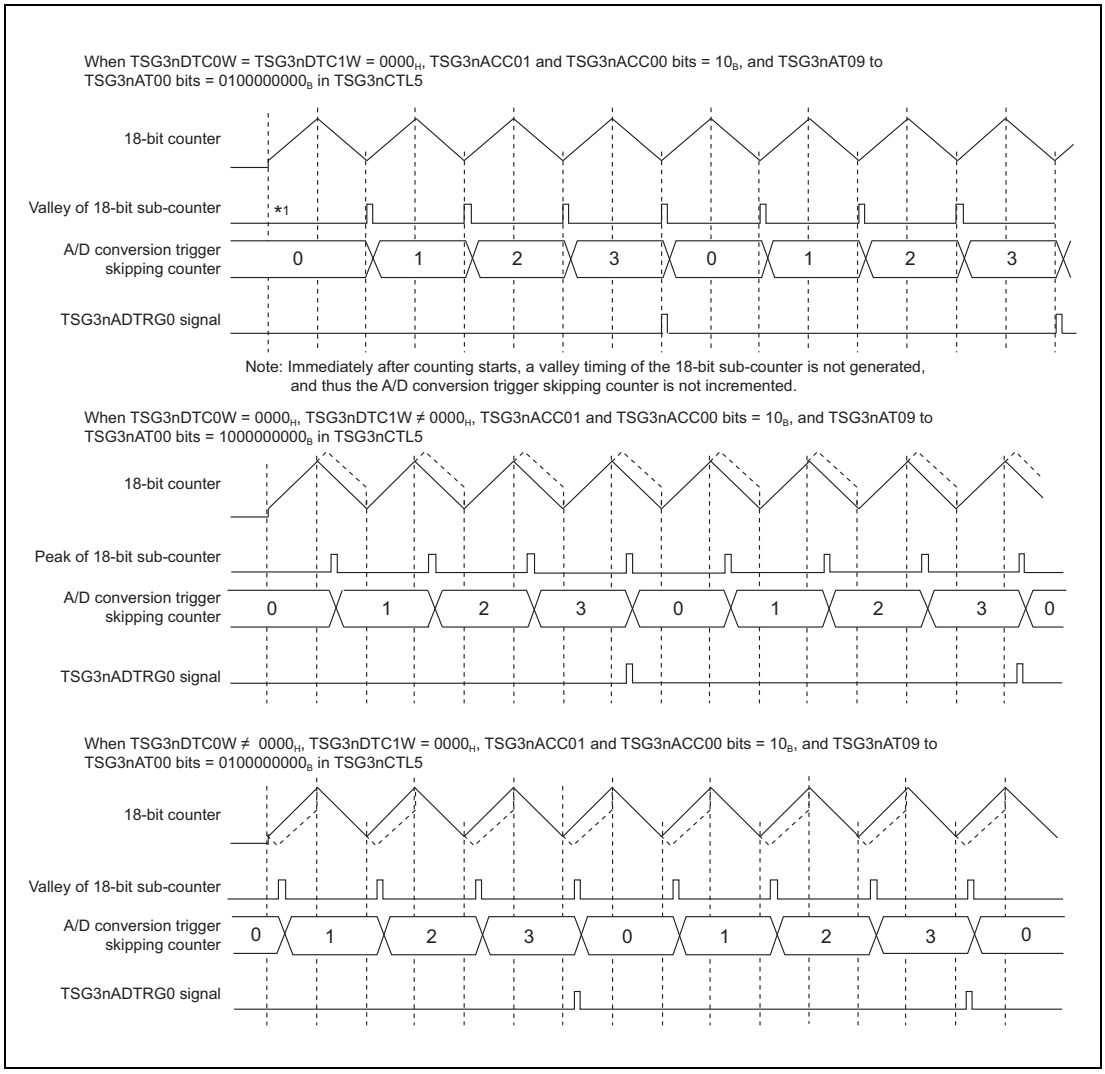


Figure 35.45 Example of Operation of A/D Conversion Trigger Skipping Function

(3) Notes on A/D Conversion Trigger

- If the same value is written to TSG3nDCMP0E, TSG3nDCMP1E or TSG3nDCMP2E, and the same condition (when the 18-bit counter increments or decrements) is set as the valid A/D conversion trigger, A/D conversion trigger skipping counter is incremented by one and one trigger pulse is output upon a match of the 18-bit counter with these registers.
- In PWM mode, SP-PWM mode, 120-DC mode, and HSP-PWM mode, a valley interrupt (INTTSG3nIVLY) is not generated. Only a peak interrupt (INTTSG3nIPEK) is valid.
- In 120-DC mode, when TSG3nS120DCO is set to 0, the 18-bit counter may be cleared during the carrier period due to switch of the output pattern. The A/D conversion trigger is not generated if TSG3nDCMP2E to TSG3nDCMP0E values do not match with the 18-bit counter value and a peak interrupt (INTTSG3nIPEK) is not generated.

35.4.6 Error/Warning Interrupt

35.4.6.1 Error Interrupt Function

If the simultaneous active state of the positive phase and inverse phase is detected after the error interrupt function is enabled (TSG3nIOC1.TSG3nEOC = 1), TSG3nSTR2.TSG3nTBF is set, and an error interrupt (INTTSG3nIER) of TSG3n is generated. Whether or not to detect an error of each phase (TSG3nO1 and TSG3nO2, TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 pins) can be selected by TSG3nCTL1.TSG3nTBA2 to TSG3nTBA0, respectively.

When an error occurs, outputs of the TSG3nO1 to TSG3nO6 pins can be set to high-impedance. For details, see **Section 36.4.1, Asynchronous Hi-Z Control Function**.

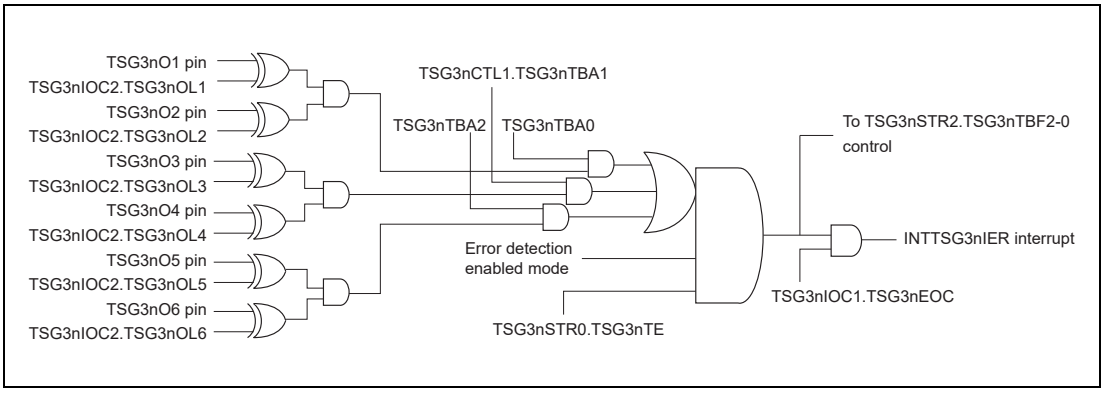


Figure 35.46 Error Interrupt (INTTSG3nIER) Generation Control Circuit

CAUTION

When an error interrupt is generated, the error status should be canceled (write 1 to TSG3nSTC.TSG3nTBR2 to TSG3nSTC.TSG3nTBR0) during error interrupt servicing. Otherwise, subsequent error interrupts are not generated.

(1) PWM Mode, 120-DC Mode and HSP-PWM Mode

In PWM mode and HSP-PWM mode, if TSG3nCMP1E and TSG3nCMP2E, and TSG3nCMP3E and TSG3nCMP4E are set so that the TSG3nO1 and TSG3nO2 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated. Likewise, if TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP9E, TSG3nCMP10E, TSG3nCMP11E, and TSG3nCMP12E are set so that the TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated.

In 120-DC mode, if TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E, TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E, TSG3nPAT0W, and TSG3nPAT1W are set so that the TSG3nO1 and TSG3nO2 pins output the active level simultaneously, an error interrupt (INTTSG3nIER) is generated. With the same setting, the TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 pins also output the active level simultaneously and an error interrupt (INTTSG3nIER) is generated.

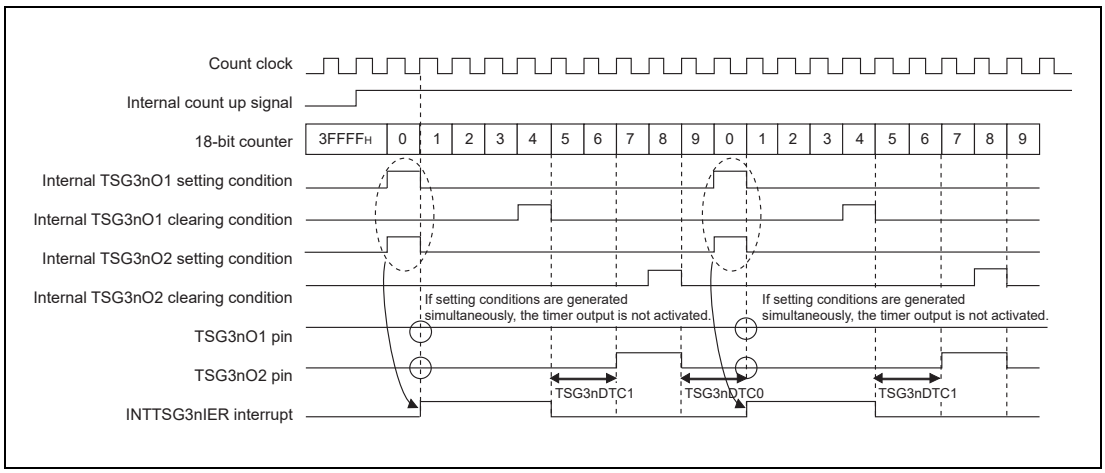


Figure 35.47 Example of Error Interrupt (INTTSG3nIER) Generation (PWM Mode)

NOTE

TSG3nO3 and TSG3nO4, and TSG3nO5 and TSG3nO6 behave the same.

When the active level of output is switched by manipulating TSG3nIOC2.TSG3nOL1 and TSG3nOL2, an error interrupt is generated as shown in the following figure.

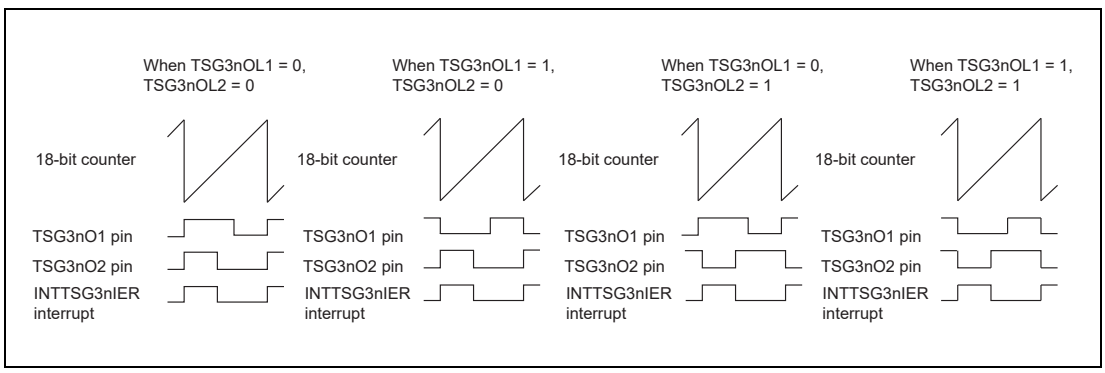


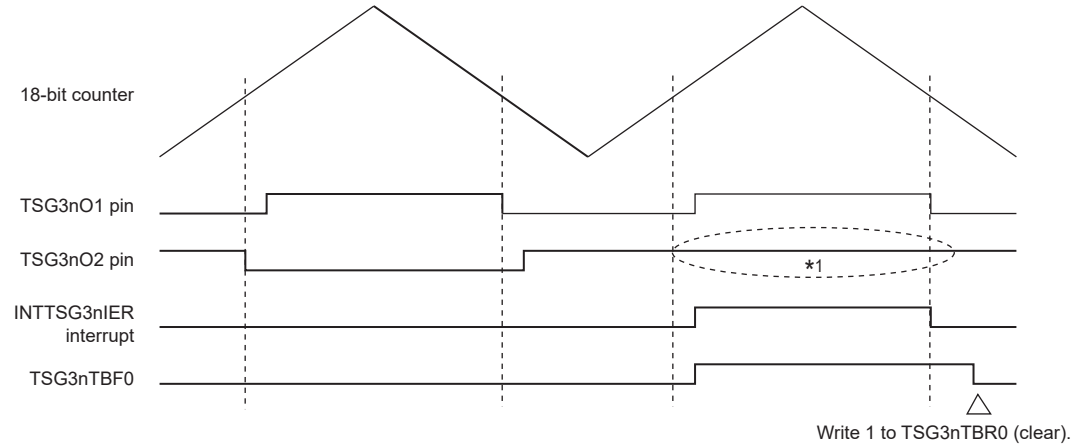
Figure 35.48 Example of Error Interrupt (INTTSG3nIER) Generation for each Active Level

(2) HT-PWM Mode and SP-PWM Mode

When either TSG3n dead time control register 0 or 1 (TSG3nDTC0W or TSG3nDTC1W) is 0000_H, an error may occur.

NOTE

If an error occurs when the dead time control function is used (both TSG3nDTC0W and TSG3nDTC1W are not 0000_H), internal circuit failure may have occurred.



Note 1. TSG3nO2 pin control circuit failure occurs.

Figure 35.49 Example of Error Interrupt Operation

35.4.6.2 Warning Interrupt Function

TSG3n has a warning interrupt (INTTSG3nIWN).

Warning interrupt (INTTSG3nIWN) is generated when any of the following conditions is detected.

For details, see **Section 35.4.3, Flags**.

- When simultaneous change in two or more pins of TSG3nPTSI2 to TSG3nPTSI0 is detected:
  See **Section 35.4.3.4, Noise Detection Flag (TSG3nNDF)**.
- When reversal is detected of the TSG3nPTSI2 to TSG3nPTSI0 pins:
  See **Section 35.4.3.7, Pattern Reversal Detection Flag (TSG3nPRF)**.
- When 000 or 111 is detected from the TSG3nPTSI2 to TSG3nPTSI0 pins:
  See **Section 35.4.3.6, Pattern Error Detection Flag (TSG3nPEF)**.
- When a toggle of the TSG3nPTSI2 to TSG3nPTSI0 pins is generated three or more times between TSG3nOPCI0 and TSG3nOPCI1 signal triggers.:
  See **Section 35.4.3.8, TSG3nPTSI2 to TSG3nPTSI0 Pin Abnormal Toggle Detection Flag (TSG3nPTF)**.
- When the TSG3nOPCI0 and TSG3nOPCI1 signal triggers are detected simultaneously:
  See **Section 35.4.3.9, TSG3nOPCI0 and TSG3nOPCI1 Signal Simultaneous Trigger Detection Flag (TSG3nTDF)**.
- When the phase difference between the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) and output pattern (TSG3nOPF2 to TSG3nOPF0) is detected:
  See **Section 35.4.3.10, Pattern Phase Difference Detection Flag (TSG3nPPF)**.

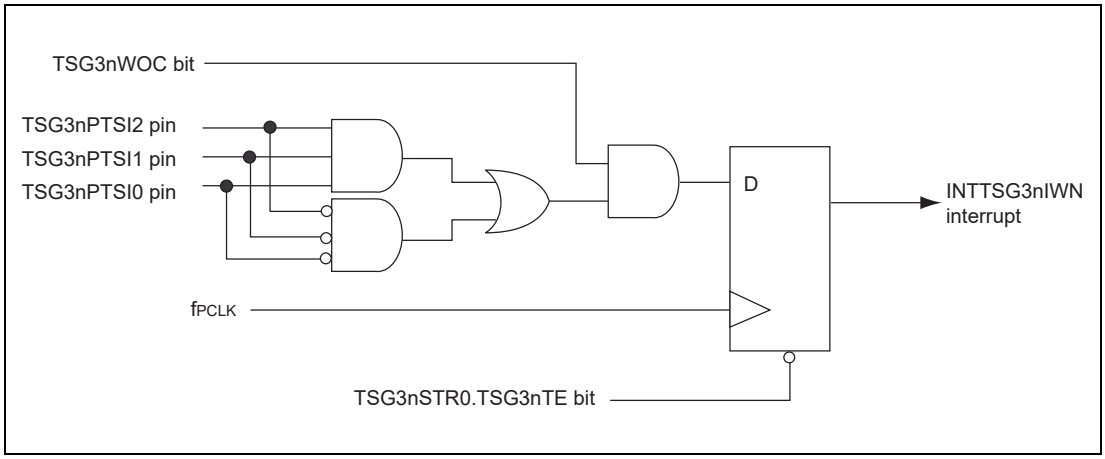


Figure 35.50 Detection of Abnormality of TSG3nPTSI2-TSG3nPTSI0 Pins

35.4.7 Operating Modes

35.4.7.1 PWM Mode

Overview

A PWM signal is output at the TSG3nO1 to TSG3nO6 pins according to set timing/clear timing of TSG3nCMP1E to TSG3nCMP12E registers with the PWM period set in the TSG3nCMP0E register.

Prerequisites

- Set the set timing to the compare register with an even number:
  TSG3nCMP2E (set timing of the TSG3nO1 output), TSG3nCMP4E (set timing of the TSG3nO2 output), TSG3nCMP6E (set timing of the TSG3nO3 output), TSG3nCMP8E (set timing of the TSG3nO4 output), TSG3nCMP10E (set timing of the TSG3nO5 output) and TSG3nCMP12E (set timing of the TSG3nO6 output)
- Set the clear timing to the compare register with an odd number:
  TSG3nCMP1E (clear timing of the TSG3nO1 output), TSG3nCMP3E (clear timing of the TSG3nO2 output), TSG3nCMP5E (clear timing of the TSG3nO3 output), TSG3nCMP7E (clear timing of the TSG3nO4 output), TSG3nCMP9E (clear timing of the TSG3nO5 output) and TSG3nCMP11E (clear timing of the TSG3nO6 output)

Functional description

Set the PWM period and set/clear timing of the TSG3nO1 to TSG3nO6 outputs. Set TSG3nTRG0.TSG3nTS = 1 to start the timer counter.

The TSG3nO1 to TSG3nO6 outputs are set to the inactive state at the same time the counting begins. The outputs are set to the active state by the match of the buffer registers TSG3nCMP2E, TSG3nCMP4E, TSG3nCMP6E, TSG3nCMP8E, TSG3nCMP10E, and TSG3nCMP12E with the 18-bit counter.

Next, the TSG3nO1 to TSG3nO6 outputs are set to the inactive state by the match of the buffer registers TSG3nCMP1E, TSG3nCMP3E, TSG3nCMP5E, TSG3nCMP7E, TSG3nCMP9E, and TSG3nCMP11E with the 18-bit counter.

During counting, a compare match interrupt (INTTSG3nI0 to INTTSG3nI12) is generated by the match of the buffer register TSG3nCMP0E-TSG3nCMP12E with the 18-bit counter.

CAUTION

Reload is executed when a value is written to the TSG3nCMP1E register while TSG3nCTL3.TSG3nRMC = 0. Therefore, even when it is needed to rewrite only the value of the TSG3nCMP0E register, a write operation to the TSG3nCMP1E register is necessary. When only the TSG3nCMP0E register is rewritten, reload is not done.

NOTE

The PWM mode is set when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 000_B.

(a) When TSG3nCMP0E and TSG3nCMP1E to TSG3nCMP12E are Not Rewritten during Timer Operation

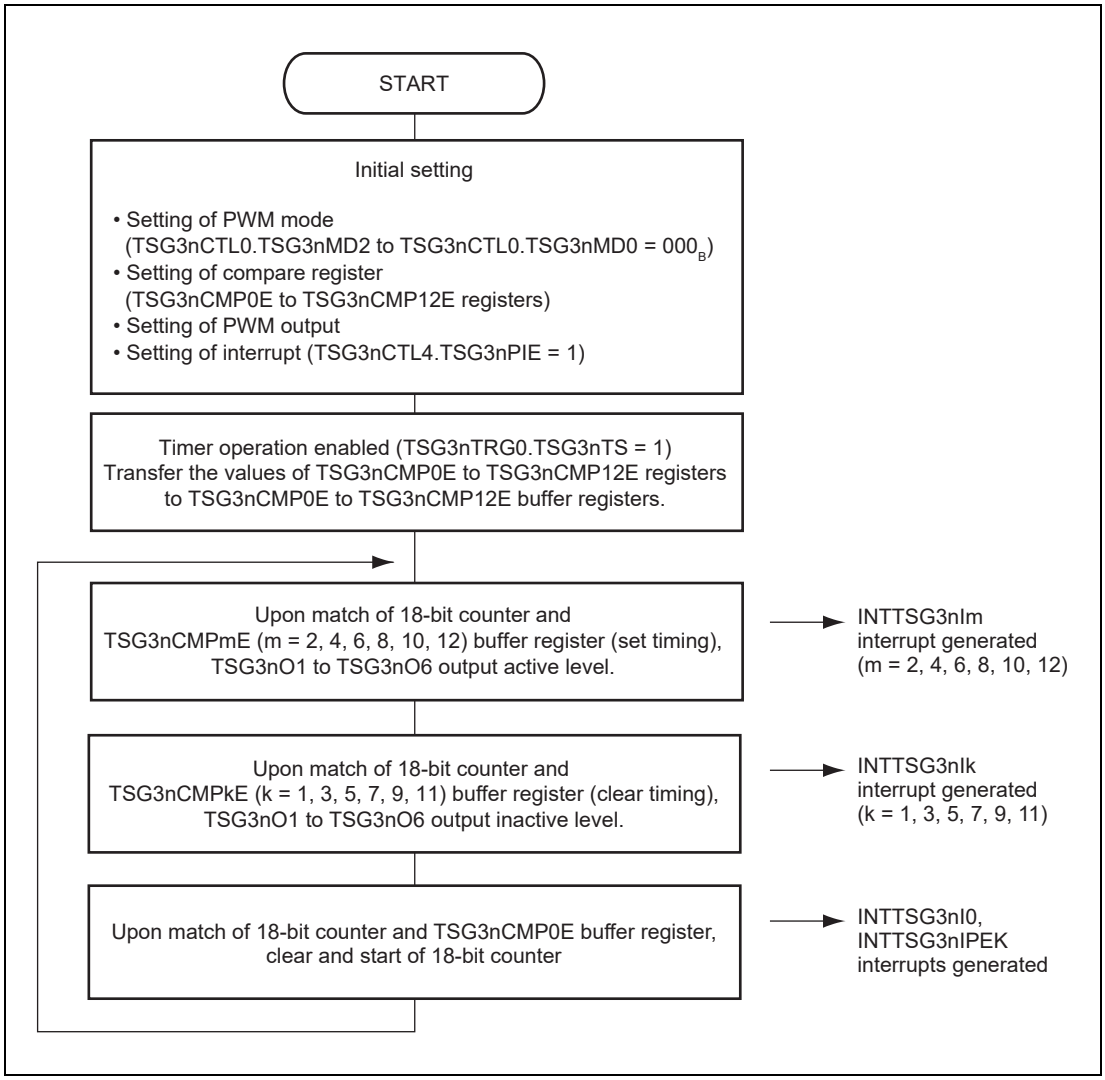
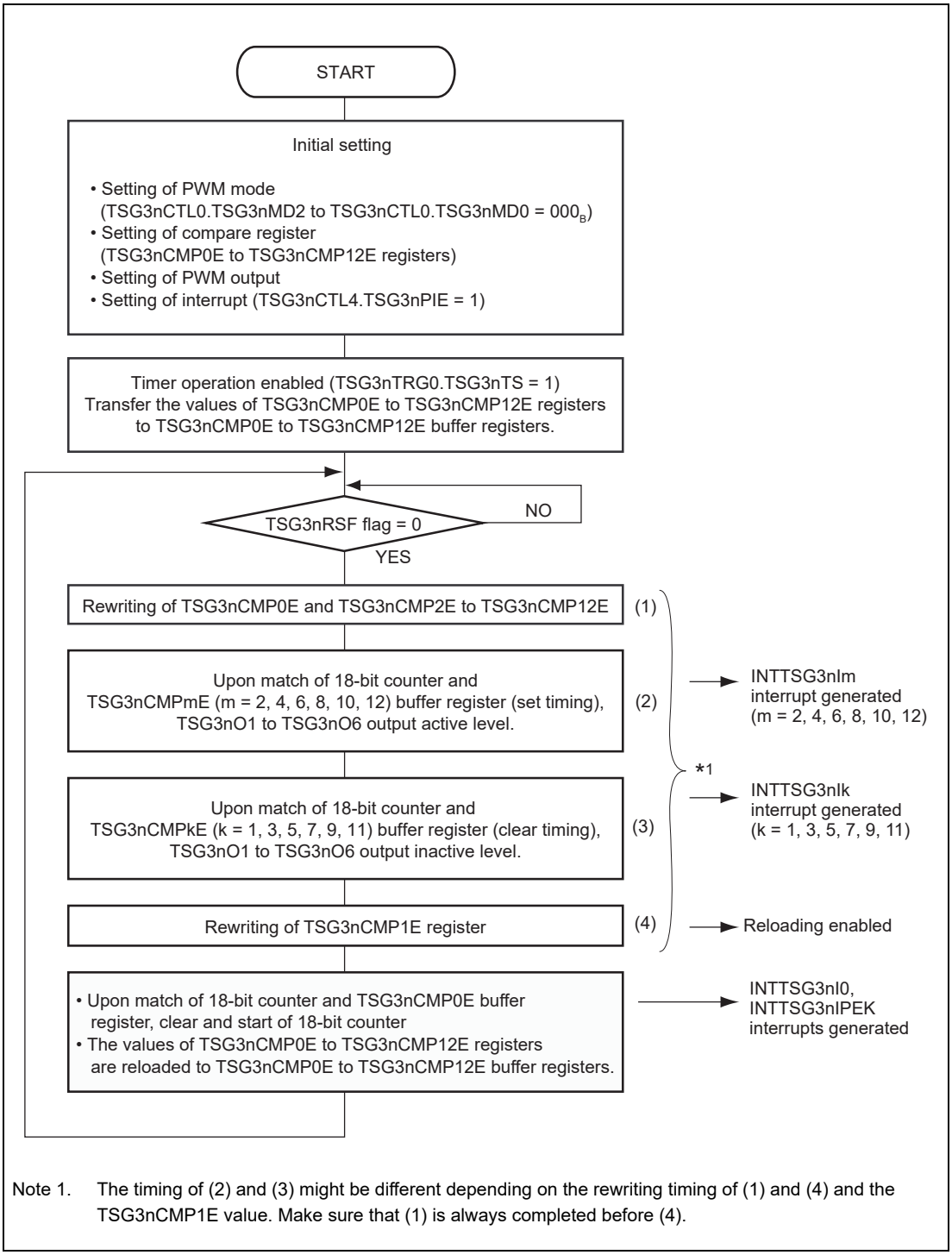


Figure 35.51 Basic Operation Flow of PWM Mode (1/2)

(b) When TSG3nCMP0E and TSG3nCMP1E to TSG3nCMP12E are Rewritten during Timer Operation



Note 1. The timing of (2) and (3) might be different depending on the rewriting timing of (1) and (4) and the TSG3nCMP1E value. Make sure that (1) is always completed before (4).

Figure 35.51 Basic Operation Flow of PWM Mode (2/2)

CAUTION

Rewrite compare registers after confirming that the reload request flag TSG3nRSF is 0.

(1) List of Operations in PWM Mode

Table 35.54 Counter Functions in PWM Mode

Operation		Setting condition
18-bit counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1, or a simultaneous start trigger
	Clear	Compare match of TSG3nCMP0E buffer register and 18-bit counter
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1

Table 35.55 Functions of Compare Registers and Dead Time Control Register in PWM Mode

Register	Rewriting Method	Rewrite during Operation	Function
TSG3nCMP0E	Reload/Anytime rewrite	Possible	Setting period
TSG3nCMPmE (m = 1 to 12)	Reload/Anytime rewrite	Possible	Setting set/clear timing
TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E	Reload/Anytime rewrite	Possible	Diagnostic output or A/D conversion trigger
TSG3nDTC0W, TSG3nDTC1W	Reload	Possible*1	Setting dead time

Note 1. For details, see **Section 35.4.7.1, (3) Controlling Dead Time in PWM Mode**.

Table 35.56 Timer Input Function in PWM Mode

Pin	Function
TSG3nCLKI	Clock enable input

Table 35.57 Timer Output Function in PWM Mode

Pin	Function
TSG3nOm (m = 1 to 6)	PWM output by compare match of TSG3nCMPkE buffer register and 18-bit counter (k = 1 to 12)
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Table 35.58 Interrupt Requests in PWM Mode

Interrupt	Function
INTTSG3nIm (m = 0 to 12)	Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 0 to 12)
INTTSG3nIER	Error (detection of simultaneous active state of TSG3nO1 and TSG3nO2, or TSG3nO3 and TSG3nO4, or TSG3nO5 and TSG3nO6)
INTTSG3nIVLY	—
INTTSG3nIPEK	Peak interrupt (generated at the same timing as INTTSG3nI0)
INTTSG3nIWN	Warning interrupt

Note: "—": Not available in PWM mode

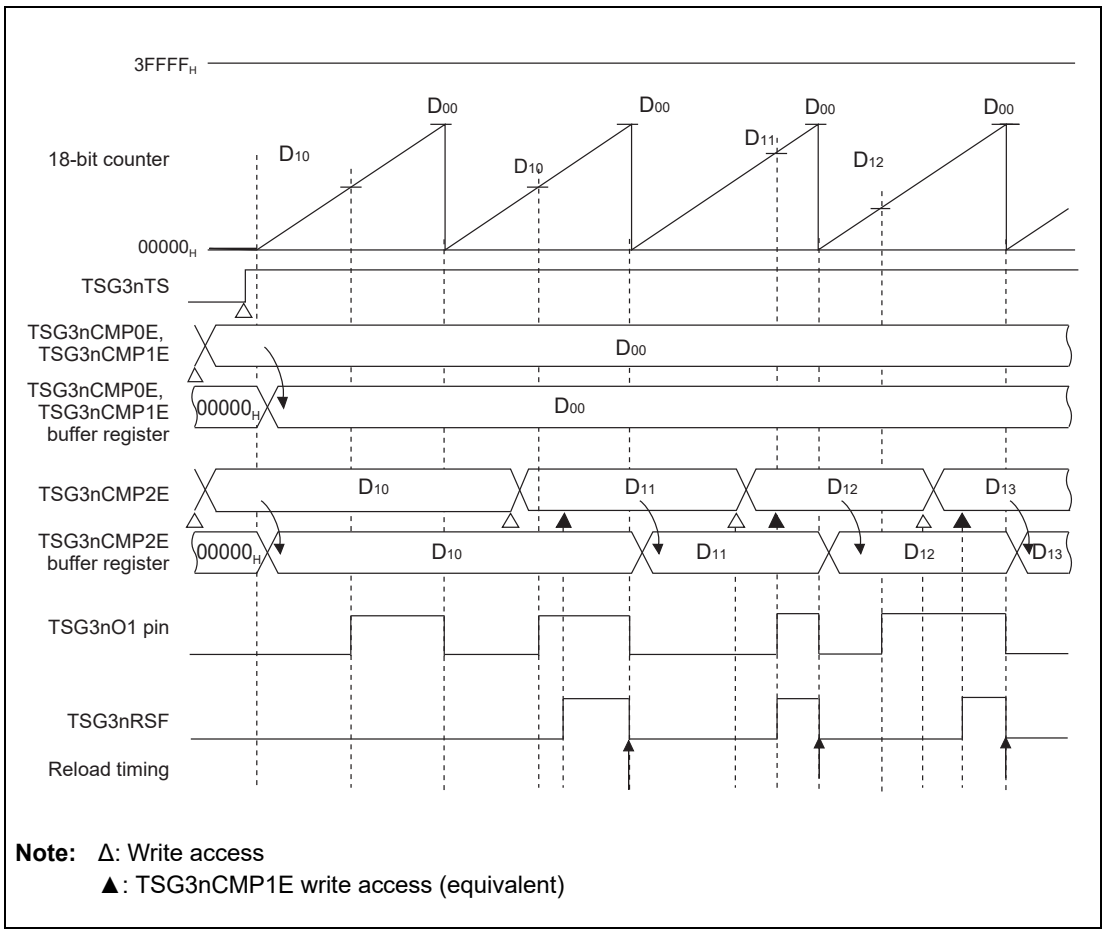
Table 35.59 Compare Match Timing in PWM Mode

Compare Match	Timing
TSG3nCMP0E	When 18-bit counter changes from TSG3nCMP0E to 00000_H
TSG3nCMPmE (m = 1 to 12)	After detecting the match of 18-bit counter and TSG3nCMPmE (m = 1 to 12)

Table 35.60 Example of Setting each Timer Output Condition in PWM Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG3nOm (m = 1 to 6)	PWM output	(TSG3nCMP0E + 1) × count clock	Output an inactive level throughout one period (duty cycle 0%)	TSG3nCMPmE = TSG3nCMP (m + 1)E or TSG3nCMP (m + 1) E> TSG3nCMP0E (m = 1, 3, 5, 7, 9, 11)
			Output an active level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m + 1)E + 1 TSG3nCMP (m + 1)E = TSG3nCMPmE - 1 (m = 1, 3, 5, 7, 9, 11)
			Output an inactive level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m + 1)E - 1 TSG3nCMP (m + 1)E = TSG3nCMPmE + 1 (m = 1, 3, 5, 7, 9, 11)
			Output an active level throughout one period (duty cycle 100%)	TSG3nCMPmE > TSG3nCMP0E TSG3nCMP (m + 1)E ≤ TSG3nCMP0E (m = 1, 3, 5, 7, 9, 11)

When only TSG3nCMP2E is rewritten and TSG3nO1 is output (TSG3nIOC0.TSG3nTOE1 = 1, TSG3nIOC2.TSG3nOL1 = 0)



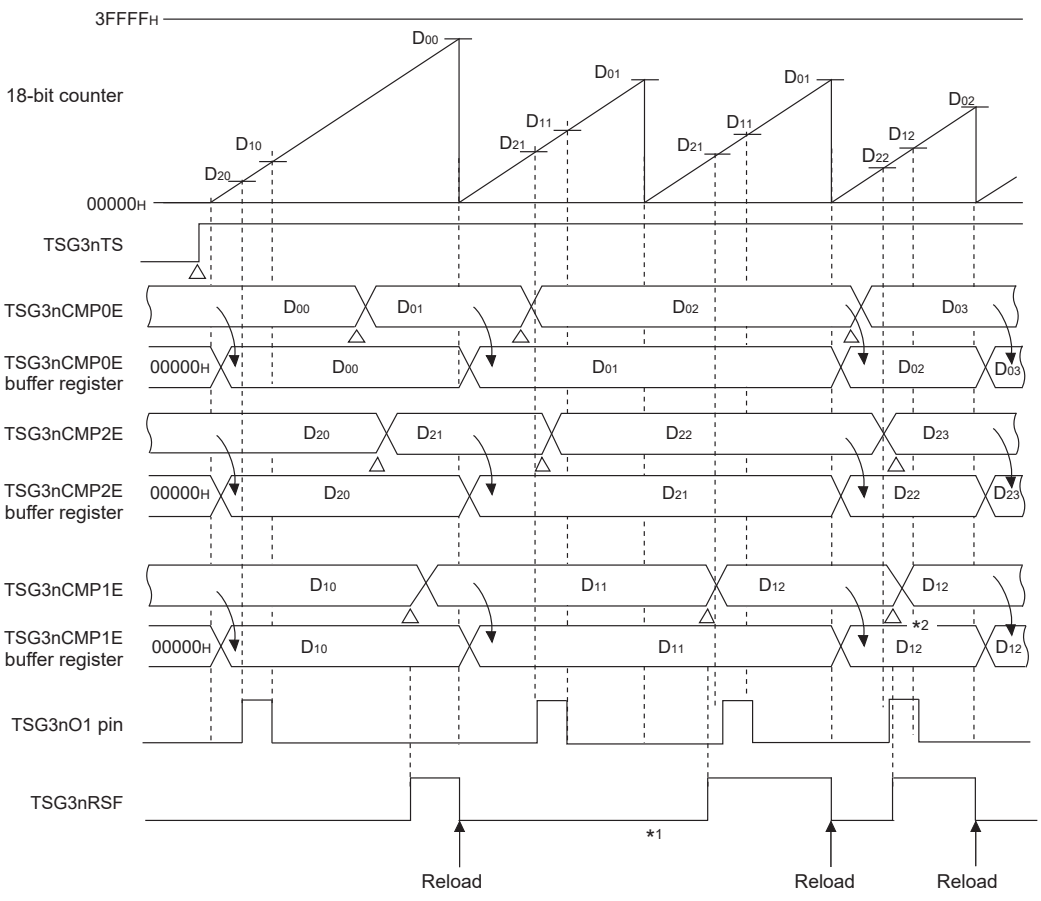
Note: Δ: Write access
▲: TSG3nCMP1E write access (equivalent)

Figure 35.52 Example of Basic Operation Timing of PWM Mode (1/2)

NOTES

1. D00: Set values of TSG3nCMP0E and TSG3nCMP1E (00000_H-$3FFFF_H$)
   D10, D11, D12 and D13: Set values of TSG3nCMP2E (00000_H-$3FFFF_H$)
2. TSG3nO1 (PWM) duty cycle = (TSG3nCMP1E-TSG3nCMP2E)×(count clock cycle)
   TSG3nO1 (PWM) period = (Set value of TSG3nCMP0E + 1)×(count clock cycle)
3. TSG3nO2-TSG3nO6 pins behave similarly to the TSG3nO1 pin

When TSG3nCMP0E-TSG3nCMP2E are rewritten, and TSG3nO1 is output (TSG3nIOC0.TSG3nTOE1 = 1, TSG3nIOC2.TSG3nOL1 = 0)



Note 1. The TSG3nCMP0E buffer register is not reloaded, because TSG3nCMP1E register was not rewritten.

Note 2. Equivalent writing

Note: Δ: Write access

Figure 35.52 Example of Basic Operation Timing of PWM Mode (2/2)

NOTES

1. D00, D01, D02, D03: Set value of TSG3nCMP0E (00000_H-$3FFFF_H$)
   D10, D11, D12, D13: Set value of TSG3nCMP1E (00000_H-$3FFFF_H$)
   D20, D21, D22, D23: Set value of TSG3nCMP2E (00000_H-$3FFFF_H$)
2. Outputs from TSG3nO2 to TSG3nO6 behave similarly to the TSG3nO1 pin.

(2) Interrupt/Reload Skipping Function in PWM Mode

By setting TSG3nCTL4.TSG3nPRE and TSG3nPIE to 1 and setting the TSG3nRCC04 to TSG3nRCC00 and TSG3nCTL3.TSG3nRIA, the reload and interrupt skipping function can be used.

By setting TSG3nPRE to 1 and setting the TSG3nRCC04 to TSG3nRCC00, the interrupt skipping function can be used.

(3) Controlling Dead Time in PWM Mode

By setting the dead time values in the TSG3nDTC0W and TSG3nDTC1W registers in PWM mode, it is possible to control the dead time. The dead time is controlled according to the switch timing of the TSG3nO1 and TSG3nO2 outputs, the TSG3nO3 and TSG3nO4 outputs, or the TSG3nO5 and TSG3nO6 outputs.

Table 35.61 Dead Time in PWM Mode

Switch Timing	Dead Time
TSG3nO1: High level to low level and TSG3nO2: Low level to high level	Value of TSG3nDTC1W register
TSG3nO2: High level to low level and TSG3nO1: Low level to high level	Value of TSG3nDTC0W register
TSG3nO3: High level to low level and TSG3nO4: Low level to high level	Value of TSG3nDTC1W register
TSG3nO4: High level to low level and TSG3nO3: Low level to high level	Value of TSG3nDTC0W register
TSG3nO5: High level to low level and TSG3nO6: Low level to high level	Value of TSG3nDTC1W register
TSG3nO6: High level to low level and TSG3nO5: Low level to high level	Value of TSG3nDTC0W register

NOTE

Dead time counter keeps operating even when operation stop (TSG3nTE = 0) setting collided with dead time insert timing, and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.

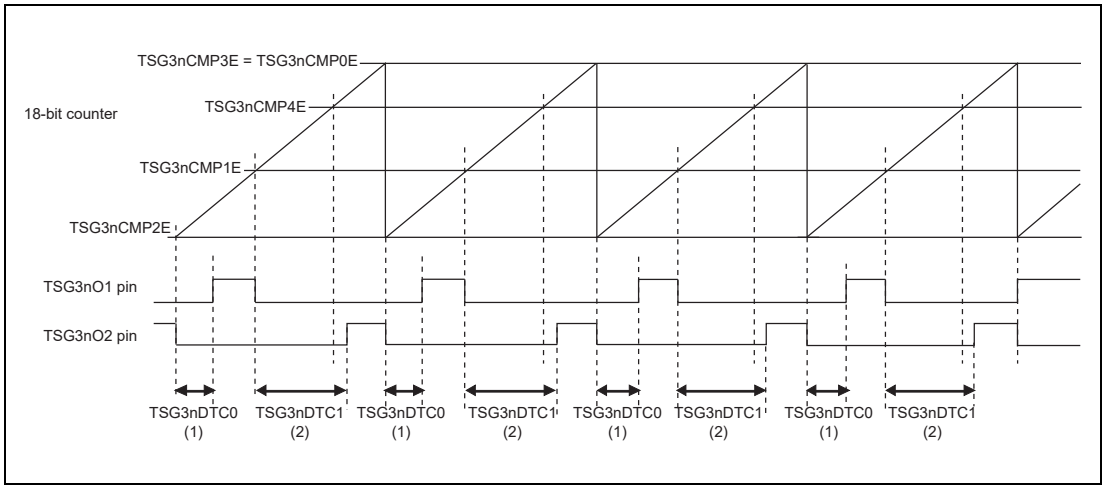


Figure 35.53 Example of Dead Time Control between TSG3nO1 and TSG3nO2 Outputs (1/2)

During (1), the dead time counter starts counting at the falling edge of the TSG3nO2 output. At this time, even after the 18-bit counter reaches 00000_H indicating that the TSG3nO1 output should be active, the TSG3nO1 output stays inactive because the dead time counter is still operating. The TSG3nO1 output becomes active at the timing when the dead time count operation ends.

At (2), the dead time counter starts counting at the falling edge of the TSG3nO1 output. Even after the match of the 18-bit counter and TSG3nCMP4E indicating that the TSG3nO2 output should be active, the TSG3nO2 output stays inactive because the dead time counter is still operating. The TSG3nO2 output becomes active at the timing when the dead time count operation ends.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

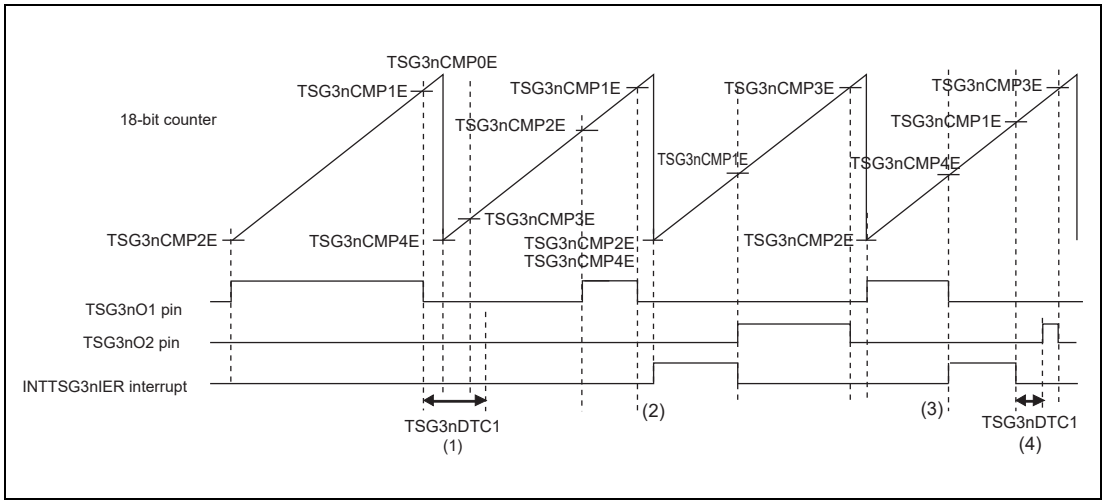


Figure 35.53 Example of Dead Time Control between TSG3nO1 and TSG3nO2 Outputs (2/2)

During (1), the dead time counter starts counting at the falling edge of the TSG3nO1 output. Even after the 18-bit counter reaches 00000_H and the match occurs between the 18-bit counter and TSG3nCMP4E indicating that the TSG3nO2 output should be active, the TSG3nO2 output stays inactive because the dead time counter is still operating. Moreover, since the TSG3nCMP3E register compare match occurs before the operation of the dead time counter ends, the TSG3nO2 output stays inactive.

TSG3nCMP1E + TSG3nDTC1 ≥ TSG3nCMP0E + TSG3nCMP3E
(TSG3nO2 stays inactive)

TSG3nCMP3E + TSG3nDTC0 ≥TSG3nCMP0E + TSG3nCMP1E
(TSG3nO1 stays inactive)

At (2), the INTTSG3nIER interrupt occurs because the TSG3nCMP2E register and the TSG3nCMP4E register are set so that the TSG3nO1 and TSG3nO2 outputs rise simultaneously. Here, both the TSG3nO1 and TSG3nO2 outputs become inactive.

At (3), compare match with the TSG3nCMP4E register while the TSG3nO1 output is active generates an INTTSG3nIER interrupt and both TSG3nO1 and TSG3nO2 outputs become inactive.

At (4), the falling edge (inactive) of the TSG3nO1 output is caused by the detection of simultaneous active state. The dead time counter starts counting after compare match with the TSG3nCMP1E register. After the end of the dead time counter operation, the TSG3nO2 output becomes active.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high.
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

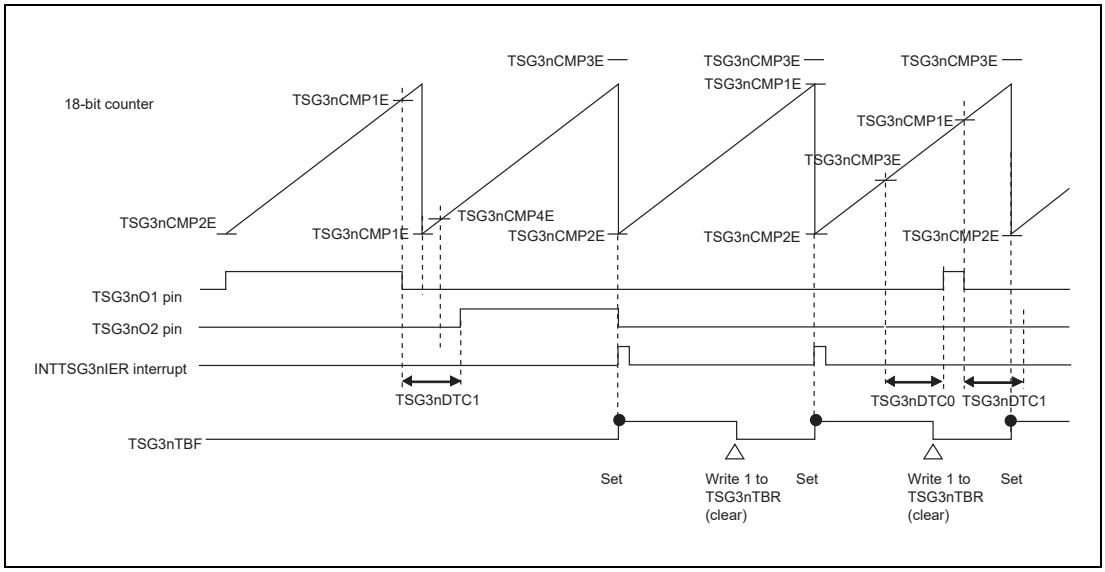


Figure 35.54 Example of 100% Duty Output at Dead Time Control

When the TSG3nO2 pin is set to duty cycle of 100% (TSG3nCMP3E ≥ TSG3nCMP0E + 1), the output of the TSG3nO1 pin is fixed to a low level. This control is intended to mask the active condition of TSG3nO1 output since the TSG3nO2 output is active before the TSG3nO1 output becomes active. In this case, the INTTSG3nIER interrupt is also generated because TSG3nO1 and TSG3nO2 outputs become high simultaneously.

NOTES

1. The TSG3nO1 and TSG3nO2 pins are set to active high.
2. The TSG3nO3 and TSG3nO4 pins and the TSG3nO5 and TSG3nO6 pins outputs behave similarly.

(4) Dead Time Rewriting during Timer Operation in PWM Mode

In PWM mode, it is possible to rewrite TSG3n dead time control registers TSG3nDTC0W and TSG3nDTC1W while counting. The new settings are active at reload timing. It is not possible to change the dead time setting by rewriting at any time.

To enable reloading, write to the TSG3nCMP1E register.

35.4.7.2 HT-PWM mode (High accuracy Triangular - Pulse Width Modulation mode)

Overview

In this mode, the 18-bit counter (up/down count by ± 2 bits, practically 17 bits) and the 18-bit compare registers (LSB is used to control additional pulse) are used to generate a 6-phase PWM signal.

Prerequisites

- Set the carrier wave period with TSG3nCMP0E.
- Set the duty cycle of the voltage data signals of the U phase, V phase, and W phase with TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE (The values set in TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE are reflected immediately to the corresponding TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10)).
- Symmetric triangular wave control is described in this section. Refer to **Section 35.4.7.2 (10), Asymmetric Triangular Wave Control in HT-PWM Mode**, for asymmetric triangular wave control.

Functional description

In this mode, the carrier period and the duty cycle of the U phase, the V phase, and the W phase are configured. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter counts up from TSG3nDTC0 as the minimum value, and counts down upon the match with the maximum value of TSG3nCMP0E + TSG3nDTC0.

The dead time is set with TSG3nDTC0 and TSG3nDTC1. TSG3nDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch and TSG3nDTC1 sets the dead time of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSG3nDTT1 to TSG3nDTT3) for dead time generation load the set values of TSG3nDTC0 and TSG3nDTC1 by the compare match of the 18-bit counter and the TSG3nCMPm buffer register (m = 1, 2, 5, 6, 9, 10), and start down-counting.

INTTSG3nIm interrupts (m = 1, 2, 5, 6, 9, 10) are generated by the compare match of the 18-bit counter with TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E buffer registers.

INTTSG3nIm interrupts (m = 3, 4, 7, 8, 11, 12) are generated by the compare match of the 18-bit counter with TSG3nCMP3E, TSG3nCMP7E, and TSG3nCMP11E buffer registers when counting down (TSG3nCUF = 1), and with TSG3nCMP4E, TSG3nCMP8E, and TSG3nCMP12E buffer registers when counting up (TSG3nCUF = 0).

NOTE

The HT-PWM mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 001_B.

(1) Block Diagram and Basic Timing Chart

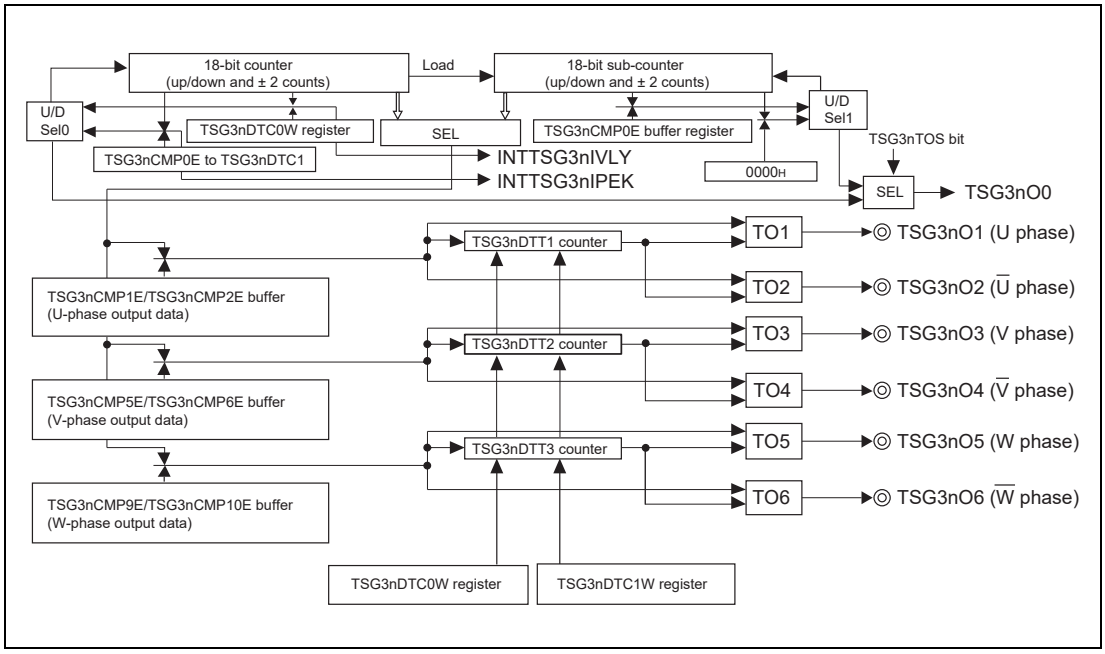


Figure 35.55 Block Diagram in HT-PWM Mode

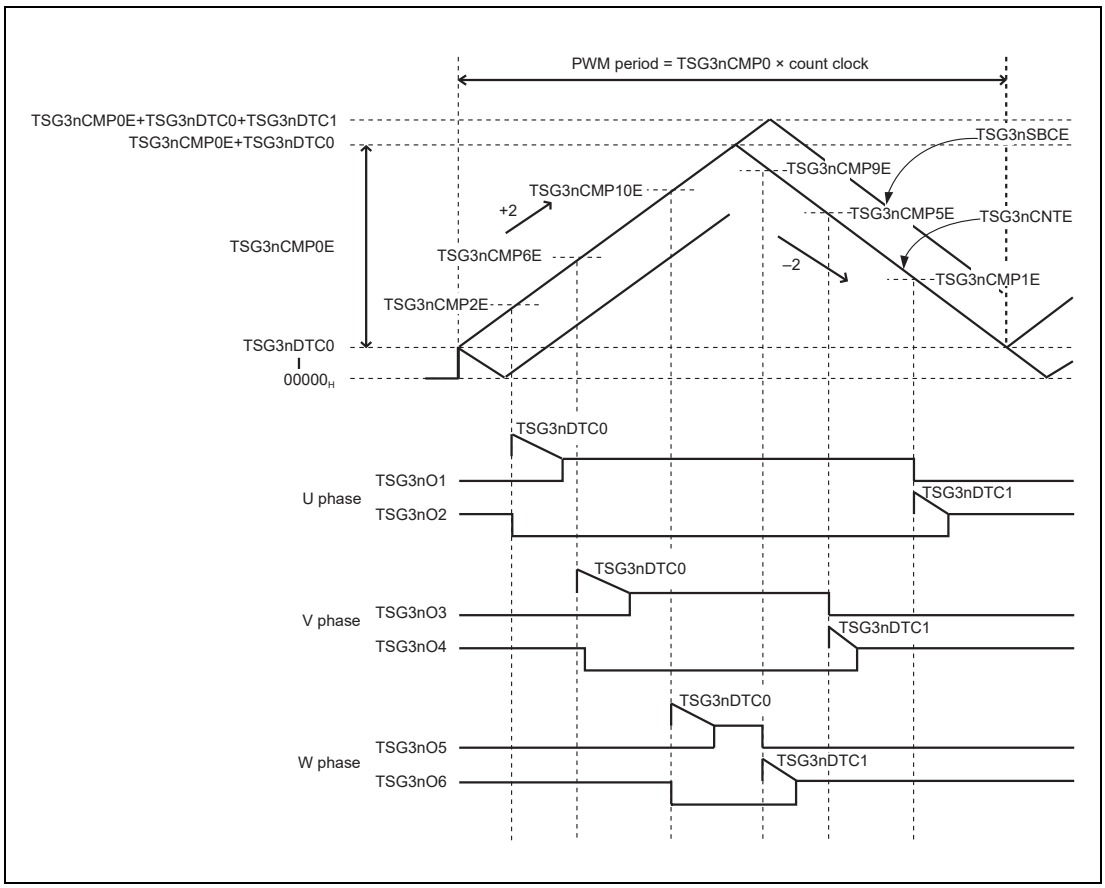


Figure 35.56 Basic Timing in HT-PWM Mode

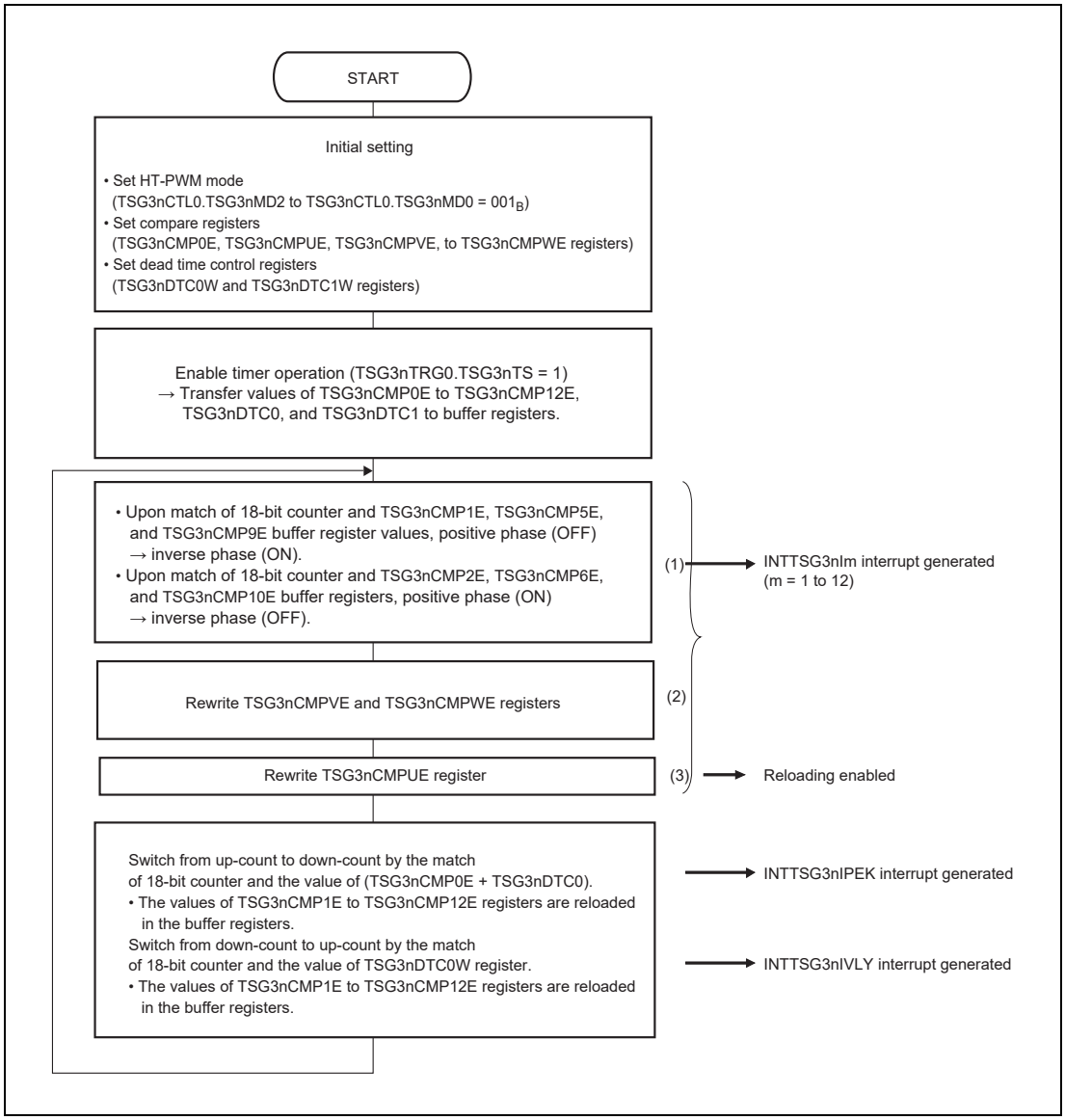


Figure 35.57 Basic Operation Flow in HT-PWM Mode

NOTE

- Write access to TSG3nCMPUE (TSG3nCMP1E) includes reloading enabling operation. Therefore, (3) must be done after (2).
- The INTTSG3nIPEK interrupt is generated only when TSG3nCTL4.TSG3nPIE = 1.
- The INTTSG3nIVLY interrupt is generated only when TSG3nCTL4.TSG3nVIE = 1.
- INTTSG3nI3, INTTSG3nI7, and INTTSG3nI11 outputs an interrupt at the match timing of TSG3nCMP3E, 7E, and 11E with TSG3nCNTE when counting down (TSG3nCUF = 1). INTTSG3nI4, INTTSG3nI8, and INTTSG3nI12 outputs an interrupt at the match timing of TSG3nCMP4E, 8E, and 12E with TSG3nCNTE when counting up (TSG3nCUF = 0).

(2) List of HT-PWM Mode Operations

Table 35.62 Counter Function in HT-PWM Mode

Operation		Setting Condition
18-bit counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger (counting up from TSG3nDTC0)
	Up count	Compare match of TSG3nDTC0 buffer register and 18-bit counter
	Down count	Compare match of TSG3nCMP0E + TSG3nDTC0 and 18-bit counter
	Clear	—
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1
18-bit sub-counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger (counting down from TSG3nDTC0)
	Up count	Underflow
	Down count	Compare match of TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 buffer register and 18-bit sub-counter
	Load	• TSG3nCMP0E + TSG3nDTC0: When value of 18-bit counter matches the value of buffer register TSG3nCMP0E + TSG3nDTC0 • TSG3nDTC0: When value of 18-bit counter matches the value of the buffer register TSG3nDTC0
	Clear	—
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1

Table 35.63 Compare Register and Dead Time Control Register Functions in HT-PWM Mode

Register	Rewrite Method	Rewrite during Operation	Function
TSG3nCMP0E	Reload/Anytime rewrite	Possible	Setting period
TSG3nCMPUE	—	Possible	PWM control for U phase
TSG3nCMP1E, TSG3nCMP2E	Reload/Anytime rewrite		
TSG3nCMPVE	—	Possible	PWM control for V phase
TSG3nCMP5E, TSG3nCMP6E	Reload/Anytime rewrite		
TSG3nCMPWE	—	Possible	PWM control for W phase
TSG3nCMP9E, TSG3nCMP10E	Reload/Anytime rewrite		
TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E	Reload/Anytime rewrite	Possible	Diagnostic signal output or A/D conversion trigger
TSG3nDTC0W, TSG3nDTC1W	Reload	Possible conditionally	Period and dead time control

NOTE

- The rewritten values of TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE are set to TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E and TSG3nCMP10E.
- For rewriting method of TSG3nDTC0 and TSG3nDTC1, see **Section 35.4.7.2(8)(a), TSG3nDTC0 and TSG3nDTC1 Rewriting**.

Table 35.64 Timer Output Function in HT-PWM Mode

Pin	Function
TSG3nO0	Active level output during up count, inactive level output at down count of the 18-bit counter/sub-counter
TSG3nO1	PWM output with dead time by compare match of TSG3nCMP1E buffer register and 18-bit counter (down count) and compare match of TSG3nCMP2E buffer register and 18-bit counter (up count) PWM output by compare match of TSG3nCMP1E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP2E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP1E < DTC0
TSG3nO2	Output negative phase with respect to TSG3nO1 (with dead time)
TSG3nO3	PWM output with dead time by compare match of TSG3nCMP5E buffer register and 18-bit counter (down count) and compare match of TSG3nCMP6E buffer register and 18-bit counter (up count). PWM output by compare match of TSG3nCMP3E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP6E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP5E < DTC0
TSG3nO4	Output negative phase with respect to TSG3nO3 (with dead time)
TSG3nO5	PWM output with dead time by compare match of TSG3nCMP9E buffer register and 18-bit counter (down count) and TSG3nCMP10E buffer register and 18-bit counter (up count). PWM output by compare match of TSG3nCMP5E buffer register and 18-bit sub-counter (during down counting) and compare match of TSG3nCMP10E buffer register and 18-bit sub-counter (during up counting) while TSG3nCMP9E < DTC0
TSG3nO6	Output negative phase with respect to TSG3nO5 (with dead time)
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger

NOTES

1. The target of TSG3nO0 status output can be switched with TSG3nIOC1.TSG3nTOS.
2. When the peak and valley values of the 18-bit sub-counter are set in TSG3nCMP1E and TSG3nCMP2E, clearing takes precedence.

Table 35.65 Interrupt Request in HT-PWM Mode

Interrupt	Function
INTTSG3nI0	Compare match of TSG3nDTC0 buffer register and 18-bit counter (periodic interrupt)
INTTSG3nIm (m = 1, 2, 5, 6, 9, 10)	Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 1, 2, 5, 6, 9, 10)
INTTSG3nIER	Error interrupt
INTTSG3nIVLY	Valley interrupt
INTTSG3nIPEK	Peak interrupt
INTTSG3nIWN	Warning interrupt

Table 35.66 Compare Match Timing in HT-PWM Mode

Compare Match	Timing
TSG3nCMP0E	When the 18-bit counter changes from TSG3nDTC0 to TSG3nDTC0 + 2
TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10)	When 18-bit counter changes from TSG3nCMPmE to TSG3nCMPmE ± 2 (m = 1, 2, 5, 6, 9, 10)

Table 35.67 Example of Setting Each Timer Output Condition in HT-PWM Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG3nO0	Toggle output	TSG3nCMP0E × count clock	Output an inactive level when counting up, and an active level when counting down.	—
TSG3nO1, TSG3nO3, TSG3nO5	PWM output	TSG3nCMP0E × count clock	Output an inactive level throughout one period (0% duty)	TSG3nCMP0E ≤ TSG3nCMPmE ≤ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 (m = U, V, W)
			Output an active level of one count clock in one period	TSG3nCMPmE = TSG3nCMP0E – 1 (m = U, V, W)
			Output an inactive level of one count clock in one period	TSG3nCMPmE = 0001_H (m = U, V, W)
			Output an active level throughout one period(100% duty)	TSG3nCMPmE = 0000_H (m = U, V, W)
TSG3nO2, TSG3nO4, TSG3nO6	PWM output		Output an inactive level throughout one period (0% duty)	TSG3nCMPmE ≤ TSG3nDTC0 + TSG3nDTC1 (m = U, V, W)
			Output an active level of one count clock in one period	TSG3nCMPm = TSG3nDTC0 + TSG3nDTC1 + 1 (m = U, V, W)
			Output an inactive level of one count clock in one period	TSG3nCMPmE = TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 – 1 (m = U, V, W)
			Output an active level throughout one period (100% duty)	TSG3nCMPmE = TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 (m = U, V, W)
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger	TSG3nCMP0E × count clock	See **Section 35.4.5, A/D Conversion Trigger Function**	

(3) Various Settings of HT-PWM Mode

Setting mode

HT-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 001_B.

Setting timer output

The output pins TSG3nO1 to TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The output pin TSG3nO0 indicates the up/down count status of the 18-bit counter or the 18-bit sub-counter. Switch between the 18-bit sub-counter and the 18-bit counter is done with the TSG3nIOC1.TSG3nTOS bit.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of the positive phase and inverse phase is enabled by setting TSG3nIOC1.TSG3nEOC to 1. In HT-PWM mode, no value specified in the compare register makes the positive phase and negative phase simultaneously active. For details, see **Section 35.4.6, Error/Warning Interrupt**.

Setting register rewriting timing with reload function

With TSG3nCTL3.TSG3nRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSG3nCTL4.TSG3nPRE or TSG3nVRE to 1.

The reload timing is not generated if both TSG3nPRE bit and TSG3nVRE bit are set to 0.

When “anytime rewrite” is specified, unintended output may be generated depending on the rewrite timing.

Setting interrupts and skipping function

Interrupts and the skipping function are set with TSG3nCTL4. TSG3nPIE should be set to 1 when peak interrupt (INTTSG3nIPEK) is required and TSG3nVIE to 1 when valley interrupt (INTTSG3nIVLY) is required. To use the skipping function for peak/valley interrupts, set TSG3nRCC4 to TSG3nRCC0.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSG3nADTRG0 signal), use TSG3nCTL5.TSG3nAT09 to TSG3nAT00. With TSG3nAT09 to TSG3nAT00, A/D conversion trigger output is enabled or disabled at the match of 18-bit counter (during up count) with TSG3nDCMP2E to TSG3nDCMP0E, the match of the 18-bit counter (during down count) with TSG3nDCMP2E to TSG3nDCMP0E, the 18-bit counter peak interrupt (INTTSG3nIPEK), the 18-bit counter valley interrupt (INTTSG3nIVLY), the 18-bit sub-counter peak timing, and 18-bit sub-counter valley timing.

To set A/D conversion trigger 1 (TSG3nADTRG1 signal), use TSG3nCTL6.TSG3nAT19 to TSG3nAT10.

To set the match timing of 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to the pertinent register.

The skipping function can be used for TSG3nADTRG0 and the TSG3nADTRG1 signals. Use TSG3nACC01, TSG3nACC00 of TSG3nCTL5, and TSG3nACC10, TSG3nACC11 of TSG3nCTL6 to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of the A/D conversion trigger is output to TSG3nO7.

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

PCLK × TSG3nDTC0

PCLK × TSG3nDTC1

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, TSG3nO5 to the inactive state and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

TSG3nDTC0 and TSG3nDTC1 can only be set to an even value.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

TSG3nCMP0E = Carrier period/count clock period (PCLK)

The following requirements regarding the dead time must be satisfied when setting the TSG3nCMP0E register:

- TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 ≤ 3FFFEH
- TSG3nCMP0E > TSG3nDTC0
- TSG3nCMP0E > TSG3nDTC1
- TSG3nCMP0E > 3 × MAX (TSG3nDTC0, TSG3nDTC1)
- TSG3nCMP0E: Even number

NOTE

MAX (A, B) indicates the larger value of A and B.

Setting duty (PWM width)

The duty of the U phase, the V phase, and the W phase is set with TSG3nCMPmE (m = U, V, W, or 1, 2, 5, 6, 9, and 10), respectively. The setting range of the compare registers is as follows:

$00000_H \leq$ TSG3nCMPmE $\leq$ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1

LSB (least significant bit) of TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE indicates the setting of an additional pulse. For example, when TSG3nCMPUE = 00003_H, the change in the inverse phase (TSG3nO2 output) occurs one count clock later compared to the TSG3nCMPUE = 00002_H setting (when the 18-bit counter is up-counting). The additional pulse cannot be set to TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, or TSG3nCMP10E (only even numbers can be set to these registers).

(4) 18-bit counter Operation in HT-PWM Mode

The 18-bit counter is initialized to 00000_H and the value of TSG3nDTC0 is loaded immediately after the TSG3n timer operation starts (TSG3nTRG0.TSG3nTS = 1), and the counter is incremented by 2. After 18-bit counter reaches the value of TSG3nCMP0E + TSG3nDTC0, it is decremented by 2.

The following figure shows 18-bit counter operation.

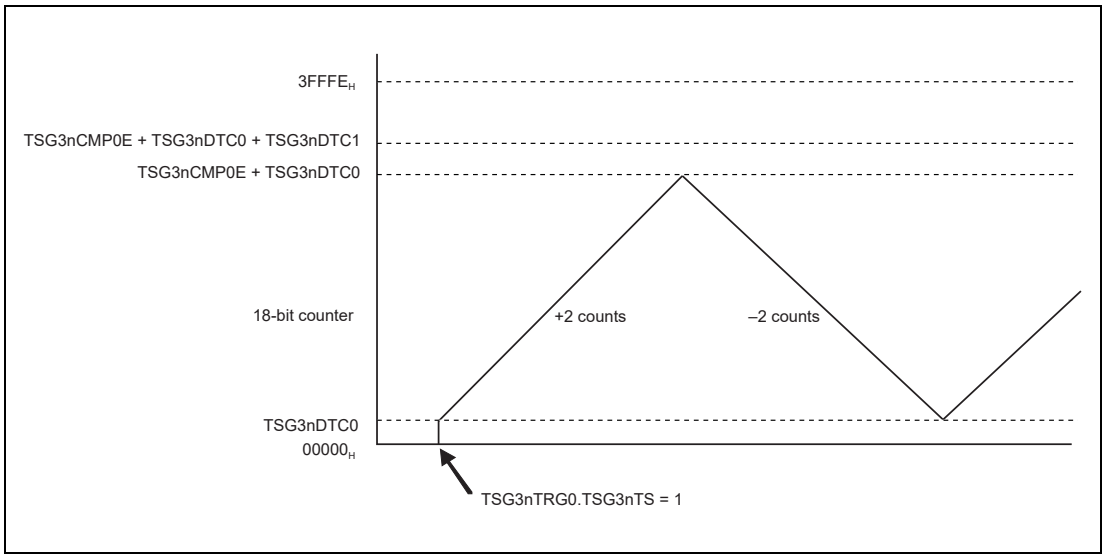


Figure 35.58 Example of 18-bit counter Operation in HT-PWM Mode

NOTE

Minimum 18-bit counter value: TSG3nDTC0
Maximum 18-bit counter value: TSG3nCMP0E + TSG3nDTC0
Carrier period: TSG3nCMP0E × count clock period (PCLK)

The 18-bit sub-counter is initialized to 00000_H and the value of TSG3nDTC0 is loaded immediately after the TSG3n timer operation starts (TSG3nTRG0.TSG3nTS = 1), and the counter is decremented by 2 until it reaches 00000_H, at which point increment by 2 begins. Next, the value of the 18-bit counter is loaded into the 18-bit sub-counter at a change timing of the 18-bit counter into the down count. Counting up by the 18-bit sub-counter continues until the value reaches the value of TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1, and then decrement by 2 begins. Similarly, when the 18-bit counter value matches the TSG3nDTC0 value, the 18-bit counter value is loaded to the 18-bit sub-counter and the down count is continued.

The following figure shows the 18-bit sub-counter operation.

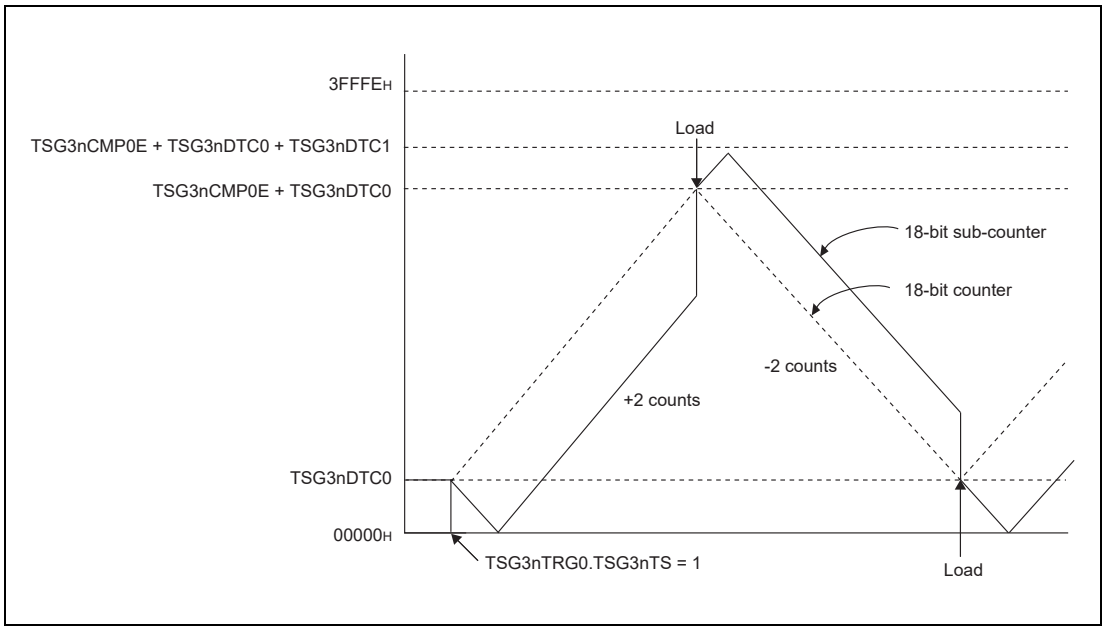


Figure 35.59 Example of 18-bit Sub-Counter Operation in HT-PWM Mode

NOTE

Minimum 18-bit sub-counter value: 00000_H
Maximum 18-bit sub-counter value: TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1

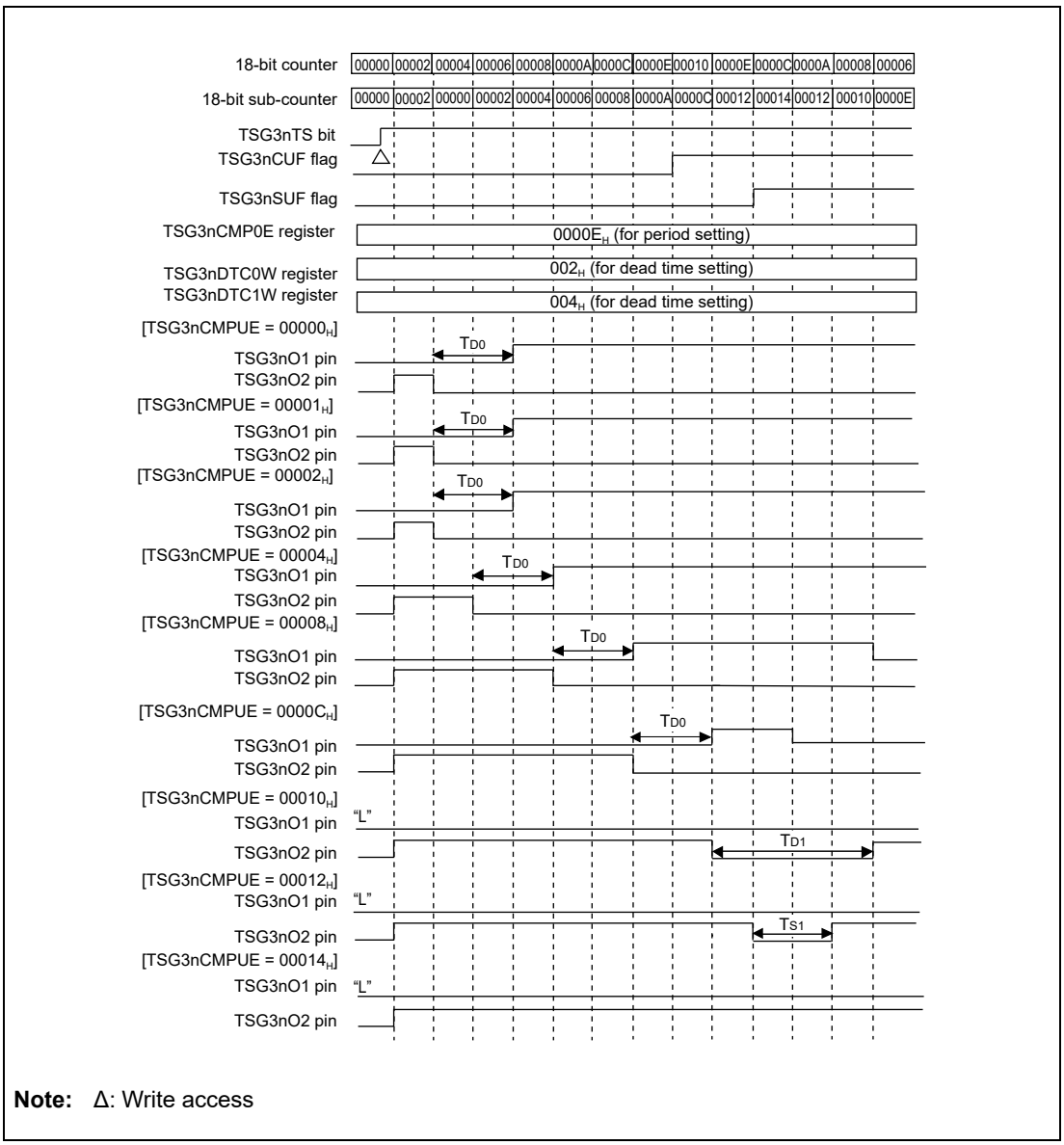
(5) Basic Operation of HT-PWM Mode

(a) Example of Timer Output Immediately after the Start of the TSG3n Timer Operation

The following figure shows the timing chart when TSG3nCMP0E = $0000E_H$, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H and TSG3nCMPUE = 00000_H to 00014_H (excerpt). In this example, TSG3nIOC2.TSG3nOL1 to TSG3nOL6 = 000000_B.

When operation starts (TSG3nTRG0.TSG3nTS = 1), the level of the TSG3nO2 pin changes to active. Afterwards, if TSG3nCMPUE ≤ TSG3nDTC0, the TSG3nO2 pin is cleared after 1 count clock cycle.

The TSG3nO2 pin is cleared upon a match of the 18-bit counter and the compare register (TSG3nCMP2E), or a match of the 18-bit sub-counter and the compare register (TSG3nCMP2E) if TSG3nCMPUE ≥ TSG3nDTC0. Afterwards, the TSG3nO1 pin is set after the set dead time period (the TSG3nO1 pin is not set if TSG3nCMPUE ≥ TSG3nCMP0E).



Note: Δ: Write access

Figure 35.60 Example of Timer Output when TSG3nTS is Set to 1 (Initial Setting) in HT-PWM Mode

NOTES

1. TSG3nCMP0E = $0000E_H$, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H
2. T_{D0}: Time depending on the dead time setting in the TSG3nDTC0W register
   T_{D1}: Time depending on the dead time setting in the TSG3nDTC1W register
   T_{S1}: Time determined by compare match of the 18-bit sub-counter and TSG3nCMPUE register, when TSG3nCMPUE >18-bit counter maximum value

(b) Example of Timer Output during TSG3n Timer Operation

The following figure shows the timing chart when TSG3nCMP0E = $0000E_H$, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H, and TSG3nCMPUE is set to 00000_H-00014_H (excerpt). In this example, TSG3nIOC2.TSG3nOL1-TSG3nOL6 = 000000_B.

The range of the active (high level) width of a positive phase (TSG3nO1) output is $00000_H \leq$ TSG3nCMPUE $\leq$ TSG3nCMP0E (for the additional pulse). The range of the active (high level) width of an inverse phase (TSG3nO2) output is TSG3nDTC0 + TSG3nDTC1 $\leq$ TSG3nCMPUE $\leq$ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1.

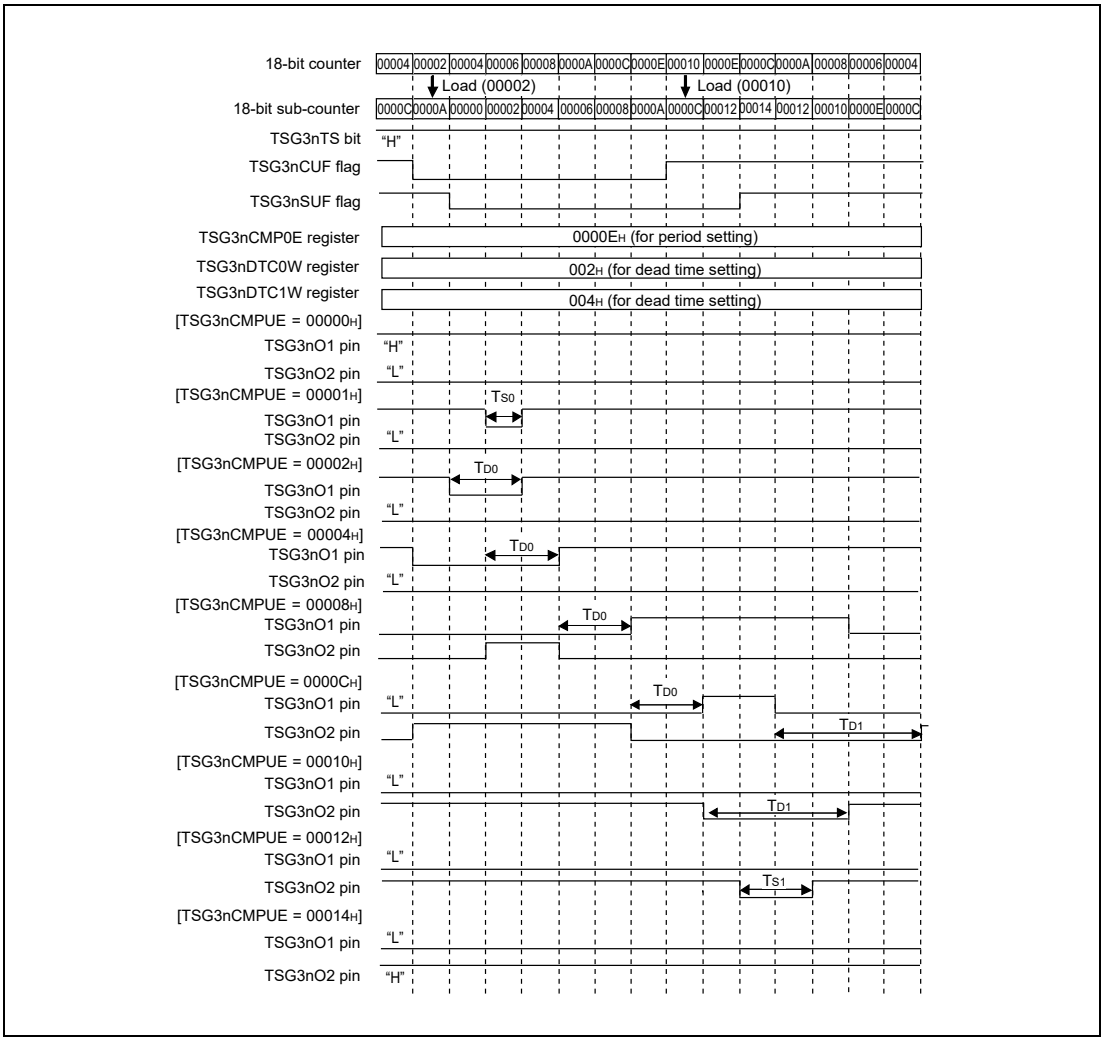


Figure 35.61 Example of Timer Output during TSG3n Operation in HT-PWM Mode

NOTES

1. TSG3nCMP0E = $0000E_H$, TSG3nDTC0 = 002_H, TSG3nDTC1 = 004_H
2. T_{D0}: Time depending on setting of the dead time in the TSG3nDTC0 register
   T_{D1}: Time depending on setting of the dead time in the TSG3nDTC1 register
   T_{S0}: Time determined by compare match of 18-bit sub-counter and the TSG3nCMPUE register, when TSG3nCMPUE < 18-bit counter minimum value
   T_{S1}: Time determined by compare match of the 18-bit sub-counter and the TSG3nCMPUE register, when TSG3nCMPUE > 18-bit counter maximum value

(6) Additional Pulse Control in HT-PWM Mode

The HT-PWM mode can generate an additional pulse by setting 1 to the LSB of the duty setting registers (TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE). This allows more precise control of the pulse duty than standard pulse control.

The following sections describe two examples of pulse output of TSG3nO1: additional pulse control is used in one example and additional pulse control is not used in another.

(a) Example of Pulse Output when Additional Pulse Control Is Used

Figure 35.62 shows the additional pulse control when an odd value is set to TSG3nCMPUE.

The arrows and numerical values show the duty cycle width of the TSG3nO1 output in one period.

As shown in **Figure 35.62**, when the additional pulse control is used, the output width of the TSG3nO1 (duty cycle) can be set within a range from 12 clock cycles to 0 clock cycles in one-clock-cycle step.

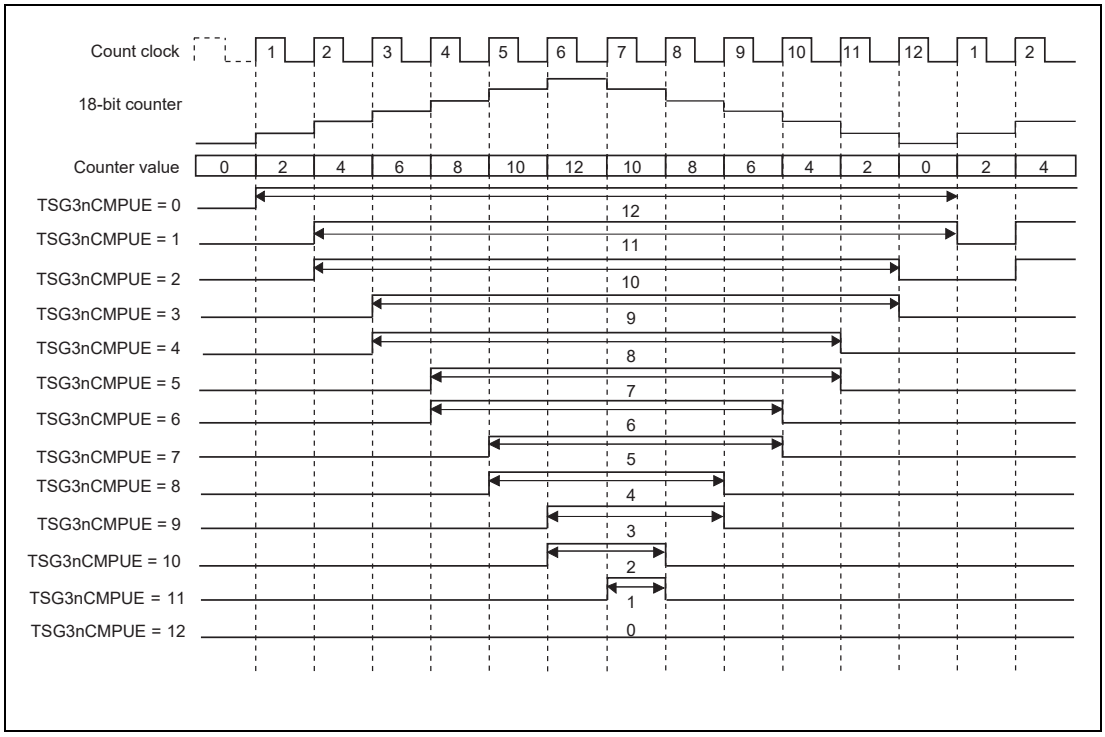


Figure 35.62 Example of TSG3nO1 Output when Additional Pulse Control Is Used in HT-PWM Mode

NOTE

TSG3nCMP0E = 12, TSG3nDTC0 = 0, TSG3nDTC1 = 0

(b) Example of Pulse Output when Additional Pulse Control Is Not Used

The arrows and numerical values in **Figure 35.63** show the duty cycle width of the TSG3nO1 output in one period.

When the additional pulse control is not used, the width of the TSG3nO1 output can be set within a range from 12 clock cycles to 0 clock cycles in two-clock-cycle step. In this case, the change in duty cycle is larger than that of the case when the additional pulse control is used.

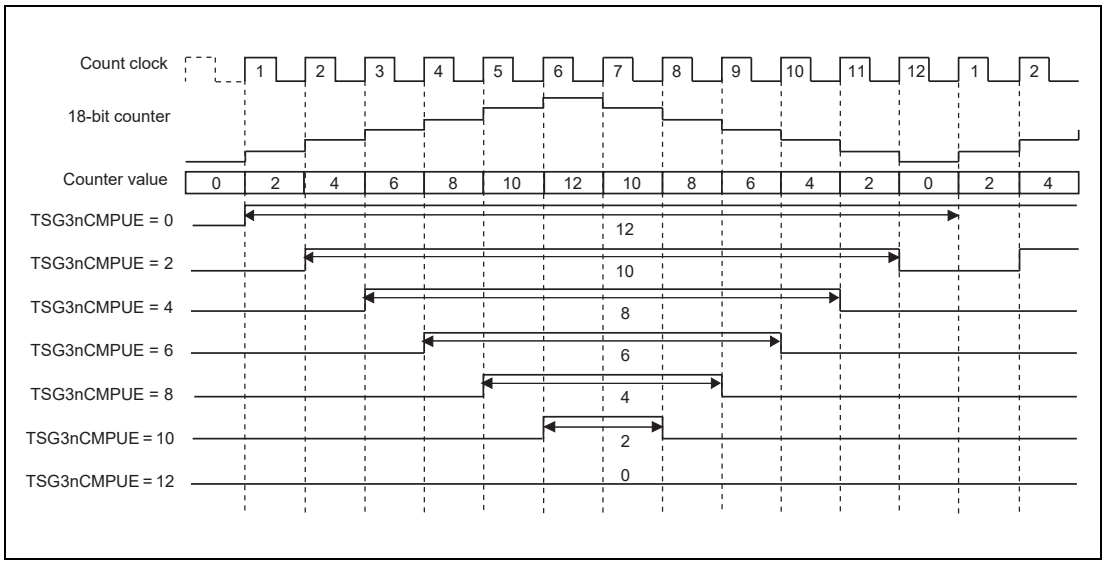


Figure 35.63 Example of TSG3nO1 Output when Additional Pulse Control Is Not Used in HT-PWM Mode

NOTE

TSG3nCMP0E = 12, TSG3nDTC0 = 0, TSG3nDTC1 = 0

(7) Dead Time Control in HT-PWM Mode

Duty setting are basically configured with the TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE registers in HT-PWM mode. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are two dead time control registers (TSG3nDTC0W and TSG3nDTC1W) and six 10-bit down counters that operate synchronously with the count clock of the 18-bit counter. TSG3nDTC0 is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSG3nDTC1 is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state.

The following figure shows the output waveform when TSG3nDTC0 = x and TSG3nDTC1 = y.

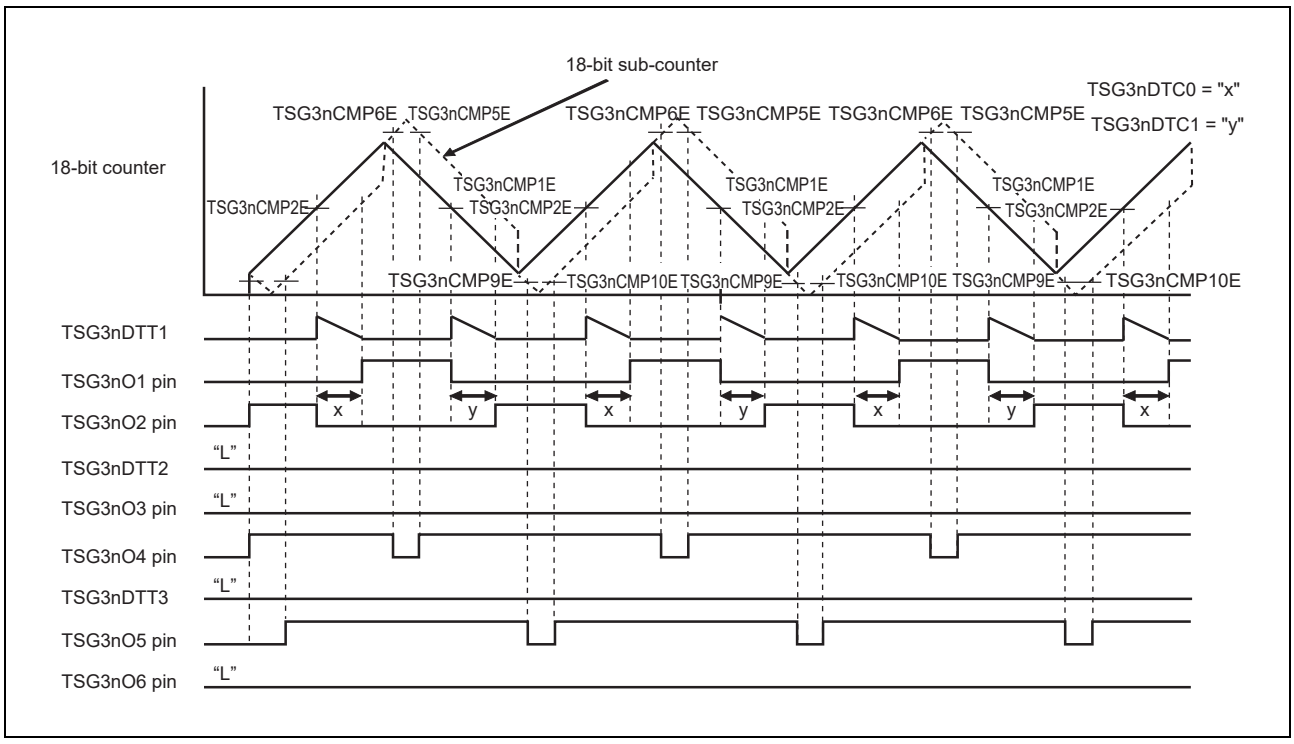


Figure 35.64 Example of Output Waveform with Dead Time in HT-PWM Mode

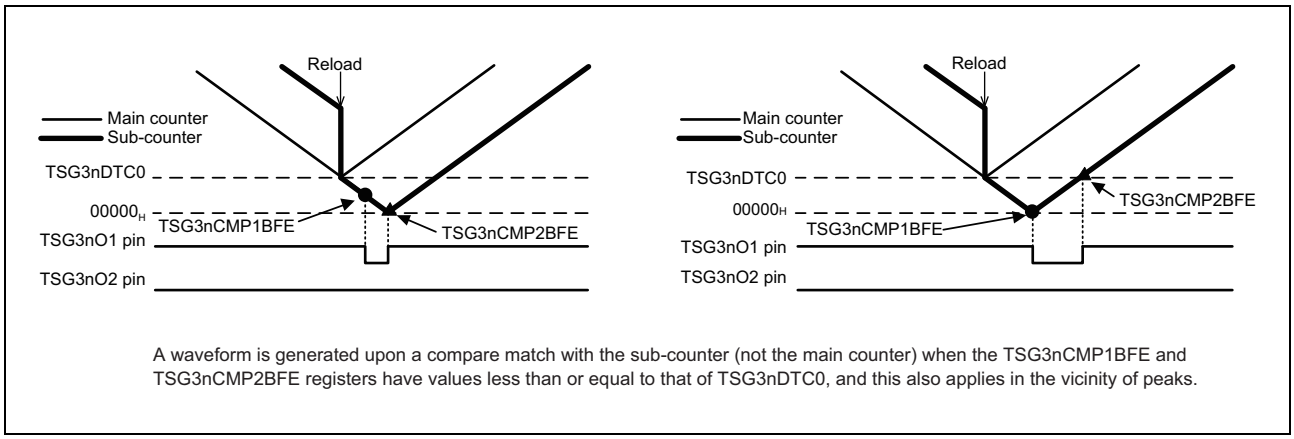


Figure 35.65 Example of Output Waveform near Trough after Reloading

(8) Notes Concerning Dead Time Control in HT-PWM Mode

(a) TSG3nDTC0 and TSG3nDTC1 Rewriting

It is possible to rewrite the dead time setting in TSG3nDTC0 and TSG3nDTC1 registers during timer operation.

CAUTIONS

1. **Rewrite TSG3nDTC0 and TSG3nDTC1 when the reload function is used (TSG3nRMC = 0).**
2. **The write protection code check function is applied when TSG3nDTC0 and TSG3nDTC1 are rewritten. For details, see the pertinent register descriptions (Section 35.3.43, Section 35.3.44, Section 35.3.63).**
3. **When the TSG3nCMP0E and TSG3nDTC1 are updated at the peak of the 18-bit counter:**
   If the set value of TSG3nCMPmE is greater than the updated TSG3nCMP0E + TSG3nDTC0 value (new maximum value of the main counter), the match interrupt (INTTSG3nIm) is not generated immediately after reloading (m = 2, 6, or 10).

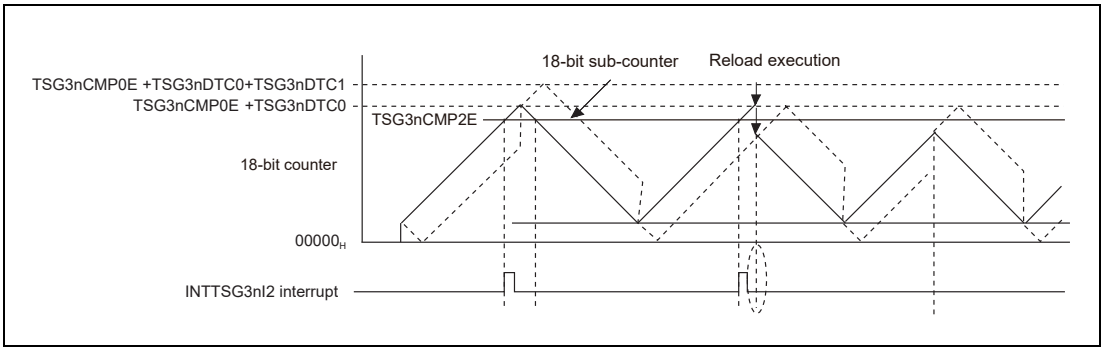


Figure 35.66 Example of Operation During Reloading at 18-Bit Counter Peak Timing

4. **When the TSG3nDTC0 is updated at the valley of the 18-bit counter:**
   If the TSG3nCMPmE set value is smaller than the updated TSG3nDTC0 value (new minimum value of the main counter), the match interrupt (INTTSG3nIm) is not generated immediately after reloading (m = 1, 5, or 9).

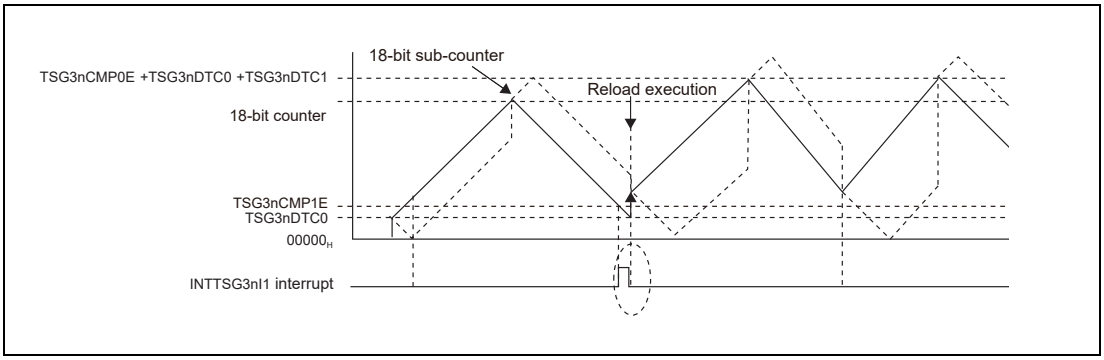


Figure 35.67 Example of Operation During Reloading at 18-Bit Counter Trough Timing

(9) Software Output Control Function in HT-PWM Mode

TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT1.TSG3nSPC2-TSG3nSPC0 are used in HT-PWM mode for software control of timer output control.

As shown in **Figure 35.68**, with TSG3nSTE = 0, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSG3nSOC is set to 0, output control is retained and at the reload timing, output control is switched to HT-PWM mode output control.

For details, refer to **Section 35.4.7.8, Software Output Control Function**.

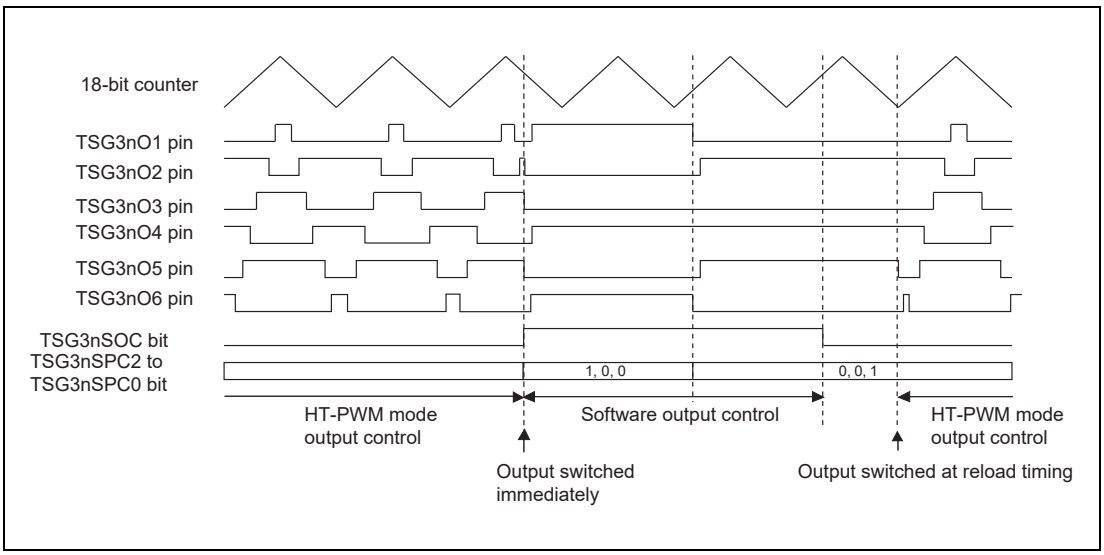


Figure 35.68 Example of Switching from HT-PWM Mode Control to Software Output Control

CAUTION

Use reload (simultaneous rewrite) mode (TSG3nCTL3.TSG3nRMC = 0) when software output control function is used.

(a) Procedure for Software Output Control

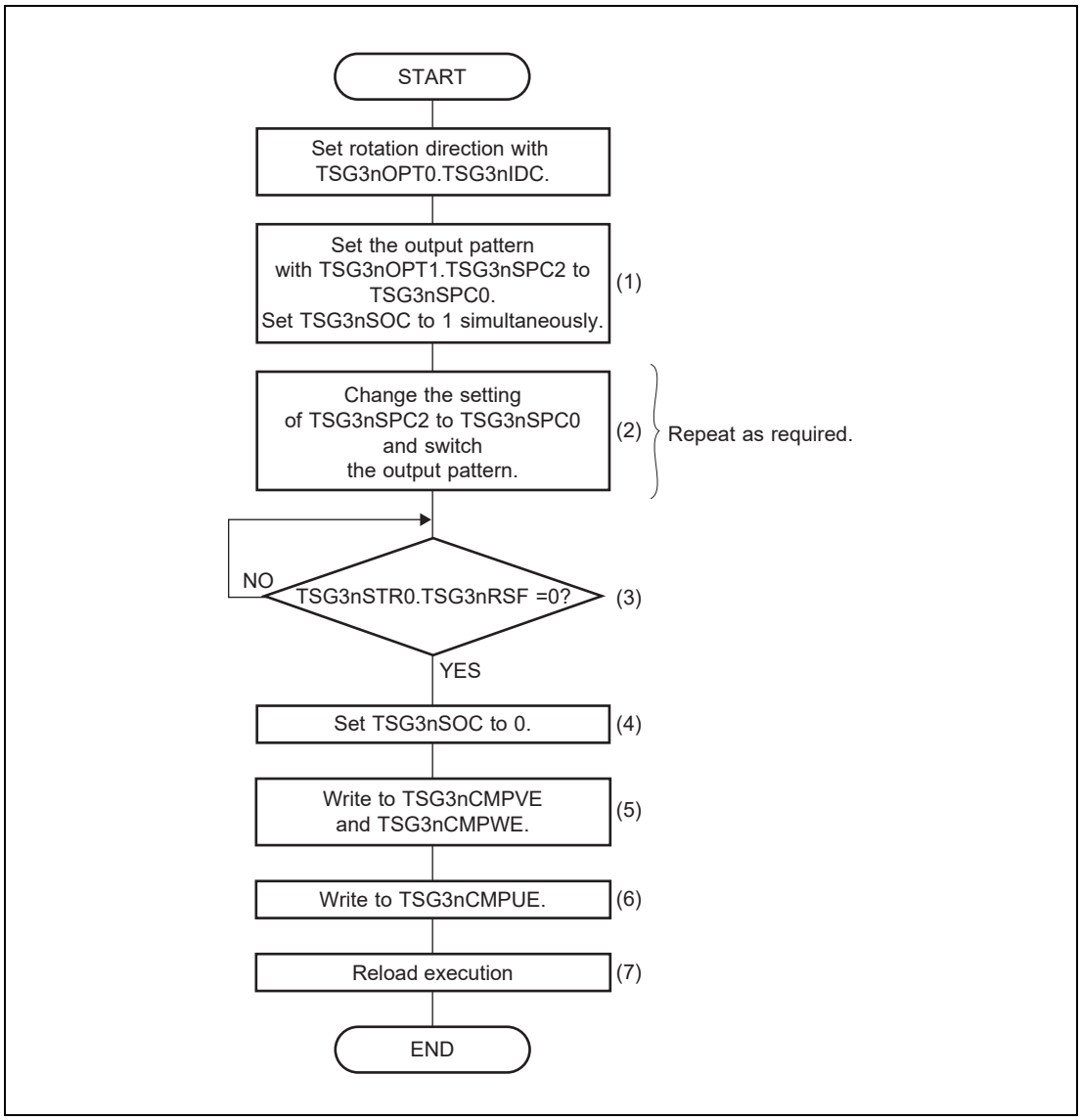


Figure 35.69 Flow of Software Output Control in HT-PWM Mode

The procedure for software output control is described below.

(1) Set the output pattern to the TSG3nOPT1.TSG3nSPC2-TSG3nSPC0. To enable software output control, set TSG3nOPT0.TSG3nSOC to 1 simultaneously.

(2) Change the output pattern setting of the TSG3nSPC2-TSG3nSPC0 to change the timer output.

(3) Confirm that reload request flag TSG3nSTR0.TSG3nRSF = 0. In case TSG3nRSF = 1, do not proceed to the following step until TSG3nRSF = 0.

(4) By setting TSG3nSOC = 0, the software control release process starts (it is not released yet at this point).

(5) Set the compare register settings that are required after the software output control is released. Proceed to the following step when the register setting is not required. When configuring the registers with the reload function, set them at this point.

(6) Write to TSG3nCMPUE (TSG3nCMP1E) to start reloading.

(7) Reload is executed and software output is released.

CAUTION

Execute reload after completing steps (3), (4), (5), and (6). When reload cannot be executed, the software output cannot be released.

(10) Asymmetric Triangular Wave Control in HT-PWM Mode

In HT-PWM mode, it is possible to control output by an asymmetric triangular waveform by setting the different timings for setting and clearing the U, V, and W phases.

The following describes the differences of the asymmetric triangular wave control from the symmetric triangular wave control.

(a) PWM Setting

When a symmetric triangular wave is used, the output control of each of the U phase, V phase, and W phase is done by setting the set timing and the clear timing to the same value in TSG3nCMPUE, TSG3nCMPVE, and TSG3nCMPWE, respectively. When an asymmetric triangular wave is used, the output control of each phase is done by setting TSG3nCMPmE as follows (m = 1, 2, 5, 6, 9, 10).

Prerequisites

- The clear timing of PWM of the voltage data signal of U, V and W phases is set with TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E.
- The set timing of PWM of the voltage data signal of U, V, and W phases is set with TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E.
- The set and clear timings of each phase can also be set with TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E.
- TSG3nCMPmE can only be set to an even value (m = 1, 2, 5, 6, 9, 10).

(b) Timer Output

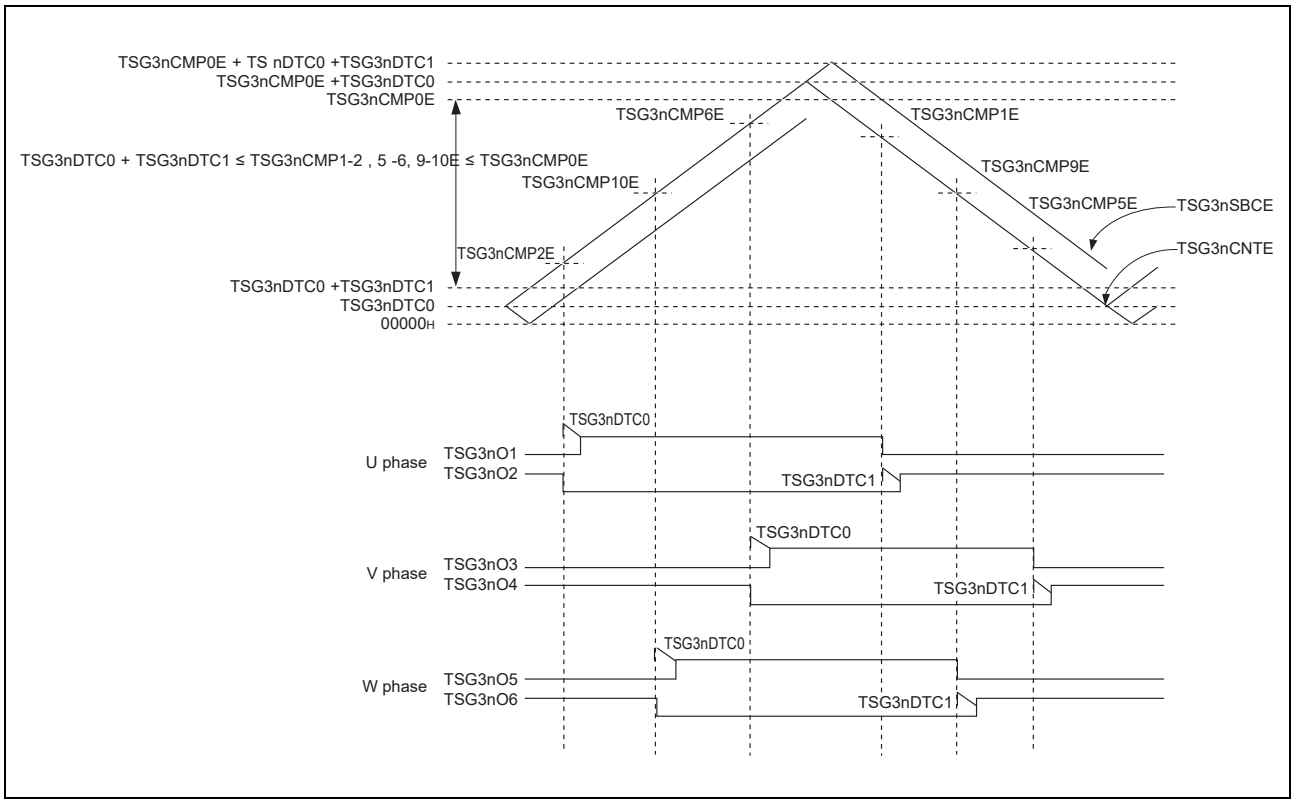


Figure 35.70 Example of Timer Output Waveform in HT-PWM Mode

NOTE

When output is controlled by the asymmetric triangular waveform, the following conditions apply to setting of TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10).

- TSG3nDTC0+TSG3nDTC1 ≤ TSG3nCMPmE ≤ TSG3nCMP0E
- Only when TSG3nCMPmE = TSG3nCMP (m + 1) E, or TSG3nCMPmE = TSG3nCMP (m + 1) E + 2, it is possible to set TSG3nCMPmE under the condition 00000_H ≤ TSG3nCMPmE ≤ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1, which also applies to the case where the symmetric triangular wave is used.

35.4.7.3 SP-PWM Mode (Shifted-pulse - Pulse Width Modulation Mode)

Overview

In this mode, a 6-phase PWM can be generated using the 18-bit counter and the 18-bit compare registers.

Prerequisites

- The PWM signal cycle is set in TSG3nCMP0E.
- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, while the clear timings of these phases are configured using TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E (when set timing and clear timing are used for control).
- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E while the active periods of these phases are configured using TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE. The sum of the set values of TSG3nCMP2E, TSG3nCMP6E, TSG3nCMP10E, and the set values of TSG3nUPWE, TSG3nVPWE, TSG3nWPWE are set to TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E respectively (when set timing and active period are used for control). The value after addition must be no greater than $3FFFF_H$. Otherwise, values having 19 or more bits are truncated.

Functional description

In this mode, the carrier period, the set timings, and the duty cycle of the U phase, V phase, and W phase are configured. The counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter counts up from 00000_H and is cleared by match with TSG3nCMP0E.

The dead time is set with TSG3nDTC0 and TSG3nDTC1. TSG3nDTC0 sets the dead time of inverse phase (OFF) to positive phase (ON) switch while TSG3nDTC1 sets that of positive phase (OFF) to inverse phase (ON) switch. The 10-bit counters (TSG3nDTT1 to TSG3nDTT3) for dead time generation load the set values of TSG3nDTC0 and TSG3nDTC1 at compare match of the 18-bit counter with the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) and start down-counting.

INTTSG3nIm interrupts (m = 1, 2, 5, 6, 9, 10) are generated by the compare match of the 18-bit counter with the TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E buffer registers.

NOTE

SP-PWM mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 010_B.

(1) Basic Timing Chart

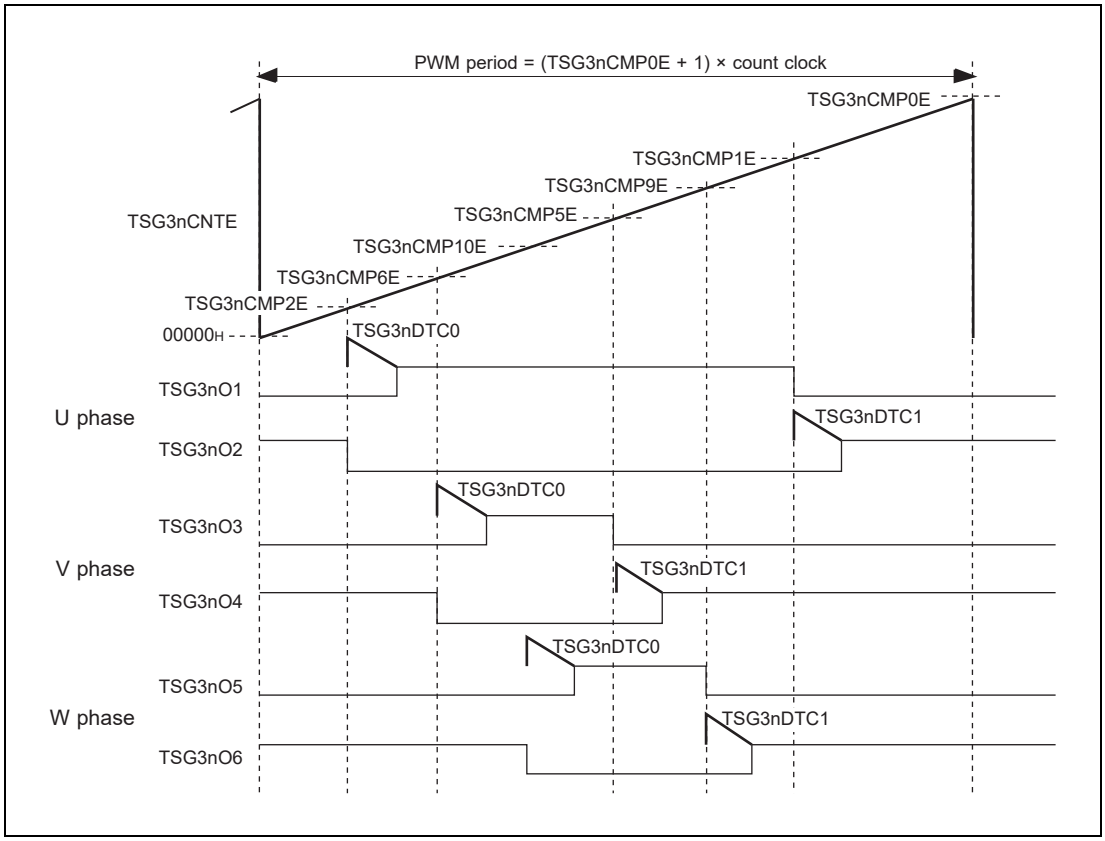


Figure 35.71 Basic Timing in SP-PWM Mode

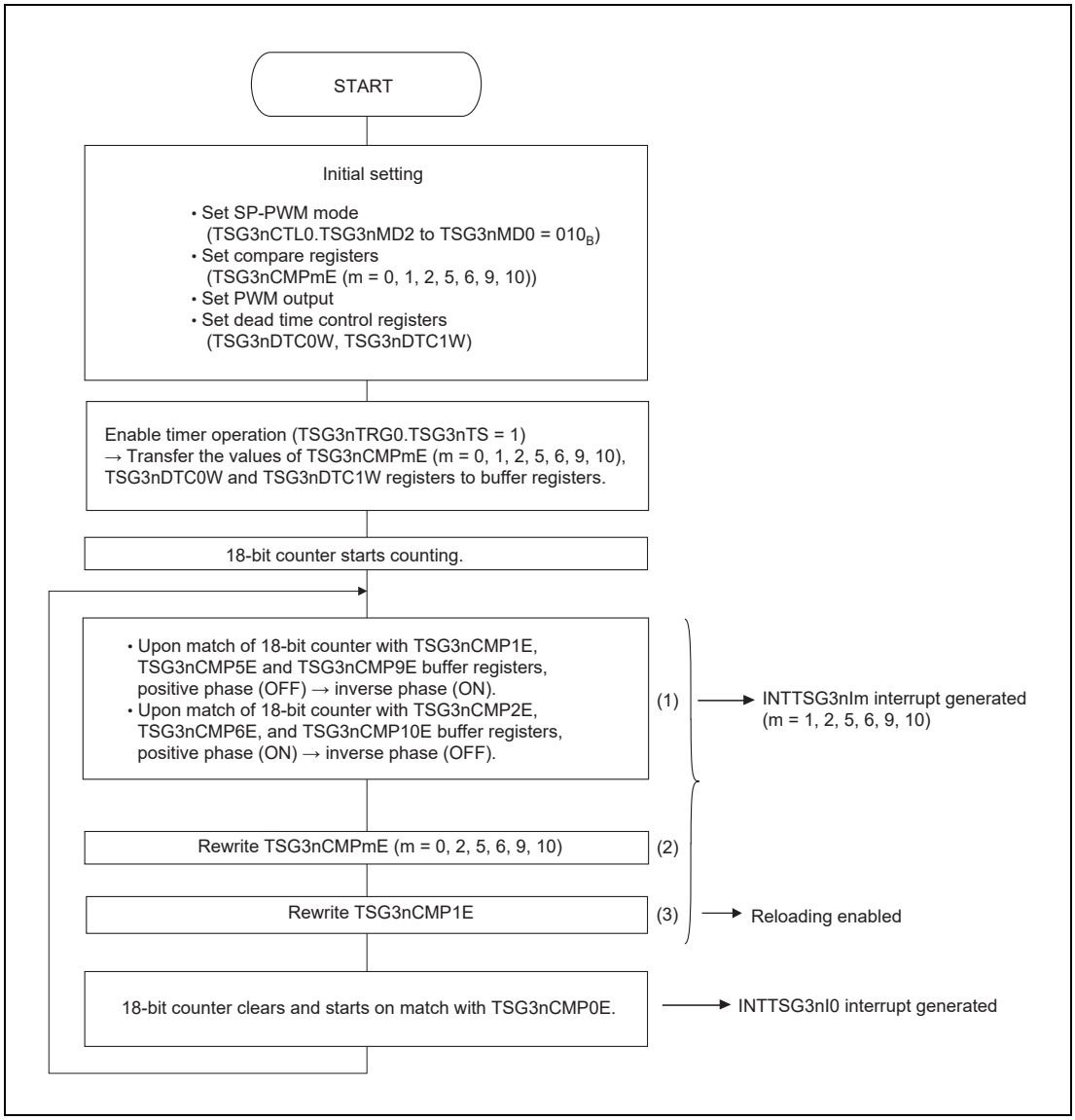


Figure 35.72 Basic Operation Flow in SP-PWM Mode

NOTE

The timing of (1) may be different depending on the rewriting timing of (2) and (3) and the TSG3nCMP1E value. Be sure to rewrite (2) before rewriting (3).

(2) List of SP-PWM Mode Operations

Table 35.68 Counter Functions in SP-PWM Mode

Operation		Setting Condition
18-bit counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger
	Clear	Compare match of TSG3nCMP0E buffer register with 18-bit counter
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1

Table 35.69 Compare Registers and Dead Time Control Register Functions in SP-PWM Mode

Register	Rewriting Method	Rewrite during Operation	Function
TSG3nCMP0E	Reload/Anytime rewrite	Possible	Setting period
TSG3nUPWE	Reload/Anytime rewrite	Possible	PWM control for U phase
TSG3nCMP1E, TSG3nCMP2E	Reload/Anytime rewrite		
TSG3nVPWE	Reload/Anytime rewrite	Possible	PWM control for V phase
TSG3nCMP5E, TSG3nCMP6E	Reload/Anytime rewrite		
TSG3nWPWE,	Reload/Anytime rewrite	Possible	PWM control for W phase
TSG3nCMP9E, TSG3nCMP10E	Reload/Anytime rewrite		
TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E	Reload/Anytime rewrite	Possible	Diagnostic output or A/D conversion trigger
TSG3nDTC0W, TSG3nDTC1W	Reload	Possible	Period and dead time

Table 35.70 Timer Input Function in SP-PWM Mode

Pin	Function
TSG3nCLKI	Clock enable input

Table 35.71 Output Functions in SP-PWM Mode

Pin	Function
TSG3nO1	PWM output with dead time by compare match of TSG3nCMP1E buffer register (clear timing) or TSG3nCMP2E buffer register (set timing) with 18-bit counter
TSG3nO2	Output inverse phase with respect to TSG3nO1 (with dead time)
TSG3nO3	PWM output with dead time by compare match of TSG3nCMP5E buffer register (clear timing) or TSG3nCMP6E buffer register (set timing) with 18-bit counter
TSG3nO4	Output inverse phase with respect to TSG3nO3 (with dead time)
TSG3nO5	PWM output with dead time by compare match of the TSG3nCMP9E buffer register (clear timing) or the TSG3nCMP10E buffer register (set timing) with the 18-bit counter
TSG3nO6	Output inverse phase with respect to TSG3nO5 (with dead time)
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Table 35.72 Interrupt Requests in SP-PWM Mode

Interrupt	Function
INTTSG3nIm (m = 0, 1, 2, 5, 6, 9, 10)	Compare match of TSG3nCMPmE buffer register with 18-bit counter (m = 0, 1, 2, 5, 6, 9, 10)
INTTSG3nIER	Error
INTTSG3nIVLY	—
INTTSG3nIPEK	Peak interrupt (generated at the same timing as INTTSG3nI0 interrupt)
INTTSG3nIWN	Warning

Table 35.73 Compare Match Timing in SP-PWM Mode

Compare Match	Timing
TSG3nCMP0E	When 18-bit counter changes from TSG3nCMP0E to 00000_H
TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10)	After match of 18-bit counter and TSG3nCMPmE is detected (m = 1, 2, 5, 6, 9, 10)

Table 35.74 Example of Setting Each Timer Output Condition in SP-PWM Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG3nO1, TSG3nO3, TSG3nO5	PWM output	(TSG3nCMP0E + 1) × count clock	Output an inactive level throughout one period (duty 0%)	TSG3nCMPmE = TSG3nCMP (m + 1) E or TSG3nCMP (m + 1) E > TSG3nCMP0E (m = 1, 5, 9)
			Output an active level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m + 1) E + 1 TSG3nCMP (m + 1) E = TSG3nCMPmE – 1 (m = 1, 5, 9)
			Output an inactive level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m + 1) E - 1 TSG3nCMP (m + 1) E = TSG3nCMPmE + 1 (m = 1, 5, 9)
			Output an active level throughout one period (duty 100%)	TSG3nCMPmE > TSG3nCMP0E TSG3nCMP (m + 1) E ≤ TSG3nCMP0E (m = 1, 5, 9)
TSG3nO2, TSG3nO4, TSG3nO6	PWM output	(TSG3nCMP0E + 1) × count clock	Output an inactive level throughout one period (duty 0%)	TSG3nCMP (m – 1) E > TSG3nCMP0E (m = 2, 6, 10)
			Output an active level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m – 1) E - 1 TSG3nCMP (m – 1) E = TSG3nCMPmE + 1 (m = 2, 6, 10)
			Output an inactive level of one count clock in one period	TSG3nCMPmE = TSG3nCMP (m – 1) E + 1 TSG3nCMP (m – 1) E = TSG3nCMPmE – 1 (m = 2, 6, 10)
			Output an active level throughout one period (duty 100%)	TSG3nCMPmE = TSG3nCMP(m – 1)E or TSG3nCMPmE > TSG3nCMP0E (m = 2, 6, 10)
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger	(TSG3nCMP0E + 1) × count clock	Refer to **Section 35.4.5, A/D Conversion Trigger Function**.	

(3) Various Settings of SP-PWM Mode

Setting mode

SP-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 010_B.

Setting timer output

The output pins TSG3nO1-TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of the positive and inverse phases is enabled by setting TSG3nIOC1.TSG3nEOC to 1. For details, refer to **Section 35.4.6, Error/Warning Interrupt**.

Setting rewriting timing of register with reload function

With the TSG3nCTL3.TSG3nRMC, reload (simultaneous rewrite) or rewrite (anytime) is specified for the registers with reload function. The default setting is 0 (reload). To reload, set TSG3nCTL4.TSG3nPRE to 1.

No reload timing is generated when TSG3nPRE = 0.

When “anytime rewrite” is specified, unintended output may be generated depending on the rewrite timing.

Setting A/D conversion trigger output

The A/D conversion trigger 0 (TSG3nADTRG0 signal) is set with TSG3nCTL5.TSG3nAT09 to TSG3nAT00.

TSG3nAT09 to TSG3nAT00 is used to enable or disable the A/D conversion trigger output at the match of TSG3nDCMP2E to TSG3nDCMP0E with the 18-bit counter (up count).

TSG3nCTL6.TSG3nAT19 to TSG3nAT10 is used to set the A/D conversion trigger 1 (TSG3nADTRG1 signal).

To set the match timing of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to each register.

The skipping function can be used for TSG3nADTRG0 and TSG3nADTRG1 signals. TSG3nACC00 and TSG3nACC01 of TSG3nCTL5, and TSG3nACC10 and TSG3nACC11 of TSG3nCTL6 can be used to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- **Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.**
- **In SP-PWM mode, no valley interrupt (INTTSG3nIVLY) is generated. Therefore, TSG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.**
- **In SP-PWM mode, the 18-bit sub-counter does not operate. Therefore, TSG3nAT09 and TSG3nAT08 in TSG3nCTL5, and TSG3nAT19 and TSG3nAT18 in TSG3nCTL6 must be set to 0.**
- **In SP-PWM mode, down counting by the 18-bit counter is not generated. Therefore, TSG3nAT07, TSG3nAT05, and TSG3nAT03 in TSG3nCTL5 and TSG3nAT17, TSG3nAT15, and TSG3nAT13 in TSG3nCTL6 should be set to 0.**

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

PCLK × TSG3nDTC0

PCLK × TSG3nDTC1

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state, and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

TSG3nCMP0E = (carrier period/count clock period) – 1

CAUTION

PWM output with 100% duty cannot be produced when TSG3nCMP0E = $3FFFF_H$.

Setting duty (PWM width)

The duty of U phase, V phase, and W phase is set with TSG3nCMPmE, TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE (m = 1, 2, 5, 6, 9, 10), respectively.

- The set timings of the U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, and the clear timings are configured using TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E (when set timing and clear timing are used for control).
- The set timings of U phase, V phase, and W phase are configured using TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E while the active periods of these phases are configured using TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE.
  The sum of the set values of TSG3nCMP2E, TSG3nCMP6E, and TSG3nCMP10E, and the set values of TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE are set to TSG3nCMP1E, TSG3nCMP5E, and TSG3nCMP9E respectively (when set timing and active period are used for control).

(4) Dead Time Control in SP-PWM mode

Duty setting registers are TSG3nCMPmE (m = 1, 2, 5, 6, 9, 10), TSG3nUPWE, TSG3nVPWE, and TSG3nWPWE and register for setting the period is TSG3nCMP0E. The 6-phase PWM waveform of a variable duty is output by using these registers. To achieve the dead time control, there are six 10-bit down counters that operate synchronously with the count clock of the 18-bit counter and two dead time control registers (TSG3nDTC0W and TSG3nDTC1W). TSG3nDTC0W is used for setting a dead time from a change of the inverse phase to the inactive state to a change of the positive phase to the active state. TSG3nDTC1W is used for setting a dead time from a change of the positive phase to the inactive state to a change of the inverse phase to the active state.

Dead time counter keeps operating even when operation stop (TSG3nTE = 0) setting collided with dead time insert timing, and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.

The following figure shows an example of the output waveform.

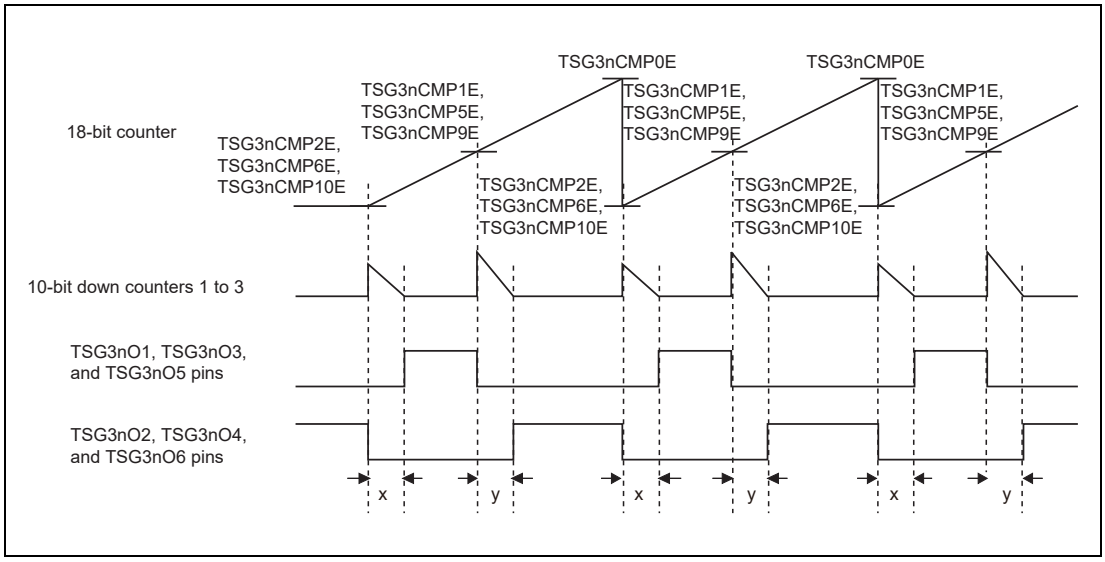


Figure 35.73 Example of Output Waveform in SP-PWM Mode

NOTE

x: TSG3nDTC0 register values, y: TSG3nDTC1 register values

(5) Software Output Control Function in SP-PWM Mode

TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are used to control timer output by software.

As shown in **Figure 35.74**, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time is guaranteed. After that, when TSG3nSOC is set to 0, output control is retained, and at the reload timing, output control is switched to SP-PWM mode output control.

For details, refer to **Section 35.4.7.8, Software Output Control Function**.

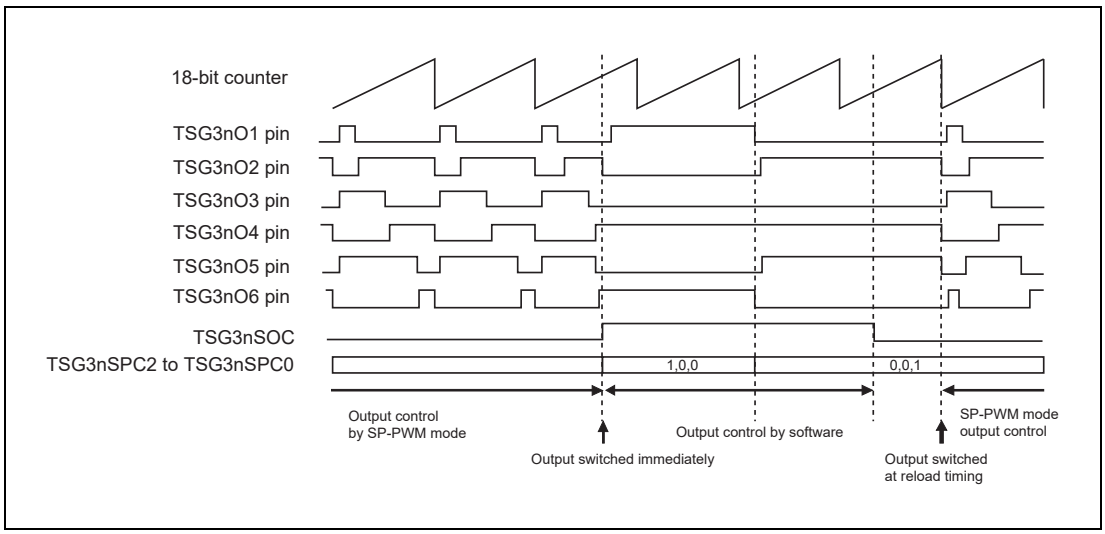


Figure 35.74 Example of Switching from SP-PWM Mode Control to Software Output Control

(a) Procedure for Software Output Control Processing

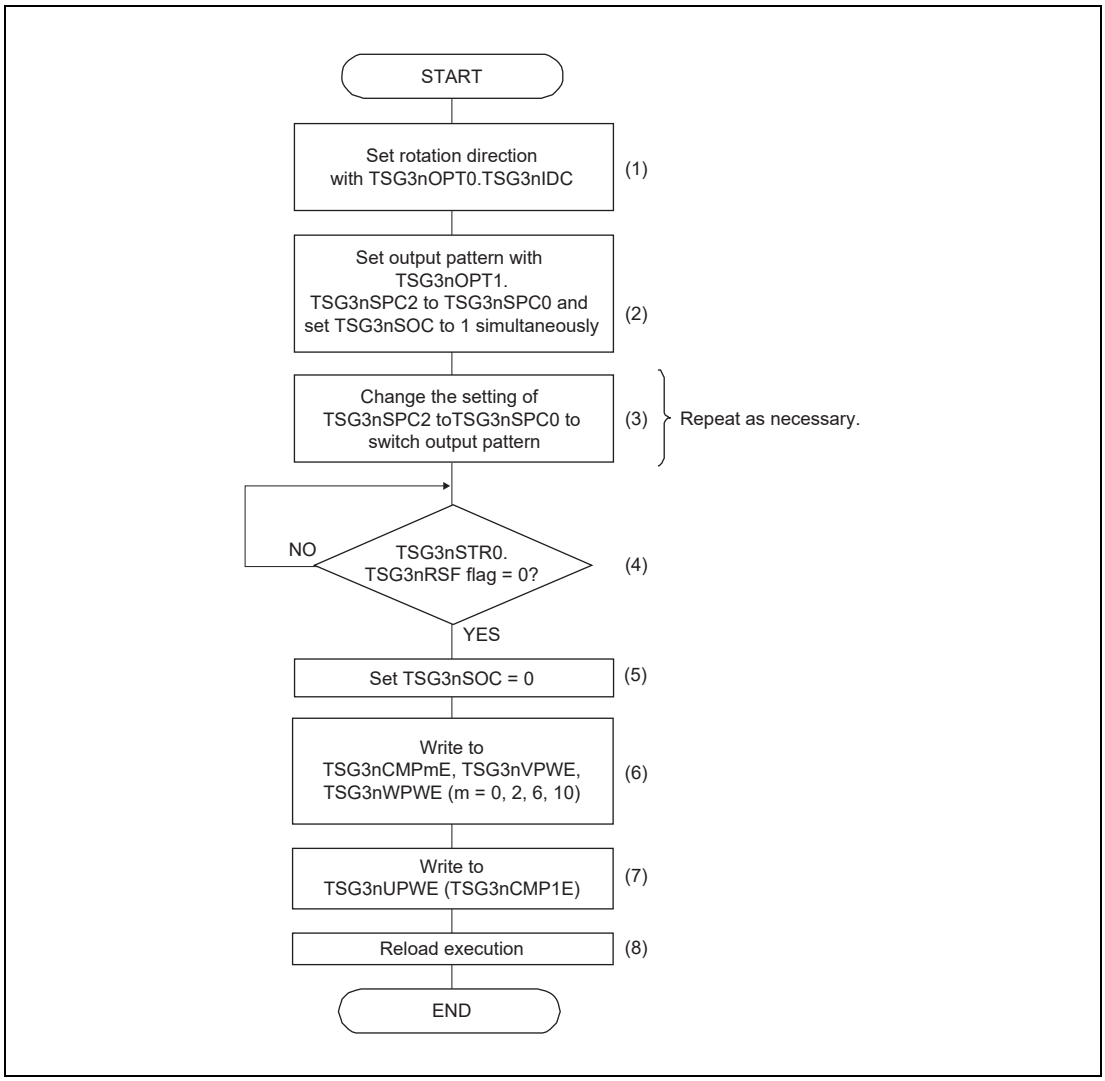


Figure 35.75 Flow of Software Output Control

The procedure for software output control is described below.

(1) Set TSG3nIDC to determine the electric current direction. The timer output has a 180-degree phase shift between when TSG3nIDC = 0 and when TSG3nIDC = 1. When this bit is rewritten with the software output control function, the output pattern will change according to the new setting at the next timer cycle.

(2) Set the output pattern in TSG3nSPC2-0. At the same time, set TSG3nSOC to 1 to switch to software output control mode.

(3) Change the output pattern setting for TSG3nSPC2-0 to change the timer output.

(4) Ensure that the reload request flag TSG3nRSF is 0. If TSG3nRSF is 1, do not proceed to the following step until TSG3nRSF = 0.

(5) By clearing TSG3nSOC to 0, software control release process starts (it is not released yet at this point).

(6) Configure the compare register settings that are required after the software output control is released. Proceed to the following step when the register configuration is not required. When configuring the registers with the reload function, configure them at this point.

(7) Write to TSG3nUPWE (TSG3nCMP1E) to start reloading.

(8) Reload is executed and software output control is released.

CAUTION

Be sure to execute reload after completing steps (4), (5), (6), and (7). Unless reload can be executed, software output control cannot be released

35.4.7.4 120-DC Mode

Overview

In this mode, PWM output period set to TSG3nCMP0E and timer output (TSG3nO1 to TSG3nO6) according to the duty cycle set with TSG3nCMP1E to TSG3nCMP12E are controlled with three types of pattern inputs (software output control method, pattern switch method, and trigger switch method) to perform 120-DC control.

Prerequisites

- Set the PWM period with TSG3nCMP0E.
- Set the PWM duty with TSG3nCMP1E to TSG3nCMP12E and set the output pattern with TSG3nPAT0W and TSG3nPAT1W.

Functional description

In this mode, configure the PWM period, the duty cycle with each compare register, and the pattern to be output with each pattern register. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter starts counting from 00000_H, and is cleared by the match with TSG3nCMP0E.

INTTSG3nI1 to INTTSG3nI12 interrupts are generated by a compare match of the 18-bit counter and TSG3nCMP1E to TSG3nCMP12E buffer registers, respectively.

NOTE

120-DC mode is enabled when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 011_B.

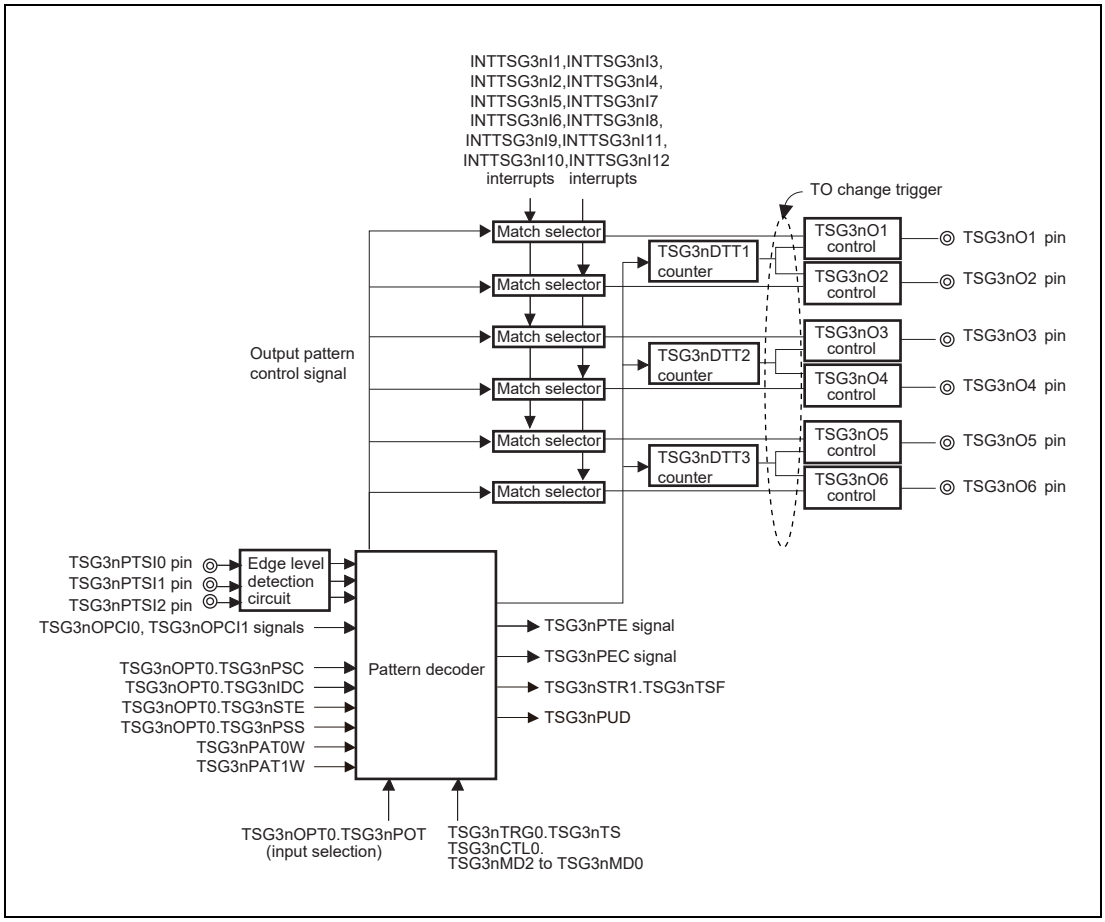


Figure 35.76 Block Diagram in 120-DC Mode

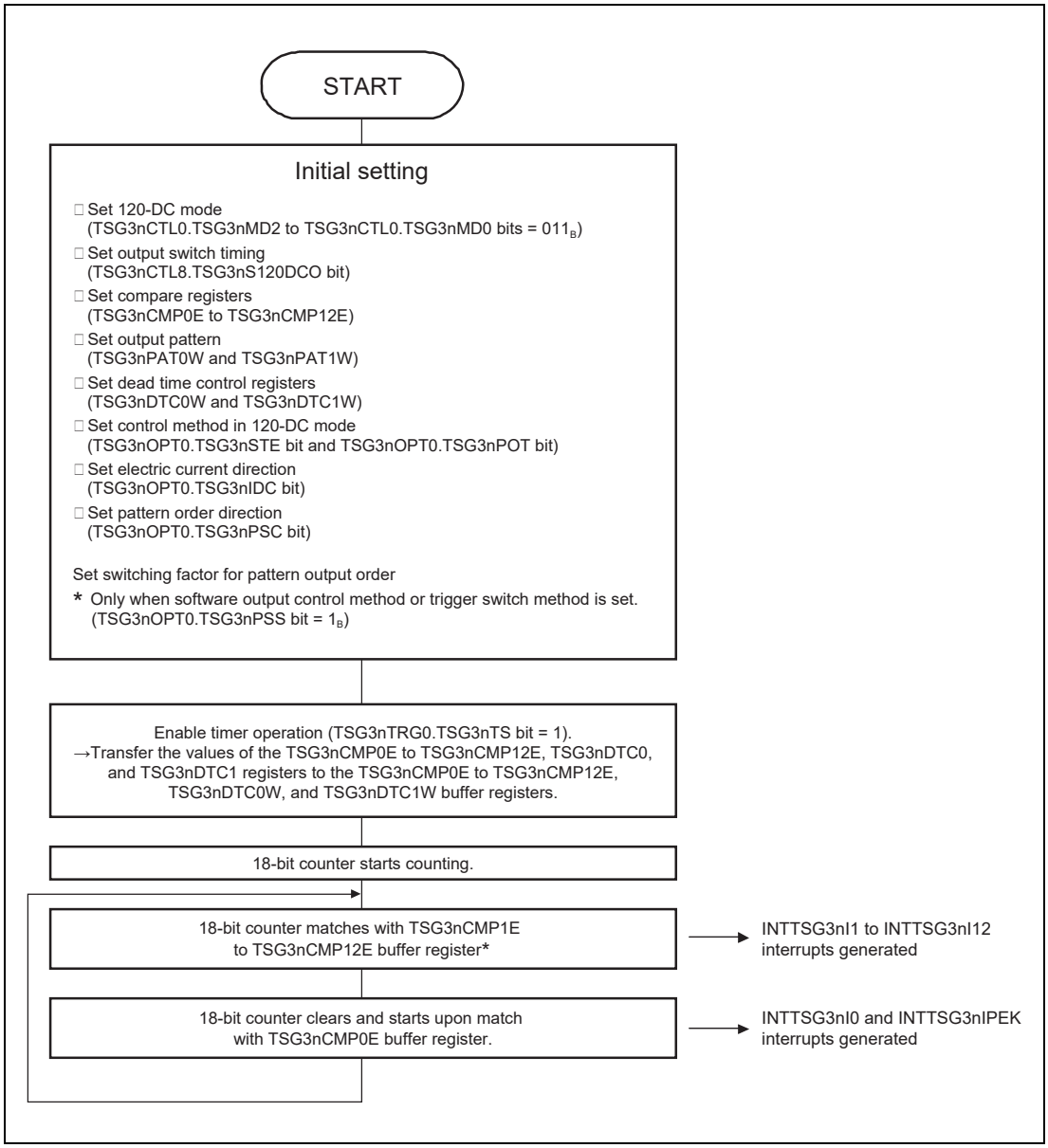


Figure 35.77 Basic Operation Flow in 120-DC Mode

NOTE

The 18-bit counter is not cleared by match of the 18-bit counter with the TSG3nCMP1E to TSG3nCMP12E buffer registers

(1) List of Operations in 120-DC Mode

Table 35.75 Counter Functions in 120-DC Mode

Operation		Setting Condition
18-bit counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger
	Clear	When TSG3nCTL8.TSG3nS120DCO is 0: timing of a match of TSG3nCMP0E value and 18-bit counter value or output pattern switch When TSG3nCTL8.TSG3nS120DCO is 1: timing of a match of TSG3nCMP0E value with 18-bit counter value
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1

Table 35.76 Functions of Compare Registers and Dead Time Control Registers in 120-DC Mode

Register	Rewrite Method	Rewrite during Operation	Function
TSG3nCMP0E	Reload	Possible	Setting period
TSG3nCMPmE (m = 1-12)	Reload	Possible	Setting PWM duty
TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E	Reload	Possible	Diagnostic signal output or A/D conversion trigger
TSG3nDTC0W, TSG3nDTC1W	Reload	Possible	Setting dead time

Table 35.77 Timer Input Function in 120-DC Mode

Pin/Signal	Function
TSG3nCLKI pin	Clock enable input
TSG3nPTSI2-TSG3nPTSI0 pins	Pattern input (3 phases)
TSG3nOPCI0, TSG3nOPCI1 signals	Trigger input

Table 35.78 Timer Output Function in 120-DC Mode

Pin/Signal	Function
TSG3nO1 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting.
TSG3nO2 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting.
TSG3nO3 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting.
TSG3nO4 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting.
TSG3nO5 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 1, 2, 5, 6, 9, 10) with the 18-bit counter and output pattern selected by TSG3nPAT0W setting.
TSG3nO6 pin	PWM output (with dead time) according to a compare match of the TSG3nCMPmE buffer register (m = 3, 4, 7, 8, 11, 12) with the 18-bit counter and output pattern selected by TSG3nPAT1W setting.
TSG3nO7 pin	Diagnostic signal output or pulse output by A/D conversion trigger
TSG3nPTE signal	Toggle signal by change in input pattern

Table 35.79 Interrupt Requests in 120-DC Mode

Interrupt	Function
INTTSG3nIm (m = 0 to 12)	Compare match of TSG3nCMPmE buffer register and 18-bit counter (m = 0 to 12)
INTTSG3nIER	Error
INTTSG3nIVLY	—
INTTSG3nIPEK	Peak interrupt (generated at the same timing as INTTSG3nI0)
INTTSG3nIWN	Warning interrupt

Table 35.80 Compare Match Timing in 120-DC Mode

Compare Match	Timing
TSG3nCMP0E	When 18-bit counter changes from TSG3nCMP0E to 00000_H
TSG3nCMPmE (m = 1 to 12)	After detecting the match of 18-bit counter and TSG3nCMPmE (m = 1 to 12)

Table 35.81 Example of Setting Each Timer Output Condition in 120-DC Mode

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TSG3nOm (m = 1 to 6)	PWM output	(TSG3nCMP0E + 1) × count clock	See **Section 35.4.7.4, (6) List of Output Patterns in 120-DC Mode**.	—
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger	(TSG3nCMP0E + 1) × count clock	See **Section 35.4.5, A/D Conversion Trigger Function**.	—

(2) Various Settings of 120-DC Mode

Setting mode

120-DC mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 011_B.

Setting timer output

The output pins TSG3nO1 to TSG3nO6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.
The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

With TSG3nIOC1.TSG3nEOC = 1, the error interrupt (INTTSG3nIER) generation is enabled when the simultaneous active state of the positive phase and inverse phase is detected. For details, see **Section 35.4.6, Error/Warning Interrupt**.

Setting register rewrite timing

Reloading the registers with the reload function is activated with TSG3nCTL3.TSG3nRMC (simultaneous rewrite; default setting is 0 = reload). To reload, set TSG3nCTL4.TSG3nPRE to 1.

The reload timing is not generated if TSG3nPRE is 0.

Setting A/D conversion trigger output

To set A/D conversion trigger 0 (TSG3nADTRG0 signal), use TSG3nCTL5.TSG3nAT09 to TSG3nAT00.
With TSG3nAT09 to TSG3nAT00, A/D conversion trigger output is enabled or disabled at the match of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E (during up count).
To set A/D conversion trigger 1 (TSG3nADTRG1 signal), use TSG3nCTL6.TSG3nAT19 to TSG3nAT10.

To set the match timing of the 18-bit counter and TSG3nDCMP2E to TSG3nDCMP0E, set the compare value to the pertinent register.
The skipping function can be used for TSG3nADTRG0 and TSG3nADTRG1 signals. Use TSG3nACC01, TSG3nACC00 of TSG3nCTL5, and TSG3nACC11, TSG3nACC10 of TSG3nCTL6 to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- **Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E to TSG3nDCMP0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.**
- **In 120-DC mode, a valley interrupt (INTTSG3nIVLY) is not generated. Therefore, TSG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.**
- **In 120-DC mode, the 18-bit sub-counter does not operate. TSG3nAT09 and TSG3nAT08 in TSG3nCTL5, and TSG3nAT19 and TSG3nAT18 in TSG3nCTL6 must be set to 0.**
- **In 120-DC mode, the 18-bit counter does not decrement. Therefore, TSG3nAT07, TSG3nAT05, and TSG3nAT03 in TSG3nCTL5 and TSG3nAT17, TSG3nAT15, and TSG3nAT13 in TSG3nCTL6 must be set to 0.**

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions

PCLK × TSG3nDTC0

PCLK × TSG3nDTC1

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state to a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Carrier period

Set the carrier period with TSG3nCMP0E according to the following expression:

TSG3nCMP0E = (carrier period/count clock period) – 1

Setting duty (PWM width)

The duty of PWM output is set with TSG3nCMP1E to TSG3nCMP12E. The setting range of the compare registers is as follows:

$00000_H \leq$ TSG3nCMPmE $\leq$ TSG3nCMP0E + 1

CAUTION

Do not set TSG3nCMPmE = TSG3nCMP0E + 1 (m = 1 to 12) only when TSG3nCMP0E + 1 < TSG3nCMPmE, and TSG3nCMP0E = $3FFFF_H$.

Setting output PWM

In 120-DC mode, the output pins TSG3nO1, TSG3nO3, and TSG3nO5 are controlled by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E, and the output pins TSG3nO2, TSG3nO4, TSG3nO6 are controlled by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E. The duty cycle of a PWM period (TSG3nCMP0E) can be set with TSG3nCMP1E to TSG3nCMP12E. Setting TSG3nCMP1E to TSG3nCMP12E to 00000_H sets the PWM duty cycle to 0%. Setting TSG3nCMP1E to TSG3nCMP12E to TSG3nCMP0E + 1 value sets the PWM duty cycle to 100%. This allows chopping output control and rectangular wave output control.

(3) Control Methods in 120-DC Mode

Control methods in 120-DC mode are listed below.

Table 35.82 Control Method in 120-DC Mode

Control Method	Function
Software output control method	Switches the output pattern according to the TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 setting made by software.
Pattern switch method	Switches the output pattern by the pattern input signal of TSG3nPTSI0 to TSG3nPTSI2.
Trigger switch method	Switches the output pattern by the trigger switch method using the trigger input signals TSG3nOPCI0 and TSG3nOPCI1 or by the pattern input setting of TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 in the constant order.

Switch timing of timer output when changing input pattern of 120-DC mode can be set by TSG3nCTL8.TSG3nS120DCO.

Table 35.83 Setting of TSG3nS120DCO and Operation in 120-DC Mode

TSG3nS120DCO	Function
0	When input patterns are changed, the main counter (TSG3nCNTE) is cleared and the change of patterns is immediately reflected to timer output.
1	When input patterns are changed, the change of input patterns is reflected to timer output at the next timer cycle (after a match of the main counter (TSG3nCNTE) with TSG3nCMP0E).

Setting software output control method

Setting TSG3nOPT0.TSG3nSTE = 0 enables switching of output pattern by software output control. The TSG3nO1 to TSG3nO6 pin output is switched according to the settings of TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0.

The pattern output order at the beginning of operation is set with TSG3nOPT0.TSG3nIDC and TSG3nOPT0.TSG3nPSC.

Operation of software output control method

The PWM output of TSG3nO1 to TSG3nO6 pins (PWM output defined by TSG3nCMP1E to TSG3nCMP12E) is selected using TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 configured by software. As for the control of the dead time, the dead time counter is activated at the falling edge of the signals in each phase and the dead time is inserted.

The 18-bit counter counts according to the carrier period set in TSG3nCMP0E. The 18-bit counter is cleared by match of the 18-bit counter and TSG3nCMP0E or by a write access to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 (when TSG3nS120DCO = 0).

In this method, the pattern, which is decoded using information consisting of the output pattern (TSG3nSPC2 to TSG3nSPC0), the electric current direction control bit (TSG3nOPT0.TSG3nIDC), and order detection control bit (TSG3nOPT0.TSG3nPSC), is output. **Figure 35.98** shows the timer output when the output pattern is changed by software output control.

Immediately after the operation starts (TSG3nTRG0.TSG3nTS = 1), the output pattern of TSG3nSPC2 to TSG3nSPC0 and the pattern set with TSG3nIDC and TSG3nPSC (TSG3nOPT0.TSG3nPSS = 1) are output.

Setting pattern switch method

Setting TSG3nOPT0.TSG3nSTE to 1 and TSG3nPOT to 0 selects the pattern switch method. The TSG3nO1 to TSG3nO6 pin output pattern is changed at the change timing of the TSG3nPTSI2 to TSG3nPTSI0 pins.

The output pattern at the beginning of operation is set with TSG3nOPT0.TSG3nIDC and with TSG3nOPT0.TSG3nPSC. However, after determining the rotation direction (after the value is set to TSG3nSTR1.TSG3nTSF), the setting of TSG3nPSC is disabled.

Operation of pattern switch method

After level detection is performed for the pins (three inputs from the hall sensor), the level-detected signals are decoded. From the decoding result, the PWM output of TSG3nO1 to TSG3nO6 pins (PWM output defined by TSG3nCMP1E to TSG3nCMP12E value) is selected. To control the dead time, the dead time counter is activated at the falling timing of signals in each phase and the dead time is inserted.

The 18-bit counter is cleared by a match of the 18-bit counter value with the TSG3nCMP0E value or by a change of the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) while TSG3nS120DCO is 0.

In this method, the pattern, which is decoded by using information on input pattern (TSG3nPTSI2 to TSG3nPTSI0), the electric current direction control bit (TSG3nOPT0.TSG3nIDC), and TSG3nPTSI2 to TSG3nPTSI0 pattern order detection flag (TSG3nSTR1.TSG3nTSF), is output. **Figure 35.80** to **Figure 35.83** show the timer output when TSG3nPTSI2 to TSG3nPTSI0 pin inputs change.

Immediately after the operation starts (TSG3nTRG0.TSG3nTS = 1), the output pattern set by the levels input on the TSG3nPTSI2 to TSG3nPTSI0 pins and by the TSG3nIDC and TSG3nPSC bits is produced. If a level on any of the TSG3nPTSI2 to TSG3nPTSI0 pins is changed, TSG3nTSF is determined by the direction of the change to the order. After the TSG3nTSF value is determined, the pattern set by the TSG3nTSF bit replaces the pattern set by the TSG3nPSC bit.

CAUTION

When connecting the three-phase pulse input signal to the TSG3nPTSI2 to TSG3nPTSI0 pins, confirm that the three-phase pulse input value and the patterns output from the TSG3nO1 to TSG3nO6 pins satisfy the expected conditions.
If the expected conditions are not satisfied, change the connection between the three-phase pulse input signal and the TSG3nPTSI2 to TSG3nPTSI0 pins.

Setting trigger switch method

Setting TSG3nOPT0.TSG3nSTE and TSG3nPOT to 1 selects the trigger switch method. The output patterns of pins TSG3nO1 to TSG3nO6 are changed at a rising edge of an external input (TSG3nOPCI1 and TSG3nOPCI0 signals).

For pattern output order, see **Section 35.4.7.4, (5) Operation in 120-DC Mode**.

The initial output pattern can be controlled with TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0.

The initial pattern is output when the TSG3n operation starts (TSG3nTRG0.TSG3nTS = 1) after TSG3nSPC2 to TSG3nSPC0 are configured. For details, see **Section 35.4.7.4, (6) List of Output Patterns in 120-DC Mode**.

Operation of trigger switch method

With the trigger input switch method, the rising edges of the TSG3nOPCI0 and TSG3nOPCI1 signals are detected and the output switch timing is generated. The initial timer output pattern is set with TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0. The subsequent output patterns are switched after rising of the TSG3nOPCI0 and TSG3nOPCI1 signals is detected. Furthermore, the output patterns can be switched by setting TSG3nOPT1.TSG3nSPC2 to TSG3nOPT1.TSG3nSPC0.

The 18-bit counter counts based on the carrier period set in TSG3nCMP0E. The 18-bit counter is cleared by match of the 18-bit counter and TSG3nCMP0E, by a write access to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0, or by detecting a rising of TSG3nOPCI0 and TSG3nOPCI1 signals. (When TSG3nS120DCO = 0)

For examples of operation in 120-DC mode when trigger input switch method is used, see **Figure 35.80** to **Figure 35.83**.

CAUTION

The initial pattern should be set according to the read input level of the port to which TSG3nPTSI2 to TSG3nPTSI0 pins are connected.

(4) Timer Output in 120-DC Mode

In 120-DC mode, the PWM output is controlled with TSG3nPAT0W, TSG3nPAT1W, and TSG3nCMP1E to TSG3nCMP12E. TSG3nPAT0W, TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, and TSG3nCMP10E are set to control the output of TSG3nO1, TSG3nO3, and TSG3nO5 pins. TSG3nPAT1W, TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, and TSG3nCMP12E are set to control the output of TSG3nO2, TSG3nO4, and TSG3nO6 pins.

With PWM output control, eight types of output patterns can be selected for each of TSG3nO1, TSG3nO3, and TSG3nO5 pins and TSG3nO2, TSG3nO4, and TSG3nO6 pins.

Table 35.84 TSG3nPAT0W Set Value and Output Control

PATmT Value	Output Control
000	Fixed to low
001	PWM output set with TSG3nCMP1E
010	PWM output set with TSG3nCMP2E
011	PWM output set with TSG3nCMP5E
100	PWM output set with TSG3nCMP6E
101	PWM output set with TSG3nCMP9E
110	PWM output set with TSG3nCMP10E
111	Fixed to high

Note: m = 0, 1, 2, 3, 4, 5

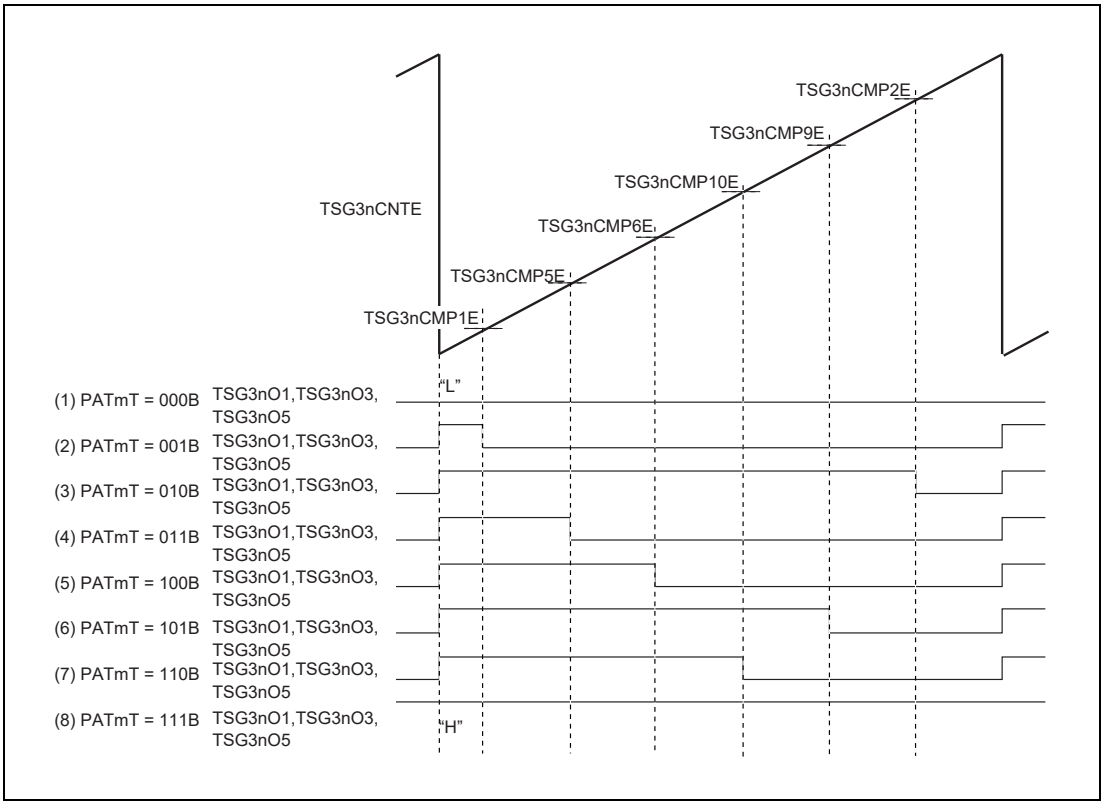


Figure 35.78 TSG3nO1, TSG3nO3, TSG3nO5 Pin Output of Each Output Pattern

Table 35.85 TSG3nPAT1W Set Value and Output Control

PATmB Value	Output Control
000	Fixed to low
001	PWM output set with TSG3nCMP3E
010	PWM output set with TSG3nCMP4E
011	PWM output set with TSG3nCMP7E
100	PWM output set with TSG3nCMP8E
101	PWM output set with TSG3nCMP11E
110	PWM output set with TSG3nCMP12E
111	Fixed to high

Note: m = 0, 1, 2, 3, 4, 5

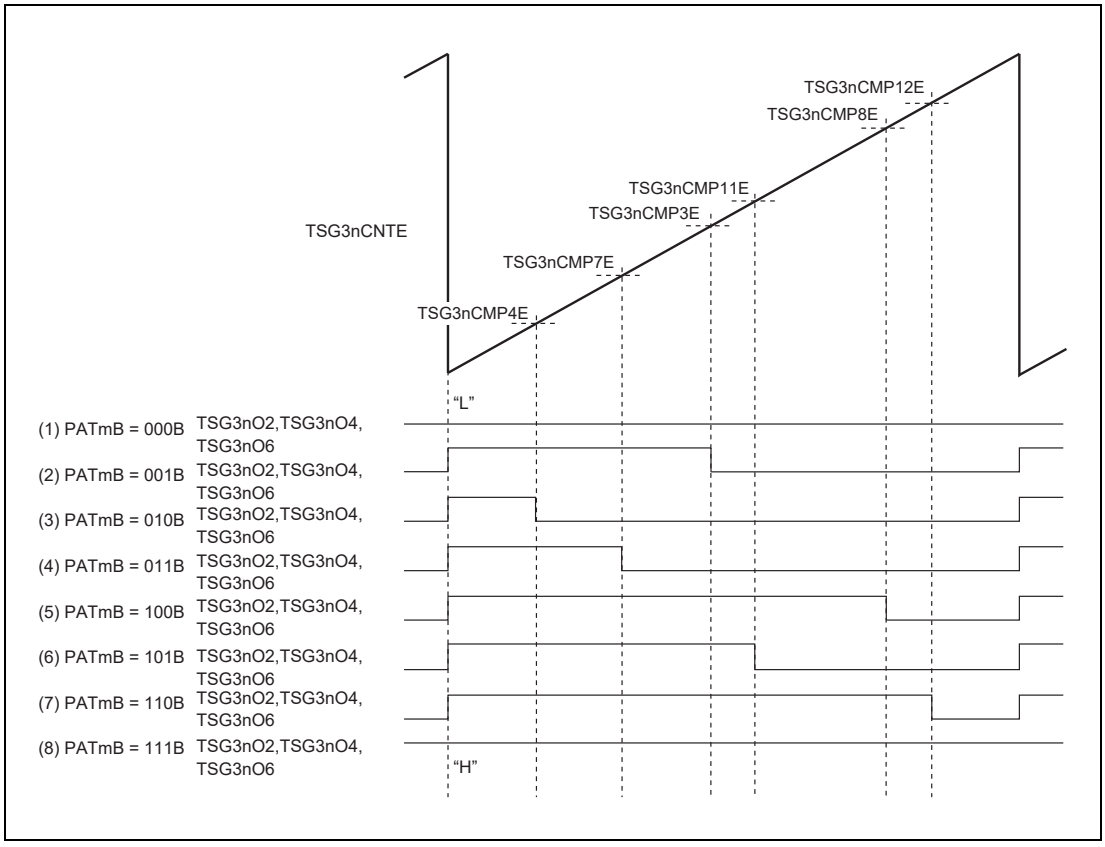


Figure 35.79 TSG3nO2, TSG3nO4, TSG3nO6 Pin Output of Each Output Pattern

(5) Operation in 120-DC Mode

Figure 35.80 to **Figure 35.83** show examples of operation in 120-DC mode.

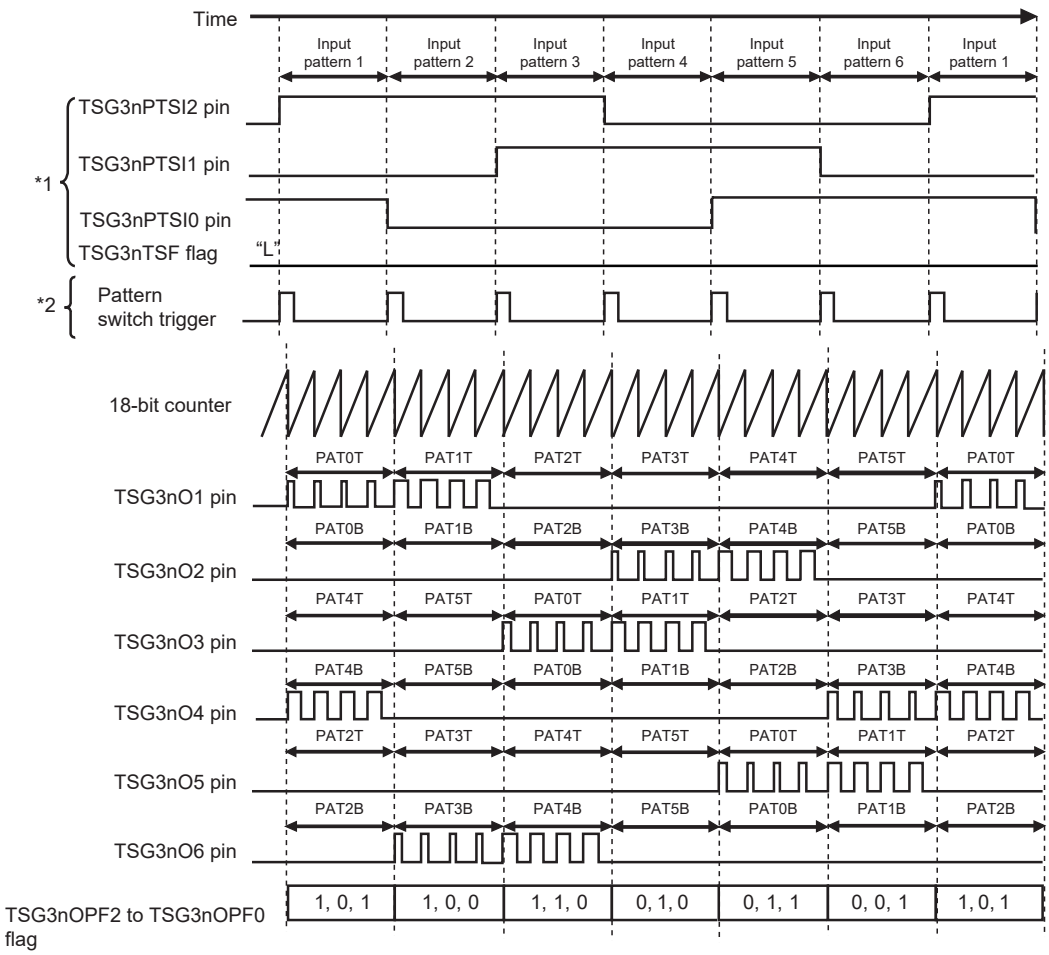
The TSG3nO1 to TSG3nO6 pins detect the input level change of the TSG3nPTSI2 to TSG3nPTSI0 pins, and then change the output pattern. The 18-bit counter produces sawtooth waveform and PWM output using TSG3nCMP0E to TSG3nCMP12E.

When TSG3nS120DCO = 0, the 18-bit counter is cleared to 00000_H each time the counter value matches with TSG3nCMP0E or a change in the TSG3nPTSI2 to TSG3nPTSI0 pins is detected. The timer output pattern is switched each time a change in the TSG3nPTSI2 to TSG3nPTSI0 pins is detected.

When TSG3nS120DCO = 1, the 18-bit counter is cleared to 00000_H when the counter value matches with TSG3nCMP0E but not cleared with a change in the TSG3nPTSI2 to TSG3nPTSI0 pins. The timer output pattern is switched to the one corresponding to the patterns of TSG3nPTSI2 to TSG3nPTSI0 pins at the next match timing of TSG3nCNTE and TSG3nCMP0E.

NOTE

PAT0T to PAT5T and PAT0B to PAT5B show PWM operation set by TSG3nCMP1E to TSG3nCMP12E, respectively.

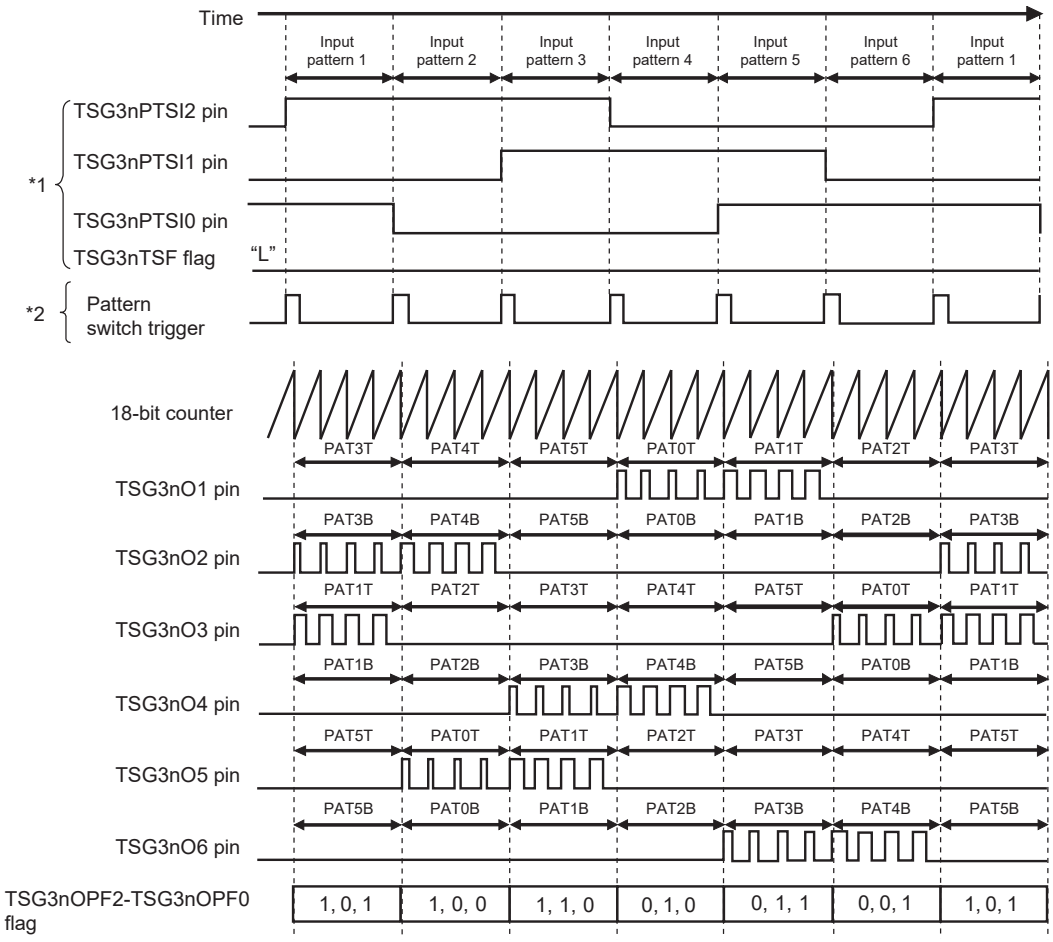


Note 1. For pattern switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 0)
Note 2. For trigger switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1)

Figure 35.80 Example of Operation in 120-DC Mode (Normal Rotation: TSG3nSTR1.TSG3nTSF = 0 and TSG3nOPT0.TSG3nIDC = 0)

NOTE

TSG3nOPT0.TSG3nSOC = 0

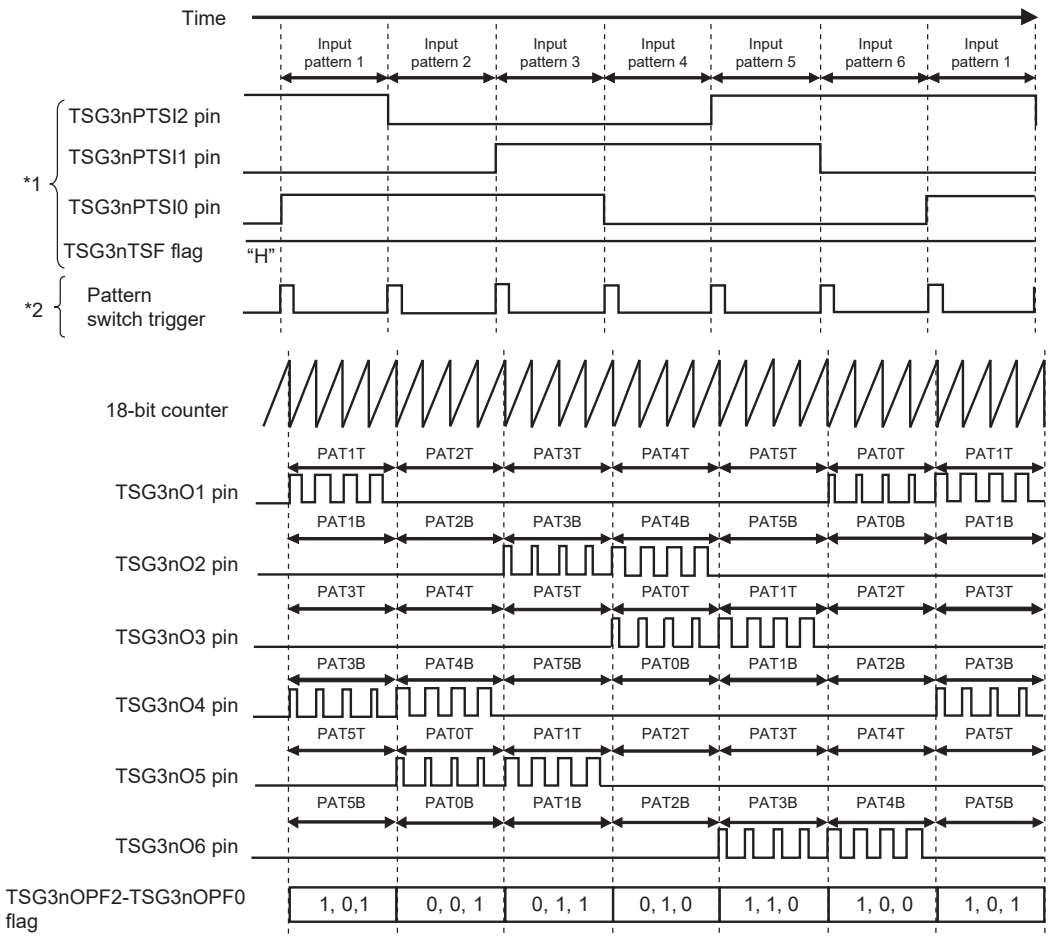


Note 1. For pattern switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 0)
Note 2. For trigger switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1)

Figure 35.81 Example of Operation in 120-DC Mode (Normal Rotation: TSG3nSTR1.TSG3nTSF = 0 and TSG3nOPT0.TSG3nIDC = 1)

NOTE

TSG3nOPT0.TSG3nSOC = 0



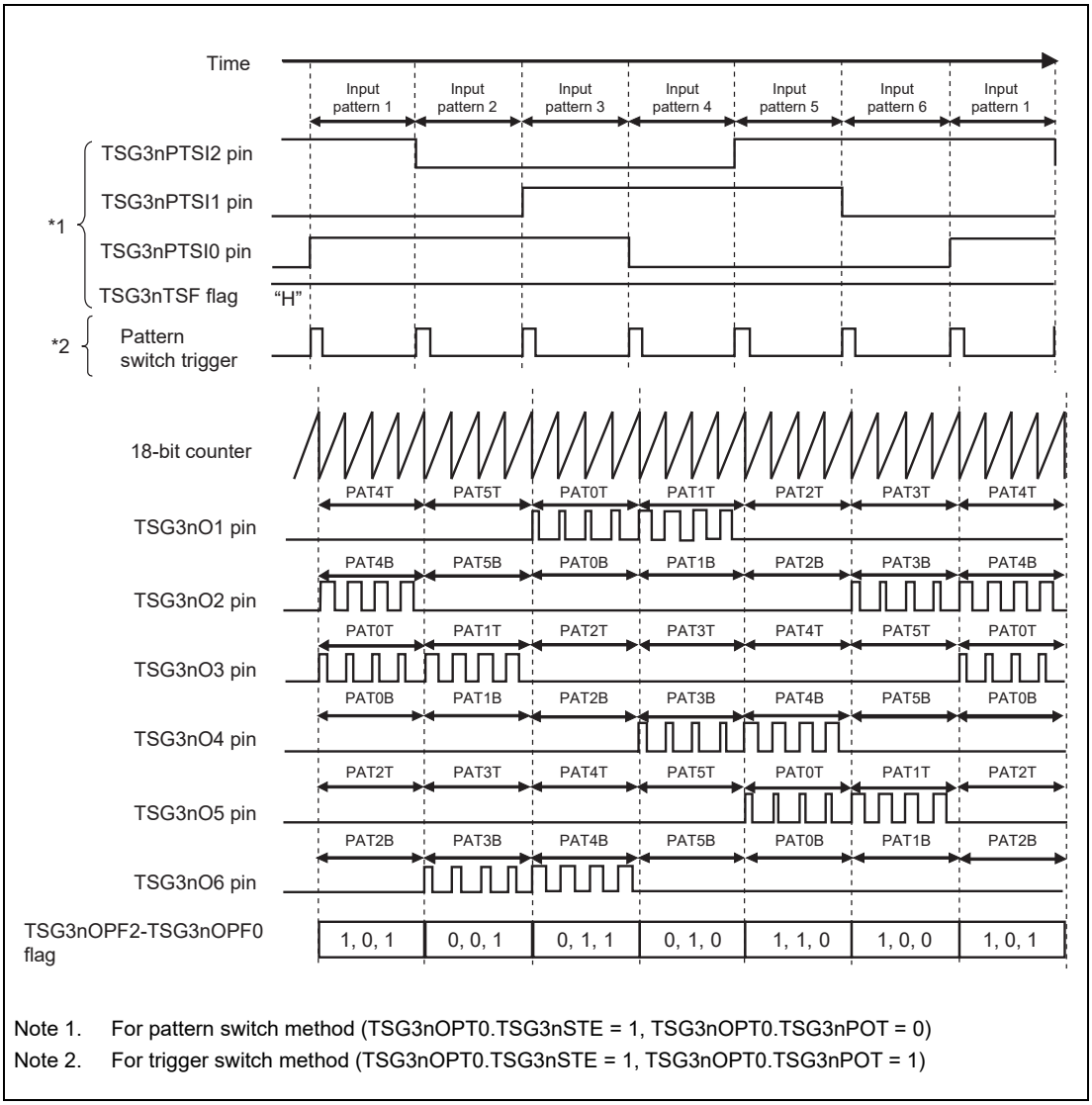
Note 1. For pattern switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 0)

Note 2. For trigger switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1)

Figure 35.82 Example of Operation in 120-DC Mode (Reverse Rotation: TSG3nSTR1.TSG3nTSF = 1 and TSG3nOPT0.TSG3nIDC = 0)

NOTE

TSG3nOPT0.TSG3nSOC = 0



Note 1. For pattern switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 0)
Note 2. For trigger switch method (TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1)

Figure 35.83 Example of Operation in 120-DC Mode (Reverse Rotation: TSG3nSTR1.TSG3nTSF = 1 and TSG3nOPT0.TSG3nIDC = 1)

NOTE

TSG3nOPT0.TSG3nSOC = 0

(6) List of Output Patterns in 120-DC Mode

In 120-DC mode, the output pattern is determined according to the electric current direction (TSG3nOPT0.TSG3nIDC) and the pattern order direction.

Table 35.86 Selection of Pattern Order Direction in 120-DC Mode

TSG3nOPT0			Pattern Order Direction
TSG3nSTE	TSG3nPOT	TSG3nPSS	
0	—	1	TSG3nPSC
1	0	—	TSG3nTSF (TSG3nPSC only for initial setting at operation start)
1	1	1	TSG3nPSC

NOTE

Set {TSG3nPOT,TSG3nPSS} = {1,0} is prohibited.

Table 35.87 List of Output Patterns in 120-DC Mode (1/4)

Electric current direction: normal (TSG3nIDC = 0)
Pattern order direction: normal (TSG3nTSF = 0 or TSG3nPSC = 0)

	TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output pins	101	100	110	010	011	001	000	111
TSG3nO1	PAT0T	PAT1T	PAT2T	PAT3T	PAT4T	PAT5T	Low	Low
TSG3nO2	PAT0B	PAT1B	PAT2B	PAT3B	PAT4B	PAT5B	Low	Low
TSG3nO3	PAT4T	PAT5T	PAT0T	PAT1T	PAT2T	PAT3T	Low	Low
TSG3nO4	PAT4B	PAT5B	PAT0B	PAT1B	PAT2B	PAT3B	Low	Low
TSG3nO5	PAT2T	PAT3T	PAT4T	PAT5T	PAT0T	PAT1T	Low	Low
TSG3nO6	PAT2B	PAT3B	PAT4B	PAT5B	PAT0B	PAT1B	Low	Low

Table 35.87 List of Output Patterns in 120-DC Mode (2/4)

Electric current direction: reverse (TSG3nIDC = 1)
Pattern order direction: normal (TSG3nTSF = 0 or TSG3nPSC = 0)

	TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output pins	101	100	110	010	011	001	000	111
TSG3nO1	PAT3T	PAT4T	PAT5T	PAT0T	PAT1T	PAT2T	Low	Low
TSG3nO2	PAT3B	PAT4B	PAT5B	PAT0B	PAT1B	PAT2B	Low	Low
TSG3nO3	PAT1T	PAT2T	PAT3T	PAT4T	PAT5T	PAT0T	Low	Low
TSG3nO4	PAT1B	PAT2B	PAT3B	PAT4B	PAT5B	PAT0B	Low	Low
TSG3nO5	PAT5T	PAT0T	PAT1T	PAT2T	PAT3T	PAT4T	Low	Low
TSG3nO6	PAT5B	PAT0B	PAT1B	PAT2B	PAT3B	PAT4B	Low	Low

Note 1. When TSG3nSPC2 to TSG3nSPC0 are written while the values of TSG3nSTE and TSG3nPOT are 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated on rising of the TSG3nOPCI0 and TSG3nOPCI1 signals, the output is switched according to the order of pattern switching. When TSG3nS120DCO = 0, the output is switched immediately. When TSG3nS120DCO = 1, the output is switched when the main counter (TSG3nCNTE) matches with TSG3nCMP0E (from the next timer cycle). TSG3nSPC2 to TSG3nSPC0 remain unchanged even if the output pattern is switched.

NOTES

1. PAT0T to PAT5T: PWM output set by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E
2. PAT0B to PAT5B: PWM output set by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E

Table 35.87 List of Output Patterns in 120-DC Mode (3/4)

Electric current direction: normal (TSG3nIDC = 0)
Pattern order direction: reverse (TSG3nTSF = 1 or TSG3nPSC = 1)

	TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output pins	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG3nO1	PAT1T	PAT0T	PAT5T	PAT4T	PAT3T	PAT2T	Low	Low
TSG3nO2	PAT1B	PAT0B	PAT5B	PAT4B	PAT3B	PAT2B	Low	Low
TSG3nO3	PAT3T	PAT2T	PAT1T	PAT0T	PAT5T	PAT4T	Low	Low
TSG3nO4	PAT3B	PAT2B	PAT1B	PAT0B	PAT5B	PAT4B	Low	Low
TSG3nO5	PAT5T	PAT4T	PAT3T	PAT2T	PAT1T	PAT0T	Low	Low
TSG3nO6	PAT5B	PAT4B	PAT3B	PAT2B	PAT1B	PAT0B	Low	Low

Table 35.87 List of Output Patterns in 120-DC Mode (4/4)

Electric current direction: reverse (TSG3nIDC = 1)
Pattern order direction: reverse (TSG3nTSF = 1 or TSG3nPSC = 1)

	TSG3nOPT1.TSG3nSPC2-TSG3nSPC0*1/TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output pins	**101**	**100**	**110**	**010**	**011**	**001**	**000**	**111**
TSG3nO1	PAT4T	PAT3T	PAT2T	PAT1T	PAT0T	PAT5T	Low	Low
TSG3nO2	PAT4B	PAT3B	PAT2B	PAT1B	PAT0B	PAT5B	Low	Low
TSG3nO3	PAT0T	PAT5T	PAT4T	PAT3T	PAT2T	PAT1T	Low	Low
TSG3nO4	PAT0B	PAT5B	PAT4B	PAT3B	PAT2B	PAT1B	Low	Low
TSG3nO5	PAT2T	PAT1T	PAT0T	PAT5T	PAT4T	PAT3T	Low	Low
TSG3nO6	PAT2B	PAT1B	PAT0B	PAT5B	PAT4B	PAT3B	Low	Low

Note 1. When TSG3nSPC2 to TSG3nSPC0 are written while the values of TSG3nSTE and TSG3nPOT are 1, the output pattern changes. Thereafter, when a pattern switch trigger is generated on rising of the TSG3nOPCI0 and TSG3nOPCI1 signals, the output is switched according to the order of pattern switching. When TSG3nS120DCO = 0, the output is switched immediately.
When TSG3nS120DCO = 1, the output is switched when the main counter (TSG3nCNTE) matches with TSG3nCMP0E (from the next timer cycle). TSG3nSPC2 to TSG3nSPC0 remain unchanged even if the output pattern is switched.

NOTES

1. PAT0T to PAT5T: PWM output set by TSG3nCMP1E, TSG3nCMP2E, TSG3nCMP5E, TSG3nCMP6E, TSG3nCMP9E, TSG3nCMP10E
2. PAT0B to PAT5B: PWM output set by TSG3nCMP3E, TSG3nCMP4E, TSG3nCMP7E, TSG3nCMP8E, TSG3nCMP11E, TSG3nCMP12E

(7) Operation Start Timing in 120-DC Mode

In 120-DC mode, when trigger switch control (TSG3nOPT0.TSG3nSTE=1, TSG3nOPT0.TSG3nPOT = 1) is used, pattern set with TSG3nOPT1.TSG3nSPC2 to TSG3nOPT1.TSG3nSPC0, TSG3nOPT0.TSG3nPSC, and TSG3nOPT0.TSG3nIDC can be output. However, when pattern switch control (TSG3nOPT0.TSG3nSTE=1, TSG3nOPT0.TSG3nPOT = 0) is used, the pattern of the TSG3nPTSI2 to TSG3nPTSI0 pins can be detected but the pattern order direction (TSG3nSTR1.TSG3nTSF) cannot be determined. Therefore, set the pattern order direction in TSG3nPSC when TSG3nTE is 0. The TSG3nPSC set value is loaded to TSG3nTSF, and the value can be used for the initial pattern setting.

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 0, TSG3nPOT = 0, TSG3nIDC = 0, TSG3nSTE = 1

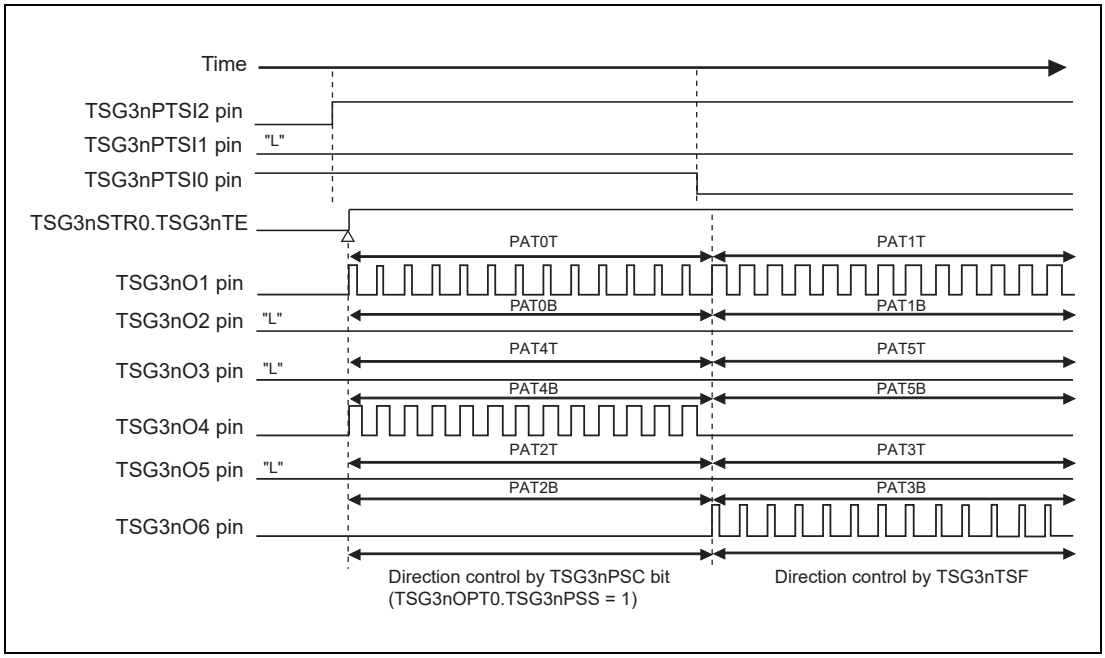


Figure 35.84 Control when Timer Output Starts in Normal Rotation (when Normal Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 1, TSG3nPOT = 0, TSG3nIDC = 1, TSG3nSTE = 1

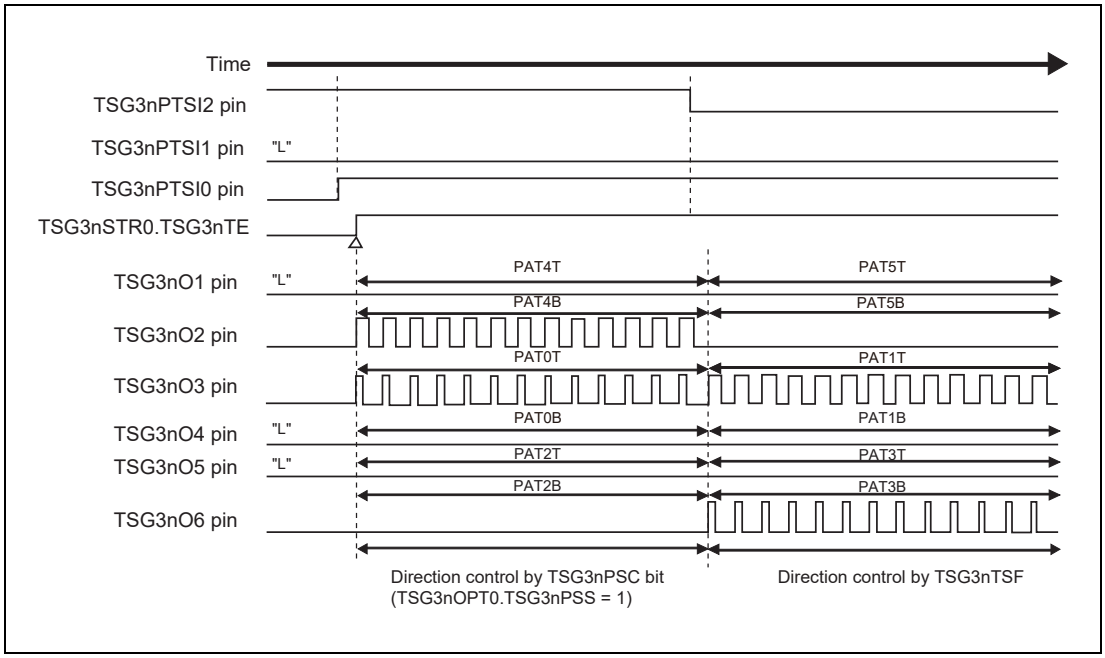


Figure 35.85 Control when Timer Output Starts in Reverse Rotation (when Normal Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 0, TSG3nPOT = 0, TSG3nIDC = 0, TSG3nSTE = 1

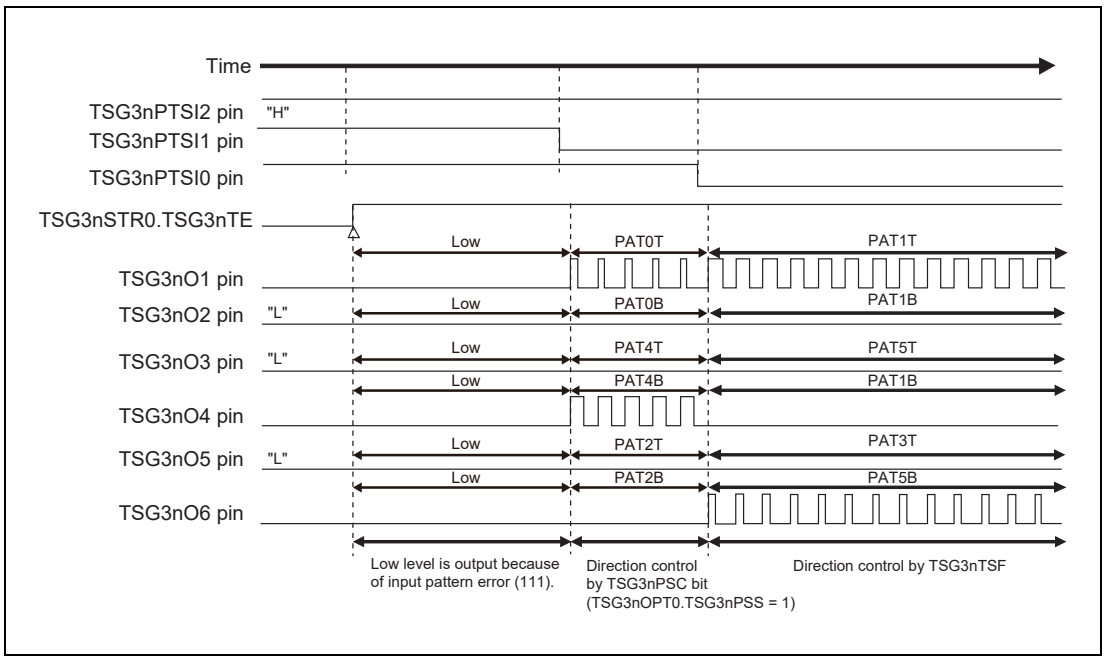


Figure 35.86 Control when Timer Output Starts in Normal Rotation (when Error Pattern is Input)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nPSC = 1, TSG3nPOT = 0, TSG3nIDC = 1, TSG3nSTE = 1

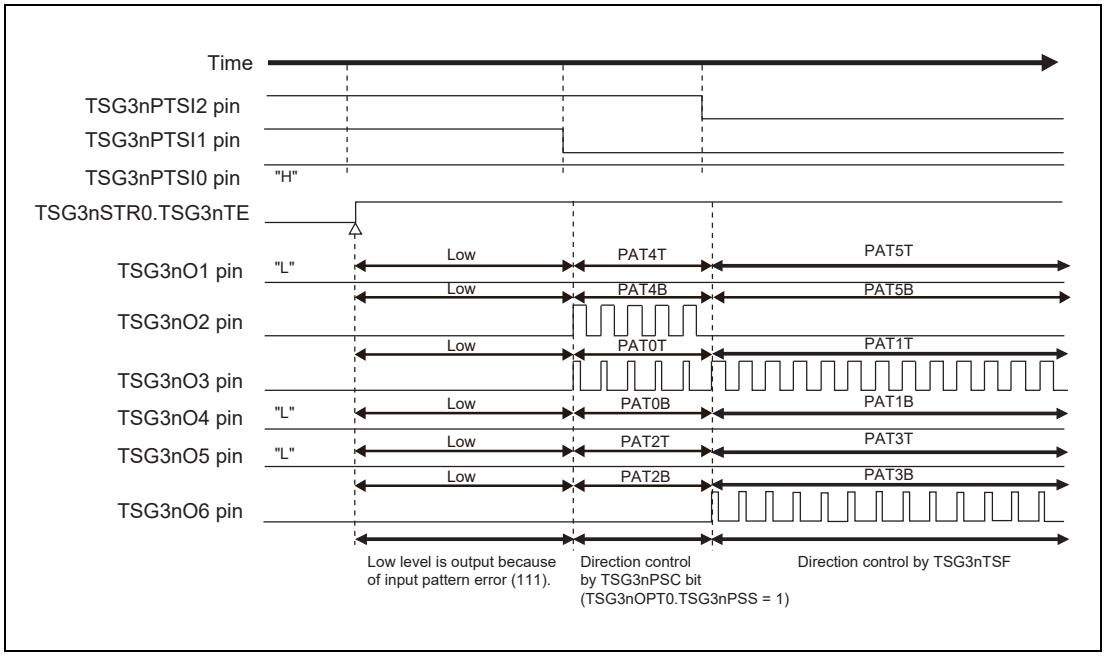


Figure 35.87 Control when Timer Output Starts in Reverse Rotation (when Error Pattern is Input)

(8) Output Switch Timing in 120-DC Mode (TSG3nS120DCO = 0)

As shown in **Figure 35.88** to **Figure 35.91**, in 120-DC mode, the external switch timing for output pattern (TSG3nOPCI0 and TSG3nOPCI1 signals, and TSG3nPTSI2 to TSG3nPTSI0 pins) is input irrespective of the 18-bit counter operation. When TSG3nS120DCO is 0, the output is switched to the new pattern by clearing the 18-bit counter using the pattern switch timing signal that is externally input.

In the pattern switch method, if a change in TSG3nPTSI2 to TSG3nPTSI0 pins occurs several times within one period, the output pattern is switched by clearing the 18-bit counter at each change. In the trigger switch method, if TSG3nOPCI0 and TSG3nOPCI1 signal trigger is input for several times within one period, the output pattern is switched by clearing the 18-bit counter each time the trigger is accepted.

If TSG3nSPC2 to TSG3nSPC0 are rewritten several times within one period, the output pattern is switched by clearing the 18-bit counter each time TSG3nSPC2 to TSG3nSPC0 are rewritten.

In case of a conflict between a rewrite to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 and TSG3nOPCI0 and TSG3nOPCI1 trigger, rewriting of TSG3nSPC2 to TSG3nSPC0 takes precedence.

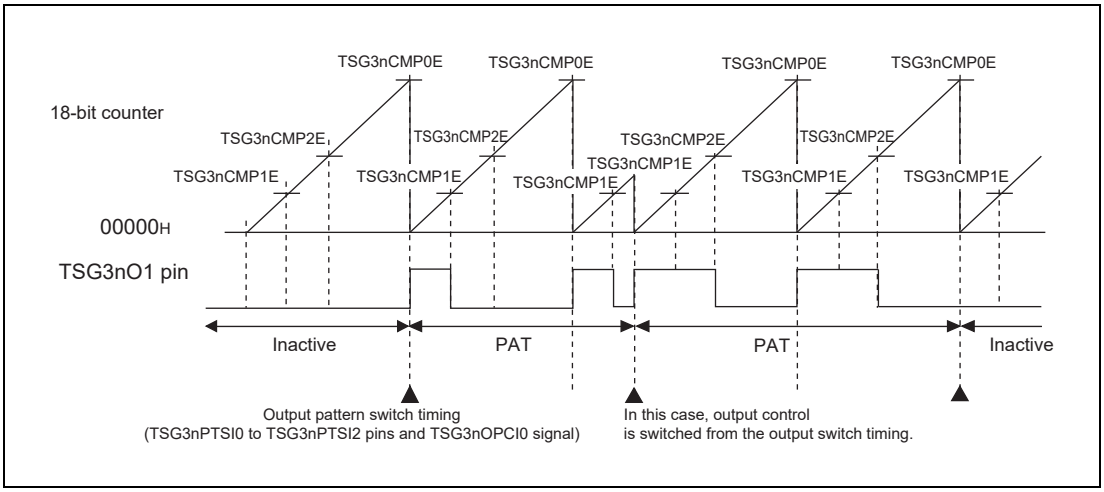
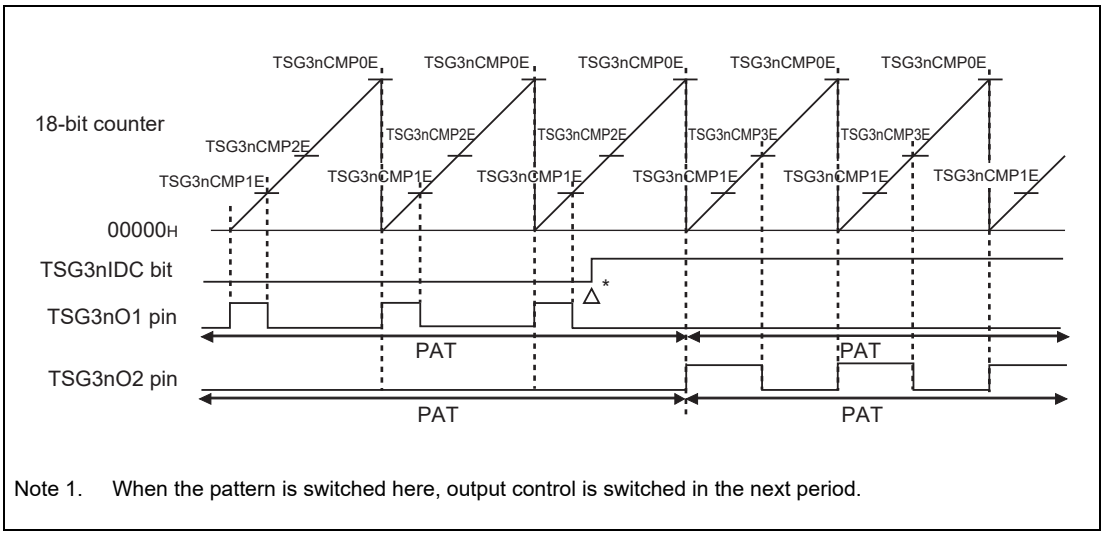


Figure 35.88 Output Switch Example (TSG3nPTSI2 to TSG3nPTSI0 Pins and TSG3nOPCI0 and TSG3nOPCI1 Signal Trigger Input)



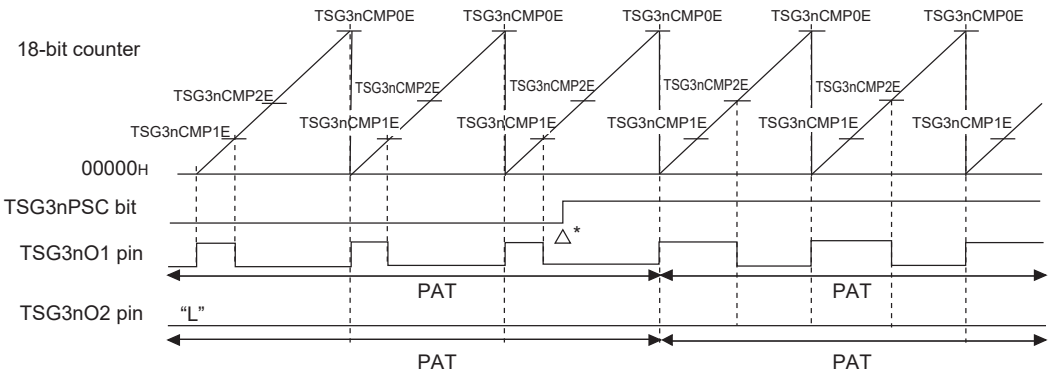
Note 1. When the pattern is switched here, output control is switched in the next period.

Figure 35.89 Output Switch Example (Switched by TSG3nOPT0.TSG3nIDC)

NOTE

If a change in the TSG3nPTSI2-0 pins occurs by the time next period when output control is switched by the TSG3nIDC bit, the 18-bit counter is cleared and output control is switched.

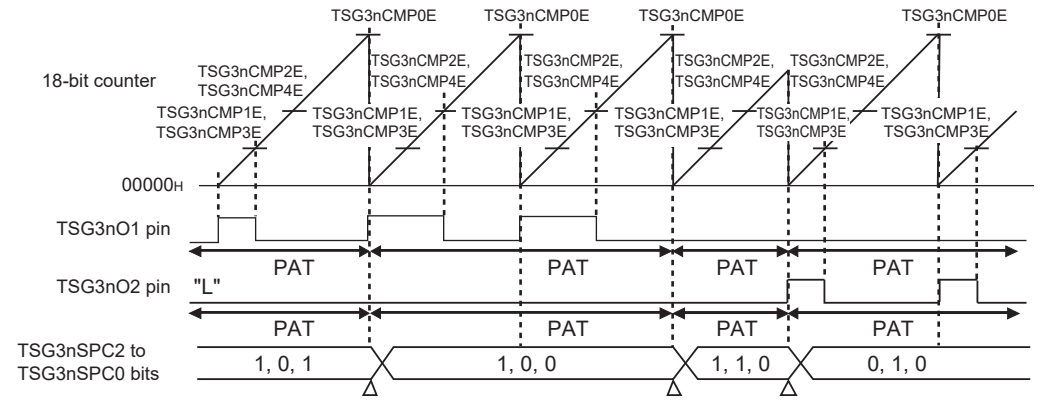
- TSG3nOPT0.TSG3nSTE =1, TSG3nOPT0.TSG3nPOT = 1



Note 1. When the pattern is switched here, output control is switched from the next period.

Figure 35.90 Output Switch Example (Switched by TSG3nOPT0.TSG3nPSC)

- TSG3nOPT0.TSG3nSOC = 0, TSG3nOPT0.TSG3nSTE =1, TSG3nOPT0.TSG3nPOT = 1



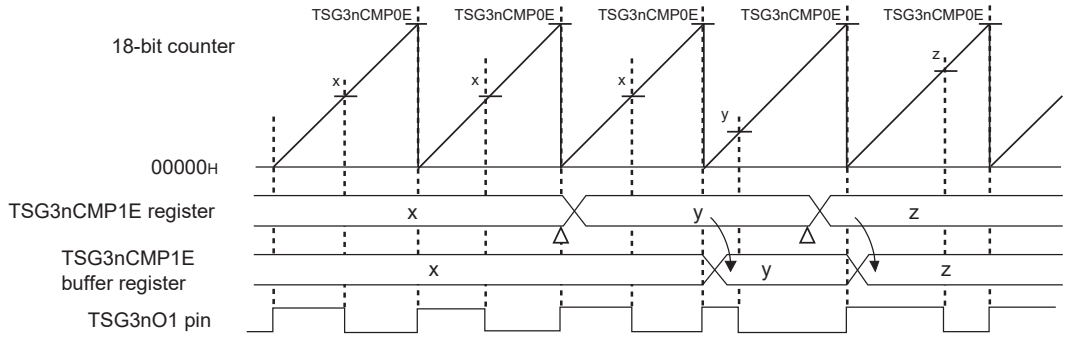
Note: Δ: Write access

Figure 35.91 Output Switch Example (Switched by TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0)

(9) Compare Register Rewrite Timing in 120-DC Mode

Example of operation when TSG3nCMP1E is reloaded (rewritten simultaneously) is shown below.

Figure 35.92 shows an output example when TSG3nCMP1E is rewritten. After TSG3nCMP1E is changed, data is not transferred to the TSG3nCMP1E buffer register (changed data does not become valid) until the next reload timing; therefore, the specified output waveform can be obtained. However, do not write to TSG3nCMP1E again while the reload is suspended (period from when TSG3nCMP1E is changed to when simultaneous rewrite is executed). Be sure to read the reload request flag (TSG3nRSF) to confirm that the flag is 0 before writing to TSG3nCMP1E.



Note: Δ: Write access

Figure 35.92 Output Example when TSG3nCMP1E is Rewritten

(10) Dead Time Control in 120-DC Mode

In 120-DC mode, the dead time is controlled on falling of each phase, and the dead time is added.

The dead time set in TSG3nDTC1W is inserted on falling of the positive phase, and the dead time set in TSG3nDTC0W is inserted on falling of the inverse phase.

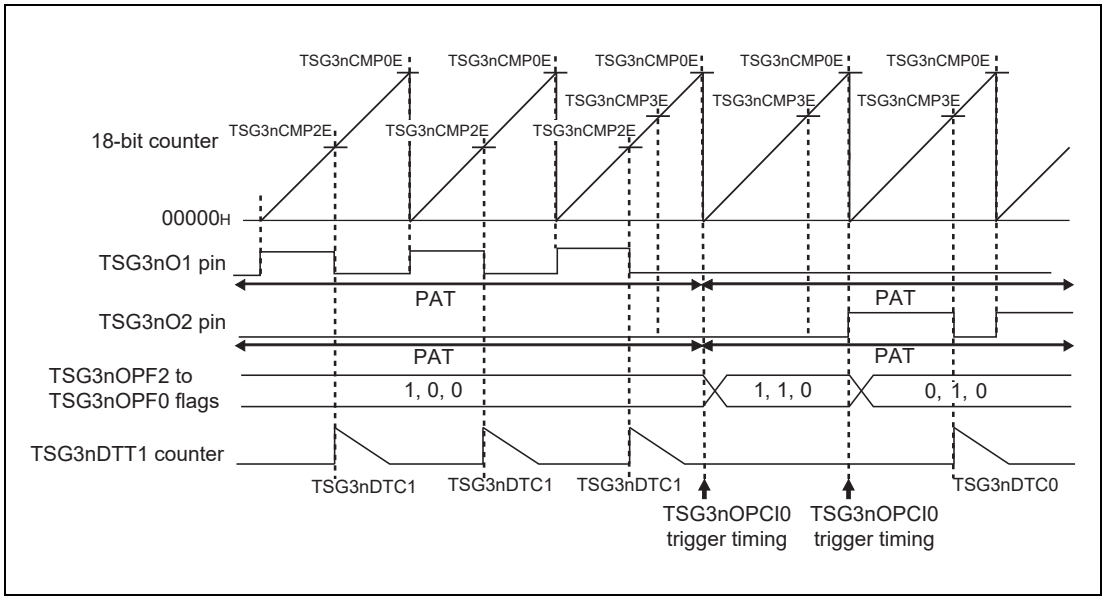


Figure 35.93 Output Switch Example

CAUTION

The dead time control method may affect the timer output. The timer output may not have the specified active level width due to the dead time control under the following conditions:

- **When noise is generated on the input pattern in the pattern switch method**
- **When a change in the input pattern occurs earlier than the PWM period in the pattern switch method**
- **When TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are changed and the output pattern is forcibly changed in the trigger switch method**
- **When switch method is changed**
- **When the current direction control bit (TSG3nOPT0.TSG3nIDC) is changed**
- **When the software output control function is used**

(11) Output Switch in 120-DC Mode

In 120-DC mode, the output pattern can be controlled by writing values to TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 when the trigger switch method (TSG3nOPT0.TSG3nSTE = 1, and TSG3nPOT = 1) is used. The dead time is secured by hardware at the switch timing.

CAUTION

When 111_B or 000_B is written to TSG3nSPC2 to TSG3nSPC0, the TSG3nO1 to TSG3nO6 pins are driven low.

(12) Operation when Noise is Generated in TSG3nPTSI2 to TSG3nPTSI0 Pins in 120-DC Mode

Input to the TSG3nPTSI2 to TSG3nPTSI0 pins is assumed to be the hall sensor signals of the brushless DC motor. Depending on the system, a noise may be generated on the TSG3nPTSI2 to TSG3nPTSI0 pins. Operation when a noise is generated is described below.

For system product design, be sure to insert a noise filter circuit between the hall sensor and the TSG3nPTSI2 to TSG3nPTSI0 pins.

Figure 35.95 shows a case when a noise is generated on the TSG3nPTSI2 to TSG3nPTSI0 pins during operation with the pattern switch method used.

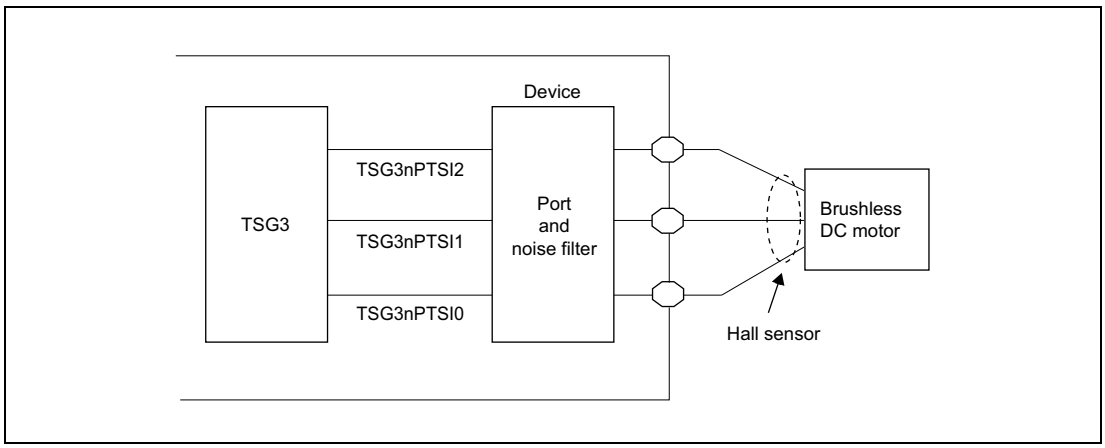


Figure 35.94 Example of Noise Filter Circuit Connection

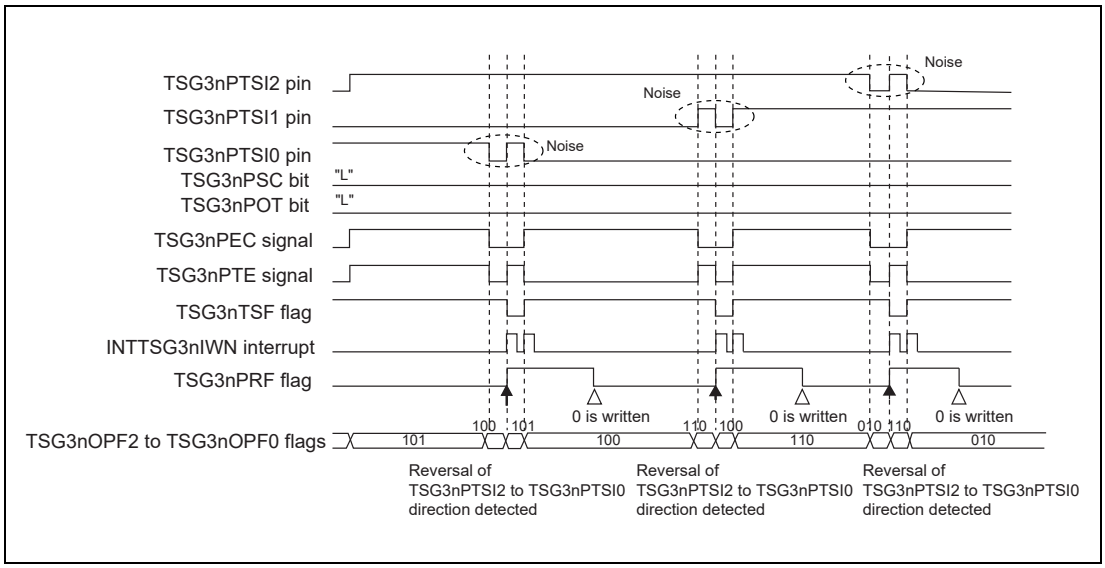


Figure 35.95 Example of Noise Generation at Level Change in TSG3nPTSI2 to TSG3nPTSI0 Pins (Pattern Switch Method)

(a) Change Timing of Input Pattern Change Detection Signal (TSG3nPTE)

The TSG3nPTE signal toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes.

CAUTION

Be sure to specify the pattern order direction with the TSG3nPSC bit in the TSG3nOPT0 register (when the TSG3nPSS bit in the TSG3nOPT0 register is 1).

When TSG3nPSC = 0:

Table 35.88 TSG3nPTE Toggle Operation when TSG3nPSC is 0

		TSG3nPTSI2-TSG3nPTSI0 Pins after change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	Toggle	—	—	—	—
	100	—	—	—	—	Toggle	—	—	—
	110	—	—	—	—	—	Toggle	—	—
	010	—	—	—	—	—	—	Toggle	—
	011	—	—	—	—	—	—	—	Toggle
	001	—	—	Toggle	—	—	—	—	—

When TSG3nPSC = 1:

Table 35.89 TSG3nPTE Toggle Operation when TSG3nPSC is 1

		TSG3nPTSI2-TSG3nPTSI0Pins after change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	—	—	—	—	Toggle
	100	—	—	Toggle	—	—	—	—	—
	110	—	—	—	Toggle	—	—	—	—
	010	—	—	—	—	Toggle	—	—	—
	011	—	—	—	—	—	Toggle	—	—
	001	—	—	—	—	—	—	Toggle	—

(b) Change Timing of Three-Phase Encode Signal (TSG3nPEC)

The TSG3nPEC signal toggles when input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes

Table 35.90 TSG3nPEC Toggle Operation

		TSG3nPTSI2-TSG3nPTSI0after change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 Pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	Toggle	—	—	—	Toggle
	100	—	—	Toggle	—	Toggle	—	—	—
	110	—	—	—	Toggle	—	Toggle	—	—
	010	—	—	—	—	Toggle	—	Toggle	—
	011	—	—	—	—	—	Toggle	—	Toggle
	001	—	—	Toggle	—	—	—	Toggle	—

(c) Change Timing of TSG3nO1 to TSG3nO6 Pins

- When the pattern switch method is used, the output pattern changes when the input signal of the TSG3nPTSI2 to TSG3nPTSI0 pins*1 changes. The output is also switched when two or more pins change simultaneously.
- When the trigger switch method is used, the output pattern changes at the rising edge of the TSG3nOPCI0 and TSG3nOPCI1 signals. The output also changes when data is written to TSG3nSPC2 to TSG3nSPC0*1 in TSG3nOPT0.

Note 1. When the input pattern changes to 000 or 111, the TSG3nO1-TSG3nO6 pins are driven low. The output pattern of TSG3nO1-TSG3nO6 changes immediately only when TSG3nS120DCO = 0. When TSG3nS120DCO = 1, the output pattern is switched when the main counter (TSG3nCNTE) matches with TSG3nCMP0E (from the next timer cycle).

(d) Change Timing of TSG3nTSF Flag

The TSG3nTSF flag toggles when the input pattern (TSG3nPTSI2 to TSG3nPTSI0 pins) changes

Table 35.91 Setting and Clearing of TSG3nTSF

		TSG3nPTSI2-TSG3nPTSI0 Pins after change							
		000	111	101	100	110	010	011	001
Current TSG3nPTSI2 to TSG3nPTSI0 Pins	000	—	—	—	—	—	—	—	—
	111	—	—	—	—	—	—	—	—
	101	—	—	—	0	—	—	—	1
	100	—	—	1	—	0	—	—	—
	110	—	—	—	1	—	0	—	—
	010	—	—	—	—	1	—	0	—
	011	—	—	—	—	—	1	—	0
	001	—	—	0	—	—	—	1	—

(e) Set Timing of TSG3nNDF Flag

This flag is set when two or more pins of the TSG3nPTSI2 to TSG3nPTSI0 pins change simultaneously, and cleared when 1 is written to the TSG3nNDR bit. The TSG3nNDF flag is valid when 1 is set to the TSG3nNDC bit.

(f) Set Timing of TSG3nPRF Flag

This flag is set when the TSG3nTSF flag changes, and cleared when 1 is written to the TSG3nPRR bit. The TSG3nPRF flag is valid when 1 is set to the TSG3nPRC bit.

(g) Set Timing of TSG3nPEF Flag

This flag is set when 000 or 111 is input to the TSG3nPTSI2 to TSG3nPTSI0 pins, and cleared when 1 is written to the TSG3nPER bit. The TSG3nPEF flag is valid when the TSG3nPEC bit is set to 1.

(13) Basic Control Flow in 120-DC Mode

In 120-DC mode, there are eight control states as listed in **Table 35.92**.

When TSG3nOPT0.TSG3nSTE = 1 and TSG3nPOT = 0, the pattern switch method is used for 120-DC control. This is defined as fixed phase control. The fixed phase control should be performed considering the factors such as delay from the hall sensor and delay from sensor level detection to timer output. However, acceleration or deceleration can be performed simply by changing the PWM duty.

When TSG3nOPT0.TSG3nSTE = 1 and TSG3nPOT = 1, the trigger switch method is selected for 120-DC control. This is defined as variable phase control. With the variable phase control, the timer output pattern is set prior to the hall sensor; therefore, acceleration or deceleration control according to the phase difference can be performed. However, control is more complex than the fixed phase control because offset width and predicted value with respect to the hall sensor should be considered. For details, **Section 41.2.3.10, Three-Phase Pulse Input Control Function**.

When TSG3nOPT0.TSG3nSTE = 1, TSG3nOPT0.TSG3nPOT = 1, and TSG3nPSS = 1, the pattern order direction of the motor can be set with the TSG3nPSC bit in the TSG3nOPT0 register. Set TSG3nPSC to 0 for normal rotation and set TSG3nPSC to 1 for reverse rotation.

The TSG3nIDC bit in the TSG3nOPT0 register sets the electric current direction. If the same value as the rotation direction of the motor (TSG3nPSC set value) is set, acceleration control is set. If the different value from the rotation direction of the motor is set, deceleration control is set.

Table 35.92 Timer Control Status

Status	TSG3nPSC in TSG3nOPT0	TSG3nTSF in TSG3nSTR1	TSG3nIDC in TSG3nOPT0	TSG3nPOT in TSG3nOPT0	Control
A	—	0	0	0	Normal rotation, acceleration, and fixed phase
B	0	—	0	1	Normal rotation, acceleration, and variable phase
C	0	—	1	1	Normal rotation, deceleration, and variable phase
D	—	0	1	0	Normal rotation, deceleration, and fixed phase
E	—	1	1	0	Reverse rotation, acceleration, and fixed phase
F	1	—	1	1	Reverse rotation, acceleration, and variable phase
G	1	—	0	1	Reverse rotation, deceleration, and variable phase
H	—	1	0	0	Reverse rotation, deceleration, and fixed phase

Generally, the state when the motor rotation stops is assumed to be the starting state and the control begins. First the fixed phase control is used to rotate the motor from the stopped state. Then, to accelerate the motor speed to the fast rotation, the variable phase control is switched on. Used in combination with encoder timer (ENCA), the variable phase control changes the timer output to a timing that is earlier than the change point (leading) of the hall sensors.

To decelerate the motor speed from fast rotation, the direction is switched to deceleration control by rewriting only TSG3nIDC in TSG3nOPT0. When the rotation count is reduced to low-speed rotation, the motor can be transitioned to the stopped state by decreasing the PWM duty.

State transition is shown in **Figure 35.96** and **Figure 35.97**.

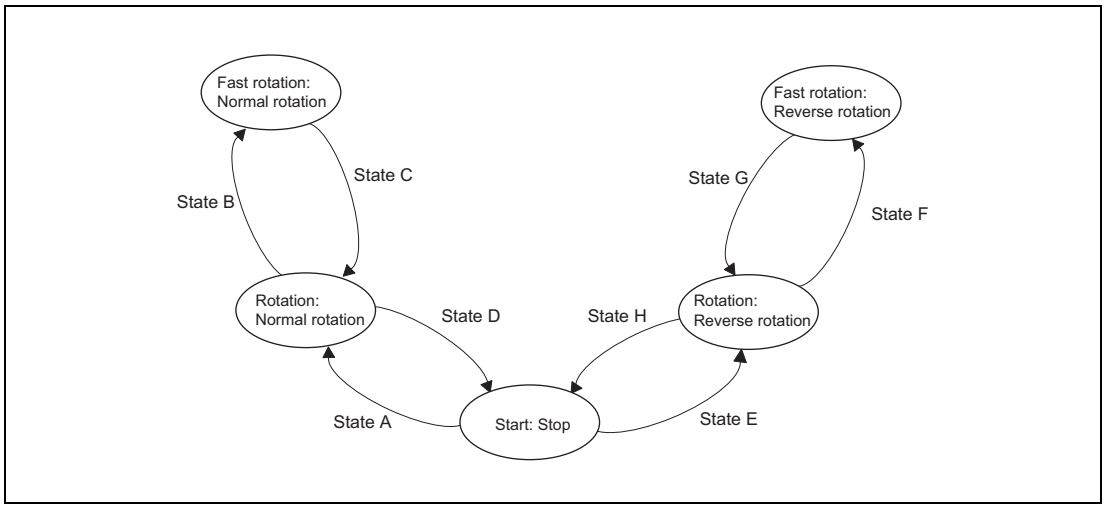


Figure 35.96 State Transition Diagram

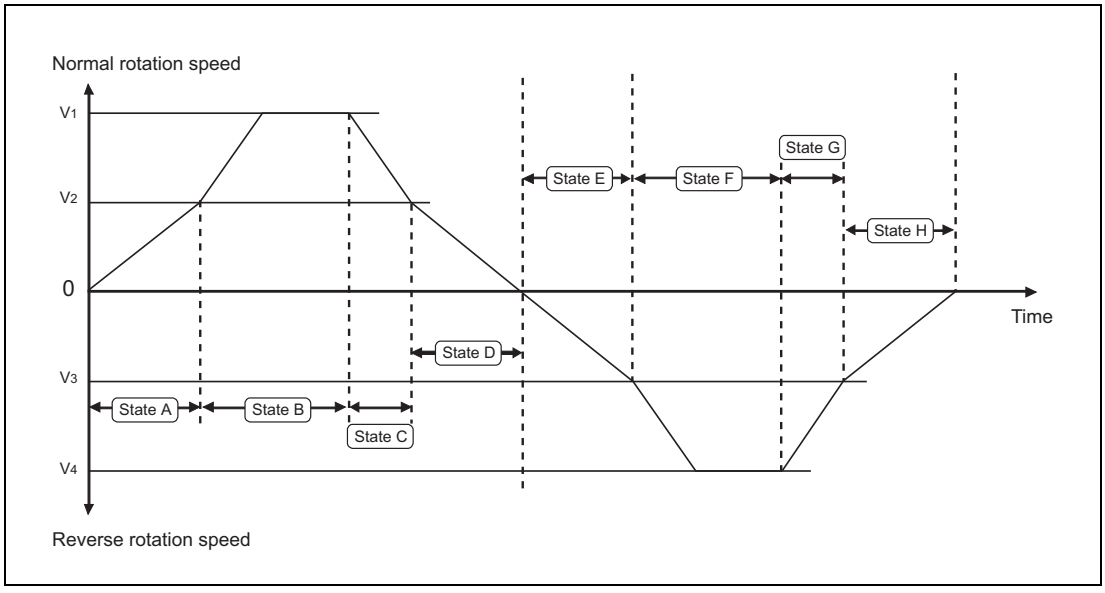


Figure 35.97 Relationship between State Transition and Rotation Speed of Motor

NOTE

V1 and V4: Fast rotation speed of normal rotation and reverse rotation

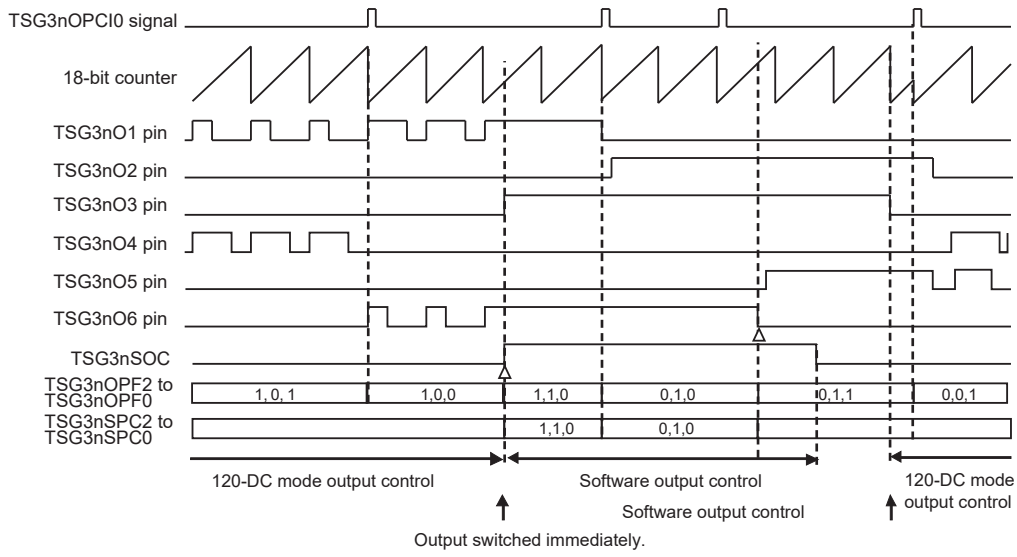
V2 and V3: Low rotation speed of normal rotation and reverse rotation

(14) Software Output Control Function in 120-DC Mode

TSG3nOPT0.TSG3nSOC and TSG3nIDC, and TSG3nOPT1.TSG3nSPC2 to TSG3nSPC0 are used in 120-DC mode for timer output control by software.

As shown in **Figure 35.98**, the output control is switched immediately when TSG3nSOC is set to 1. If the dead time is set, the period of the dead time period is guaranteed. After that, to switch the output control from software output control to 120-DC control, set TSG3nSOC to 0. At this timing, output control is retained and at the reload timing, output control is switched to 120-DC mode.

For details on software output control function, see **Section 35.4.7.8, Software Output Control Function**.



Note: Δ: Write access

Figure 35.98 Example of Switching from 120-DC Mode to Software Output Control Function

(a) Procedure for Software Output Control

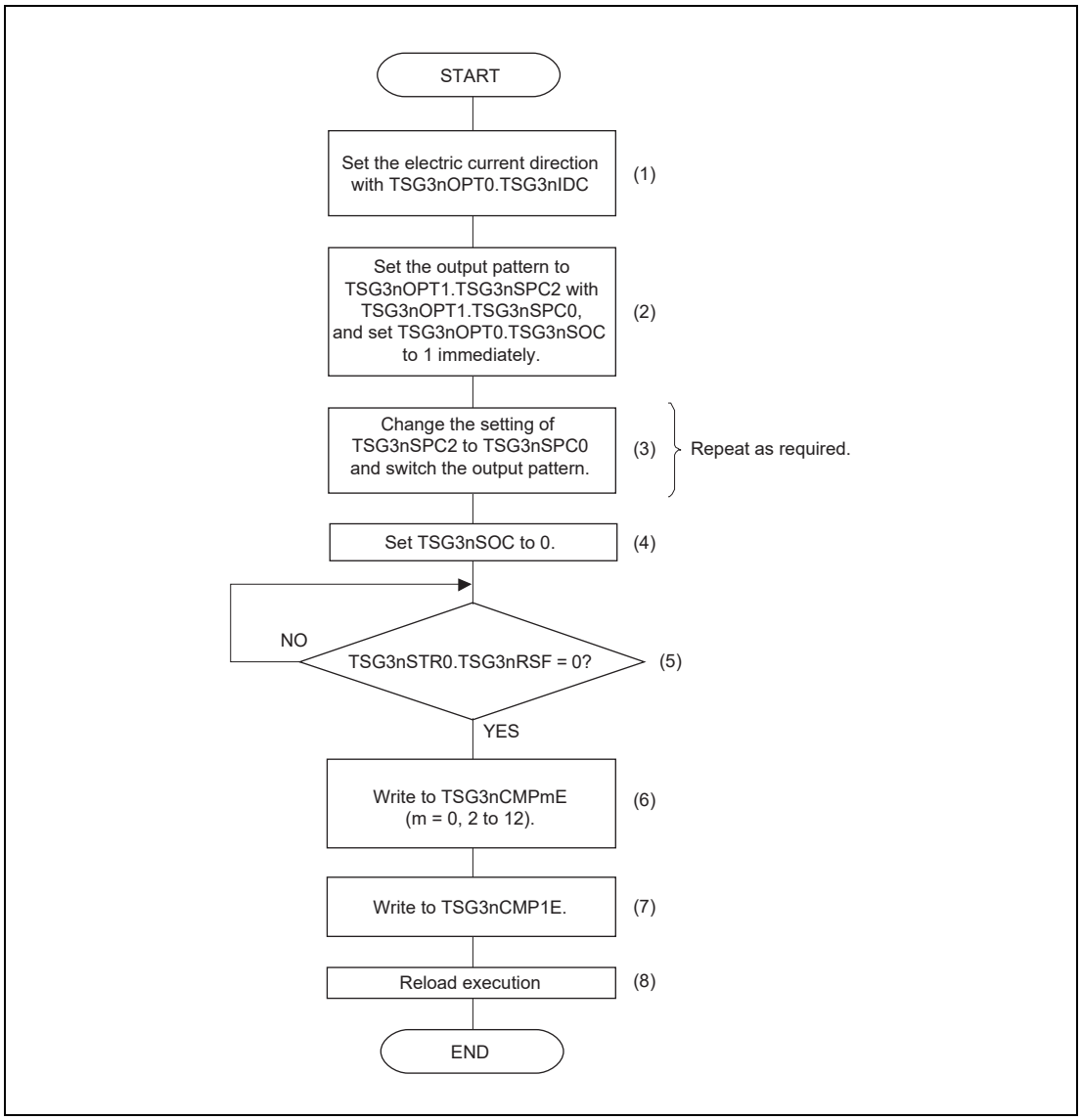


Figure 35.99 Process Flow of Software Output Control

The procedure for software output control is described below.

(1) Set the TSG3nIDC bit. The phase of the timer output with TSG3nIDC = 0 is different by 180 degrees from that with TSG3nIDC = 1. With the software output control function, the timer output will not change only by rewriting this bit. However, if the period match occurs before step (2), the output pattern of 120-DC control changes. Therefore, specify a schedule that will prevent the period match from occurring before step (2).

(2) Set the output pattern with TSG3nSPC2 to TSG3nSPC0. To enable software output control, set TSG3nSOC to 1 simultaneously.

(3) Change the output pattern setting of TSG3nSPC2 to TSG3nSPC0 to change the timer output.

(4) Confirm that the reload request flag (TSG3nRSF) = 0. If TSG3nRFS = 1, do not proceed to the following step until TSG3nRSF = 0.

(5) By setting TSG3nSOC = 0 the software control release process starts (it is not released yet at this point).

(6) Configure the compare register settings that are required after the software output control is released. Proceed to the following step when the register setting is not required. When setting the registers with the reload function, set them at this point.

(7) Write to TSG3nCMP1E to start reloading.

(8) Reload is executed and software output is released.

CAUTION

Execute reload after completing steps (4) to (7). When reload cannot be executed, the software output cannot be released.

35.4.7.5 HSP-PWM Mode (High accuracy Shifted-pulse - Pulse Width Modulation Mode)

Overview

In this mode, the 18-bit counter and the 18-bit compare register are used to generate a high accuracy sawtooth waveform PWM signal.

Prerequisites

- Set the PWM period to TSG3nCMP0E.
- Set PWM output width with the TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE registers. Set PWM shift width with the TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE registers. Set dead time with the TSG3nDTC0W and TSG3nDTC1W registers. The values set in these registers are immediately reflected to the corresponding TSG3nCMPmE (m = 1 to 12) based on the calculation described later.

Functional description

Set the PWM period, the dead time, and the PWM shift width. Then set the PWM output width. Counting up begins when TSG3nTRG0.TSG3nTS is set to 1.

The 18-bit counter starts counting from 00000_H and is cleared by the match with TSG3nCMP0E.

During counting, a compare match interrupt (INTTSG3nI0 to INTTSG3nI12) is generated by the match of the buffer register TSG3nCMP0E to TSG3nCMP12E with the 18-bit counter.

NOTE

The HSP-PWM mode is set when TSG3nCTL0.TSG3nMD2-TSG3nMD0 = 100_B.

(1) Basic Timing Chart

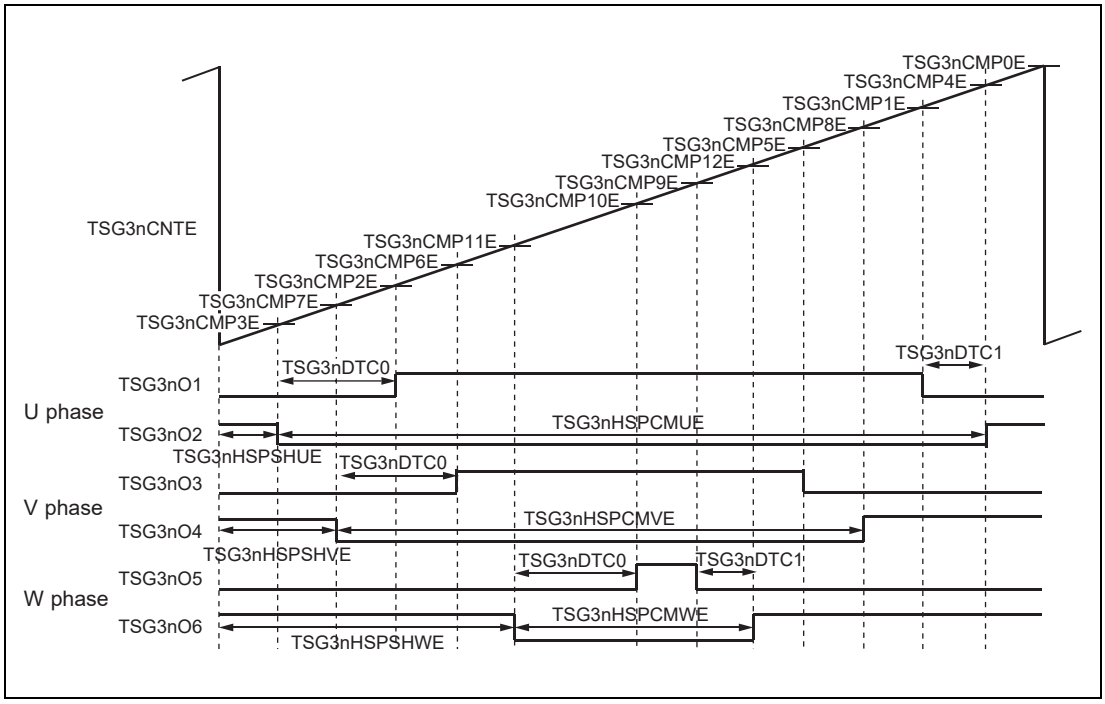


Figure 35.100 Basic Timing in HSP-PWM Mode

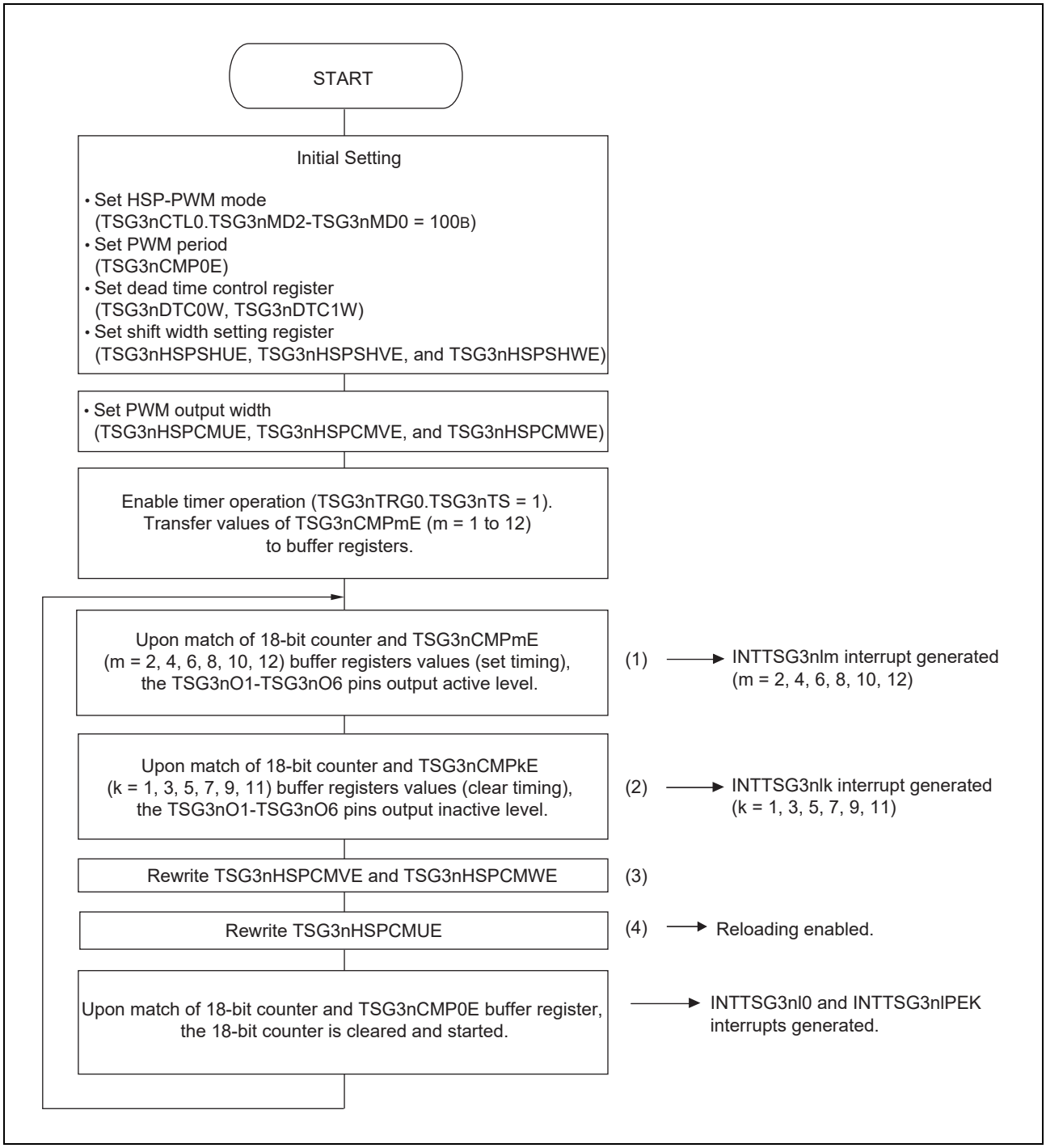


Figure 35.101 Basic Operation Flow in HSP-PWM Mode

NOTE

The timing may be different from the orders described above but be sure to execute (4) after (3).

CAUTIONS

1. **When changing the settings for PWM output width in the TSG3nHSPCMUE, VE, and WE registers during operation, set the TSG3nHSPCMUE register last. When only the settings for the V phase and W phase are changed but the PWM output width in the U phase is not, the existing value should be written back to the TSG3nHSPCMUE register.**
2. **After changing the PWM cycle by using the TSG3nCMP0E register, the PWM output width settings in the TSG3nHSPCMUE, VE, and WE registers must be reconfigured. Changing the settings for the PWM shift width in the TSG3nHSPSHUE, VE, and WE registers and the settings for the PWM cycle in the TSG3nCMP0E register at the same time is not allowed.**
3. **Reconfigure the settings for PWM output width in the TSG3nHSPCMUE, VE, and WE registers after changing the settings for PWM shift width in the TSG3nHSPSHUE, VE, and WE registers.**

(2) List of Operation in HSP-PWM

Table 35.93 Counter Function in HSP-PWM

Operation		Setting Condition
18-bit counter	Start	TSG3nTRG0.TSG3nTS = 0 → 1 or simultaneous start trigger
	Clear	Compare match of TSG3nCMP0E buffer register and 18-bit counter
	Stop	TSG3nTRG1.TSG3nTT = 0 → 1

Table 35.94 Functions of Compare Registers, Shift Width Setting Register, and Dead Time Control Register in HSP-PWM

Register	Rewriting Method	Rewrite during Operation	Function
TSG3nCMP0E	Reload	Possible	Setting period
TSG3nHSPCMUE	Reload	Possible	PWM control for U phase
TSG3nHSPSHUE			
TSG3nHSPCMVE	Reload	Possible	PWM control for V phase
TSG3nHSPSHVE			
TSG3nHSPCMWE	Reload	Possible	PWM control for W phase
TSG3nHSPSHWE			
TSG3nDCMP0E, TSG3nDCMP1E, TSG3nDCMP2E	Reload	Possible	Diagnostic output or A/D conversion trigger
TSG3nDTC0W, TSG3nDTC1W	Reload	Prohibited	Dead time

Table 35.95 Timer Output in HSP-PWM Mode

Pin	Function
TSG3nO1	PWM output by compare match of the TSG3nCMP1E buffer register (clear timing) or the TSG3nCMP2E buffer register (set timing) with the 18-bit counter.
TSG3nO2	PWM output by compare match of the TSG3nCMP3E buffer register (clear timing) or the TSG3nCMP4E buffer register (set timing) with the 18-bit counter.
TSG3nO3	PWM output by compare match of the TSG3nCMP5E buffer register (clear timing) or the TSG3nCMP6E buffer register (set timing) with the 18-bit counter.
TSG3nO4	PWM output by compare match of the TSG3nCMP7E buffer register (clear timing) or the TSG3nCMP8E buffer register (set timing) with the 18-bit counter.
TSG3nO5	PWM output by compare match of the TSG3nCMP9E buffer register (clear timing) or the TSG3nCMP10E buffer register (set timing) with the 18-bit counter.
TSG3nO6	PWM output by compare match of the TSG3nCMP11E buffer register (clear timing) or the TSG3nCMP12E buffer register (set timing) with the 18-bit counter.
TSG3nO7	Diagnostic signal output or pulse output by A/D conversion trigger

Table 35.96 Interrupt Request in HSP-PWM Mode

Interrupt	Function
INTTSG3nIm (m = 0 to 12)	Compare match of the TSG3nCMPmE buffer register with 18-bit counter
INTTSG3nIER	Error
INTTSG3nIVLY	—
INTTSG3nIPEK	Peak interrupt (generated at the same timing as INTTSG3nI0)
INTTSG3nIWN	Error

Table 35.97 Compare Match Timing in HSP-PWM Mode

Compare Match	Timing
TSG3nCMP0E	When 18-bit counter changes from TSG3nCMP0E to 00000_H
TSG3nCMPmE (m = 1 to 12)	After match of 18-bit counter and TSG3nCMPmE

(3) Various Settings of HSP-PWM Mode

Setting mode

HSP-PWM mode is entered by setting TSG3nCTL0.TSG3nMD2-TSG3nMD0 to 100_B.

Setting timer output

The output pins TSG3nO1-6 are controlled by setting TSG3nIOC0, TSG3nIOC2, and TSG3nIOC3.

The TSG3nO7 pin provides output pulse as diagnostic output or A/D conversion trigger. The pin should be set as necessary.

Enabling error interrupt generation

Error interrupt (INTTSG3nIER) due to the detection of the simultaneous active state of positive and inverse phases is enabled by setting TSG3nIOC1.TSG3nEOC to 1. For details, see **Section 35.4.6, Error/Warning Interrupt**.

Setting rewriting timing of register with reload function

This function is only available in reload mode. Set TSG3nCTL3.TSG3nRMC to 0.

Setting A/D conversion trigger output

The A/D conversion trigger 0 (TSG3nADTRG0 signal) is set with TSG3nCTL5.TSG3nAT09-TSG3nAT00.

TSG3nAT09-TSG3n00 is used to enable or disable the A/D conversion trigger output upon match of TSG3nDCMP2E-0E with 18-bit counter (up count).

TSG3nCTL6.TSG3nAT19-TSG3nAT10 is used to set the A/D conversion trigger 1 (TSG3nADTRG1 signal).

To set the match timing of the 18-bit counter and TSG3nDCMP2E-TSG3nDCMP0E, set the compare value to each register.

The skipping function can be used for TSG3nADTRG0, TSG3nADTRG1 signals. TSG3nACC01 and TSG3nACC00 of TSG3nCTL5, and TSG3nACC11 and TSG3nACC10 of TSG3nCTL6 can be used to select the skipping rate from 1/1 (no skipping), 1/2, 1/4, and 1/8.

CAUTION

- **Be sure to set TSG3nCTL5, TSG3nCTL6, and TSG3nDCMP2E-0E correctly when the timing pulse of A/D conversion trigger is output to TSG3nO7.**
- **In HSP-PWM mode, no valley interrupt (INTTSG3nIVLY) is generated. Therefore, SG3nCTL5.TSG3nAT00 and TSG3nCTL6.TSG3nAT10 must be set to 0.**
- **In HSP-PWM mode, the 18-bit sub-counter does not operate. Therefore, TSG3nCTL5.TSG3nAT09 and TSG3nAT08, and TSG3nCTL6.TSG3nAT19 and TSG3nAT18 must be set to 0.**
- **In HSP-PWM mode, down counting by the 18-bit counter is not generated. Therefore, TSG3nCTL5.TSG3nAT07, TSG3nAT05, and TSG3nAT03, and TSG3nCTL6.TSG3nAT17, TSG3nAT15, and TSG3nAT13 must be set to 0.**

Setting PWM period

Set the PWM period with TSG3nCMP0E.

The PWM period is calculated by the following expression:

PCLK × (TSG3nCMP0E + 1)

Setting PWM output width

The PWM output width is set with TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE (TSG3nCMP1E-12E).

Satisfy the following requirements when setting the TSG3nHSPCMUE, TSG3nHSPCMVE, and TSG3nHSPCMWE registers:

0 ≤ TSG3nHSPCMUE,TSG3nHSPCMVE, TSG3nHSPCMWE ≤ TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1

Set the PWM output width after setting TSG3nCMP0E, TSG3nHSPSHUE, TSG3nHSPSHVE, TSG3nHSPSHWE, TSG3nDTC0, and TSG3nDTC1.

Setting PWM shift width

PWM shift width is set with TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE.

Satisfy the following requirements when setting TSG3nHSPSHUE, TSG3nHSPSHVE, and TSG3nHSPSHWE.

TSG3nHSPSHUE, TSG3nHSPSHVE, TSG3nHSPSHWE ≤ TSG3nCMP0E

Setting dead time

The dead time can be set with TSG3nDTC0 and TSG3nDTC1.

The dead time is calculated by the following expressions:

PCLK × TSG3nDTC0

PCLK × TSG3nDTC1

TSG3nDTC0 can set the time between a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the inactive state and a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the active state, respectively.

TSG3nDTC1 can set the time between a change of TSG3nO1, TSG3nO3, and TSG3nO5 to the inactive state and a change of TSG3nO2, TSG3nO4, and TSG3nO6 to the active state, respectively.

Satisfy the following requirements when setting TSG3nDTC0 and TSG3nDTC 1.

(TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1) < $3FFFF_H$

TSG3nCMP0E > 3 × TSG3nDTC0

TSG3nCMP0E > 3 × TSG3nDTC1

CAUTION

Do not modify the settings of TSG3nDTC0 and TSG3nDTC1 during timer operation in HSP-PWM (TSG3nTE = 1). Set TSG3nDTC0 and TSG3nDTC1 while TSG3nTE = 0.

Set dead time in HSP-PWM mode. Do not set 0 to TSG3nDTC0 and TSG3nDTC1.

Settings prohibited in HSP-PWM Mode

In HSP-PWM mode, use the following registers and bits with the settings shown below.

Do not modify the settings during operation (TSG3nTE = 1).

Table 35.98 Setting Prohibited in HSP-PWM Mode

Bit Name	Setting Value	Description
TSG3nCTL3.TSG3nRMC	0	Available only in reload mode.
TSG3nIOC3.TSG3nTOL6-1	000000_B	Logical inverse of set/clear of PWM is prohibited (HSP-PWM mode limitation)
TSG3nOPT0.TSG3nSOC	0	Switching to software control function is prohibited.
TSG3nOPT0.TSG3nSTE	0	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nOPT0.TSG3nPOT	0	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nOPT0.TSG3nPSS	0	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nOPT0.TSG3nIDC	0	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nOPT0.TSG3nPSC	0	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nOPT1.TSG3nSPC2-0	000_B	Operation setting of 120-DC mode and software control function (value after reset).
TSG3nPAT0W	0000000_H	Operation setting of 120-DC mode (value after reset).
TSG3nPAT1W	0000000_H	Operation setting of 120-DC mode (value after reset).

CAUTION

In HSP-PWM mode, no setting should be made directly to the TSG3nCMPmE register (m = 1 to 12).

The PWM output width and the PWM shift width should be set with TSG3nHSPCMUE, VE, and WE registers and TSG3nHSPSHUE, VE, and WE registers.

35.4.7.6 Compare Register Setting in HSP-PWM Mode

In HSP-PWM mode, the PWM output width is set with TSG3nHSPCMUE, VE and WE.

With a write access to TSG3nHSPCMUE, VE, and WE, TSG3 calculates and sets the values to TSG3nCMP1E-12E based on the values written to the followings:

- TSG3nCMP0E (PWM period setting)
- TSG3nDTC0 (dead time control 0)
- TSG3nDTC1 (dead time control 1)
- TSG3nHSPSHUE/VE/WE (PWM shift width setting)
- TSG3nHSPCMUE/VE/WE (PWM output width setting)

The high accuracy PWM is realized by these values.

The algorithm to set compare register values are listed in the following table.

Table 35.99 Algorithm for Compare Setting in HSP-PWM Mode

Value Set In HSPCMUE					CMP4E	CMP3E	CMP2E	CMP1E
HSPCMUE = 0					if (HSPSHUE = 0) 0 else HSPCMUE – 1 + HSPSHUE	CMP0E + 1	0	0
0	<	HSPCMUE	≤	DTC0 + DTC1	HSPCMUE – 1 + HSPSHUE	if (HSPSHUE = 0) CMP0E else HSPSHUE – 1	0	0
DTC0 + DTC1	<	HSPCMUE	≤	CMP0E	HSPCMUE – 1 + HSPSHUE	if (HSPSHUE = 0) CMP0E else HSPSHUE – 1	DTC0 – 1 + HSPSHUE	HSPCMUE – DTC1 – 1 + HSPSHUE
CMP0E	<	HSPCMUE	≤	CMP0E + DTC1 + 1	0	0	DTC0 – 1 + HSPSHUE	HSPCMUE – DTC1 – 1 + HSPSHUE
CMP0E + DTC1 + 1	<	HSPCMUE	<	CMP0E + DTC0 + DTC1 + 1	0	0	DTC0 – 1 + HSPSHUE	HSPCMUE – CMP0E – DTC1 – 2 + HSPSHUE
HSPCMUE = CMP0E + DTC0 + DTC1 + 1					0	0	DTC0 – 1 + HSPSHUE	CMP0E + 1

: For the colored cells, subtract CMP0E + 1 when the calculated result is greater than CMP0E.

NOTE

"TSG3n" is omitted from the register names used in the calculation.

35.4.7.7 Timer Output Operation in HSP-PWM Mode

TSG3nO1-6 output is set by compare match of TSG3nCNTE with TSG3nCMP2E, 4E, 6E, 8E, 10E, and 12E, respectively, and cleared by compare match of TSG3nCNTE with TSG3nCMP1E, 3E, 5E, 7E, 9E, and 11E, respectively.

When PWM output width is set in TSG3nHSPCMUE,VE, and WE, the value is set in the TSG3nCMP1E-12E registers based on the calculation described in **Section 35.4.7.6, Compare Register Setting in HSP-PWM Mode** that enables a high accuracy PWM output from 0% to 100%.

PWM output timing can be shifted by setting desired width to TSG3nHSPSHUE, VE, WE.

(1) When TSG3nHSPCMUE, VE, and WE (PWM output width setting) is set to 0

When 0 is set to TSG3nHSPCMUE (U phase PWM output width setting), 0 is set to TSG3nCMP1E, 2E, and 4E, and "TSG3nCMP0E+1" is set to TSG3nCMP3E.

Setting of TSG3nO1 by the match of TSG3nCNTE with TSG3nCMP2E and clearing by the match with TSG3nCMP1E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO1 output is fixed to inactive.

Setting of TSG3nO2 by the match of TSG3nCNTE with TSG3nCMP4E occurs when TSG3nCNTE = 0. Since TSG3nCMP0E + 1 is set to TSG3nCMP3E, TSG3nCNTE does not match TSG3nCMP3E. Therefore, TSG3nO2 output is fixed to active.

The same operation applies to TSG3nO3-6, which are the output pins for V phase and W phase when 0 is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

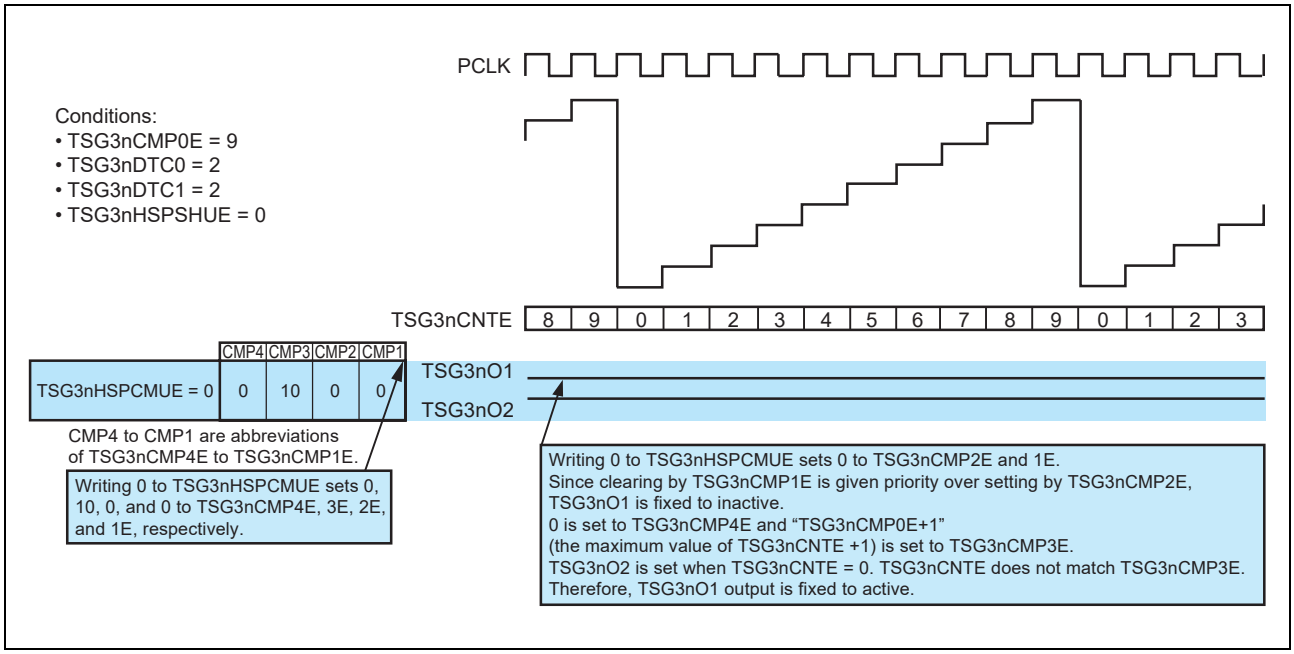


Figure 35.102 Waveform in HSP-PWM Mode (0 is set to TSG3nHSPCMUE)

(2) When TSG3nHSPCMUE/VE/WE (PWM output width setting) is set to the range 0 < TSG3nHSPCMUE ≤ TSG3nDTC0 + TSG3nDTC1

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), 0 is set to TSG3nCMP1E and 2E, “TSG3nHSPCMUE set value - 1” is set to TSG3nCMP4E, and the value of the TSG3nCMP0E is set to TSG3nCMP3E.

Setting of TSG3nO1 by the match of TSG3nCNTE with TSG3nCMP2E and clearing by the match with TSG3nCMP1E occurs simultaneously, but clearing takes precedence. Therefore, TSG3nO1 output is fixed to inactive.

TSG3nO2 is cleared by the match of TSG3nCNTE with TSG3nCMP3E and set by the match with TSG3nCMP4E. Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO2 is inactive for one cycle during PWM period when 1 is set, two cycles when 2 is set, and three cycles when 3 is set, and so on.

When the value other than 0 is set to TSG3nHSPSHUE (PWM shift width), set/clear timing of TSG3nO2 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “0 < TSG3nHSPCMUE ≤ TSG3nDTC0 + TSG3nDTC1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

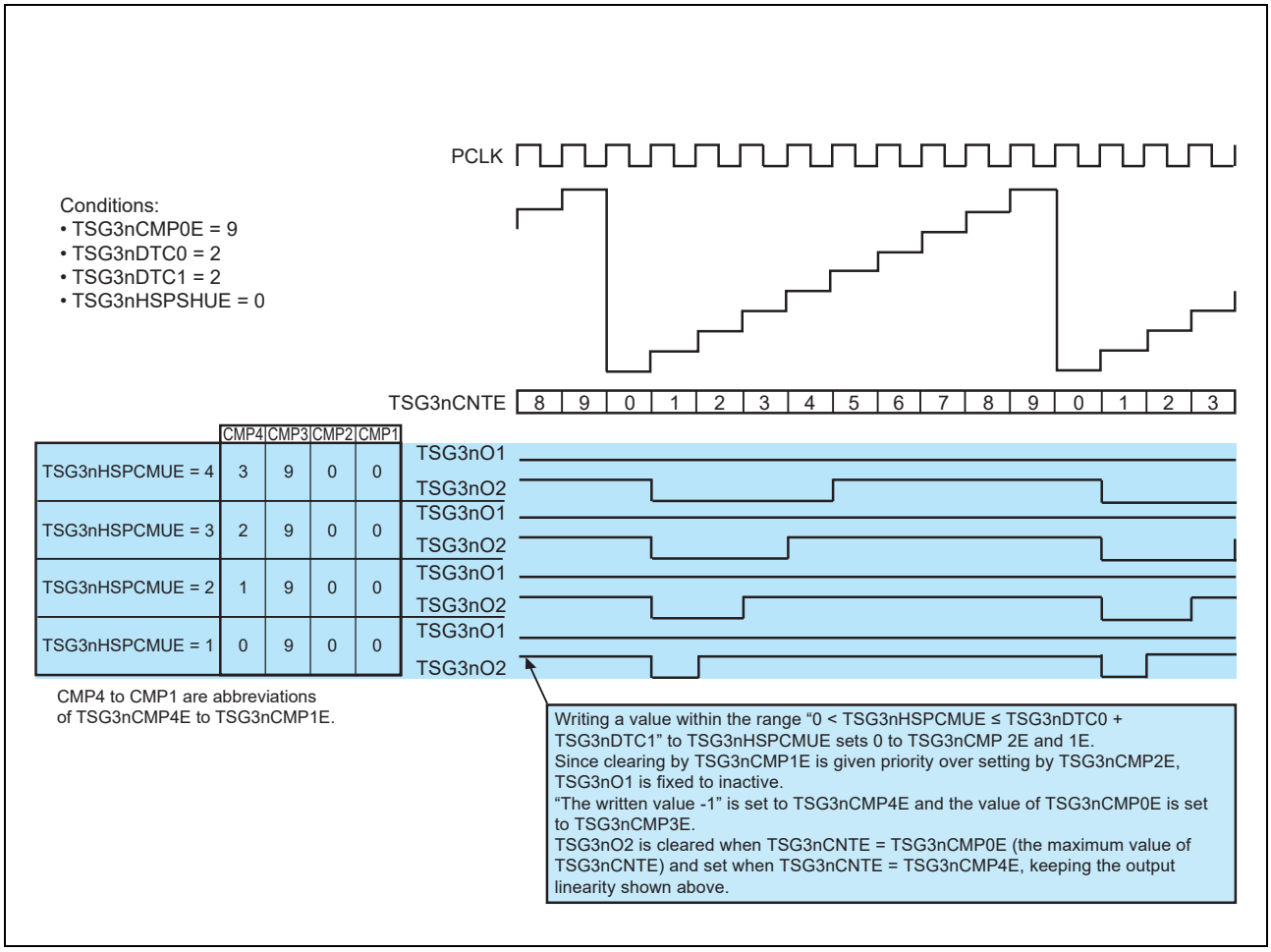


Figure 35.103 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to 0 < TSG3nHSPCMUE ≤ TSG3nDTC0 + TSG3nDTC1)

(3) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range TSG3nDTC0 + TSG3nDTC1 <TSG3nHSPCMUE/VE/WE ≤ TSG3nCMP0E

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTCD0 -1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nDTC1 - 1” is set to TSG3nCMP1E, “TSG3nHSPCMUE - 1” is set to TSG3nCMP4E, and the value of TSG3nCMP0E is set to TSG3nCMP3E.

TSG3nO1 is set by the match of TSG3nCNTE with TSG3nCMP2E and cleared by the match with TSG3nCMP1E while TSG3nO2 is set by the match of TSG3nCNTE with TSG3nCMP4E and cleared by the match with TSG3nCMP3E

Here, the outputs are shifted according to the set value in TSG3nHSPCMUE, that is, when “TSG3nDTC0 + TSG3nDTC1 + 1” is set, TSG3nO1 is active for one cycle during PWM period and TSG3nO2 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1+ 1” during PWM period. When “TSG3nDTC0 + TSG3nDTC1 + 2” is set, TSG3nO1 is active for two cycles during PWM period and TSG3nO2 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1+ 2” during PWM period, and so on.

When the value other than 0 is set to TSG3nHSPSHUE (PWM shift width), set/clear timing of TSG3nO1 and TSG3nO2 are shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nDTC0 + TSG3nDTC1 <TSG3nHSPCMUE/VE/WE ≤ TSG3nCMP0E” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

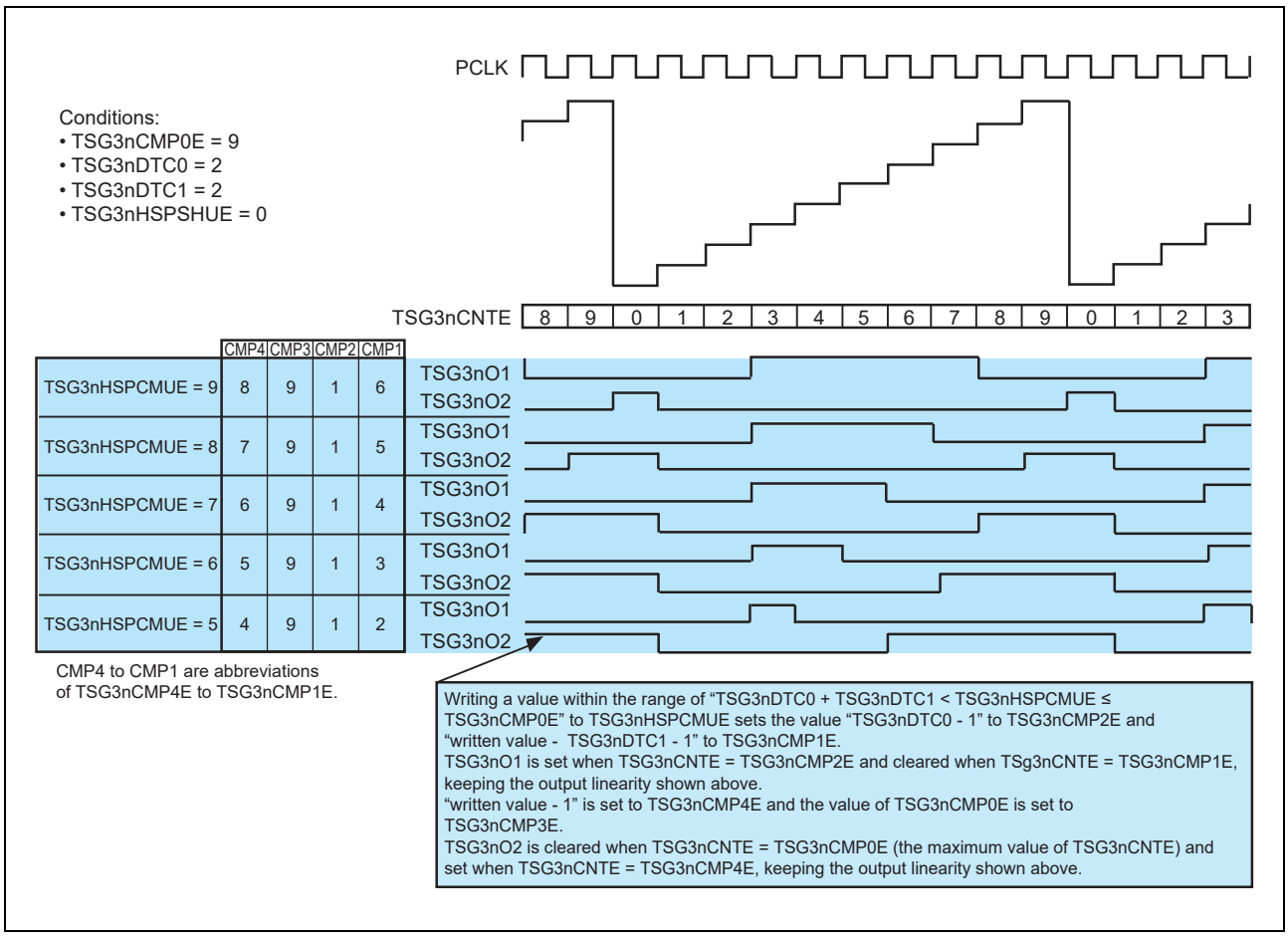


Figure 35.104 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to TSG3nDTC0 + TSG3nDTC1 < TSG3nHSPCMUE ≤ TSG3nCMP0E)

(4) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range TSG3nCMP0E < TSG3nHSPCMUE/VE/ WE < TSG3nCMP0E + TSG3nDTC1 + 1

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nDTC1 - 1” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

TSG3nO1 is cleared by the match of TSG3nCNTE with TSG3nCMP1E and set by the match with TSG3nCMP2E.

Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO1 is inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1” during PWM period when “TSG3nCMP0E + 1” is set, inactive for the cycles of “TSG3nDTC0 + TSG3nDTC1 - 1” during PWM period when “TSG3nCMP0E + 2” is set, and so on.

Setting of TSG3nO2 by the match of TSG3nCNTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE (U phase PWM shift width), set/clear timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nCMP0E < TSG3nHSPCMUE/VE/ WE < TSG3nCMP0E + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

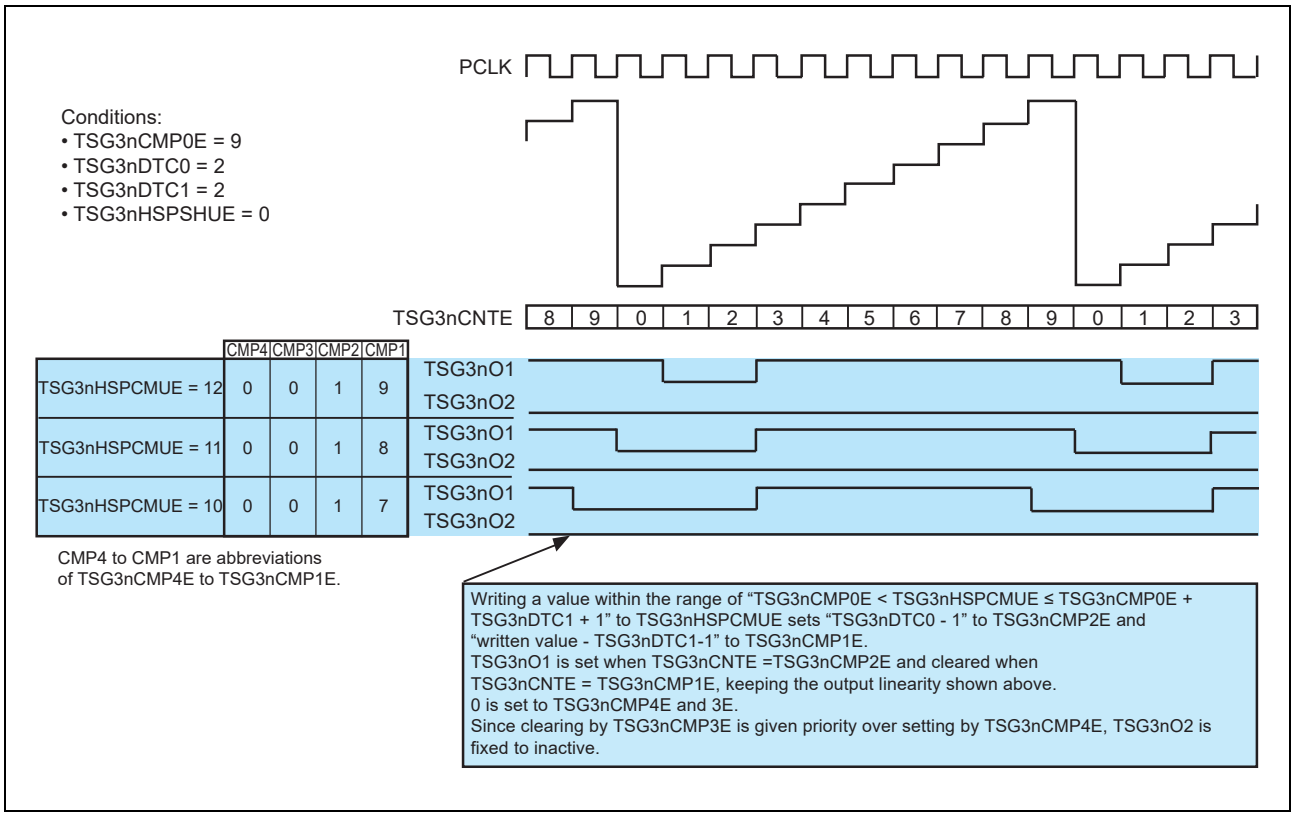


Figure 35.105 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to TSG3nCMP0E < TSG3nHSPCMUE < TSG3nCMP0E + TSG3nDTC1 + 1)

(5) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to the range TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE/VE/WE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1

When the value within this range is set to TSG3nHSPCMUE (U phase PWM output width setting), “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nHSPCMUE - TSG3nCMP0E - TSG3nDTC1 - 2” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

TSG3nO1 is cleared by the match of TSG3nCNTE with TSG3nCMP1E and set by the match with TSG3nCMP2E. Here, the output is shifted according to the set value in TSG3nHSPCMUE, that is, TSG3nO1 is inactive for one cycle during PWM period when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1” (the maximum value of PWM output width - 1) is set, inactive for two cycles during PWM period when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 - 1” (the maximum value of PWM output width - 2) is set, and so on.

Setting of TSG3nO2 by the match of TSG3nCNTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE (U Phase PWM shift width), set/clear timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when the value of “TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE/VE/WE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

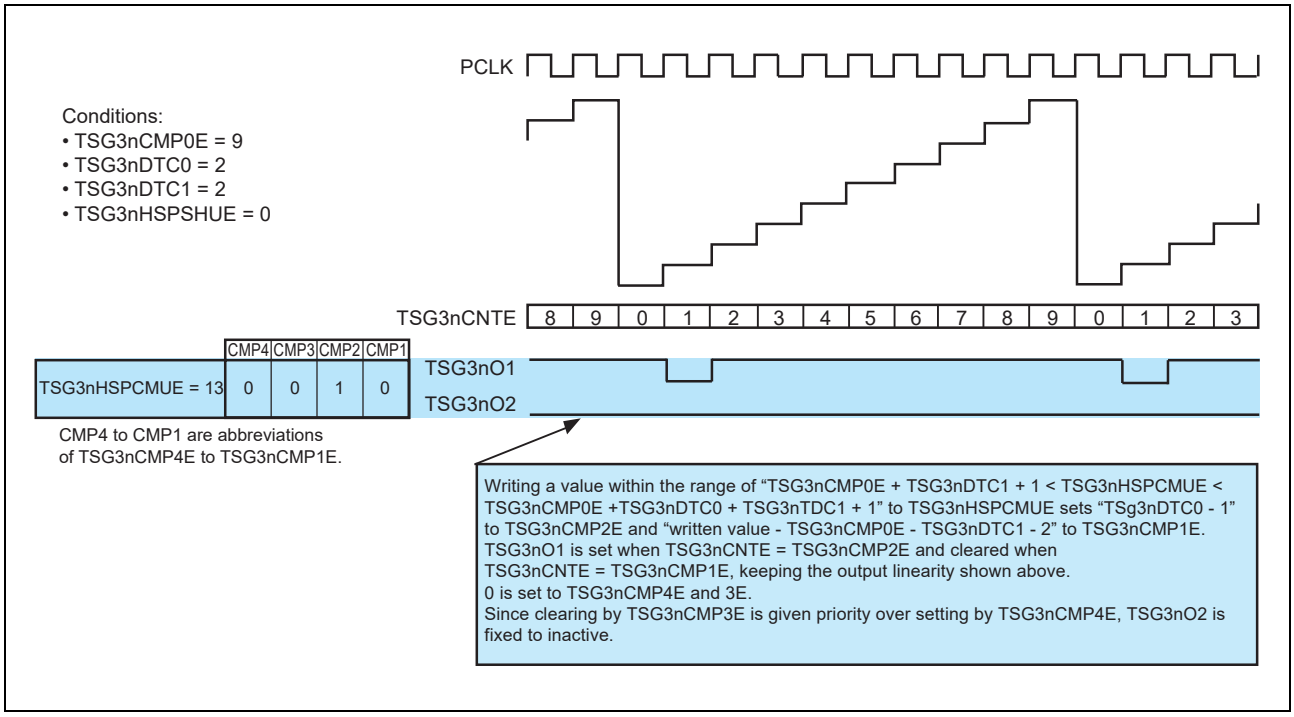


Figure 35.106 Waveform in HSP-PWM Mode (TSG3nHSPCMUE is set to TSG3nCMP0E + TSG3nDTC1 + 1 < TSG3nHSPCMUE < TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1)

(6) When TSG3nHSPCMUE/VE/WE (PWM Output Width Setting) is set to TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1

When “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1” (the maximum value of PWM output width) is set to TSG3nHSPCMUE, “TSG3nDTC0 - 1” is set to TSG3nCMP2E, “TSG3nCMP0E + 1” is set to TSG3nCMP1E, and 0 is set to TSG3nCMP3E and 4E.

Setting of TSG3nO1 by the match of TSG3nCNTE with TSG3nCMP2E occurs when TSG3nCNTE = TSG3nDTC0 - 1.

Since TSG3nCMP0E + 1 is set to TSG3nCMP3E, TSG3nCNTE does not match TSG3nCMP3E. Therefore, TSG3nO1 output is fixed to active.

Setting of TSG3nO2 by the match of TSG3nCNTE with TSG3nCMP4E and clearing by the match with TSG3nCMP3E occur simultaneously, but clearing takes precedence. Therefore, TSG3nO2 output is fixed to inactive.

When the value other than 0 is set to TSG3nHSPSHUE, set timing of TSG3nO1 is shifted to the right for the number of cycles set in TSG3nHSPSHUE. Note that the set timing is shifted only at operation start (TSG3nTE from 0 to 1) because TSG3nO1 output is fixed to active. For the operation when operation start, see **Section 35.4.7.7, (8) Timer Output Operation at Operation Start in HSP-PWM Mode**.

The same condition applies to TSG3nO3-6, which are the output pins for V phase and W phase when “TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1” is set to TSG3nHSPCMVE (V phase PWM output width setting) and TSG3nHSPCMWE (W phase PWM output width setting).

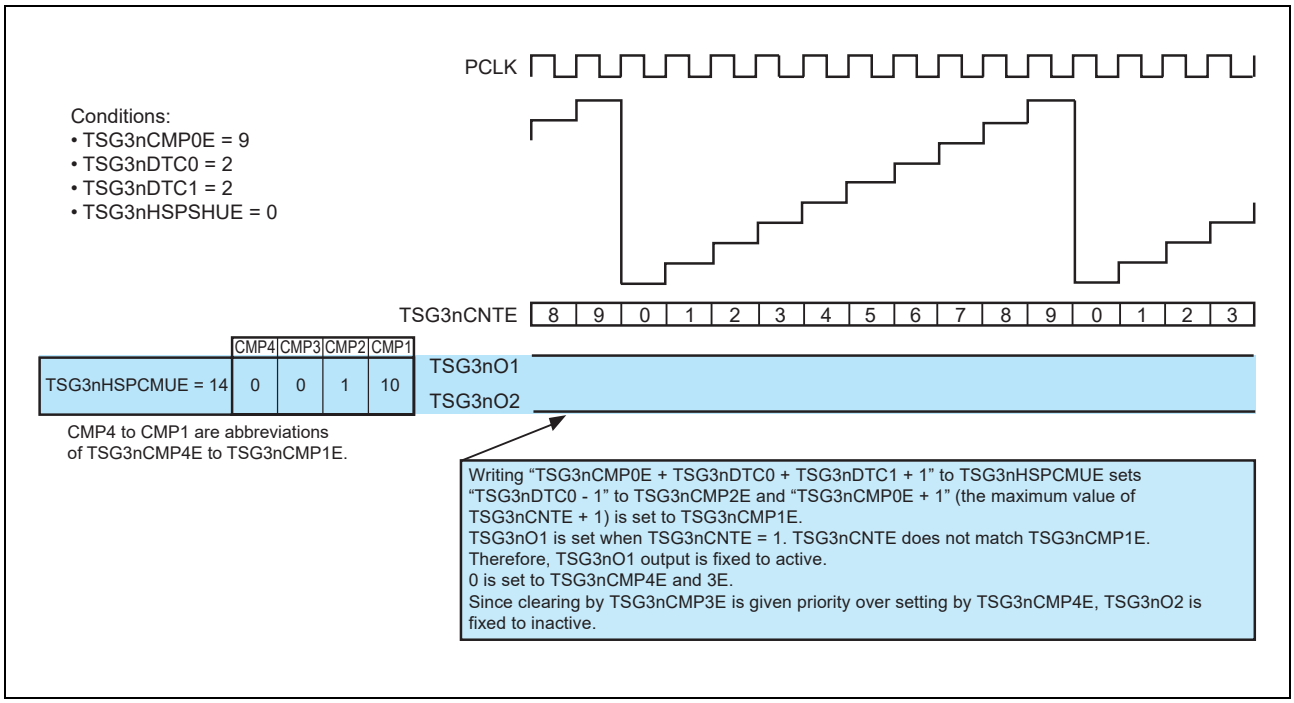


Figure 35.107 Waveform in HSP-PWM (TSG3nHSPCMUE is set to TSG3nCMP0E + TSG3nDTC0 + TSG3nDTC1 + 1)

NOTES

1. In HSP-PWM mode, TSG3nO1 and TSG3nO2 can be set at the same timing by setting the same value to TSG3nCMP4E and 2E. In this case, both TSG3nO1 and TSG3nO2 become inactive.
2. In HSP-PWM mode, dead time counter keeps operating even when TSG is stopped (TSG3nTE = 0) and the dead time set in TSG3nO1 and TSG3nO2 are always inserted.

(7) PWM Adjustment at Reload Timing in HSP-PWM Mode

In HSP-PWM mode, PWM output width is adjusted when TSG3nHSPCMUE/VE/WE (PWM output width setting) is modified during operation. TSG3nO1-6 are set/cleared at reload timing and immediately switched to the output reflecting the new PWM output width.

TSG3nO1-6 are forcibly set/cleared according to the following formula. Even for adjustment, the dead time set with TSG3nDTC0 and 1 is always inserted to TSG3nO1-6. If values are directly written to TSG3nCMP1-12E, output adjustment is not performed at reload timing.

Formula for reload adjustment operation

Table 35.100 Formula for Reload Adjustment Operation When TSG3nHSPSHUE/VE/WE is 0 (PWM shift width is set to 0)

Pin	Set	Clear
TSG3nO1/3/5	CMP0E + DTC1 + 1 < HSPCMUE/VE/WE	HSPCMUE/VE/WE ≤ CMP0E + DTC1 + 1
TSG3nO2/4/6	HSPCMUE/VE/WE = 0	0 < HSPCMUE/VE/WE

Table 35.101 Formula for Reload Adjustment Operation When TSG3nHSPSHUE/VE/WE is not 0 (PWM shift width is not set to 0)

Pin	Set	Clear
TSG3nO1/3/5	(i) CMP0E + DTC1 + 1 – HSPSHUE/VE/WE < HSPCMUE/VE/WE	HSPCMUE/VE/WE ≤ CMP0E + DTC1 + 1 – HSPSHUE/VE/WE
	(ii) (CMP0E + 1) × 2 + DTC1 – HSPSHUE/VE/WE < HSPCMUE/VE/WE	HSPCMUE/VE/WE ≤ (CMP0E + 1) × 2 + DTC1 – HSPSHUE/VE/WE
TSG3nO2/4/6	HSPCMUE/VE/WE ≤ CMP0E + 1 – HSPSHUE/VE/WE	CMP0E + 1 – HSPSHUE/VE/WE < HSPCMUE/VE/WE

If the value other than 0 is set as PWM shift width, set/clear condition for positive phase is determined whether the set shift width is greater than “CMP0E + DTC0 – 1” or not.

(i) HSPSHUE/VE/WE ≤ CMP0 – DTC0 + 1

(ii) HSPSHUE/VE/WE > CMP0 – DTC0 + 1

“TSG3n” is omitted from the register names used in the calculation.

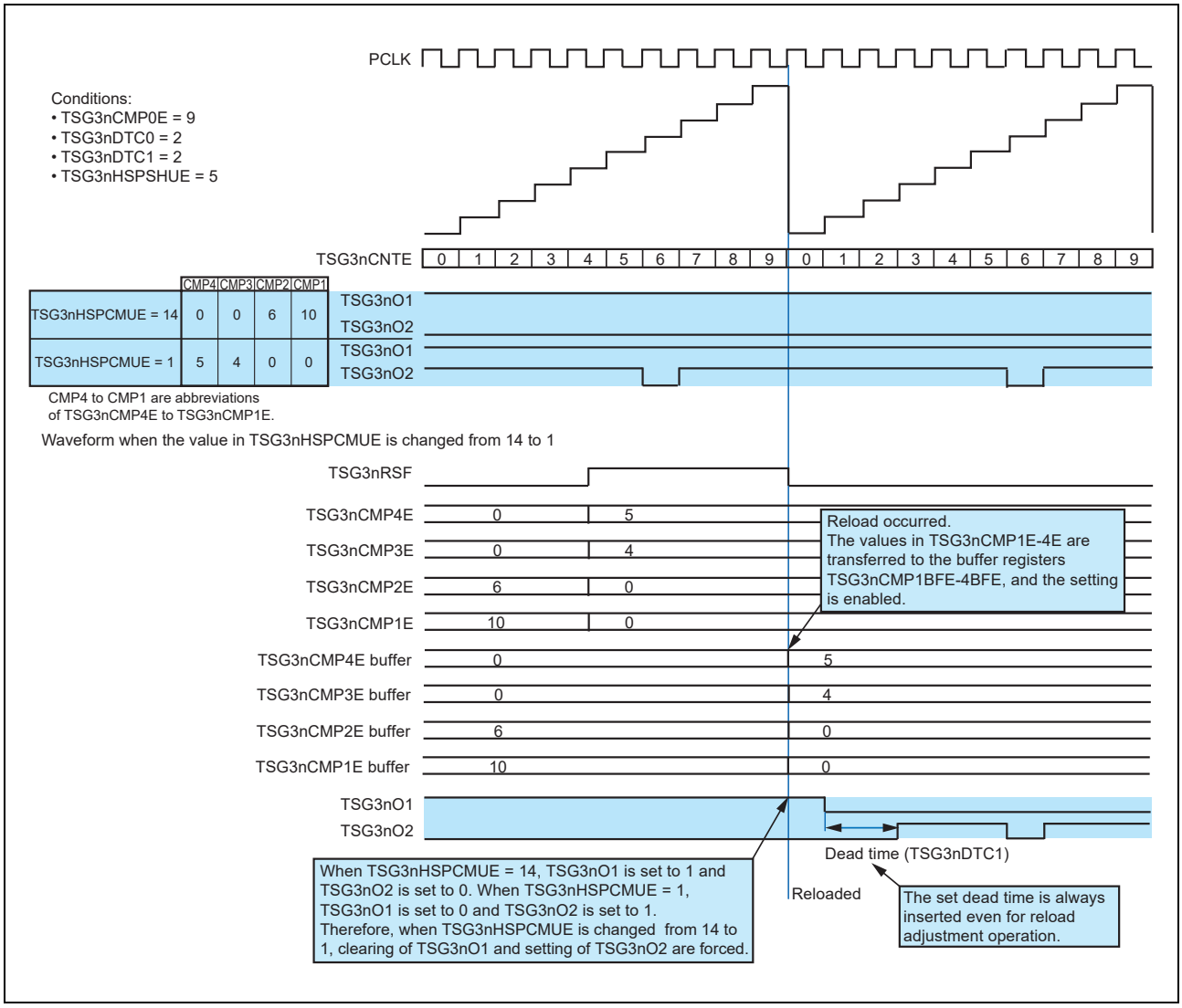


Figure 35.108 Detailed Timing Diagram of Reload Adjustment (Change of TSG3nHSPCMUE from 14 to 1)

(8) Timer Output Operation at Operation Start in HSP-PWM Mode

In HSP-PWM mode, TSG3nO1-6 are cleared when operation starts.

Then, TSG3nO1-6 are set/cleared as TSG3nCNTE counts up depending on the values set in TSG3nHSPCMUE/VE/WE (TSG3nCMP1E to 12E).

Even if TSG3nO1-6 are set before operation start and cleared at operation start, and then set again, the set dead time is always inserted.

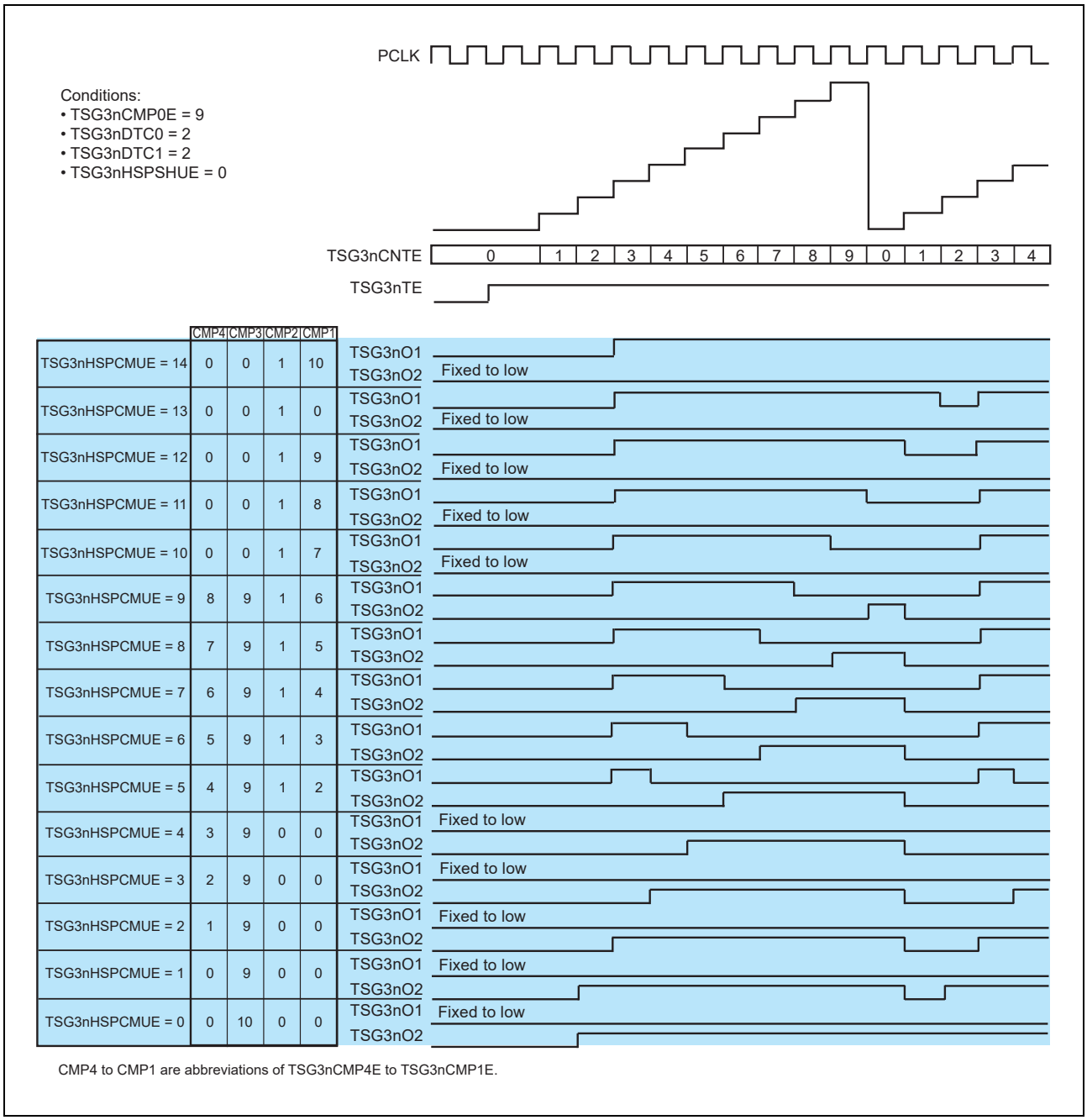


Figure 35.109 Timing Diagram of Operation Start in HSP-PWM Mode (TSG3nHSPSHUE = 0 (shift width is set to 0))

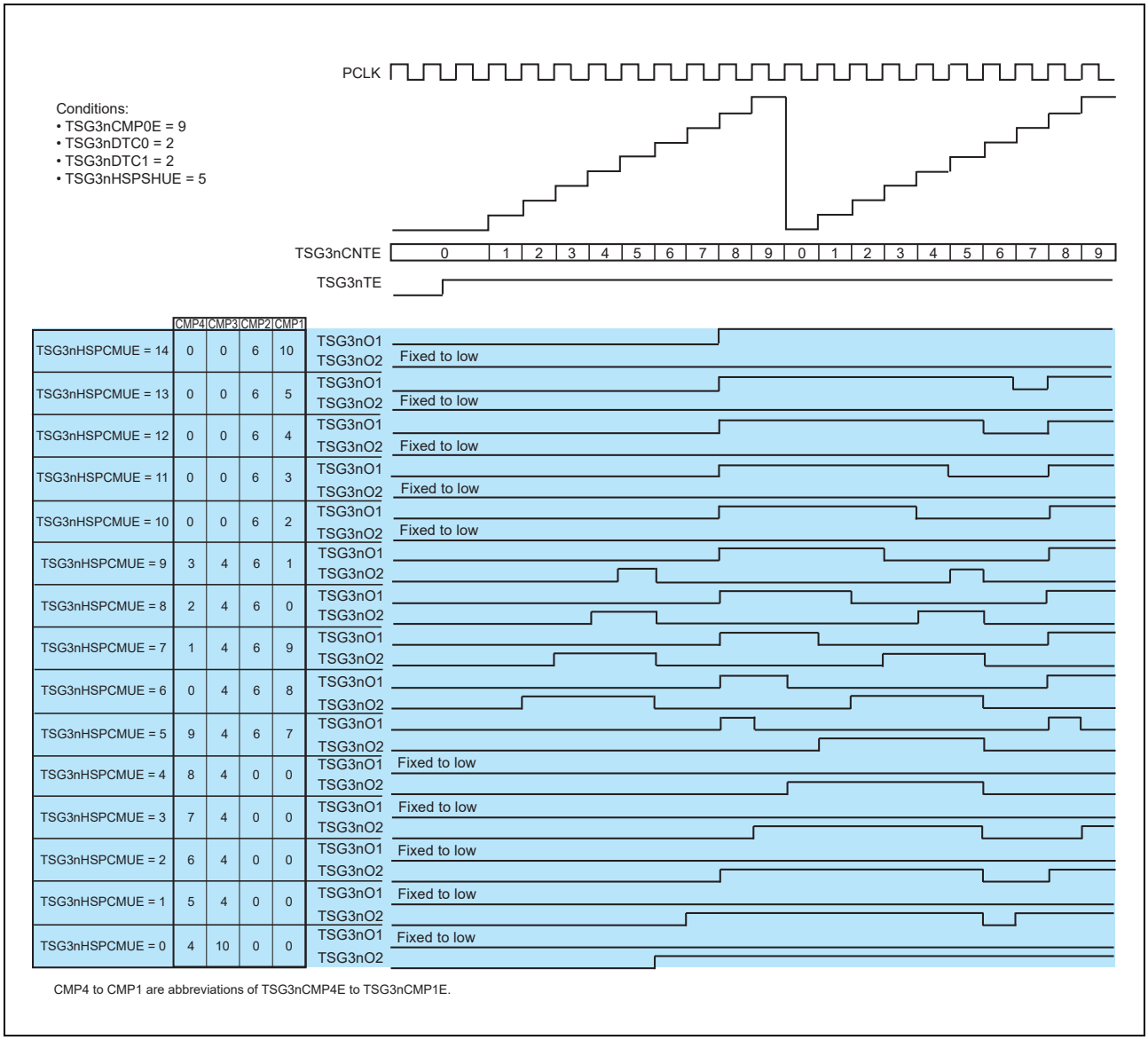


Figure 35.110 Timing Diagram of Operation Start in HSP-PWM Mode (TSG3nHSPSHUE = 5 (shift width is set to five clock cycles))

35.4.7.8 Software Output Control Function

Software output control function is available in any mode except HSP-PWM mode. This function can switch between six output patterns for the TSG3nO1 to TSG3nO6 pins using TSG3nOPT0.TSG3nSOC, TSG3nIDC, and TSG3nOPT0.TSG3nSPC2-0.

When TSG3nSOC is switched from 0 to 1, the output control method of the TSG3nO1 to TSG3nO6 pins is switched to the software output control immediately. On the other hand, when TSG3nSOC is switched from 1 to 0, the software output control is released at the reload timing.

Table 35.102 Registers Associated with Software Output Control Function

Register	Operation
TSG3nOPT0.TSG3nSOC	TSG3nSOC = 1
TSG3nOPT0.TSG3nSTE	TSG3nSTE = 0
TSG3nOPT1.TSG3nSPC2-TSG3nSPC0	Sets output patterns listed in the following **Table 35.103** and **Table 35.104**.
TSG3nOPT0.TSG3nIDC	Sets output pattern (electric current direction).

Table 35.103 Output Patterns by Software Output Control (TSG3nOPT0.TSG3nIDC = 0)

TSG3nOPT0.TSG3nSOC = 1, TSG3nSTE = 0, TSG3nIDC = 0

	TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output Pin	101	100	110	010	011	001	000	111
TSG3nO1	ACT	ACT	ACT	INACT	INACT	INACT	INACT	ACT
TSG3nO2	INACT	INACT	INACT	ACT	ACT	ACT	ACT	INACT
TSG3nO3	INACT	INACT	ACT	ACT	ACT	INACT	INACT	ACT
TSG3nO4	ACT	ACT	INACT	INACT	INACT	ACT	ACT	INACT
TSG3nO5	ACT	INACT	INACT	INACT	ACT	ACT	INACT	ACT
TSG3nO6	INACT	ACT	ACT	ACT	INACT	INACT	ACT	INACT

Note: ACT: Indicates active level is output.
INACT: Indicates inactive level is output.

Table 35.104 Output Patterns by Software Output Control (TSG3nOPT0.TSG3nIDC = 1)

TSG3nOPT0.TSG3nSOC = 1, TSG3nSTE = 0, TSG3nIDC = 1

	TSG3nSTR1.TSG3nOPF2-TSG3nOPF0							
Output Pin	101	100	110	010	011	001	000	111
TSG3nO1	INACT	INACT	INACT	ACT	ACT	ACT	ACT	INACT
TSG3nO2	ACT	ACT	ACT	INACT	INACT	INACT	INACT	ACT
TSG3nO3	ACT	ACT	INACT	INACT	INACT	ACT	ACT	INACT
TSG3nO4	INACT	INACT	ACT	ACT	ACT	INACT	INACT	ACT
TSG3nO5	INACT	ACT	ACT	ACT	INACT	INACT	ACT	INACT
TSG3nO6	ACT	INACT	INACT	INACT	ACT	ACT	INACT	ACT

Note: ACT: Indicates active level is output.
INACT: Indicates inactive level is output.

Section 36 Timer Option (TAPA)

This section contains a generic description of the Timer Option (TAPA).

The first part in this section describes the RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the TAPA.

36.1 Features TAPA for RH850/U2A-EVA

36.1.1 Number of Units

This microcontroller has the following number of TAPA units.

Table 36.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of units	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	4 (n = 0 to 3)	3 (n = 0, 2, 3)	2 (n = 0, 3)	2 (n = 0, 3)
Name	TAPAn									

Table 36.2 Index

Index	Description
n	Throughout this section, the individual TAPA units are identified by the index “n” (n = 0 to 3), for example, TAPAnFLG for the TAPAn flag register.

36.1.2 Register Base Addresses

TAPA register addresses are given as offsets from the given base address. Each of the TAPAn base addresses <TAPAn_base> is listed in the following table.

Table 36.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<TAPA0_base>	FFBF 9000$_H$	P-Bus Group 5
<TAPA1_base>	FFBF 9100$_H$	P-Bus Group 5
<TAPA2_base>	FFBF 9200$_H$	P-Bus Group 5
<TAPA3_base>	FFBF 9300$_H$	P-Bus Group 5

36.1.3 Clock Supply

Clock supply by and to TAPA is listed in the following table.

Table 36.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAPAn	PCLK	CLK_HSB

36.1.4 Interrupt Requests and Error Notifications

TAPAn interrupt requests are listed in the following table.

Table 36.5 Interrupt Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
TAPA0				
INTTAPA0IPEK0	Peak interrupt 0	27	—	—
INTTAPA0IVLY0	Valley interrupt 0	28	—	—

This module has no error notifications.

36.1.5 Reset Sources

TAPAn reset sources are listed in the following table. TAPAn is initialized by these reset sources.

Table 36.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
TAPAn	All registers	√	√	√	√	√	√	—

36.1.6 Internal Input/Output Signal

The internal input/output signals of TAPAn are listed below.

Table 36.7 Internal Input/Output Signals

TAPAn Signal	Function	Connected to
TAPA0		
TAPA0THASIN	TAPA0 asynchronous Hi-Z control signal*1	PIC1
TAPA0TSIM0	TAUD master channel interrupt signal (TAUD0: INTTAUD0I0,I NTTAUD0I2, INTTAUD0I8)	PIC1
TAPA0TUDCM0	TAUD master up/down signal (TAUD0: TAUD0UDC0, TAUD0UDC2, TAUD0UDC8)	PIC1
TAPA0TCDENS0	TAUD slave 0 match detect	PIC2
TAPA0TCDENS1	TAUD slave 1 match detect	PIC2
$\overline{\text{TAPA0THZOUT0}}$	TAPA0UP/TAPA0UN output buffers Hi-Z control output	Port
$\overline{\text{TAPA0THZOUT1}}$	TAPA0VP/TAPA0VN output buffers Hi-Z control output	Port
$\overline{\text{TAPA0THZOUT2}}$	TAPA0WP/TAPA0WN output buffers Hi-Z control output	Port
TAPA0TADOUT0	A/D trigger signal 0 output	PIC2
TAPA0TADOUT1	A/D trigger signal 1 output	PIC2
TAPA1		
TAPA1THASIN	TAPA1 asynchronous Hi-Z control signal	PIC1
$\overline{\text{TAPA1THZOUT0}}$	TAUD1O10, TAUD1O11 output buffers Hi-Z control output	Port
$\overline{\text{TAPA1THZOUT1}}$	TAUD1O12, TAUD1O13 output buffers Hi-Z control output	Port
$\overline{\text{TAPA1THZOUT2}}$	TAUD1O14, TAUD1O15 output buffers Hi-Z control output	Port
TAPA2		
TAPA2THASIN	TAPA2 asynchronous Hi-Z control signal	PIC1
$\overline{\text{TAPA2THZOUT0}}$	TSG30O1-TSG30O6 output buffers Hi-Z control output	Port
TAPA3		
TAPA3THASIN	TAPA3 asynchronous Hi-Z control signal	PIC1
$\overline{\text{TAPA3THZOUT0}}$	TSG31O-TSG31O6 output buffers Hi-Z control output	Port

Note 1. See **Section 41.2.3.13, Hi-Z Control Function** for details.

36.1.7 Peripheral Configuration

The following figure shows the peripheral configuration of TAPA.

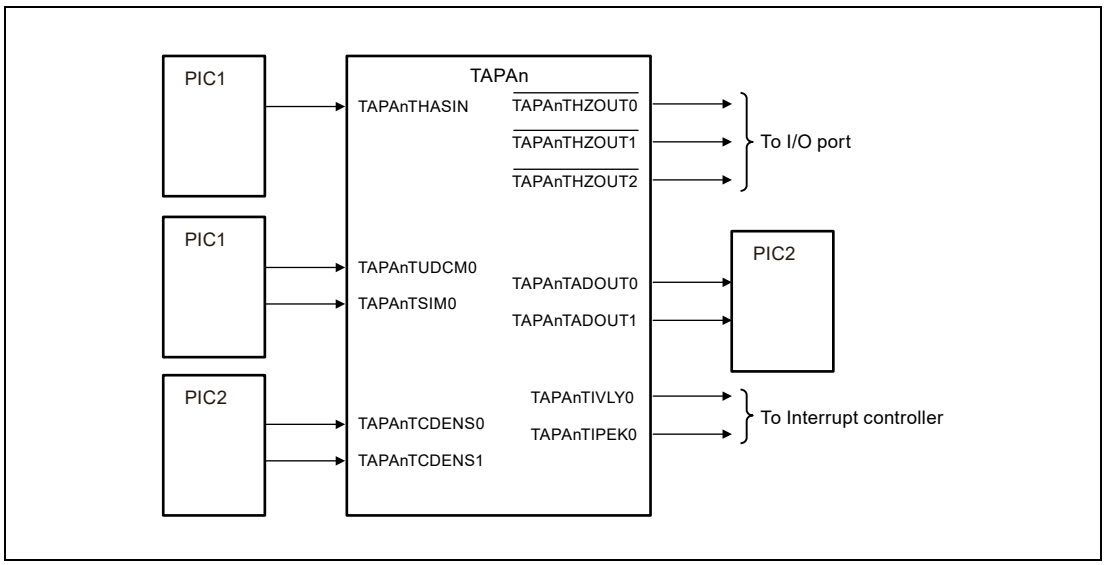


Figure 36.1 Peripheral Configuration of TAPA

The following describes the peripheral configuration of TAPA.

- TAPAnTHASIN: Hi-Z control asynchronous input signal

  This signal controls Hi-Z by the source selected in PIC1.

  For the sources that can be selected in PIC1, see **Section 41.2.3.13, Hi-Z Control Function**.

- TAPAnTUDCM0: TAUD master channel up/down input

- TAPAnTSIM0: TAUD master 0 INT input

  Peak and valley interrupts can be generated on TAUDn channels selected in PIC1.

  For TAUDn channels that can be selected in PIC1, see **Section 41.2.2.18, PIC1REG200 — Timer Input/Output Control Register 200** and **Section 41.2.2.19, PIC1REG210 — Timer Input/Output Control Register 210**.
  Only TAPA0 has this connection.

- TAPAnTCDENS0, TAPAnTCDENS1: TAUD slave channel match detection input

  The TAUDn channel interrupts selected in PIC2 can be handled as A/D conversion trigger outputs 0 and 1 (TAPAnTADOUT1 and TAPAnTADOUT0). For the TAUDn channel interrupts that can be selected in PIC, see **Section 36.4.3, A/D Converter Conversion Trigger Selection Function**.
  Only TAPA0 has this connection.

- TAPAnTADOUT1-0: A/D conversion trigger outputs 1 and 0

  The A/D conversion trigger signals generated on TAPAn are outputs to PIC2. The setting of the PIC2 can also be used to set these signals as triggers for A/D conversion. For the register specifications in the PIC2, see **Section 41.3.3.1, ADCJ Trigger Select Function**.
  Only TAPA0 has this connection.

36.2 Overview

36.2.1 Functional Overview

- Asynchronous Hi-Z control function
  Hi-Z control for TAUDn (n = 0, 1), TSG3n (n = 0, 1) output can be performed by using pin input or error signals.
- Interrupt signals output function
  Request signals for two types of interrupts, peak interrupts and valley interrupts, can be output by the INTn signals output by TAUDn (n = 0).
- A/D conversion start trigger selection function
  An A/D conversion start trigger can be output by the INTn signals output by TAUDn (n = 0).

36.2.2 Block Diagram

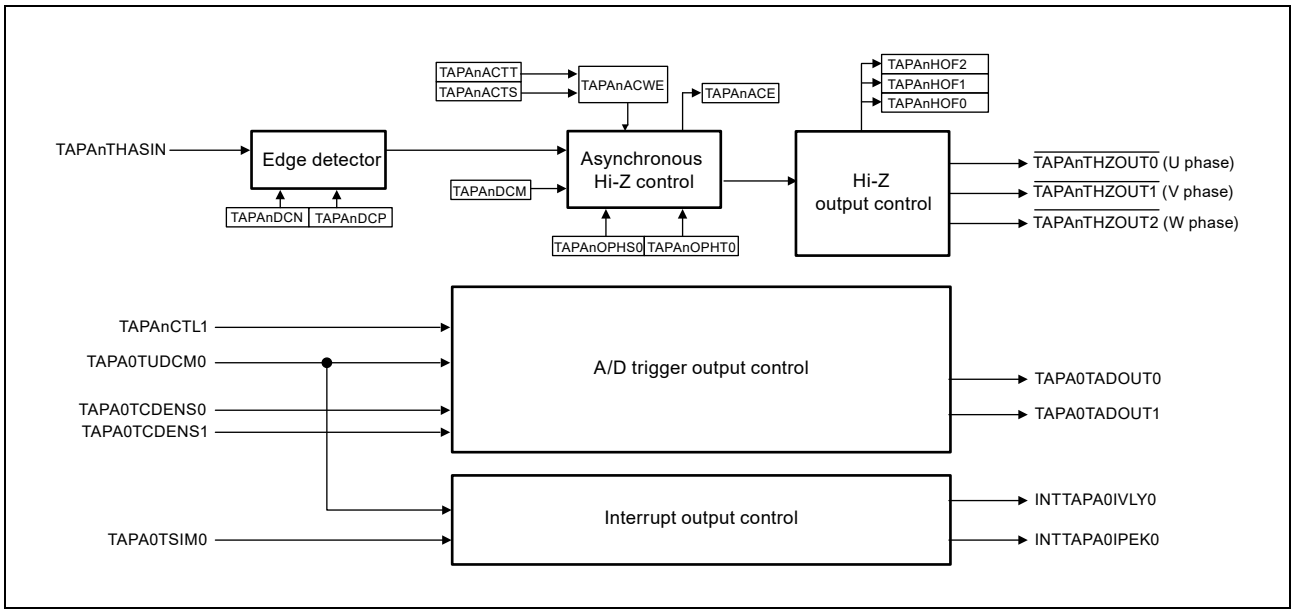


Figure 36.2 Block Diagram

36.2.3 Definition of Terms

Peak and valley interrupts - Peak and valley of timer counter.

In this document, the period from a TAUD counting-up status to generation of INT from the master channel is defined as a peak period, and this INT is defined as a peak interrupt.

In contrast, the period from a TAUD counting-down status to generation of INT from the master channel is defined as a valley period, and this INT is defined as a valley interrupt.

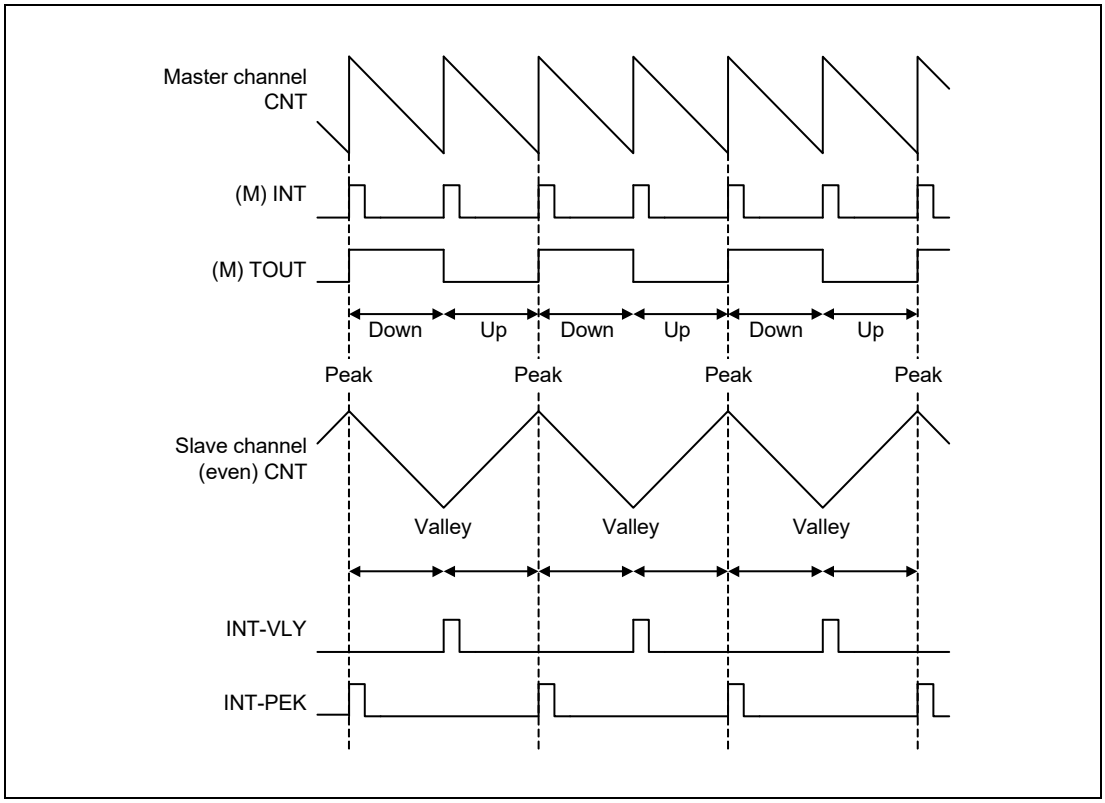


Figure 36.3 Peak and Valley Interrupts

36.3 Registers

36.3.1 List of Registers

TAPAn (n = 0 to 3) registers are listed in the following table.

<TAPAn_base> is defined in **Section 36.1.2, Register Base Addresses**.

Table 36.8 List of Registers

Module	Register Name	Symbol	Address	Access Protection	
				PBG	Other
TAPAn	TAPAn control register 0	TAPAnCTL0	<TAPAn_base> + 20_H	*2	—
TAPAn	TAPAn control register 1	TAPAnCTL1*1	<TAPAn_base> + 24_H	*2	—
TAPAn	TAPAn flag register	TAPAnFLG	<TAPAn_base> + 00_H	*2	—
TAPAn	TAPAn asynchronous control write enable register	TAPAnACWE	<TAPAn_base> + 04_H	*2	—
TAPAn	TAPAn asynchronous control start trigger register	TAPAnACTS	<TAPAn_base> + 08_H	*2	—
TAPAn	TAPAn asynchronous control stop trigger register	TAPAnACTT	<TAPAn_base> + $0C_H$	*2	—
TAPAn	TAPAn Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base> + 14_H	*2	—
TAPAn	TAPAn Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base> + 18_H	*2	—

Note 1. TAPAnCTL1 is only available in TAPA0 (n = 0 only).

Note 2. n = 0: PBG51#10
n = 1: PBG51#11
n = 2: PBG51#12
n = 3: PBG51#13

36.3.2 TAPAnCTL0 — TAPAn Control Register 0

- This register is used to set up the asynchronous Hi-Z control function.
- The values of this register can be rewritten only when TAPAnFLG.TAPAnACE is 0 and TAUDnTEm for the corresponding TAUD's master channel is 0.

Access This register can be read or written in 16-bit units.

Address <TAPAn_base> + 20_H

Value after reset 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TAPAnDCM	TAPAnDCN	TAPAnDCP	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 36.9 TAPAnCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TAPAnDCM	Clear Condition Configuration This control bit specifies the clear conditions for Hi-Z control output. 0: Enables manipulation of TAPAnOPHT0 regardless of the TAPAnTHASIN signal input level. 1: Enables manipulation of TAPAnOPHT0 only if the TAPAnTHASIN signal input is inactive.
3, 2	TAPAnDCN, TAPAnDCP	Hi-Z Input Edge Selection These control bits specify the valid edge of TAPAnTHASIN.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

TAPAn DCN	TAPAn DCP	Description
0	0	Does not detect valid edges
0	1	Detects a rising edge as the valid edge (active level = high)
1	0	Detects a falling edge as the valid edge (active level = low)
1	1	Setting is prohibited.

36.3.3 TAPAnCTL1 — TAPAn Control Register 1

This register is used to set up the asynchronous Hi-Z control function.

Access This register can be read or written in 8-bit units.

Address <TAPAn_base> + 24_H

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAPAnATS3	TAPAnATS2	TAPAnATS1	TAPAnATS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 36.10 TAPAnCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	TAPAnATS3, TAPAnATS2	A/D Conversion Trigger 1 Selection These are control bits that specify A/D conversion trigger output 1 (TAPA0TADOUT1).

TAPAnATS3	TAPAnATS2	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 signal

Bit Position	Bit Name	Function
1, 0	TAPAnATS1, TAPAnATS0	A/D Conversion Trigger 0 Selection These are control bits that specify A/D conversion trigger output 0 (TAPA0TADOUT0).

TAPAnATS1	TAPAnATS0	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt INTTAPA0IVLY0 signal while the triangle wave is rising (counting up) or falling (counting down)

36.3.4 TAPAnFLG — TAPAn Flag Register

This flag register is for asynchronous Hi-Z control.

Access This register is a read-only register that can be read in 16-bit units.

Address <TAPAn_base> + 00_H

Value after reset 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TAPAnHOF2	TAPAnHOF1	TAPAnHOF0	—	—	—	—	—	—	—	TAPAnACE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.11 TAPAnFLG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned.
10	TAPAnHOF2	W phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of $\overline{\text{TAPAnTHZOUT2}}$ is high level. 1: The present output of $\overline{\text{TAPAnTHZOUT2}}$ is low level.
9	TAPAnHOF1	V phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of $\overline{\text{TAPAnTHZOUT1}}$ is high level. 1: The present output of $\overline{\text{TAPAnTHZOUT1}}$ is low level.
8	TAPAnHOF0	U phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of $\overline{\text{TAPAnTHZOUT0}}$ is high level. 1: The present output of $\overline{\text{TAPAnTHZOUT0}}$ is low level.
7 to 1	Reserved	When read, the value after reset is returned.
0	TAPAnACE	Asynchronous Hi-Z Control Enable 0: Indicates that the asynchronous Hi-Z control is stopped. 1: Indicates that the asynchronous Hi-Z control is enabled. The conditions for setting or clearing this bit are as follows: Clear condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1 Set condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1

Note: TAPAnHOFm (m = 1, 2) is enabled when TAPAn (n = 0, 1).

36.3.5 TAPAnACWE — TAPAn Asynchronous Control Write Enable Register

This register is used to enable writing for asynchronous Hi-Z control.

Access This register can be read or written in 8-bit units.

Address <TAPAn_base> + 04_H

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACWE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 36.12 TAPAnACWE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TAPAnACWE	Asynchronous Control Write Enable This is a write-enable bit for asynchronous Hi-Z control. After 1 is written, this bit is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

36.3.6 TAPAnACTS — TAPAn Asynchronous Hi-Z Control Start Trigger Register

This register is used to enable the start trigger for asynchronous Hi-Z control.

Access This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 08_H

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.13 TAPAnACTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTS	Asynchronous Hi-Z Control Start Trigger This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Enables asynchronous Hi-Z control if TAPAnACWE = 1.

36.3.7 TAPAnACTT — TAPAn Asynchronous Hi-Z Control Stop Trigger Register

This bit enables the stop trigger for asynchronous Hi-Z control.

Access This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + $0C_H$

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnACTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.14 TAPAnACTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTT	Asynchronous Hi-Z Control Stop Trigger This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Disables asynchronous Hi-Z control if TAPAnACWE = 1.

36.3.8 TAPAnOPHS — TAPAn Hi-Z Start Trigger Register

This software trigger register is used to start Hi-Z control for motor control output pins.

Access This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 14_H

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.15 TAPAnOPHS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHS0	Hi-Z Control Start Trigger This bit starts Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Starts Hi-Z control.

36.3.9 TAPAnOPHT — TAPAn Hi-Z Stop Trigger Register

This software trigger register is used to stop Hi-Z control for motor control output pins.

Access This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base> + 18_H

Value after reset 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.16 TAPAnOPHT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHT0	Hi-Z Control Stop Trigger This bit stops Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Stops Hi-Z control. Whether the setting of this bit is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

36.4 Operation

36.4.1 Asynchronous Hi-Z Control Function

Abnormal operation in timer motor-control under CPU control leads to rotation of the externally connected motor also becoming abnormal. In such cases, this function forcibly sets the motor control output to the Hi-Z state, independently of control by the CPU.

36.4.1.1 Overview

The following method is available for controlling Hi-Z.

- Asynchronous input Hi-Z control by Hi-Z control asynchronous input signal (TAPAnTHASIN)
  - Controls the Hi-Z control output signals $\overline{\text{TAPAnTHZOUT0}}$ (U phase), $\overline{\text{TAPAnTHZOUT1}}$ (V phase), $\overline{\text{TAPAnTHZOUT2}}$ (W phase) asynchronously at TAPAn (n = 0, 1).
  - Controls the Hi-Z control output signals $\overline{\text{TAPAnTHZOUT0}}$ asynchronously at TAPAn (n = 2, 3).
- Motor control output in a Hi-Z state can be resumed by writing the Hi-Z stop trigger register (TAPAnOPHT0).
- The Hi-Z state of motor control output can also be specified by writing the Hi-Z control start trigger register (TAPAnOPHS).
- Setting PIC can enable or disable Hi-Z control input when an error occurs.

Table 36.17 Asynchronous Hi-Z Control and Its Operation

Function	Operation
Asynchronous Hi-Z control corresponding to the pin input	This function detects asynchronous pin inputs and forcibly stops TOUTn output from the corresponding timer module (TAUD, TSG3) in response. Device port outputs become Hi-Z while TAPAnTHASIN is active and until software sends a stop request.

36.4.2 System Configuration Example

A system configuration example is shown below, where an external error detection signal (the TAPAnESO signal, n = 0, 1) is used for Hi-Z control of the motor control outputs (TAPA0UP/ TAPA0UN/TAPA0VP/TAPA0VN/TAPA0WP/TAPA0WN, TAUD1O10/TAUD1O11/TAUD1O12/ TAUD1O13/TAUD1O14/TAUD1O15, TSG30O1/TSG30O2/TSG30O3/TSG30O4/TSG30O5/ TSG30O6, TSG31O1/TSG31O2/TSG31O3/TSG31O4/TSG31O5/TSG31O6).

When valid edges of the external error detection signal are detected, the level of the motor control outputs is set to Hi-Z.

Because the microcontroller might freeze when an error occurs, external error detection signals are continuously processed so that the motor control timer outputs can be set to Hi-Z even if no clock is supplied.

Note that an error is detected only when the valid edge of the error detection signal is detected. Therefore, no error is detected if the output level is fixed and the signal level does not change.

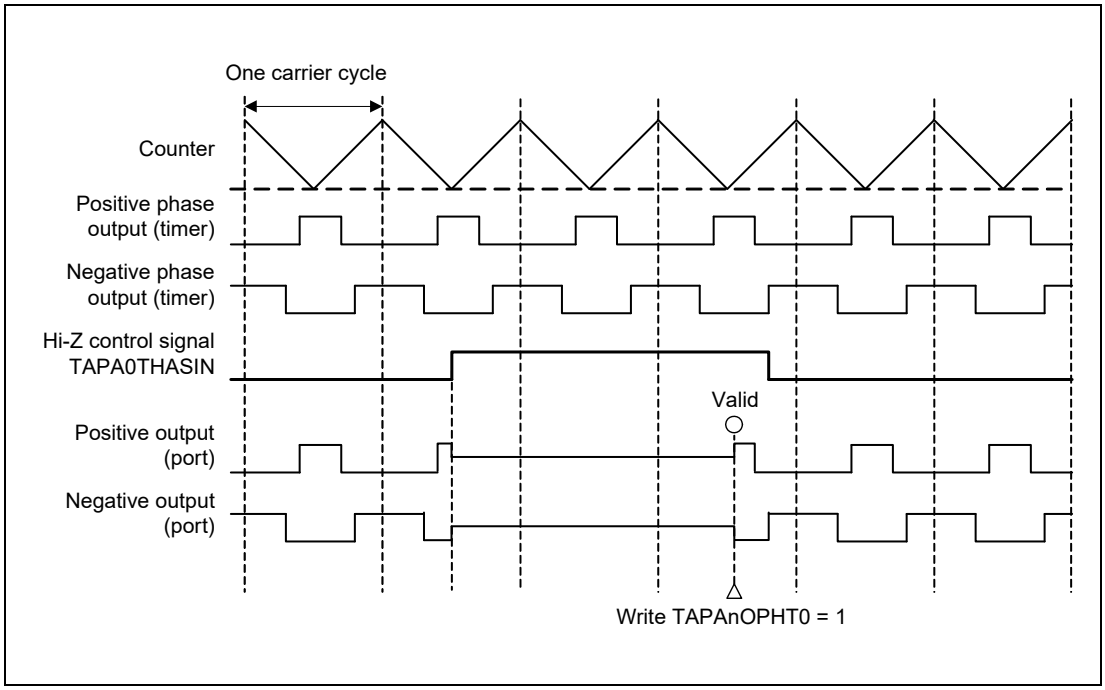
36.4.2.1 Basic Operation

Hi-Z control for motor control output pins can be started as follows:

- Detecting the valid edge of asynchronous Hi-Z control signal (TAPAnTHASIN)
- Setting the start trigger bit TAPAnOPHS.TAPAnOPHS0 of the Hi-Z control signal

The levels of the motor control output pins are set to Hi-Z until the stop trigger bit of the Hi-Z control signal (TAPAnOPHT.TAPAnOPHT0) is set. Note that whether the setting of TAPAnOPTH0 is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

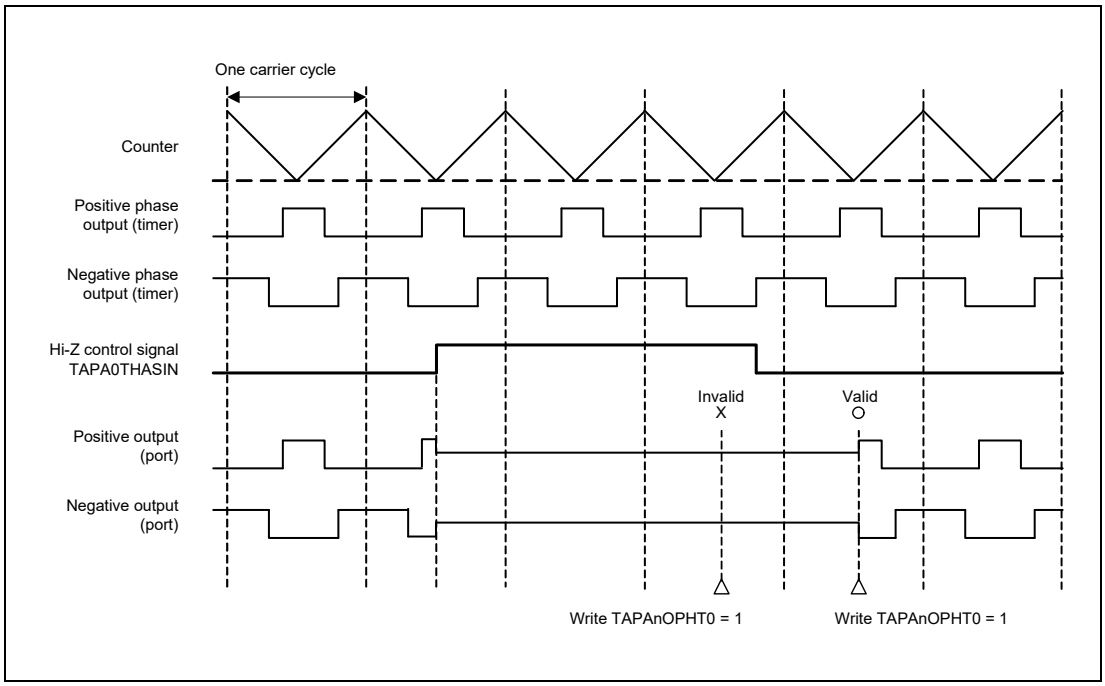
(1) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPAnTHASIN is detected.

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0, regardless of the level of TAPAnTHASIN.

(2) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPAnTHASIN is detected.

Writing 1 to the stop trigger bit (TAPAnOPHT.TAPAnOPHT0) of the Hi-Z control signal is ignored while TAPAnTHASIN is active (high level because TAPAnCTL0.TAPAnDCP is 1).

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0 after TAPAnTHASIN becomes inactive (low level because TAPAnCTL0.TAPAnDCP is 1).

36.4.2.2 Asynchronous Hi-Z Control Using Software Trigger

Hi-Z control for motor control output is possible by using the Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0 and Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0.

(1) Function of Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0

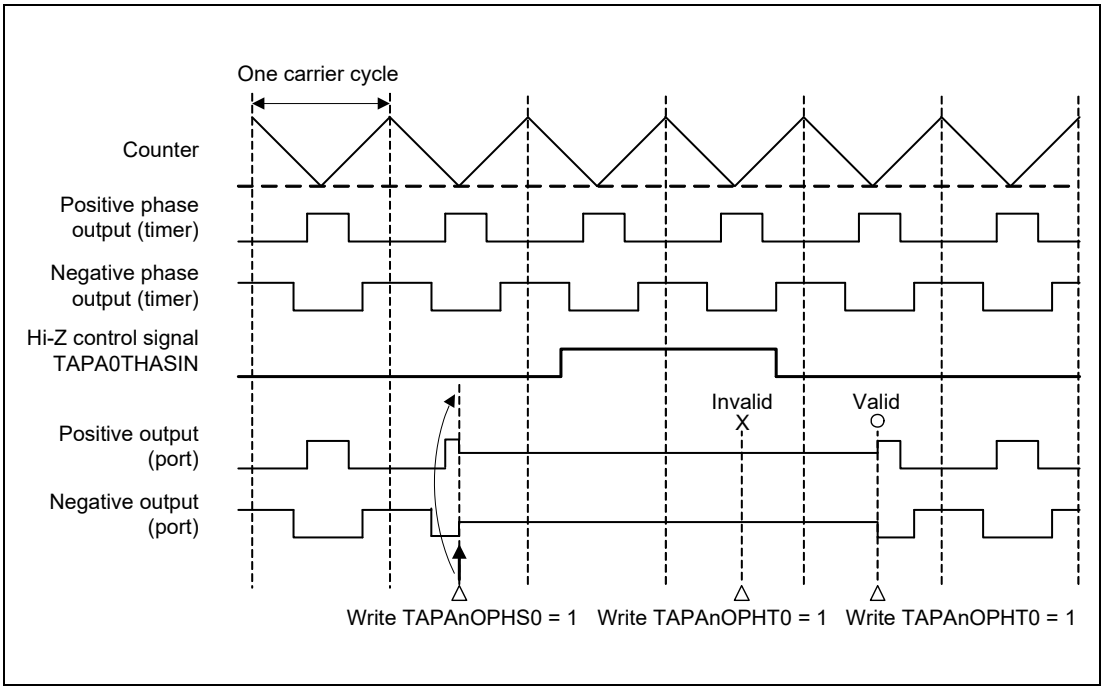
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the motor control output (Hi-Z output).

(2) Function of Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0

Whether the Hi-Z control stop trigger is valid or invalid depends on the conditions below:

TAPAnDCM	Function
0	Writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output.
1	If TAPAnTHASIN is inactive, writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output. If TAPAnTHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

(3) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control output (Hi-Z output) is forcibly stopped when 1 is written to TAPAnOPHS0.

After that, the level of the motor control output remains Hi-Z even if the rising edge of TAPAnTHASIN is detected.

Writing to TAPAnOPHT0 is ignored while TAPA0THASIN is active (high level because TAPAnDCN is 0 and TAPAnDCP is 1).

After detection of the falling edge of TAPA0THASIN, the motor control output restarts when 1 is written to TAPAnOPHT0 while TAPA0THASIN is inactive (low level because TAPAnDCN is 0 and TAPAnDCP is 1).

36.4.2.3 Operating Procedure

The operating procedure for the asynchronous input Hi-Z control function is shown below:

	Operation	Status of TAPA
Initial setup	Set up the TAPAnCTL0 register. Specify TAPAnDCP and TAPAnDCN to select the input edge. Specify TAPAnDCM to select the clear mode.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)
Start operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTS register. Set TAPAnACTS to 1.	Writing to TAPAnACTS is enabled. Asynchronous Hi-Z control enabled (TAPAnFLG.TAPAnACE = 1)
During operation	Hi-Z control for the timer function outputs can be started by controlling the following: • TAPAnOPHS register • Asynchronous Hi-Z control signal (TAPA0THASIN) Hi-Z control for the timer function outputs can be stopped by controlling the following: • TAPAnOPHT register (If TAPAnDCM is 1, control by the TAPAnOPHT register is enabled only while TAPA0THASIN is inactive.) The TAPA operating status can always be read using the TAPAnFLG register.	Hi-Z control for the motor control output pins is started by detecting the valid edge of the asynchronous Hi-Z control signal (TAPA0THASIN) or by setting the Hi-Z control start trigger bit TAPAnOPHS0 to 1. Hi-Z control for the motor control output pins is stopped by setting the Hi-Z control stop trigger bit TAPAnOPHT0 to 1 according to the operation mode specified by the TAPAnDCM bit.
Stop operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTT register. Set TAPAnACTT to 1.	Writing to TAPAnACTT is enabled. Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

Restart

36.4.3 A/D Converter Conversion Trigger Selection Function

This function outputs the A/D converter conversion trigger signals TAPA0TADOUT0 and TAPA0TADOUT1 from the TAPA0TCDENS0 and TAPA0TCDENS1 signals, which are connected to a compare match interrupt based on the triangular carrier cycle of TAUD, or a valley interrupt signal (INTTAPA0IVLY0).

36.4.3.1 Configuration of A/D Converter Conversion Trigger Selection Function

Table 36.18 Signals Used for TAPA0TADOUT Generation

Output Signal	Slave Match Detection Signal	Valley Interrupt Signal
TAPA0TADOUT0	TAPA0TCDENS0	INTTAPA0IVLY0
TAPA0TADOUT1	TAPA0TCDENS1	INTTAPA0IVLY0

Table 36.19 Operation of TAPA0TADOUT1 According to the Setting of TAPAnCTL1.TAPAnATS[3:2]

TAPAnATS3	TAPAnATS2	Description
0	0	Outputs the INT signal from TAPA0TADOUT1 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPA0TADOUT1 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPA0TADOUT1 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 from TAPA0TADOUT1.

Table 36.20 Operation of TAPA0TADOUT0 According to the Setting of TAPAnCTL1.TAPAnATS[1:0]

TAPAnATS1	TAPAnATS0	Description
0	0	Outputs the INT signal from TAPA0TADOUT0 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPA0TADOUT0 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPA0TADOUT0 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt INTTAPA0IVLY0 from TAPA0TADOUT0.

36.4.3.2 Block Diagram

The interrupt request signal of each channel of TAUD0 is masked and OR'ed, and output to TAPA0 as the TAUD0 trigger signal.

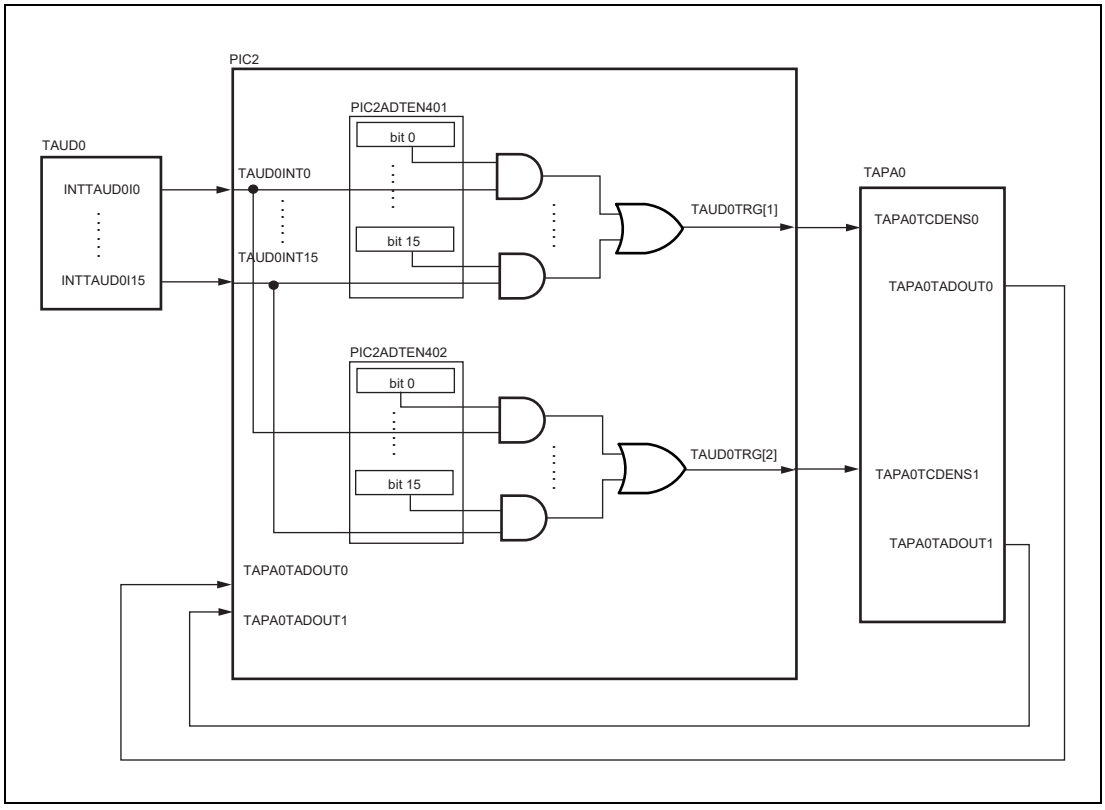


Figure 36.4 Block Diagram of A/D Conversion Trigger Selection Function

CAUTION

This function is only available for the scan groups 1 and 2 of ADCJ0-1. When using this function with the scan groups 1 and 2, select the TAUD trigger via TAPA. Do not select the TAUD trigger directly by the ADCJ trigger select function.

36.4.3.3 Waveforms of A/D Converter Conversion Trigger Output Control Operation in Triangle PWM Mode

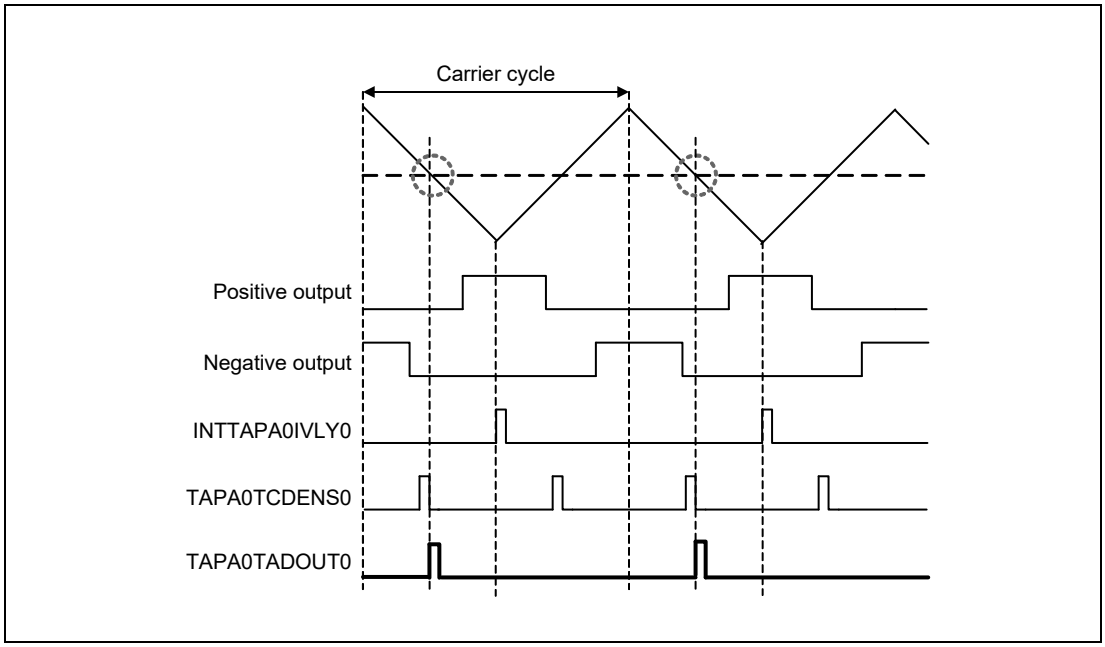


Figure 36.5 TAPAnATS[1:0] bits = 00_B: Output INT Signal while the Triangle Wave is Falling (Counting Down)

While the triangle wave is falling (counting down), the signals TAPA0TCDENS0 and TAPA0TCDENS1 are output as the A/D converter conversion trigger signals TAPA0TADOUT0 and TAPA0TADOUT1.

In this case, no A/D converter conversion trigger signal is output while the triangle wave is rising (counting up).

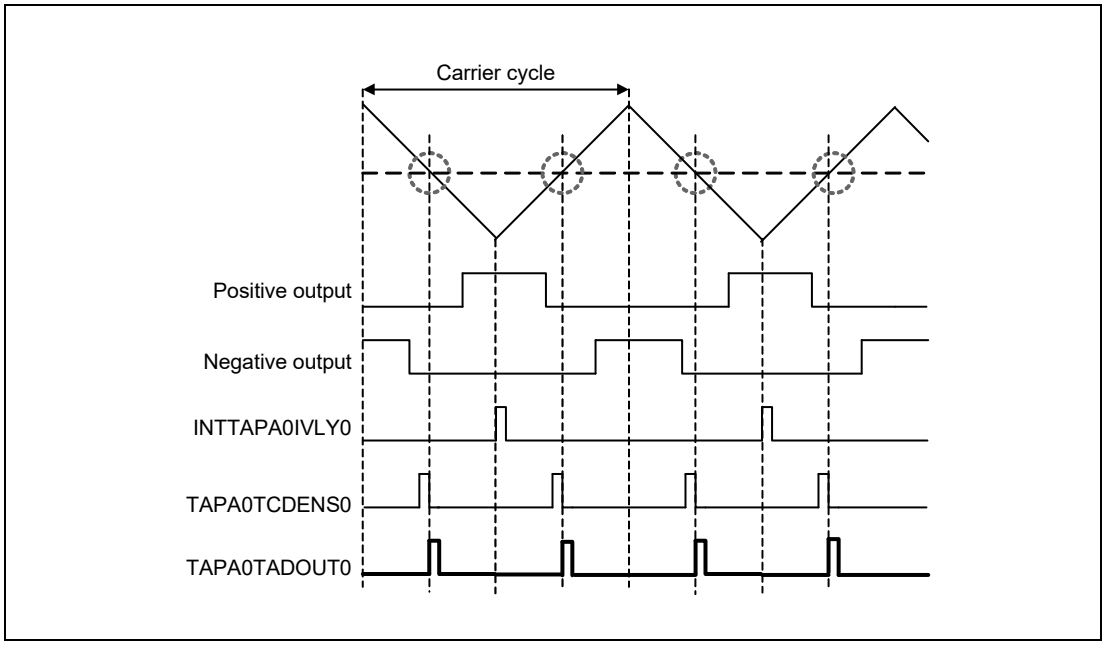


Figure 36.6 TAPAnATS[1:0] bits = 10_B: Output INT Signal while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPA0TCDENS0 and TAPA0TCDENS1 are output as the A/D converter conversion trigger signals TAPA0TADOUT0 and TAPA0TADOUT1.

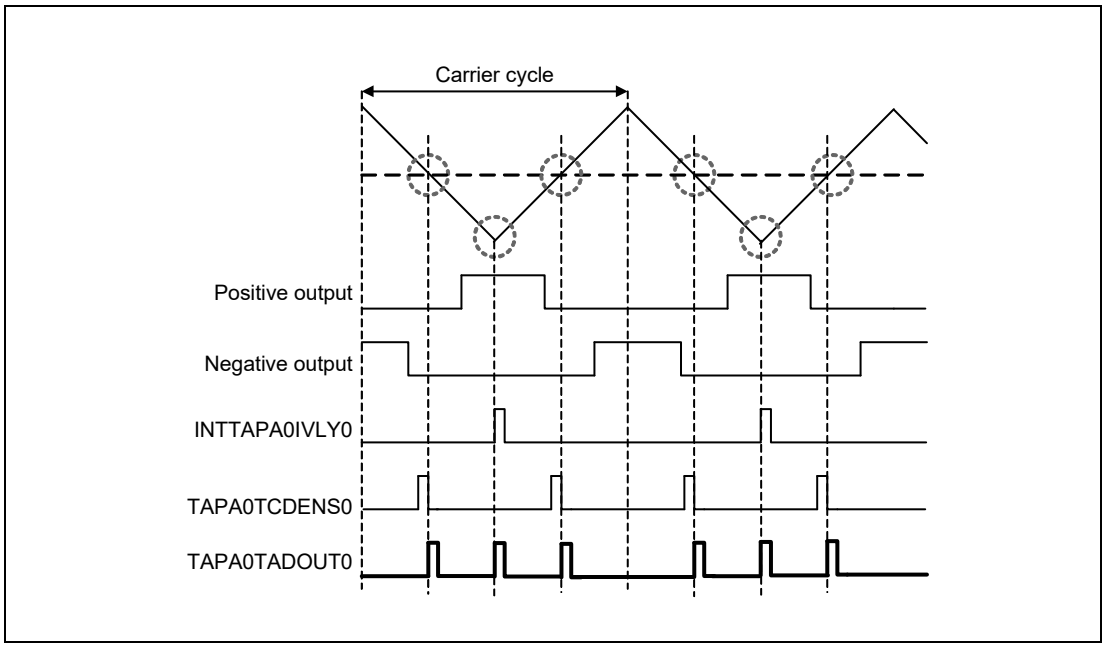


Figure 36.7 TAPAnATS[1:0] bits = 11B: Output of INT Signal and Valley Interrupt while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPA0TCDENS0 and TAPA0TCDENS1 and valley interrupt INTTAPA0IVLY0 are output as the A/D converter conversion trigger signals TAPA0TADOUT0 and TAPA0TADOUT1.

36.4.3.4 Operating Procedure for A/D Converter Conversion Trigger Selection Function

The operating procedure for the A/D converter conversion trigger selection function is shown below.

	Operation	Status of TAUD and TAPA
Initial setup	Initialize TAUD. Specify the timer operation mode. Set up the TAPAnCTL1 register. Specify TAPAnATS[1:0] (TAPA0TADOUT0 setting). Specify TAPAnATS[3:2] (TAPA0TADOUT1 setting). Set up the PIC2ADTEN4nj and PIC0REG2n0 registers according to the signal to be used. Specify PIC2ADTEN4nj (TAPA0TCDENS0 or TAPA0TCDENS1 setting). Specify PIC0REG2n0 (INTTAPA0IVLY0 setting).	TAUD and TAPA stop the operation.
Start operation	Start the TAUD operation.	TAUD starts the count operation.
During operation	TAUD operates according to the setting of each function.	The A/D conversion trigger selection function outputs either TAPA0TADOUT0 according to the setting of TAPAnATS[1:0] or TAPA0TADOUT1 according to the setting of TAPAnATS[3:2], based on the interrupt TAPA0TCDENS1 or TAPA0TCDENS0, which is input from TAUD, and the valley interrupt INTTAPA0IVLY0, which is generated by TAPA.
Stop operation	Stop the TAUD operation.	TAUD stops the count operation.

Restart

Section 37 Timer Pattern Buffer (TPBA)

The Timer Pattern Buffer (TPBA) is 16-bit PWM timer with the duty setting buffer. TPBA is implemented 2 units.

37.1 Features of TPBA for RH850/U2A-EVA

37.1.1 Units and Channels

This microcontroller has the following number of TPBA units.

Each TPBA unit has one channel interface.

Table 37.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of units	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)
Name	TPBAn									

Table 37.2 Index

Index	Description
n	Throughout this section, the individual TPBA units are identified by the index “n” (n = 0, 1); for example, TPBAnCTL is the TPBAn control register.
m	The number of buffers are indicated by the index “m” (m = 00 to 63).

37.1.2 Register Base Addresses

TPBA base addresses are listed in the following table.

TPBA register addresses are given as offsets from the base addresses in general.

Table 37.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<TPBA0_base>	FFBF A000$_H$	P-Bus Group 5
<TPBA1_base>	FFBF A800$_H$	P-Bus Group 5

37.1.3 Clock Supply

Clock supply by and to TPBA is listed in the following table.

Table 37.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
TPBAn	PCLK	CLK_HSB	Peripheral high speed clock

37.1.4 Interrupt Requests and Error Notifications

TPBA interrupt requests are listed in the following table.

Table 37.5 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
TPBA0				
INTTPBA0IPRD	TPBA0 Period match detection interrupt	528	Group0-167	Group3-92
INTTPBA0IDTY	TPBA0 Duty match detection interrupt	529	Group0-168	Group3-93
INTTPBA0IPAT	TPBA0 Pattern number matching detection interrupt	530	Group0-169	Group3-94
TPBA1				
INTTPBA1IPRD	TPBA1 Period match detection interrupt	531	Group0-170	Group3-95
INTTPBA1IDTY	TPBA1 Duty match detection interrupt	532	Group0-171	Group3-96
INTTPBA1IPAT	TPBA1 Pattern number matching detection interrupt	533	Group0-172	Group3-97

This module has no error notifications.

37.1.5 Reset Sources

TPBA reset sources are listed in the following table. TPBAn is initialized by these reset sources.

Table 37.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
TPBAn	All registers	√	√	√	√	√	√	—

37.1.6 External Input/Output Signals

External input/output signals of TPBA are listed in the following table.

Table 37.7 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
TPBA0			
TPBA0O	O	Timer output	TPBA0O
TPBA1			
TPBA1O	O	Timer output	TPBA1O

37.2 Overview

37.2.1 Functional Overview

TPBAn is a 16-bit PWM timer with the duty setting buffer.

- Count clock resolution: Min. 12.5 ns (count clock: 80 MHz)
- 16-bit counter
- 16-bit duty register
- 16-bit period setting register
- 7-bit address counter register
- 7-bit pattern number setting register
- Interrupt request signals
  - Period-matched detection interrupt
  - Duty-cycle-matched detection interrupt
  - Number-of-patterns matched detection interrupt
- Number of duty patterns
  - 64 patterns (16 bits) or 128 patterns (8 bits)
- Automatic duty generation according to the number of patterns
- Output control by software
- The count clock can be selected from PCLK, PCLK/2, PCLK/4, and PCLK/8 according to the prescaler set value.
- Synchronous start with another timer

37.2.2 Block Diagram

The following block diagram shows the main components of TPBA.

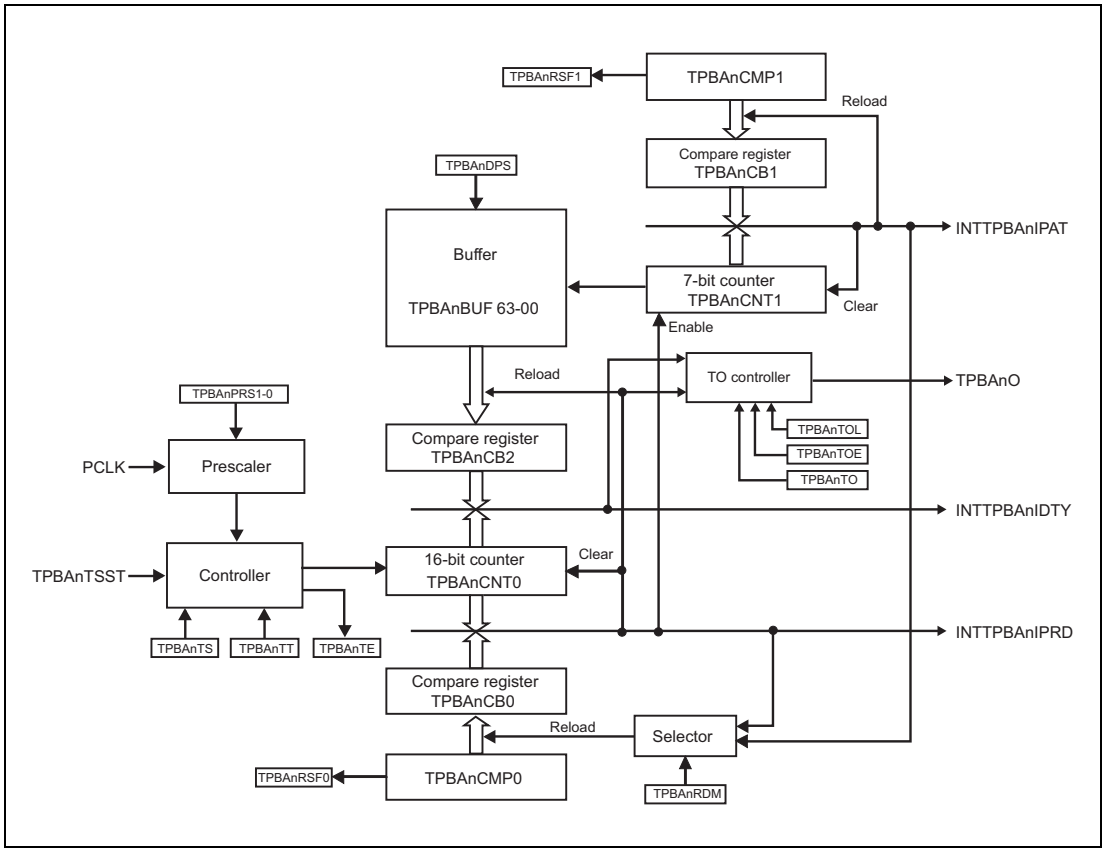


Figure 37.1 Block Diagram of TPBA

- TPBAnTSST Simultaneous start trigger (input by PIC function)

37.3 Registers

37.3.1 List of Registers

TPBA registers are listed in the following table.

For information on <TPBAn_base>, see **Section 37.1.2, Register Base Addresses**.

Table 37.8 List of Registers

Module Name	Register Name	Symbol	Address	Access size	Access Protection	
					PBG	Other
TPBAn	TPBAn control register	TPBAnCTL	<TPBAn_base> + 200_H	8	*1	—
	TPBAn reload data mode register	TPBAnRDM	<TPBAn_base> + 118_H	8	*1	—
	TPBAn reload status register	TPBAnRSF	<TPBAn_base> + 110_H	8	*1	—
	TPBAn reload data trigger register	TPBAnRDT	<TPBAn_base> + 114_H	8	*1	—
	TPBAn timer output enable register	TPBAnTOE	<TPBAn_base> + 120_H	8	*1	—
	TPBAn timer output register	TPBAnTO	<TPBAn_base> + $11C_H$	8	*1	—
	TPBAn timer output level register	TPBAnTOL	<TPBAn_base> + 124_H	8	*1	—
	TPBAn period setting register	TPBAnCMP0	<TPBAn_base> + 100_H	16	*1	—
	TPBAn duty setting register	TPBAnBUFm	<TPBAn_base> + m × 4_H	16	*1	—
	TPBAn pattern number setting register	TPBAnCMP1	<TPBAn_base> + 104_H	8	*1	—
	TPBAn timer counter register	TPBAnCNT0	<TPBAn_base> + 108_H	16	*1	—
	TPBAn address counter register	TPBAnCNT1	<TPBAn_base> + $10C_H$	8	*1	—
	TPBAn enable status register	TPBAnTE	<TPBAn_base> + 128_H	8	*1	—
	TPBAn start trigger register	TPBAnTS	<TPBAn_base> + $12C_H$	8	*1	—
	TPBAn stop trigger register	TPBAnTT	<TPBAn_base> + 130_H	8	*1	—

Note 1. n = 0: PBG51#14
n = 1: PBG51#15

37.3.2 TPBAnCTL — TPBAn Control Register

This register specifies the operation of the TPBAn.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 200_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	TPBAnPRS[1:0]		—	—	—	TPBAnDPS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R/W

Table 37.9 TPBAnCTL Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	TPBAnPRS[1:0]	Selects the count clock.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TPBAnDPS	Selects the duty setting pattern mode. 0: 16 bits × 64 patterns mode 1: 8 bits × 128 patterns mode

TPBAnPRS1	TPBAnPRS0	Description
0	0	PCLK is selected.
0	1	PCLK/2 is selected.
1	0	PCLK/4 is selected.
1	1	PCLK/8 is selected.

CAUTION

This register should be set when the timer is stopped (TPBAnTE = 0). If this register is erroneously rewritten, set the register again after stopping the timer.

37.3.3 TPBAnRDM — TPBAn Reload Data Mode Register

This register controls the reload timing of the TPBAn period setting register and TPBAn timer output level register values.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 118_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnRDM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.10 TPBAnRDM Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TPBAnRDM0	Controls the reload timing of the TPBAn period setting register (TPBAnCMP0) and TPBAn timer output level register (TPBAnTOL) values. 0: Reloads the values synchronously with a number-of-patterns matched detection interrupt (INTTPBAnIPAT). 1: Reloads the values synchronously with a period-matched detection interrupt (INTTPBAnIPRD).

CAUTION

Although this register can be rewritten during operation, the rewritten value is reflected immediately. Accordingly, during operation, this register should be rewritten when the reload request flag (TPBAnRSF) is 0.

37.3.4 TPBAnRSF — TPBAn Reload Status Register

This register indicates whether or not reload requests from the corresponding registers have been generated.

Access: This register can be read in 8-bit units.

Address: <TPBAn_base> + 110_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TPBAnRSF1	TPBAnRSF0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.11 TPBAnRSF Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TPBAnRSF1	Indicates whether or not a reload request from TPBAnCMP1 has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT1 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed.
0	TPBAnRSF0	Indicates whether or not a reload request from TPBAnCMP0 and TPBAnTOL has been generated. 0: No reload request generated or reload completed. 1: Reload request has been generated. This bit is set to 1 when 1 is written to the TPBAnRDT0 bit in the TPBAnRDT register. This bit is cleared at the timing when reload is performed.

37.3.5 TPBAnRDT — TPBAn Reload Data Trigger Register

This register enables reload of the register values.

Access: This register can be written in 8-bit units. It is always read as 0.

Address: <TPBAn_base> + 114_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TPBAnRDT1	TPBAnRDT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 37.12 TPBAnRDT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	TPBAnRDT1	Enables reload of the TPBAnCMP1 values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF1 is set to 1). The values are updated simultaneously at the next reload timing (reload).
0	TPBAnRDT0	Enables reload of the TPBAnCMP0 and TPBAnTOL values. 0: Write access is ignored. 1: Reload is enabled (TPBAnRSF0 is set to 1). The values are updated simultaneously at the next reload timing (reload).

37.3.6 TPBAnTOE — TPBAn Timer Output Enable Register

This register enables or disables the timer output.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 120_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTOE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.13 TPBAnTOE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TPBAnTOE0	Enables or disables the timer output (TPBAnO). 0: Disables the timer output based on counter operation. 1: Enables the timer output based on counter operation. • When the timer output is disabled, the level specified in TPBAnTO is output from the TPBAnO pin, and can be controlled by software. • When the timer output is enabled, TPBAnTO is set or cleared by the timer operation, and a PWM signal is output. Write access is prohibited (ignored).

37.3.7 TPBAnTO — TPBAn Timer Output Register

This register controls or reads timer output level.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + $11C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTO0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.14 TPBAnTO Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TPBAnTO0	Sets or indicates the output level of TPBAnO pin • When the timer output is disabled (TPBAnTOE.TPBAnTOE0 = 0) 0: Outputs low level. 1: Outputs high level. The output level can be controlled by rewriting this register while the timer output is disabled. • When the timer output is enabled (TPBAnTOE.TPBAnTOE0 = 1) 0: Low level is being output by the timer output. 1: High level is being output by the timer output. When the timer output is enabled, rewrite to this register is ignored.

37.3.8 TPBAnTOL — TPBAn Timer Output Level Register

This register controls the timer output level.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 124_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTOL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 37.15 TPBAnTOL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TPBAnTOL0	Specifies the active level of the timer output. 0: High 1: Low. • Setting of this bit is enabled when the timer output is enabled (TPBAnTOE.TPBAnTOE0 = 1). • Setting of this bit is reflected when the timer output is started, and change of the output level is reflected at the next reload timing.

CAUTION

This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 37.4.2, Compare Register Rewrite Operation**.

37.3.9 TPBAnCMP0 — TPBAn Period Setting Register

This is a 16-bit compare register for setting the PWM period.

Access: This register can be read/written in 16-bit units.

Address: <TPBAn_base> + 100_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPBAnCMP0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.16 TPBAnCMP0 Register Setting

Operating Mode	PWM Period	Minimum Value (Period)	Maximum Value (Period)
8 bits	TPBAnCMP0 + 1	1	100_H
16 bits	TPBAnCMP0 + 1	1	10000_H

CAUTION

- The PWM period is (TPBAnCMP0 + 1) count clock periods. Accordingly, for PWM output with 100% duty cycle, the maximum settable value is $FFFE_H$ (FE_H).
- This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 37.4.2, Compare Register Rewrite Operation**.

37.3.10 TPBAnBUFm — TPBAn Duty Setting Register

This register is a 16 × 64 buffer register for duty setting.

Access: This register can be read/written in 16-bit units.

Address: <TPBAn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPBAnBUFm															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.17 TPBAnBUFm Register Contents

Bit Position	Bit Name	Function
15 to 0	TPBAnBUFm15 to TPBAnBUFm0	Sets the duty value. This register can set the duty value either in 16 bits × 64 patterns mode (TPBAnDPS = 0) or 8 bits × 128 patterns mode (TPBAnDPS = 1) by setting the TPBAnDPS bit. In either mode, this register is accessed in 16-bit units by the CPU. For details, see **Section 37.4.3, Duty Rewrite Operation**.

CAUTION

The value set to this register is transferred to the duty setting buffer register (TPBAnCB2) synchronously with a period-matched detection interrupt (INTTPBAnIPRD). Rewrite during timer operation is reflected immediately. For details, see **Section 37.4.3, Duty Rewrite Operation**.

- When duty setting register with 8 bits × 128 patterns is used, the duty is set in the range from 00_H to FF_H.

  The formula to output a waveform of duty 100% is: TPBAnBUFm = TPBAnCMP0 + 1 ≤ $00FF_H$.
  Therefore, when PWM output of duty 100% is required, the maximum value of TPBAnCMP0 is $00FE_H$. When TPBAnBUFm is greater than TPBAnCMP0 + 1, duty value exceeds 100%, but the output is restricted to 100%.
- When duty setting register with 16 bits × 64 patterns is used, the duty is set in the range from 0000H to $FFFF_H$.

  The formula to output a waveform of duty 100% is: TPBAnBUFm = TPBAnCMP0 + 1 ≤ $FFFF_H$.
  Therefore, when PWM output of duty 100% is required, the maximum value of TPBAnCMP0 is $FFFE_H$.
  When TPBAnBUFm is greater than TPBAnCMP0 + 1, the duty value exceeds 100%, but the output is restricted to 100%.

37.3.11 TPBAnCMP1 — TPBAn Pattern Number Setting Register

This register sets the number of PWM output patterns.

Access: This register can be read/written in 8-bit units.

Address: <TPBAn_base> + 104_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	TPBAnCMP1						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.18 TPBAnCMP1 Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	TPBAnCMP1 [6:0]	Sets the number of patterns within the following range. TPBAnDPS = 0: 0 to 63 TPBAnDPS = 1: 0 to 127

CAUTION

- This register is a register to be reloaded. Rewrite during timer operation is reflected at the next reload timing. For details on reload, see **Section 37.4.2, Compare Register Rewrite Operation**.
- If 64 or a greater number is set as the number of patterns when the duty setting pattern is in 16 bits × 64 patterns mode (TPBAnDPS = 0), the address pointer changes from 63 to 00, and the duty value is transferred from 00 again. A number-of-patterns matched detection interrupt signal (INTTPBAnIPAT) is output by the match of the specified number of patterns and the lower 7-bit values of TPBAnCNT1.

37.3.12 TPBAnCNT0 — TPBAn Timer Counter Register

This register is a timer counter register that generates PWM output.

Access: This register can only be read in 16-bit units.

Address: <TPBAn_base> + 108_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPBAnCNT0															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

16-bit counter

This register is a counter register through which the 16-bit counter value can be read.

37.3.13 TPBAnCNT1 — TPBAn Address Counter Register

This register is a counter register that indicates the address pointer to the duty setting register.

Access: This register can only be read in 8-bit units.

Address: <TPBAn_base> + $10C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	TPBAnCNT1						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

7-bit counter

This register is a counter register that indicates the address of the TPBAnBUFm register.

37.3.14 TPBAnTE — TPBAn Enable Status Register

This register indicates whether the timer counter is operating or stopped.

Access: This register can only be read in 8-bit units.

Address: <TPBAn_base> + 128_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 37.19 TPBAnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	TPBAnTE0	Indicates whether the timer counter is operating or stopped. 0: The timer counter is stopped. 1: The timer counter is operating. • The TPBAnTE0 bit is set to 1 when 1 is written to the TPBAnTS bit or when a simultaneous start trigger is input. • The TPBAnTE0 bit is cleared to 0 when 1 is written to the TPBAnTT bit.

37.3.15 TPBAnTS — TPBAn Start Trigger Register

This register controls the timer counter start trigger.

Access: This register can only be written in 8-bit units. It is always read as 0.

Address: <TPBAn_base> + $12C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 37.20 TPBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TPBAnTS0	This bit is a trigger bit that enables the timer counter. 0: Write access is ignored. 1: Starts counting (TPBAnTE = 1).

CAUTION

Write access to this register during counting (TPBAnTE = 1) is ignored.

37.3.16 TPBAnTT — TPBAn Stop Trigger Register

This register controls the timer counter stop trigger.

Access: This register can only be written in 8-bit units. It is always read as 0.

Address: <TPBAn_base> + 130_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TPBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 37.21 TPBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TPBAnTT0	This bit is a trigger bit that disables the timer counter. 0: Write access is ignored. 1: Disables counting (TPBAnTE = 0).

37.4 Function

37.4.1 Basic Operation

37.4.1.1 Basic Operation of 16-Bit Counter (TPBAnCNT0)

Counting start

The 16-bit counter (TPBAnCNT0) starts counting from the value after reset $FFFF_H$.

Counter clear

The 16-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB0) set value of TPBAnCMP0.

Counter read during counting

The 16-bit counter value during counting can be read through TPBAnCNT0.

37.4.1.2 Basic Operation of 7-Bit Counter (TPBAnCNT1)

Counting start

The 7-bit counter (TPBAnCNT1) is initialized to 00_H and starts counting. Subsequently, the counter value is incremented synchronously with a period-matched detection interrupt (INTTPBAnIPRD).

Counter clear

The 7-bit counter is cleared by the match of the counter value and the buffer register (TPBAnCB1) set value of TPBAnCPM1.

Counter read during counting

The 7-bit counter value during counting can be read through TPBAnCNT1. The read value indicates TPBAnBUFm in which the duty value to be transferred next is stored.

37.4.2 Compare Register Rewrite Operation

The following registers are rewritten by reload.

- TPBAnCMP0
- TPBAnCMP1
- TPBAnTOL

Reload mode (simultaneous rewrite function)

Writing to TPBAnRDT enables reload of the registers corresponding to the set bits (sets the reload request flag (TPBAnRSF.TPBAnRSFk)), and the values of all the registers to be reloaded are updated simultaneously at the next reload timing (reload).

The reload timing of TPBAnCMP0 and TPBAnTOL is set by TPBAnRDM.

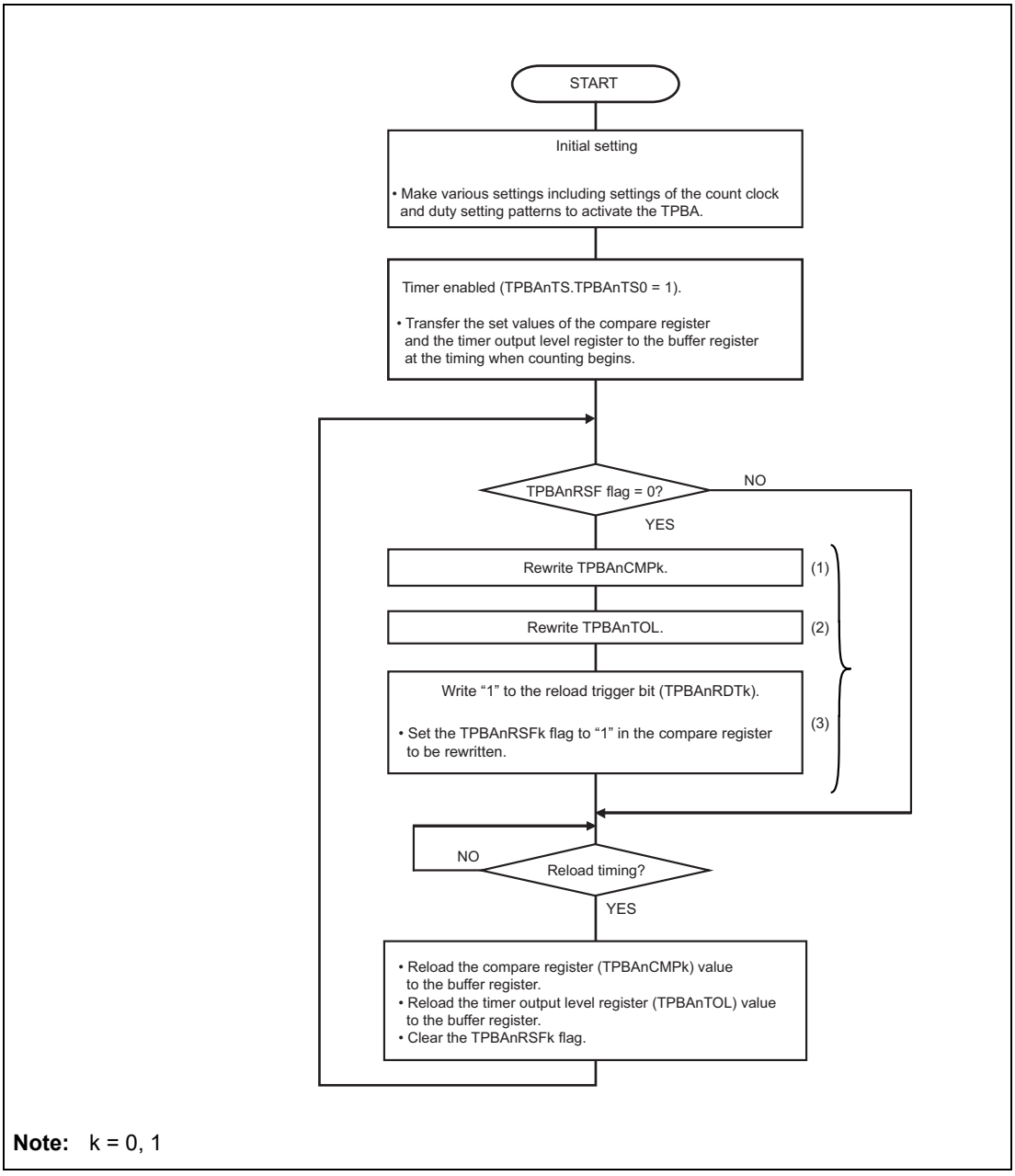
The reload timing of TPBAnCMP1 is the match timing (INTTPBAnIPAT) of the 7-bit counter (TPBAnCNT1) and the buffer register (TPBAnCB1) of TPBAnCMP1.

The registers to be reloaded should be rewritten when the reload request flag (TPBAnRSF.TPBAnRSFk) is 0.

Note: k = 0, 1

Setting Flow for Registers to Be Reloaded

The rewritten values of the registers to be reloaded (TPBAnCMP0, TPBAnCMP1, and TPBAnTOL) can be transferred to the respective buffer registers simultaneously at the reload timing.



Note: k = 0, 1

Figure 37.2 Basic Operation Flow of Reload (Simultaneous Rewrite Function)

CAUTION

Setting the TPBAnRDTk bit to 1 enables reload. Accordingly, the TPBAnRDTk bit should be rewritten after the registers to be reloaded have been rewritten.

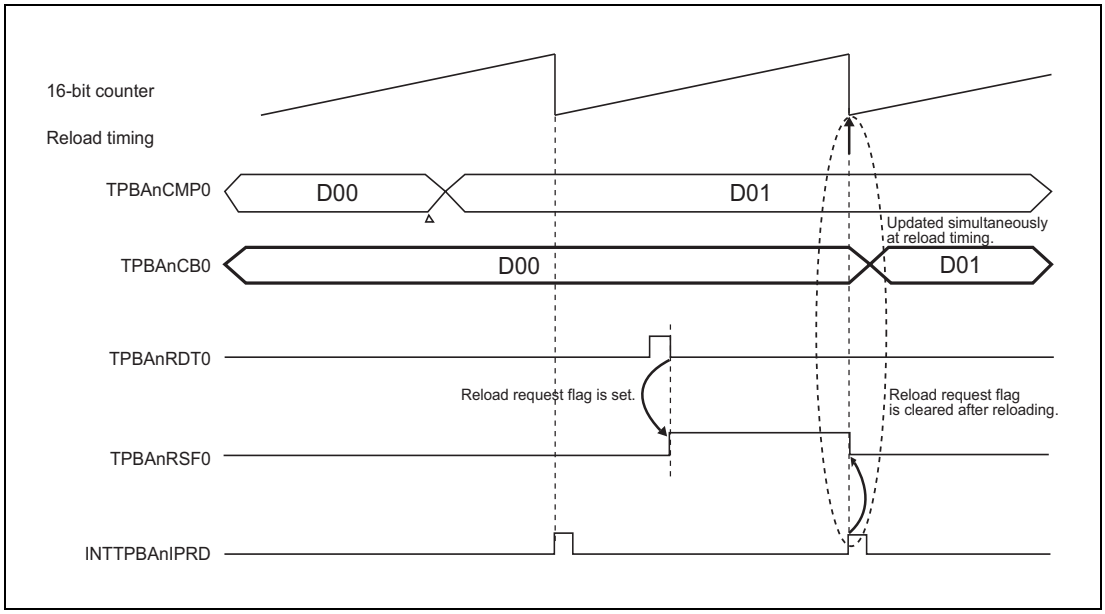


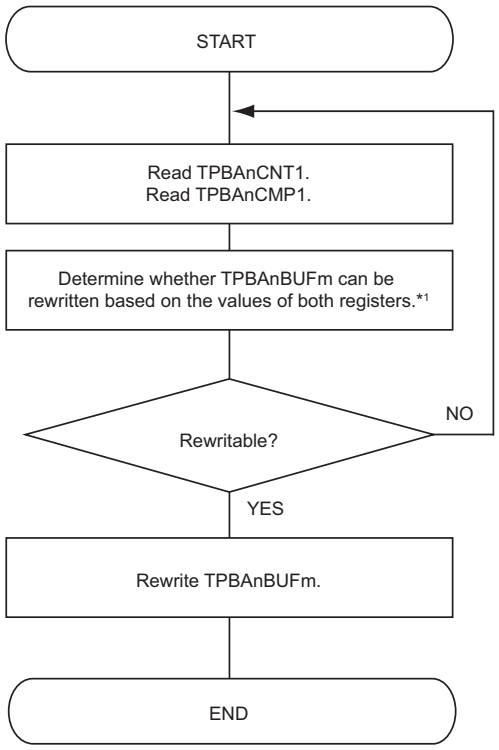
Figure 37.3 Simultaneous Rewrite Timing (TPBAnDPS = 0, TPBAnRDM = 0, and TPBAnTOL = 0)

37.4.3 Duty Rewrite Operation

TPBAnBUFm can be rewritten during operation.

The rewritten setting is reflected immediately.

37.4.3.1 TPBAnBUFm Setting Flow



Note 1. Since TPBAnBUFm can be written immediately, it is recommended that this register be rewritten during the last duty cycle output (TPBAnCNT1 = 00_H) of the pattern period (the number of patterns specified by TPBAnCMP1).

Figure 37.4 Basic Rewrite Flow of TPBAnBUFm

37.4.3.2 Access to TPBAnBUFm

TPBAnBUFm is accessed in 16 bit units. The following shows the access in 16 bits × 64 patterns mode and the access in 8 bits × 128 patterns mode.

- In 16 bits × 64 patterns mode (TPBAnDPS = 0)
  This register is accessed by the CPU in units of one 16-bit pattern.

15 0	
Pattern 64	$00FC_H$
Pattern 63	$00F8_H$
:	:
Pattern 3	0008_H
Pattern 2	0004_H
Pattern 1	0000_H

- In 8 bits × 128 patterns mode (TPBAnDPS = 1)
  This register is accessed by the CPU in units of two 8-bit patterns.

15 8	7 0	
Pattern 128	Pattern 127	$00FC_H$
Pattern 126	Pattern 125	$00F8_H$
:	:	:
Pattern 6	Pattern 5	0008_H
Pattern 4	Pattern 3	0004_H
Pattern 2	Pattern 1	0000_H

37.4.3.3 Relationship between TPBAnCNT1 Read Value and TPBAnBUFm

The duty value of the currently output PWM waveform can be obtained by reading the TPBAnCNT1 count value during operation. TPBAnBUFm in which the currently output duty value is stored can be found by one of the following formulas.

Table 37.22 TPBAnBUFm Formula

TPBAnDPS Bit	Formula		
	TPBAnCNT1 ≠ 00_H		**TPBAnCNT1 = 00_H**
0: 16 bits × 64 patterns mode	TPBAnCNT1 - 01_H(1)		TPBAnCMP1(2)
1: 8 bits × 128 patterns mode	TPBAnCNT1 value is an odd number	TPBAnCNT1/2(3)	TPBAnCMP1/2(5)
	TPBAnCNT1 value is an even number	(TPBAnCNT1 / 2) - 01_H(4)	

(1) When TPBAnDPS = 0 and the TPBAnCNT1 ≠ 00_H
The applicable register is found by the formula TPBAnCNT1 – 01_H.
(Example) When TPBAnCNT1 = 08_H: 08_H – 01_H = 07_H → TPBA0BUF07

(2) When TPBAnDPS = 0 and the TPBAnCNT1 = 00_H
The applicable register is found by the TPBAnCMP1 value.
(Example) When TPBAnCMP1 = 08_H: TPBAnBUF08

(3) When TPBAnDPS = 1 and the TPBAnCNT1 = an odd number
The applicable register is found by the formula TPBAnCNT1 / 2
(Example) When TPBAnCNT1 = 07_H: 07_H / 02_H = 03_H → TPBAnBUF03 (lower 8 bits)

(4) When TPBAnDPS = 1 and the TPBAnCNT1 = an even number
The applicable register is found by the formula (TPBAnCNT1 / 2) – 01_H.
(Example) When TPBAnCNT1 = 08_H: (08_H / 02_H) – 01_H = 03_H → TPBAnBUF03 (upper 8 bits)

(5) When TPBAnDPS = 1 and the TPBAnCNT1 = 00_H
The applicable register is found by the formula TPBAnCMP1 / 2.
(Example) When TPBAnCMP1 = 08_H: 08_H / 2 = 04_H → TPBAnBUF04 (lower 8 bits)

37.4.4 Basic Operation Example

Overview

A PWM signal is output from the TPBAnO pin according to the PWM period set in the TPBAnCMP0 register and duty cycle set in the TPBAnBUF00 to TPBAnBUF63 registers.

Prerequisites

- Select 16 bits × 64 patterns mode or 8 bits × 128 patterns mode by setting TPBAnDPS.
- Set the duty cycle to TPBAnBUF00 to TPBAnBUF63.
- Set the number of patterns to TPBAnCMP1.

Functional description

Set the PWM period, the number of patterns, duty cycle, and level to be output. Set TPBAnTS.TPBSnTS0 = 1 (or input a simultaneous start trigger) to start incrementing the timer counter value.

The TPBAnO output is set to the active level at the same time the counting begins. TPBAnCNT1 is incremented, and points to the address of the buffer in which the subsequent duty value is stored.

The output is set to the inactive level by the match of the 16-bit counter and the TPBAnBUFm buffer register (TPBAnCB2).

The duty value is then transferred from TPBAnBUFm to the buffer register (TPBAnCB2) by the match of the 16-bit counter and the TPBAnCMP0 buffer register (TPBAnCB0). Then, TPBAnCNT1 is incremented, and a period-matched detection interrupt (INTTPBAnIPRD) is generated. The TPBAnO output is set to the active level after one count clock.

During counting, a duty-cycle-matched detection interrupt (INTTPBAnIDTY) is generated by the match of the 16-bit counter and the buffer register (TPBAnCB2) of TPBAnBUFm.

A number-of-patterns matched detection interrupt (INTTPBAnIPAT) is generated by the match of the 7-bit counter and the TPBAnCMP1 buffer register (TPBAnCB1).

37.4.4.1 List of Operations

Table 37.23 16-Bit Counter Function

Operation		Setting Condition
16-bit counter	Start	Writing 1 to TPBAnTS or set simultaneous start trigger.
	Clear	Compare match of TPBAnCMP0 buffer register and 16-bit counter
	Stop	Writing 1 to TPBAnTT

Table 37.24 7-Bit Counter Function

Operation		Setting Condition
7-bit counter	Start	Writing 1 to TPBAnTS or set simultaneous start trigger.
	Clear	Compare match of TPBAnCMP1 buffer register and 7-bit counter
	Stop	Writing 1 to TPBAnTT

Table 37.25 Functions of Compare Registers and Buffer Registers

Register (Data)	Buffer Register	Rewrite Method	Rewrite during Operation	Function
TPBAnCMP0	TPBAnCB0	Reload	Possible	Setting period
TPBAnCMP1	TPBAnCB1	Reload	Possible	Setting number of patterns
TPBAnBUFm	TPBAnCB2	Rewrite immediately	Possible	Setting duty
TPBAnTOL	TPBAnTOLB	Reload	Possible	Setting output level

Buffer Registers

The registers that specify period, the number of patterns, duty, and timer output level consist of data registers that a user can directly set and buffer registers that a user cannot directly set.

Table 37.26 Timer Output Function

Pin	Function
TPBAnO	• When output is enabled (TPBAnTOE = 01_H) PWM output by compare match of the TPBAnBUFm buffer register (TPBAnCB2) and the 16-bit counter • When output is disabled (TPBAnTOE = 00_H) TPBAnTO set value

Table 37.27 Interrupt Requests

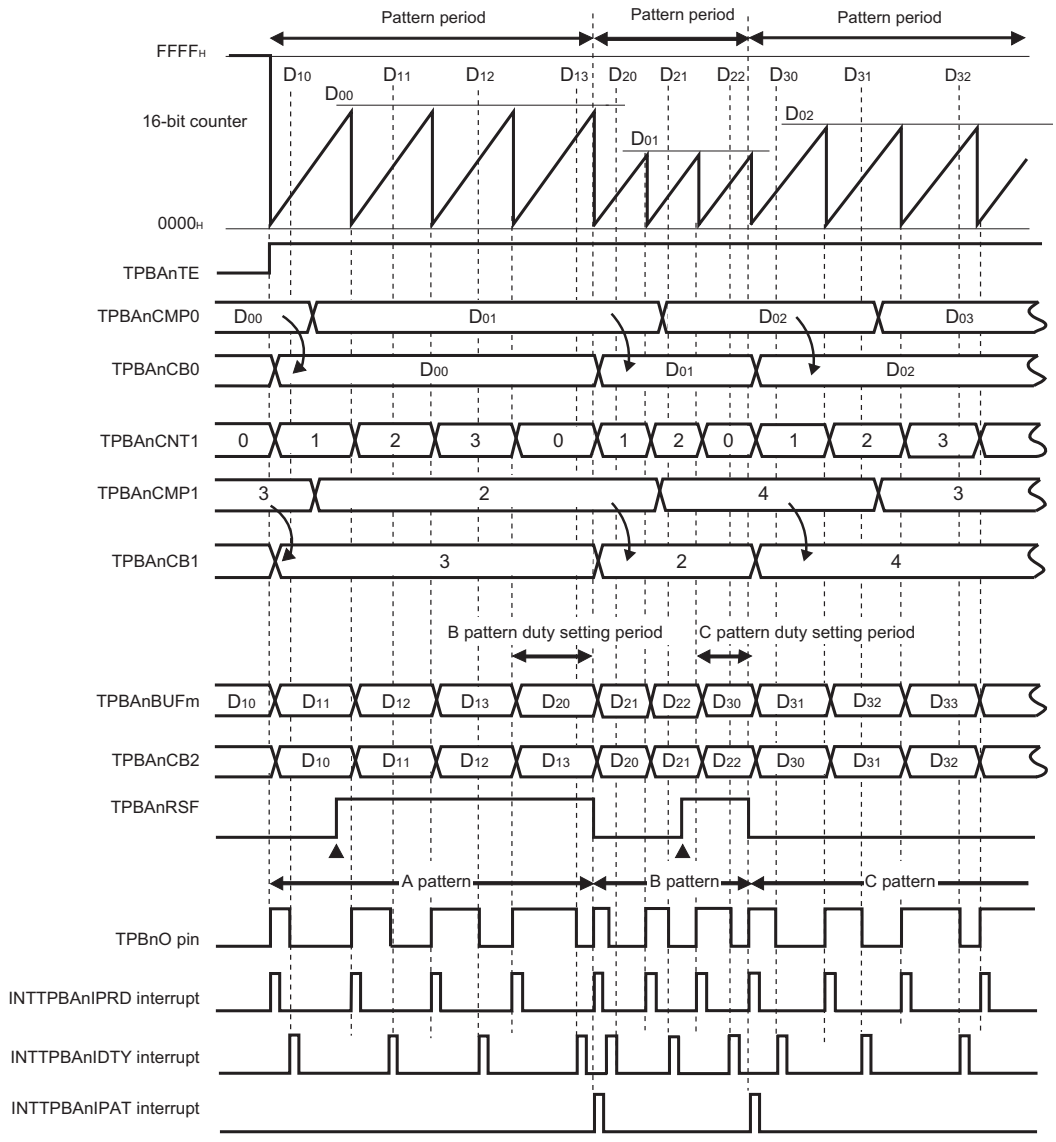
Interrupt	Function
INTTPBAnIPRD	Period-matched detection interrupt
INTTPBAnIDTY	Duty-cycle-matched detection interrupt
INTTPBAnIPAT	Number-of-patterns matched detection interrupt

Table 37.28 Compare Match Timing

Compare Match	Timing
TPBAnCMP0	When the 16-bit counter changes from match with TPBAnCMP0 to 0000_H.
TPBAnCMP1	When the 7-bit counter changes from match with TPBAnCMP1 to 01_H.
TPBAnBUFm	When the 16-bit counter matches with the buffer register (TPBAnCB2).

Table 37.29 Example of Setting Each Timer Output Condition

Pin	Item	Output Period	Output Duty	
			Output Condition	Setting Condition
TPBAnO	PWM output	(TPBAnCMP0 + 1) × count clock	Outputs an inactive level throughout one period (duty cycle 0%).	TPBAnBUFm = 0000_H
			Outputs an active level of one count clock in one period.	TPBAnBUFm = 0001_H
			Outputs an inactive level of one count clock in one period	TPBAnBUFm = TPBAnCMP0
			Outputs an active level throughout one period (duty cycle 100%).	TPBAnBUFm ≥ TPBAnCMP0 + 1



Note 1. PWM duty cycle = (TPBAnBUFm) / (TPBAnCMP0 set value + 1)
PWM period = (TPBAnCMP0 set value + 1) × (count clock period)

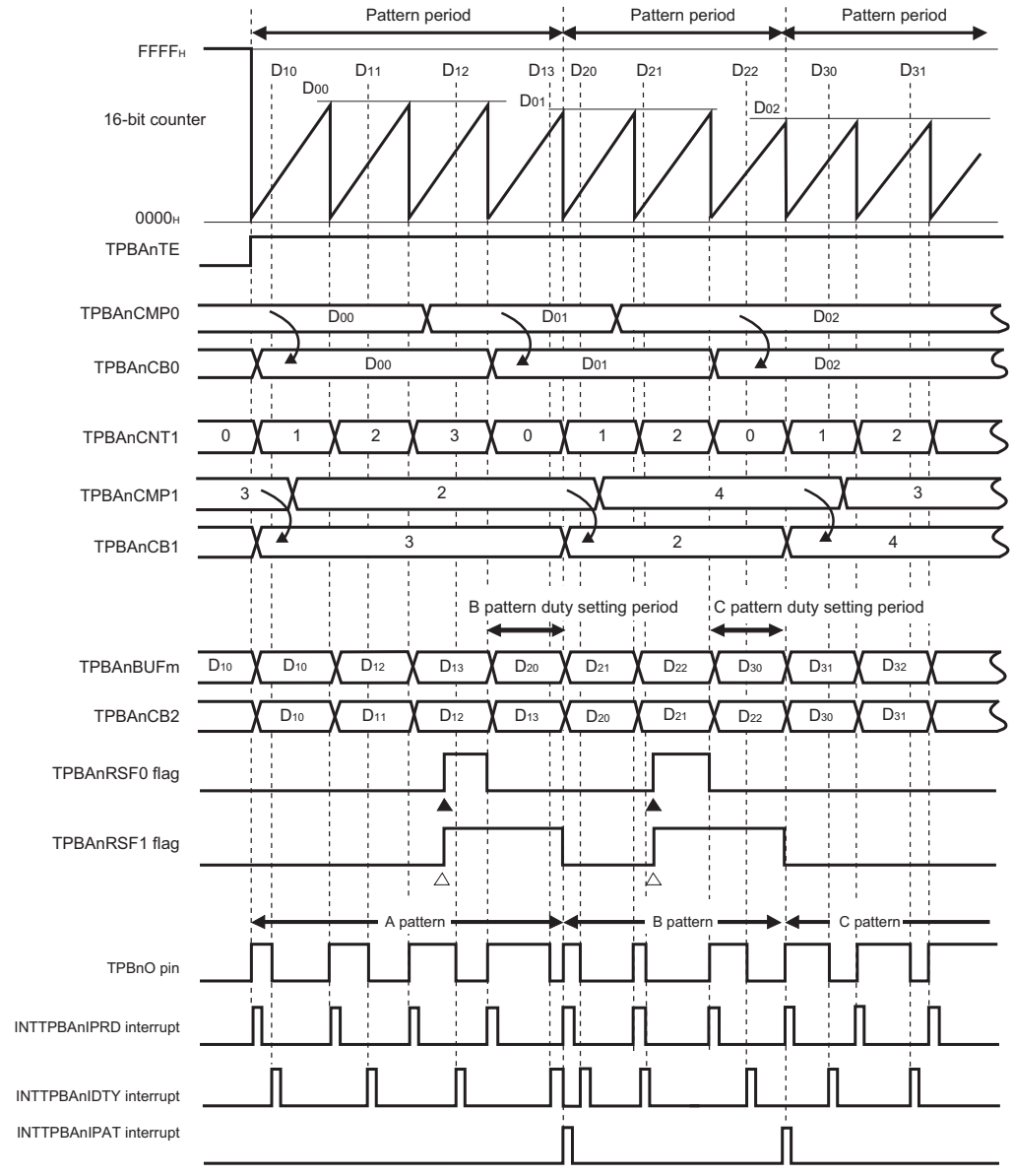
Note 2. ▲: Writing 1 to TPBAnRDT

Figure 37.5 Example of Basic Timing (1/2)

CAUTION

TPBAnO outputs active level 1 count clock after output of INTTPBAnIPRD and outputs inactive level at INTTPBAnIDTY output timing.

When a number-of-patterns matched detection interrupt is used as a trigger of the TPBAnCMP0 and TPBAnTOL reload timing (TPBAnIRDM.TPBAnRDM0 = 0 and TPBAnTOL = 0)



Note 1. PWM duty cycle = (TPBAnBUFm) / (TPBAnCMP0 set value + 1)
PWM period = (TPBAnCMP0 set value + 1) × (count clock period)
Note 2. ▲: Writing 1 to TPBAnRDT0
Note 3. Δ: Writing 1 to TPBAnRDT1

Figure 37.6 Example of Basic Timing (2/2)

CAUTION

TPBAnO outputs active level 1 count clock after output of INTTPBAnIPRD and outputs inactive level at INTTPBAnIDTY output timing.

When a period-matched detection interrupt is used as a trigger of the TPBAnCMP0 and TPBAnTOL reload timing (TPBAnIRDM.TPBAnRDM0 = 1 and TPBAnTOL = 0)

Section 38 Generic Timer Module (GTM)

Generic timer module (GTM) is a modular timer unit used to support body control and chassis & safety applications. It is designed to unload the CPU from a high interrupt load. Most of the task of GTM can run independently from the CPU.

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the units constituting GTM.

NOTE

Ignore the description of AXI Master/Slave function and submodules that are not supported in RH850/U2A-EVA Group. Regarding supported submodules in RH850/U2A-EVA Group, refer to **Section 38.28.1, GTM Device 358 Configuration**. Regarding actual register reset value, see **Section 38.28, GTM Device 358**.

38.1 Features of GTM for RH850/U2A-EVA

38.1.1 Number of Units

This microcontroller has the following number of GTM channels.

Table 38.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	GTMn									

Table 38.2 Index

Index	Description
n	Throughout this section, the individual GTM units are identified by the index “n” (n = 0).

38.1.2 IP version

The table below shows IP version in each device.

Table 38.3 IP version

Product Name	IP	Device	Revision
RH850/U2A-EVA(516 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A16(516 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A16(373 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A16(292 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A8(373 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A8(292 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A6(292 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A6(176 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A6(156 pins)	GTM-IP	GTM-IP_358	v3.5.0.2
RH850/U2A6(144 pins)	GTM-IP	GTM-IP_358	v3.5.0.2

38.1.3 Number of Sub-Units and Channels

The table below shows the availability of the GTM.

Table 38.4 Sub-Units and Channels

Sub-module	Number of Instances	Instance name	Channels
ARU	1	—	—
BRC	0	—	—
PSM (FIFO, AFD, F2A)	0	—	—
CMU	1	—	—
CCM	4	—	—
TBU	1	—	3 (n = 0 to 2)
TIM	4	TIM0 to TIM3	8 (n = 0 to 7)
TOM	0	—	—
ATOM	4	ATOM0 to ATOM3	8 (n = 0 to 7)
DTM	8	CDTM0_DTM[i](i = 4, 5) CDTM1_DTM[i](i = 4, 5) CDTM2_DTM[i](i = 4, 5) CDTM3_DTM[i](i = 4, 5)	4 (n = 0 to 3)
MCS	4	MCS0 to MCS3	8 (n = 0 to 7)
MCFG	1	—	—
MAP	0	—	—
DPLL	0	—	—
SPE	0	—	—
ICM	1	—	—
CMP	1	—	—
MON	1	—	—

Table 38.5 Memory Size

	RH850/U2A-EVA, U2A16, U2A8, U2A6
Sub-module	Size
MCS RAM0	8 KB/instance
MCS RAM1	4 KB/instance

38.1.4 Register/RAM Base Address

GTM base addresses are listed in the following table. GTM register and RAM addresses are given as offsets from the base addresses in general.

Table 38.6 Register and RAM Base Addresses

Base Address Name	Base Address	Bus Group
<GTM0_base>	FF60 0000_H	P-Bus Group 6H
<GTM0_1_base>	FF70 6000_H	P-Bus Group 6H

38.1.5 Clock Supply

GTM clock supplies are listed in the following table.

Table 38.7 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
GTM Cluster 0 to 3 (MCS0-3 and associated RAM)	GTM Main Clock	CLK_UHSB Switchable between 1/1 (high) and 1/2 (low) frequency.
	Register access clock	CLK_HSB

38.1.6 Interrupt Requests and Error Notifications

Mapping of ICM outputs to interrupts and DMA/DTS triggers is shown in the following table.

Table 38.8 Interrupt and DMA/DTS Requests (1/3)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
GTMn				
INTGTMA0TIM00	TIM0 Shared interrupts (TIM0_IRQ0)	79	Group1-140	Group2-44
INTGTMA0TIM01	TIM0 Shared interrupts (TIM0_IRQ1)	80	Group1-141	Group2-45
INTGTMA0TIM02	TIM0 Shared interrupts (TIM0_IRQ2)	81	Group1-142	Group2-46
INTGTMA0TIM03	TIM0 Shared interrupts (TIM0_IRQ3)	82	Group1-143	Group2-47
INTGTMA0TIM04	TIM0 Shared interrupts (TIM0_IRQ4)	83	Group1-144	Group2-48
INTGTMA0TIM05	TIM0 Shared interrupts (TIM0_IRQ5)	84	Group1-145	Group2-49
INTGTMA0TIM06	TIM0 Shared interrupts (TIM0_IRQ6)	85	Group1-146	Group2-50
INTGTMA0TIM07	TIM0 Shared interrupts (TIM0_IRQ7)	86	Group1-147	Group2-51
INTGTMA0TIM10	TIM1 Shared interrupts (TIM1_IRQ0)	87	Group1-148	Group2-52
INTGTMA0TIM11	TIM1 Shared interrupts (TIM1_IRQ1)	88	Group1-149	Group2-53
INTGTMA0TIM12	TIM1 Shared interrupts (TIM1_IRQ2)	89	Group1-150	Group2-54
INTGTMA0TIM13	TIM1 Shared interrupts (TIM1_IRQ3)	90	Group1-151	Group2-55
INTGTMA0TIM14	TIM1 Shared interrupts (TIM1_IRQ4)	91	Group1-152	Group2-56
INTGTMA0TIM15	TIM1 Shared interrupts (TIM1_IRQ5)	92	Group1-153	Group2-57
INTGTMA0TIM16	TIM1 Shared interrupts (TIM1_IRQ6)	93	Group1-154	Group2-58
INTGTMA0TIM17	TIM1 Shared interrupts (TIM1_IRQ7)	94	Group1-155	Group2-59
INTGTMA0TIM20	TIM2 Shared interrupts (TIM2_IRQ0)	95	Group1-156	Group2-60
INTGTMA0TIM21	TIM2 Shared interrupts (TIM2_IRQ1)	96	Group1-157	Group2-61
INTGTMA0TIM22	TIM2 Shared interrupts (TIM2_IRQ2)	97	Group1-158	Group2-62
INTGTMA0TIM23	TIM2 Shared interrupts (TIM2_IRQ3)	98	Group1-159	Group2-63
INTGTMA0TIM24	TIM2 Shared interrupts (TIM2_IRQ4)	99	Group1-160	Group2-64
INTGTMA0TIM25	TIM2 Shared interrupts (TIM2_IRQ5)	100	Group1-161	Group2-65
INTGTMA0TIM26	TIM2 Shared interrupts (TIM2_IRQ6)	101	Group1-162	Group2-66
INTGTMA0TIM27	TIM2 Shared interrupts (TIM2_IRQ7)	102	Group1-163	Group2-67
INTGTMA0TIM30	TIM3 Shared interrupts (TIM3_IRQ0)	103	Group1-164	Group2-68
INTGTMA0TIM31	TIM3 Shared interrupts (TIM3_IRQ1)	104	Group1-165	Group2-69
INTGTMA0TIM32	TIM3 Shared interrupts (TIM3_IRQ2)	105	Group1-166	Group2-70
INTGTMA0TIM33	TIM3 Shared interrupts (TIM3_IRQ3)	106	Group1-167	Group2-71
INTGTMA0TIM34	TIM3 Shared interrupts (TIM3_IRQ4)	107	Group1-168	Group2-72
INTGTMA0TIM35	TIM3 Shared interrupts (TIM3_IRQ5)	108	Group1-169	Group2-73
INTGTMA0TIM36	TIM3 Shared interrupts (TIM3_IRQ6)	109	Group1-170	Group2-74
INTGTMA0TIM37	TIM3 Shared interrupts (TIM3_IRQ7)	110	Group1-171	Group2-75
INTGTMA0MCS00	MCS0 Interrupt for channel (MCS0_IRQ0)	111	Group1-172	Group2-76
INTGTMA0MCS01	MCS0 Interrupt for channel (MCS0_IRQ1)	112	Group1-173	Group2-77
INTGTMA0MCS02	MCS0 Interrupt for channel (MCS0_IRQ2)	113	Group1-174	Group2-78
INTGTMA0MCS03	MCS0 Interrupt for channel (MCS0_IRQ3)	114	Group1-175	Group2-79
INTGTMA0MCS04	MCS0 Interrupt for channel (MCS0_IRQ4)	115	Group1-176	Group2-80
INTGTMA0MCS05	MCS0 Interrupt for channel (MCS0_IRQ5)	116	Group1-177	Group2-81
INTGTMA0MCS06	MCS0 Interrupt for channel (MCS0_IRQ6)	117	Group1-178	Group2-82

Table 38.8 Interrupt and DMA/DTS Requests (2/3)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
INTGTMA0MCS07	MCS0 Interrupt for channel (MCS0_IRQ7)	118	Group1-179	Group2-83
INTGTMA0MCS10	MCS1 Interrupt for channel (MCS1_IRQ0)	119	Group1-180	Group2-84
INTGTMA0MCS11	MCS1 Interrupt for channel (MCS1_IRQ1)	120	Group1-181	Group2-85
INTGTMA0MCS12	MCS1 Interrupt for channel (MCS1_IRQ2)	121	Group1-182	Group2-86
INTGTMA0MCS13	MCS1 Interrupt for channel (MCS1_IRQ3)	122	Group1-183	Group2-87
INTGTMA0MCS14	MCS1 Interrupt for channel (MCS1_IRQ4)	123	Group1-184	Group2-88
INTGTMA0MCS15	MCS1 Interrupt for channel (MCS1_IRQ5)	124	Group1-185	Group2-89
INTGTMA0MCS16	MCS1 Interrupt for channel (MCS1_IRQ6)	125	Group1-186	Group2-90
INTGTMA0MCS17	MCS1 Interrupt for channel (MCS1_IRQ7)	126	Group1-187	Group2-91
INTGTMA0MCS20	MCS2 Interrupt for channel (MCS2_IRQ0)	127	Group1-188	Group2-92
INTGTMA0MCS21	MCS2 Interrupt for channel (MCS2_IRQ1)	128	Group1-189	Group2-93
INTGTMA0MCS22	MCS2 Interrupt for channel (MCS2_IRQ2)	129	Group1-190	Group2-94
INTGTMA0MCS23	MCS2 Interrupt for channel (MCS2_IRQ3)	130	Group1-191	Group2-95
INTGTMA0MCS24	MCS2 Interrupt for channel (MCS2_IRQ4)	131	Group1-192	Group2-96
INTGTMA0MCS25	MCS2 Interrupt for channel (MCS2_IRQ5)	132	Group1-193	Group2-97
INTGTMA0MCS26	MCS2 Interrupt for channel (MCS2_IRQ6)	133	Group1-194	Group2-98
INTGTMA0MCS27	MCS2 Interrupt for channel (MCS2_IRQ7)	134	Group1-195	Group2-99
INTGTMA0MCS30	MCS3 Interrupt for channel (MCS3_IRQ0)	135	Group1-196	Group2-100
INTGTMA0MCS31	MCS3 Interrupt for channel (MCS3_IRQ1)	136	Group1-197	Group2-101
INTGTMA0MCS32	MCS3 Interrupt for channel (MCS3_IRQ2)	137	Group1-198	Group2-102
INTGTMA0MCS33	MCS3 Interrupt for channel (MCS3_IRQ3)	138	Group1-199	Group2-103
INTGTMA0MCS34	MCS3 Interrupt for channel (MCS3_IRQ4)	139	Group1-200	Group2-104
INTGTMA0MCS35	MCS3 Interrupt for channel (MCS3_IRQ5)	140	Group1-201	Group2-105
INTGTMA0MCS36	MCS3 Interrupt for channel (MCS3_IRQ6)	141	Group1-202	Group2-106
INTGTMA0MCS37	MCS3 Interrupt for channel (MCS3_IRQ7)	142	Group1-203	Group2-107
INTGTMA0ATOM00	ATOM0 Shared interrupts (ATOM0_IRQ0)	143	Group1-204	Group2-108
INTGTMA0ATOM01	ATOM0 Shared interrupts (ATOM0_IRQ1)	144	Group1-205	Group2-109
INTGTMA0ATOM02	ATOM0 Shared interrupts (ATOM0_IRQ2)	145	Group1-206	Group2-110
INTGTMA0ATOM03	ATOM0 Shared interrupts (ATOM0_IRQ3)	146	Group1-207	Group2-111
INTGTMA0ATOM10	ATOM1 Shared interrupts (ATOM1_IRQ0)	147	Group1-216	Group2-112
INTGTMA0ATOM11	ATOM1 Shared interrupts (ATOM1_IRQ1)	148	Group1-217	Group2-113
INTGTMA0ATOM12	ATOM1 Shared interrupts (ATOM1_IRQ2)	149	Group1-218	Group2-114
INTGTMA0ATOM13	ATOM1 Shared interrupts (ATOM1_IRQ3)	150	Group1-219	Group2-115
INTGTMA0ATOM20	ATOM2 Shared interrupts (ATOM2_IRQ0)	151	Group1-228	Group2-116
INTGTMA0ATOM21	ATOM2 Shared interrupts (ATOM2_IRQ1)	152	Group1-229	Group2-117
INTGTMA0ATOM22	ATOM2 Shared interrupts (ATOM2_IRQ2)	153	Group1-230	Group2-118
INTGTMA0ATOM23	ATOM2 Shared interrupts (ATOM2_IRQ3)	154	Group1-231	Group2-119
INTGTMA0ATOM30	ATOM3 Shared interrupts (ATOM3_IRQ0)	155	Group1-240	Group2-120
INTGTMA0ATOM31	ATOM3 Shared interrupts (ATOM3_IRQ1)	156	Group1-241	Group2-121
INTGTMA0ATOM32	ATOM3 Shared interrupts (ATOM3_IRQ2)	157	Group1-242	Group2-122
INTGTMA0ATOM33	ATOM3 Shared interrupts (ATOM3_IRQ3)	158	Group1-243	Group2-123
INTGTMA0AEI	AEI Shared interrupt	159	—	—
INTGTMA0ARU0	ARU_NEW_DATA0 Interrupt	160	Group1-252	—

Table 38.8 Interrupt and DMA/DTS Requests (3/3)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number
INTGTMA0ARU1	ARU_NEW_DATA1 Interrupt	161	Group1-253	—
INTGTMA0ARU2	ARU_ACC_ACK Interrupt	162	Group1-254	—
INTGTMA0CMP	CMP Shared interrupt	163	Group1-255	—
INTGTMA0ERR	GTM Error Interrupt	164	—	—
ATOMDREQ00	ATOM0_OUT Timer output trigger (ATOM0_OUT0) *1	—	Group1-208	Group3-50
ATOMDREQ01	ATOM0_OUT Timer output trigger (ATOM0_OUT1) *1	—	Group1-209	Group3-51
ATOMDREQ02	ATOM0_OUT Timer output trigger (ATOM0_OUT2) *1	—	Group1-210	Group3-52
ATOMDREQ03	ATOM0_OUT Timer output trigger (ATOM0_OUT3) *1	—	Group1-211	Group3-53
ATOMDREQ04	ATOM0_OUT Timer output trigger (ATOM0_OUT4) *1	—	Group1-212	Group3-54
ATOMDREQ05	ATOM0_OUT Timer output trigger (ATOM0_OUT5) *1	—	Group1-213	Group3-55
ATOMDREQ06	ATOM0_OUT Timer output trigger (ATOM0_OUT6) *1	—	Group1-214	Group3-56
ATOMDREQ07	ATOM0_OUT Timer output trigger (ATOM0_OUT7) *1	—	Group1-215	Group3-57
ATOMDREQ10	ATOM1_OUT Timer output trigger (ATOM1_OUT0) *1	—	Group1-220	Group3-58
ATOMDREQ11	ATOM1_OUT Timer output trigger (ATOM1_OUT1) *1	—	Group1-221	Group3-59
ATOMDREQ12	ATOM1_OUT Timer output trigger (ATOM1_OUT2) *1	—	Group1-222	Group3-60
ATOMDREQ13	ATOM1_OUT Timer output trigger (ATOM1_OUT3) *1	—	Group1-223	Group3-61
ATOMDREQ14	ATOM1_OUT Timer output trigger (ATOM1_OUT4) *1	—	Group1-224	Group3-62
ATOMDREQ15	ATOM1_OUT Timer output trigger (ATOM1_OUT5) *1	—	Group1-225	Group3-63
ATOMDREQ16	ATOM1_OUT Timer output trigger (ATOM1_OUT6) *1	—	Group1-226	Group3-64
ATOMDREQ17	ATOM1_OUT Timer output trigger (ATOM1_OUT7) *1	—	Group1-227	Group3-65
ATOMDREQ20	ATOM2_OUT Timer output trigger (ATOM2_OUT0) *1	—	Group1-232	Group3-66
ATOMDREQ21	ATOM2_OUT Timer output trigger (ATOM2_OUT1) *1	—	Group1-233	Group3-67
ATOMDREQ22	ATOM2_OUT Timer output trigger (ATOM2_OUT2) *1	—	Group1-234	Group3-68
ATOMDREQ23	ATOM2_OUT Timer output trigger (ATOM2_OUT3) *1	—	Group1-235	Group3-69
ATOMDREQ24	ATOM2_OUT Timer output trigger (ATOM2_OUT4) *1	—	Group1-236	Group3-70
ATOMDREQ25	ATOM2_OUT Timer output trigger (ATOM2_OUT5) *1	—	Group1-237	Group3-71
ATOMDREQ26	ATOM2_OUT Timer output trigger (ATOM2_OUT6) *1	—	Group1-238	Group3-72
ATOMDREQ27	ATOM2_OUT Timer output trigger (ATOM2_OUT7) *1	—	Group1-239	Group3-73
ATOMDREQ30	ATOM3_OUT Timer output trigger (ATOM3_OUT0) *1	—	Group1-244	Group3-74
ATOMDREQ31	ATOM3_OUT Timer output trigger (ATOM3_OUT1) *1	—	Group1-245	Group3-75
ATOMDREQ32	ATOM3_OUT Timer output trigger (ATOM3_OUT2) *1	—	Group1-246	Group3-76
ATOMDREQ33	ATOM3_OUT Timer output trigger (ATOM3_OUT3) *1	—	Group1-247	Group3-77
ATOMDREQ34	ATOM3_OUT Timer output trigger (ATOM3_OUT4) *1	—	Group1-248	Group3-78
ATOMDREQ35	ATOM3_OUT Timer output trigger (ATOM3_OUT5) *1	—	Group1-249	Group3-79
ATOMDREQ36	ATOM3_OUT Timer output trigger (ATOM3_OUT6) *1	—	Group1-250	Group3-80
ATOMDREQ37	ATOM3_OUT Timer output trigger (ATOM3_OUT7) *1	—	Group1-251	Group3-81

Note 1. Only rising edge of timer output is effective.

This module has no error notification.

38.1.7 Reset Sources

The following table shows GTM reset sources and GTM responses to each reset source.

Table 38.9 Reset Sources

Register Name	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
All registers	√	√	√	√	√	√	—

Note: GTM can be reset by a dedicated bit, which allows the software to set GTM in reset state without using external reset.

38.1.8 External Input and Output Pins

External input/output pins of GTM are listed below.

Table 38.10 Pin Function Information (1/2)

Pin Name	I/O	Description	Alternative Port Pin Signal
GTMn			
TIM0_IN0 TIM0_IN1 TIM0_IN2 TIM0_IN3 TIM0_IN4 TIM0_IN5 TIM0_IN6 TIM0_IN7	I	Timer input signals for TIM0	GTM0I0 GTM0I1 GTM0I2 GTM0I3 GTM0I4 GTM0I5 GTM0I6 GTM0I7
TIM1_IN0 TIM1_IN1 TIM1_IN2 TIM1_IN3 TIM1_IN4 TIM1_IN5 TIM1_IN6 TIM1_IN7	I	Timer input signals for TIM1	GTM1I0 GTM1I1 GTM1I2 GTM1I3 GTM1I4 GTM1I5 GTM1I6 GTM1I7
TIM2_IN0 TIM2_IN1 TIM2_IN2 TIM2_IN3 TIM2_IN4 TIM2_IN5 TIM2_IN6 TIM2_IN7	I	Timer input signals for TIM2	GTM2I0 GTM2I1 GTM2I2 GTM2I3 GTM2I4 GTM2I5 GTM2I6 GTM2I7
TIM3_IN0 TIM3_IN1 TIM3_IN2 TIM3_IN3 TIM3_IN4 TIM3_IN5 TIM3_IN6 TIM3_IN7	I	Timer input signals for TIM3	GTM3I0 GTM3I1 GTM3I2 GTM3I3 GTM3I4 GTM3I5 GTM3I6 GTM3I7
ATOM0_OUT0 ATOM0_OUT1 ATOM0_OUT2 ATOM0_OUT3 ATOM0_OUT4 ATOM0_OUT5 ATOM0_OUT6 ATOM0_OUT7	O	Timer output signals for ATOM0*1	GTMAT0O0 GTMAT0O1 GTMAT0O2 GTMAT0O3 GTMAT0O4 GTMAT0O5 GTMAT0O6 GTMAT0O7

Table 38.10 Pin Function Information (2/2)

Pin Name	I/O	Description	Alternative Port Pin Signal
ATOM1_OUT0 ATOM1_OUT1 ATOM1_OUT2 ATOM1_OUT3 ATOM1_OUT4 ATOM1_OUT5 ATOM1_OUT6 ATOM1_OUT7	O	Timer output signals for ATOM1*1	GTMAT1O0 GTMAT1O1 GTMAT1O2 GTMAT1O3 GTMAT1O4 GTMAT1O5 GTMAT1O6 GTMAT1O7
ATOM2_OUT0 ATOM2_OUT1 ATOM2_OUT2 ATOM2_OUT3 ATOM2_OUT4 ATOM2_OUT5 ATOM2_OUT6 ATOM2_OUT7	O	Timer output signals for ATOM2*1	GTMAT2O0 GTMAT2O1 GTMAT2O2 GTMAT2O3 GTMAT2O4 GTMAT2O5 GTMAT2O6 GTMAT2O7
ATOM3_OUT0 ATOM3_OUT1 ATOM3_OUT2 ATOM3_OUT3 ATOM3_OUT4 ATOM3_OUT5 ATOM3_OUT6 ATOM3_OUT7	O	Timer output signals for ATOM3*1	GTMAT3O0 GTMAT3O1 GTMAT3O2 GTMAT3O3 GTMAT3O4 GTMAT3O5 GTMAT3O6 GTMAT3O7
ATOM0_OUT0_N ATOM0_OUT1_N ATOM0_OUT2_N ATOM0_OUT3_N ATOM0_OUT4_N ATOM0_OUT5_N ATOM0_OUT6_N ATOM0_OUT7_N	O	Inverted timer output signals for ATOM0*1	GTMAT0O0N GTMAT0O1N GTMAT0O2N GTMAT0O3N GTMAT0O4N GTMAT0O5N GTMAT0O6N GTMAT0O7N
ATOM1_OUT0_N ATOM1_OUT1_N ATOM1_OUT2_N ATOM1_OUT3_N ATOM1_OUT4_N ATOM1_OUT5_N ATOM1_OUT6_N ATOM1_OUT7_N	O	Inverted timer output signals for ATOM1*1	GTMAT1O0N GTMAT1O1N GTMAT1O2N GTMAT1O3N GTMAT1O4N GTMAT1O5N GTMAT1O6N GTMAT1O7N
ATOM2_OUT0_N ATOM2_OUT1_N ATOM2_OUT2_N ATOM2_OUT3_N ATOM2_OUT4_N ATOM2_OUT5_N ATOM2_OUT6_N ATOM2_OUT7_N	O	Inverted timer output signals for ATOM2*1	GTMAT2O0N GTMAT2O1N GTMAT2O2N GTMAT2O3N GTMAT2O4N GTMAT2O5N GTMAT2O6N GTMAT2O7N
ATOM3_OUT0_N ATOM3_OUT1_N ATOM3_OUT2_N ATOM3_OUT3_N ATOM3_OUT4_N ATOM3_OUT5_N ATOM3_OUT6_N ATOM3_OUT7_N	O	Inverted timer output signals for ATOM3*1	GTMAT3O0N GTMAT3O1N GTMAT3O2N GTMAT3O3N GTMAT3O4N GTMAT3O5N GTMAT3O6N GTMAT3O7N
CMU_ECLK0 CMU_ECLK1 CMU_ECLK2	O	External Clock	GTMECLK0 GTMECLK1 GTMECLK2

Note 1. Output buffer of these signals could be controlled to Hi-Z by PIC2.

38.2 Overview

38.2.1 Functional Overview

GTM consists of the following sub-modules.

- Advanced Routing Unit (ARU)
  - Data routing between GTM sub-modules.
- Clock Management Unit (CMU)
  - Clock pre-scaler for GTM internal and external clocks.
- Cluster Configuration Module (CCM)
  - Controlling cluster configurations:
    - Cluster’s clock frequency.
    - Module clock gating.
    - Status monitoring of the cluster's MCS bus master.
    - Address range protection.
- Time Base Unit (TBU)
  - Three independent 24-bit time bases.
  - One module counter for angular conversions.
  - Supporting the following modes:
    - Free Running Counter Mode.
    - Forward/Backward Counter Mode.
- Timer Input Module (TIM)
  - Filter, capture/compare with a time stamp and input signal timeout.
  - Input from other peripherals to GTM.
- ARU-connected Timer Output Module (ATOM)
  - Complex output waveform generation without CPU interactions with 24-bit time base counter.
  - Input signal to other peripherals.
- Dead Time Module (DTM)
  - Hardware support for dead time generation.
- Multi Channel Sequencer (MCS)
  - Programmable sequencer with RISC-like instruction set and dedicated RAM.
- Memory Configuration (MCFG)
  - Mapping MCS physical memory to MCS instances.
- Interrupt Concentrator Module (ICM)
  - Bundling GTM internal interrupts before they are routed to INTC and DMAC.
- Output Compare Unit (CMP)

– Support for safety relevant applications, compare of DTM outputs.

- Monitoring Unit (MON)
  – Monitoring GTM internal clocks and MCS functionality.

38.2.2 Block Diagram

The following figure shows interconnections between GTM-IP and other peripherals.

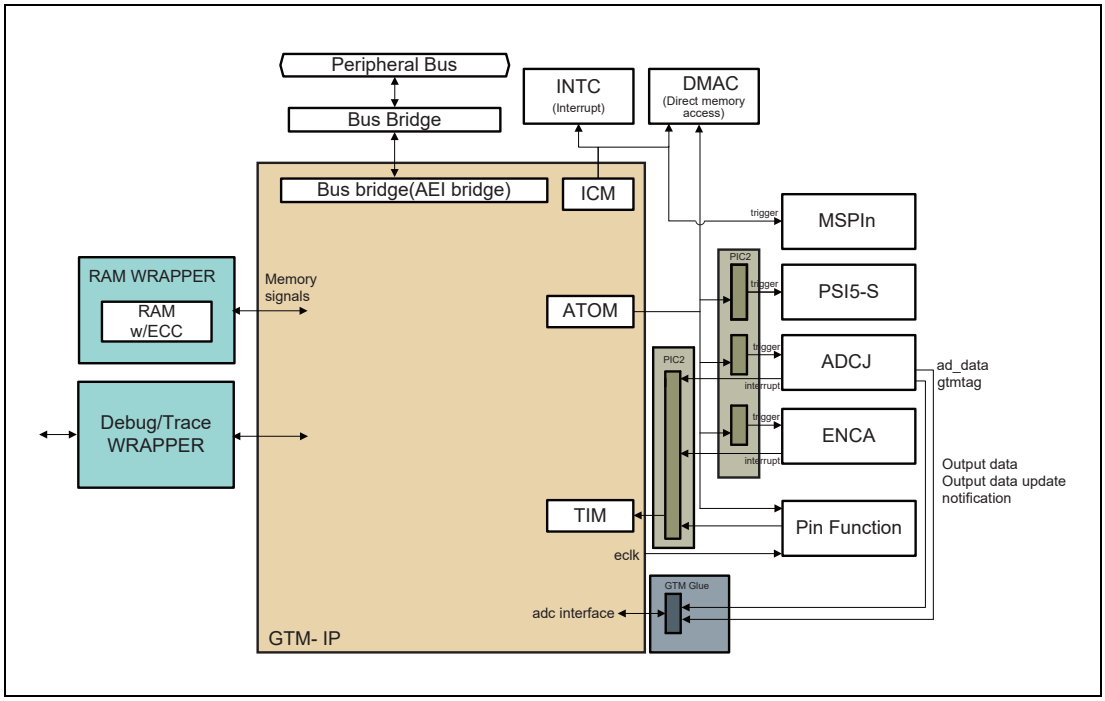


Figure 38.1 GTM integration diagram

38.3 Registers

38.3.1 List of Registers

GTM registers are listed in the following table.

For details about <GTM0_base> and <GTM0_1_base>, see **Section 38.1.4, Register/RAM Base Address**.

Table 38.11 List of Registers

Module name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GTM0	All of GTM-IP registers	*1	<GTM0_base> + *1	32	PBG6H0#2	—
GTM0_1	GTM ADCI channel selection register 0	GTM_ADCI_CHSEL0	<GTM0_1_base> + 000_H	8, 16, 32	PBG6H0#2	—
	GTM ADCI channel selection register 1	GTM_ADCI_CHSEL1	<GTM0_1_base> + 004_H	8, 16, 32	PBG6H0#2	—

Note 1. For details of GTM-IP registers, refer to **Section 38.28.6, GTM Register and Memory Addresses**. Also, the access size of RAM has to be 32-bit wide.

38.3.2 GTM_ADCI_CHSELn — GTM ADCI channel selection register n

Results of A/D conversion are stored in ADC_CH[y]_DATA registers (y = 0 to 31). Either registers 0 to 15 or registers 16 to 31 can be used based on GTM_ADCI_CHSELn.CHS. In addition a register designation is also based on GTMTAG (ADCJ). For more information about GTMTAG configuration, refer to **Section 43, Analog to Digital Converter (ADCJ)**. (n = 0, 1)

Access: This register can be read or written in 32-bit units.

Address: <GTM0_1_base> + n × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.12 GTM_ADCI_CHSELn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CHS	The analog to digital conversion data storage territory related in GTMTAG of ADCJ is designated. 0: AD data is stored in ADC_CH[y]_DATA (y = 0 to 15) 1: AD data is stored in ADC_CH[y]_DATA (y = 16 to 31)

38.4 Introduction

38.4.1 Overview

This document is the specification for the Generic Timer Module (GTM). It contains a module framework with sub-modules of different functionality. These sub-modules can be combined in a configurable manner to form a complex timer module that serves different application domains and different classes within one application domain. Because of this scalability and configurability the timer is called generic.

The scalability and configurability is reached with an architecture philosophy where dedicated hardware sub-modules are located around a central routing unit (called Advanced Routing Unit (ARU)). The ARU can connect the sub-modules in a flexible manner. The connectivity is software programmable and can be configured during runtime.

Nevertheless, the GTM-IP is designed to unload the CPU or a peripheral core from a high interrupt load. Most of the tasks inside the GTM-IP can run-once setup by an external CPU-independent and in parallel to the software. There may be special situations, where the CPU has to take action but the goal of the GTM design was to reduce these situations to a minimum.

The hardware sub-modules have dedicated functionality's, e.g. there are timer input modules where incoming signals can be captured and characterized together with a notion of time. By combination of several sub-modules through the ARU complex functions can be established.

E.g. the signals characterized at an input module can be routed to a signal processing unit where an intermediate value about the incoming signal frequency can be calculated.

The modules that help to implement such complex functions are called infrastructure components further on. These components are present in all GTM variants. However, the number of these components may vary from device to device.

Other sub-modules have a more general architecture and can fulfill typical timer functions, e.g. there are PWM generation units. The third classes of sub-modules are those fulfilling a dedicated functionality for a certain application domain, e.g. the DPLL serves engine management applications. A fourth group of sub-modules is responsible for supporting the implementation of safety functions to fulfill a defined safety level. The module ICM is responsible for interrupt services and defines the fifth group.

The master communication from GTM to CPU works over module AXIM. The slave communication from CPU to GTM is doing by AXIS which is describing in an extra document called Module-Integration-Guide. Both module are collecting as interface components, the sixth group.

Each GTM-IP is build up therefore with sub-modules coming from those six groups. The application class is defined by the amount of components of those sub-modules integrated into the implemented GTM-IP.

38.4.2 Document Structure

The structure of this document is motivated out of the aforementioned sub-module classes. **Section 38.5, GTM Architecture** describes the dedicated GTM-IP implementation this specification is written for. It gives an overview about the implemented sub-modules.

The master interface to the CPU is describing in **Section 38.6, AXI Master**. The following **Section 38.7, Advanced Routing Unit (ARU)** up to **Section 38.14, Time Base Unit (TBU)** deals with the so called infrastructure components for routing, clock management and common time base functions. **Section 38.15, Timer Input Module (TIM)** to **Section 38.18, Dead Time Module (DTM)** describe the signal input and output modules while the following **Section 38.19, Multi Channel Sequencer (MCS)** explains the signal processing and generation sub-module with **Section 38.20, Memory Configuration (MCFG)** its memory configuration. The next **Section 38.21, TIM0 Input Mapping Module (MAP)** to **Section 38.23, Sensor Pattern Evaluation (SPE)** provides a detailed description of application specific modules like the MAP, DPLL and SPE. **Section 38.24, Interrupt Concentrator Module (ICM)** describes a module that bundles several interrupts coming from the other sub-modules and connect them to the outside world. The last **Section 38.25, Output Compare Unit (CMP)** to **Section 38.26, Monitor Unit (MON)** provides to safety related modules like CMP and MON sub-modules.

These sub-module groups are shown in the following table:

Chapter	Sub-module	Group
38.6	AXI Master	Master interface to the CPU
38.7	Advanced Routing Unit (ARU)	Infrastructural components
38.8	Broadcast Module (BRC)	Infrastructural components
38.9	First In First Out Module (FIFO)	Infrastructural components
38.10	AEI to FIFO Data Interface (AFD)	Infrastructural components
38.11	FIFO to ARU Unit (F2A)	Infrastructural components
38.12	Clock Management Unit (CMU)	Infrastructural components
38.13	Cluster Configuration Module (CCM)	Infrastructural components
38.14	Time Base Unit (TBU)	Infrastructural components
38.15	Timer Input Module (TIM)	IO Modules
38.16	Timer Output Module (TOM)	IO Modules
38.17	ARU-connected Timer Output Module (ATOM)	IO Modules
38.18	Dead Time Module (DTM)	IO Modules
38.19	Multi Channel Sequencer (MCS)	Signal generation and processing
38.20	Memory Configuration (MCFG)	Memory for signal generation and processing
38.21	TIM0 Input Mapping Module (MAP)	Application specific modules
38.22	Digital PLL Module (DPLL)	Application specific modules
38.23	Sensor Pattern Evaluation (SPE)	Application specific modules
38.24	Interrupt Concentrator Module (ICM)	Interrupt services
38.25	Output Compare Unit (CMP)	Safety features
38.26	Monitoring Unit (MON)	Safety features

38.5 GTM Architecture

38.5.1 Overview

As already mentioned in **Section 38.4, Introduction** the GTM-IP forms a generic timer platform that serves different application domains and different classes within these application domains. Depending on these multiple requirements of application domains multiple device configurations with different number of sub-modules (i.e. ATOM, BRC, MCS, PSM, SPE, TIM, TOM, DTM) and different number of channel per sub-module (if applicable) are possible. The device dependent configuration (i.e. the number of sub-modules) is listed in the device specific **Section 38.28, GTM Device 358**. The Parameter Storage Module (PSM) is only a virtual hierarchy and consists of the sub-modules F2A, FIFO and AFD.

The Cluster Dead Time Module (CDTM) is also a virtual hierarchy and consists of up to six DTM modules. It depends on the GTM device configuration which of the six DTM instances are available. Refer to **Section 38.28, GTM Device 358** for list of available DTM instances. In general, the first four DTM modules inside a CDTM[n] hierarchy are connected to the outputs of the TOM instance [n] of the cluster [n], the other two DTM instances are connected to the outputs of the ATOM instance [n] of this cluster [n].

The cluster view of a GTM-IP architecture is depicted in **Figure 38.2**. This is a generic figure which shows an exemplify GTM-IP device configuration.

The device dependent configuration (i.e. the count of sub-modules and channels per sub-module) is listed in the device specific **Section 38.28, GTM Device 358**.

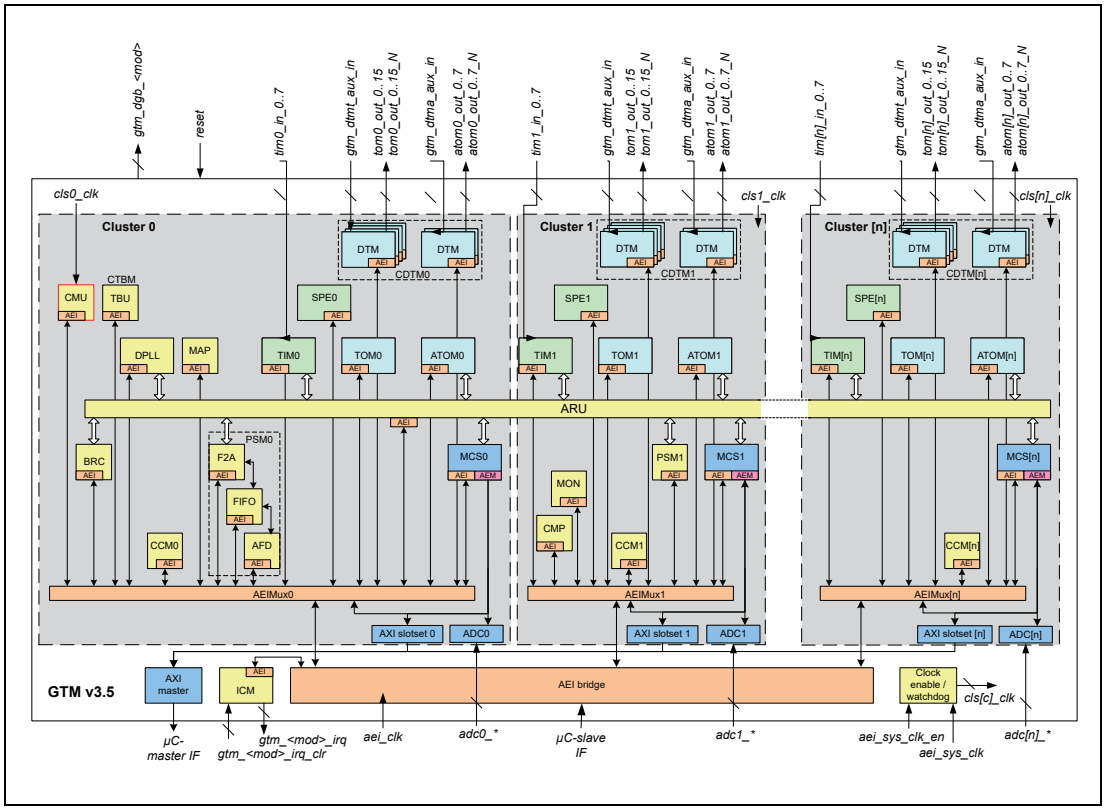


Figure 38.2 GTM Architecture Block Diagram

The GTM-IP is divided in multiple clusters 0 to n. A certain amount of modules exist in each cluster. The operating frequency of a cluster can be configured to OFF, aei_sys_clk or aei_sys_clk/2. The clock enable generation can be implemented internal to the GTM_IP or external. In case of an external enable generation aei_sys_clk_en is used to generate the internal clocks. In addition an enable watchdog is implemented to monitor the correctness of the external applied enable signals aei_sys_clk_en.

The central component of the GTM-IP is the Advanced Routing Unit (ARU) where most of the sub-modules are located around and connected to. This ARU forms together with the Broadcast (BRC) and the Parameter Storage Module (PSM) the infrastructural part of the GTM. The ARU is able to route data from a connected source sub-module to a connected destination sub-module. The routing is done in a deterministic manner with a round-robin scheduling scheme of connected channels which receive data from ARU and with a worst case round−trip time.

The routed data word size of the ARU is 53 bit. The data word can logically be split into three parts. These parts are shown in **Figure 38.3, ARU Data Word**. Bits 0 to 23 and bits 24 to 47 typically hold data for the operation registers of the GTM-IP. This can be, for example, the duty cycle and period duration of a measured PWM input signal or the output characteristic of an output PWM to be generated. Another possible content of Data0 and Data1 can be two 24 bit values of the GTM-IP time bases TBU_TS0, TBU_TS1 and TBU_TS2. Bits 48 to 52 can contain control bits to send control information from one sub-module to another. These ARU Control Bits (ACB) can have a different meaning for different sub-modules. It is also possible to route data from a source to a destination and the destination can act later on as source for another destination. These routes through the GTM-IP are further on called data streams. For a detailed description of the ARU sub-module, refer to **Section 38.7, Advanced Routing Unit (ARU)**.

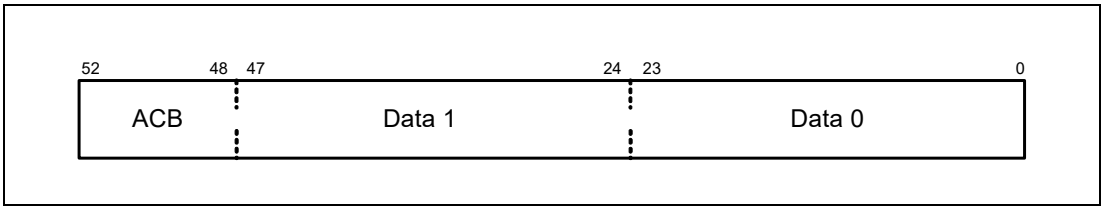


Figure 38.3 ARU Data Word

The BRC is able to distribute data from one source module to more than one destination modules connected to the ARU. The PSM sub-module consists of three sub-units, the AEI-to-FIFO Data Interface (AFD), FIFO-to-ARU Interface (F2A) and the FIFO itself. The PSM can serve as a data storage for incoming data characteristics or as parameter storage for outgoing data. This data is stored in a RAM that is logically located inside the FIFO sub-unit, but physically the RAM is implemented and integrated by the silicon vendor with his RAM implementation technology. Therefore, the GTM-IP provides the interface to the RAM at its module boundary. The AFD sub-unit is the interface between the FIFO and the GTM SoC system bus interface AEI (see **Section 38.5.2.1, GTM-IP Generic Bus Interface (AEI)** for detailed discussion). The F2A sub-unit is the interface between the FIFO sub-unit and the ARU.

Signals are transferred into the GTM-IP at the Timer Input Modules (TIM). These modules are able to filter the input signals and annotate additional information. Each channel is for example able to measure pulse high or low times and the period of a PWM signal in parallel and route the values to ARU for further processing. The internal operation registers of the TIM sub-module are 24 bits wide.

The Clock Management Unit (CMU) serves up to 13 different clocks for the GTM and up to three external clock pins GTM_ECLK0 to 2. It acts as a clock divider for the system clock. The counters implemented inside other sub-modules are typically driven from this sub-module.

Note that the CMU clocks are implemented as enable signals for the counters while the whole system runs with the GTM global clock SYS_CLK. This global clock typically corresponds to the micro controller bus clock the GTM-IP is connected to and should not exceed 100 MHz because of the power dissipation of the used transistors where the GTM is implemented with.

The TBU provides up to three independent common time bases for the GTM-IP. In general, the number of time bases depends on the implemented device. If three time bases are implemented, two of these time bases can also be clocked with the digital PLL (DPLL) sub_inc1 c and sub_inc2c outputs. The DPLL generates the higher frequent clock signals sub_inc1, sub_inc2, sub_inc1c and sub_inc2c on behalf of the frequencies of up to two input signals. These two input signals can be selected out of six incoming signals from the TIM0 sub-module. In this sub-module the incoming signals are filtered and transferred to the MAP sub-module where two of these six signals are selected for further processing inside the DPLL.

Signal outputs are generated with the Dead Time Module (DTM), Timer Output Modules (TOM) and the ARU-connected TOMs (ATOM). Each TOM channel is able to generate a PWM signal at its output. Because of the integrated shadow register even the generation of complex PWM outputs is possible with the TOM channels by serving the parameters with the CPU. It is possible to trigger TOM channels for a successor TOM sub-module through a trigger line between TOM(x)_CH(15) and TOM(x+1)_CH(0). But to avoid long trigger paths the GTM-IP integrator can configure after which TOM sub-module instance a register is placed into the trigger signal chain. Each register results in one SYS_CLK cycle delay of the trigger signal. Refer to device specification of silicon vendor for unregistered trigger chain length.

In addition, each TOM sub-module can integrate functions to drive one BLDC engine. This BLDC support is established together with the TIM and Sensor Pattern Evaluation (SPE) sub-module.

The ATOMs offer the additional functionality to generate complex output signals without CPU interaction by serving these complex waveform characteristics by other sub-modules that are connected to the ARU like the PSM or Multi Channel Sequencer (MCS). While the internal operation and shadow registers of the TOM channels are 16 bit wide, the operation and shadow registers of the ATOM channels are 24 bit wide to have a higher resolution and to have the opportunity to compare against time base values coming from the TBU.

It is possible to trigger ATOM channels for a successor ATOM sub-module through a trigger line between ATOM(x)_CH(7) and ATOM(x+1)_- CH(0). But to avoid long trigger paths the GTM-IP integrator can configure after which ATOM sub-module instance a register is placed into the trigger signal chain. Each register results in one SYS_CLK cycle delay of the trigger signal. Refer to device specification of silicon vendor for unregistered trigger chain length.

Together with the MCS the ATOM is able to generate an arbitrary predefined output sequence at the GTM-IP output pins. The output sequence is defined by instructions located in RAM connected to the MCS sub-module. The instructions define the points were an output signal should change or to react on other signal inputs. The output points can be one or two time stamps (or even angle stamp in case of an engine management system) provided by the TBU. Since the MCS is able to read data from the ARU it is also able to operate on incoming data routed from the TIM. Additionally, the MCS can process data that is located in its connected RAMs. The MCS RAM is located logically inside the MCS while the silicon vendor has to implement its own RAM technology there.

The two modules Compare Module (CMP) and Monitor Module (MON) implement safety related features. The CMP compares two output channels of the DTM and sends the result to the MON sub-module were the error is signaled to the CPU. The MON module is also able to monitor the ARU and CMU activities.

In the described implementation the sub-modules of the GTM-IP have a huge amount of different interrupt sources. These interrupt sources are grouped and concentrated by the Interrupt Concentrator Module (ICM) to form a much easier manageable bunch of interrupts that are visible outside of the GTM-IP.

On the GTM-IP top level there are some configurable signal connections from the signal output of the DTM modules to the input signals of the TIM modules.

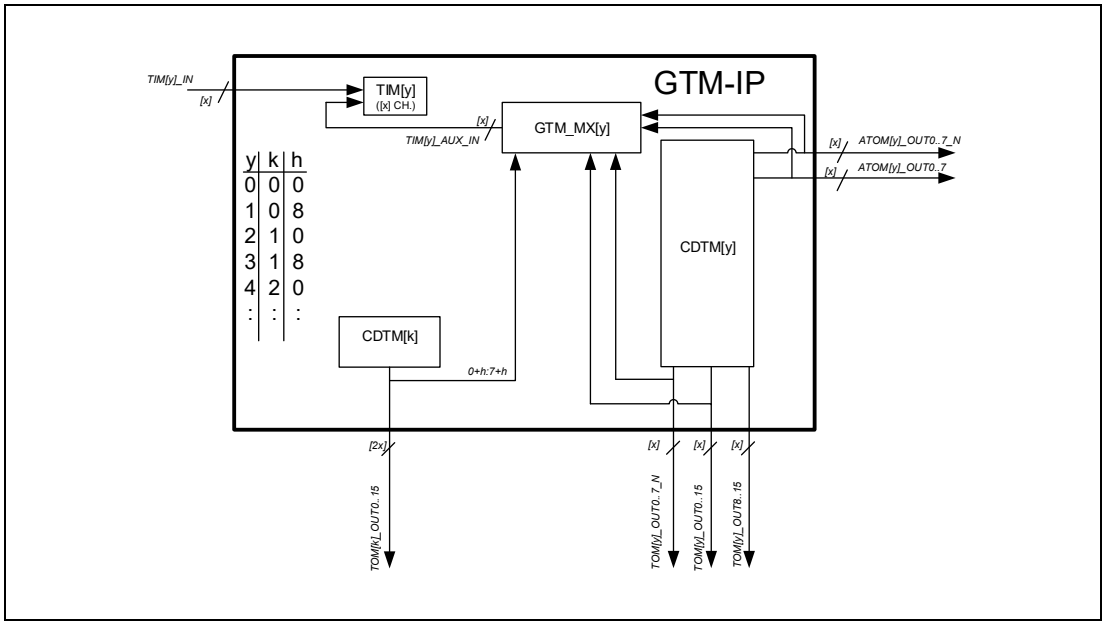


Figure 38.4 GTM-IP signal multiplex

The next diagram gives an overview of the connectivity for different configuration of GTM global bit SRC_IN_MUX of register GTM_CFG and the cluster configuration register CCM[y]_TIM_AUX_IN_SRC. The source selection is defined per channel with the bit SRC_CH[x] and SEL_OUT_N_CH[x] in the register CCM[y]_TIM_AUX_IN_SRC.

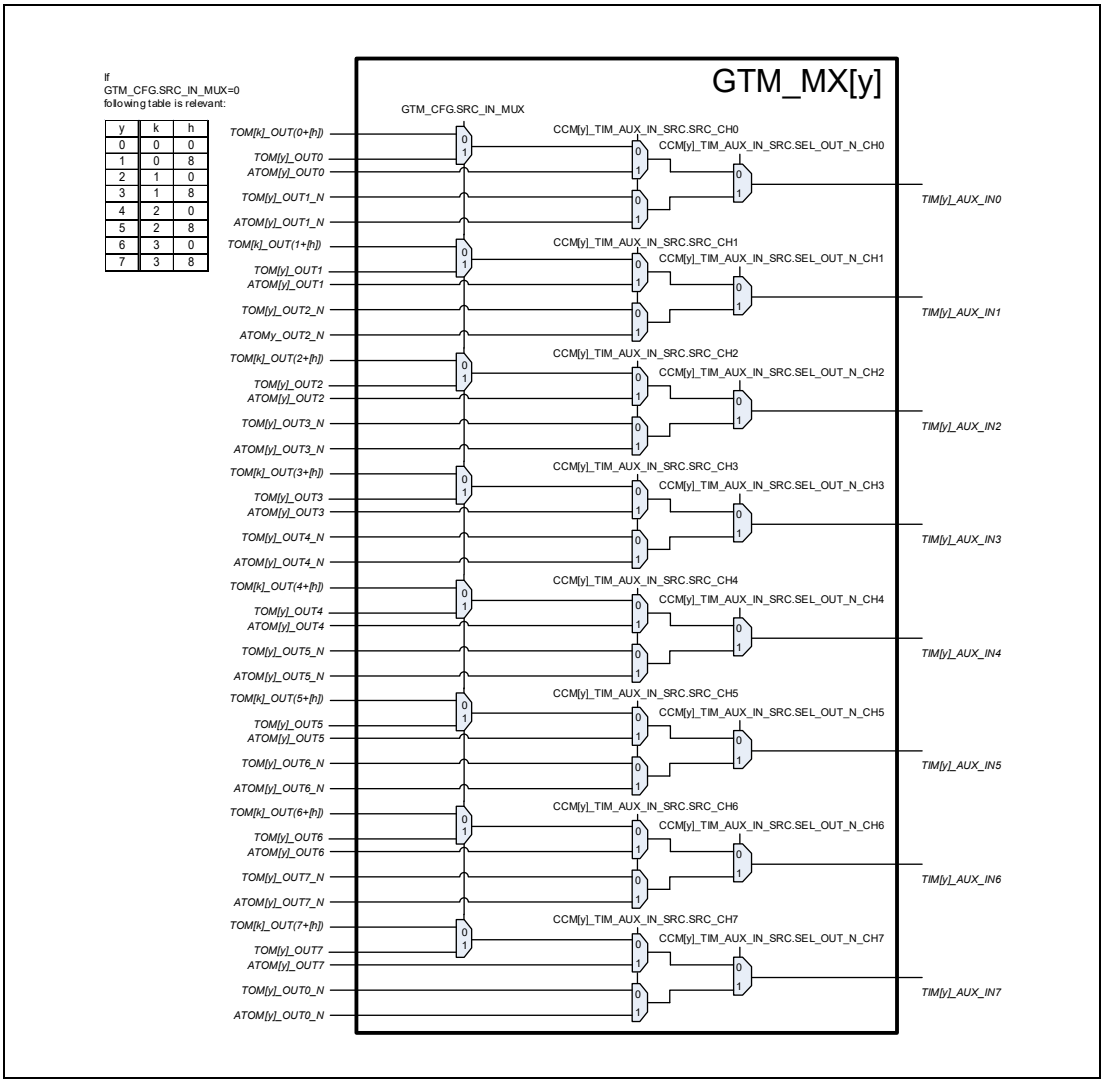


Figure 38.5 TIM auxiliary input multiplexing

The trigger out of TIM (i.e. the signals TIM[i]_EXT_CAPTURE(7:0) of each TIM instance i) are routed to ATOM instance [i] and TOM instance [i] with i=0...cITIM-1 (cITIM defines the number of available TIM instances, refer to device specific **Section 38.28, GTM Device 358**). This TIM trigger can be used to trigger inside the ATOM or TOM instance either a channel or the global control register of AGC or TGC0/TGC1 unit.

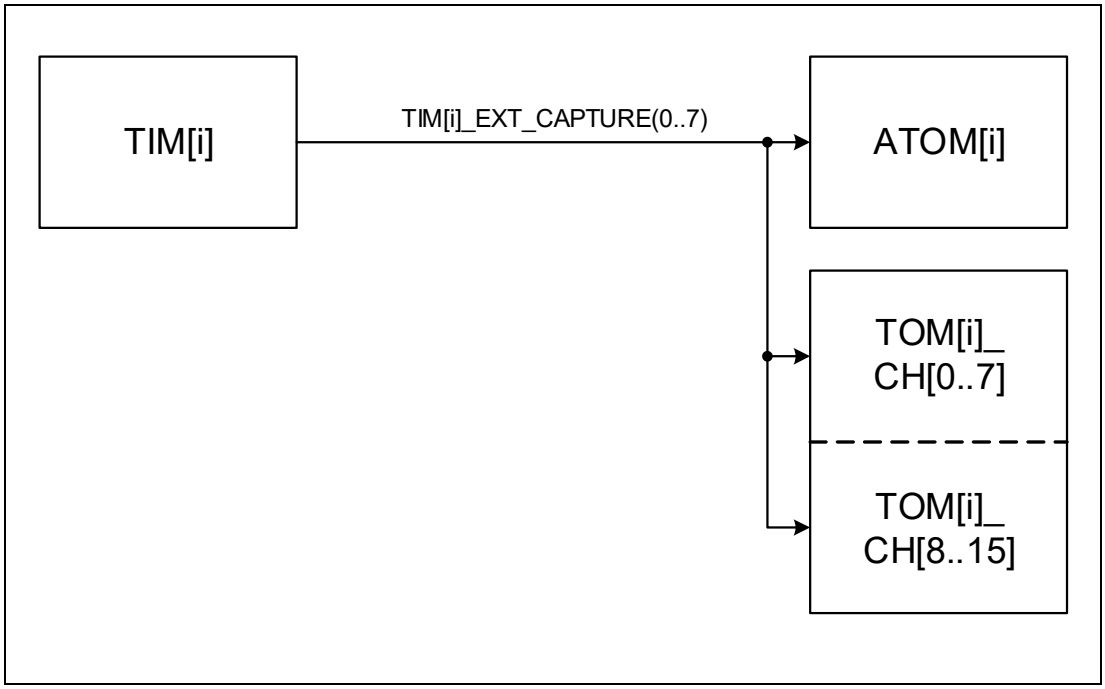


Figure 38.6 TIM external capture forwarding to TOM and ATOM

The trigger out of TIM (i.e. the signals TIM[i]_EXT_CAPTURE(7:0) of each TIM instance i) are additionally routed to the MCS instance [i]. This trigger forwarding can be enabled by register CCM[i]_EXT_CAP_EN.

38.5.2 GTM-IP Interfaces

In general the GTM-IP can be divided into four interface groups. Two interface groups represent the ports of the GTM-IP where incoming signals are assembled and outgoing signals are created. These interfaces are therefore connected to the GTM-IP input sub-module TIM and to the GTM-IP output sub-modules DTM.

Another interface is the bus master interface where the GTM-IP can be connected to the SoC system bus and actively drive read and write operations onto the SoC system bus. This generic bus interface is described in more detail in **Section 38.5.2.1, GTM-IP Generic Bus Interface (AEI)**.

Another interface is the bus slave interface where the GTM-IP can be connected to the SoC system bus to allow access to the GTM internal resources. Two options are provided for the slave interface: A generic bus interface being described in more detail in **Section 38.5.2.1**, and an AXI slave described in **Section 38.5.2.2, AXI Slave**.

The last interface is the interrupt controller interface. The GTM-IP provides several interrupt lines coming from the various sub-modules. These interrupt lines are concentrated inside the ICM and have to be adapted to the dedicated micro controller environment where each interrupt handling can look different. The interrupt concept is described in more detail in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.5.2.1 GTM-IP Generic Bus Interface (AEI)

The GTM-IP is equipped with a generic bus interface that can be widely adapted to different SoC bus systems. This generic bus interface is called AE-Interface (AEI). The adaptation of the AEI to SoC buses is typically done with a bridge module translating the AEI signals to the SoC bus signals of the silicon vendor. The AEI bus signals are depicted in the following table:

Table 38.13 AEI bus signals

Signal name	I/O	Description	Bit width
AEI_SEL	I	GTM-IP select line	1
AEI_ADDR	I	GTM-IP address	32
AEI_PIPE	I	AEI Address phase signal	1
AEI_W1R0	I	Read/Write access	1
AEI_WDATA	I	Write data bus	32
AEI_RDATA	O	Read data bus	32
AEI_READY	O	Data ready signal	1
AEI_STATUS	O	AEI Access status	2

The AEI Status Signal may drive one of the following values:

Table 38.14 AEI Status Signal

AEI_STATUS	Description
00_B	No Error
01_B	Illegal Byte Addressing
10_B	Illegal Address Access
11_B	Unsupported Address

The signal value 00_B is returned if no error occurred during AEI access.

The signal value 01_B is returned if the bus address is not an integer multiple of 4 (byte addressing).

The signal value 11_B is returned if the address is not handled in the GTM.

The signal value 10_B is returned if the written register contains a protected bit field and the protection is active or if the register is temporarily not writable because of sub-module internal state or the clock of the relevant cluster is disabled or if a bit field which is not defined as read-only is written to a value unequal to "all zero" while the device dependency attribute of that bit field evaluates to false.

In case of an illegal write access signaled by status 10_B the register will not be modified.

NOTE

Exception for register CMU_CLK_CTRL. In case of write access signalled by aei_status 10_B the register will be modified each completely disabled bit.

CAUTION

aei_status 01_B is not supported in RH850/U2A-EVA Group.
When aei_status 10_B or 11_B is returned, error response will be notified to access bus master.

Reading registers will never return status 10_B.

The detailed list of register addresses with return status 10_B can be found in **Section 38.28, GTM Device 358**.

If a bit field of a register is defined as reserved, any write access to this bit field has no side effect and a read access to this bit field will always return the value 0.

If one or more encoding values of a bit field are defined as reserved values and nothing else is specified, the following default behavior is defined: The written value can be read back and the entire behavior of the circuit is undefined.

38.5.2.2 AXI Slave

(1) Functional Characteristics

The AXI slave to AEI split protocol adapter implements an AXI compliant slave interface and converts the access to be AEI split compatible. In addition it implements checks for the correctness of the AXI transaction within the given constraints of the AEI.

The block can be configured for different use-cases with generics. The supported generics are described in the following table:

- Full AMBA AXI 3.0 compliant
- Only 32 bit accesses
- 32 bit address
- 32 or 64 bit data
- No byte/halfword support
- Only aligned accesses
- Alignment check
- Single ID support
- No interleaved ID accesses
- Single burst address counter

- ID check for write data is executed
- Cache, protection and lock signals are not available (are ignored)
- Single clock operation
- Single low active asynchronous reset signal
- AEI split protocol
- Additional AEI sideband signals
- Support for two AXI Debugger IDs

(2) AXI to AEI Split Adapter Block

(a) Functional View

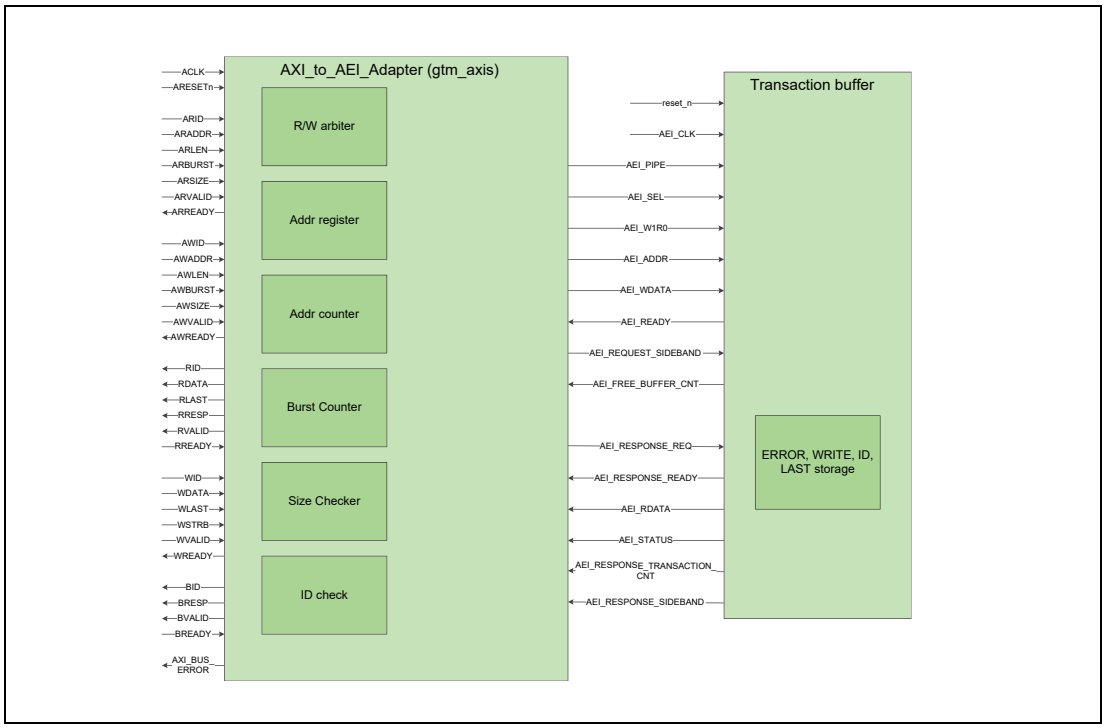


Figure 38.7 Block Diagram AXI slave

Figure 38.7 shows the basic modules of the adapter block. The transaction buffer is not part of the adapter block, but part of the block the adapter is integrated in.

(b) Entity Table

The block has an AXI slave interface on the "left" side and an AEI split master interface on the "right" side. **Table 38.15** described the block entity pins.

Table 38.15 Pin List (1/2)

Port Name	Width	Direction	Description
General AXI Signals			
ACLK	1	input	AXI reference clock
ARESETn	1	input	AXI reset signal (low active)
Read Address Channel			
ARID	Generic	input	AXI transaction ID, ID width can be selected via generic
ARADDR	32	input	AXI read address
ARLEN	4	input	AXI read burst length
ARBURST	2	input	AXI read burst type
ARSIZE	3	input	AXI read size, must always be b010, as we only allow word accesses
ARVALID	1	input	AXI read valid
ARREADY	1	output	AXI read ready
Write Address Channel			
AWID	Generic	input	AXI transaction ID, ID width can be selected via generic
AWADDR	32	input	AXI write address
AWLEN	4	input	AXI write burst length
AWBURST	2	input	AXI write burst type
AWSIZE	3	input	AXI write data size (must always be b010, as we only support 32bit accesses)
AWVALID	1	input	AXI write data valid
AWREADY	1	output	AXI write data ready
Read Data Channel			
RID	Generic	output	AXI transaction ID, ID width can be selected via generic
RDATA	Generic	output	AXI read data
RLAST	1	output	AXI real last in a burst signal
RRESP	2	output	AXI read response
RVALID	1	output	AXI read data valid
RREADY	1	input	AXI read data ready
Write	Data	Channel	
WID	Generic	input	AXI transaction ID, ID width can be selected via generic
WDATA	Generic	input	AXI write data
WLAST	1	input	AXI write last in a burst signal
WSTRB	4	input	AXI byte strobes (must by all high, as we only support full word access)
WVALID	1	input	AXI write data valid

Table 38.15 Pin List (2/2)

Port Name	Width	Direction	Description
WREADY	1	output	AXI write data ready
Write Response Channel			
BID	Generic	output	AXI transaction ID, ID width can be selected via generic
BRESP	2	output	AXI write response
BVALID	1	output	AXI write response valid
BREADY	1	input	AXI wire response ready
AEI Master Signals			
AEI_PIPE	1	output	AEI pipe signal (always 1)
AEI_SEL	1	output	AEI select
AEI_W1R0	1	output	AEI write/read selection
AEI_DEBUG_ACCESS	1	output	AEI debugger access
AEI_ADDR	32	output	AEI address
AEI_WDATA	32	output	AEI write data
AEI_READY	1	input	AEI ready
AEI_REQUEST_SIDEBAND	Generic	output	AEI request sideband signals, the slave shall provide this value when providing the related response
AEI_FREE_BUFFER_CNT	Generic	input	AEI number of free request entries
AEI_RESPONSE_REQ	1	output	AEI response request
AEI_RESPONSE_READY	1	input	AEI response ready
AEI_RDATA	32	input	AEI read data
AEI_STATUS	2	input	AEI transaction status AEI number of available response transactions Same width as AEI_FREE_CNT
AEI_RESPONSE_SIDEBAND	Generic	input	AEI response sideband signals. Here the slave shall provide the value from the related request
General AXI Signals			
AXI_BUS_ERROR	1	output	Illegal AXI transaction detected (system sideband signal)

38.5.3 ARU Routing Concept

One central concept of the GTM-IP is the routing mechanism of the ARU sub-module for data streams. Each data word transferred between the ARU and its connected sub-module is 53-bit wide. It is important to understand this concept in order to use the resources of the GTM-IP effectively. Each module that is connected to the ARU may provide an arbitrary number of ARU write channels and an arbitrary number of ARU read channels. In the following, the ARU write channels are named data sources and the ARU read channels are named data destinations.

The concept of the ARU intends to provide a flexible and resource efficient way for connecting any data source to an arbitrary data destination. In order to save resource costs, the ARU does not implement a switch matrix, but it implements a data router with serialized connectivity providing the same interconnection flexibility. **Figure 38.8** shows the ARU data routing principle. Data sources are marked with a green rectangle and the data destinations are marked with yellow rectangles. The dashed lines in the ARU depict the configurable connections between data sources and data destinations. A connection between a data source and a data destination is also called a data stream.

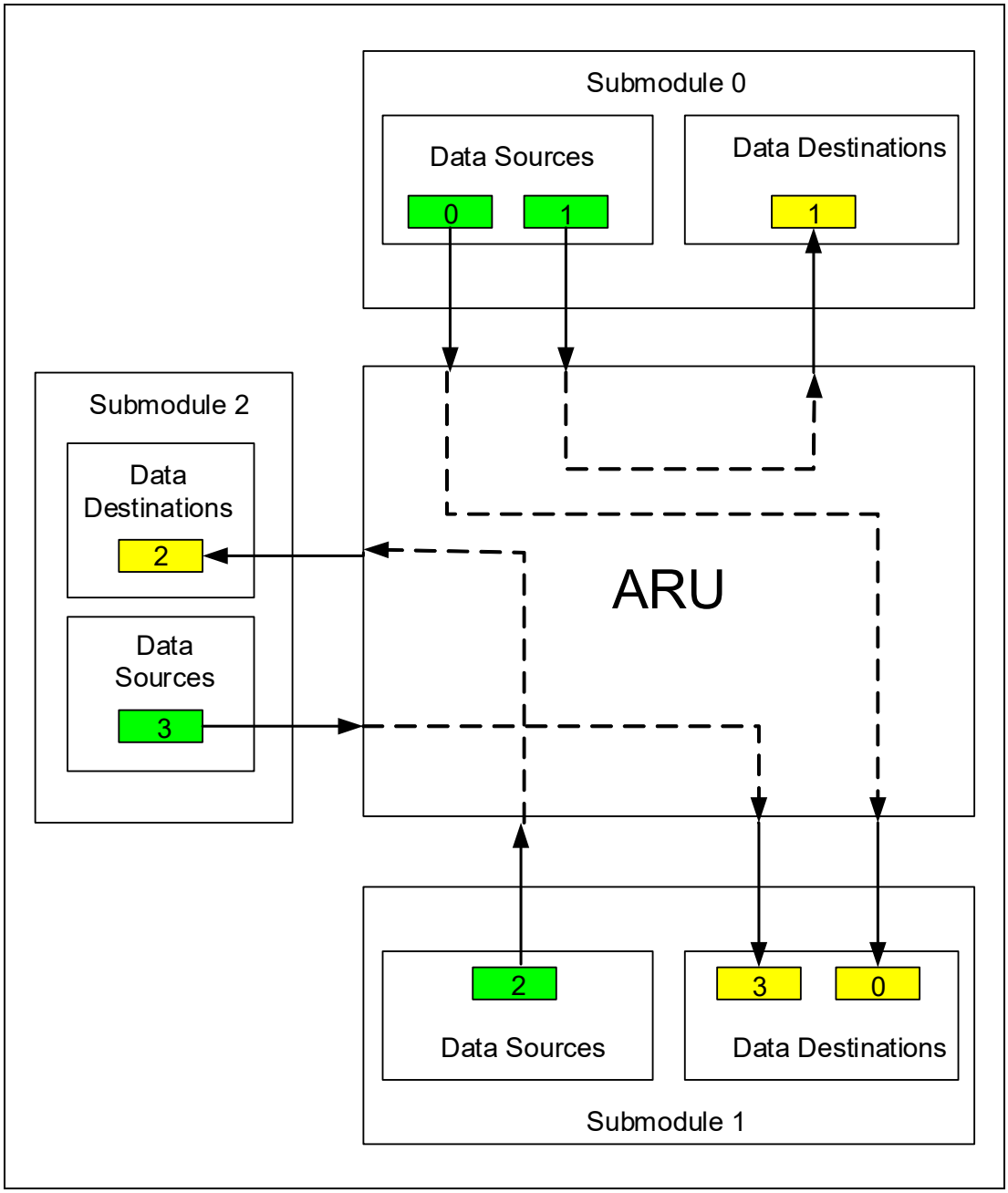


Figure 38.8 Principle of data routing using ARU

The configuration of the data streams is realized according to the following manner: Each data source has its fixed and unique source address: The ARU read ID. The fixed address of each data source is pointed out by the numbers in the green boxes of **Figure 38.8**. The address definitions of all available data sources in the GTM-IP can be obtained from **Table 38.520, ARU Write Addresses**. The connection from a specific data source to a specific data destination is defined by configuring the corresponding address of a data source (i.e. the ARU read ID) in the desired data destination. The configured address of each data destination is pointed out by the numbers in the yellow boxes of **Figure 38.8**.

Normally, the destination is idle and waits for data from the source. If the source offers new data, the destination does a destructive read, processes the data and goes idle again. The same data is never read twice.

There is one sub-module for which this destructive read access does not hold. This is the BRC sub-module configured in Maximal Throughput Mode. For a detailed description of this module, refer to **Section 38.8, Broadcast Module (BRC)**.

The functionality of the ARU is as follows: The ARU sequentially polls the data destinations of the connected modules in a round−robin order.

If a data destination requests new data from its configured data source and the data source has data available, the ARU delivers the data to the destination and it informs both, the data source and destination that the data is transferred. The data source marks the delivered ARU data as invalid which means that the destination consumed the data.

It should be noted that each data source should only be connected to a single data destination. This is because the destinations consume the data. If two destinations would reference the same source one destination would consume the data before the other destination could consume it. Since the data transfers are blocking, the second destination would block until it receives new data from the source. If a data source should be connected to more than one data destination the sub-module Broadcast (BRC) has to be used. On the other hand, the transfer from a data source to the ARU is also blocking, which means that the source channel can only provide new data to the ARU when an old data word is consumed by a destination. In order to speed up the process of data transfers, the ARU handles two different data destinations in parallel.

Following table gives an overview about the number of data destinations and data sources of each GTM-IP instance type.

Table 38.16 ARU source and destination address count per instance

Sub-module	Number of data sources per instance	Number of data destinations per instance
ARU	1	0
DPLL	24	24
TIM	8	0
MCS	24	8
BRC	22	12
TOM	0	0
ATOM	8	8
DTM	0	0
PSM	8	8
ICM	0	0
CPM	0	0
MON	0	0
CCM	0	0

(1) ARU Round Trip Time

The ARU uses a round-robin arbitration scheme with a fixed round trip time for all connected data destinations. This means that the time between two adjacent read requests resulting from a data destination channel always takes the round trip time, independently if the read request succeeds or fails.

(2) ARU Blocking Mechanism

Another important concept of the ARU is its blocking mechanism that is implemented for transferring data from a data source to a data destination. This mechanism is used by ARU connected sub-modules to synchronize the sub-modules to the routed data streams. **Figure 38.9** explains the blocking mechanism.

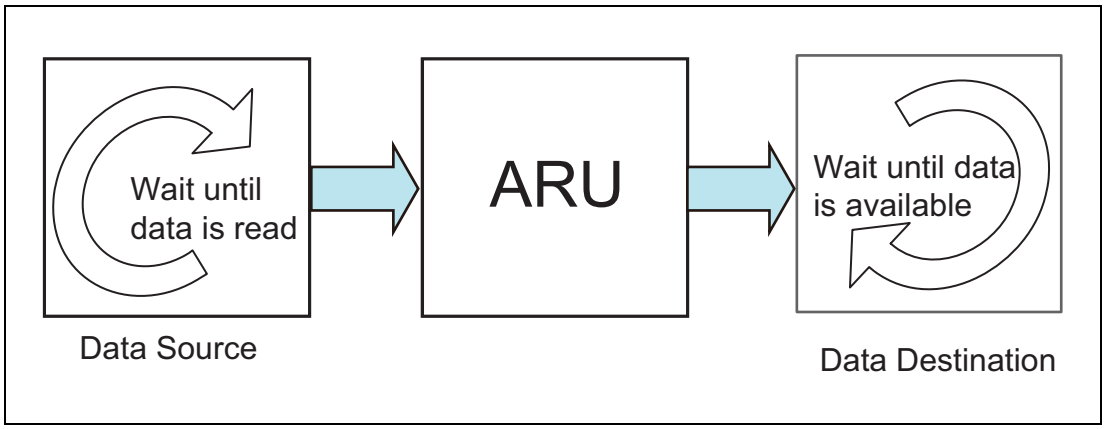


Figure 38.9 Graphical representation of ARU blocking mechanism

If a data destination requests data from a data source over the ARU but the data source does not have any data yet, it has to wait until the data source provides new data. In this case the sub-module that owns the data destination may perform other tasks. When a data source produces new data faster than a data destination can consume the data the source raises an error interrupt and signals that the data could not be delivered in time. The new data is marked as valid for further transfers and the old data is overwritten.

In any case the round trip time for the ARU has a fixed reset value for a specific device configuration. The end value of the roundtrip counter can be changed with a configuration register ARU_CADDR_END inside the ARU. For more details see the ARU specific chapter. Refer also to device specific **Section 38.28, GTM Device 358** of this specification for detailed information.

It is possible to reset the ARU roundtrip counter ARU_CADDR manually synchronous to CMU clock enable from configuration register inside CMU module. Refer to CMU specific chapter for more details. One exception is the BRC sub-module when configured in Maximal Throughput Mode. Refer to **Section 38.8, Broadcast Module (BRC)** for a detailed description.

38.5.4 GTM-IP Clock and Time Base Management (CTBM)

Inside the GTM-IP several subunits are involved in the clock and time base management of the whole GTM. **Figure 38.10** shows the connections and sub blocks involved in these tasks. The sub blocks involved are called Clock and Time Base Management (CTBM) modules further on.

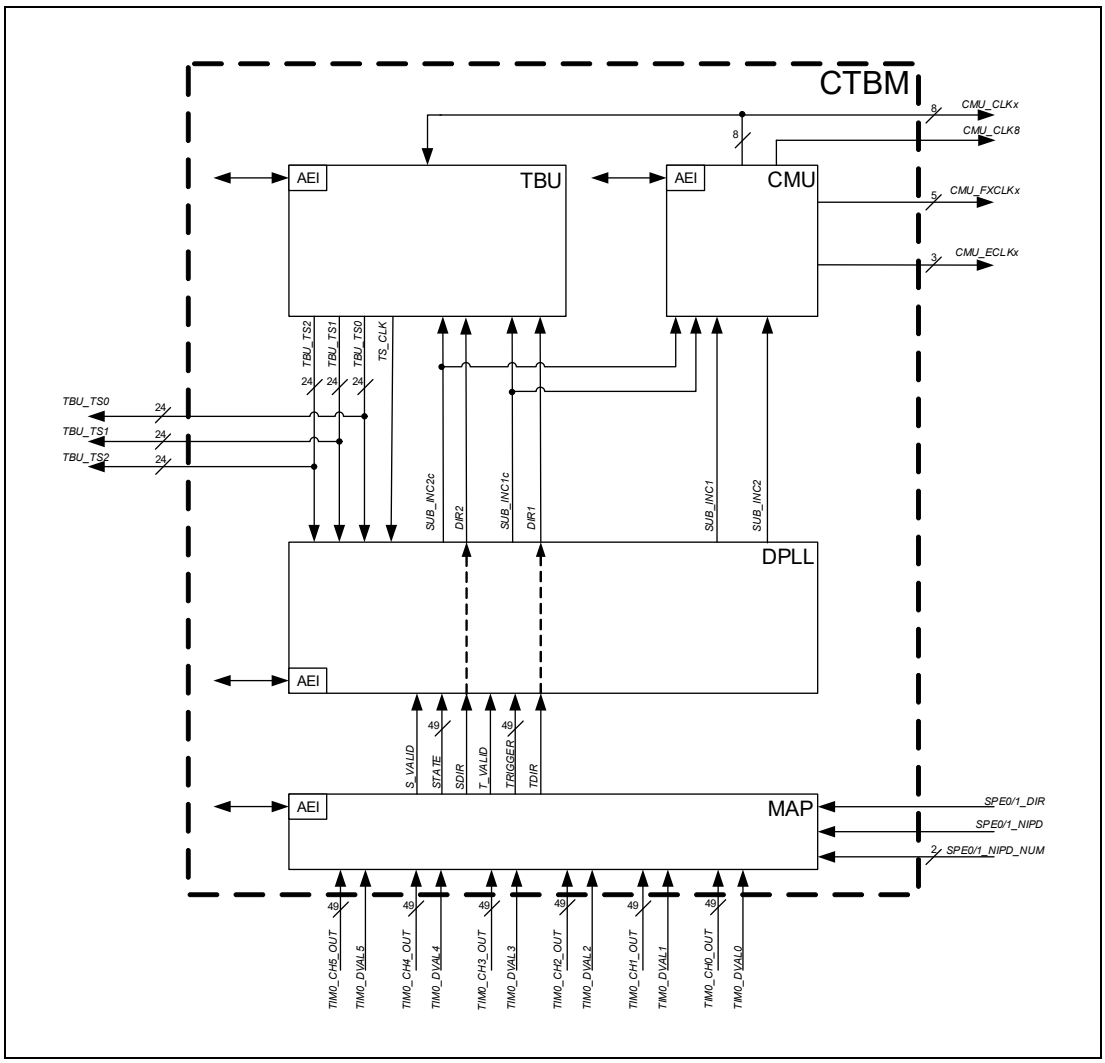


Figure 38.10 GTM-IP Clock and time base management architecture

One important module of the CTBM is the Clock Management Unit (CMU) which generates up to 14 clocks for the sub-modules of the GTM and up to three GTM external clocks CMU_ECLK[z] (z: 0 to 2). For a detailed description of the CMU functionality and clocks, refer to **Section 38.12, Clock Management Unit (CMU)**.

The five CMU_FXCLK[y] (y: 0 to 4) clocks are used by the TOM sub-module for PWM generation.

A maximum of nine CMU_CLK[x] (x: 0 to 8) clocks are used by other sub-modules of the GTM for signal generation.

Inside the Time Base Unit (TBU) one of CMU_CLK[x] (x: 0 to 7) clocks is used per channel to generate a common time base for the GTM.

Besides the CMU_CLK[x] signals, the TBU can use the compensated SUB_INC[i]c (i: 1, 2) signals coming from the DPLL sub-module for time base generation. This time base then typically represents an angle clock for an engine management system. For the meaning of compensated (SUB_INC[i]c) and

uncompensated (SUB_INC[i]) DPLL signals, refer to the DPLL **Section 38.22, Digital PLL Module (DPLL)**. The SUB_INC[i]c signals in combination with the two direction signal lines DIR[i] the TBU time base can be controlled to run forwards or backwards. The TBU functionality is described in **Section 38.14, Time Base Unit (TBU)**.

The TBU sub-module generates the three time base signals TBU_TS0, TBU_TS1 and TBU_TS2 which are widely used inside the GTM as common time bases for signal characterization and generation.

Besides the time base 1 and 2 which may represent a relative angle clock for an engine management system it is helpful to have an absolute angle clock for CPU/MCS internal angle algorithm calculations. This absolute angle clock is represented by the TBU base 3. The TBU channel 0 up to 2 are widely used inside the GTM as common time (channel 0, 1 and/or 2) or angle (channel 1 and/or 2) bases for signal characterization and generation. The TBU channel 3 is only configurable and readable by MCS0 or CPU.

As stated before, the DPLL sub-module provides the four clock signals SUB_INC[i] and SUB_INC[i]c which can be seen as a clock multiplier generated out of the two input signal vectors TRIGGER and STATE coming from the MAP sub-module. For a detailed description of the DPLL functionality refer to **Section 38.22, Digital PLL Module (DPLL)**.

The MAP sub-module is used to select the TRIGGER and STATE signals for the DPLL out of six input signals coming from TIM0 sub-module.

Besides this, the MAP sub-module is able to generate a TDIR (TRIGGER Direction) and SDIR (STATE Direction) signal for the DPLL and TBU coming from the SPE0 and SPE1 signal lines. The direction signals are generated out of a defined input pattern. For a detailed description of the MAP sub-module, refer to **Section 38.21, TIM0 Input Mapping Module (MAP)**.

38.5.4.1 Cyclic Event Compare

With the time base module (TBU) the GTM provides three counters, where the counter of TBU_CH0 represents a time and the counter TBU_CH1 and TBU_CH2 may represent a time (if clock source is CMU_CLK generated inside CMU) or an angle (if clock source is a DPLL sub_inc signal provided via CMU).

From application point of view it is necessary to divide the cyclic event counter representing time or angle into two parts, the past and the future. The border of past/future is a moving border depending on current time or angle value. The cyclic event counting and the moving border of past/future is depicted in the figure below.

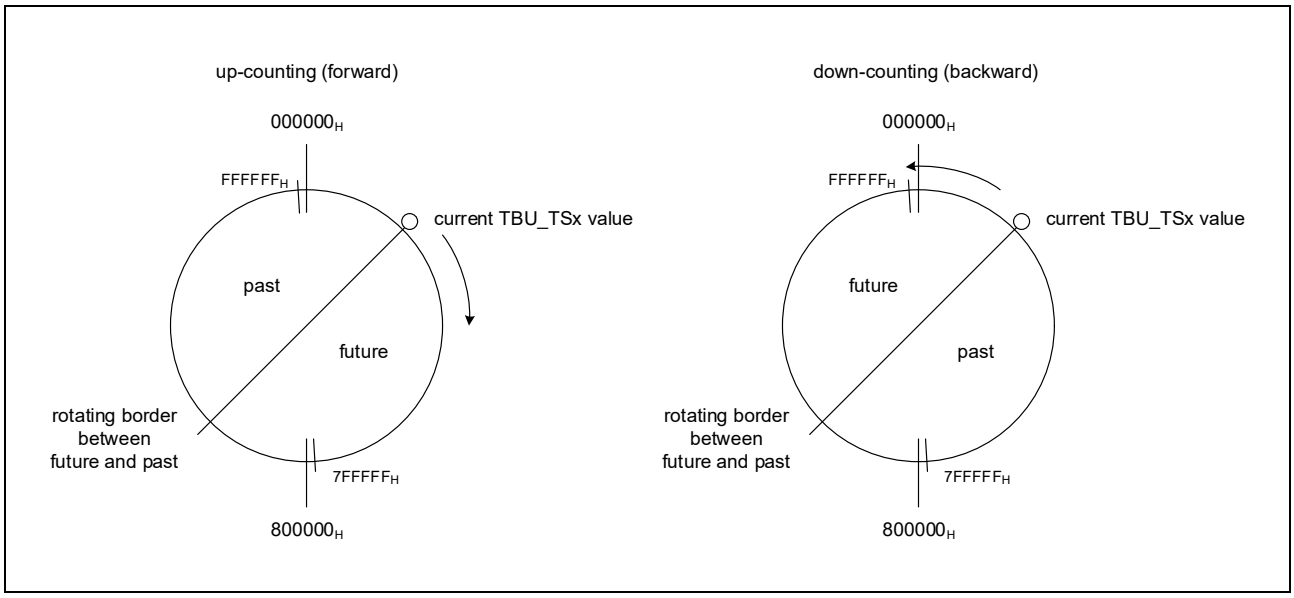


Figure 38.11 Cyclic event counter representing time or angle

Inside different sub-modules of GTM a greater−equal compare (in case of up-counting) or a less-equal compare (in case of down-counting) against a TBU base value (representing time or angle) always means that it is checked if the reference value is in relation to the current TBU value in the future or in the past.

38.5.5 GTM-IP Interrupt Concept

The sub-modules of the GTM-IP can generate thousands of interrupts on behalf of internal events. This high amount of interrupts is combined inside the Interrupt Concentrator Module (ICM) into interrupt groups. In these interrupt groups the GTM-IP sub-module interrupt signals are bundled to a smaller set of interrupts. From these interrupt sets, a smaller amount of interrupt signals is created and signaled outside of the GTM-IP as a signal GTM_<MOD>_IRQ, where <MOD> identifies the name of the corresponding GTM-IP sub-module.

Moreover, each output signal GTM_<MOD>_IRQ has a corresponding input signal GTM_<MOD>_IRQ_CLR that can be used for clearing the interrupts. These input signals can be used by the surrounding micro controller system as:

- acknowledge signal from a DMA controller
- validation signal from ADC
- clear signal from an GTM-external interrupt controller to do an atomic clear while entering an ISR routine

The controlling of the individual interrupts is done inside the sub-modules. If a sub-module consists of several sub-module channels that are most likely to work independent from each other (like TIM, PSM, MCS, TOM, and ATOM), each sub-module channel has its own interrupt control and status register set, named as interrupt set in the following. Other sub-modules (SPE, ARU, DPLL, BRC, CMP and global GTM functionality) have a common interrupt set for the whole sub-module.

The interrupt set consists of four registers: The IRQ_EN register, the IRQ_NOTIFY register, the IRQ_FORCINT register, and the IRQ_MODE register. While the registers IRQ_EN, IRQ_NOTIFY, and IRQ_FORCINT signalize the status and allow controlling of each individual interrupt source within an interrupt set, the register IRQ_MODE configures the interrupt mode that is applied to all interrupts that belong to the same interrupt set.

In order to support a wide variety of micro controller architectures and interrupt systems with different interrupt signal output characteristics and internal interrupt handling the following four modes can be configured:

- Level mode
- Pulse mode
- Pulse-Notify mode
- Single-Pulse mode

These interrupt modes are described in more details in the following subsections.

The register IRQ_EN allows the enabling and disabling of an individual interrupt within an interrupt set. Independent of the configured mode, only enabled interrupts can signalize an interrupts on its signal GTM_<MOD>_IRQ.

The register IRQ_NOTIFY collects the occurrence of interrupt events. The behavior for setting a bit in this register depends on the configured mode and thus it is described later on in the mode descriptions. Independent of the configured mode any write access with value ‘1’ to a bit in the register IRQ_NOTIFY always clears the corresponding IRQ_NOTIFY bit.

Moreover, the enabling of a disabled interrupt source with a write access to the register IRQ_EN also clears the corresponding bit in the IRQ_NOTIFY register but only if the error interrupt source EIRQ_EN is disabled. However, if the enabling of a disabled interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the register IRQ_NOTIFY is not cleared.

Additionally, each write access to the register IRQ_MODE, clears all bits in the IRQ_NOTIFY register. It should be notified that the clearing of IRQ_NOTIFY is applied independently of the written data (e.g. no mode change).

Thus, a secure way for reconfiguring the interrupt mode of an interrupt set, is to disable all interrupts of the interrupt set with the register IRQ_EN, define the new interrupt mode by writing register IRQ_MODE, followed by enabling the desired interrupts with the register IRQ_EN.

Thus, a secure way for reconfiguring the interrupt mode of an error interrupt set, is to disable all error interrupts of the error interrupt set with the register EIRQ_EN, define the new interrupt mode by writing register IRQ_MODE, followed by enabling the desired error interrupts with the register EIRQ_EN.

The register IRQ_FORCINT is used by software for triggering individual interrupts with a write access with value ‘1’. Since a write access to IRQ_FORCINT only generates a single pulse, IRQ_FORCINT is not implemented as a true register and thus any read access to IRQ_FORCINT always results with a value of ‘0’.

The mechanism for triggering interrupts with IRQ_FORCINT is globally disabled after reset. It has to be explicitly enabled by clearing the bit RF_PROT in the register GTM_CTRL (see **Section 38.5.9.3, GTM_CTRL**).

For the modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE and CMP each interrupt may configured to raise instead of the normal interrupt an error interrupt if enabled by the corresponding error interrupt enable bit in register EIRQ_EN. It is possible for one source to enable the normal interrupt and the error interrupt in parallel. Because both interrupt clear signals could reset the notify bit this is expected to cause problems in a system and therefore it is strongly recommended to not enable both interrupt types at the same point in time.

Similar to enabling an interrupt, the enabling of a disabled error interrupt source with a write access to the register EIRQ_EN also clears the corresponding bit in the IRQ_NOTIFY register only if the interrupt source IRQ_EN is disabled. However, if the enabling of a disabled error interrupt is simultaneous to an incoming interrupt event, the interrupt event is dominant and the register IRQ_NOTIFY is not cleared.

All enabled error interrupts are OR-combined inside the ICM and assigned to the dedicated GTM port gtm_err_irq. A corresponding input gtm_err_irq_clr allows the reset of this error interrupt from outside the GTM (hardware clear).

To be able to detect the module source of the error interrupt the ICM provides the register ICM_IRQG_MEI.

The error interrupt causing channel can be determined for the module FIFO by evaluating the ICM register ICM_IRQG_CEI0.

The error interrupt causing channel can be determined for the modules TIM by evaluating the ICM register ICM_IRQG_CEI1 to 2.

The error interrupt causing channel can be determined for the modules MCS with all possible channel by evaluating the ICM register

ICM_IRQG_MCS[i]_CEI. In case of usage only the first 8 channels of each MCS the error interrupt causing channel can be determined by evaluating the ICM register ICM_IRQG_CEI3 to 4.

38.5.5.1 Level interrupt mode

The default interrupt mode is the Level Interrupt Mode. In this mode each occurred interrupt event is collected in the register IRQ_NOTIFY, independent of the corresponding enable bit of register IRQ_EN and EIRQ_EN.

An interrupt event, which is defined as a pulse on the signal Int_out of **Figure 38.12**, may be triggered by the interrupt source of the sub-module or by software performing a write access to the corresponding register IRQ_FORCINT, with a disabled bit RF_PROT in register GTM_CTRL.

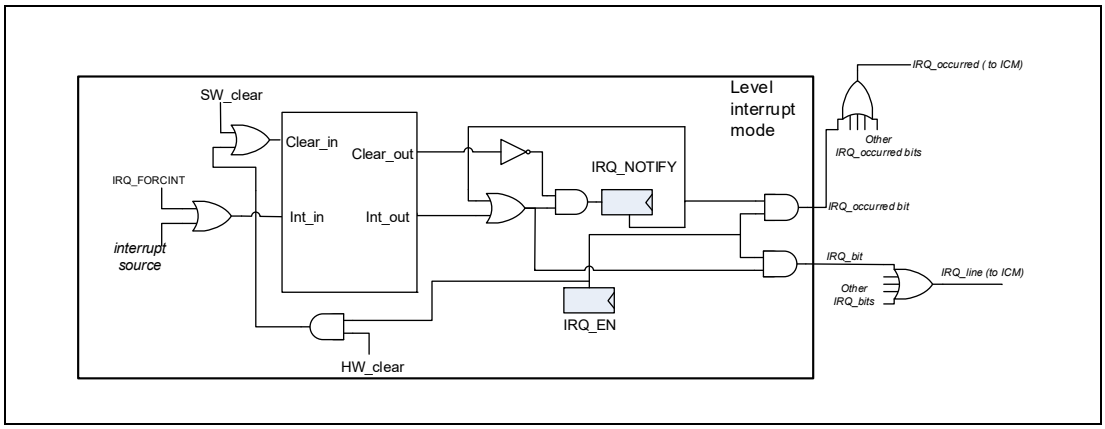


Figure 38.12 Level interrupt mode scheme

A collected interrupt bit in register IRQ_NOTIFY may be cleared by a clear event, which is defined as a pulse on signal Clear_out of **Figure 38.12**. A clear event can be performed by writing '1' to the corresponding bit in the register IRQ_NOTIFY leading to a pulse on signals SW_clear.

A clear event may also result from an externally connected signal GTM_<MOD>_IRQ_CLR, which is routed to the signal HW_clear of **Figure 38.12**. However, the hardware clear mechanism is only possible, if the corresponding interrupt is enabled by register IRQ_EN.

As **Table 38.17, Priority of Interrupt Events and Clear Events** shows, interrupt events are dominant in the case of a simultaneous interrupt event and clear event.

Table 38.17 Priority of Interrupt Events and Clear Events

Int_in	Clear_in	Int_out	Clear_out
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

As shown in **Figure 38.12** an occurred interrupt event is signaled as a constant signal level with value 1 to the signal IRQ_bit, if the corresponding interrupt is enabled in register IRQ_EN.

With exception of the sub-modules ARU and DPLL, the signal IRQ_bit is OR-combined with the neighboring IRQ_bit signals of the same interrupt set and they are routed as a signal IRQ_line to the interrupt concentrator module (ICM). The interrupt signals IRQ_bit of the sub-modules DPLL and ARU are routed directly as a signal IRQ_line to the sub-module ICM. In some cases (sub-modules TOM and ATOM) the ICM may further OR-combine several IRQ_line signals to an outgoing interrupt signal GTM_<MOD>_IRQ. In the other cases the IRQ_line signals are directly connected to the outgoing signals GTM_<MOD>_IRQ, within the sub-module ICM.

The signal IRQ_occurred is connected in a similar way as the signal IRQ_line, however this signal is used for monitoring the interrupt state of the register IRQ_NOTIFY in the registers of the ICM.

The additional error interrupt enable mechanism for level interrupt is shown below.

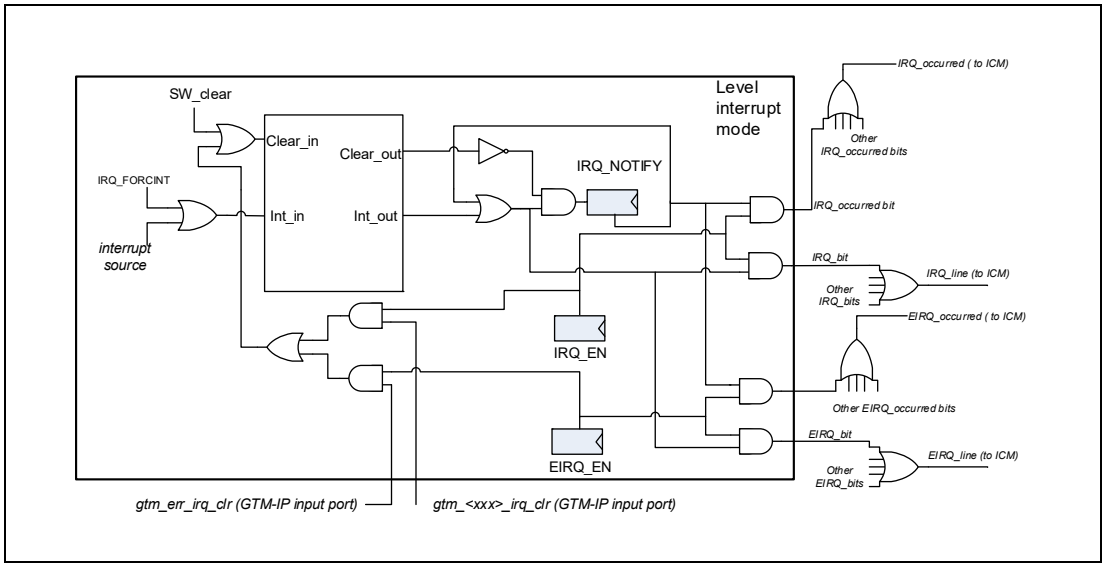


Figure 38.13 Level interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

A collected interrupt bit in register IRQ_NOTIFY may be cleared by a clear event, which is defined as a pulse on signal Clear_out of **Figure 38.13**. A clear event can be performed by writing '1' to the corresponding bit in the register IRQ_NOTIFY leading to a pulse on signals SW_clear. A clear event may also result from the externally connected signal gtm_<MOD>_irq_clr or gtm_err_irq_clr, which is routed as an HW_clear to Clear_in of **Figure 38.13**. However, the hardware clear mechanism is only possible, if the corresponding interrupt or error interrupt is enabled by register IRQ_EN or EIRQ_EN.

As it can be seen from the **Figure 38.13** an occurred interrupt event is signaled as a constant signal level with value 1 to the signal IRQ_bit, if the corresponding interrupt is enabled in register IRQ_EN.

CAUTION

Level mode is not supported in RH850/U2A-EVA Group.

38.5.5.2 Pulse interrupt mode

The Pulse interrupt mode behavior can be observed from **Figure 38.14**.

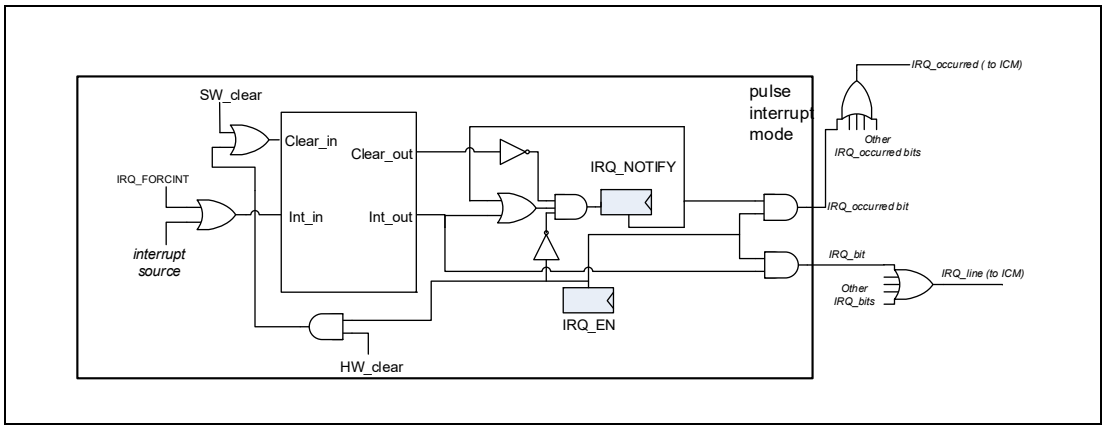


Figure 38.14 Pulse interrupt mode scheme

In Pulse Interrupt Mode each Interrupt Event will generate a pulse on the IRQ_bit signal if IRQ_EN is enabled. As it can be seen from the figure, the interrupt bit in IRQ_NOTIFY register is always cleared if IRQ_EN is enabled. However, if an interrupt is disabled in the register IRQ_EN, an occurred interrupt event is captured in the register IRQ_NOTIFY, in order to allow polling for disabled interrupts by software. Disabled interrupts may be cleared by an interrupt clear event.

In Pulse interrupt mode, the signal IRQ_occurred is always 0.

The additional error interrupt enable mechanism for pulse interrupt is shown below.

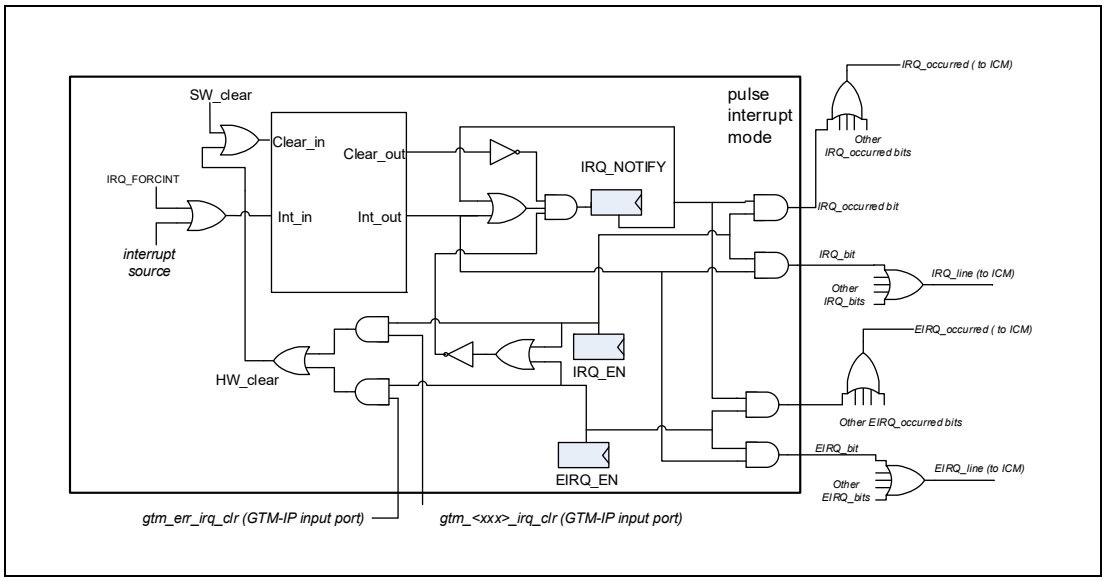


Figure 38.15 Pulse interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse Interrupt Mode each Interrupt Event will generate a pulse on the EIRQ_bit signal if EIRQ_EN is enabled.

As it can be seen from the figure, the interrupt bit in IRQ_NOTIFY register is always cleared if EIRQ_EN or IRQ_EN are enabled.

However, if an error interrupt is disabled in the register EIRQ_EN, an occurred error interrupt event is captured in the register IRQ_NOTIFY, in order to allow polling for disabled error interrupts by software.

Disabled error interrupts may be cleared by an error interrupt clear event.

In Pulse interrupt mode, the signal EIRQ_occurred is always 0.

38.5.5.3 Pulse-notify interrupt mode

In Pulse-notify Interrupt mode, all interrupt events are captured in the register IRQ_NOTIFY. If an interrupt is enabled by the register IRQ_EN, each interrupt event will also generate a pulse on the IRQ_bit signal. The signal IRQ_occurred will be high if interrupt is enabled in register IRQ_EN and the corresponding bit of register IRQ_NOTIFY is set. The Pulse-notify interrupt mode is shown in **Figure 38.16**.

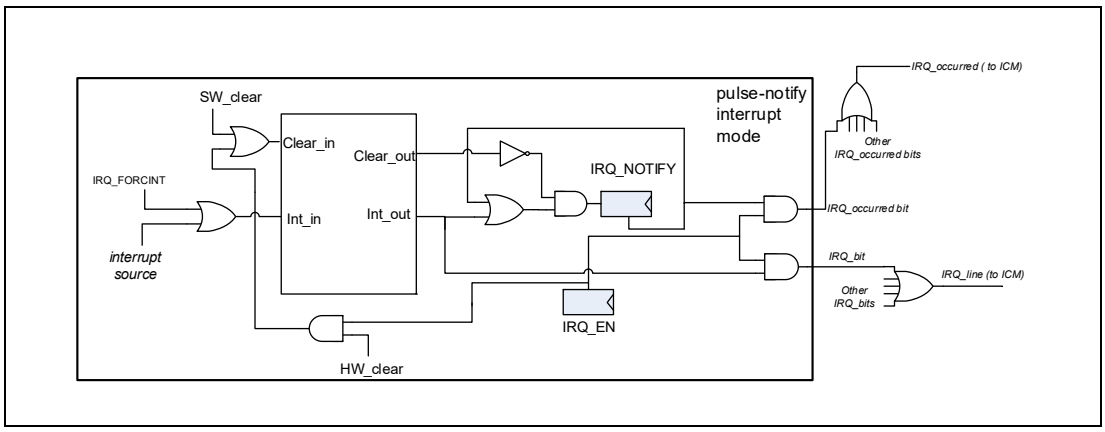


Figure 38.16 Pulse-notify interrupt mode scheme

The additional error interrupt enable mechanism for pulse-notify interrupt is shown below.

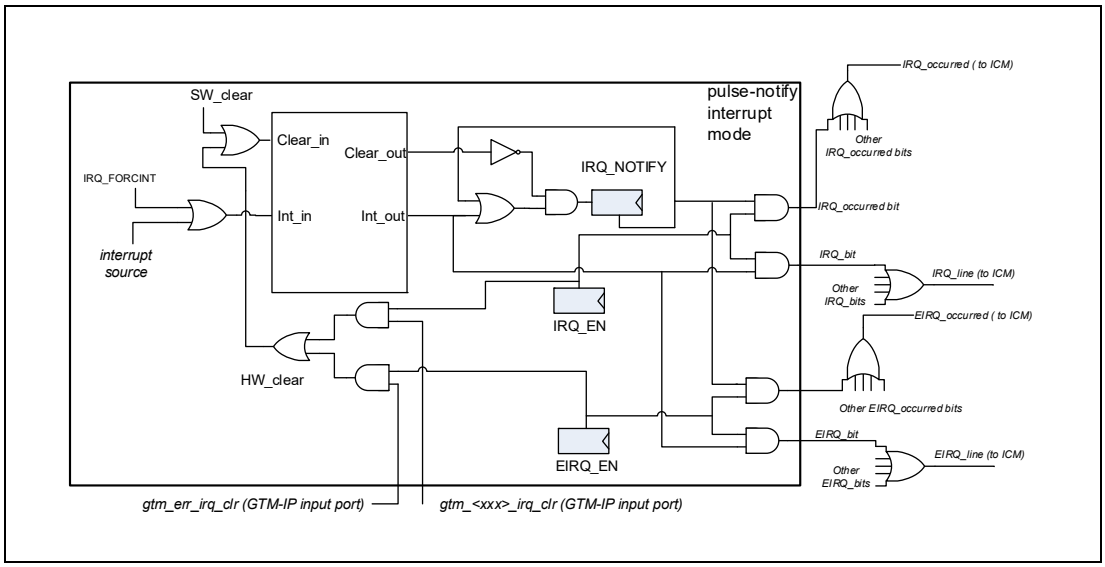


Figure 38.17 Pulse-notify interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Pulse-notify Interrupt mode, all error interrupt events are captured in the register IRQ_NOTIFY. If an error interrupt is enabled by the register EIRQ_EN, each error interrupt event will also generate a

pulse on the EIRQ_bit signal. The signal EIRQ_occurred will be high if error interrupt is enabled in register EIRQ_EN and the corresponding bit of register IRQ_NOTIFY is set. The Pulse-notify interrupt mode for error interrupts is shown in **Figure 38.17**.

38.5.5.4 Single-pulse interrupt mode

In Single-pulse Interrupt Mode, an interrupt event is always captured in the register IRQ_NOTIFY, independent of the state of IRQ_EN. However, only the first interrupt event of an enabled interrupt within a common interrupt set is forwarded to signal IRQ_line. Additional interrupt events of the same interrupt set cannot generate pulses on the signal IRQ_line, until the corresponding bits in register IRQ_NOTIFY of enabled interrupts are cleared by a clear event. The IRQ_occurred signal line will be high, if the IRQ_EN and the IRQ_NOTIFY register bits are set. The Single-pulse interrupt mode is shown in **Figure 38.18**.

The only exceptions are the modules ARU and DPLL. In these modules the IRQ_occurred bit of each interrupt is directly connected (without OR-conjunction of neighboring IRQ_occurred bits) to the inverter for suppressing further interrupt pulses.

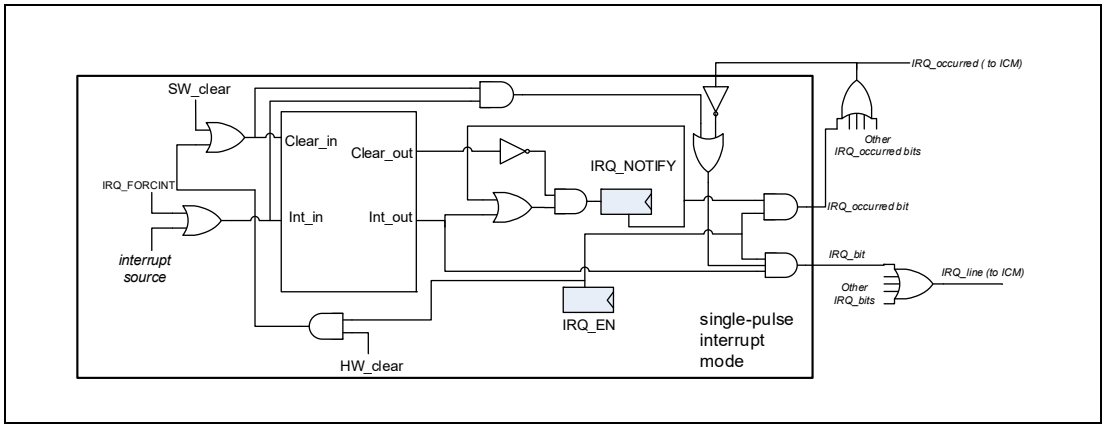


Figure 38.18 Single-pulse interrupt mode scheme

To avoid unexpected IRQ behavior in the single pulse mode, all desired interrupt sources should be enabled by a single write access to IRQ_EN and the notification bits should be cleared by a single write access to the register IRQ_NOTIFY.

The additional error interrupt enable mechanism for single-pulse interrupt is shown below

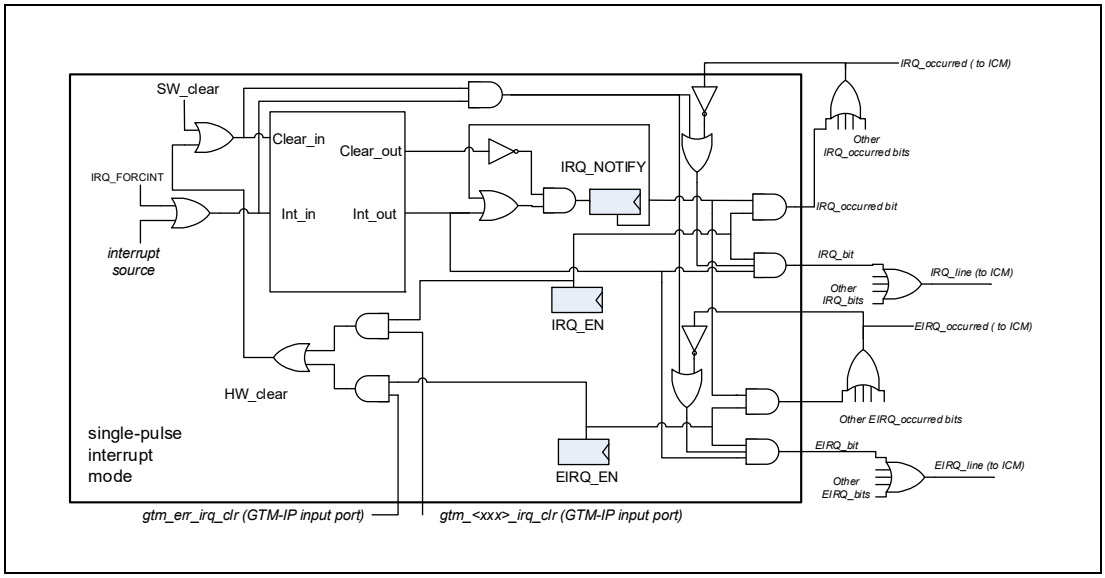


Figure 38.19 Single-pulse interrupt scheme for modules AEI-bridge, BRC, FIFO, TIM, MCS, DPLL, SPE, CMP

In Single-pulse Interrupt Mode, an error interrupt event is always captured in the register IRQ_NOTIFY, independent of the state of EIRQ_EN. However, only the first error interrupt event of an enabled error interrupt within a common error interrupt set is forwarded to signal EIRQ_line. Additional error interrupt events of the same error interrupt set cannot generate pulses on the signal EIRQ_line, until the corresponding bits in register IRQ_NOTIFY of enabled error interrupts are cleared by a clear event. The EIRQ_occurred signal line will be high, if the EIRQ_EN and the IRQ_NOTIFY register bits are set. The Single-pulse interrupt mode for error interrupts is shown in **Figure 38.19**.

To avoid unexpected EIRQ behavior in the single pulse mode, all desired error interrupt sources should be enabled by a single write access to EIRQ_EN and the notification bits should be cleared by a single write access to the register IRQ_NOTIFY.

The only exceptions are the modules ARU and DPLL. In these modules the EIRQ_occurred bit of each error interrupt is directly connected (without OR-conjunction of neighboring EIRQ_occurred bits) to the inverter for suppressing further error interrupt pulses.

38.5.5.5 GTM-IP Interrupt concentration method

Because of the grouping of interrupts inside the ICM, it can be necessary for the software to access the ICM sub-module first to determine the interrupt set that is responsible for an interrupt. A second access to the responsible register IRQ_NOTIFY is then necessary to identify the interrupt source, serve it and to reset the interrupt flag in register IRQ_NOTIFY afterwards. The interrupt flags are never reset by an access to the ICM. For a detailed description of the ICM sub-module, refer to **Section 38.24, Interrupt Concentrator Module (ICM)**.

38.5.6 GTM-IP Software Debugger Support

For software debugger support the GTM-IP comes with several features. E.g. status register bits must not be altered by a read access from a software debugger. To avoid this behavior to reset a status register bit by software, the CPU has to write a '1' explicitly to the register bit to reset its content.

The **Table 38.18** describes the behavior of some GTM-IP registers with special functionality on behalf of read accesses from the AEI bus interface.

Table 38.18 Register behavior in case of Software Debugger accesses

Module	Register	Description
AFD	AFD[i]_CH[z]_BUF_ACC	The FIFO read access pointers are not altered on behalf of a Debugger read access to this register
TIM	TIM[i]_CH[x]_GPR0/1	The overflow bit is not altered in case of a Debugger read access to this register.
TIM	TIM[i]_CH[x]_ECNT	The register is not cleared in case of a Debugger read access to this register.
ATOM	ATOM[i]_CH[x]_SR0/1	In SOMC mode a read access to this register by the Debugger does not release the channel for a new compare/match event.

Further on, some important states inside the GTM-IP sub-module have to be signaled to the outside world, when reached and should for example trigger the software debugger to stop program execution. For this internal state signaling, refer to the GTM-IP module integration guide.

The GTM provides an external signal gtm_halt, which disables clock signal SYS_CLK for debugging purposes. If SYS_CLK is disabled, a connected debugger can read any GTM related register and the GTM internal RAMs using AEI. Moreover, the debugger can also perform write accesses to the internal RAMs and to all GTM related registers in order to enable advanced debugging features (e.g. modifications of register contents in single step mode).

38.5.7 GTM-IP Programming conventions

To serve different application domains the GTM-IP is a highly configurable module with many configuration modes. In principle the sub-modules of the GTM-IP are intended to be configured at system startup to fulfill certain functionality for the application domain the micro controller runs in.

For example, a TIM input channel can be used to monitor an application specific external signal, and this signal has to be filtered. Therefore, the configuration of the TIM channel filter mode will be specific to the external signal characteristic. While it can be necessary to adapt the filter thresholds during runtime an adaptation of the filter mode during runtime is not reasonable. Thus, the change of the filter mode during runtime can lead to an unexpected behavior.

In general, the programmer has to be careful when reprogramming configuration registers of the GTM-IP sub-modules during runtime. It is recommended to disable the channels before reconfiguration takes place to avoid unexpected behavior of the GTM-IP.

38.5.8 GTM-IP TOP-Level Configuration Registers Overview

GTM-IP TOP-level contains following configuration registers:

Table 38.19 Register list

Symbol	Register Name	Details in Section
GTM_REV	GTM-IP Version control register	**38.5.9.1**
GTM_RST	GTM-IP Global reset register	**38.5.9.2**
GTM_CTRL	GTM-IP Global control register	**38.5.9.3**
GTM_AEI_ADDR_XPT	GTM-IP AEI Timeout exception address register	**38.5.9.4**
GTM_AEI_STA_XPT	GTM-IP Interrupt notification register	**38.5.9.5**
GTM_IRQ_NOTIFY	GTM-IP Interrupt notification register	**38.5.9.6**
GTM_IRQ_EN	GTM-IP Interrupt enable register	**38.5.9.7**
GTM_IRQ_FORCINT	GTM-IP Software interrupt generation register	**38.5.9.8**
GTM_IRQ_MODE	GTM-IP top level interrupts mode selection.	**38.5.9.9**
GTM_BRIDGE_MODE	GTM-IP AEI bridge mode register	**38.5.9.10**
GTM_BRIDGE_PTR1	GTM-IP AEI bridge pointer 1 register	**38.5.9.11**
GTM_BRIDGE_PTR2	GTM-IP AEI bridge pointer 2 register	**38.5.9.12**
GTM_MCS_AEM_DIS	GTM-IP MCS master port disable register	**38.5.9.13**
GTM_EIRQ_EN	GTM-IP Error interrupt enable register	**38.5.9.14**
GTM_CLS_CLK_CFG	GTM-IP Cluster Clock Configuration	**38.5.9.15**
GTM_CFG	GTM Configuration register	**38.5.9.16**

38.5.9 GTM-IP TOP-Level Configuration Registers Description

38.5.9.1 GTM_REV

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEV_CODE2				DEV_CODE1				DEV_CODE0				MAJOR			
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINOR				NO				STEP							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.20 GTM_REV Register Contents

Bit Position	Bit Name	Function
31 to 28	DEV_CODE2	Device encoding digit 2.
27 to 24	DEV_CODE1	Device encoding digit 1.
23 to 20	DEV_CODE0	Device encoding digit 0.
19 to 16	MAJOR	Major version number Define major version number of GTM-IP specification.
15 to 12	MINOR	Minor version number Define minor version number of GTM-IP specification.
11 to 8	NO	Delivery number Define delivery number of GTM-IP specification.
7 to 0	STEP	Release step GTM Release step.

NOTE

See device specific **Section 38.28, GTM Device 358** for reset value.

38.5.9.2 GTM_RST

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	BRIDGE_MODE_WRDIS	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.21 GTM_RST Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	BRIDGE_MODE_WRDIS	GTM_BRIDGE_MODE write disable 0: Writing of GTM_BRIDGE_MODE register is enabled 1: Writing of GTM_BRIDGE_MODE register is disabled **NOTE** This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
26 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RST	GTM-IP Reset 0: No reset action 1: Initiate reset action for all sub-modules **NOTES** 1. This bit is automatically cleared by hardware after it was written. Therefore, the register is always read as zero (0) by the software. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**

38.5.9.3 GTM_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00008_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEIM_CLUSTER				—	—	—	TO_VAL					—	—	TO_MODE	RF_PROT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 38.22 GTM_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 12	AEIM_CLUSTER	AEIM_CLUSTER: AEIM cluster number **NOTES** 1. These bits show the number of the AEI master port cluster which throws the interrupts AEIM_USP_ADDR, AEIM_IM_ADDR and AEIM_USP_BE depending on the AEI master port access status. 2. If one of the corresponding irq notify bits (6:4) is set, this bit field will be frozen until the interrupt notify bits (6:4) are cleared.
11 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 4	TO_VAL	AEI timeout value. These bits define the number of cycles after which a timeout event occurs. When TO_VAL equals zero (0) the AEI timeout functionality is disabled.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TO_MODE	AEI timeout mode. 0: Observe: If timeout_counter = 0 the address and rw signal in addition with timeout flag will be stored to the GTM_AEI_ADDR_XPT register. Following timeout_counter = 0 accesses will not overwrite the first entry in the aei_addr_timeout register. Clearing the timeout flag/aei_status error_code will reenable the storing of a next faulty access. 1: Abort: In addition to observe mode the pending access will be aborted by signalling an illegal module access on aei_status and sending ready. In case of a read deliver as data 0 by serving of next AEI accesses.
0	RF_PROT	RST and FORCINT protection. 0: SW RST (global), SW interrupt FORCINT, and SW RAM reset functionality is enabled 1: SW RST (global), SW interrupt FORCINT, and SW RAM reset functionality is disabled

38.5.9.4 GTM_AEI_ADDR_XPT

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0000C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TO_W1R0	TO_ADDR			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TO_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.23 GTM_AEI_ADDR_XPT Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20	TO_W1R0	AEI timeout Read/Write flag. This bit defines the AEI Read/Write flag for which the AEI timeout event occurred.
19 to 0	TO_ADDR	AEI timeout address. This bit field defines the AEI address for which the AEI timeout event occurred.

38.5.9.5 GTM_AEI_STA_XPT

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0002C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	W1R0	ADDR			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.24 GTM_AEI_STA_XPT Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20	W1R0	AEI exception Read/Write flag. This bit field captures the address of the first AEI access resulting with a non-zero AEI status signal. The bit field can be cleared by clearing the interrupt flags AEI_USP_ADDR, AEI_USP_BE, and AEI_IM_ADDR in the register GTM_IRQ_NOTIFY.
19 to 0	ADDR	AEI exception address. This bit field captures the address of the first AEI access resulting with a non-zero AEI status signal. The bit field can be cleared by clearing the interrupt flags AEI_USP_ADDR, AEI_USP_BE, and AEI_IM_ADDR in the register GTM_IRQ_NOTIFY.

38.5.9.6 GTM_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLK_EN_EXP_STATE				CLK_EN_ERR_STATE				—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLK_PER_ERR	CLK_EN_ERR	AEIM_USP_BE	AEIM_USP_BE	AEIM_USP_ADDR	AEI_USP_BE	AEI_IM_ADDR	AEI_USP_ADDR	AEI_TO_XPT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.25 GTM_IRQ_NOTIFY Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	CLK_EN_EXP_STATE	Expected clock enable state. This bit field defines the GTM expected clk enable state at occurrence of the CLK_EN_ERR event. Function only available with INT_CLK_EN_GEN = 0_B: 0000_B: Enable state for internal clock aei_sys_clk 0001_B: Enable state for internal clock aei_sys_clk/2 Other than above: Setting prohibited. **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
27 to 24	CLK_EN_ERR_STATE	Erroneous clock enable state. This bit field defines the GTM external clk enable state at occurrence of the CLK_EN_ERR event. Function only available with INT_CLK_EN_GEN = 0_B: 0000_B: Enable state for internal clock aei_sys_clk 0001_B: Enable state for internal clock aei_sys_clk/2 Other than above: Setting prohibited. **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
23 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CLK_PER_ERR	Clock period error interrupt. 0: No interrupt occurred 1: CLK_PER_ERR interrupt was raised by clock enable watchdog **NOTES** 1. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 2. Read as zero in case of INT_CLK_EN_GEN = 1_B.

Table 38.25 GTM_IRQ_NOTIFY Register Contents (2/3)

Bit Position	Bit Name	Function
7	CLK_EN_ERR	Clock enable error interrupt. 0: No interrupt occurred 1: CLK_EN_ERR interrupt was raised by clock enable watchdog **NOTES** 1. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 2. Read as zero in case of INT_CLK_EN_GEN = 1_B.
6	AEIM_USP_BE	AEI master port unsupported byte enable interrupt. 0: No interrupt occurred 1: AEIM_USP_BE interrupt was raised by the AEI master port interface **NOTE** This bit will be cleared on a CPU write access of value ‘1’. A read access leaves the bit unchanged.
5	AEIM_IM_ADDR	AEI master port illegal Module address interrupt. 0: No interrupt occurred 1: AEIM_IM_ADDR interrupt was raised by the AEI master port interface **NOTE** This bit will be cleared on a CPU write access of value ‘1’. A read access leaves the bit unchanged.
4	AEIM_USP_ADDR	AEI master port unsupported address interrupt. 0: No interrupt occurred 1: AEIM_USP_ADDR interrupt was raised by the AEI master port interface **NOTE** This bit will be cleared on a CPU write access of value ‘1’. A read access leaves the bit unchanged.
3	AEI_USP_BE	AEI unsupported byte enable interrupt. 0: No interrupt occurred 1: AEI_USP_BE interrupt was raised by the AEI interface **NOTE** This bit will be cleared on a CPU write access of value ‘1’. A read access leaves the bit unchanged.
2	AEI_IM_ADDR	AEI illegal Module address interrupt. 0: No interrupt occurred 1: AEI_IM_ADDR interrupt was raised by the AEI interface **NOTE** This bit will be cleared on a CPU write access of value ‘1’. A read access leaves the bit unchanged.

Table 38.25 GTM_IRQ_NOTIFY Register Contents (3/3)

Bit Position	Bit Name	Function
1	AEI_USP_ADDR	AEI unsupported address interrupt. 0: No interrupt occurred 1: AEI_USP_ADDR interrupt was raised by the AEI interface **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	AEI_TO_XPT	AEI timeout exception occurred. 0: No interrupt occurred 1: AEI_TO_XPT interrupt was raised by the AEI Timeout detection unit **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

38.5.9.7 GTM_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLK_PER_ERR_IRQ_EN	CLK_EN_ERR_IRQ_EN	AEIM_USP_BE_IRQ_EN	AEIM_IM_ADDR_IRQ_EN	AEIM_USP_ADDR_IRQ_EN	AEI_USP_BE_IRQ_EN	AEI_IM_ADDR_IRQ_EN	AEI_USP_ADDR_IRQ_EN	AEI_TO_XPT_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.26 GTM_IRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CLK_PER_ERR_IRQ_EN	CLK_PER_ERR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
7	CLK_EN_ERR_IRQ_EN	CLK_EN_ERR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
6	AEIM_USP_BE_IRQ_EN	AEIM_USP_BE_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
5	AEIM_IM_ADDR_IRQ_EN	AEIM_IM_ADDR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
4	AEIM_USP_ADDR_IRQ_EN	AEIM_USP_ADDR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
3	AEI_USP_BE_IRQ_EN	AEI_USP_BE_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
2	AEI_IM_ADDR_IRQ_EN	AEI_IM_ADDR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
1	AEI_USP_ADDR_IRQ_EN	AEI_USP_ADDR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

Table 38.26 GTM_IRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
0	AEI_TO_ XPT_IRQ_EN	AEI_TO_XPT_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

38.5.9.8 GTM_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TRG_CLK_PER_ERR	TRG_CLK_EN_ERR	TRG_AEIM_USP_BE	TRG_AEIM_IM_ADDR	TRG_AEIM_USP_AD	TRG_AEI_USP_BE	TRG_AEI_IM_ADDR	TRG_AEI_USP_ADDR	TRG_AEI_TO_XPT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.27 GTM_IRQ_FORCINT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	TRG_CLK_PER_ERR	Trigger CLK_PER_ERR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert CLK_PER_ERR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL** 3. Read as zero in case of INT_CLK_EN_GEN = 1_B.
7	TRG_CLK_EN_ERR	Trigger CLK_EN_ERR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert CLK_EN_ERR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL** 3. Read as zero in case of INT_CLK_EN_GEN = 1_B.
6	TRG_AEIM_USP_BE	Trigger AEIM_USP_BE_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEIM_USP_BE_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**

Table 38.27 GTM_IRQ_FORCINT Register Contents (2/2)

Bit Position	Bit Name	Function
5	TRG_AEIM_IM_ADDR	Trigger AEIM_IM_ADDR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEIM_IM_ADDR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
4	TRG_AEIM_USP_ADDR	Trigger AEIM_USP_ADDR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEIM_USP_ADDR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
3	TRG_AEI_USP_BE	Trigger AEI_USP_BE_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEI_USP_BE_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
2	TRG_AEI_IM_ADDR	Trigger AEI_IM_ADDR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEI_IM_ADDR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
1	TRG_AEI_USP_ADDR	Trigger AEI_USP_ADDR_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEI_USP_ADDR_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**
0	TRG_AEI_TO_XPT	Trigger AEI_TO_XPT_IRQ interrupt by software. 0: No interrupt triggering 1: Assert AEI_TO_XPT_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of **Section 38.5.9.3, GTM_CTRL**

38.5.9.9 GTM_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0001C_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.28 GTM_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IRQ_MODE	Interrupt strategy mode selection for the AEI timeout and address monitoring interrupts. 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTES** 1. The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**. 2. This mode selection is only valid for the interrupts described in **Section 38.5.9.6**.

38.5.9.10 GTM_BRIDGE_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00030_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BUFF_DPT								—	—	—	—	—	—	—	BRG_RST
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SYNC_INPUT_REG	—	—	BUFF_OVL	MODE_UP_PGR	—	—	—	—	—	BYPASS_SYNC	MSK_WR_RSP	BRG_MODE
Value after reset	0	0	0	—	0	0	—	—	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.29 GTM_BRIDGE_MODE Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	BUFF_DPT	Buffer depth of AEI bridge. Signals the buffer depth of the GTM AEI bridge implementation. **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor.
23 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	BRG_RST	Bridge software reset. 0: No bridge reset request. 1: Bridge reset request. **NOTE** This bit is cleared automatically after write.
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SYNC_INPUT_REG	Additional pipelined stage in synchronous bridge mode 0: No additional pipelined stage implemented. 1: Additional pipelined stage implemented. All accesses in synchronous mode will be increased by one clock cycle. **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	BUFF_OVL	Buffer overflow register. 0: No buffer overflow occurred. 1: Buffer overflow occurred. A buffer overflow can occur while multiple aborts are issued by the external bus or a pipelined instruction is started while FBC = 0 (see **Section 38.5.9.11, GTM_BRIDGE_PTR1**).

Table 38.29 GTM_BRIDGE_MODE Register Contents (2/2)

Bit Position	Bit Name	Function
8	MODE_UP_PGR	Mode update in progress. 0: No update in progress. 1: Update in progress.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	BYPASS_SYNC	Bypass synchronizer flipflops. Function only available with BRG_MODE = 1 0: synchronizer flip-flops in use, latency increase due to synchronization (aei_clk -> aei_sys_clk and back aei_sys_clk -> aei_clk). This setting must be used if aei_clk and aei_sys_clk operate fully asynchronous by independent clock sources. 1: synchronizer flip-flops are bypassed. No additional latency due to synchronization. This setting can be used if aei_clk and aei_sys_clk are generated by clock gating or clock division out of a common clock source. Clock edges on aei_clk and aei_sys_clk generated out of the same clock edge of the common clock source must have zero skew.
1	MSK_WR_RSP	Mask write response. 0: Do not mask the write response. Depending on the selected address the latency for execution can vary due to GTM internal arbitration. After this time the status of the access will be signaled by the signal AEI_STATUS to the bus interface. 1: Mask write response. The write buffer of the bridge is activated, the actual access will be stored to the write buffer and without latency on the bus interface the acceptance of the access is signaled. AEI_STATUS = 00_B will be signaled. In case of a full write buffer the actual access will be postponed until the next write buffer entry becomes free. **NOTES** 1. The status of the executed write accesses can be observed by using the notify bits AEI_USP_ADDR,AEI_IM_ADDR, AEI_USP_BE in the register GTM_IRQ_NOTIFY. 2. With active write buffer MSK_WR_RSP = 1, execution of actions can be delayed due to previous inserted write actions in the transaction buffer which wait to be serviced. This can lead to the fact that an access on the bus to a different peripheral than the GTM might be executed earlier in time than the write access buffered in the GTM. Applications must be setup up with this in mind otherwise unexpected operation can happen.
0	BRG_MODE	Defines the operation mode for the AEI bridge. 0: AEI bridge operates in sync_bridge mode 1: AEI bridge operates in async_bridge mode **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor.

NOTE

All writable bits are protected by bit BRIDGE_MODE_WRDIS at **Section 38.5.9.2, GTM_RST**.

38.5.9.11 GTM_BRIDGE_PTR1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00034_H

Value after reset: $0xx0\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSP_TRAN_RDY						FBC						ABT_TRAN_PGR			
Value after reset	0	0	0	0	0	0	–	–	–	–	–	–	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABT_TRAN_PGR	TRAN_IN_PGR					FIRST_RSP_PTR					NEW_TRAN_PTR				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.30 GTM_BRIDGE_PTR1 Register Contents

Bit Position	Bit Name	Function
31 to 26	RSP_TRAN_RDY	Response transactions ready. Amount of ready response transactions.
25 to 20	FBC	Free buffer count. Number of free buffer entries. **NOTE** Initial value depends on the hardware configuration chosen by silicon vendor. (see BUFF_DPT in **Section 38.5.9.10, GTM_BRIDGE_MODE**).
19 to 15	ABT_TRAN_PGR	Aborted transaction in progress pointer.
14 to 10	TRAN_IN_PGR	Transaction in progress pointer (acquire)
9 to 5	FIRST_RSP_PTR	First response pointer. Signals the actual value of first response pointer.
4 to 0	NEW_TRAN_PTR	New transaction pointer. Signals the actual value of the new transaction pointer.

NOTES

1. This register operates on the AEI_CLK domain.
2. This register holds diagnosis information about the AEI bus bridge. Each access to the GTM_IP will update the defined pointer bit fields. Depending on the mode of GTM_MODE_BRIDGE (BRG_MODE, MSK_WR_RESP), the AEI protocol and operating frequency which is use, the 4 pointer bit fields will change at different clock cycles relative to the start of the transaction. This leads to the fact that reading the register can show values not equal to the defined Initial Value, even directly after a write to GTM_BRIDGE_MODE with BRG_RST=1 was done.

38.5.9.12 GTM_BRIDGE_PTR2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TRAN_IN_PGR2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.31 GTM_BRIDGE_PTR2 Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4 to 0	TRAN_IN_PGR2	Transaction in progress pointer (acquire2)

Note: This register operates on the GTM_CLK domain.

38.5.9.13 GTM_MCS_AEM_DIS

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0003C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DIS_CLS11		DIS_CLS10		DIS_CLS9		DIS_CLS8	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DIS_CLS7		DIS_CLS6		DIS_CLS5		DIS_CLS4		DIS_CLS3		DIS_CLS2		DIS_CLS1		DIS_CLS0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.32 GTM_MCS_AEM_DIS Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, an undefined value is returned. When writing, write the value after reset.
23 to 0	DIS_CLS[k]	Disable MCS AEIM access in cluster k Multicore encoding in use (DIS_CLS[k](1) defines the state of the signal) 00_B = State is 0; MCS AEM access in cluster x enabled (ignore write access) 01_B = Change state to 0 10_B = Change state to 1 11_B = State is 1; MCS AEM access in cluster x disabled (ignore write access) **NOTE** Any read access to a DIS_CLS[k] bit field will always result in a value 00_B or 11B indicating current state. A modification of the state is only performed with the values 01_B and 10_B. Writing the values 00_B and 11_B is always ignored.

38.5.9.14 GTM_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00020_H

Value after reset: $0000\ 0180_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLK_PER_ERR_EIRQ_EN	CLK_EN_ERR_EIRQ_EN	AEIM_USP_BE_EIRQ_EN	AEIM_IM_ADDR_EIRQ_EN	AEIM_USP_ADDR_EIRQ_EN	AEI_USP_BE_EIRQ_EN	AEI_IM_ADDR_EIRQ_EN	AEI_USP_ADDR_EIRQ_EN	AEI_TO_XPT_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.33 GTM_EIRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CLK_PER_ERR_EIRQ_EN	CLK_PER_ERR_EIRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
7	CLK_EN_ERR_EIRQ_EN	CLK_EN_ERR_EIRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP **NOTE** Read as zero in case of INT_CLK_EN_GEN = 1_B.
6	AEIM_USP_BE_EIRQ_EN	AEIM_USP_BE_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP
5	AEIM_IM_ADDR_EIRQ_EN	AEIM_IM_ADDR_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP
4	AEIM_USP_ADDR_EIRQ_EN	AEIM_USP_ADDR_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP
3	AEI_USP_BE_EIRQ_EN	AEI_USP_BE_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP
2	AEI_IM_ADDR_EIRQ_EN	AEI_IM_ADDR_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP
1	AEI_USP_ADDR_EIRQ_EN	AEI_USP_ADDR_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP

Table 38.33 GTM_EIRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
0	AEI_TO_XPT_EIRQ_EN	AEI_TO_XPT_EIRQ error interrupt enable. 0: Disable error interrupt, interrupt is not visible outside GTM-IP 1: Enable error interrupt, interrupt is visible outside GTM-IP

38.5.9.15 GTM_CLS_CLK_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $000B0_H$

Value after reset: $0000\ 00AA_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLS3_CLK_DIV		CLS2_CLK_DIV		CLS1_CLK_DIV		CLS0_CLK_DIV	
Value after reset	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.34 GTM_CLS_CLK_CFG Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CLS[k]_CLK_DIV	Cluster k Clock Divider 00_B: Cluster is disabled. 01_B: Cluster is enabled without clock divider. 10_B: Cluster is enabled with clock divider 2. 11_B: Setting prohibited. **NOTES** 1. These bits are only writable if bit field RF_PROT of register GTM_CTRL is cleared. 2. The configuration value CLS0_CLK_DIV defines the primary input clock period for CMU. 3. If CLS0_CLK_DIV is configured to a value 10_B (i.e. clock divider 2), the maximum CMU clock frequency for all other cluster c = 1..n is also limited to configured CMU clock frequency of cluster 0.

NOTES

1. The availability of configuration bits is indicated by value of bit CFG_CLOCK_RATE in register CCM[c]_HW_CFG.
   - If CFG_CLOCK_RATE=0, only the values 00_B and 01_B are valid for bit fields CLS[c]_CLK_DIV.
   - If CFG_CLOCK_RATE=1, only the values 00_B, 01_B and 10_B are valid for bit fields CLS[c]_CLK_DIV.
   - If CFG_CLOCK_RATE=1, limitation for some cluster are possible to disable high frequency clock usage.

   In this case only values 00_B and 10_B are valid for bit fields CLS[c]_CLK_DIV. Refer also to **Section 38.28, GTM Device 358** of this specification for detailed information.
2. Writing a value to a bit field CLS[c]_CLK_DIV that is not available in the device, an AEI status 10_B is returned. (with c = 0 to NCCM-1)

38.5.9.16 GTM_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00028_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRC_IN_MUX
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.35 GTM_CFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRC_IN_MUX	GTM_TIM[i]_AUX_IN input source selection 0: Use for TIM[i] output of TOM[k] 1: Use for TIM[i] output of TOM[i] (same cluster)

38.6 AXI Master

38.6.1 Functional Characteristics

The AXI transaction generator implements multiple AEI slave interfaces and an AXI compliant master interface. It includes the necessary registers, arbitration and bus state-machines.

- AEI standard protocol compliant
- 32 bit AEI interface
- Zero wait state register access
- Max 10 AEI slave interfaces (configurable via generic)
- Max 8 AXI transaction slots per AEI slave (configurable via generic)
- One common register set per AEI interface
- Full AMBA AXI 3.0 compliant
- 32 bit address
- 32 bit data
- Single accesses only (length is fixed to 1)
- All other AXI signals are programmable
- ID width configurable
- One "interrupt" line per slot

38.6.2 AXI Transaction Generator Block

38.6.2.1 Functional View

Figure 38.20 shows the basic modules of the AXI Transaction Generator block.

For each AEI interface one cluster module is implemented. This cluster module includes the register module. The transactions are then arbitrated and finally used to create the bus transactions.

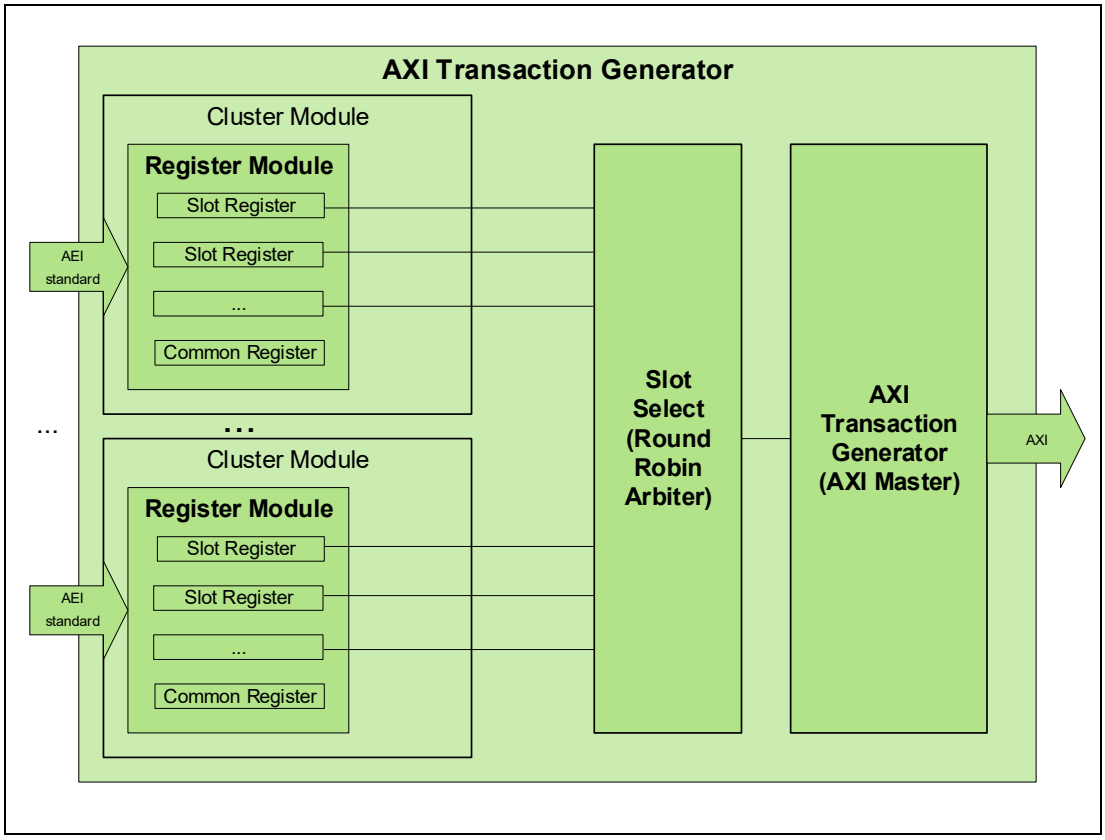


Figure 38.20 Block Diagram

38.6.3 Block Entity Table

The IP has a configurable number of AEI standard slave interfaces on the “left” side and an AXI master interface on the “right” side. **Table 38.36** describes the blocks entity pins.

(1) Pin list

Table 38.36 Pin list (1/3)

Port name	Width	Direction	Description
General Signals			
ACLK	1	Input	AXI port clock
ARESETN	1	Input	AXI port reset (always low active). Release of ARESTN must be synchronized to ACLK.
CLK	generic	input	AXIM port clocks. One clock per AEIM interface. ACLK/CLK must all fulfill synchronous timing relationships, but CLK may be gated.

Table 38.36 Pin list (2/3)

Port name	Width	Direction	Description
ENCLK	generic	Input	Gate enabled for AXIM clocks. Used to synchronize data between CLK/ACLK domains.
CLKOFF	generic	Input	AXIM port is in clock off state. This signals works as a synchronous reset for AEIM ports and overrules ENCLK to avoid hang-ups in ACLK/CLK synchronization.
RESET	1	input	Common AXIM ports reset signal. Active level is defined by generic. Release of RESET must be synchronized to CLK
GTM_HALT	1	Input	Debug signal. While GTM_HALT is asserted, no new requests are accepted by the AXI port. Pending AXI cycles are still completed.
AEI Signals (one set per AEI interface)			
AEIM_AXIM_DEBUG_ACCESS	1	Input	AEI Debug Access
AEIM_AXIM_SEL	1	input	AEI select
AEIM_AXIM_W1R0	1	input	AEI write/read selection
AEIM_AXIM_ADDR	8	input	AEI address MSBs
AEIM_AXIM_ADDR_LSB	2	Input	AEI address LSBs
AEIM_AXIM_WDATA	32	input	AEI write data
AEIM_AXIM_READY	1	output	AEI ready
AEIM_AXIM_RDATA	32	output	AEI read data
AEIM_AXIM_STATUS	2	output	AEI transaction status
AEIM_AXIM_IRQ	generic	output	AEI interrupt request
AEIM_AXIM_IRQ_CLR	generic	input	AEI interrupt clear
Read Address Channel			
ARID	generic	output	AXI transaction ID, ID width can be selected via generic
ARADDR	32	output	AXI read address
ARLEN	4	output	AXI read burst length (fixed to 0x0, as we only support single transactions)
ARBURST	2	output	AXI read burst type (fixed to 0x1 (INCR))
ARSIZE	3	output	AXI read size
ARLOCK	2	output	AXI lock type
ARCACHE	4	output	AXI cache type
ARPROT	3	output	AXI protection type
ARVALID	1	output	AXI read valid
ARREADY	1	input	AXI read ready
Write Address Channel			
AWID	generic	output	AXI transaction ID, ID width can be selected via generic
AWADDR	32	output	AXI write address

Table 38.36 Pin list (3/3)

Port name	Width	Direction	Description
AWLEN	4	output	AXI write burst length (fixed to 0x0, as we only support single transactions)
AWBURST	2	output	AXI write burst type (fixed to 0x1 (INCR))
AWSIZE	3	output	AXI write data size
AWLOCK	2	output	AXI lock type
AWCACHE	4	output	AXI cache type
AWPROT	3	output	AXI protection type
AWVALID	1	output	AXI write data valid
AWREADY	1	input	AXI write data ready
Read Data Channel			
RID	generic	input	AXI transaction ID, ID width can be selected via generic
RDATA	32	input	AXI read data
RLAST	1	input	AXI real last in a burst signal
RRESP	2	input	AXI read response
RVALID	1	input	AXI read data valid
RREADY	1	output	AXI read data ready
Write Data Channel			
WID	generic	output	AXI transaction ID, ID width can be selected via generic
WDATA	32	output	AXI write data
WLAST	1	output	AXI write last in a burst signal
WSTRB	4	output	AXI byte strobes
WVALID	1	output	AXI write data valid
WREADY	1	input	AXI write data ready
Write Response Channel			
BID	generic	input	AXI transaction ID, ID width can be selected via generic
BRESP	2	input	AXI write response
BVALID	1	input	AXI write response valid
BREADY	1	output	AXI wire response ready

38.6.3.1 Block Configuration

The block can be configured for different use-cases with generics. The support generics are described in the following **Table 38.37**.

(1) Generic List

Table 38.37 Generic list

Name	Type	Description
AXIM_ID_WIDTH_C	Integer	Width of the ID busses on the AXI slave interface. Range: 0 to 16, Default: 4
AXIM_DATA_WIDTH_C	Integer	Width of the DATA busses on the AXI interface; Allowed values are 32 and 64
AXI_ADDR_WIDTH_C	Integer	Width of the ADDR busses on the AXI interface; Allowed value is 32.
AXIM_INST_C	Integer	Number of supported AEI masters. Range: 1 to 16
AXIM_USED_C	Integer	Actual number of AEI masters: Range 1 to AXIM_INST_C
AXIM_SLOTS_C	Integer Array	Number of Slots per AEI master. One entry per AEI master. Range: 0 to 24
AXIM_POSTED_WRITE_C	Boolean	Enable posted write support
AXIM_SEC_ACC_C	Boolean	Allow AXI secure bus cycles (AxPROT[1]=0); Should be set to FALSE for most applications
AXIM_PRIV_ACC_C	Boolean	Allow AXI privileged bus cycles (AxPROT[0]=1); Should be set to FALSE or TRUE depending on application

Note: These generic are not implemented with VHDL generics statements. Instead the generics from below table represent constants declared inside the "gtm_scale_pack.vhd" VHDL package.

38.6.4 Detailed Architecture

38.6.4.1 Cluster Module

The cluster module holds all components which belong to one cluster of the GTM. It is intended to ease the layout separation of the different components. Each cluster implements a separate clock input, but clocks of all clusters and AXI port clock must fulfill synchronous timing relationships. Cluster clocks must be generated from the same base clock as AXI port clock (ACLK), but some clock cycles may be gated. Therefore the clock rate for each cluster can be different: CLK(cluster) = ACLK / n; n = 1...X (typical X=2) Signal synchronization between CLK and ACLK domains is supported by the ENCLK inputs. ENCLK(cluster)=”1” will identify active cluster clock edges to ACLK domain logic.

Cluster clocks may be disabled completely for longer time frames (e.g.: debugging). A disabled cluster clock is signaled by CLKOFF(cluster) input signal. This signal will overrule ENCLK(cluster) for CLK/ ACLK synchronization logic. Otherwise synchronization logic may hang if CLK is disabled. After a clock off phase, cluster state is reset. Therefore CLKOFFF must be active for at least one active clock cycle when clock off state is released. When one cluster leaves clock off state, system software must ensure that no active AXI cycle is pending for any slot of this cluster. Otherwise AXI responses from requests issued before entering clock off state may collide with newer slot requests, issued after release of clock off state.

38.6.4.2 Register Module

For each AEI slave interface one register module is instantiated. The register module is built from two building blocks, the slot registers, where one set for each available slot is implemented and a common register area where the slot allocation takes place. All register provide 1 cycle accesses; this means AEIM_AXIM_READY must be asserted in the same cycle as AEIM_AXIM_SEL is asserted. This is required for correct MCS operation. In debug mode (AEIM_AXIM_DEBUG_ACCESS=1) read accesses to registers must not have any side effects:

A debug read to the data register will not start a new AXI read cycle in auto-increment mode

A debug read to the REQUEST register will not allocate a new slot.

38.6.4.3 Slot Registers

In the slot registers all information needed for the AXI transaction is stored. The goal is to use as less registers as possible to reduce the number of required accesses by the controlling program. The Slot register block also includes a state machine to communicate with the following arbiter stage and the AXI transaction generator.

(1) Slot Registers

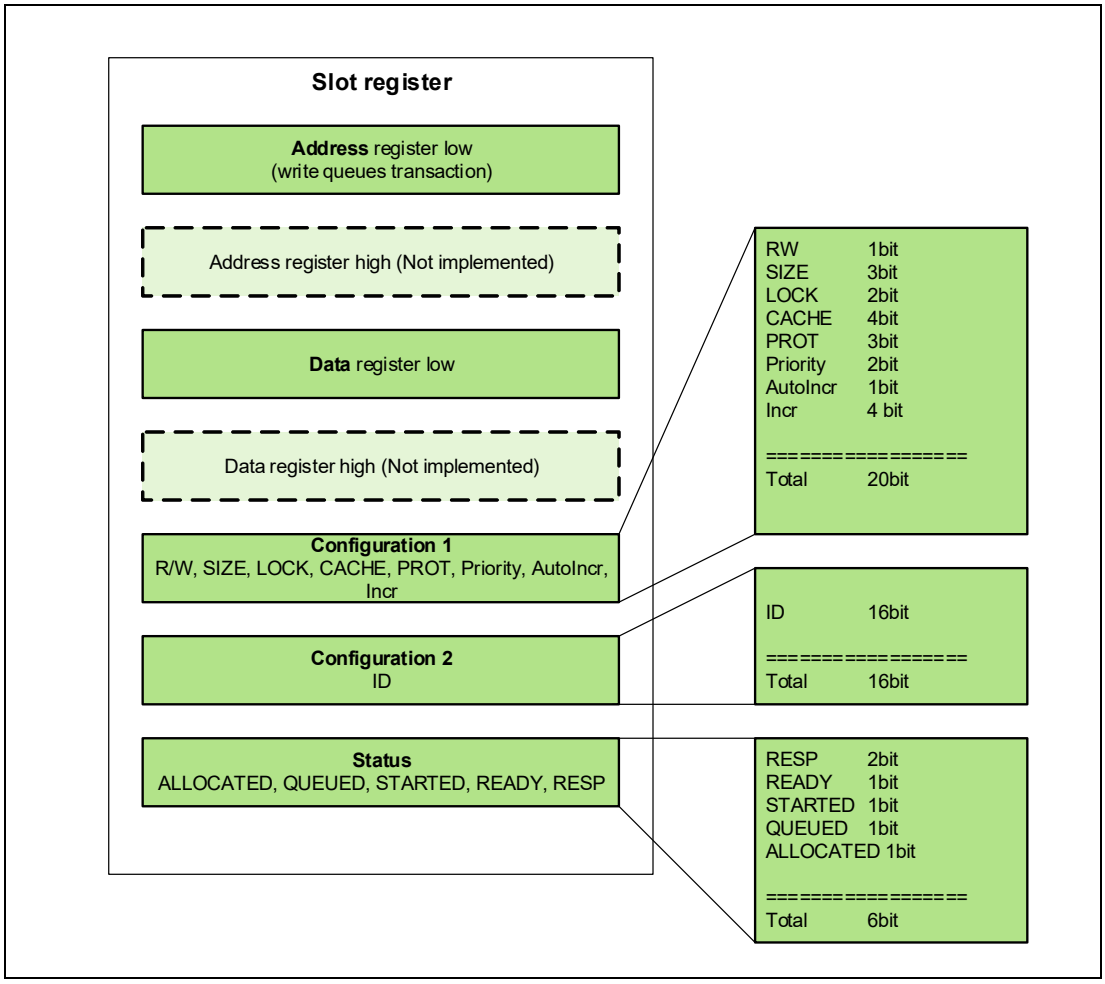


Figure 38.21 Slot Register

(2) Slot State Machine

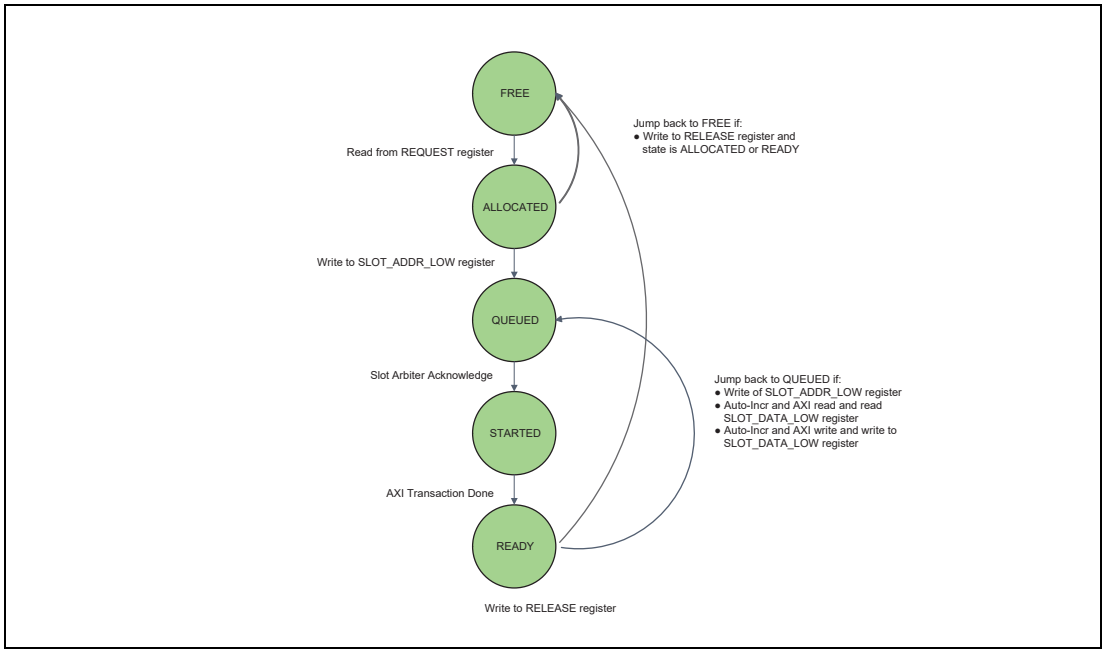


Figure 38.22 Slot State Machine

38.6.4.4 Interrupt Handling

Each slot provides one interrupt line to the corresponding MCS (AEIM_AXIM_IRQ). This interrupt line is set when the slot state changes from STARTED to READY. The interrupt is cleared by asserting the external input line “AEIM_AXIM_IRQ_CLR” for this slot.

38.6.4.5 Auto Increment

The module offers an address auto increment feature. If it is enabled, the address (inside the ADDR register) is incremented automatically after each transaction according to the INCR bit field in the CFG1 Register: New_ADDR <= ADDR + INCR To disable auto-increment, the auto-increment bit in the configuration register has to be cleared. There is one auto-increment adder per cluster. Each slot will request auto-increment processing from this shared resource whenever the last transaction has finished (Slot state changed from STARTED to READY). The new address is available inside the address register two clock cycles later.

38.6.4.6 Common Registers

The common registers are used for slot management. If an MCS thread wants to allocate a transaction slot it reads the REQUEST register. If a slot is available the return value is the number of the allocated slot in two codings. The lower 24bit are one-hot coded, each bit representing one slot.

The upper 8 bit are the allocated slot number in binary format. If no slot was available the maximum value of 0xFF is returned, the lower bytes are then 0x00 0x00 0x00. To release a slot, MCS must write a "1" to the corresponding bit in the RELEASE register. Via the FREE register it is possible to check which slots are still free. One bit for each implemented slot exists.

(1) Common Registers

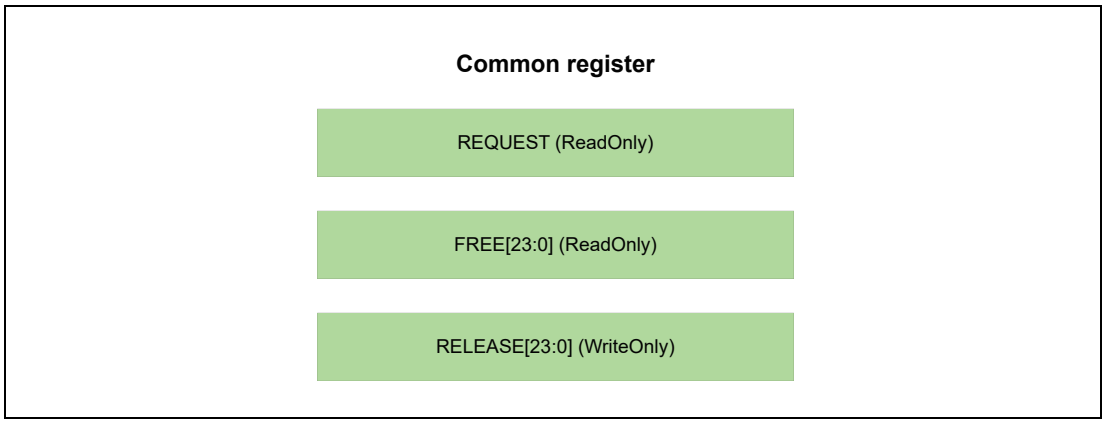


Figure 38.23 Common Registers

38.6.4.7 Arbitration Scheme

The slot arbiter should balance all slots in an equal manner. To achieve an equal load balancing over all slots, the arbiter must process all slots in parallel (flat). Also each slot request can be assigned to a priority level between 0 and 3 and the arbiter must select slots depending on this priority level. The slot arbiter will only accept new slot requests while input signal GTM_HALT="0". Slot requests already accepted when GTM_HALT changes from "1" to "0" are still forwarded to the AXI transaction generator. Therefore after asserting GTM_HALT up to two pending AXI transactions may still be issued to the AXI port.

(1) Arbiter implementation

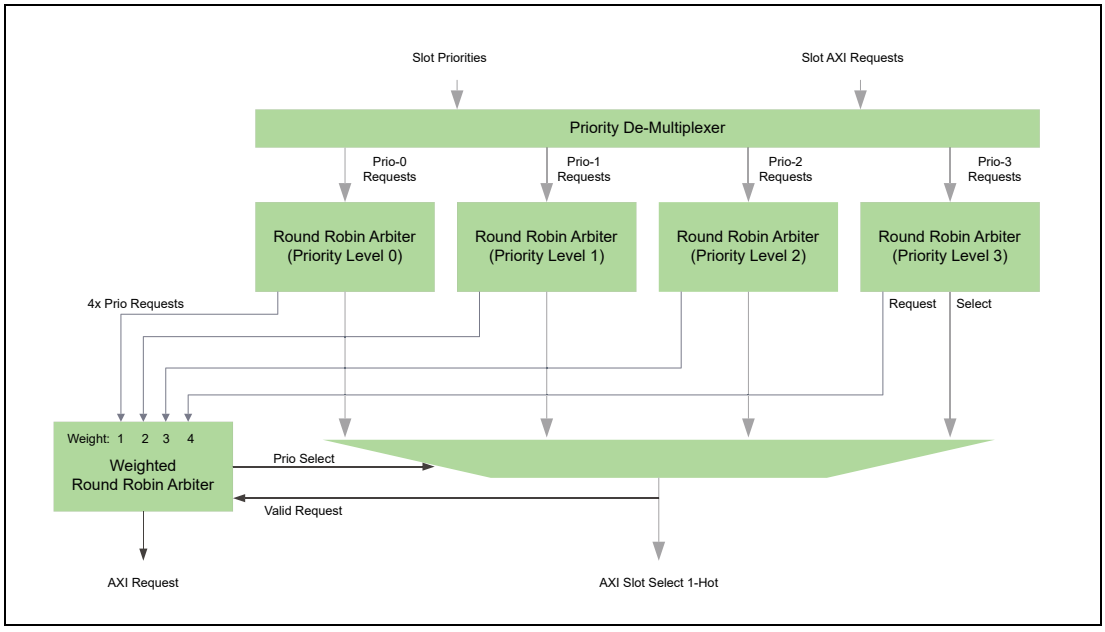


Figure 38.24 Arbiter Implementation

38.6.4.8 Round Robin Arbiter

The Round Robin Arbiter for each priority class must be configurable for the actual number of slots. For maximum GTM configurations this will be up to 80 inputs (10x8).

(1) Round Robin Arbiter

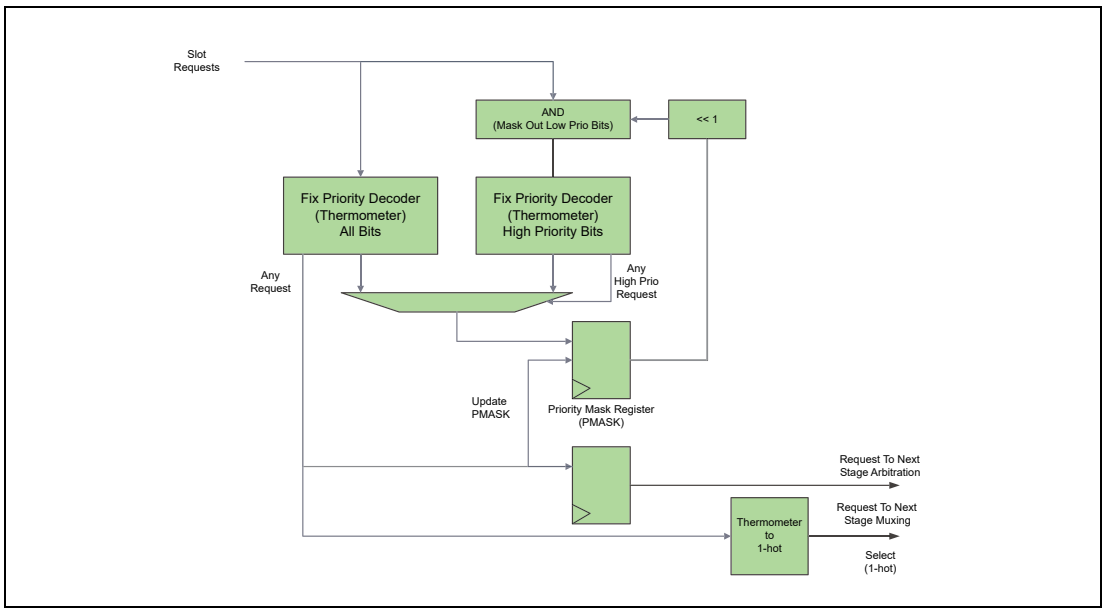


Figure 38.25 Round Robin Arbiter

38.6.4.9 Transaction Generator

The Transaction Generator creates the actual AXI bus transaction. It applies the transaction values to the right AXI lines and observes the bus until the transaction has been completed. If available the next transaction is immediately started. The Transaction Generator has output as well as input Flip-Flops to the AXI bus to ease the timing towards the NOC. Due to the pipelining of AXI handshake signals, the Transaction Generator can only initiate new transaction every second clock cycle. This is no limitation, because also the arbiter can deliver new requests with a maximum rate of every second clock cycle. The Transaction Generator will check the parameters for every requested AXI transaction for validity. If an invalid request is detected, the AXI Transaction Generator will not send a request to the AXI bus, instead it will directly respond to the requesting slot with an error response code (11_B).

Invalid transactions are:

1. Secure AXI cycle requested (PROT[1] = 0) and VHDL constant "axim_sec_acc_c" is set to false (default). In systems supporting ARM Trustzone technology, secure AXI bus cycles are reserved for the so call "Secure Monitor" software. Other bus master like GTM should not issue secure bus cycles.
2. Privileged AXI cycle requested (PROT[0] = 1) and VHDL constant "axim_priv_acc_c" is set to false. If privileged bus cycles are allowed depends on application. By default "axim_priv_acc_c" is set to true.
3. LOCK = 11_B (reserved value) or LOCK = 10_B (Hard Lock): An AXI cycle with LOCK = 10_B is a hard lock. It will lock all other bus cycles on the NOC interconnect until the bus lock is removed with an unlock bus cycle. This is dangerous. To avoid this, GTM must only use "soft locks" or "exclusive access locks" (LOCK = 01_B).

4. AXI cycles with SIZE greater 2 (greater 4 byte): AXI Transaction Generator will only support transactions not larger that the external data bus width, which is currently fixed to 32 bit.
5. Misaligned 2-byte bus cycles: If AxSIZE is configured for 2 bytes (001_B), the address must be aligned to 2 bytes (AxADDR[0] = 0).
6. Misaligned 4-byte bus cycles: If AxSIZE is configured for 4 bytes (010_B), the address must be aligned to 4 bytes (AxADDR[1:0] = 00_B).

38.6.4.10 Data Alignment

AXI Transaction Generator will shift the data from the slot DATA register to the correct AXI data bus lines, according to ADDR. Software must expect the data inside the data register always aligned to bit 0. For AXI operations with a size of less than 32 bits, the Transaction Generator will shift the data bytes to the according positions on the AXI data bus. For AXI read operations, the Transaction Generator will mask unused bytes. E.g.: If software wants to initiate a one byte AXI write transaction to address 3, it will program the data byte to bit 7.0 of the DATA register. AXI Transaction Generator will shift this byte to bits 31.24 of the AXI data bus. For AXI write operations, the Transaction Generator will calculate the correct AXI write strobe signals (WSTRB) depending on SIZE and ADDR.

38.6.5 AXIM Configuration Registers Description

Table 38.38 Axim Configuration Registers Description

Register name	Description	Details in Section
AXIM[i]_FREE	Slot allocation status	**38.6.6.1**
AXIM[i]_REQUEST	Slot request (allocation)	**38.6.6.2**
AXIM[i]_RELEASE	Slot release (de-allocation)	**38.6.6.3**
AXIM[i]_SLOT[x]_ADDR_LOW	Address bits 31:0 of AXI transaction	**38.6.6.4**
AXIM[i]_SLOT[x]_DATA_LOW	Data bits 31:0 of AXI transaction	**38.6.6.5**
AXIM[i]_SLOT[x]_CFG1	Slot configuration 1	**38.6.6.6**
AXIM[i]_SLOT[x]_CFG2	Slot configuration 2	**38.6.6.7**
AXIM[i]_SLOT[x]_STATUS	Slot status	**38.6.6.8**

38.6.6 AXIM Configuration Registers description

38.6.6.1 AXIM[i]_FREE

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 14000_H + (400_H × i)

Value after reset: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FREE[k]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.39 AXIM[i]_FREE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	FREE[k]	Each bit represents the allocation status of one slot 0: Slot is allocated, or not available 1: Slot is free

38.6.6.2 AXIM[i]_REQUEST

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 14004_H + (400_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	REQID								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REQ1HOT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.40 AXIM[i]_REQUEST Register Contents

Bit Position	Bit Name	Function
31 to 24	REQID	This bit field shows the new allocated slot as binary encoded index. If no slot could be allocated this bit field is read as all 1 (FF_H).
23 to 1	Reserved	When read, the value after reset is returned.
0	REQ1HOT0	A read to the REQUEST register will allocate a new slot if any slot is available. This bit field shows the new allocated slot as 1−hot encoded vector. If no slot could be allocated this bit field is read as all 0.

38.6.6.3 AXIM[i]_RELEASE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> +14008_H + (400_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RELREQ0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.41 AXIM[i]_RELEASE Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RELREQ0	A write to RELREQ will de-allocateone or more slots. Each bit in RELREQ represents one slot. 0: Don't change state of Slot(n) 1: De−allocate Slot(n)

Note: This bit field is self.clearing and always read as zero.

Note: Software should check if a release request will really result in slot de.allocation. Slot will only change state at release request if the slot has no pending request to the AXI Transaction Generator (Slot states ALLOCATED or READY).

38.6.6.4 AXIM[i]_SLOT[x]_ADDR_LOW

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> +14020_H + (400_H × i) + (20_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AXI_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AXI_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.42 AXIM[i]_SLOT[x]_ADDR_LOW Register Contents

Bit Position	Bit Name	Function
31 to 0	AXI_ADDR	Address for the AXI transaction. If the auto−increment mode is enabled (AUTO_INCR = 1), this register is updated after each AXI transaction: AXI_ADDR ≤ AXI_ADDR + INCR If the slot is in ALLOCATED or READY state. A write to this register will start a new AXI request (Slot state changes to QUEUED).

38.6.6.5 AXIM[i]_SLOT[x]_DATA_LOW

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> +14028_H + (400_H × i) + (20_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AXI_DATA_LOW															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AXI_DATA_LOW															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.43 AXIM[i]_SLOT[x]_DATA_LOW Register Contents

Bit Position	Bit Name	Function
31 to 0	AXI_DATA_LOW	AXI write operation (AXI_RW = 0): Data that is send out via AXI transaction. Software must configure this register before the AXI write operation is started. Data in AXI_DATA_LOW is aligned to bit 0. AXI Transaction Generator will apply the data to the correct AXI data lines depending on AXI_ADDR. If auto-increment is enabled (AUTO_INCR = 1) and the slot is in READY state. A write to this register will start a new AXI transaction (Slot state will change to QUEUED). AXI read operation (AXI_RW = 1): Data read by last AXI transaction. Software can read the received data after the AXI transaction has finished (Slot state = READY). Data in AXI_DATA_LOW is always aligned to bit 0. AXI Transaction Generator will shift the AXI data bus lines according to AXI_ADDR. Unused bytes (identified by AXI_SIZE) will be masked and are read as all 0. If auto-increment is enabled (AUTO_INCR = 1) and the slot is in READY state. A read to this register will start a new AXI transaction (Slot state will change to QUEUED)

38.6.6.6 AXIM[i]_SLOT[x]_CFG1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> +14030_H + (400_H × i) + (20_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	AXI_RW	AXI_SIZE			—	—	AXI_LOCK		AXI_CACHE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AXI_CACHE		AXI_PROT			—	—	—	—	PRIO		AUTO_INCR	INCR			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.44 AXIM[i]_SLOT[x]_CFG1 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	AXI_RW	0: AXI write transaction 1: AXI read transaction
24 to 22	AXI_SIZE	000_B: Transmit 1 data byte 001_B: Transmit 2 data bytes 010_B: Transmit 4 data bytes 011_B: Not allowed for GTM 100_B: Not allowed for GTM 101_B: Not allowed for GTM 110_B: Not allowed for GTM 111_B: Not allowed for GTM
21, 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19, 18	AXI_LOCK	00_B: No lock 01_B: Exclusive access ("soft lock") 10_B: Access with bus lock (not allowed for GTM) 11_B: Reserved (not allowed for GTM)
17 to 14	AXI_CACHE	AxCACHE bit field (see ARM AXI−3 spec)
13 to 11	AXI_PROT	AXI_PROT[0]: Privileged (1) or user (0) mode access AXI_PROT[1]: Secure (0) or none−secure (1) mode access AXI_PROT[2]: Data (0) or instruction (1) mode access
10 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 5	PRIO	Slot priority. Priority used for slot arbitration. Lowest priority is 0, highest priority is 3.
4	AUTO_INCR	Enable or disable auto−increment mode 0: Disable 1: Enable
3 to 0	INCR	Address increment for auto−increment mode

38.6.6.7 AXIM[i]_SLOT[x]_CFG2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> +14034_H + (400_H × i) + (20_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AXI_ID															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.45 AXIM[i]_SLOT[x]_CFG2 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	AXI_ID	AXI ID for transaction. If posted writes are enabled (generic), the lower bit of the AXI_ID selects between posted and none−posted writes.

38.6.6.8 AXIM[i]_SLOT[x]_STATUS

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 14038_H + (400_H × i) + (20_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	RESP		READY	STARTED	QUEUED	ALLOC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.46 AXIM[i]_SLOT[x]_STATUS Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5, 4	RESP	AXI response from last AXI transaction 00_B: OK 01_B: EXOKAY 10_B: SLVERR 11_B: DECERR; If AXI Transaction Generator detects any invalid requests or responses (unexpected response ID), it will force response to this value.
3	READY	This bit represents the slot READY state. 0: AXI Transaction Generator has no finished request 1: AXI Transaction Generator has finished request
2	STARTED	This bit represents the slot STARTED state. 0: AXI request is not accepted by arbiter 1: AXI request is accepted by arbiter
1	QUEUED	This bit represents the slot QUEUED state. 0: AXI request is not queued for arbitration 1: AXI request is queued for arbitration
0	ALLOC	0: Slot is free 1: Slot is allocated **NOTE** This bit is asserted as long as the slot is allocated, also during states QUEUED, STARTED and READY. The slot state ALLOCATED is represented by ALLOC = 1 and QUEUED = STARTED = READY = 0

38.7 Advanced Routing Unit (ARU)

38.7.1 Overview

The Advanced Routing Unit (ARU) is a flexible infrastructure component for transferring 53-bit wide data (five control bits and two 24-bit values) between several sub-modules of the GTM core in a configurable manner.

Since the concept of the ARU has already been described in **Section 38.5.3, ARU Routing Concept**, this section only describes additional ARU features that can be used by the software for configuring and debugging ARU related data streams. Also the definition of 'streams' and 'channels' in the ARU context is done in **Section 38.5.3, ARU Routing Concept**.

The principle of ARU data routing is described in **Section 38.5.3, ARU Routing Concept**. In the real GTM implementation the ARU serves in parallel per clock period two individual data destinations, one destination at port ARU-0 and at port ARU-1. Both ARU ports ARU-0 and ARU-1 are running by default in parallel but can be configured in dynamic routing mode (see below) to run in an individual mode.

As already defined in the **Section 38.5.3, ARU Routing Concept**, the ARU read ID is the address of the data source that is configured in the data destination module. These ARU read ID's are selected by the individual counter of ARU ports ARU-0 and ARU-1. Via the ARU ports ARU-0 and ARU-1 with each ARU read ID two independent GTM sub-modules are addressed and served. The combination of ARU port (ARU-0 or ARU-1) and the ARU read ID addresses one ARU wdata source (i.e. the ARU write port of a GTM sub-module).

The assignment of ARU write ports of GTM sub-modules to the ARU ports ARU-0 and ARU-1 and the ARU read ID's is device depending and can be found in **Section 38.28, GTM Device 358**.

38.7.2 Special Data Sources

Besides the addresses of the sub-module related data sources as described in **Section 38.28.10.1, ARU Write Address Overview**, the ARU provides two special data sources that can be used for the configuration of data streams. These data sources are defined as follows:

Address $1FF_H$: Data source that provides always a 53 bit data word with zeros. A read access to this memory location will never block a requesting data destination.

Address $1FE_H$: Data source that never provides a data word. A read access to this memory location will always block a requesting data destination. This is the reset value of the read registers inside the data destinations.

Address 000_H: This address is reserved and can be used to bring data through the ARU registers ARU_DATA_H and ARU_DATA_L into the system by writing the write address 000_H into the ARU_ACCESS register. This means that software test data can be brought into the GTM-IP by the CPU.

38.7.3 ARU Access via AEI

Besides the data transfer between the connected sub-modules, there are two possibilities to access ARU data via the AEI.

38.7.3.1 Default ARU Access

The default ARU access incorporates the registers ARU_ACCESS, which is used for initiation of a read or write request and the registers ARU_DATA_H and ARU_DATA_L that provide the ARU data word to be transferred.

The status of a read or write transfer can be determined by polling specific bits in register ARU_ACCESS. Furthermore the acc_ack bit in the interrupt notify register is set after the read or write access is performed to avoid data loss e.g. on access cancelation.

A pending read or write request may also be canceled by clearing the associated bit.

In the case of a read request, the AEI access behaves as a read request initiated by a data destination of a module. The read request is served by the ARU immediately when no other destination has a pending read request. This means, that an AEI read access does not take part in the scheduling of the destination channels and that the time between two consecutive read accesses is not limited by the round trip time.

On the other hand, the AEI access has the lowest priority behind the ARU scheduler that serves the destination channels. Thus, in worst case, the read request is served after one round trip of the ARU, when all destination channels would request data at the same point in time.

In the case of the write request, the ARU provides the write data at the address defined by the ADDR bit field inside the ARU_ACCESS register.

To avoid data loss, the reserved ARU address 0x0 has to be used to bring data into the system. Otherwise, in case the address specified inside the ADDR bit field is defined for another sub-module that acts as a source at the ARU data loss may occur and no deterministic behavior is guaranteed.

This is because the regular source sub-module is not aware that its address is used by the ARU itself to provide data to a destination.

It is guaranteed that the ARU write data is send to the destination in case of both modules want to provide data at the same time.

Configuring both read and write request bits results in a read request, if the write request bit inside the register isn't already set. The read request bit will be set but not the write request bit. The following table describes the important cases of the bit 12 (RREQ) and bit 13 (WREQ) of the ARU_ACCESS register:

Table 38.47 ARU Access

AEI write access: aei_wdata (13:12)	Actual value of GTM0ARUACCESS (13:12)	Next value of GTM0ARUACCESS (13:12)	Comment
0 0	0 1	0 0	CanCel read request
0 0	1 0	0 0	Cancel write request
0 1	1 0	1 0	Unchanged register
1 0	0 1	0 1	Unchanged register
1 1	0 0	0 1	Both read and write request results in a read request
1 1	1 0	1 0	As before but WREQ bit is already set → unchanged register

38.7.3.2 Debug Access

The debug access mode enables to inspect routed data of configured data streams during runtime.

The ARU provides two independent debug channels, whereas each is configured by a dedicated ARU read address in register ARU_DBG_ACCESS0 and ARU_DBG_ACCESS1 respectively.

The registers ARU_DBG_DATA0_H and ARU_DBG_DATA0_L (ARU_DBG_DATA1_H and ARU_DBG_DATA1_L) provide read access to the latest data word that the corresponding data source sent through the ARU.

Any time when data is transferred through the ARU from a data source to the destination requesting the data the interrupt signal ARU_NEW_DATA0 _IRQ (ARU_NEW_DATA1 _IRQ) is raised.

For advanced debugging purposes, the interrupt signal can also be triggered by software using the register ARU_IRQ_FORCINT.

NOTE

The debug mechanism should not be used by the application, when a HW-Debugger is used to trace the ARU communication. In that case, the debug registers are used by the HW-Debugger to specify the ARU streams that should be traced.

38.7.4 ARU dynamic routing

A dynamic routing feature of the ARU is implemented and can be configured using the additional AEI registers: ARU_CTRL, ARU_[z]_DYN_CTRL, ARU_[z]_DYN_RDADDR, ARU_[z]_DYN_ROUTE_LOW, ARU_[z]_DYN_ROUTE_HIGH, ARU_[z]_DYN_ROUTE_SR_LOW and ARU_[z]_DYN_ROUTE_SR_HIGH. For further information see the register part of this chapter.

38.7.4.1 Dynamic routing - CPU controlled

The dynamic routing feature can be enabled separately for ARU-0 and ARU-1 by setting the corresponding bit fields of the register ARU_CTRL.

The enabling of the dynamic routing feature is synchronized to the normal routing scheme if ARU master ID-0 is addressed. The dynamic route will started with additional ARU master DYN_READ_ID0.

With the dynamic routing feature it is possible to insert additional ARU master ID's, DYN_READ_IDy (y:0.5), in a defined manner into the normal ARU routing scheme.

While inserting additional ARU master ID's the normal ARU routing scheme is paused. Therefore consider that inserting additional ARU master ID's will lengthen the normal routing scheme.

It is possible to configure 6 additional ARU master ID's in the ARU_[z]_DYN_ROUTE_LOW/_HIGH registers for both ARU-0 and ARU-1.

In the bit field DYN_CLK_WAIT of ARU_[z]_DYN_ROUTE_HIGH register the number of clock cycles has to be configured, after which one of the additional ARU master ID's will be inserted.

After each configured number of clock cycles the defined ARU master ID's will be inserted cyclic one after each other in the following manner: ... → DYN_READ_ID0 → DYN_READ_ID1→ DYN_READ_ID2 → DYN_READ_ID3 → DYN_READ_ID4 → DYN_READ_ID5 → DYN_READ_ID0 → ...

In the shadow registers ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH further 6 ARU master, DYN_READ_IDy (y:6.11), can be configured.

The bit DYN_UPDATE_EN in the ARU_[z]_DYN_ROUTE_SR_HIGH register controls whether the ARU_[z]_DYN_ROUTE_LOW/_HIGH registers are updated from its shadow registers except DYN_UPDATE_EN, it is not updated. The update is executed once after writing ARU_[z]_DYN_ROUTE_SR_HIGH. If update started DYN_UPDATE_EN is reset.

With the DYN_ROUTE_SWAP option in the ARU_[z]_DYN_CTRL register it is possible to swap the registers ARU_[z]_DYN_ROUTE_LOW/HIGH with its shadow registers ARU_[z]_DYN_ROUTE_SR_LOW / HIGH. The swapping is executed always after the 6 ARU master DYN_READ_ID's are inserted. So it is possible to insert a maximum of 12 ARU master DYN_READ_ID's cyclic after each configured number of clock cycles. If swap started DYN_UPDATE_EN is reset.

Setting the bit field DYN_CLK_WAIT of ARU_[z]_DYN_ROUTE_HIGH register to zero, only the defined ARU master DYN_READ_ID's will be executed. The normal ARU routing scheme is stopped.

Setting the bit field DYN_CLK_WAIT of ARU_[z]_DYN_ROUTE_HIGH register to 15, only the normal ARU routing scheme is executed. Inserting of additional ID's is stopped.

To reset the ARU caddr counter and ARU dynamic route counter set bit ARU_ADDR_RSTGLB of CMU_GLB_CTRL following by a write access to register CMU_CLK_EN.

(1) Dynamic routing ring mode

In dynamic routing ring mode it is possible to use all 24 DYN_READ_ID's from both ARU-0 and ARU-1 by setting bit field ARU_DYN_RING_MODE in ARU_CTRL register to 1. In this mode all 4 registers ARU_[z]_DYN_ROUTE_LOW/_HIGH and ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH are connected as a ring, so all 24 DYN_READ_ID's can be used from both ARU's. The ring structure is shown in **Figure 38.26**. The data register shift direction is shown by the arrows in the ring.

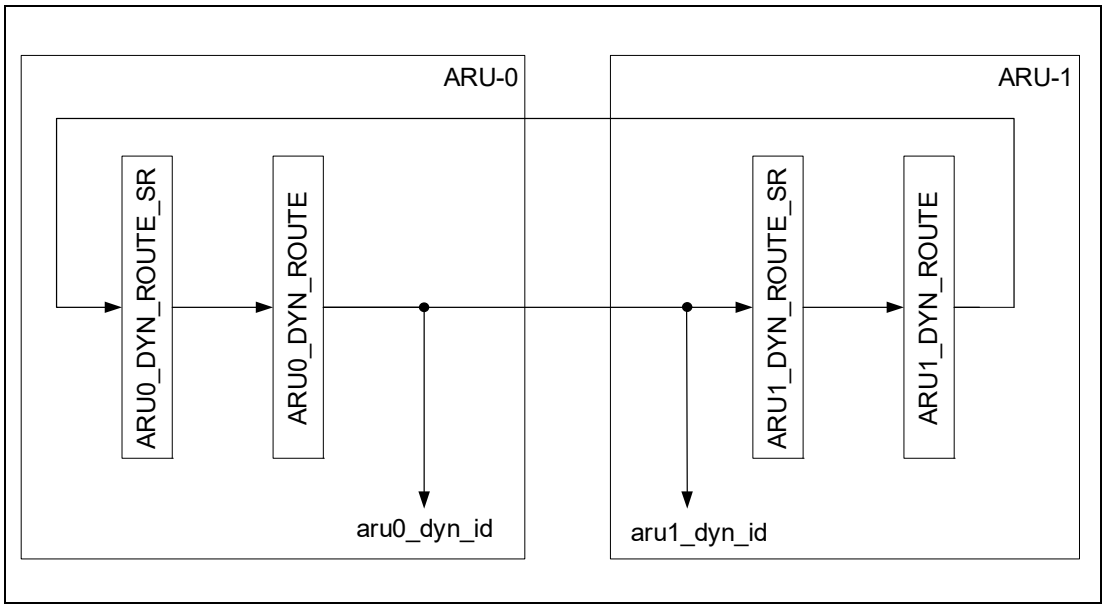


Figure 38.26 ARU dynamic routing - ring mode

Enabling the dynamic routing ring mode will automatically reset the caddr counter of both ARU-0 and ARU-1. This is necessary to synchronize both ARU's in this mode.

Enabling the dynamic routing ring mode will ignored DYN_ROUTE_SWAP and DYN_UPDATE_EN.

NOTE

DYN_ARU_UPDATE_EN should be disabled in dynamic routing ring mode.

It is possible to enable the dynamic routing ring mode for both ARU-0 and ARU-1 or only for one of the ARU's by setting the corresponding bit field ARU_0_DYN_EN/ARU_1_DYN_EN of the register ARU_CTRL. Because of the fact that to each ARU port ARU-0 and ARU-1 with the same ARU read ID two different GTM sub-modules are served it may make sense to enable ARU dynamic routing only for one port ARU-0 or ARU-1 if configured to ring-mode. The other port is then served in the default round robin manner.

In dynamic routing ring mode ARU_[z]_DYN_ROUTE_LOW/_HIGH and ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH are not write-protected.

NOTE

Avoid modification of ARU_[z]_DYN_ROUTE_LOW/_HIGH and ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH in active dynamic routing ring mode.

38.7.4.2 Dynamic routing – ARU controlled

Furthermore it is possible to reload the ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH registers by ARU itself.

Therefore the ARU has its own master port which will be served in the normal ARU routing scheme. The ARU read address for this master port has to be configured in the register ARU_[z]_DYN_RDADDR.

This feature can be enabled by setting bit DYN_ARU_UPDATE_EN of ARU_[z]_DYN_CTRL register.

The following mapping of the ARU word to the ARU_[z]_DYN_ROUTE_LOW/_HIGH registers is implemented: ARU_[z]_DYN_ROUTE_SR_LOW(23:0) = aru_data(23:0)
ARU_[z]_DYN_ROUTE_SR_HIGH(28:0) = aru_data(52:24)

The bit field aru_data(51:48) controls the configuration bits DYN_CLK_WAIT and the bit aru_data(52) controls the configuration bit DYN_UPDATE_EN. Both functions are described in the **Section 38.7.4.1**.

In opposite to the dynamic routing scheme controlled from CPU/AEI (only the 6 additional ARU master DYN_REA_ID's are inserted) two additional ID's are served. One is the ARU master ID itself for reloading and the other is the default ID-0. The ID-0 is only added to the inserted routing scheme if bit field DYN_CLK_WAIT of ARU_[z]_DYN_ROUTE_HIGH is set to zero (only the inserted routing scheme is executed). This ensures that a debug access can take place even if only the inserted routing scheme is executed.

The following dynamic routing scheme is executed for 15 > DYN_CLK_WAIT > 0: ... → ARU-master_ID → DYN_READ_ID0 → DYN_READ_ID1

→ DYN_READ_ID2 → DYN_READ_ID3 → DYN_READ_ID4 → DYN_READ_ID5 → ARU-master_ID → ...

The following dynamic routing scheme is executed for DYN_CLK_WAIT = 0 : ... → ARU-master_ID → DYN_READ_ID0 → DYN_READ_ID1 → DYN_READ_ID2 → DYN_READ_ID3 → DYN_READ_ID4 → DYN_READ_ID5 → default_ID0 → ARU-master_ID → ...

With the possibility of reloading the dynamic routing scheme over ARU, a FIFO or MCS is able to deliver the dynamic routing scheme data.

38.7.5 ARU Interrupt Signals

The following table describes ARU interrupt signals:

Table 38.48 ARU Interrupt Signals

Signal	Description
ARU_NEW_DATA0_IRQ	Indicates that data is transferred through the ARU using debug channel ARU_DBG_ACCESS0.
ARU_NEW_DATA1_IRQ	Indicates that data is transferred through the ARU using debug channel ARU_DBG_ACCESS1.
ARU_ACC_ACK_IRQ	ARU access acknowledge IRQ.

38.7.6 ARU Configuration Registers Overview

The following table shows a conclusion of configuration registers address offsets and initial values.

Table 38.49 Register List

Symbol	Register Name	Details in Section
ARU_ACCESS	ARU access register	**38.7.7.1**
ARU_DATA_H	ARU access register upper data word	**38.7.7.2**
ARU_DATA_L	ARU access register lower data word	**38.7.7.3**
ARU_DBG_ACCESS0	Debug access channel 0	**38.7.7.4**
ARU_DBG_DATA0_H	Debug access 0 transfer register upper data word	**38.7.7.5**
ARU_DBG_DATA0_L	Debug access 0 transfer register lower data word	**38.7.7.6**
ARU_DBG_ACCESS1	Debug access channel 0	**38.7.7.7**
ARU_DBG_DATA1_H	Debug access 1 transfer register upper data word	**38.7.7.8**
ARU_DBG_DATA1_L	Debug access 1 transfer register lower data word	**38.7.7.9**
ARU_IRQ_NOTIFY	ARU Interrupt notification register	**38.7.7.10**
ARU_IRQ_EN	ARU Interrupt enable register	**38.7.7.11**
ARU_IRQ_FORCINT	Register for forcing the ARU_NEW_DATA_IRQ interrupt	**38.7.7.12**
ARU_IRQ_MODE	IRQ mode configuration register	**38.7.7.13**
ARU_CADDR_END	ARU caddr counter end value	**38.7.7.14**
ARU_CADDR	ARU caddr counter value	**38.7.7.15**
ARU_CTRL	ARU enable dynamic routing	**38.7.7.16**
ARU_[z]_DYN_CTRL	ARU z dynamic routing control register	**38.7.7.17**
ARU_[z]_DYN_RDADDR	ARU z read ID for dynamic routing	**38.7.7.18**
ARU_[z]_DYN_ROUTE_LOW	ARU z lower bits of DYN_ROUTE register	**38.7.7.19**
ARU_[z]_DYN_ROUTE_HIGH	ARU z higher bits of DYN_ROUTE register	**38.7.7.20**
ARU_[z]_DYN_ROUTE_SR_LOW	ARU z shadow register for ARU_[z]_DYN_ROUTE_LOW	**38.7.7.21**
ARU_[z]_DYN_ROUTE_SR_HIGH	ARU z shadow register for ARU_[z]_DYN_ROUTE_HIGH	**38.7.7.22**

38.7.7 ARU Configuration Registers Description

38.7.7.1 ARU_ACCESS

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00280_H

Value after reset: 0000 $01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	WREQ	RREQ	—	—	—	ADDR								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.50 ARU_ACCESS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	WREQ	Initiate write request 0: No write request is pending 1: Mark data in registers ARU_DATA_H and ARU_DATA_L as valid **NOTES** 1. This bit is cleared automatically after transaction. Moreover, it can be cleared by software to cancel a write request. 2. WREQ bit is only writable if RREQ bit is zero, so to switch from WREQ to RREQ a cancel request has to be performed before. 3. Configuring both RREQ and WREQ bits results in a read request, so WREQ bit will not be set 4. The data is provided at address ADDR. This address has to be programmed as the source address in the destination sub-module channel. In worst case, the data is provided after one full ARU round trip.
12	RREQ	Initiate read request 0: No read request is pending 1: Set read request to source channel addressed by ADDR **NOTES** 1. This bit is cleared automatically after transaction. Moreover, it can be cleared by software to cancel a read request. 2. RREQ bit is only writable if WREQ bit is zero, so to switch from RREQ to WREQ a cancel request has to be performed before. 3. Configuring both RREQ and WREQ bits results in a read request, so RREQ bit will be set if the WREQ bit of the register isn't already set. 4. The ARU read request on address ADDR is served immediately when no other destination has actually a read request when the RREQ bit is set by CPU. In a worst case scenario, the read request is served after one round trip of the ARU, but this is only the case when every destination channel issues a read request at consecutive points in time.

Table 38.50 ARU_ACCESS Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ADDR	ARU address Define the ARU address used for transferring data **NOTES** 1. For an ARU write request, the preferred address 0_H have to be used. 2. A write request to the address $1FF_H$ (always full address) or $1FE_H$ (always empty address) are ignored and doesn't have any effect. 3. ARU address bits ADDR are only writable if RREQ and WREQ bits are zero

Note: The register ARU_ACCESS can be used either for reading or for writing at the same point in time.

38.7.7.2 ARU_DATA_H

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00284_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.51 ARU_DATA_H Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	DATA	Upper ARU data word Transfer upper ARU data word addressed by ADDR. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register

38.7.7.3 ARU_DATA_L

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00288_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.52 ARU_DATA_L Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	DATA	Lower ARU data word **NOTES** 1. Transfer lower ARU data word addressed by ADDR. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word are mapped to the data bits 24 to 28 of this register when data is read by the CPU. 2. For writing data into the ARU by the CPU the bits 24 to 28 are not transferred to bit 48 to 52 of the ARU word. Only bits 0 to 23 are written to bits 0 to 23 of the ARU word.

38.7.7.4 ARU_DBG_ACCESS0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0028C_H$

Value after reset: $0000\ 01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADDR								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.53 ARU_DBG_ACCESS0 Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ADDR	ARU debugging address Define address of ARU debugging channel 0.

38.7.7.5 ARU_DBG_DATA0_H

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00290_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.54 ARU_DBG_DATA0_H Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28 to 0	DATA	Upper debug data word **NOTES** 1. Transfer upper ARU data word addressed by register DBG_ACCESS0. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register. 2. The interrupt ARU_NEW_DATA0_IRQ is raised if a new data word is available.

38.7.7.6 ARU_DBG_DATA0_L

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00294_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.55 ARU_DBG_DATA0_L Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28 to 0	DATA	Lower debug data word **NOTES** 1. Transfer lower ARU data word addressed by register DBG_ACCESS0. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word is mapped to the data bits 24 to 28 of this register. 2. The interrupt ARU_NEW_DATA0_IRQ is raised if a new data word is available.

38.7.7.7 ARU_DBG_ACCESS1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00298_H

Value after reset: $0000\ 01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADDR								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.56 ARU_DBG_ACCESS1 Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ADDR	ARU debugging address Define address of ARU debugging channel 1.

38.7.7.8 ARU_DBG_DATA1_H

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0029C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.57 ARU_DBG_DATA1_H Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28 to 0	DATA	Upper debug data word **NOTES** 1. Transfer upper ARU data word addressed by register DBG_ACCESS1. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register. 2. The interrupt ARU_NEW_DATA1_IRQ is raised if a new data word is available.

38.7.7.9 ARU_DBG_DATA1_L

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $002A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.58 ARU_DBG_DATA1_L Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28 to 0	DATA	Lower debug data word **NOTES** 1. Transfer lower ARU data word addressed by register DBG_ACCESS1. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word is mapped to the data bits 24 to 28 of this register. 2. The interrupt ARU_NEW_DATA1_IRQ is raised if a new data word is available.

38.7.7.10 ARU_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACC_ACK	NEW_DATA1	NEW_DATA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.59 ARU_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ACC_ACK	AEI to ARU access finished, on read access data are valid **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
1	NEW_DATA1	Data was transferred for addr ARU_DBG_ACCESS1 0: No interrupt occurred 1: ARU_NEW_DATA1_IRQ interrupt was raised by the ARU **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	NEW_DATA0	Data was transferred for addr ARU_DBG_ACCESS0 0: No interrupt occurred 1: ARU_NEW_DATA0_IRQ interrupt was raised by the ARU **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

38.7.7.11 ARU_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACC_ACK_IRQ_EN	NEW_DATA1_IRQ_EN	NEW_DATA0_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.60 ARU_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ACC_ACK_IRQ_EN	ACC_ACK_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
1	NEW_DATA1_IRQ_EN	ARU_NEW_DATA1_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
0	NEW_DATA0_IRQ_EN	ARU_NEW_DATA0_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

38.7.7.12 ARU_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002AC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG_ACC_ACK	TRG_NEW_DAT1	TRG_NEW_DAT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.61 ARU_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TRG_ACC_ACK	Trigger ACC_ACK interrupt 0: Corresponding bit in status register will not be forced 1: Assert corresponding field in ARU_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.
1	TRG_NEW_DAT1	Trigger new data 1 interrupt 0: Corresponding bit in status register will not be forced 1: Assert corresponding field in ARU_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.
0	TRG_NEW_DAT0	Trigger new data 0 interrupt 0: Corresponding bit in status register will not be forced 1: Assert corresponding field in ARU_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.

38.7.7.13 ARU_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002B0_H$

Value after reset: $0000\ 000X_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.62 ARU_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.7.7.14 ARU_CADDR_END

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002B4_H$

Value after reset: $0000\ 00XX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CADDR_END						
Value after reset	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.63 ARU_CADDR_END Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CADDR_END	Set end value of ARU caddr counter **NOTES** 1. The ARU roundtrip counter aru_caddr runs from zero to caddr_end value. 2. Shorten the ARU roundtrip cycle by setting a smaller number than the defined reset value will cause that not all ARU-connected modules will be served. 3. Making the roundtrip cycle longer than the reset value would cause longer ARU roundtrip time and as a result some ARU-connected modules will not be served as fast as possible for this device. 4. The reset value is device-specific. For more information see **Section 38.28, GTM Device 358**. 5. This bit is write protected by bit RF_PROT of register GTM_CTRL.

38.7.7.15 ARU_CADDR

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $002FC_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CADDR_1						
Value after reset	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CADDR_0						
Value after reset	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.64 ARU_CADDR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 16	CADDR_1	CADDR_1 value of ARU-0 caddr counter
15 to 7	Reserved	When read, the value after reset is returned.
6 to 0	CADDR_0	CADDR_0 value of ARU-1 caddr counter

Note: The registers CADDR_0 and CADDR_1 start incrementing with each clock cycle just after reset. Due to this the initial reset value cannot be read back.

38.7.7.16 ARU_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002BC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ARU_DYN_RING_MODE	ARU_1_DYN_EN		ARU_0_DYN_EN	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.65 ARU_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ARU_DYN_RING_MODE	ARU_DYN_RING_MODE enable dynamic routing ring mode 0_B: different dynamic routing scheme for ARU-0 and ARU-1 1_B: same dynamic routing scheme for ARU-0 and ARU-1 with 24 possible read-ID's (dynamic routing ring mode) Dynamic routing ring mode for both ARU-0 and ARU-1
3, 2	ARU_1_DYN_EN	ARU_1_DYN_EN enable dynamic routing for ARU-1 00_B: No change 01_B: Disable dynamic routing 10_B: Enable dynamic routing 11_B: No change Dynamic routing enable of ARU-1. Write of following double bit values is possible: **NOTE** If dynamic routing is disabled, the normal ARU routing scheme for ARU-1 is executed.
1, 0	ARU_0_DYN_EN	ARU_0_DYN_EN enable dynamic routing for ARU-0 00_B: No change 01_B: Disable dynamic routing 10_B: Enable dynamic routing 11_B: No change Dynamic routing enable of ARU-0. Write of following double bit values is possible: **NOTE** If dynamic routing is disabled, the normal ARU routing scheme for ARU-0 is executed.

38.7.7.17 ARU_[z]_DYN_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002C0_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DYN_ROUTE_SWAP	DYN_ARU_UPDATE_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.66 ARU_[z]_DYN_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DYN_ROUTE_SWAP	DYN_ROUTE_SWAP enable swapping DYN_ROUTE_SR with DYN_ROUTE register Enable swapping DYN_ROUTE_SR with DYN_ROUTE register.
0	DYN_ARU_UPDATE_EN	DYN_ARU_UPDATE_EN enable reload of DYN_ROUTE register from ARU itself Enable reload of DYN_ROUTE register from ARU itself.

38.7.7.18 ARU_[z]_DYN_RDADDR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002E8_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DYN_ARU_RDADDR								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.67 ARU_[z]_DYN_RDADDR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	DYN_ARU_RDADDR	DYN_ARU_RDADDR ARU read address ID to reload the DYN_ROUTE register ARU read address ID to reload the DYN_ROUTE register from ARU itself.

38.7.7.19 ARU_[z]_DYN_ROUTE_LOW

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002C8_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DYN_READ_ID2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DYN_READ_ID1								DYN_READ_ID0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.68 ARU_[z]_DYN_ROUTE_LOW Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	DYN_READ_ID2	DYN_READ_ID2 ARU read ID 2 ARU read ID 2 for dynamic routing
15 to 8	DYN_READ_ID1	DYN_READ_ID1 ARU read ID 1 ARU read ID 1 for dynamic routing
7 to 0	DYN_READ_ID0	DYN_READ_ID0 ARU read ID 0 ARU read ID 0 for dynamic routing

38.7.7.20 ARU_[z]_DYN_ROUTE_HIGH

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002D0_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DYN_CLK_WAIT				DYN_READ_ID5							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DYN_READ_ID4								DYN_READ_ID3							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.69 ARU_[z]_DYN_ROUTE_HIGH Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	DYN_CLK_WAIT	DYN_CLK_WAIT number of clk cycles for dynamic routing Defines the number of clk cycles between each dynamic routing ID.
23 to 16	DYN_READ_ID5	DYN_READ_ID5 ARU read ID 5 ARU read ID 5 for dynamic routing
15 to 8	DYN_READ_ID4	DYN_READ_ID4 ARU read ID 4 ARU read ID 4 for dynamic routing
7 to 0	DYN_READ_ID3	DYN_READ_ID3 ARU read ID 3 ARU read ID 3 for dynamic routing

38.7.7.21 ARU_[z]_DYN_ROUTE_SR_LOW

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002D8_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DYN_READ_ID8							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DYN_READ_ID7								DYN_READ_ID6							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.70 ARU_[z]_DYN_ROUTE_SR_LOW Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	DYN_READ_ID8	DYN_READ_ID8 ARU read ID 8 ARU read ID 8 for dynamic routing **NOTE** These bits are mapped to ARU data bits aru_data(23:16).
15 to 8	DYN_READ_ID7	DYN_READ_ID7 ARU read ID 7 ARU read ID 7 for dynamic routing **NOTE** These bits are mapped to ARU data bits aru_data(15:8).
7 to 0	DYN_READ_ID6	DYN_READ_ID6 ARU read ID 6 ARU read ID 6 for dynamic routing **NOTE** These bits are mapped to ARU data bits aru_data(7:0).

NOTE

This is the shadow register for **Section 38.7.7.20, ARU_[z]_DYN_ROUTE_HIGH**.

38.7.7.22 ARU_[z]_DYN_ROUTE_SR_HIGH

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $002E0_H$ + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DYN_UPDATE_EN	DYN_CLK_WAIT				DYN_READ_ID11							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DYN_READ_ID10								DYN_READ_ID9							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.71 ARU_[z]_DYN_ROUTE_SR_HIGH Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	DYN_UPDATE_EN	DYN_UPDATE_EN update enable from shadow register Enable update ARU_[z]_DYN_ROUTE_LOW/_HIGH registers from shadow registers ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH. **NOTE** This bit is mapped to ARU data bits aru_data(52).
27 to 24	DYN_CLK_WAIT	DYN_CLK_WAIT number of clk cycles for dynamic routing Defines the number of clk cycles between each dynamic routing ID. **NOTE** These bits are mapped to ARU data bits aru_data(51:48).
23 to 16	DYN_READ_ID11	DYN_READ_ID11 ARU read ID 11 ARU read ID 11 for dynamic routing. **NOTE** These bits are mapped to ARU data bits aru_data(47:40).
15 to 8	DYN_READ_ID10	DYN_READ_ID10 ARU read ID 10 ARU read ID 10 for dynamic routing. **NOTE** These bits are mapped to ARU data bits aru_data(39:32).
7 to 0	DYN_READ_ID9	DYN_READ_ID9 ARU read ID 9 ARU read ID 9 for dynamic routing. **NOTE** These bits are mapped to ARU data bits aru_data(31:24).

NOTE

This is the shadow register for register **Section 38.7.7.20, ARU_[z]_DYN_ROUTE_HIGH**.

38.8 Broadcast Module (BRC)

38.8.1 Overview

Since each write address for the sub-module channels of the GTM-IP that are able to write to the ARU can only be read by a single module, it is impossible to provide a data stream to different modules in parallel (This statement holds not for sources, which do not invalidate their data after the data were read by any consumer, e.g. DPLL).

To overcome this issue for regular modules, the sub-module Broadcast (BRC) enables to duplicate data streams multiple times.

The BRC sub-module provides 12 input channels as well as 22 output channels.

In order to clone an incoming data stream, the corresponding input channel can be mapped to zero or more output channels.

When mapped to zero no channel is read.

To destroy an incoming data stream, the EN_TRASHBIN bit inside the BRC_SRC_[z]_DEST register has to be set.

The total number of output channels that are assigned to a single input channel is variable. However, the total number of assigned output channels must be less than or equal to 22.

38.8.2 BRC Configuration

As it is the case with all other sub-modules connected to the ARU, the input channels can read arbitrary ARU address locations and the output channels provide the broadcast data to fixed ARU write address locations.

The associated write addresses for the BRC sub-module are fixed and can be obtained from **Section 38.27, Appendix A**.

The read address for each input channel is defined by the corresponding register BRC_SRC_[z]_ADDR (z: 0 to 11).

The mapping of an input channel to several output channels is defined by setting the appropriate bits in the register BRC_SRC_[z]_DEST (z: 0 to 11). Each output channel is represented by a single bit in the register BRC_SRC_[z]_DEST. The address of the output channel is defined in **Section 38.27, Appendix A**.

If no output channel bit is set within a register BRC_SRC_[z]_DEST, no data is provided to the corresponding ARU write address location from the defined read input specified by BRC_SRC_[z]_ADDR. This means that the channel does not broadcast any data and is disabled (reset state).

Besides the possibility of mapping an input channel to several output channels, the bit EN_TRASHBIN of register BRC_SRC_[z]_DEST may be set, which results in dropping an incoming data stream. In this case the data of an input channel defined by BRC_SRC_[z]_ADDR is consumed by the BRC module and not routed to any succeeding sub-module. In consequence, the output channels defined in the register BRC_SRC_[z]_DEST are ignored. Therefore, the bits 0 to 21 are set to zero (0) when trash bin functionality is enabled.

In general, the BRC sub-module can work in two independent operation modes. In the first operation mode the data consistency is guaranteed since a BRC channel requests only new data from a source when all destination channels for the BRC have consumed the old data value. This mode is called Data Consistency Mode (DCM).

In a second operation mode the BRC channel always requests data from a source and distributes this data to the destination regardless whether all destinations have already consumed the old data. This mode is called Maximum Throughput Mode (MTM).

MTM ensures that always the newest available data is routed through the system, while it is not guaranteed data consistency since some of the destination channels can be provided with the old data while some other destination channels are provided with the new data. If this is the case, the Data Inconsistency Detected Interrupt BRC_DID_IRQ[x] is raised but the channel continues to work.

Furthermore in MTM mode it is guaranteed that it is not possible to read a data twice by a read channel. This is blocked.

The channel mode can be configured inside the BRC_SRC_[z]_ADDR register.

To avoid invalid configurations of the registers BRC_SRC_[z]_DEST, the BRC also implements a plausibility check for these configurations.

If the software assigns an already used output channel to a second input channel, BRC performs an auto correction of the lastly configured register BRC_SRC_[z]_DEST and it triggers the interrupt BRC_DEST_ERR_IRQ.

Consider the following example for clarification of the auto correction mechanism. Assume that the following configuration of the 22 lower significant bits for the registers

BRC_SRC_[z]_DEST :

BRC_SRC_0_DEST : 00 0000 0000 1000 1000 0000_B

BRC_SRC_1_DEST : 00 0000 0000 0100 0000 0100_B

BRC_SRC_2_DEST : 00 0000 0000 0001 0100 0010_B

BRC_SRC_3_DEST : 00 0000 0000 0010 0001 1001_B

If the software overwrites the value for register BRC_SRC_2_DEST with BRC_SRC_2_DEST : 00 0000 0000 1001 0010 0010_B (changed bits are underlined), then the BRC releases an BRC_DEST_ERR_IRQ interrupt since bit 11 is already assigned in register BRC_SRC_0_DEST. The auto correction forces bit 11 to be cleared. The modifications of the bits 5 and 6 are accepted, since there is no violation with previous configurations. So the result of the write access mentioned above results in the following modified register configuration: BRC_SRC_2_DEST : 00 0000 0000 0001 0010 0010_B

For debug purposes, the interrupt BRC_DEST_ERR_IRQ can also be released by writing to register BRC_IRQ_FORCINT. Nevertheless, the interrupt has to be enabled to be visible outside of the GTM-IP.

38.8.3 BRC Interrupt Signals

Signal	Description
BRC_DEST_ERR_IRQ	Indicating configuration errors for BRC module
BRC_DID_IRQ[x]	Data inconsistency occurred in MTM mode (x:0..11)

38.8.4 BRC Configuration Registers Overview

BRC contains following configuration registers:

Table 38.72 Register List

Symbol	Register Name	Details in Section
BRC_SRC_[z]_ADDR	BRC read address for input channel z	**38.8.5.1**
BRC_SRC_[z]_DEST	BRC destination channels for input channel z	**38.8.5.2**
BRC_IRQ_NOTIFY	BRC interrupt notification register	**38.8.5.3**
BRC_IRQ_EN	BRC interrupt enable register	**38.8.5.4**
BRC_IRQ_FORCINT	BRC force interrupt register	**38.8.5.5**
BRC_IRQ_MODE	BRC interrupt mode configuration register	**38.8.5.6**
BRC_EIRQ_EN	BRC error interrupt enable register	**38.8.5.7**
BRC_RST	BRC software reset register	**38.8.5.8**

38.8.5 BRC Configuration Registers Description

38.8.5.1 BRC_SRC_[z]_ADDR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00400_H + (8_H × z)

Value after reset: 0000 $01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	BRC_MODE	—	—	—	ADDR								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.73 BRC_SRC_[z]_ADDR Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	BRC_MODE	BRC Operation mode select. 0: Consistency Mode (DCM) selected 1: Maximum Throughput Mode (MTM) selected This bit field is only writable if channel is disabled.
11 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ADDR	Source ARU address. Define an ARU read address used as data source for input channel z (z = 0 to 11). This bit field is only writable if channel is disabled.

38.8.5.2 BRC_SRC_[z]_DEST

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00404_H + (8_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	EN_TRASHBIN	EN_DEST21	EN_DEST20	EN_DEST19	EN_DEST18	EN_DEST17	EN_DEST16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EN_DEST15	EN_DEST14	EN_DEST13	EN_DEST12	EN_DEST11	EN_DEST10	EN_DEST9	EN_DEST8	EN_DEST7	EN_DEST6	EN_DEST5	EN_DEST4	EN_DES3	EN_DEST2	EN_DEST1	EN_DEST0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.74 BRC_SRC_[z]_DEST Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	EN_TRASHBIN	EN_TRASHBIN: Control trash bin functionality. 0: Trash bin functionality disabled 1: Trash bin functionality enabled **NOTE** When bit EN_TRASHBIN is enabled bits 0 to 21 are ignored for this input channel. Therefore, the bits 0 to 21 are set to zero (0) when trash bin functionality is enabled.
21 to 0	EN_DEST[21:0]	Enable BRC destination address k 0: Destination address k not mapped to source BRC_SRC_[z]_ADDR 1: Destination address k mapped to source BRC_SRC_[z]_ADDR **NOTE** The destination address k for BRC channel is defined in **Section 38.28.10.1**.

NOTES

1. The bits 0 to 21 are cleared by auto correction mechanism if a destination channel is assigned to multiple source channels.
2. When a BRC input channel is disabled (all EN_DEST[k] (k = 0 to 21) bits are reset to zero (0)), the internal states are reset to their reset value.

38.8.5.3 BRC_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00460_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DID11	DID10	DID9	DID8	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0	DEST_ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.75 BRC_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 1	DID[11:0]	Data inconsistency occurred in MTM mode. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	DEST_ERR	Configuration error interrupt for BRC sub-module 0: No BRC configuration error occurred 1: BRC configuration error occurred **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

38.8.5.4 BRC_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00464_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DID_IRQ_EN11	DID_IRQ_EN10	DID_IRQ_EN9	DID_IRQ_EN8	DID_IRQ_EN7	DID_IRQ_EN6	DID_IRQ_EN5	DID_IRQ_EN4	DID_IRQ_EN3	DID_IRQ_EN2	DID_IRQ_EN1	DID_IRQ_EN0	DEST_ERR_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.76 BRC_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 1	DID_IRQ_EN [11:0]	Enable BRC_DID_IRQ for channel 11 to 0, see bit 0 for description.
0	DEST_ERR_IRQ_EN	BRC_DEST_ERR_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

38.8.5.5 BRC_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00468_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TRG_DID11	TRG_DID10	TRG_DID9	TRG_DID8	TRG_DID8	TRG_DID6	TRG_DID5	TRG_DID4	TRG_DID3	TRG_DID2	TRG_DID1	TRG_DID0	TRG_DEST_ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.77 BRC_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 1	TRG_DID [11:0]	Trigger DID interrupt, see bit 0 for description.
0	TRG_DEST_ERR	Trigger destination error interrupt. 0: corresponding bit in status register will not be forced. 1: Assert corresponding field in BRC_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.

38.8.5.6 BRC_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0046C_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.78 BRC_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1 to 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.8.5.7 BRC_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00474_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DID_EIRQ_EN11	DID_EIRQ_EN10	DID_EIRQ_EN9	DID_EIRQ_EN8	DID_EIRQ_EN7	DID_EIRQ_EN6	DID_EIRQ_EN6	DID_EIRQ_EN4	DID_EIRQ_EN3	DID_EIRQ_EN2	DID_EIRQ_EN1	DID_EIRQ_EN0	DEST_ERR_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.79 BRC_EIRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 1	DID_EIRQ_EN [11:0]	Enable BRC_DID_EIRQ for channel 11 to 0, see bit 0 for description.
0	DEST_ERR_EIRQ_EN	BRC_DEST_ERR_EIRQ error interrupt enable 0: Disable error interrupt, error interrupt is not visible outside GTM-IP 1: Enable error interrupt, error interrupt is visible outside GTM-IP

38.8.5.8 BRC_RST

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00470_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.80 BRC_RST Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	RST	Software reset 0: No action 1: Reset BRC **NOTE** This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately.

38.9 First In First Out Module (FIFO)

38.9.1 Overview

The FIFO unit is the storage part of the PSM sub-module. The F2A described in **Section 38.11, FIFO to ARU Unit (F2A)** and the AFD described in **Section 38.10, AEI to FIFO Data Interface (AFD)** implement the interface part of the FIFO sub-module to the ARU and the AEI bus. Each FIFO unit embeds eight logical FIFOs. These logical FIFOs are configurable in the following manner:

1. FIFO size (defines start and end address)
2. FIFO operation modes (normal mode or ring buffer operation mode)
3. Fill level control / memory region read protection

Each logical FIFO represents a data stream between the sub-modules of the GTM and the microcontroller connected to AFD sub-module (see **Section 38.10**). The FIFO RAM counts 1K words, where the word size is 29 bit. This gives the freedom to program or receive 24 bit of data together with the five control bits inside an ARU data word.

The FIFO unit provides three ports for accessing its content. One port is connected to the F2A interface, one port is connected to the AFD interface and one port has its own AEI bus interface.

The AFD interface has always the highest priority. Accesses to the FIFO from AFD interface and direct AEI interface in parallel — which means at the same time — is not possible, because both interfaces are driven from the same AEI bus interface of the GTM.

The priority between F2A and direct AEI interface can be defined by software. This can be done by using the register FIFO[i]_CH[z]_CTRL for all FIFO channels of the sub-module.

The FIFO is organized as a single RAM that is also accessible through the FIFO AEI interface connected to one of the FIFO ports. To provide the direct RAM access, the RAM is mapped into the address space of the microcontroller. The addresses for accessing the RAM via AEI can be found in **Section 38.28, GTM Device 358**.

After reset, the FIFO RAM isn't initialized by hardware.

The FIFO channels can be flushed individually. Each of the eight FIFO channels can be used either in normal FIFO operation mode or in ring buffer operation mode.

Beside the possibility of flushing each FIFO channel directly, a write access to FIFO[i]_CH[z]_END_ADDR or to FIFO[i]_CH[z]_START_ADDR will also flush the regarding channel which means that the read and write pointer and also the fill level of the regarding channel will be reset. In consequence of this existing data in the concerned FIFO channel are not longer valid- thereafter the channel is empty.

38.9.2 Operation Modes

38.9.2.1 FIFO Operation Mode

In normal FIFO operation mode the content of the FIFO is written and read in first-in first-out order, where the data is destroyed after it is delivered to the system bus or the F2A sub-module (see **Section 38.11**).

The upper and lower watermark registers (registers FIFO[i]_CH[z]_UPPER_WM and FIFO[i]_CH[z]_LOWER_WM) are used for controlling the FIFO s fill level. If the fill level falls below the lower watermark or it exceeds the upper watermark, an interrupt signal is triggered by the FIFO sub-module if enabled inside the FIFO[i]_IRQ_EN.

The interrupt signals are sent to the Interrupt Concentrator Module (ICM) (see **Section 38.24**). The ICM can also initiate specific DMA transfers.

38.9.2.2 Ring Buffer Operation Mode

The ring buffer mode can be used to provide a continuous data or configuration stream to the other GTM sub-modules without CPU interaction. In ring buffer mode the FIFO provides a continuous data stream to the F2A sub-module. The first word of the FIFO is delivered first and after the last word is provided by the FIFO to the ARU, the first word can be obtained again.

If in ring buffer mode the read pointer reaches the write pointer it will be set again to the configured start address. So the read pointer always rotates cyclic between the configured start address of the regarding FIFO channel (first written data) and the write pointer which points to the last written data of the channel.

It is possible to add data to the FIFO channel via the AEI to FIFO interface (AFD) using the register AFD[i]_CH[z]_BUF_ACC while running in ring buffer mode. The new written data will be added in the next ring buffer cycle. However, the register AFD[i]_CH[z]_BUF_ACC should not be read in ring buffer mode.

It is recommended to fill the FIFO channel first before enabling the data stream in the FIFO to ARU interface (F2A).

Modifications of the continuous data stream can be achieved by using direct memory access which is provided by the FIFO AEI interface.

38.9.2.3 DMA Hysteresis Mode

The DMA hysteresis mode can be enabled by setting bit DMA_HYSTERESIS = 1 in the FIFO[i]_CH[z]_IRQ_MODE register.

In the DMA hysteresis mode the lower and upper watermark will be masked to generate the DMA request (= fifo_irq) in the following manner.

If a DMA is writing data to a FIFO (configured by setting bit DMA_HYST_DIR = 1 in register FIFO[i]_CH[z]_IRQ_MODE), the DMA request will be generated by the lower watermark. The upper watermark does not generate a DMA request. The next DMA request will be generated by the next lower watermark until the upper watermark was reached.

If a DMA is reading data from a FIFO (configured by setting bit DMA_HYST_DIR = 0 in register FIFO[i]_CH[z]_IRQ_MODE), the DMA request will be generated by the upper watermark. The lower watermark does not generate a DMA request. The next DMA request will be generated by the next upper watermark until the lower watermark was reached.

NOTE

The watermarks have to achieve the following condition depending on the irq mode.

1. Level/Pulse/Pulse-Notify mode : upper watermark > lower watermark
2. Single-Pulse mode : upper watermark > lower watermark + 1

38.9.3 FIFO Interrupt Signals

Signal	Description
FIFO[i]*_CH[z]_EMPTY*	Indicating empty FIFO z (z = 0 to 7) was reached
FIFO[i]*_CH[z]_FULL*	Indicating full FIFO z (z = 0 to 7) was reached
FIFO[i]*_CH[z]_LOWER_WM*	Indicating FIFO z (z = 0 to 7) reached lower watermark.
FIFO[i]*_CH[z]_UPPER_WM*	Indicating FIFO z (z = 0 to 7) reached upper watermark.

38.9.4 FIFO Configuration Registers Overview

FIFO contains following configuration registers:

Table 38.81 Register List

Symbol	Register Name	Details in Section
FIFO[i]_CH[z]_CTRL	FIFOi channel z control register	**38.9.5.1**
FIFO[i]_CH[z]_END_ADDR	FIFOi channel z end address register	**38.9.5.2**
FIFO[i]_CH[z]_START_ADDR	FIFOi channel z start address register	**38.9.5.3**
FIFO[i]_CH[z]_UPPER_WM	FIFOi channel z upper watermark register	**38.9.5.4**
FIFO[i]_CH[z]_LOWER_WM	FIFOi channel z lower watermark register	**38.9.5.5**
FIFO[i]_CH[z]_STATUS	FIFOi channel z status register	**38.9.5.6**
FIFO[i]_CH[z]_FILL_LEVEL	FIFOi channel z fill level register	**38.9.5.7**
FIFO[i]_CH[z]_WR_PTR	FIFOi channel z write pointer register	**38.9.5.8**
FIFO[i]_CH[z]_RD_PTR	FIFOi channel z read pointer register	**38.9.5.9**
FIFO[i]_CH[z]_IRQ_NOTIFY	FIFOi channel z interrupt notification register	**38.9.5.10**
FIFO[i]_CH[z]_IRQ_EN	FIFOi channel z interrupt enable register	**38.9.5.11**
FIFO[i]_CH[z]_IRQ_FORCINT	FIFOi channel z force interrupt register	**38.9.5.12**
FIFO[i]_CH[z]_IRQ_MODE	FIFOi channel z interrupt mode control register	**38.9.5.13**
FIFO[i]_CH[z]_EIRQ_EN	FIFOi channel z error interrupt enable register	**38.9.5.14**
FIFO[i]_MEMORY	FIFO memory location	**38.9.5.15**

38.9.5 FIFO Configuration Registers Description

38.9.5.1 FIFO[i]_CH[z]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18400_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	WULOCK	FLUSH	RAP	RBM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.82 FIFO[i]_CH[z]_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	WULOCK	RAM write unlock. Enable/disable direct RAM write access to the memory mapped FIFO region. 0: Direct RAM write access disabled 1: Direct RAM write access enabled **NOTE** Only the bit WULOCK of register FIFO[i]_CH0_CTRL enables/disables the direct RAM write access for all FIFO channel (whole FIFO RAM). The WULOCK bits of the other channels are writable but have no effect.
2	FLUSH	FIFO Flush control 0: Normal operation 1: Execute FIFO flush (bit is automatically cleared after flush) **NOTE** A FIFO Flush operation resets the FIFO[i]_CH[z]_FILL_LEVEL, FIFO[i]_CH[z]_WR_PTR and FIFO[i]_CH[z]_RD_PTR registers to their initial values.
1	RAP	RAM access priority 0: FIFO ports have higher access priority than AEI-IF 1: AEI-IF has higher access priority than FIFO ports **NOTE** The RAP bit is only functional in register FIFO_0_CTRL. The priority is defined for all FIFO channels there.
0	RBM	Ring buffer mode enable 0: Normal FIFO operation mode 1: Ring buffer mode

38.9.5.2 FIFO[i]_CH[z]_END_ADDR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18404_H + (4000_H × i) + (40_H × z)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.83 FIFO[i]_CH[z]_END_ADDR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	ADDR	End address for FIFO channel z (z = 0 to 7) **NOTES** 1. Value for ADDR is calculated as ADDR = 128 × (z + 1) - 1 2. A write access will flush the regarding channel

38.9.5.3 FIFO[i]_CH[z]_START_ADDR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18408_H + (4000_H × i) + (40_H × z)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.84 FIFO[i]_CH[z]_START_ADDR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	ADDR	Start address for FIFO channel z (z = 0 to 7) **NOTES** 1. Initial value for ADDR is calculated as ADDR = 128 × z 2. A write access will flush the regarding channel

38.9.5.4 FIFO[i]_CH[z]_UPPER_WM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $1840C_H$ + (4000_H × i) + (40_H × z)

Value after reset: $0000\ 0060_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.85 FIFO[i]_CH[z]_UPPER_WM Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	ADDR	Upper watermark address. **NOTE** The upper watermark is configured as a relative fill level of the FIFO. ADDR must be in range: 0 ≤ ADDR ≤ FIFO[i]_CH[z]_END_ADDR – FIFO[i]_CH[z]_START_ADDR. Initial value for ADDR is defined as ADDR = 60_H.

38.9.5.5 FIFO[i]_CH[z]_LOWER_WM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18410_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0020_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.86 FIFO[i]_CH[z]_LOWER_WM Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	ADDR	Lower watermark address. **NOTE** The lower watermark is configured as a relative fill level of the FIFO. ADDR must be in range: 0 ≤ ADDR ≤ FIFO[i]_CH[z]_END_ADDR – FIFO[i]_CH[z]_START_ADDR. Initial value for ADDR is defined as ADDR = 20_H.

38.9.5.6 FIFO[i]_CH[z]_STATUS

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 18414_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	UP_WM	LOW_WM	FULL	EMPTY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.87 FIFO[i]_CH[z]_STATUS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	UP_WM	Upper watermark reached 0: Fill level < upper watermark 1: Fill level ≥ upper watermark **NOTE** Bit only applicable in normal mode.
2	LOW_WM	Lower watermark reached 0: Fill level > lower watermark 1: Fill level ≤ lower watermark **NOTE** Bit only applicable in normal mode.
1	FULL	FIFO is full 0: Fill level < FIFO[i]_CH[z]_END_ADDR – FIFO[i]_CH[z]_START_ADDR + 1 1: Fill level = FIFO[i]_CH[z]_END_ADDR – FIFO[i]_CH[z]_START_ADDR + 1 **NOTE** Bit only applicable in normal mode.
0	EMPTY	FIFO is empty. 0: Fill level > 0 1: Fill level = 0 **NOTE** Bit only applicable in normal mode.

38.9.5.7 FIFO[i]_CH[z]_FILL_LEVEL

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 18418_H + (4000_H × i) + (40_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LEVEL										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.88 FIFO[i]_CH[z]_FILL_LEVEL Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10 to 0	LEVEL	Fill level of the current FIFO **NOTE** LEVEL is in range: 0 ≤ LEVEL ≤ FIFO[i]_CH[z]_END_ADDR - FIFO[i]_CH[z]_START_ADDR + 1. Register content is compared to the upper and lower watermark values for this channel to detect watermark over– and underflow.

38.9.5.8 FIFO[i]_CH[z]_WR_PTR

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $1841C_H$ + (4000_H × i) + (40_H × z)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.89 FIFO[i]_CH[z]_WR_PTR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9 to 0	ADDR	Position of the write pointer **NOTE** ADDR must be in range 0 ≤ ADDR ≤ 1023. Initial value for ADDR is defined as ADDR = FIFO[i]_CH[z]_START_ADDR.

38.9.5.9 FIFO[i]_CH[z]_RD_PTR

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 18420_H + (4000_H × i) + (40_H × z)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ADDR									
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.90 FIFO[i]_CH[z]_RD_PTR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9 to 0	ADDR	Position of the read pointer **NOTE** ADDR must be in range 0 ≤ ADDR≤ 1023. Initial value for ADDR is defined as ADDR = FIFO[i]_CH[z]_START_ADDR

38.9.5.10 FIFO[i]_CH[z]_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18424_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FIFO_UWM	FIFO_LWM	FIFO_FULL	FIFO_EMPTY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.91 FIFO[i]_CH[z]_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FIFO_UWM	FIFO Upper watermark was overrun. See bit 0.
2	FIFO_LWM	FIFO Lower watermark was under–run. See bit 0.
1	FIFO_FULL	FIFO is full. See bit 0.
0	FIFO_EMPTY	FIFO is empty 0: No interrupt occurred. 1: FIFO is empty interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.

38.9.5.11 FIFO[i]_CH[z]_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18428_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FIFO_UWM_IRQ_EN	FIFO_LWM_IRQ_EN	FIFO_FULL_IRQ_EN	FIFO_EMPTY_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.92 FIFO[i]_CH[z]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FIFO_UWM_IRQ_EN	Interrupt enable. See bit 0.
2	FIFO_LWM_IRQ_EN	Interrupt enable. See bit 0.
1	FIFO_FULL_IRQ_EN	Interrupt enable. See bit 0.
0	FIFO_EMPTY_IRQ_EN	Interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.

38.9.5.12 FIFO[i]_CH[z]_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $1842C_H$ + (4000_H × i) + (40_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	TRG_FIFO_UWM	TRG_FIFO_LWM	TRG_FIFO_FULL	TRG_FIFO_EMPTY
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.93 FIFO[i]_CH[z]_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	TRG_FIFO_UWM	Force interrupt of upper watermark. See bit 0.
2	TRG_FIFO_LWM	Force interrupt of lower watermark. See bit 0.
1	TRG_FIFO_FULL	Force interrupt of FIFO full status. See bit 0.
0	TRG_FIFO_EMPTY	Force interrupt of FIFO empty status. 0: Corresponding bit in status register will not be forced 1: Assert corresponding field in FIFO[i]_CH[z]_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.

38.9.5.13 FIFO[i]_CH[z]_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18430_H + (4000_H × i) + (40_H × z)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DMA_HYST_DIR	DMA_HYSTERESIS	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.94 FIFO[i]_CH[z]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	DMA_HYST_DIR	DMA direction in hysteresis mode. 0: DMA direction read in hysteresis mode. 1: DMA direction write in hysteresis mode. **NOTES** 1. In the case of DMA writing data to a FIFO the DMA requests must be generated by the lower watermark. If the DMA hysteresis is enabled, the FIFO does not generate a new DMA request until the upper watermark is reached. 2. In the case of DMA reading data from FIFO the DMA requests must be generated by the upper watermark. If the DMA hysteresis is enabled, the FIFO does not generate a new DMA request until the lower watermark is reached.
2	DMA_HYSTERESIS	Enable DMA hysteresis mode. 0: Disable FIFO hysteresis for DMA access. 1: Enable FIFO hysteresis for DMA access.
1, 0	IRQ_MODE	IRQ mode selection. 00_B: Level mode 01_B: Pulse mode 10_B: Pulse–Notify mode 11_B: Single–Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.9.5.14 FIFO[i]_CH[z]_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18434_H + (4000_H × i) + (40_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FIFO_UWM_EIRQ_EN	FIFO_LWM_EIRQ_EN	FIFO_FULL_EIRQ_EN	FIFO_EMPTY_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.95 FIFO[i]_CH[z]_EIRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FIFO_UWM_EIRQ_EN	Interrupt enable. See bit 0.
2	FIFO_LWM_EIRQ_EN	Interrupt enable. See bit 0.
1	FIFO_FULL_EIRQ_EN	Interrupt enable. See bit 0.
0	FIFO_EMPTY_EIRQ_EN	Error interrupt enable 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.

38.9.5.15 FIFO[i]_MEMORY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 19000_H + (4000_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.96 FIFO[i]_MEMORY Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	DATA	FIFO memory location. **NOTE** Read as zero, should be written as zero.

38.10 AEI to FIFO Data Interface (AFD)

38.10.1 Overview

The AFD sub-module implements a data interface between the AEI bus and the FIFO sub-module, which consists of eight logical FIFO channels.

The AFD sub-module provides one buffer registers that are dedicated to the logical channels of the FIFO. Access to the corresponding FIFO channel is given by reading or writing this buffer registers AFD[i]_CH[z]_BUF_ACC.

An AEI write access to the buffer register where the corresponding FIFO channel is full will be ignored. The data will be lost.

An AEI read access to the buffer register where the corresponding FIFO channel is empty will be served with zero data.

38.10.2 AFD Configuration Registers Overview

AFD contains following configuration registers:

Table 38.97 Register List

Symbol	Register Name	Details in Section
AFD[i]_CH[z]_BUF_ACC	AFD i FIFO z buffer access register	**38.10.3.1**

38.10.3 AFD Configuration Register Description

38.10.3.1 AFD[i]_CH[z]_BUF_ACC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18080_H + (4000_H × i) + (10_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DATA												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.98 AFD[i]_CH[z]_BUF_ACC Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	DATA	Read/write data from/to FIFO

38.11 FIFO to ARU Unit (F2A)

38.11.1 Overview

The F2A is the interface between the ARU and the FIFO sub-module. Since the data width of the ARU (ARU word) is 53 bit (two 24 bit values and five control bits) and the data width of the FIFO is only 29 bit, the F2A has to distribute the data from and to the FIFO channels in a configurable manner.

The data transfer between FIFO and ARU is organized with eight different streams that are connected to the eight different channels of the corresponding FIFO module. A stream represents a data flow from/to ARU to/from the FIFO via the F2A.

The general definition of 'channels' and 'streams' in the ARU context is done in **Section 38.5.3**.

Each FIFO channel can act as a write stream (data flow from FIFO to ARU) or as a read stream (data flow from ARU to FIFO).

Within these streams the F2A can transmit/receive the lower, the upper or both 24 bit values of the ARU together with the ARU control bits according to the configured transfer modes as described in **Section 38.11.2**.

Each stream can be enabled/disabled separately within the register F2A[i]_ENABLE. If a stream will be disabled, the stream data which are stored inside the F2A will be deleted. This is necessary to ensure, that no old data are transferred after enabling a stream.

38.11.2 Transfer modes

The F2A unit provides several transfer modes to map 29 bit data of the FIFO from/to 53 bit data of the ARU. E.g. it is configurable that the 24 bit FIFO data is written to the lower ARU data entry (means bits 0 to 23) or to the higher 24 bit ARU data entry (means bits 24 to 47). Bits 24 to 28 of the FIFO data entry (the five control bits) are written/read in both cases to/from bits 48 to 52 of the ARU entry.

When both values of the ARU have to be stored in the FIFO the values are stored behind each other inside the FIFO if the FIFO is not full.

If there is only space for one 24 bit data word plus the five control bits, the F2A transfers one part of the 53 bits first and then waits for transferring the second part before new data is requested from the ARU.

When two values from the FIFO have to be written to one ARU location the words have to be located behind each other inside the FIFO.

The transfer to ARU is only established when both parts could be read out of the FIFO otherwise if only one 29 bit word was provided by the FIFO the F2A waits until the second part is available before the data is made available at the ARU.

Figure 38.27 shows the data ordering of the FIFO when both ARU values must be transferred between ARU and FIFO.

When reading from the ARU the F2A first writes the lower word to the FIFO.

In case of writing to the ARU the F2A reads the lower word first from the FIFO, thus the lower word must be written first to the FIFO through the AFD interface.

NOTE

The five control bits (bits 48 to 52 of the ARU data word) are duplicated as bit 24 to 28 of both FIFO words in case of reading from ARU.

In the case of writing to the ARU, bits 24 to 28 of the last written FIFO word (the higher ARU word) are copied to bits 48 to 52 of the corresponding ARU location.

The transfer modes can be configured with the TMODE bits of registers F2A[i]_CH[z]_STR_CFG (x: 0 to 7).

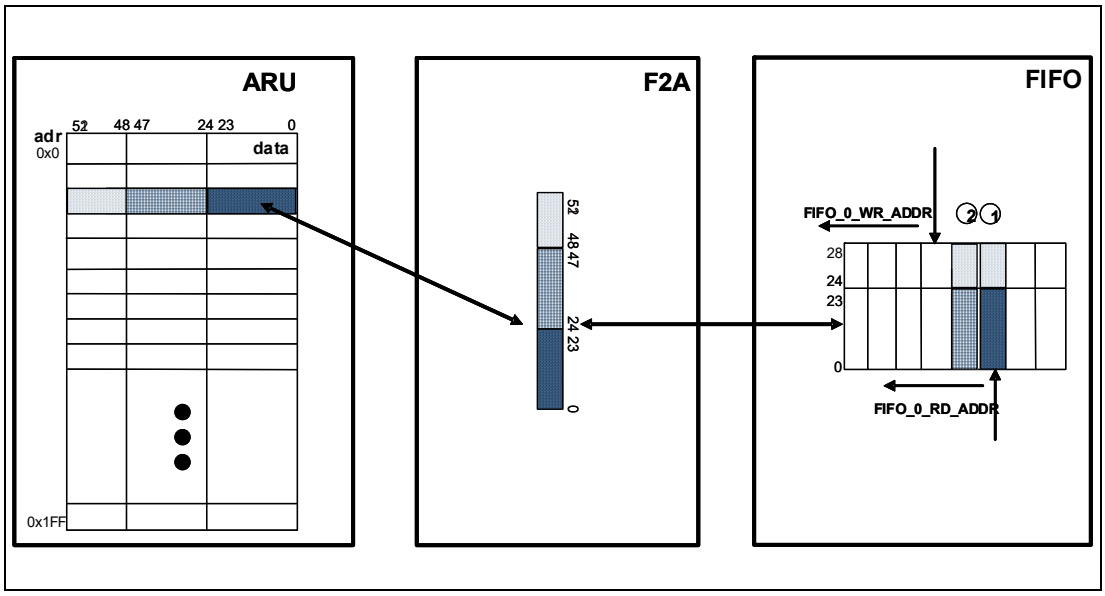


Figure 38.27 Data transfer of both ARU words between ARU and FIFO

38.11.3 Internal buffer mode

It is possible to use a FIFO channel as a buffer which is accessed only internally from ARU side. To do this, a read and a write stream of the F2A to one FIFO channel are needed. Therefore it is possible to reconfigure the upper 4.

F2A streams 4 to 7 to the lower 4 FIFO channels 0 to 3 in the following manner:

F2A stream 4 (+ F2A stream 0) → FIFO channel 0
F2A stream 5 (+ F2A stream 1) → FIFO channel 1
F2A stream 6 (+ F2A stream 2) → FIFO channel 2
F2A stream 7 (+ F2A stream 3) → FIFO channel 3

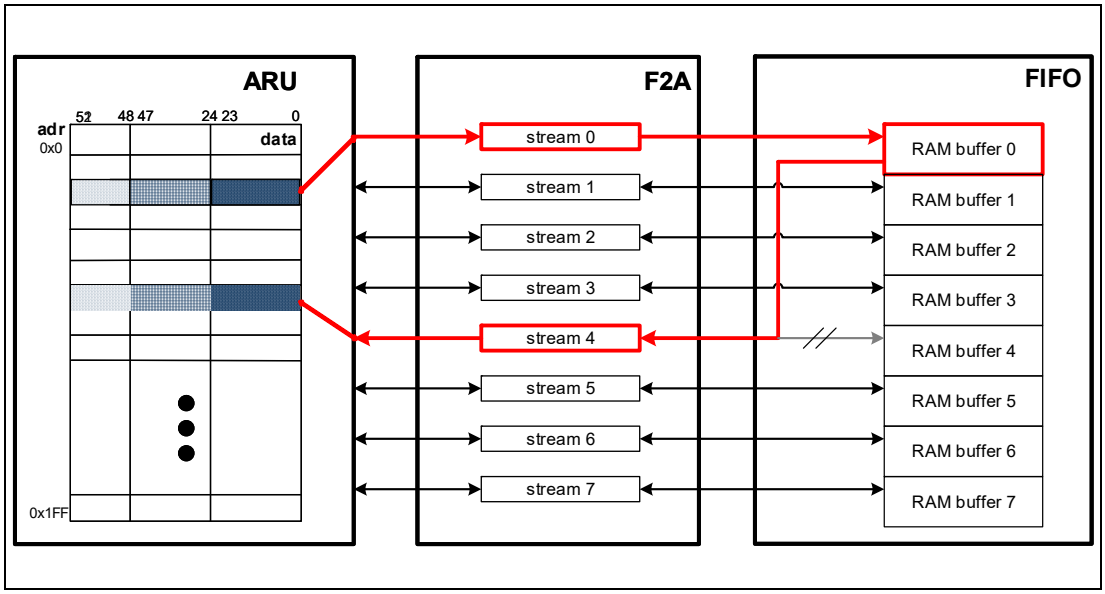


Figure 38.28 Reconfiguration of F2A stream 4 to FIFO channel 0

The configuration of each 4 upper F2A streams can be done separately for each stream in the configuration register F2A[i]_CTRL.

NOTE

The corresponding upper FIFO channel (4 to 7) cannot be used in this configuration.

38.11.4 F2A Configuration Registers Overview

F2A contains following configuration registers:

Table 38.99 Register List

Symbol	Register Name	Details in Section
F2A[i]_ENABLE	F2Ai stream activation register	**38.11.5.1**
F2A[i]_CH[z]_ARU_RD_FIFO	F2Ai ARU Read address register	**38.11.5.2**
F2A[i]_CH[z]_STR_CFG	F2Ai stream z configuration register	**38.11.5.3**
F2A[i]_CTRL	F2Ai stream control register	**38.11.5.4**

38.11.5 F2A Configuration Registers Description

38.11.5.1 F2A[i]_ENABLE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18040_H + (4000_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STR7_EN		STR6_EN		STR5_EN		STR4_EN		STR3_EN		STR2_EN		STR1_EN		STR0_EN	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.100 F2A[i]_ENABLE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	STR[7:0]_EN	Enable/disable stream k READ access: 00_B: Stream disabled 01_B: — 10_B: — 11_B: Stream enabled WRITE access: 01_B: Stream 0 is disabled and internal states are reset 10_B: Stream 0 is enabled Other than above: Setting prohibited. **NOTE** Stream data inside F2A will be deleted on stream disabling.

38.11.5.2 F2A[i]_CH[z]_ARU_RD_FIFO

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18000_H + (4000_H × i) + (4_H × z)

Value after reset: 0000 $01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADDR								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.101 F2A[i]_CH[z]_ARU_RD_FIFO Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ADDR	ARU Read address **NOTE** This bit field is only writable if channel is disabled.

38.11.5.3 F2A[i]_CH[z]_STR_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18020_H + (4000_H × i) + (4_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	DIR	TMODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.102 F2A[i]_CH[z]_STR_CFG Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	DIR	Data transfer direction 0: Transport from ARU to FIFO 1: Transport from FIFO to ARU
17, 16	TMODE	Transfer mode for 53 bit ARU data from/to FIFO. 00_B: Transfer low word (ARU bits 23:0) from/to FIFO 01_B: Transfer high word (ARU bits 47:24) from/to FIFO 10_B: Transfer both words from/to FIFO 11_B: Setting prohibited.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

NOTE

The write protected bits of register F2A[i]_STR_[z]_CFG are only writable if the corresponding enable bit STRz_EN of register F2A_ENABLE is cleared.

38.11.5.4 F2A[i]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 18044_H + (4000_H × i) + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	STR7_CONF		STR6_CONF		STR5_CONF		STR4_CONF	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.103 F2A[i]_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	STR[7:4]_CONF	Reconfiguration of stream k to FIFO channel k–4 READ access: 00_B: Stream k is mapped to FIFO buffer k 01_B: — 10_B: — 11_B: Stream k is mapped to FIFO buffer k–4 WRITE access: 01_B: Stream k is mapped to FIFO buffer k 10_B: Stream k is mapped to FIFO buffer k–4 Other than above: Setting prohibited. **NOTE** The write protected bits of register F2A[i]_CTRL are only writable if the corresponding enable bit STR[k–4]_EN and STR[k]_EN of register F2A_ENABLE is cleared.

38.12 Clock Management Unit (CMU)

38.12.1 Overview

The Clock Management Unit (CMU) is responsible for clock generation of the counters and of the GTM-IP. The CMU consists of three sub-units that generate different clock sources for the whole GTM-IP. The primary clock source for this sub-module is the cluster 0 clock signal cls0_clk which is defined by the value of bit field CLS0_CLK_DIV in register GTM_CLS_CLK_CFG. **Figure 38.29** shows a block diagram of the CMU.

The Configurable Clock Generation (CFGU) sub-unit provides eight dedicated clock sources for the following GTM modules: TIM, ATOM, TBU, and MON. Each instance of such a module can choose an arbitrary clock source, in order to specify wide-ranging time bases.

The Fixed Clock Generation (FXU) sub-unit generates predefined non-configurable clocks CMU_FXCLK[y] (y: 0 to 4) for the TOM modules and the MON module. The CMU_FXCLK[y] signals are derived from the CMU_GCLK_EN signal generated by the Global Clock Divider. The dividing factors are defined as 2^0, 2^4, 2^8, 2^{12}, and 2^{16}.

The External Clock Generation (EGU) sub-unit is able to generate up to three chip external clock signals visible at CMU_ECLK[z] (z: 0 to 2) with a duty cycle of about 50%.

The External Clock Generation (EGU) sub-unit is able to generate clock CMU_CLK8 for module CCM to manage 2 clock domains.

The clock source signals CMU_CLK[x] (x: 0 to 7) and CMU_FXCLK[y] are implemented in form of enable signals for the corresponding registers, which means that the actual clock signal of all registers always use the CLS0_CLK signal.

The four configurable clock signals CMU_CLK0, CMU_CLK1, CMU_CLK6 and CMU_CLK7 are used for the TIM filter counters.

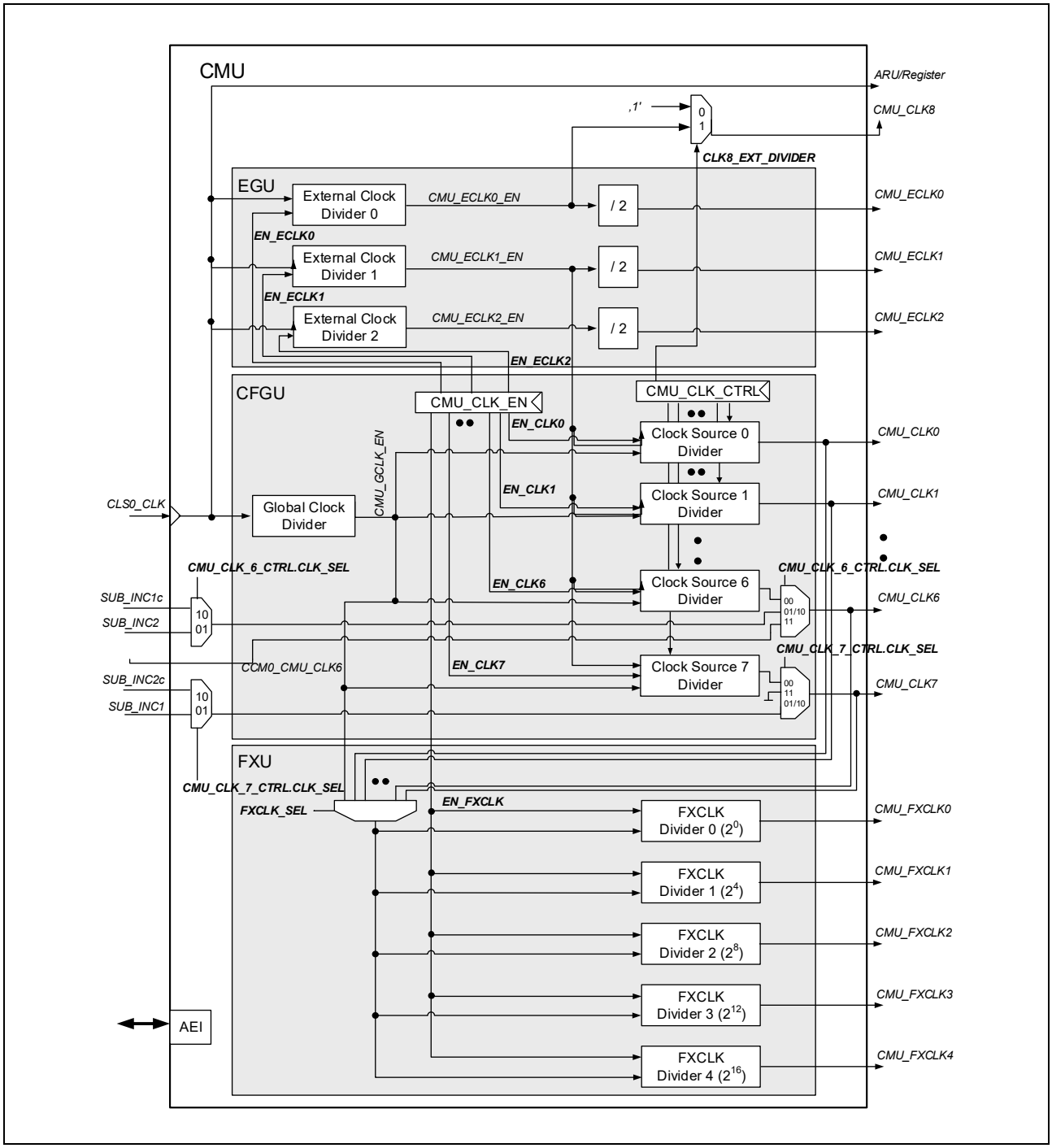


Figure 38.29 CMU Block Diagram

38.12.2 Global Clock Divider

The sub block Global Clock Divider can be used to divide the CMU primary source signal CLS0_CLK into a common subdivided clock signal.

The divided clock signal of the sub block Global Clock Divider is implemented as an enable signal that enables dedicated clocks from the CLS0_CLK signal to generate the user specified divided clock frequency.

The resulting fractional divider (Z/N) specified through equation:

$T_{CMU\_GCLK\_EN} = (Z/N) * T_{CLS0\_CLK}$ is implemented according the following algorithm

(Z: CMU_GCLK_NUM(23:0) ; N: CMU_GCLK_DEN(23:0) ; $Z \geq N > 0$):

(1) Set remainder (R), operand 1 (OP1) and operand 2 (OP2) register during INIT-phase (with implicit conversion to signed):
R = Z, OP1 = N, OP2 = N-Z;

(2) After leaving INIT-phase (at least one CMU_CLK[x] has been enabled) the sign of remainder R for each CLS0_CLK cycle will be checked:

(3) If $R \geq 0$ keep updating remainder and keep CMU_GCLK_EN = 0:
R = R-OP1;

(4) If R<0 update remainder and set CMU_GCLK_EN = 1:
R = R-OP2;

After at most (Z/N +1) subtractions (3) there will be a negative R and an active phase of the generated clock enable (for one cycle) will be triggered (4). The remainder R is a measure for the distance to a real Z/N clock and will be regarded for the next generated clock enable cycle phase. The new R value will be R = R + (Z-N). In the worst case the remainder R will sum up to an additional cycle in the generated clock enable period after Z-cycles. In the other cases equally distributed additional cycles will be inserted for the generated clock enable. If Z is an integer multiple of N no additional cycles will be included for the generated clock enable at all.

NOTE

For a better resource sharing all arithmetic has been reduced to subtractions and the initialization of the remainder R uses the complement of (Z-N).

38.12.3 Configurable Clock Generation Sub-unit (CFGU)

The CMU sub-unit CFGU provides up to eight configurable clock divider blocks that divide the common C MU_GCLK_EN signal into dedicated enable signals for the GTM-IP sub blocks.

The configuration of the eight different clock signals CMU_CLK[x] (x: 0 to 7) always depends on the configuration of the global clock enable signal CMU_GCLK_EN. Additionally, each clock source has its own configuration data, provided by the control register CMU_CLK_[x]_CTRL (x: 0 to 7).

According to the configuration of the Global Clock Divider, the configuration of the Clock Source x Divider is done by setting an appropriate value in the bit field CLK_CNT[x] of the register CMU_CLK_[x]_CTRL.

The frequency fx = 1/Tx of the corresponding clock enable signal CMU_CLK[x] can be determined by the unsigned representation of CLK_CNT[x] of the register CMU_CLK_[x]_CTRL in the following way:

$T_{CMU\_CLK[x]} = (CLK\_CNT[x] + 1) \times T_{CMU\_GCLK\_EN}$.

The corresponding wave form is shown in **Figure 38.30**.

Each clock signal CMU_CLK[x] can be enabled individually by setting the appropriate bit field EN_CLK[x] in the register CMU_CLK_EN.

However, individual enabling of CMU_CLK6 and CMU_CLK7 is only possible if the bit field CLK_SEL of register CMU_CLK_6_CTRL or CMU_CLK_7_CTRL is set to 0.

Alternatively, clock source six and seven (CMU_CLK6 and CMU_CLK7) may provide the signal SUB_INC1 and SUB_INC2 coming from module DPLL as clock enable signal depending on the bit field CLK_SEL of the register CMU_CLK_6_CTRL and CMU_CLK_7_CTRL.

CMU_CLK8 is switched by CLK8_EXT_DIVIDER of the register CMU_CLK_CTRL between CLS0_CLK and CMU_ECLK0.

To switch the clock reference CMU_GCLK_EN with CMU_ECLK1_EN an input selector are used in all Clock Source Divider. The CMU_ECLK1_EN source is enabled by setting the appropriate bit field CMU[x]_EXT_DIVIDER in the register CMU_CLK_CTRL.

To avoid unexpected behavior of the hardware, the configuration of register CMU_CLK_[x]_CTRL and CMU_CLK_CTRL can only be changed, when the corresponding clock signal CMU_CLK[x] and CMU_ECLK[1] is disabled.

Further, any changes to the registers CMU_GCLK_NUM and CMU_GCLK_DEN can only be performed, when all clock enable signals CMU_CLK[x] and the EN_FXCLK bit inside the CMU_CLK_EN register are disabled.

The clock source signals CMU_CLK[x] (x: 0 to 7) and CMU_FXCLK[y] are implemented in form of enable signals for the corresponding registers, which means that the actual clock signal of all registers always use the CLS0_CLK signal.

The hardware guarantees that all clock signals CMU_CLK[x] (x: 0 to 7), which were enabled simultaneous, are synchronized to each other.

Simultaneous enabling does mean that the bits EN_CLK[x] in the register CMU_CLK_EN are set by the same write access.

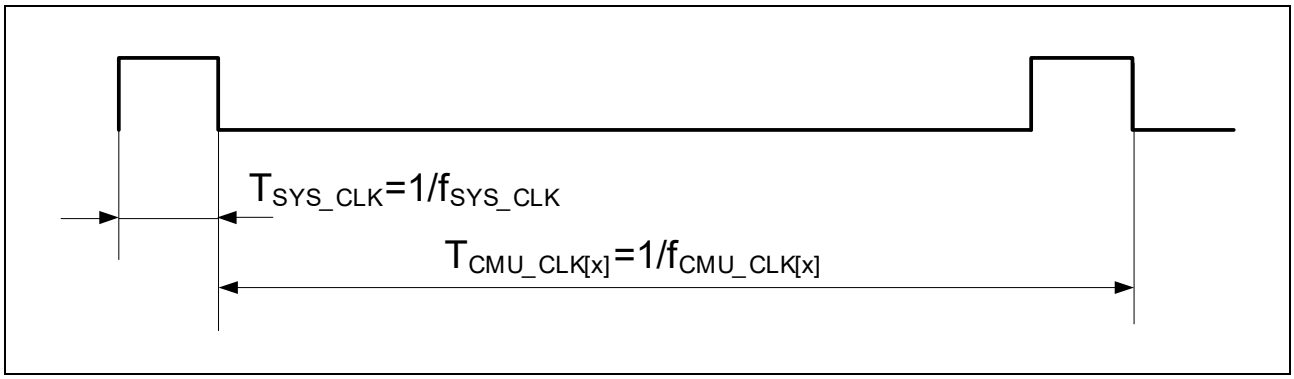


Figure 38.30 Wave Form of Generated Clock Signal CMU_CLK[x]

38.12.4 Fixed Clock Generation (FXU)

The FXU sub-unit generates fixed clock enables out of the CMU_GCLK_EN or one of the eight CMU_CLK[x] enable signal depending on the FXCLK_SEL bit field of the CMU_FXCLK_CTRL register. These clock enables are used for the PWM generation inside the TOM modules.

All clock enables CMU_FXCLK[y] can be enabled or disabled simultaneous by setting the appropriate bit field EN_FXCLK in the register CMU_CLK_EN.

The dividing factors are defined as 2^0, 2^4, 2^8, 2^{12}, and 2^{16}. The signals CMU_FXCLK[y] are implemented in form of enable signals for the corresponding registers (see also **Figure 38.30**).

38.12.5 External Generation Unit (EGU)

The EGU sub-unit generate up to three separate clock output signals CMU_ECLK[z] (z: 0 to 2). Each of these clock signals is derived from the corresponding External Clock Divider z sub block, which generates a clock signal derived from the GTM-IP input clock CLS0_CLK.
In contrast to the signals CMU_CLK[x] and CMU_FXCLK[y], which are treated as simple enable signals for the registers, the signals CMU_ECLK[z] have a duty cycle of about 50% that is used as a true clock signal for external peripheral components.
To manage a second global frequency CMU_GCLK_EN could be replaced by ECLK[1]_EN for CMU_CLK[x](x: 0 to 7). Also the all-time enabled CMU_CLK8 could be replaced by ECLK[0]_EN.

Each of the external clocks divider are enabled and disabled by setting the appropriate bit field EN_ECLK[z] in the register CMU_CLK_EN.
The clock frequencies $f_{CMU\_ECLK[z]} = 1/T_{CMU\_ECLK[z]}$ of the external clocks are controlled with the registers CMU_ECLK_[z]_NUM and CMU_ECLK_[z]_DEN as follows:

$T_{CMU\_ECLK[z]\_EN} = (ECLK[z]\_NUM / ECLK[z]\_DEN) \times T_{CLS0\_CLK}$

and is implemented according the following algorithm (Z: CMU_ECLK_[z]_NUM(23:0); N: CMU_ECLK_[z]_DEN(23:0); $Z \geq N > 0$; $Z \geq N$; CMU_ECLK[z] = '0'):

(1) Set remainder (R), operand 1 (OP1) and operand 2 (OP2) register during INIT-phase (with implicit conversion to signed): R = Z, OP1 = N, OP2 = N-Z;

(2) After leaving INIT-phase (CMU_ECLK[z] has been enabled) the sign of remainder R for each CLS0_CLK cycle will be checked:

(3) If $R \geq 0$ keep updating remainder and keep CMU_ECLK[z]: R = R-OP1;

(4) If $R < 0$ update remainder and toggle CMU_ECLK[z]: R = R-OP2;

After at most (Z/N +1) subtractions (3) there will be a negative R and an active phase of the generated clock enable (for one cycle) will be triggered (4). The remainder R is a measure for the distance to a real Z/N clock and will be regarded for the next generated clock toggle phase. The new R value will be R = R+(Z-N). In the worst case the remainder R will sum up to an additional cycle in the generated clock toggle period after Z-cycles. In the other cases equally distributed additional cycles will be inserted for the generated clock toggle. If Z is an integer multiple of N no additional cycles will be included for the generated clock toggle at all.

NOTE

For a better resource sharing all arithmetic has been reduced to subtractions and the initialization of the remainder R uses the complement of (Z–N).

The default value of the CMU_ECLK[z] output is low.

38.12.6 CMU Configuration Registers Overview

CMU contains following configuration registers:

Table 38.104 Register List

Symbol	Register Name	Details in Section
CMU_CLK_EN	CMU clock enable	**38.12.7.1**
CMU_GCLK_NUM	CMU global clock control numerator	**38.12.7.2**
CMU_GCLK_DEN	CMU global clock control denominator	**38.12.7.3**
CMU_CLK_[z]_CTRL	CMU control for clock source z	**38.12.7.4**
CMU_CLK_6_CTRL	CMU control for clock source 6	**38.12.7.5**
CMU_CLK_7_CTRL	CMU control for clock source 7	**38.12.7.6**
CMU_ECLK_[z]_NUM	CMU external clock z control numerator	**38.12.7.7**
CMU_ECLK_[z]_DEN	CMU external clock z control denominator	**38.12.7.8**
CMU_FXCLK_CTRL	CMU control FXCLK sub-unit input clock	**38.12.7.9**
CMU_GLB_CTRL	CMU synchronizing ARU and clock source	**38.12.7.10**
CMU_CLK_CTRL	CMU control for clock source selection	**38.12.7.11**

38.12.7 CMU Configuration Registers Description

38.12.7.1 CMU_CLK_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00300_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EN_FXCLK		EN_ECLK2		EN_ECLK1		EN_ECLK0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EN_CLK7		EN_CLK6		EN_CLK5		EN_CLK4		EN_CLK3		EN_CLK2		EN_CLK1		EN_CLK0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.105 CMU_CLK_EN Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23, 22	EN_FXCLK	Enable all CMU_FXCLK, see bits 1, 0 **NOTE** An enable to EN_FXCLK from disable state will be reset internal fixed clock counters.
21, 20	EN_ECLK2	Enable ECLK 2 generation sub-unit, see bits 1, 0
19, 18	EN_ECLK1	Enable ECLK 1 generation sub-unit, see bits 1, 0
17, 16	EN_ECLK0	Enable ECLK 0 generation sub-unit, see bits 1, 0
15, 14	EN_CLK7	Enable clock source 7, see bits 1, 0
13, 12	EN_CLK6	Enable clock source 6, see bits 1, 0
11,10	EN_CLK5	Enable clock source 5, see bits 1, 0
9, 8	EN_CLK4	Enable clock source 4, see bits 1, 0
7, 6	EN_CLK3	Enable clock source 3, see bits 1, 0
5, 4	EN_CLK2	Enable clock source 2, see bits 1, 0
3, 2	EN_CLK1	Enable clock source 1, see bits 1, 0
1, 0	EN_CLK0	Enable clock source 0 00_B: Clock source is disabled (ignore write access) 01_B: Disable clock signal and reset internal states 10_B: Enable clock signal 11_B: Clock signal enabled (ignore write access) **NOTES** 1. Any read access to an EN_CLK[z], EN_ECLK[z] or EN_FXCLK bit field will always result in a value 00_B or 11_B indicating current state. A modification of the state is only performed with the values 01_B and 10_B. Writing the values 00_B and 11_B is always ignored. 2. Any disabling to EN_CLK[x] will be reset internal counters for configurable clocks.

38.12.7.2 CMU_GCLK_NUM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00304_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMU_GCLK_NUM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMU_GCLK_NUM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.106 CMU_GCLK_NUM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CMU_GCLK_NUM	Numerator for global clock divider. Defines numerator of the fractional divider. **NOTES** 1. Value can only be modified when all clock enables EN_CLK[x] and the EN_FXCLK are disabled. 2. The CMU hardware alters the content of CMU_GCLK_NUM and CMU_GCLK_DEN automatically to 1_H, if CMU_GCLK_NUM is specified less than CMU_GCLK_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_GCLK_NUM followed by a single write to register CMU_GCLK_DEN.

38.12.7.3 CMU_GCLK_DEN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00308_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GCLK_DEN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GCLK_DEN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.107 CMU_GCLK_DEN Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	GCLK_DEN	Denominator for global clock divider. Defines denominator of the fractional divider. **NOTES** 1. Value can only be modified when all clock enables EN_CLK[x] and the EN_FXCLK are disabled. 2. The CMU hardware alters the content of CMU_GCLK_NUM and CMU_GCLK_DEN automatically to 1_H, if CMU_GCLK_NUM is specified less than CMU_GCLK_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_GCLK_NUM followed by a single write to register CMU_GCLK_DEN.

38.12.7.4 CMU_CLK_[z]_CTRL

Access: This register can be read or written in 32-bit units.

Address: CMU_CLK_0_CTRL: <GTM_base> + $0030C_H$
CMU_CLK_1_CTRL: <GTM_base> + 00310_H
CMU_CLK_2_CTRL: <GTM_base> + 00314_H
CMU_CLK_3_CTRL: <GTM_base> + 00318_H
CMU_CLK_4_CTRL: <GTM_base> + $0031C_H$
CMU_CLK_5_CTRL: <GTM_base> + 00320_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CLK_CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK_CNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.108 CMU_CLK_[z]_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CLK_CNT	Clock count. Defines count value for the clock divider. **NOTE** Value can only be modified when clock enable EN_CLK[z] and EN_ECLK1 are disabled.

38.12.7.5 CMU_CLK_6_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00324_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CLK_SEL		CLK_CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK_CNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.109 CMU_CLK_6_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	CLK_SEL	Clock source selection. 00_B: Use Clock Source 6 Divider 01_B: Use signal SUB_INC2 of module DPLL 10_B: Use signal SUB_INC1c of module DPLL 11_B: Use signal CCM0_CMU_CLK6 of sub-module CCM0 **NOTE** Value can only be modified when clock enable EN_CLK6 and EN_ECLK1 are disabled.
23 to 0	CLK_CNT	Clock count. Defines count value for the clock divider of clock source CMU_CLK6. **NOTE** Value can only be modified when clock enable EN_CLK6 and EN_ECLK1 are disabled.

38.12.7.6 CMU_CLK_7_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00328_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CLK_SEL		CLK_CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLK_CNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.110 CMU_CLK_7_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	CLK_SEL	Clock source selection. 00_B: Use Clock Source 7 Divider 01_B: Use signal SUB_INC1 of module DPLL 10_B: Use signal SUB_INC2c of module DPLL 11_B: Reserved, no clock is selected **NOTE** Value can only be modified when clock enable EN_CLK7 and EN_ECLK1 are disabled.
23 to 0	CLK_CNT	Clock count. Defines count value for the clock divider of clock source CMU_CLK7. **NOTE** Value can only be modified when clock enable EN_CLK7 and EN_ECLK1 are disabled.

38.12.7.7 CMU_ECLK_[z]_NUM

Access: This register can be read or written in 32-bit units.

Address: CMU_ECLK_0_NUM: <GTM_base> + $0032C_H$
CMU_ECLK_1_NUM: <GTM_base> + 00334_H
CMU_ECLK_2_NUM: <GTM_base> + $0033C_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ECLK_NUM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECLK_NUM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.111 CMU_ECLK_[z]_NUM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ECLK_NUM	Numerator for external clock divider. Defines numerator of the fractional divider. **NOTES** 1. Value can only be modified when clock enable EN_ECLK[z] is disabled. 2. The CMU hardware alters the content of CMU_ECLK_[z]_NUM and CMU_ECLK_[z]_DEN automatically to 1_H, if CMU_ECLK_[z]_NUM is specified less than CMU_ECLK_[z]_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_ECLK_[z]_NUM followed by a single write to register CMU_ECLK_[z]_DEN.

38.12.7.8 CMU_ECLK_[z]_DEN

Access: This register can be read or written in 32-bit units.

Address: CMU_ECLK_0_DEN: <GTM_base> + 00330_H
CMU_ECLK_1_DEN: <GTM_base> + 00338_H
CMU_ECLK_2_DEN: <GTM_base> + 00340_H

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ECLK_DEN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECLK_DEN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.112 CMU_ECLK_[z]_DEN Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ECLK_DEN	Denominator for external clock divider. Defines denominator of the fractional divider. **NOTES** 1. Value can only be modified when clock enable EN_ECLK[z] is disabled. 2. The CMU hardware alters the content of CMU_ECLK_[z]_NUM and CMU_ECLK_[z]_DEN automatically to 1_H, if CMU_ECLK_[z]_NUM is specified less than CMU_ECLK_[z]_DEN or one of the values is specified with a value zero. Thus, a secure way for altering the values is writing twice to the register CMU_ECLK_[z]_NUM followed by a single write to register CMU_ECLK_[z]_DEN.

38.12.7.9 CMU_FXCLK_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00344_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FXCLK_SEL			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.113 CMU_FXCLK_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	FXCLK_SEL	Input clock selection for EN_FXCLK line. 0000_B: CMU_GCLK_EN selected. 0001_B: CMU_CLK0 selected. 0010_B: CMU_CLK1 selected. 0011_B: CMU_CLK2 selected. 0100_B: CMU_CLK3 selected. 0101_B: CMU_CLK4 selected. 0110_B: CMU_CLK5 selected. 0111_B: CMU_CLK6 selected. 1000_B: CMU_CLK7 selected. **NOTES** 1. This value can only be written, when the CMU_FXCLK generation is disabled. See bits 23 and 22 in register CMU_CLK_EN. 2. Other values for FXCLK_SEL are reserved and should not be used, but they behave like FXCLK_SEL = 0.

38.12.7.10 CMU_GLB_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00348_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARU_ADDR_RSTGLB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.114 CMU_GLB_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARU_ADDR_RSTGLB	Reset ARU caddr counter and ARU dynamic route counter. **NOTES** 1. Writing value 1 to this bit field results in a request to reset the ARU caddr counter and ARU dynamic route counter. The next following write access to register CMU_CLK_EN applies the ARU caddr counter reset, ARU dynamic route counter reset and resets this bit. This feature can be used to synchronize the ARU round trip time to the CMU clocks. 2. This bit is write protected. Before writing to this bit set bit RF_PROT of register GTM_CTRL to 0.

38.12.7.11 CMU_CLK_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0034C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLK8_EXT_DIVIDER	CLK7_EXT_DIVIDER	CLK6_EXT_DIVIDER	CLK5_EXT_DIVIDER	CLK4_EXT_DIVIDER	CLK3_EXT_DIVIDER	CLK2_EXT_DIVIDER	CLK1_EXT_DIVIDER	CLK0_EXT_DIVIDER
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.115 CMU_CLK_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CLK8_EXT_DIVIDER	Clock source selection for CMU_CLK8. 0: Use Clock Source CLS0_CLK 1: Use Clock Source CMU_ECLK0 **NOTE** Value can only be modified when EN_ECLK0 is disabled.
7 to 0	CLK[7:0]_EXT_DIVIDER	Clock source selection for CMU_CLK_[k]_CTRL. 0: Use Clock Source CMU_GCLK_EN 1: Use Clock Source CMU_ECLK1 **NOTE** Value can only be modified when clock enable EN_CLK[k] and EN_ECLK1 are disabled.

38.13 Cluster Configuration Module (CCM)

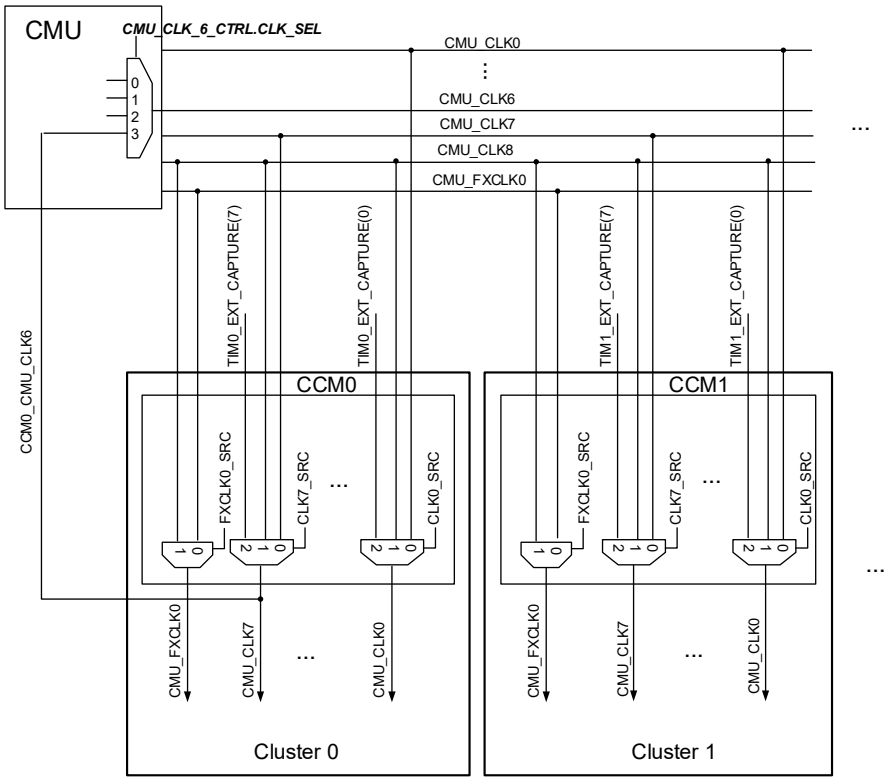
38.13.1 Overview

As already mentioned in **Section 38.5**, each sub-module of the GTM is aligned explicitly to a cluster. The Cluster Configuration Module (CCM) enables the configuration of several cluster specific options namely.

1. Cluster's clock frequency
2. Module clock gating
3. Status observation of the cluster's MCS bus master (AEM)
4. Address range protection
5. Global architecture configuration

The register CCM[i]_CFG allows disabling the system clock signal for unused sub modules of the i-th cluster. The registers CCM[i]_CMU_CLK_CFG and CCM[i]_CMU_FXCLK_CFG allows the configuration of various cluster clock frequencies.

Figure 38.31 shows important details about the wiring of the cluster's local clock signals.



The register CCM[i]_AEIM_STA captures the address and the reason of the first invalid AEIM bus master access of the cluster's MCS module.

Figure 38.31 Cluster Clock Signal Wiring

38.13.2 Address Range Protection

The CCM also provides up to NARP so called address range protectors (ARPs), where the number NARP depends on the actual device configuration (defined in **Section 38.28, GTM Device 358**). An ARP can be used to define a configurable write protected address range in order to support enhanced safety features. The address width of and ARP is also device dependent and it is determined by the parameter AAW as defined in **Section 38.28, GTM Device 358**.

The protected address range is mapped to the address range of the cluster's MCS RAM port.

Each ARP z (with z = 0 to NARP-1) can be configured by the registers CCM[i]_ARP[z]_CTRL and CCM[i]_ARP[z]_PROT, where the register CCM[i]_ARP[z]_CTRL enables to configure the size and base address offset for an ARP and the register CCM[i]_ARP[z]_PROT configures, that an MCS channel x with a set bit field WPROTx cannot write to the corresponding z-th ARP. Whenever an MCS channel x is writing to an ARP that does not allow a write access from channel x by the configuration register CCM[i]_ARP[z]_PROT, the write access is discard. The bit field WPROT_AEI of register CCM[i]_ARP[z]_CTRL allows to configure if a CPU write access (via AEI slave interface) to the z-th ARP is protected. If the CPU wants to write to the z-th ARP while WPROT_AEI is set, the write access will be discard and the AEI status signal will signalize an invalid module access.

NOTE

Considering the size and base address of an ARP, it should be noted that the configuration possibilities are limited. Details about the configuration can be found in the register description of MCS[i]_ARP[z]_CTRL.

The bit field DIS_PROT of register CCM[i]_ARP[z]_CTRL changes the meaning of an ARP configuration in a way that it explicitly allows an MCS channel x with a set bit field WPROTx to write to the z–th ARP. Accordingly, if the bit DIS_PROT is set while the bit WPROT_AEI is also set in the register CCM[i]_ARP[z]_CTRL, the z-th ARP explicitly allows a write access from the CPU to the z-th ARP. A meaningful application of an ARP z with a set bit field DIS_PROT for an MCS channel x has another ARP with a surrounding wider address range that is defining a write protection for MCS channel x and some other MCS channels. Since the address range of an ARP can surround another ARP it is possible to configure contradictory conditions for MCS channels or the CPU within the overlapping area (e.g. if ARP y surrounds ARP z and ARP y allows a write access for an MCS channel x but ARP z prohibits a write access for MCS channel x). In order to resolve this ambiguity, the following rule is defined: A write protection for a specific address c concerning MCS channel x (the CPU) is active, if and only if, address c is covered by at least one ARP with a cleared bit DIS_PROT and a set bit WPROTx (WPROT_AEI) and there exists no ARP covering address c with a set bit field DIS_PROT and a set bit field WPROTx (WPROT_AEI).

38.13.3 CCM Configuration Registers Overview

CCM contains following configuration registers:

Table 38.116 Register List

Symbol	Register Name	Details in Section
CCM[i]_PROT	CCMi Protection Register	**38.13.4.1**
CCM[i]_CFG	CCMi Configuration Register	**38.13.4.2**
CCM[i]_CMU_CLK_CFG	CCMi CMU Clock Configuration Register	**38.13.4.3**
CCM[i]_CMU_FXCLK_CFG	CCMi CMU Fixed Clock Configuration Register	**38.13.4.4**
CCM[i]_AEIM_STA	CCMi MCS Bus Master Status Register	**38.13.4.5**
CCM[i]_ARP[z]_CTRL	CCMi Address Range Protector z Control Register	**38.13.4.6**
CCM[i]_ARP[z]_PROT	CCMi Address Range Protector z Protection Register	**38.13.4.7**
CCM[i]_HW_CONF	CCMi Hardware Configuration Register	**38.13.4.8**
CCM[i]_TIM_AUX_IN_SRC	CCMi TIM AUX Input Source Register	**38.13.4.9**
CCM[i]_EXT_CAP_EN	CCMi External Capture Enable Register	**38.13.4.10**
CCM[i]_TOM_OUT	CCMi TOM Output Register	**38.13.4.11**
CCM[i]_ATOM_OUT	CCMi ATOM Output Register	**38.13.4.12**
CCM[i]_HW_CONF2	CCMi 2 Hardware Configuration Register	**38.13.4.13**

38.13.4 CCM Configuration Register Description

38.13.4.1 CCM[i]_PROT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21FC_H$ + (200_H × i)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLS_PROT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.117 CCM[i]_PROT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CLS_PROT	Cluster Protection 0: Write Protection of cluster configuration registers disabled. 1: Write Protection of cluster configuration registers enabled.

38.13.4.2 CCM[i]_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21F8_H$ + (200_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBU_DIR2	TBU_DIR1	—	—	—	—	—	—	—	—	—	—	—	—	CLS_CLK_DIV	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	EN_CMP_MON	EN_PSM	EN_BRC	EN_DPLL_MAP	EN_MCS	EN_ATOM_ADTM	EN_TOM_SPE_TDTM	EN_TIM
Value after reset	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.118 CCM[i]_CFG Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	TBU_DIR[k]	DIR1 input signal of module TBU. 0: Indicating forward direction 1: Indicating backward direction
29 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	CLS_CLK_DIV	Cluster Clock Divider. 00_B: Cluster is disabled 01_B: Cluster is enabled without clock divider 10_B: Cluster is enabled with clock divider 2 11_B: Setting prohibited. **NOTE** The value of this bit field mirrors the bit field CLS[i]_CLK_DIV of register GTM_CLS_CLK_CFG, whereas i equals the cluster index.
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	EN_CMP_MON	Enable CMP and MON 0: Disable clock signal for modules CMP and MON. 1: Enable clock signal for modules CMP and MON. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
6	EN_PSM	Enable PSM 0: Disable clock signal for module PSM. 1: Enable clock signal for module PSM. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

Table 38.118 CCM[i]_CFG Register Contents (2/2)

Bit Position	Bit Name	Function
5	EN_BRC	EN_BRC: Enable BRC 0: Disable clock signal for module BRC. 1: Enable clock signal for module BRC. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
4	EN_DPLL_MAP	EN_DPLL_MAP: Enable DPLL and MAP 0_B: Disable clock signal for modules DPLL and MAP. 1_B: Enable clock signal for modules DPLL and MAP. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
3	EN_MCS	EN_MCS Enable MCS 0_B: Disable clock signal for module MCS. 1_B: Enable clock signal for module MCS. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
2	EN_ATOM_ADTM	EN_ATOM_ADTM: Enable ATOM and ADTM 0_B: Disable clock signal for modules ATOM and its related DTM modules. 1_B: Enable clock signal for modules ATOM and its related DTM modules. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
1	EN_TOM_SPE_TDTM	EN_TOM_SPE_TDTM: Enable TOM, SPE and TDTM 0_B: Disable clock signal for modules TOM, SPE, and its related DTM modules. 1_B: Enable clock signal for modules TOM, SPE, and its related DTM modules. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
0	EN_TIM	Enable TIM 0_B: Disable clock signal for sub module TIM. 1_B: Enable clock signal for sub module TIM. **NOTE** This bit is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

Note: The module specific clock enable registers (bit field EN_*) are only implemented if the corresponding module is available in the i-th cluster.

38.13.4.3 CCM[i]_CMU_CLK_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21F0_H$ + (200_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CLK7_SRC		—	—	CLK6_SRC		—	—	CLK5_SRC		—	—	CLK4_SRC	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CLK3_SRC		—	—	CLK2_SRC		—	—	CLK1_SRC		—	—	CLK0_SRC	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 38.119 CCM[i]_CMU_CLK_CFG Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	CLK7_SRC	Clock 7 source signal selector 00_B: Use CMU_CLK7 signal of CMU as CCM[i]_CLK7_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK7_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE7 signal as CCM[i]_CLK7_EN signal within Cluster. 11_B: Setting prohibited.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	CLK6_SRC	Clock 6 source signal selector 00_B: Use CMU_CLK6 signal of CMU as CCM[i]_CLK6_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK6_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE6 signal as CCM[i]_CLK6_EN signal within Cluster. 11_B: Setting prohibited.
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	CLK5_SRC	Clock 5 source signal selector 00_B: Use CMU_CLK5 signal of CMU as CCM[i]_CLK5_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK5_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE5 signal as CCM[i]_CLK5_EN signal within Cluster. 11_B: Setting prohibited.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.119 CCM[i]_CMU_CLK_CFG Register Contents (2/2)

Bit Position	Bit Name	Function
17, 16	CLK4_SRC	Clock 4 source signal selector 00_B: Use CMU_CLK4 signal of CMU as CCM[i]_CLK4_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK4_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE4 signal as CCM[i]_CLK4_EN signal within Cluster. 11_B: Setting prohibited.
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	CLK3_SRC	Clock 3 source signal selector 00_B: Use CMU_CLK3 signal of CMU as CCM[i]_CLK3_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK3_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE3 signal as CCM[i]_CLK3_EN signal within Cluster. 11_B: Setting prohibited.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	CLK2_SRC	Clock 2 source signal selector 00_B: Use CMU_CLK2 signal of CMU as CCM[i]_CLK2_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK2_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE2 signal as CCM[i]_CLK2_EN signal within Cluster. 11_B: Setting prohibited.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	CLK1_SRC	Clock 1 source signal selector 00_B: Use CMU_CLK1 signal of CMU as CCM[i]_CLK1_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK1_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE1 signal as CCM[i]_CLK1_EN signal within Cluster. 11_B: Setting prohibited.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CLK0_SRC	Clock 0 source signal selector 00_B: Use CMU_CLK0 signal of CMU as CCM[i]_CLK0_EN signal within Cluster. 01_B: Use CMU_CLK8 signal of CMU as CCM[i]_CLK0_EN signal within Cluster. 10_B: Use TIM[i]_EXT_CAPTURE0 signal as CCM[i]_CLK0_EN signal within Cluster. 11_B: Setting prohibited.

Note: The bit fields of this register are only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

38.13.4.4 CCM[i]_CMU_FXCLK_CFG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21F4_H$ + (200_H × i)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	FXCLK0_SRC			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.120 CCM[i]_CMU_FXCLK_CFG Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	FXCLK0_SRC	Fixed clock 0 source signal selector 0000_B: Use CMU_FXCLK0_EN signal of CMU as CCM[i]_FXCLK0_EN signal within Cluster. 0001_B: Use CMU_CLK8_EN signal of CMU as CCM[i]_FXCLK0_EN signal within Cluster. Other than above: Setting prohibited. **NOTE** These bits are only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

38.13.4.5 CCM[i]_AEIM_STA

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21D8_H$ + (200_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	AEIM_XPT_STA		—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEIM_XPT_ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.121 CCM[i]_AEIM_STA Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	AEIM_XPT_STA	AEIM_XPT_STA: AEIM Exception status. 00_B: No invalid MCS bus master access occurred 01_B: Invalid byte addressing of MCS bus master access. 10_B: Illegal module access of MCS bus master access. 11_B: Invalid MCS bus master access to an unsupported address.
23 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	AEIM_XPT_ADDR	AEIM_XPT_ADDR: Exception Address. Invalid bus master (AEIM) address of MCS module.

Notes:

1. Only the first invalid AEIM bus master access of the MCS is updating this register with the invalid AEIM address (bit field AEIM_XPT_ADDR) and the reason of the invalid access (bit field AEIM_XPT_STA). A write access to this register (independent of the written data), always resets the bit fields AEIM_XPT_STA and AEIM_XPT_ADDR and the next invalid AEIM access is captured by this register, again.
2. If the i-th cluster does not provide an MCS module, this register is not available.

38.13.4.6 CCM[i]_ARP[z]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E2000_H$ + (200_H × i) + (8_H × z)

Value after reset: 0003 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WPROT_AEI	—	—	—	—	—	—	DIS_PROT	—	—	—	—	SIZE			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.122 CCM[i]_ARP[z]_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31	WPROT_AEI	AEI slave write protection. 0: Write protection to address range from AEI slave is disabled. 1: Write protection to address range from AEI slave is enabled.
30 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	DIS_PROT	DIS_PROT: Disable ARP protection. 0: Bit WPROTx (WPROT_AEI) defines write protection for selected address range. 1: Bit WPROTx (WPROT_AEI) explicitly allows write access to selected address range. **NOTE** This bit field is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	SIZE	Size of ARP Size of memory range protector z. **NOTES** 1. The actual size of a protected memory range is defined as 2 SIZE address locations, whereas the bit field SIZE is interpreted as an unsigned integer number. 2. The values 0, 1 and 2 are not supported for this bit field and cannot be configured. A write access with such a value to this register signalizes an invalid module access at the AEI status signal and it sets the bit field SIZE to the value 3. 3. This bit field is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

Table 38.122 CCM[i]_ARP[z]_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
15 to 0	ADDR	ARP base address. Base address for address range protector z. **NOTES** 1. Only the bits 5 to AAW-1 of this bit field are implemented as registers. The bits AAW to 15 are reserved and always read and written as zeros. The bits 0 to 4 are functionally used for the definition of an ARP but they are always read and written as zeros. 2. The actual base address for a protected address range is only defined by the upper AAW-(SIZE +2) bits (bit position 2+ SIZE to bit position AAW-1) of bit field ADDR. The lower SIZE +2 bits (bit 0 to SIZE +1) are ignored for the address calculation and assumed as zeros. 3. This bit field is only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.

Note: The address range interval that is protected by this ARP can be calculated as [(ADDR AND NOT 4*(2SIZE-1)); (ADDR AND NOT 4*(2SIZE-1)) + 4*(2SIZE-1)] assuming a byte wise addressing, an unsigned integer representation for the bit fields SIZE and ADDR. NOT and AND are bitwise logical operators. The incrementation interval for neighboring memory location is always 4.

38.13.4.7 CCM[i]_ARP[z]_PROT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E2004_H$ + (200_H × i) + (8_H × z)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WPROT7	WPROT6	WPROT5	WPROT4	WPROT3	WPROT2	WPROT1	WPROT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.123 CCM[i]_ARP[z]_PROT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	WPROT[7:0]	Write Protection MCS channel k. 0: Write protection to ARP's address range for MCS channel 0 is disabled. 1: Write protection to ARP's address range for MCS channel 0 is enabled.

Notes:

1. Only the first T bits of this register (bit 0 to T-1) are functionally implemented. The other bits (bit T to 31) are reserved. Parameter T reflects the number of available MCS channels in the cluster's MCS module.
2. The bit fields of this register are only writable if bit field CLS_PROT of register CCM[i]_PROT is cleared.
3. The meaning of the bit fields WPROTx can be changed by the bit field DIS_PROT of register CCM[i]_ARP[z]_CTRL.

38.13.4.8 CCM[i]_HW_CONF

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $E21DC_H$ + (200_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	INT_CLK_EN_GEN	TOM_TRIG_INTCHAIN					ATOM_TRIG_INTCHAIN				IRQ_MODE_SINGLE_PULSE	IRQ_MODE_PULSE_NOTIFY	IRQ_MODE_PULSE	IRQ_MODE_LEVEL
Value after reset	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ARU_CONNECT_CONFIG	ERM	RAM_INIT_RST	TOM_TRIG_CHAIN			TOM_OUT_RST	ATOM_TRIG_CHAIN			ATOM_OUT_RST	CFG_CLOCK_RATE	SYNC_INPUT_REG	BRIDGE_MODE_RST	GRSTEN
Value after reset	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.124 CCM[i]_HW_CONF Register Contents (1/3)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	INT_CLK_EN_GEN	INT_CLK_EN_GEN: Internal clock enable generation 0: GTM external clock enable signals in use 1: GTM internal clock enable signals in use
28 to 24	TOM_TRIG_INTCHAIN	TOM internal trigger chain length without synchronization register It defines after which TOM channel count a synchronization register is introduced into trigger chain. Valid values are 1 to 16. 8 means that in channel 8 of the tom instances a synchronization register is placed.
23 to 20	ATOM_TRIG_INTCHAIN	ATOM internal trigger chain length without synchronization register It defines after which ATOM channel count a synchronization register is introduced into trigger chain. Valid values are 1 to 8. 4 means that in channel 4 of the atom instances a synchronization register is placed.
19	IRQ_MODE_SINGLE_PULSE	IRQ_MODE_SINGLE_PULSE 0: Single pulse mode not available 1: Single pulse mode available
18	IRQ_MODE_PULSE_NOTIFY	IRQ_MODE_PULSE_NOTIFY 0: Pulse notify mode not available 1: Pulse notify mode available
17	IRQ_MODE_PULSE	IRQ_MODE_PULSE 0: Pulse mode not available 1: Pulse mode available
16	IRQ_MODE_LEVEL	IRQ_MODE_LEVEL 0: Level mode not available 1: Level mode available
15	Reserved	When read, the value after reset is returned.
14	ARU_CONNECT_CONFIG	Defines number of parallel ARU ports 0: 2 ARU ports available (two independent counter) 1: 1 ARU port available

Table 38.124 CCM[i]_HW_CONF Register Contents (2/3)

Bit Position	Bit Name	Function
13	ERM	Enable RAM1 MSB for available MCS modules 0: MSB of MCS RAM1 address not used 1: MSB of MCS RAM1 address used **NOTE** The bit reflects the state of the configuration parameter ERM mentioned in the specification of MCFG.
12	RAM_INIT_RST	RAM initialization from reset 0: RAM is not initialized after reset 1: RAM is initialized after reset
11 to 9	TOM_TRIG_CHAIN	TOM trigger chain length without synchronization register It defines after which TOM instance count a synchronization register is introduced into trigger chain (after TOM_TRIG_<i> output if instance i and TOM_TRIG_<i+1> input of instance i+1). Valid values are 1 to 7. 1 means that after each instance a synchronization register is placed.
8	TOM_OUT_RST	TOM_OUT reset level 0: TOM_OUT reset level is '0' 1: TOM_OUT reset level is '1' **NOTE** This value represents the TOM output level after reset. The inverse value of this bit is the reset value of bit SL in all TOM channels.
7 to 5	ATOM_TRIG_CHAIN	ATOM trigger chain length without synchronization register It defines after which ATOM instance count a synchronization register is introduced into trigger chain (after ATOM_TRIG_<i> output if instance i and ATOM_TRIG_<i+1> input of instance i+1). Valid values are 1 to 7. 1 means that after each instance a synchronization register is placed.
4	ATOM_OUT_RST	ATOM_OUT reset level 0: ATOM_OUT reset level is '0' 1: ATOM_OUT reset level is '1' **NOTE** This value represents the ATOM output level after reset. The inverse value of this bit is the reset value of bit SL in all ATOM channels.
3	CFG_CLOCK_RATE	Clocks per ARU transfer 0: Each system clock an ARU transfer is scheduled 1: Each second system clock an ARU transfer is scheduled. ARU transfer rate is half the system clock frequency. **NOTE** This value defines also the availability of configuration bits in register GTM_CLS_CLK_CFG. if CFG_CLOCK_RATE = 0_B, only the values 00_B and 01_B are valid for bit fields CLS[x]x_CLK_DIV. If CFG_CLOCK_RATE = 1_B, only the values 00_B, 01_B and 10_B are valid for bit fields CLS[x]x_CLK_DIV.
2	SYNC_INPUT_REG	Additional pipelined stage in synchronous bridge mode 0: No additional pipelined stage implemented. 1: Additional pipelined stage implemented. All accesses in synchronous mode will be increased by one clock cycle. **NOTE** This register is only relevant (if existing) for synchronous bridge mode.

Table 38.124 CCM[i]_HW_CONF Register Contents (3/3)

Bit Position	Bit Name	Function
1	BRIDGE_MODE_RST	Bridge mode after reset 0: Bridge starts in synchronous mode after reset 1: Bridge starts in asynchronous mode after reset
0	GRSTEN	Global Reset Enable 0: Global GTM reset register disabled 1: Global GTM reset register enabled

38.13.4.9 CCM[i]_TIM_AUX_IN_SRC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21E0_H$ + (200_H × i)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SEL_OUT_N_CH7	SEL_OUT_N_CH6	SEL_OUT_N_CH5	SEL_OUT_N_CH4	SEL_OUT_N_CH3	SEL_OUT_N_CH2	SEL_OUT_N_CH1	SEL_OUT_N_CH0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SRC_CH7	SRC_CH6	SRC_CH5	SRC_CH4	SRC_CH3	SRC_CH2	SRC_CH1	SRC_CH0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.125 CCM[i]_TIM_AUX_IN_SRC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	SEL_OUT_N_CH[7:0]	Use DTM_OUT or DTM_OUT_N signals as AUX_IN source of TIM[i] channel [7:0] 0: Use DTM_OUT signal as AUX_IN source of TIM[i] 1: Use DTM_OUT_N signal as AUX_IN source of TIM[i]
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SRC_CH7	Defines AUX_IN source of TIM[i] channel 7 SEL_OUT_N_CH7 = 0 / SEL_OUT_N_CH7 = 1: 0: CDTM[i].DTM1 Output DTM_OUT3 selected/ CDTM[i].DTM0 Output DTM_OUT0_N selected 1: CDTM[i].DTM5 Output DTM_OUT3 selected/ CDTM[i].DTM4 Output DTM_OUT0_N selected
6	SRC_CH6	Defines AUX_IN source of TIM[i] channel 6 SEL_OUT_N_CH6 = 0 / SEL_OUT_N_CH6 = 1: 0: CDTM[i].DTM1 Output DTM_OUT2 selected/ CDTM[i].DTM1 Output DTM_OUT3_N selected 1: CDTM[i].DTM5 Output DTM_OUT2 selected/ CDTM[i].DTM5 Output DTM_OUT3_N selected
5	SRC_CH5	Defines AUX_IN source of TIM[i] channel 5 SEL_OUT_N_CH5 = 0 / SEL_OUT_N_CH5 = 1: 0: CDTM[i].DTM1 Output DTM_OUT1 selected/ CDTM[i].DTM1 Output DTM_OUT2_N selected 1: CDTM[i].DTM5 Output DTM_OUT1 selected/ CDTM[i].DTM5 Output DTM_OUT2_N selected
4	SRC_CH4	Defines AUX_IN source of TIM[i] channel 4 SEL_OUT_N_CH4 = 0 / SEL_OUT_N_CH4 = 1: 0: CDTM[i].DTM1 Output DTM_OUT0 selected/ CDTM[i].DTM1 Output DTM_OUT1_N selected 1: CDTM[i].DTM5 Output DTM_OUT0 selected/ CDTM[i].DTM5 Output DTM_OUT1_N selected
3	SRC_CH3	Defines AUX_IN source of TIM[i] channel 3 SEL_OUT_N_CH3 = 0 / SEL_OUT_N_CH3 = 1: 0: CDTM[i].DTM0 Output DTM_OUT3 selected/ CDTM[i].DTM1 Output DTM_OUT0_N selected 1: CDTM[i].DTM4 Output DTM_OUT3 selected/ CDTM[i].DTM5 Output DTM_OUT0_N selected

Table 38.125 CCM[i]_TIM_AUX_IN_SRC Register Contents (2/2)

Bit Position	Bit Name	Function
2	SRC_CH2	Defines AUX_IN source of TIM[i] channel 2 SEL_OUT_N_CH2 = 0 / SEL_OUT_N_CH2 = 1: 0: CDTM[i].DTM0 Output DTM_OUT2 selected/ CDTM[i].DTM0 Output DTM_OUT3_N selected 1: CDTM[i].DTM4 Output DTM_OUT2 selected/ CDTM[i].DTM4 Output DTM_OUT3_N selected
1	SRC_CH1	Defines AUX_IN source of TIM[i] channel 1 SEL_OUT_N_CH1 = 0 / SEL_OUT_N_CH1 = 1: 0: CDTM[i].DTM0 Output DTM_OUT1 selected/ CDTM[i].DTM0 Output DTM_OUT2_N selected 1: CDTM[i].DTM4 Output DTM_OUT1 selected/ CDTM[i].DTM4 Output DTM_OUT2_N selected
0	SRC_CH0	Defines AUX_IN source of TIM[i] channel 0 SEL_OUT_N_CH0 = 0 / SEL_OUT_N_CH0 = 1: 0: CDTM[i].DTM0 Output DTM_OUT0 selected/ CDTM[i].DTM0 Output DTM_OUT1_N selected 1: CDTM[i].DTM4 Output DTM_OUT0 selected/ CDTM[i].DTM4 Output DTM_OUT1_N selected

38.13.4.10 CCM[i]_EXT_CAP_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E21E4_H$ + (200_H × i)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIM_IP1_EXT_CAP_EN								TIM_I_EXT_CAP_EN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.126 CCM[i]_EXT_CAP_EN Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	TIM_IP1_EXT_CAP_EN	TIM[i+1]_EXT_CAPTURE signal forwarding enable 0: Disable forwarding of signal TIM[i+1]_EXT_CAPTURE to MCS[i] 1: Enable forwarding of signal TIM[i+1]_EXT_CAPTURE to MCS[i]
7 to 0	TIM_I_EXT_CAP_EN	TIM[i]_EXT_CAPTURE signal forwarding enable 0: Disable forwarding of signal TIM[i]_EXT_CAPTURE to MCS[i] 1: Enable forwarding of signal TIM[i]_EXT_CAPTURE to MCS[i] **NOTE** The trigger event forwarding is possible from TIM[i] and TIM[i+1] to MCS[i].

38.13.4.11 CCM[i]_TOM_OUT

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $E21E8_H$ + (200_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOM_OUT_N															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM_OUT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.127 CCM[i]_TOM_OUT Register Contents

Bit Position	Bit Name	Function
31 to 16	TOM_OUT_N	Output level snapshot of TOM[i]_OUT_N all channels actual level of primary output ports TOM[i]_OUT_N of channel 0 to 15 (after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.
15 to 0	TOM_OUT	Output level snapshot of TOM[i]_OUT all channels actual level of primary output ports TOM[i]_OUT of channel 0 to 15 (after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.

38.13.4.12 CCM[i]_ATOM_OUT

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $E21EC_H$ + (200_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ATOM_IP1_OUT_N								ATOM_IP1_OUT							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATOM_I_OUT_N								ATOM_I_OUT							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.128 CCM[i]_ATOM_OUT Register Contents

Bit Position	Bit Name	Function
31 to 24	ATOM_IP1_OUT_N	Output level snapshot of ATOM[i+1]_OUT_N all channels Actual level of primary output ports ATOM[i+1]_OUT_N of channel 0 to 7 (after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.
23 to 16	ATOM_IP1_OUT	Output level snapshot of ATOM[i+1]_OUT all channels Actual level of primary output ports ATOM[i+1]_OUT of channel 0 to 7 (after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.
15 to 8	ATOM_I_OUT_N	Output level snapshot of ATOM[i]_OUT_N all channels Actual level of primary output ports ATOM[i]_OUT_N of channel 0 to 7 (after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.
7 to 0	ATOM_I_OUT	Output level snapshot of ATOM[i]_OUT all channels Actual level of primary output ports ATOM[i]_OUT of channel 0 to 7(after DTM) **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor. See CCM[i]_HW_CONF for chosen value.

38.13.4.13 CCM[i]_HW_CONF2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $E21D4_H$ + (200_H × i)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AXIM_POSTED_WRITE	AXIM_SEC_ACC	AXIM_PRIV_ACC	AXIM_ID_WIDTH				
Value after reset	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.129 CCM[i]_HW_CONF2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	AXIM_POSTED_WRITE	Write transaction without response 0: Always use none–posted write. 1: Use none–posted write.
6	AXIM_SEC_ACC	Secure AXI master access constant 0: Secure AXI master access is not allowed. 1: Secure AXI master access is allowed.
5	AXIM_PRIV_ACC	Privileged AXI master access constant 0: Privileged AXI master access is not allowed. 1: Privileged AXI master access is allowed.
4 to 0	AXIM_ID_WIDTH	Defines which LSB of AXIM_ID are send to the bus How many lower of AXIM_D will be send to the bus

38.14 Time Base Unit (TBU)

38.14.1 Overview

The Time Base Unit TBU provides common time bases for the GTM-IP. The TBU sub-module is organized in channels, where the number of channels is device dependent. There are up to four channels implemented inside the TBU. The time base register TBU_CH0_BASE of TBU channel 0 is 27 bits wide and it is configurable whether the lower 24 bit or the upper 24 bit are provided to the GTM as signal TBU_TS0. The two TBU channels 1 and 2 have a time base register TBU_CH[y]_BASE (y: 1, 2) of 24 bit length. The time base register value TBU_TS[y] is provided to subsequent sub-modules of the GTM. The time base register of TBU channel 3 TBU_CH3_BASE is 24 bits wide. It is used as a modulo counter by TBU_CH3_BASE_MARK to get a relative angle clock to TBU_CH[y]_BASE. The absolute angle clock value for the current TBU_CH3_BASE is captured in TBU_CH3_BASE_CAPTURE. TBU_CH[y]_BASE = TBU_CH3_BASE_CAPTURE + ... + TBU_CH3_BASE - DIRy × TBU_CH3_BASE_MARK DIRy : direction value for time base y (y:1, 2) 0 up counter 1 down counter

NOTE

The right-hand sum is limited to 24 bit.

The TBU_UP[y] (y: 1, 2) signals are set to high for a single SYS_CLK period, whenever the corresponding signal TBU_TS[y] (y: 1, 2) is getting updated. The signal TBU_UP0_L is set to high for a single SYS_CLK period if the signal TBU_TS0 and TBU_TS0x is getting updated and TBU_UP0_H is set to high for a single SYS_CLK period, whenever the upper 24 bit of TBU_TS0 are updated.

The time base channels can run independently of each other and can be enabled and disabled synchronously by control bits in a global TBU channel enable register TBU_CHEN. **Figure 38.32** shows a block diagram of the Time Base Unit.

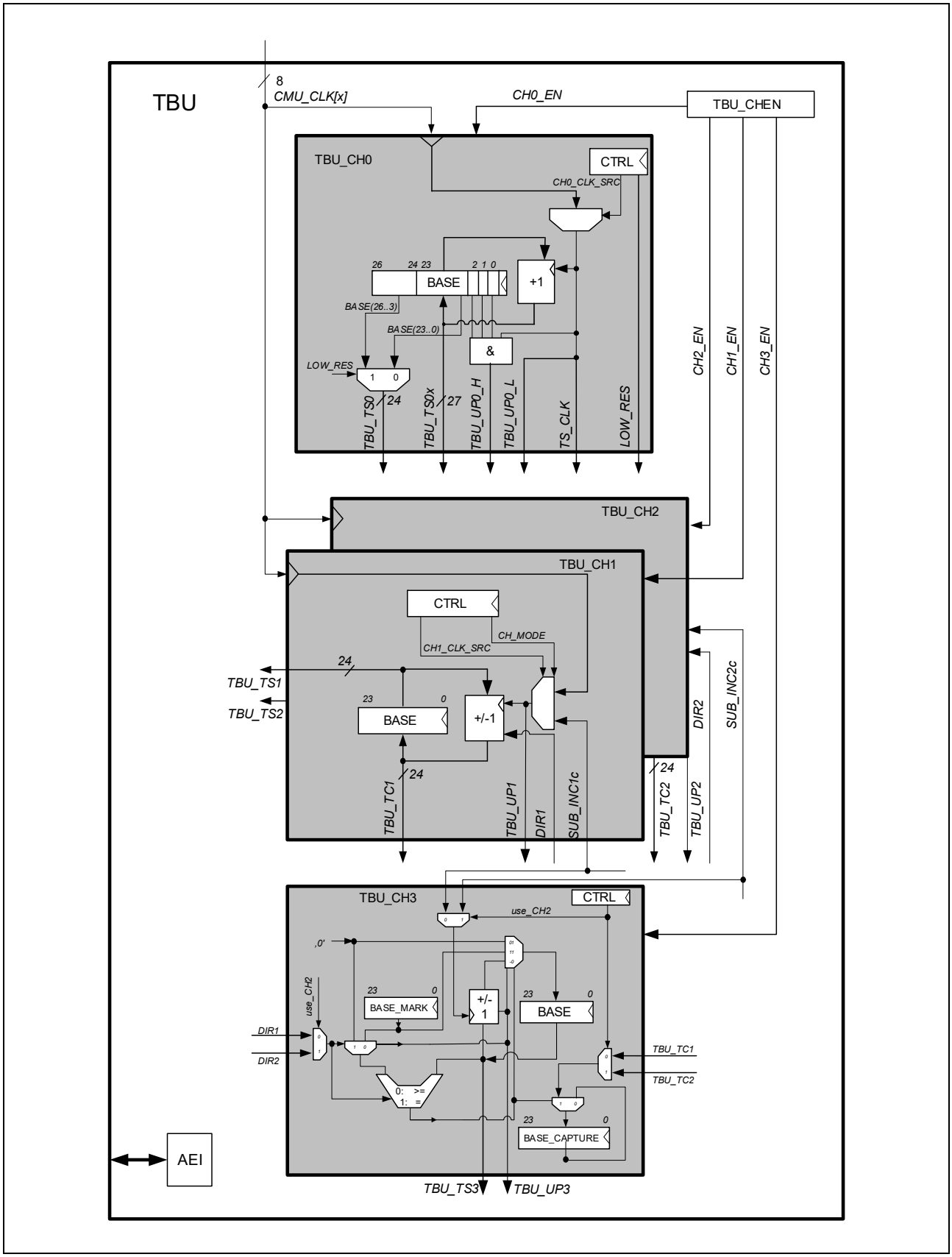


Figure 38.32 TBU Block Diagram

Dependent on the device a third TBU channel exists which offers the same functionality as the time base channel 1.

The configuration of the independent time base channels TBU_CH[z]_BASE is done via the AEI interface. The TBU channel 0 to 2 may select one of the eight CMU_CLK[x] (x: 0 to 7) signals coming from the CMU sub-module.

For TBU channels 1 and 2 an additional clock signal SUB_INC[y]c (y: 1, 2) coming from the DPLL can be selected as input clock for the TBU_CH[y]_BASE. This clock in combination with the DIR[y] signals determines the counter direction of the TBU_CH[y]_BASE.

The selected time stamp clock signal for the TBU_CH0 sub-unit is served via the TS_CLK signal line to the DPLL sub-module. The TS_CLK signal equals the signal TBU_UP0

38.14.2 TBU Channels

The time base values are generated within the TBU time base channels in two independent and one dependent operation modes.

In all modes, the time base register TBU_CH[z]_BASE (z: 0 to 3) can be initialized with a start value just before enabling the corresponding TBU channel.

Moreover, the time base register TBU_CH[z]_BASE (z: 0 to 3) can always be read in order to determine the actual value of the counter.

38.14.2.1 Independent Modes

(1) Free Running Counter Mode

TBU channel 0 provides a 27 bit counter in a free running counter mode. Dependent on the bit field LOW_RES of register TBU_CH0_CTRL, the lower 24 bits (bit 0 to 23) or the upper 24 bits (bits 3 to 26) are provided to the GTM sub-modules.

TBU channel 1 and 2 provides a 24 bit counter in a free running counter mode enabled by reset CH_MODE of register TBU_CH[y]_CTRL (y:1, 2).

In TBU Free running counter mode, the time base register TBU_CH[v]_BASE (v: 0 to 2) is updated on every specified incoming clock event by the selected signal CMU_CLK[x] (x: 0 to 8) (dependent on TBU_CH[v]_CTRL (v:0 to 2) register). In general the time base register TBU_CH[v]_BASE is incremented on every CMU_CLK[x] clock tick.

(2) Forward/Backward Counter Mode

TBU channel 1 and 2 provides a 24 bit forward/backward counter enabled by set CH_MODE of register TBU_CH[y]_CTRL (y:1, 2). In this mode the DIR[y] signal provided by the DPLL is taken into account.

The value of the time base register TBU_CH[y]_BASE is incremented in case when the DIR[y] signal equals 0 and decremented in case when the DIR[y] signal is 1.

38.14.2.2 Dependent Mode

(1) Modulo Counter Mode

TBU channel 3 provides a 24 bit forward/backward modulo counter. The clock SUB_INC[y]c and counter direction DIR[y] provided by DPLL is selected by use_CH2 of register TBU_CH3_CTRL.

The modulo value is defined in TBU_CH3_BASE_MARK. In forward counter mode if TBU_CH3_BASE value is reaching TBU_CH3_BASE_MARK TBU_CH3_BASE is reset and TBU_TS[y] is captured in TBU_CH3_BASE_CAPTURE. In backward counter mode if TBU_CH3_BASE value is reaching '0' TBU_CH3_BASE is set to TBU_CH3_BASE_MARK and TBU_TS[y] is captured in TBU_CH3_BASE_CAPTURE.

38.14.3 TBU Configuration Registers Overview

TBU contains following configuration registers:

Table 38.130 Register List

Symbol	Register Name	Details in Section
TBU_CHEN	TBU global channel enable	**38.14.4.1**
TBU_CH0_CTRL	TBU channel 0 control	**38.14.4.2**
TBU_CH0_BASE	TBU channel 0 base	**38.14.4.3**
TBU_CH1_CTRL	TBU channel 1 control	**38.14.4.4**
TBU_CH2_CTRL	TBU channel 2 control	**38.14.4.5**
TBU_CH[y]_BASE	TBU channel y base	**38.14.4.6**
TBU_CH3_CTRL	TBU channel 3 control	**38.14.4.7**
TBU_CH3_BASE	TBU channel 3 base	**38.14.4.8**
TBU_CH3_BASE_MARK	TBU channel 3 modulo value	**38.14.4.9**
TBU_CH3_BASE_CAPTURE	TBU channel 3 base captured	**38.14.4.10**

Note: In a typical application the Time Base Unit (TBU) considers channels 0, 1 and 3 only. In this case register addresses 0x20...0x2C are reserved and shall be read as zero. Channel 2 can be additionally implemented on special high-end application requirements.

38.14.4 TBU Configuration Registers Description

38.14.4.1 TBU_CHEN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00100_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ENDIS_CH3		ENDIS_CH2		ENDIS_CH1		ENDIS_CH0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.131 TBU_CHEN Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7, 6	ENDIS_CH3	TBU channel 3 enable/disable control. See bits 1, 0
5, 4	ENDIS_CH2	TBU channel 2 enable/disable control. See bits 1, 0
3, 2	ENDIS_CH1	TBU channel 1 enable/disable control. See bits 1, 0
1, 0	ENDIS_CH0	TBU channel 0 enable/disable control. READ access: 00_B: Channel is disabled 11_B: Channel is enabled WRITE access: 01_B: Disable channel 10_B: Enable channel

38.14.4.2 TBU_CH0_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00104_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CH_CLK_SRC			LOW_RES
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.132 TBU_CH0_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 1	CH_CLK_SRC	Clock source for channel x (x = 0 to 2) time base counter 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected **NOTE** This value can only be modified if channel 0 is disabled.
0	LOW_RES	TBU_CH0_BASE register resolution. 0: TBU channel uses lower counter bits (bit 0 to 23) 1: TBU channel uses upper counter bits (bit 3 to 26) **NOTES** 1. The two resolutions for the TBU channel 0 can be used in the TIM channel 0 and the DPLL sub-modules. 2. This value can only be modified if channel 0 is disabled.

38.14.4.3 TBU_CH0_BASE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00108_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	BASE										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.133 TBU_CH0_BASE Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 0	BASE	Time base value for channel 0. **NOTES** 1. The value of BASE can only be written if the TBU channel 0 is disabled. 2. If channel 0 is enabled, a read access to this register provides the current value of the underlying 27 bit counter.

38.14.4.4 TBU_CH1_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0010C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CH_CLK_SRC			CH_MODE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.134 TBU_CH1_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 1	CH_CLK_SRC	Clock source for channel 1 time base counter 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected **NOTE** This value can only be modified if channel 1 was disabled.
0	CH_MODE	Channel mode 0: Free running counter mode 1: Forward/backward counter mode **NOTE** This value can only be modified if channel 1 is disabled. In Free running counter mode the CMU clock source specified by CH_CLK_SRC is used for the counter. In Forward/Backward counter mode the SUB_INC1c clock signal in combination with the DIR1 input signal is used to determine the counter direction and clock frequency.

38.14.4.5 TBU_CH2_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00114_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CH_CLK_SRC			CH_MODE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.135 TBU_CH1_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 1	CH_CLK_SRC	Clock source for channel 2 time base counter 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected **NOTE** This value can only be modified if channel 2 was disabled.
0	CH_MODE	Channel mode 0: Free running counter mode 1: Forward/backward counter mode **NOTE** This value can only be modified if channel 2 is disabled. In Free running counter mode the CMU clock source specified by CH_CLK_SRC is used for the counter. In Forward/Backward counter mode the SUB_INC2c clock signal in combination with the DIR2 input signal is used to determine the counter direction and clock frequency.

38.14.4.6 TBU_CH[y]_BASE

Access: This register can be read or written in 32-bit units.

Address: TBU_CH1_BASE: <GTM_base> + 00110_H
TBU_CH2_BASE: <GTM_base> + 00118_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	BASE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.136 TBU_CH[y]_BASE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	BASE	Time base value for channel y (y = 1, 2) **NOTES** 1. The value of BASE can only be written if the corresponding TBU channel y is disabled. 2. If the corresponding channel y is enabled, a read access to this register provides the current value of the underlying counter.

38.14.4.7 TBU_CH3_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0011C_H$

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	USE_CH2	—	—	—	CH_MODE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Table 38.137 TBU_CH3_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	USE_CH2	Channel selector for modulo counter RWRITE access: 0: TBU_CH1 values used. (SUB_INC1c for clock, DIR1 for counter direction, TBU_TC1 for capturing) 1: TBU_CH2 values used. (SUB_INC2c for clock, DIR2 for counter direction, TBU_TC2 for capturing) **NOTE** This value can only be modified if channel 3 was disabled.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CH_MODE	Channel mode 1: Forward/backward counter mode

38.14.4.8 TBU_CH3_BASE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00120_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	BASE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.138 TBU_CH3_BASE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	BASE	Time base value for channel 3 **NOTES** 1. The value of BASE can only be written if the corresponding TBU channel 3 is disabled. 2. If the corresponding channel 3 is enabled, a read access to this register provides the current value of the underlying counter.

38.14.4.9 TBU_CH3_BASE_MARK

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00120_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	BASE_MARK							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE_MARK															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.139 TBU_CH3_BASE_MARK Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	BASE_MARK	Modulo value for channel 3 **NOTE** The value of BASE_MARK can only be written if the corresponding TBU channel 3 is disabled.

38.14.4.10 TBU_CH3_BASE_CAPTURE

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00128_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	BASE_CAPTURE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE_CAPTURE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.140 TBU_CH3_BASE_CAPTURE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	BASE_CAPTURE	Captured value of time base channel 1 or 2 **NOTE** When USE_CH2 = 0 TBU_TC1 is captured and if USE_CH2 is setting TBU_TC2 is captured.

38.15 Timer Input Module (TIM)

38.15.1 Overview

The Timer Input Module (TIM) is responsible for filtering and capturing input signals of the GTM. Several characteristics of the input signals can be measured inside the TIM channels. For advanced data processing the detected input characteristics of the TIM module can be routed through the ARU to subsequent processing units of the GTM.

Input characteristics mean either time stamp values of detected input rising or falling edges together with the new signal level or the number of edges received since channel enable together with the actual time stamp or PWM signal duration for a whole PWM period.

The architecture of TIM is shown in **Figure 38.33**.

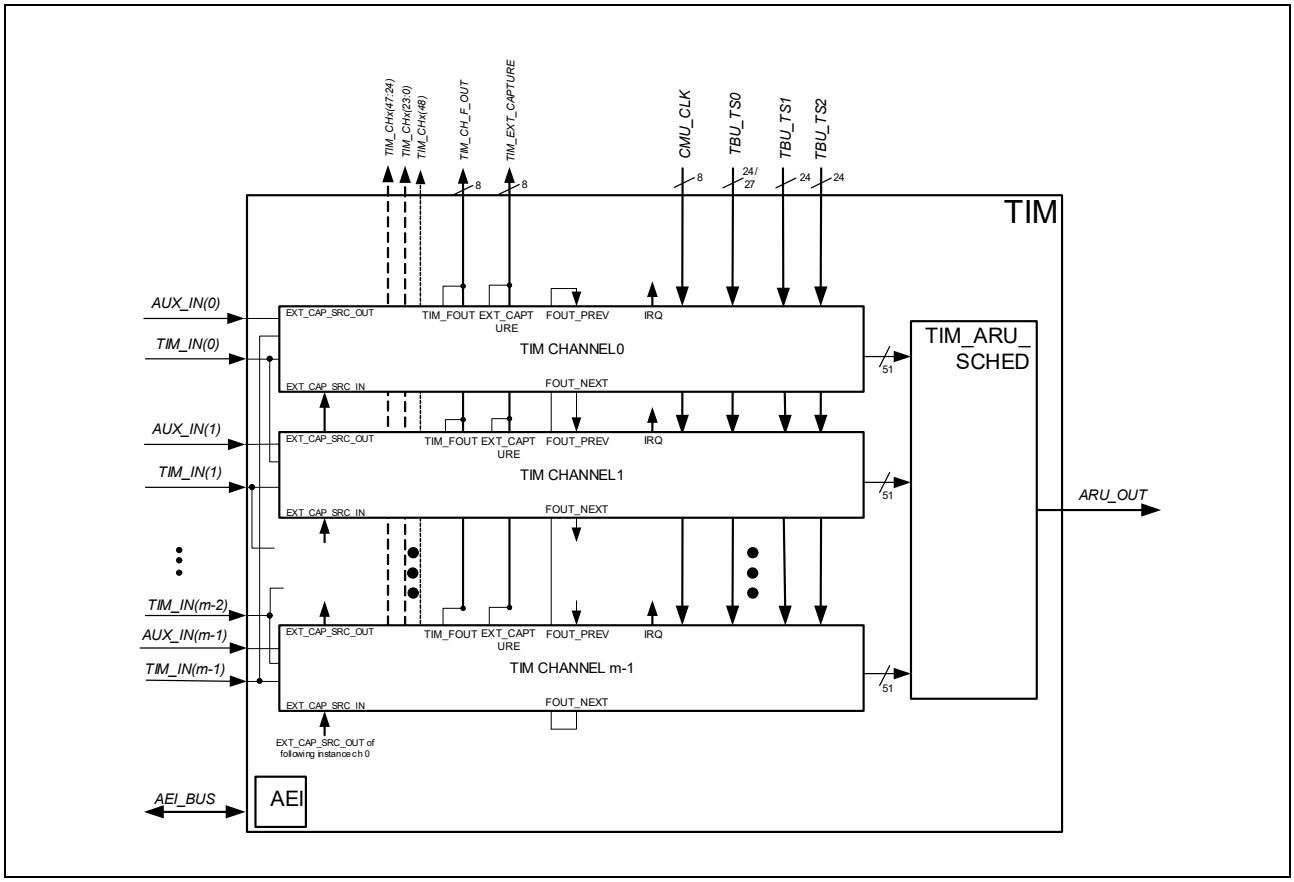


Figure 38.33 TIM Block Diagram

The number of channels m inside a TIM sub-module depends on the device.

Each of the m dedicated input signals are filtered inside the FLTx sub-unit of the TIM Module.

NOTE

The incoming input signals are synchronized to the clock SYS_CLK, resulting in a delay of two SYS_CLK periods for the incoming signals.

The measurement values can be read by the CPU directly via the AEI–Bus or they can be routed through the ARU to other sub-modules of the GTM.

For the GTM-IP TIM0 sub-module only, the dashed signal outputs TIM[i]_CH[x](23:0), TIM[i]_CH[x](47:24) and TIM[i]_CH[x](48) come from the TIM0 sub-module channels zero (0) to five (5) and are connected to MAP sub-module. There, they are used for further processing and for routing to the DPLL.

The two (three) time bases coming from the TBU are connected to the TIM channels to annotate time stamps to incoming signals. For TIM0 the extended 27 bit width time base TBU_TS0 is connected to the TIM channels, and the user has to select if the lower 24 bits (TBU_TS0(23:0)) or the higher 24 bits (TBU_TS0(26:3)) are stored inside the GPR0 and GPR1 registers.

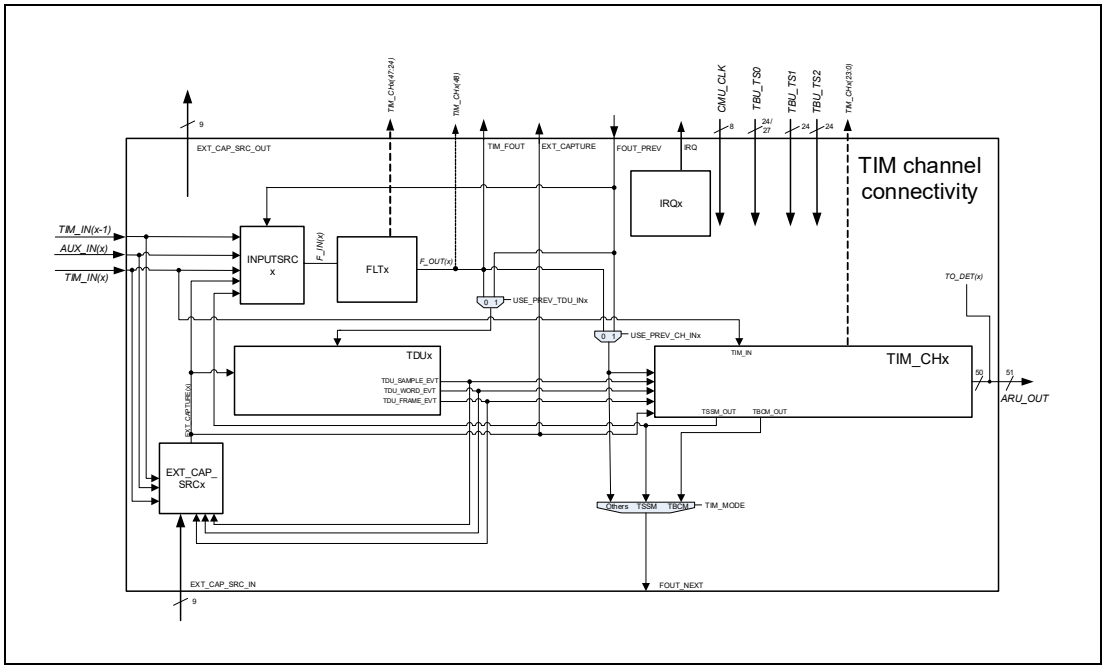


Figure 38.34 TIM channel internal connectivity

Above figure gives an overview of the channel internal connectivity of the sub units. The sub units with the major functionality are listed next: INPUT_SRCx: Select signal for processing by the Filter unit FLTx FLTx: The filter unit provides different filter mechanisms described in more detail in Chapter 13.2. TDUx: Timeout detection unit (no subsequent edge detected during a specified duration) TIM_CHx: Measurement unit; different measurements strategies configurable on the filtered signal IRQx: Local interrupt controller (enabling, status, ..) EXT_CAP_SRCx: Selects a local signal ext_capture(x) which is needed by certain functions Details are given in the next chapters.

Depending on the values of the configuration bit fields USE_PREV_TDU_INx, USE_PREV_CH_INx it is possible to operate on the signal of the local channel x or the previous channel x-1.

Depending on the value of the configuration bit field TIM_MODEx it is possible to provide different signals (via FOUT_NEXT) to the next channel.

In TBCM mode each capture event selected by the sensitive edges (CNTS) will be forwarded with the value of ECNT[0] to the following channel (via FOUT_NEXT).

38.15.1.1 Input source selection INPUTSRCx

It can be configured which source shall be used for processing in the FLT, TDU, TIM_CH units. It can be selected by the bit fields CICTRL and MODE_x, VAL_x in the register TIM[i]_IN_SRC which source is in use.

Alternatively the signal F_IN(x) can be generated by a 8 bit lookup table, which allows to define any function of 3 input sources.

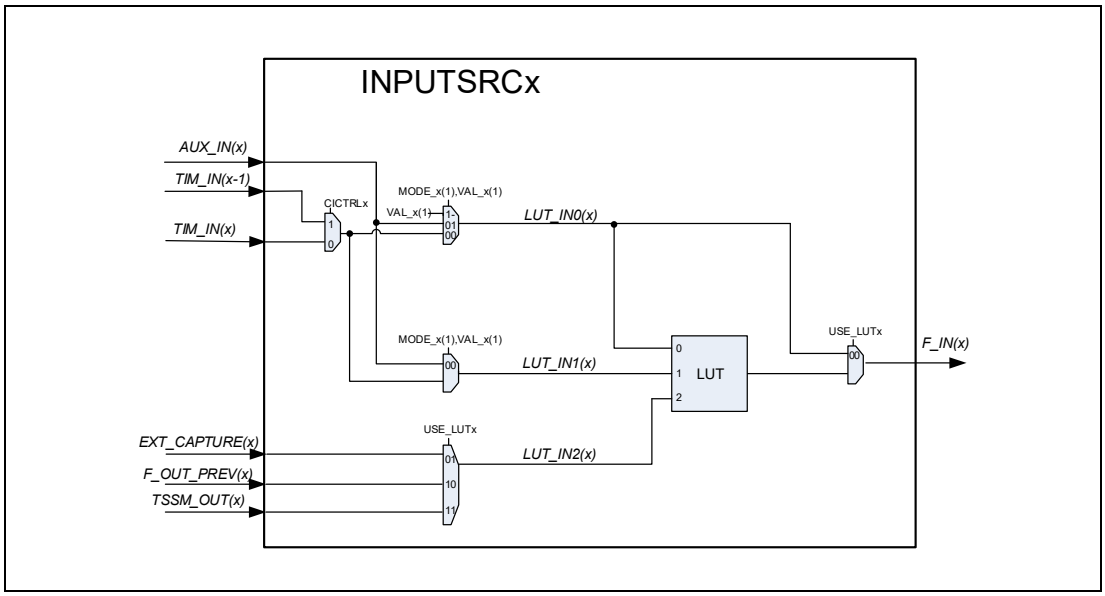


Figure 38.35 INPUTSRC Block Diagram

If USE_LUT = 00_B is set the lookup table signal generation is bypassed and the signal selection is performed as follows:

In a certain MODE_x, VAL_x combination the input signal F_IN(x) can be driven by VAL_x(1) with 0 or 1 directly.

Due to the fact that all 8 channels are bundled in the register TIM[i]_IN_SRC a synchronous control of all 8 input channels is possible.

Two adjacent channels can be combined by setting the CICTRL bit field in the corresponding TIM[i]_CH[x]_CTRL register. This allows for a combination of complex measurements on one input signal with two TIM channels.

The additional signal AUX_IN[x] can be selected as an input signal. The source of this signal is defined in **Figure 38.5, TIM auxiliary input multiplexing**.

If USE_LUT! = 00_B is set, the lookup table signal generation with following inputs is in use. See **Figure 38.35**:

Input LUT_INO(x) selection: TIM_IN(x) if CICTRLx = 0 and MODE_x (1) = 0 and VAL_x (1) = 0 TIM_IN(x–1) if CICTRLx = 1 and MODE_x (1) = 0

and VAL_x (1) = 0 AUX_IN(x) if MODE_x (1) = 0 and VAL_x (1) = 1 VAL_x (1) if MODE_x (1) = 1

Input LUT_IN1(x) selection: AUX_IN(x) if MODE_x (1) = 0 and VAL_x (1) = 0 TIM_IN(x) if CICTRLx = 0 TIM_IN(x–1) if CICTRLx = 1

Input LUT_IN2(x) selection: EXT_CAPTURE(x) if USE_LUT = 01_B FOUT_PREV(x) if USE_LUT = 10_B TSSM_OUT(x) if USE_LUT = 11_B

The lookup table is defined by the contents of the bit field TO_CNT2x. The lookup_table_index is defined by LUT_IN2(x) & LUT_IN1(x) &LUT_IN0(x). The signal F_IN(x) is generated by TO_CNT2x [lookup_table_index].

If USE_LUT! = 00_B is set, only limited functionality is available in the TDU. See bit field Slicing in the register TIM[i]_CH[x]_TDUV.

38.15.1.2 Input observation

It is possible to observe for all channels of one instance by reading TIM_INP_VAL the actual signal values of the following processing stages:

1. TIM_IN(7:0) signals after TIM input synchronization
2. TIM F_IN(7:0) signals after TIM INPUTSRC selection (input to TIM_FLT)
3. TIM F_OUT(7:0) signals after TIM filter functionality (output of TIM_FLT)

38.15.1.3 External capture source selection EXTCAPSRCx

Each channel can operate on an external capture signal EXT_CAPTURE. The source to use for this signal can be configured by the bit field EXT_CAP_SRCx in the register TIM[i]_CH[x]_ECTRL.

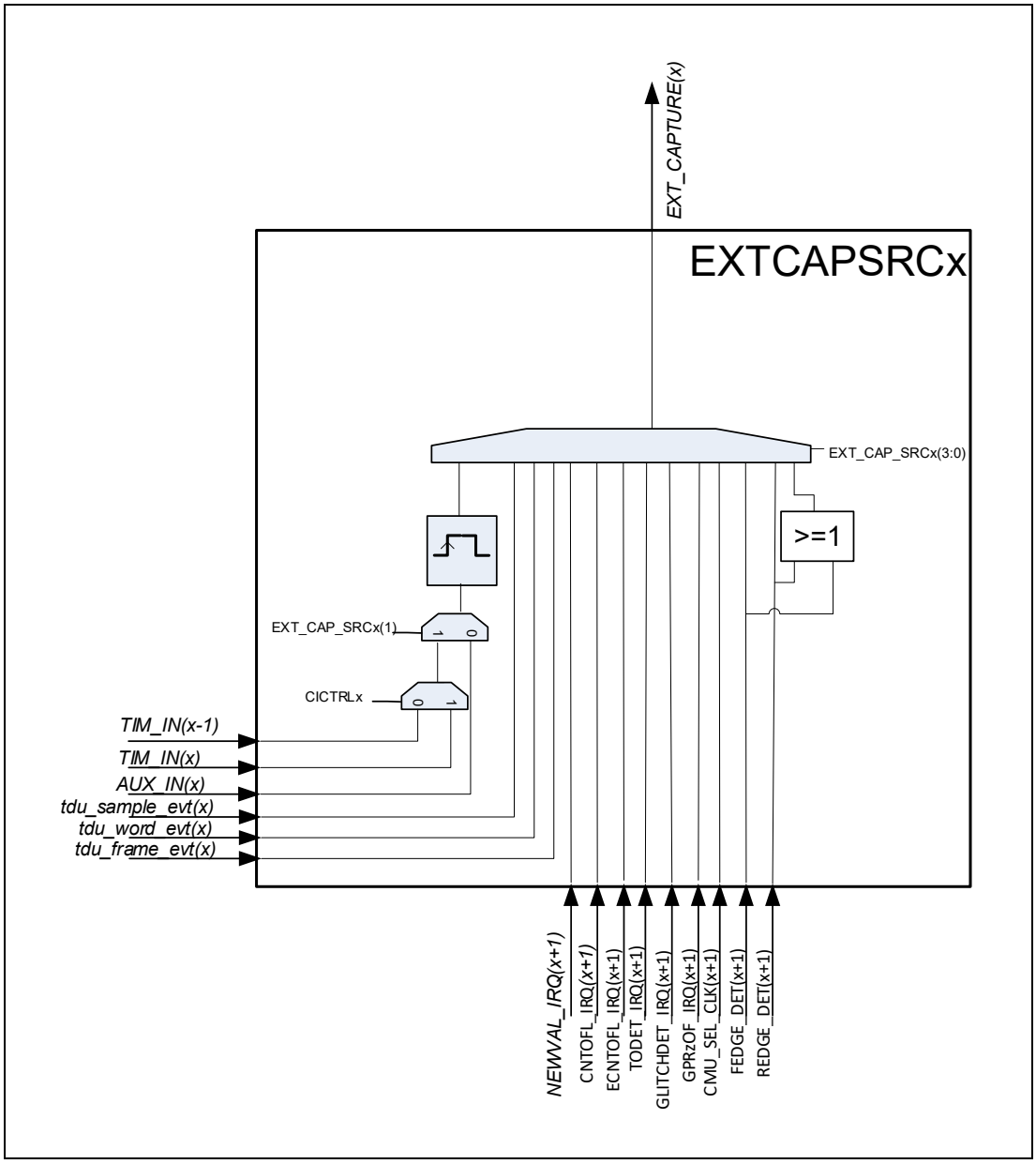


Figure 38.36 EXTCAPSRC Block Diagram

The external capture functionality can be enabled for the TIM channel x with the bit EXT_CAP_EN in the register TIM[i]_CH[x]_CTRL, it will trigger on each rising edge. A pulse generation for each rising edge of the selected input signal TIM_IN[x] and AUX_IN[x] is applied.

The six TIM channel interrupt sources can be triggered by the operation in the certain TIM channel modes. Alternatively they can be issued by a soft trigger using the corresponding bits in the register TIM[i]_CH[x+1]_FORCINT.

38.15.2 TIM Filter Functionality (FLT)

38.15.2.1 Overview

The TIM sub-module provides a configurable filter mechanism for each input signal. These filter mechanism is provided inside the FLT sub-unit.

FLT architecture is shown in **Figure 38.37**.

The filter includes a clock synchronization unit (CSU), an edge detection unit (EDU), and a filter counter associated to the filter unit (FLTU).

The CSU is synchronizing the incoming signal F_IN to the selected filter clock frequency, which is controlled with the bit field FLT_CNT_FRQ of register TIM[i]_CH[x]_CTRL.

The synchronized input signal F_IN_SYNC is used for further processing within the filter.

NOTE

Glitches with a duration go less than the selected CMU clock period is lost.

The filter modes can be applied individually to the falling and rising edges of an input signal. The following filter modes are available:

- Immediate edge propagation mode
- Individual de-glitch time mode (up/down counter)
- Individual de-glitch time mode (hold counter)
- Individual de-glitch time mode (reset counter)

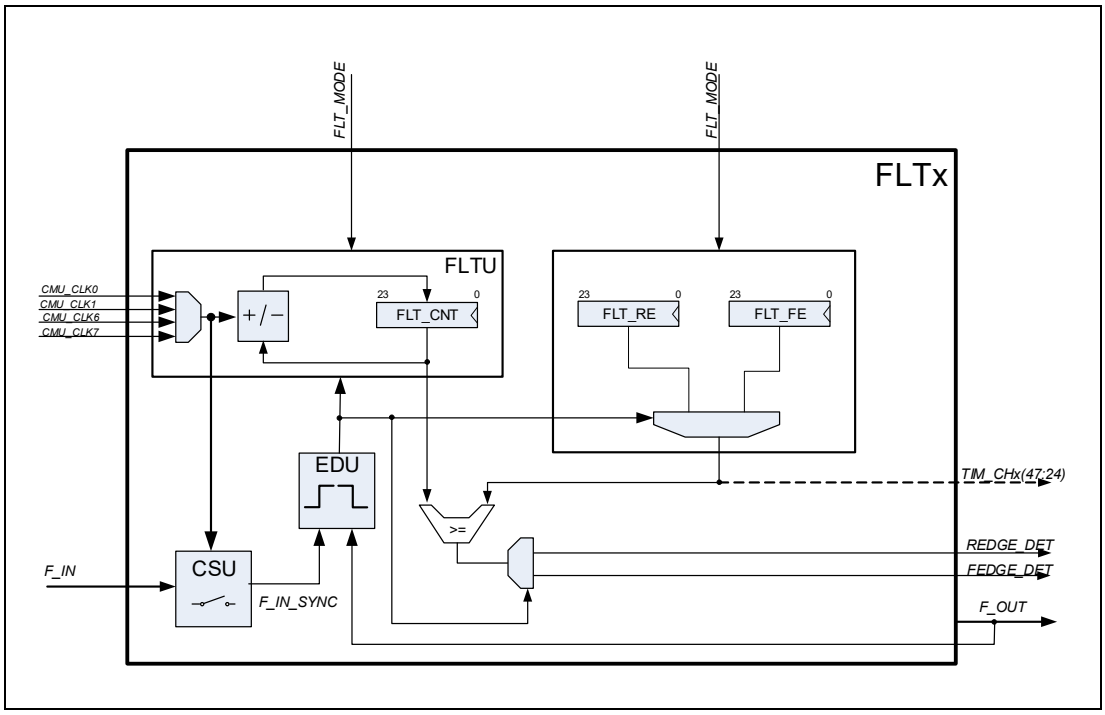


Figure 38.37 FLT Architecture

The filter parameters (deglitch and acceptance time) for the rising and falling edge can be configured inside the two filter parameter registers FLT_RE (rising edge) and FLT_FE (falling edge). The exact meaning of the parameter depends on the filter mode.

However the delay time T of both filter parameters FLT_xE can always be determined by: T = (FLT_xE +1)* T, whereas T is the clock period of the selected CMU clock signal in bit field FLT_CNT_FRQ of register TIM[i]_CH[x]_CTRL.

When a glitch is detected on an input signal a status flag GLITCHDET is set inside the TIM[i]_CH[x]_IRQ_NOTIFY register.

Table 38.141 gives an overview about the meanings for the registers FLT_RE and FLT_FE. In the individual deglitch time modes, the actual filter threshold for a detected regular edge is provided on the TIM[i]_CH[x](47:24) output line. In the case of immediate edge propagation mode, a value of zero is provided on the TIM[i]_CH[x](47:24) output line.

The TIM[i]_CH[x](47:24) output line is used by the MAP sub-module for further processing (see **Section 38.21**).

Table 38.141 Filter Parameter summary for the different Filter Modes

Filter mode	Meaning of FLT_RE	Meaning of FLT_FE
Immediate edge propagation	Acceptance time for rising edge	Acceptance time for falling edge
Individual de-glitch time (up/ down counter)	De-glitch time for rising edge	De-glitch time for falling edge
Individual de-glitch time (hold counter)	De-glitch time for rising edge	De-glitch time for falling edge
Individual de-glitch time (reset counter)	De-glitch time for rising edge	De-glitch time for falling edge

A counter FLT_CNT is used to measure the glitch and acceptance times.

The frequency of the FLT_CNT counter is configurable in bit field FLT_CNT_FRQ of register TIM[i]_CH[x]_CTRL.

The counter FLT_CNT can either be clock with the CMU_CLK0, CMU_CLK1, CMU_CLK6 or the CMU_CLK7 signal. These signals are coming from the CMU sub-module.

The FLT_CNT, FLT_FE and FLT_RE registers are 24-bit width. For example, when the resolution of the CMU_CLK0 signal is 50ns this allows maximal de-glitch and acceptance times of about 838ms for the filter.

38.15.2.2 TIM Filter Modes

(1) Immediate edge propagation mode

In immediate edge propagation mode after detection of an edge the new signal level on F_IN_SYNC is propagated to F_OUT with a delay of one T period and the new signal level remains unchanged until the configured acceptance time expires.

For each edge type the acceptance time can be specified separately in the FLT_RE and FLT_FE registers.

Each signal change on the input F_IN_SYNC during the duration of the acceptance time has no effect on the output signal level F_OUT of the filter but it sets the glitch GLITCHDET bit in the TIM[i]_CH[x]_IRQ_NOTIFY register.

After it expires an acceptance time the input signal F_IN_SYNC is observed and on signal level change the filter raises a new detected edge and the new signal level is propagated to F_OUT.

Independent of a signal level change the value of F_OUT is always set to F_IN_SYNC, when the acceptance time expires (see also **Figure 38.39**).

Figure 38.38 shows an example for the immediate edge propagation mode, in the case of rising edge detection. Both, the signal before filtering (F_IN) and after filtering (F_OUT) are shown. The acceptance time at1 is specified in the register FLT_RE.

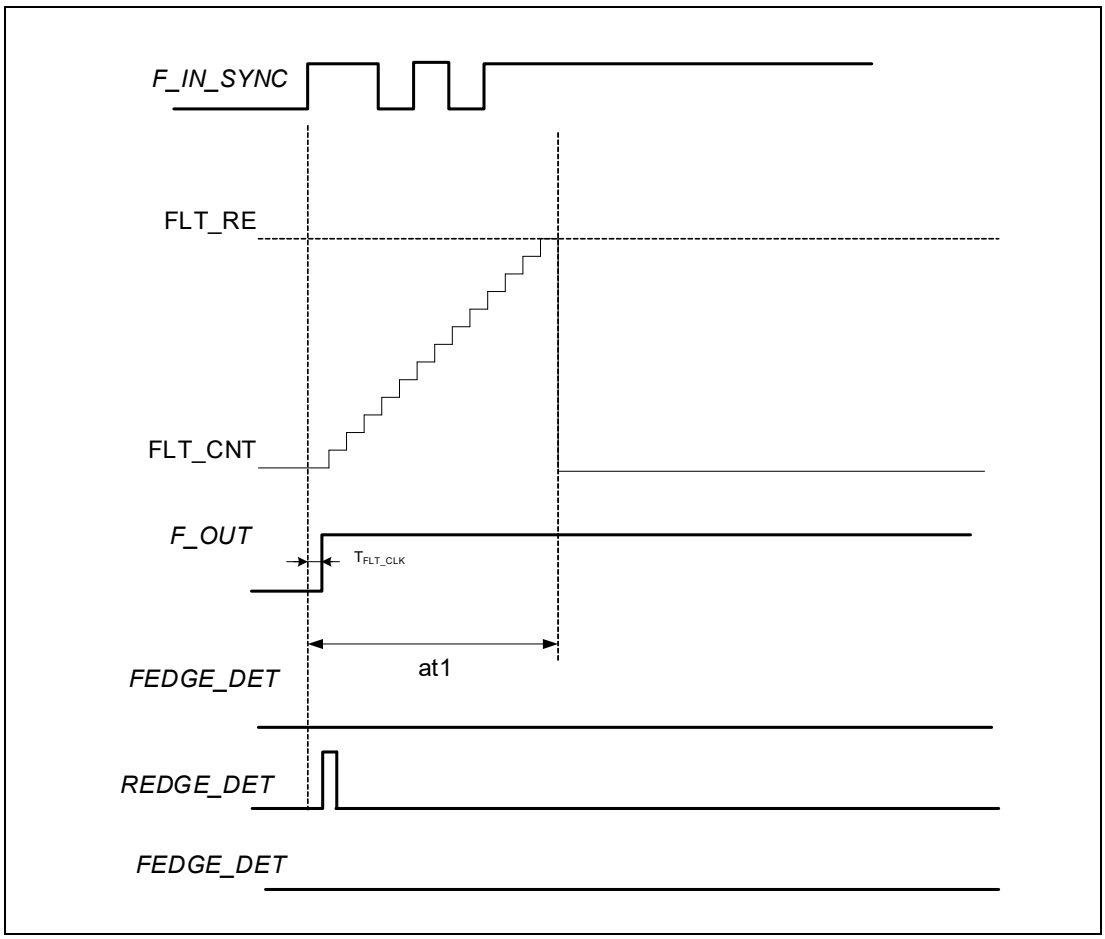


Figure 38.38 Immediate Edge Propagation Mode in the case of a rising edge

In immediate edge propagation mode the glitch measurement mechanism is not applied to the edge detection. Detected edges on F_IN_SYNC are transferred directly to F_OUT.

The counter FLT_CNT is incremented until acceptance time threshold is reached.

Figure 38.39 shows a more complex example of the TIM filter, in which both, rising and falling edges are configured in immediate edge propagation mode.

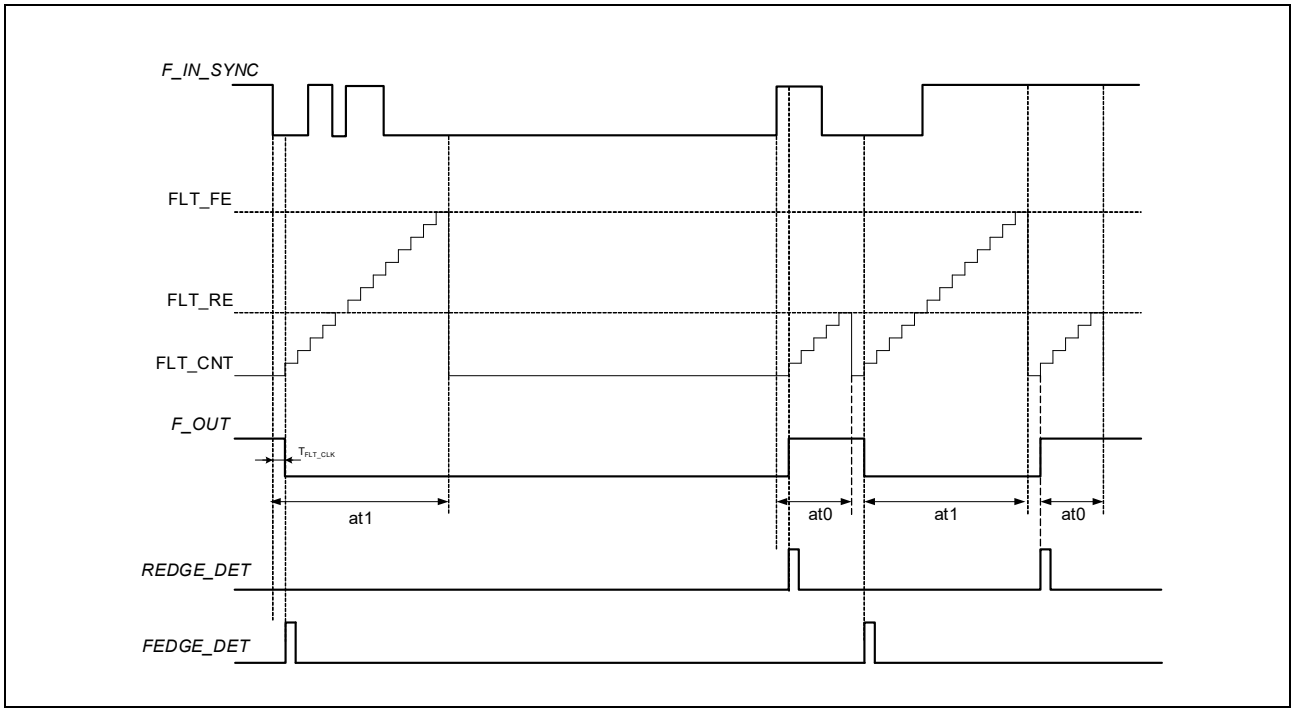


Figure 38.39 Immediate Edge Propagation Mode in the case of a rising and falling edge

If the FLT_CNT has reached the acceptance time for a specific signal edge and the signal F_IN_SYNC has already changed to the opposite level of F_OUT, the opposite signal level is set to F_OUT and the acceptance time measurement is started immediately. **Figure 38.39** shows this scenario at the detection of the first rising edge and the second falling edge.

(2) Individual de-glitch time mode (up/down counter)

In individual de-glitch time mode (up/down counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers FLT_RE and FLT_FE, respectively.

The filter counter register FLT_CNT is incremented when the signal level on F_IN_SYNC is unequal to the signal level on F_OUT and decremented if F_IN_SYNC equals F_OUT.

After FLT_CNT has reached a value of zero during decrementation the counter is stopped immediately.

If a glitch is detected a glitch detection bit GLITCHDET is set in the TIM[i]_CH[x]_IRQ_NOTIFY register.

The detected edge signal together with the new signal level is propagated to F_OUT after the individual de-glitch threshold is reached. **Figure 38.40** shows the behavior of the filter in individual de-glitch time (up/down counter) mode in the case of the rising edge detection.

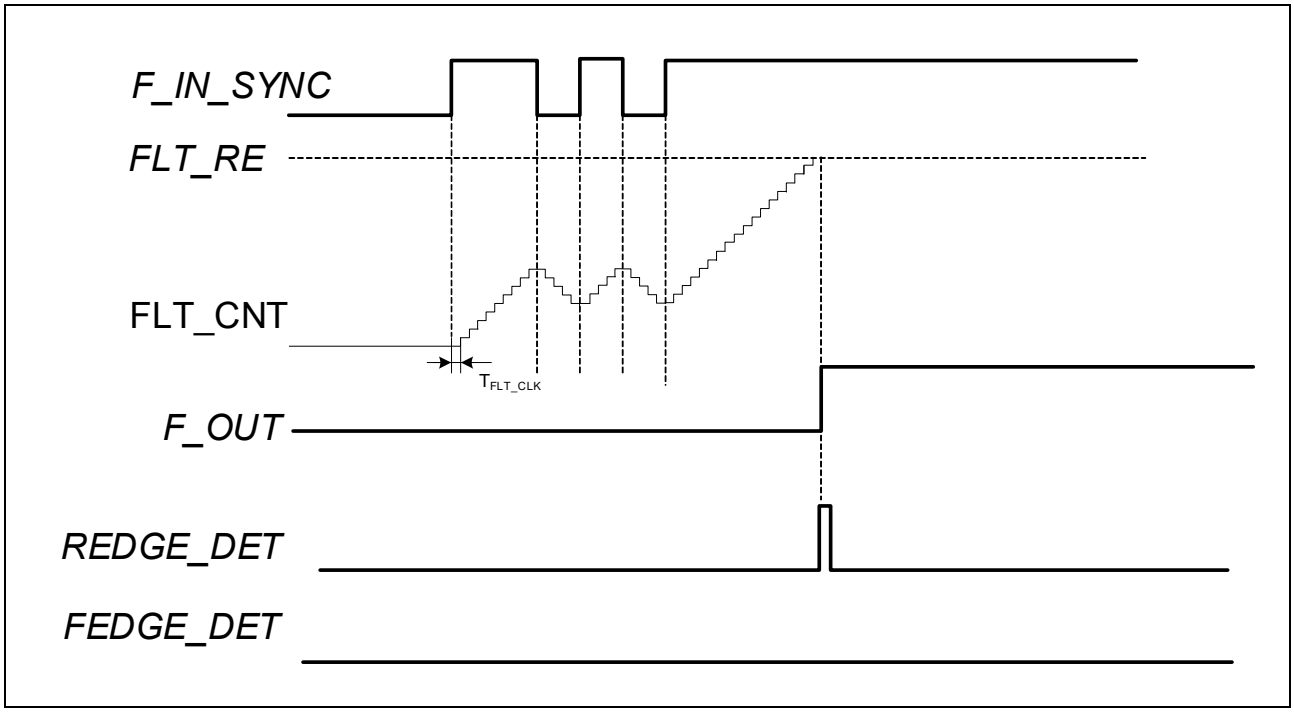


Figure 38.40 Individual De-glitch Time Mode (up/down counter) in the case of a rising edge

(3) Individual de-glitch time mode (hold counter)

In individual de-glitch time mode (hold counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers FLT_RE and FLT_FE, respectively.

The filter counter register FLT_CNT is incremented when the signal level on F_IN_SYNC is unequal to the signal level on F_OUT and the counter value of FLT_CNT is hold if F_IN equals F_OUT.

If a glitch is detected the glitch detection bit GLITCHDET is set in the TIM[i]_CH[x]_IRQ_NOTIFY register.

The detected edge signal together with the new signal level is propagated to F_OUT after the individual de-glitch threshold is reached. **Figure 38.41** shows the behavior of the filter in individual de-glitch time (hold counter) mode in the case of the rising edge detection.

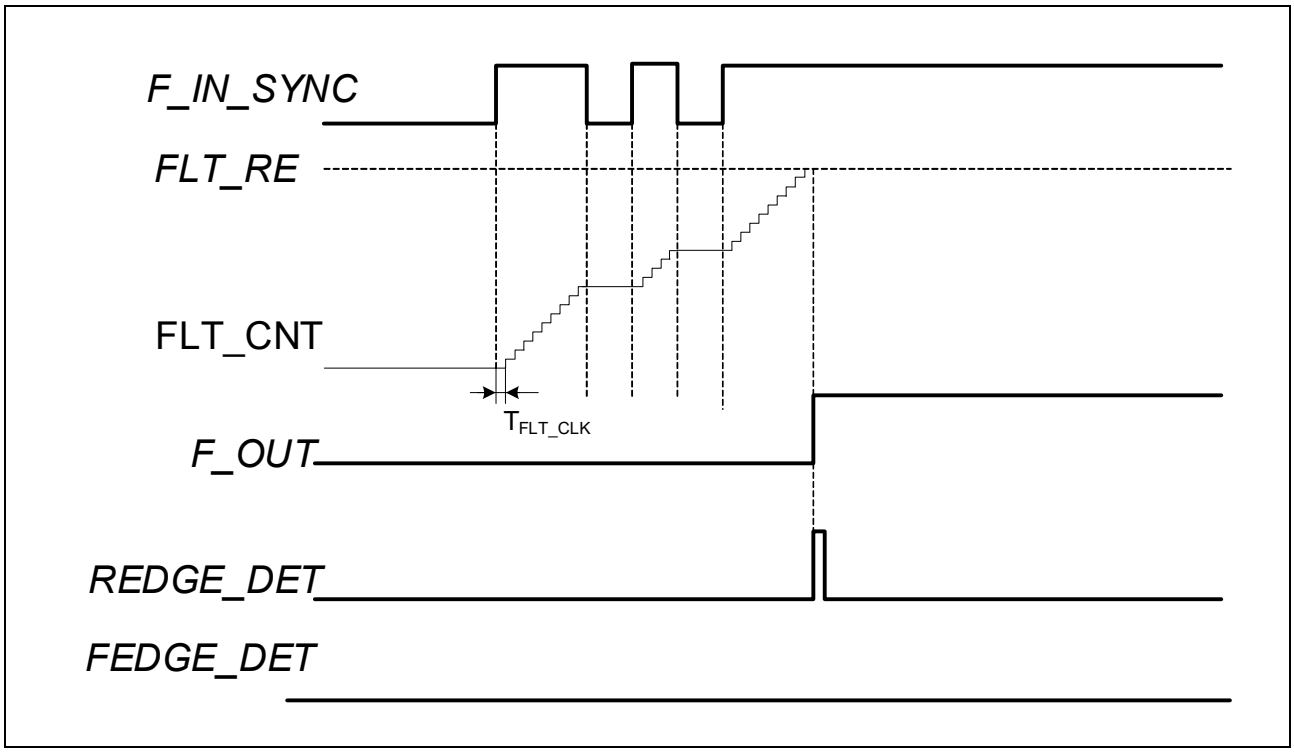


Figure 38.41 Individual De-Glitch Time Mode (hold counter) in the case of a rising edge

(4) Individual de-glitch Time Mode (reset counter)

In individual de-glitch time mode (reset counter) each edge of an input signal can be filtered with an individual de-glitch threshold filter value mentioned in the registers FLT_RE and FLT_FE, respectively.

The filter counter register FLT_CNT is incremented when the signal level on F_IN_SYNC is unequal to the signal level on F_OUT and the counter value of FLT_CNT is reset to 000000_H if F_IN equals F_OUT.

If a glitch is detected the glitch detection bit GLITCHDET is set in the TIM[i]_CH[x]_IRQ_NOTIFY register.

The detected edge signal together with the new signal level is propagated to F_OUT after the individual de-glitch threshold is reached. **Figure 38.42** shows the behavior of the filter in individual de-glitch time (reset counter) mode in the case of the rising edge detection.

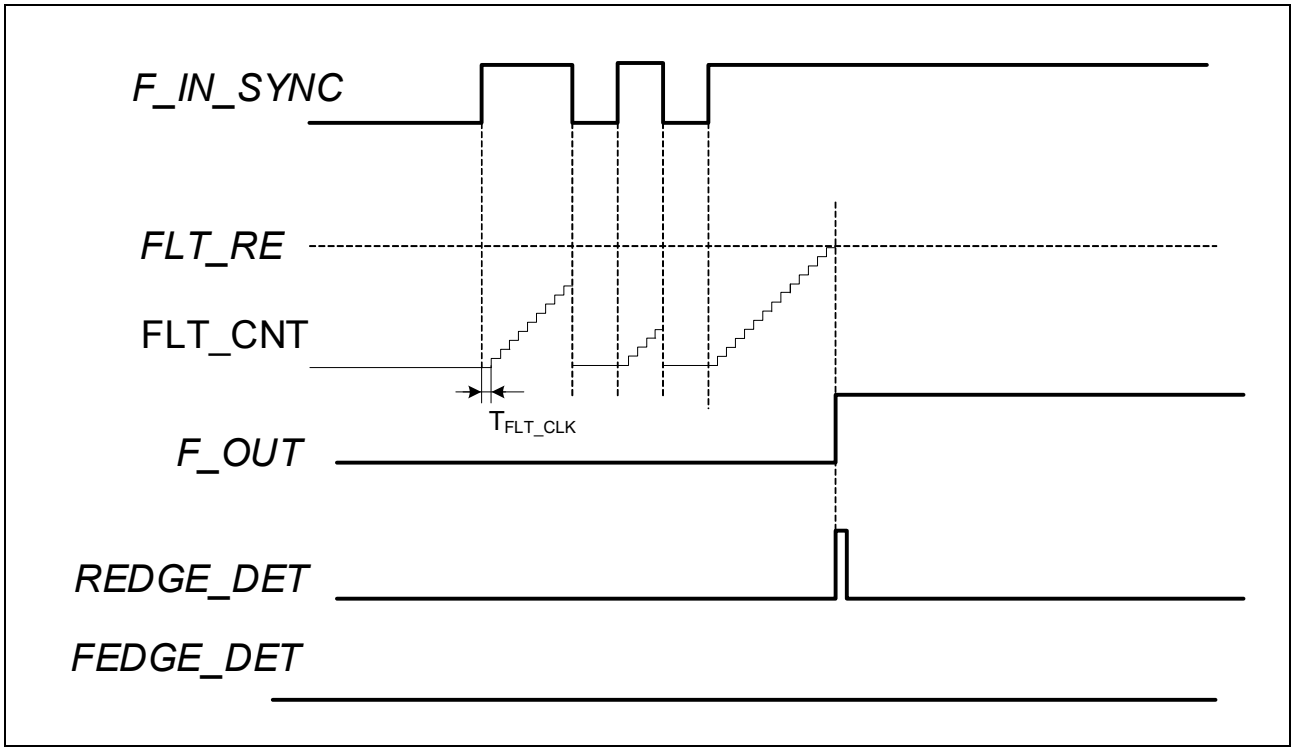


Figure 38.42 Individual De-glitch Time Mode (reset counter) in the case of a rising edge

(5) Immediate edge propagation and individual de-glitch mode

As already mentioned, the four different filter modes can be applied individually to each edge of the measured signal.

However, if one edge is configured with immediate edge propagation and the other edge with an individual deglitch mode (whether up/down counter or hold counter) a special consideration has to be applied.

Assume that the rising edge is configured for immediate edge propagation and the falling edge with individual deglitch mode (up/down counter) as shown in **Figure 38.43**.

If the falling edge of the incoming signal already occurs during the measuring of the acceptance time of the rising edge, the measurement of the deglitch time on the falling edge is started delayed, but immediately after the acceptance time measurement phase of the rising edge has finished.

Consequently, the deglitch counter can not measure the time T_{ERROR}, as shown in **Figure 38.43**.

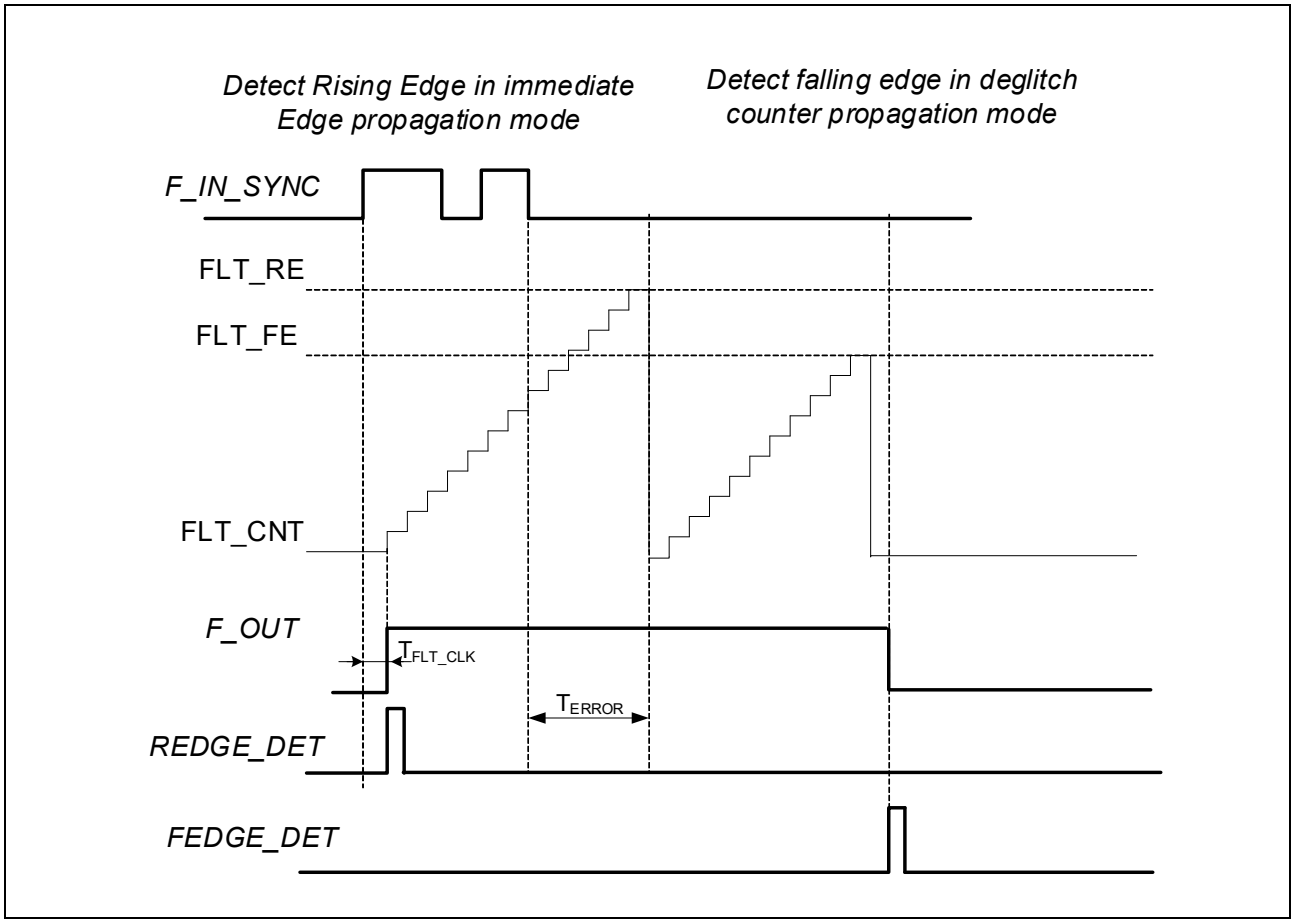


Figure 38.43 Mixed mode measurement

38.15.2.3 TIM Filter reconfiguration

If FLT_EN = 1 a change of FLT_RE or FLT_FE will take place immediately.

If FLT_EN = 1 a change of FLT_MODE_RE or FLT_MODE_FE will be used with the next occurring corresponding edge. If the mode is changed while the filter unit is processing a certain mode, it will end this edge filtering in the mode as started.

If FLT_EN = 1 a change of FLT_CTR_RE, FLT_CTR_FE, EFLT_CTR_RE or EFLT_CTR_FE will take place immediately.

38.15.3 Timeout Detection Unit (TDU)

The Timeout Detection Unit (TDU) is responsible for timeout detection of the TIM input signals.

Each channel of the TIM sub-module has its own Timeout Detection Unit (TDU) where a timeout event can be set up on the filtered input signal of the corresponding channel.

In each timeout unit exist 3 8 bit counter/comparator slices. A counter/comparator slice is shown below. The counter TO_CNT will increment by signal INC. The counter can be loaded with the value LOAD_VAL if LOAD = 1. GT_EVT will be 1 if TO_CNT > TOV is fulfilled. EQ_EVT will be 1 if TO_CNT = TOV is fulfilled.

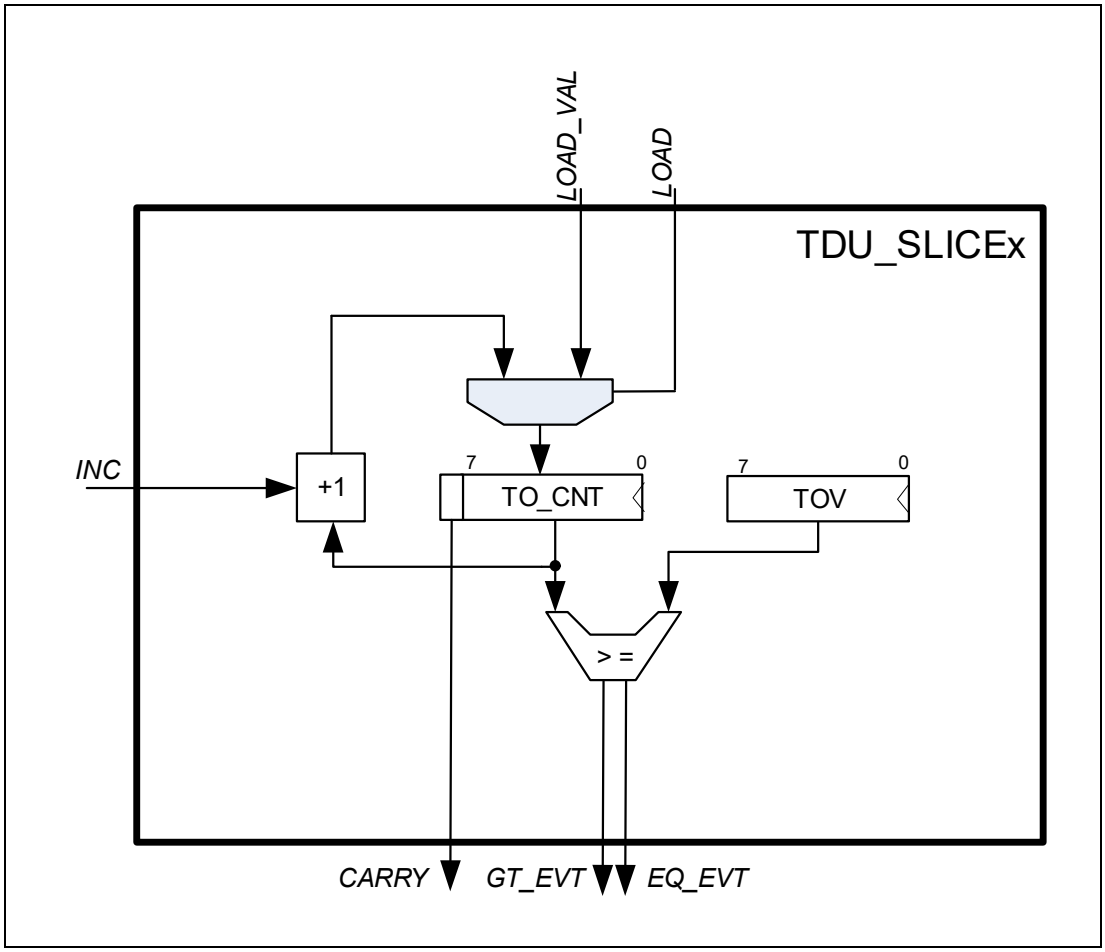


Figure 38.44 Counter/Comparator Slice

The counter/comparator slices can be cascaded depending on the application needs to operate as : 3 × 8 bit counter 1 × 16 bit counter and 1 × 8 bit counter 1 × 24 bit counter 2 × 8 bit counter. This allows the user to use the functions timeout on input signals local CMU clock prescaler 8 bit trigger event generation 8 bit (external capture, todet_irq) in parallel. With usage of the 3 × 8 bit counter it is possible to define different timeout values for the 2 signal levels.

Following table shows which functions can be used in parallel.

Table 38.142 Used Parallel Functions

Counter Type	Timeout Functionality	Generate Local TIM CMU clk	Source for External Capture to Previous Channel	Source for TODET_IRQ
24 bit	24 bit	no	tdu_timeout_evt tdu_sample_evt	tdu_timeout_evt tdu_sample_evt
1 × 8 bit 1 × 16 bit	16 bit local clk tdu_sample_evt usable	yes	tdu_timeout_evt, tdu_frame_evt, tdu_sample_evt	tdu_timeout_evt, tdu_frame_evt, tdu_sample_evt
3 × 8 bit	8 bit local clk tdu_sample_evt usable	yes	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt
3 × 8 bit	no	yes	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt
2 × 8 bit	no	no	tdu_timeout_evt, tdu_word_evt, tdu_frame_evt	tdu_timeout_evt, tdu_word_evt, tdu_frame_evt

Next table shows which of the available 8 bit resources are cascaded with a chosen SLICING.

Table 38.143 Which of the Available 8 bit Resources Are Cascaded with a Chosen SLICING (1/2)

Counter Type	Counters Count on	Counter Resource Generates	CLK Selection
24 bit	CNT on TCS	CNT= TO_CNT2 & TO_CNT1 & TO_CNT; TCMP = TOV2 & TOV1 &TOV; CNT ≥ TCMP generates tdu_sample_evt tdu_timeout_evt = tdu_sample_evt tdu_frame_evt = 0 tdu_word_evt = 0	TCS selected
3 × 8 bit	TO_CNT2 on TCS TO_CNT on tdu_sample_evt TO_CNT1 on tdu_word_evt	TO_CNT2 ≥ TOV2 generates tdu_sample_evt TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt	TO_CNT2: TCS selected TO_CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EV T= 1 TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK = 0
3 × 8 bit	TO_CNT2 on TCS TO_CNT on tdu_sample_evt TO_CNT1 on tdu_sample_evt	TO_CNT2 ≥ TOV2 generates tdu_sample_evt TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt or tdu_frame_evt	TO_CNT2: TCS selected TO_CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EVT = 1 TO_CNT1: tdu_sample_evt selected with TDU_SAME_CNT_CLK = 1
3 × 8 bit	TO_CNT2 on TCS TO_CNT on TCS TO_CNT1 on tdu_word_evt	TO_CNT2 ≥ TOV2 generates tdu_sample_evt TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt	TO_CNT2: TCS selected TO_CNT: TCS selected with TCS_USE_SAMPLE_EVT = 0 TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK = 0
3 × 8 bit	TO_CNT2 on TCS TO_CNT on TCS TO_CNT1 on TCS	TO_CNT2 ≥ TOV2 generates tdu_sample_evt TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt or tdu_frame_evt	TO_CNT2: TCS selected TO_CNT: TCS selected with TCS_USE_SAMPLE_EVT = 0 TO_CNT1: TCS selected with TDU_SAME_CNT_CLK = 1

Table 38.143 Which of the Available 8 bit Resources Are Cascaded with a Chosen SLICING (2/2)

Counter Type	Counters Count on	Counter Resource Generates	CLK Selection
2 × 8 bit	TO_CNT on TCS TO_CNT1 on tdu_word_evt	TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt tdu_sample_evt = 0	TO_CNT: TCS selected TO_CNT1: tdu_word_evt selected with TDU_SAME_CNT_CLK = 0
2 × 8 bit	TO_CNT on TCS TO_CNT1 on TCS	TO_CNT ≥ TOV generates tdu_word_evt TO_CNT1 ≥ TOV1 generates tdu_frame_evt tdu_timeout_evt = tdu_word_evt tdu_sample_evt = 0	TO_CNT: TCS selected TO_CNT1:TCS selected with TDU_SAME_CNT_CLK = 1
1 × 8 bit 1 × 16 bit	TO_CNT2 on TCS CNT on TCS	TO_CNT2 ≥ TOV2 generates tdu_sample_evt CNT = TO_CNT1 & TO_CNT; TCMP = TOV1 & TOV; CNT ≥ TCMP generates tdu_frame_evt tdu_timeout_evt = tdu_frame_evt tdu_word_evt = 0	TO_CNT2: TCS selected CNT: TCS selected with TCS_USE_SAMPLE_EVT = 0
1 × 8 bit 1 × 16 bit	TO_CNT2 on TCS CNT on tdu_sample_evt	TO_CNT2 ≥ TOV2 generates tdu_sample_evt CNT = TO_CNT1 & TO_CNT; TCMP = TOV1 & TOV; CNT ≥ TCMP generates tdu_frame_evt tdu_timeout_evt = tdu_frame_evt tdu_word_evt = 0	TO_CNT2: TCS selected CNT: tdu_sample_evt selected with TCS_USE_SAMPLE_EVT = 1

Based on a chosen counter configuration by SLICING it is possible to control the start behavior of the counters by TDU_START in multiple ways. In addition the stopping of the counters can be controlled by TDU_STOP. Depending on the application needs it can be decided how the individual counter slices can be reset/reloaded by the configuration field TDU_RESYNC.

Depending on the counter configuration, up to 4 internal compare events tdu_timeout_evt, tdu_sample_evt, tdu_word_evt, tdu_frame_evt out of the 3 comparator slices can be generated. It can be chosen by TODET_IRQ_SRC which shall be used as TIM_TODETx_IRQ signal which will be accessible by the TODET bit inside the TIM[i]_CH[x]_IRQ_NOTIFY register The TDU architecture is shown in **Figure 38.45**.

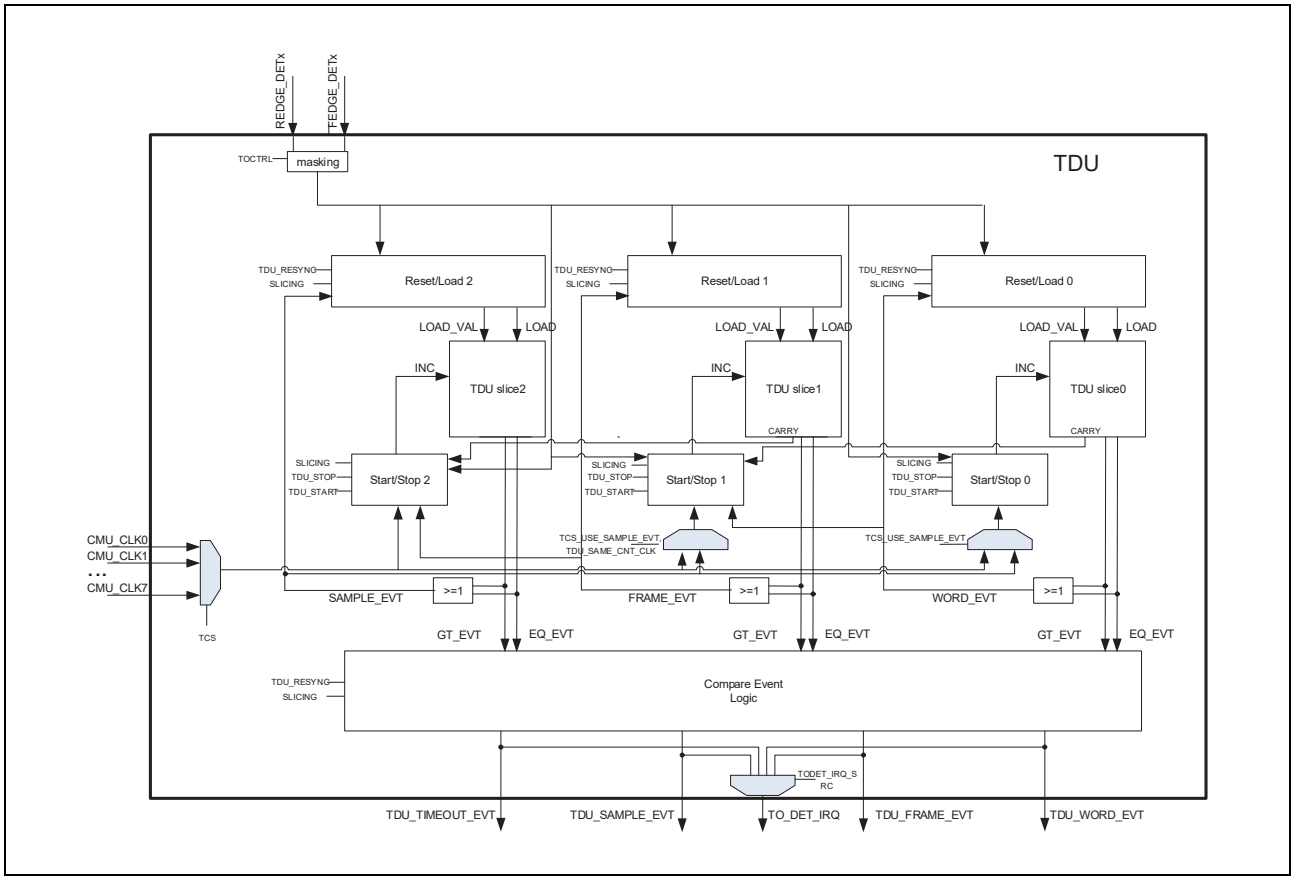


Figure 38.45 Architecture of the TDU Sub-unit

Each TDU_slice has its own start/stop control, based on the chosen configuration it will decide if the counter inside the TDU slice will increment on the resolution of the applied clock/event. The reset/load control decides based on the configuration settings and the compare result of the TDU slices those the counters TO_CNT, TO_CNT1, TO_CNT2 have to be reloaded. Depending on the chosen counter/ compare configuration the compare event logic will generate based on the compare results of the 3 TDU slices and the chosen resolution the events tdu_sample_evt, tdu_word_evt, tdu_frame_evt.

The primary resolution on which the TDU is working can be specified with the bit field TCS of the register TIM[i]_CH[x]_TDUV. The corresponding input signal CMU_CLKx will be used to clock the TDU. The individual timeout/counter values have to be specified in number of ticks of the selected input clock signal in the fields TOV, TOV1, TOV2 of the timeout value register TIM[i]_CH[x]_TDUV of the TIM channel x.

In case of cascading the bit slices by usage of SLICING and TCS_USE_SAMPLE_EVT and TDU_SAME_CLK the resolution for counting can be switched to the events tdu_sample_evt or tdu_word_evt. More details see table above.

The counter compare units start operation on occurrence of the first "start event" configured by TDU_START. They continue their operation until the first "stop event" configured by TDU_STOP occurs.

In case of occurrence of a start event and a compare/count resolution event in the same clock cycle, the counters will increment or reload/reset based on TDU_RESYNC immediately. No tdu_sample_evt, tdu_word_evt, tdu_frame_evt will be generated.

In case of occurrence of a stop event the counters will not change their values. In case of occurrence of a stop event and a compare/count resolution event in the same clock cycle the corresponding events tdu_sample_evt, tdu_word_evt, tdu_frame_evt will be generated.

In case of occurrence of a start event and a stop event in the same clock cycle the counters will not change their values. No tdu_sample_evt, tdu_word_evt, tdu_frame_evt will be generated.

The function of the timeout unit (configured to TDU_RESYNC = 0000_B, TDU_START = 000_B) can be started or stopped inside the TIM[i]_CH[x]_CTRL register by setting/resetting the TOCTRL bit.

Timeout detection can be enabled to be sensitive to falling, rising or both edges of the input signal by writing the corresponding values to the bit field TOCTRL.

The TDU generates an interrupt signal TIM_TODETx_IRQ whenever a timeout is detected for an individual input signal, and the TODET bit is set inside the TIM[i]_CH[x]_IRQ_NOTIFY register.

In addition, when the ARU access is enabled with the ARU_EN bit inside the TIM[i]_CH[x]_CTRL register, the actual values stored inside the registers TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1 are sent together with the last stored signal level to the ARU if a timeout event TDU_TIMEOUT_EVT occurs.

To signal that a timeout occurred, the ARU_OUT(50) bit (ACB(2)) is set. The bit ACB(0) will be updated with the timeout event to the signal level on which the timeout was detected. Timeout signaling with ACB(2) is only possible with TODET_IRQ_SRC = 0000_B.

Thus, a destination could determine if a timeout occurred at the TIM input by evaluating ACB bit 2.

Since the TIM channel still monitors its input pin although the timeout happened, a valid edge could occur at the input pin while the timeout information is still valid at the ARU. In that case, the new edge associated data is stored inside the registers TIM[i]_CH[x]_GPR0 and TIM[i]_CH[x]_GPR1, the GPR overflow detected bit is set together in the ACB field (ACB(1)) with the timeout bit (ACB(2)) and the values are marked as valid to the ARU.

The ACB bit 2 is cleared, when a successful ARU write access by the TIM channel took place.

The ACB bit 1 is cleared, when a successful ARU write access by the TIM channel took place.

When a valid edge initiates an ARU write access which has not ended while a new timeout occurs the GPR overflow detected bit (ACB(1)) is set. The bit ACB(0) will be updated to the level on which the timeout occurred.

When a timeout occurred and initiates an ARU write access which has not ended while a new timeout occurs the GPR overflow detected bit (ACB(1)) is not set.

The following table clarifies the meaning of the ACB Bits for valid data provided by a TIM channel:

Table 38.144 The ACB Bits for Valid Data Provided by a TIM Channel

ACB4/3	ACB2	ACB1	ACB0	Description
dc	0	0	SL	Valid edge detected
dc	0	1	SL	Input edge overwritten by subsequent edge
dc	1	0	SL	Timeout detected without valid edge
dc	1	1	SL	Timeout detected with subsequent valid edge detected

38.15.4 TIM Channel Architecture

38.15.4.1 Overview

Each TIM channel consist of an input edge counter ECNT, a Signal Measurement Unit (SMU) with a counter CNT, a counter shadow register CNTS for SMU counter and two general purpose registers GPR0 and GPR1 for value storage.

The value TOV of the timeout register TIM[i]_CH[x]_TDU is provided to TDU sub-unit of each individual channel for timeout measurement.

The architecture of the TIM channel is depicted in **Figure 38.46**.

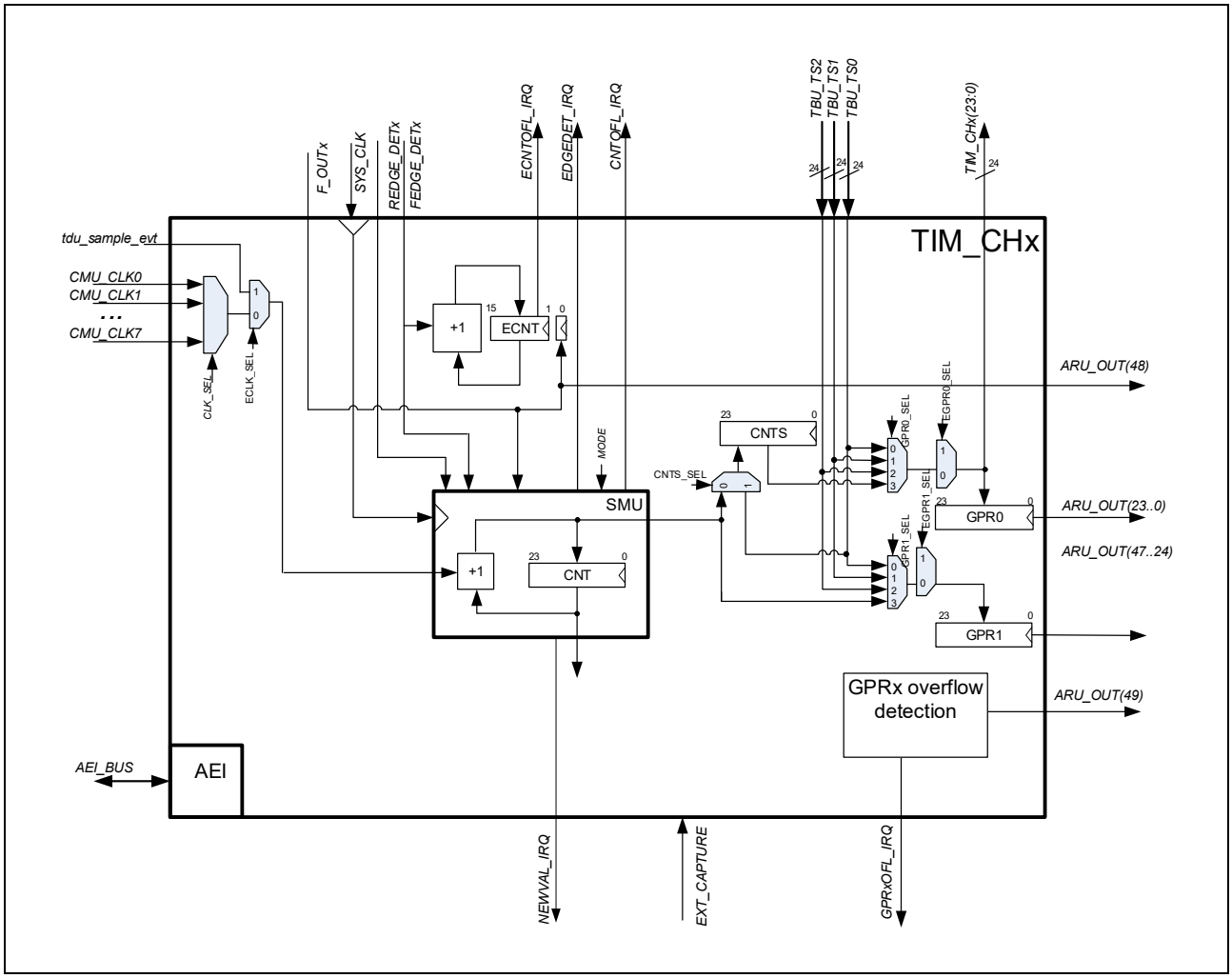


Figure 38.46 TIM Channel Architecture

Each TIM channel receives both input trigger signals REDGE_DETx and FEDGE_DETx, generated by the corresponding filter module in order to signalize a detected echo of the input signal F_INx. The signal F_OUTx shows the filtered signal of the channel's input signal F_INx.

The edge counter ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. (However, the actual counter implementation counts only falling edges on ECNT[n:1] bits. It generates ECNT by composing the ECNT[n:1] bits with F_OUTx as bit 0).

Thus, the whole ECNT counter value is always odd, when a positive edge was received and always even, when a negative edge was received.

The current ECNT[7:0] register content is made visible on the bits 31 down to 24 of the registers GPR0, GPR1, and CNTS. This allows the software to detect inconsistent read accesses to registers GPR0, GPR1, and CNTS. However, the update strategy of these registers depends on the selected TIM modes, and thus the consistency check has to be adapted carefully.

It can be chosen with the bit field FR_ECNT_OFL when an ECNT overflow is signaled on ECNTOFL. An ECNT overflow can be signaled on 8 bit or full range resolution.

While reading the register TIM[i]_CH[x]_ECNT the bit ECNT[0] shows the input signal value F_OUTx independent of the state (enabled/disabled) of the channel. If a channel gets disabled (OSM mode or resetting TIM_EN) the content of TIM[i]_CH[x]_ECNT will be frozen until a read of the register takes place. This read will reset the ECNT counter. Continuing reads will show the input signal value in bit ECNT[0] again.

When new data is written into GPR0 and GPR1 the NEWVAL bit is set in TIM[i]_CH[x]_IRQ_NOTIFY register and depending on corresponding enable bit value the NEWVALx_IRQ interrupt is raised.

Each TIM input channel has an ARU connection for providing data via the ARU to the other GTM sub-modules. The data provided to the ARU depends on the TIM channel mode and its corresponding adjustments (e.g. multiplexer configuration).

The bit ARU_EN of register TIM[i]_CH[x]_CTRL decides, whether the measurement results of registers GPR0 and GPR1 are consumed by another sub-module via ARU (ARU_EN = 1) or the CPU via AEI (ARU_EN = 0).

To guarantee a consistent delivery of data from the GPR0 and GPR1 registers to the ARU or the CPU each TIM channel has to ensure that the data is consumed before it is overwritten with new values.

If new data was produced by the TIM channel (bit NEWVAL is set inside TIM[i]_CH[x]_IRQ_NOTIFY register) while the old data is not consumed by the ARU (ARU_EN = 1) or CPU (ARU_EN = 0), the TIM channel sets the GPROFL bit inside the status register TIM[i]_CH[x]_IRQ_NOTIFY and it overwrites the data inside the registers GPR0 and GPR1. In addition when ARU_EN = 1 the bit ACB(1) is set to 1 to indicate the overflow in the ARU data.

If the CPU is selected as consumer for the registers GPR0 and GPR1 (ARU_EN = 0), the acknowledge for reading out data is performed by a read access to the register GPR0. Thus, register GPR1 should be read always before GPR0.

If the ARU is selected as consumer for the registers GPR0 and GPR1 (ARU_EN = 1), the acknowledge for reading out data is performed by the ARU itself. However, the registers GPR0 and GPR1 could be read by CPU without giving an acknowledge.

38.15.4.2 TIM Channel Modes

The TIM provides seven different measurement modes that can be configured with the bit field TIM_MODE of register TIM[i]_CH[x]_CTRL.

The measurement modes are described in the following subsections. Besides these different basic measurement modes, there exist distinct configuration bits in the register TIM[i]_CH[x]_CTRL for a more detailed controlling of each mode. The meanings of these bits are as follows:

- DSL: control the signal level for the measurement modes (e.g. if a measurement is started with rising edge or falling edge, or if high level pulses or low level pulses are measured.
- EGPR0_SEL, GPR0_SEL and EGPR1_SEL, GPR1_SEL: control the actual content of the registers GPR0 and GPR1 after a measurement has finished.
- CNTS_SEL: control the content of the registers CNTS. The actual time for updating the CNTS register is mode dependent.
- OSM: activate measurement in one-shot mode or continuous mode. In one-shot mode only one measurement cycle is performed and after that the channel is disabled.
- NEWVAL: The NEWVAL IRQ interrupt is triggered at the end of a measurement cycle, signaling that the registers GPR0 and GPR1 are updated.
- ARU_EN: enables sending of the registers GPR0 and GPR1 together with the actual signal level (in bit 48) and the overflow signal GPROFL (in bit 49), and the timeout status information (bit 50) to the ARU.
- EXT_CAP_EN: forces an update of the registers GPR0 and GPR1 and CNTS (TIM channel mode dependent) only on each rising edge of the EXT_CAPTURE signal and triggers a NEWVAL IRQ interrupt. If this mode is disabled the NEWVAL IRQ interrupt is triggered at the end of each measurement cycle.

For each channel the source of the EXT_CAPTURE signal can be configured with the bit fields EXT_CAP_SRC in the register TIM[i]_CH[x]_ECTRL.

(1) TIM PWM Measurement Mode (TPWM)

In TIM PWM Measurement Mode the TIM channel measures duty cycle and period of an incoming PWM signal. The DSL bit defines the polarity of the PWM signal to be measured.

When measurement of pulse high time and period is requested (PWM with a high level duty cycle, DSL = 1) and IMM_START = 0, the channel starts measuring after the first rising edge is detected by the filter.

If IMM_START = 1 the measurement starts immediately after activating the channel by TIM_EN = 1.

Measurement is done with the CNT register counting with the configured clock coming from CMU_CLKx until a falling edge is detected.

Assume: SWAP_CAPTURE =0, ECNT_RESET = 0

Then the counter value is stored inside the shadow register CNTS (if CNTS_SEL = 0) and the counter CNT counts continuously until the next rising edge is reached.

On this following rising edge the content of the CNTS register is transferred to GPR0 and the content of CNT register is transferred to GPR1, assuming settings for the selectors EGPR0_SEL =0, GPR0_SEL = 11 and EGPR1_SEL = 0, GPR1_SEL = 11. By this, GPR0 contains the duty cycle length and GPR1 contains the period.

NOTE

The bits 1 to 7 of the ECNT may be used to check data consistency of the registers GPR0 and GPR1.

In addition the CNT register is cleared NEWVAL status bit inside of TIM[i]_CH[x]_IRQ_NOTIFY status register and depending on corresponding interrupt enable condition TIM_NEWVALx_IRQ interrupt is raised.

The CNTS register update is not performed until the measurement is started. Afterwards each edge leaving the level defined by DSL is performing a CNTS register update.

If a PWM with a low level duty cycle should be measured (DSL = 0) and IMM_START = 0, the channel waits for a falling edge until measurement is started. On this edge the low level duty cycle time is stored first in CNTS and then finally in GPR0 and the period is stored in GPR1.

When a PWM period was successfully measured, the data in the registers GPR0 and GPR1 is marked as valid for reading by the ARU when the ARU_EN bit is set inside TIM[i]_CH[x]_CTRL register, the NEWVAL bit is set inside the TIM[i]_CH[x]_IRQ_NOTIFY register, and a new measurement is started.

If the preceding PWM values were not consumed by a reader attached to the ARU (ARU_EN bit enabled) or by the CPU the TIM channel set GPROFL status bit in TIM[i]_CH[x]_IRQ_NOTIFY and depending on corresponding interrupt enable bit value raises a GPROFL_IRQ and overwrites the old values in GPR0 and GPR1. A new measurement is started afterwards.

If the register CNT produces an overflow during the measurement, the bit CNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_CNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

If the register ECNT produces an overflow during the measurement, the bit ECNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_ECNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

If ECNT_RESET = 0 the counter CNT will be reset to 0 on active edge (defined by DSL) of the input signal. If ECNT_RESET = 1 the counter CNT will be reset to 0 on each edge of the input signal.

Assume EXT_CAP_EN = 0 and SWAP_CAPTURE = 0: On every input edge to the active level defined by DSL will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1. Every edge to the inactive level will capture the data selected by CNTS_SEL to the registers CNTS.

Assume EXT_CAP_EN = 0 and SWAP_CAPTURE = 1: On every input edge to the inactive level defined by DSL will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1. Every edge to the active level will capture the data selected by CNTS_SEL to the registers CNTS.

(a) External capture TIM PWM Measurement Mode (TPWM)

If external capture is enabled EXT_CAP_EN = 1, the pwm measurement is done continuously. The actual measurement values are captured to GPRx if an external capture event occurs.

On every external capture event the data selected by CNTS_SEL, EGPR0_SEL, GPR0_SEL will be captured to the registers CNTS, GPR0.

If SWAP_CAPTURE = 0 every external capture event will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1. Every input edge to the level ! = DSL will capture the data selected by CNTS_SEL to the registers CNTS.

If SWAP_CAPTURE = 1 every input edge to the inactive level ! = DSL will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1.

Assume SWAP_CAPTURE = 0:

Operation is done depending on CMU clock, ISL, DSL bit and the input signal value defined in next table (Assume CNTS_SEL= 0):

Table 38.145 Operation is done depending on CMU clock, ISL, DSL bit and the input signal value defined in next table (Assume CNTS_SEL= 0)

Input signal F_OUTx	Selected CMU Clock	External capture	ISL	DSL	Action description
0	1	0	—	0	CNT++
1	1	0	—	0	no
Rising edge	—	0	0	0	capture CNT value in CNTS
Falling edge	—	0	0	0	CNT = 0
Rising edge	—	0	1	0	no
Falling edge	—	0	1	0	capture CNT value in CNTS; CNT = 0
1	1	0	—	1	CNT++
0	1	0	—	1	no
Falling edge	—	0	0	1	capture CNT value in CNTS
Rising edge	—	0	0	1	CNT = 0
Falling edge	—	0	1	1	no
Rising edge	—	0	1	1	capture CNT value in CNTS; CNT = 0
—	—	Rising edge	—	—	do GPRx capture; issue NEWVAL_IRQ
—	0	0	—	—	no

The CNTS register update is not performed until the measurement is started (first edge defined by DSL is detected). Afterwards the update of the CNTS register is defined by ISL, DSL combinations in the table above.

(2) TIM Pulse Integration Mode (TPIM)

In TIM Pulse Integration Mode each TIM channel is able to measure a sum of pulse high or low times on an input signal, depending on the selected signal level bit DSL of register TIM[i]_CH[x]_CTRL register.

If IMM_START = 0 the pulse integration measurement is started with occurrence of the first edge defined by DSL on the input signal. If IMM_START = 1 the measurement starts immediately after activating the channel by TIM_EN = 1.

The pulse times are measured by incrementing the TIM channel counter CNT until the counter is stopped with occurrence of a input signal edge to the opposite signal level defined by DSL.

The counter CNT counts with the CMU_CLKx clock specified by the CLK_SEL bit field of the TIM[i]_CH[x]_CTRL register.

The CNT register is reset at the time the channel is activated (enabling via AEI write access) and it accumulates pulses while the channel is staying enabled.

Assume EXT_CAP_EN = 0 and SWAP_CAPTURE = 0: After measurement is started, every falling (DSL = 1) or rising (DSL = 0) input edge will issue a TIM_NEWVALx_IRQ interrupt, and the registers CNTS, GPR0 and GPR1 are updated according to settings of its corresponding input multiplexers, using the bits EGPR0_SEL, EGPR1_SEL, GPR0_SEL, GPR1_SEL and CNTS_SEL.

NOTE

The bits 1 to 7 of the ECNT may be used to check data consistency of the registers GPR0 and GPR1.

Assume EXT_CAP_EN = 0 and SWAP_CAPTURE = 1: After measurement is started, every falling (DSL-1) or rising (DSL = 0) input edge will issue a TIM_NEWVALx_IRQ interrupt, and the registers CNTS, GPR0 are updated according to settings of its corresponding input multiplexers, using the bits EGPR0_SEL, GPR0_SEL and CNTS_SEL. Every input edge to the active level defined by DSL (rising DSL = 1; falling DSL = 0) will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1.

When the ARU_EN bit is set inside the TIM[i]_CH[x]_CTRL register the measurement results of the registers GPR0 and GPR1 can be send to subsequent sub-modules attached to the ARU.

(a) External capture TIM Pulse Integration Mode (TPIM)

If external capture is enabled EXT_CAP_EN = 1, the pulse integration is done until next external capture event occurs.

On every external capture event the data selected by CNTS_SEL, EGPR0_SEL, GPR0_SEL will be captured to the registers CNTS, GPR0.

If SWAP_CAPTURE = 0 every external capture event will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1.

If SWAP_CAPTURE = 1 every input edge to the inactive level ! = DSL will capture the data selected by EGPR1_SEL, GPR1_SEL to the registers GPR1.

Assume SWAP_CAPTURE = 0; IMM_START = 0:

Operation is done depending on CMU clock, DSL bit and the input signal value defined in next table (inc_cnt = false if TIM channel is enabled):

Table 38.146 Operation Depending on CMU Clock, DSL and the Input Signal Value (inc_cnt = false if TIM Channel Is Enabled)

Input signal F_OUTx	Selected CMU Clock	External capture	ISL	DSL	Action description
Falling edge	—	0	—	0	inc_cnt = true
Rising edge	—	0	—	0	if inc_cnt = true then { do capture GPRx, CNTS; issue NEWVAL_IRQ } inc_cnt = false
Rising edge	—	0	—	1	inc_cnt = true
Falling edge	—	0	—	1	if inc_cnt = true then { do capture GPRx, CNTS; issue NEWVAL_IRQ } inc_cnt = false
—	1	0	—	—	if inc_cnt = true then CNT++;
—	—	Rising edge	—	—	do capture GPRx, CNTS; issue NEWVAL_IRQ; CNT=0
—	0	0	—	—	no

(3) TIM Input Event Mode (TIEM)

In TIM Input Event Mode the TIM channel is able to count edges.

It is configurable if rising, falling or both edges should be counted. This can be done with the bit fields DSL and ISL in TIM[i]_CH[x]_CTRL register.

In addition, a TIM[i]_NEWVAL[x]_IRQ interrupt is raised when the configured edge was received and this interrupt was enabled.

The counter register CNT is used to count the number of edges, and the bit fields EGPR0_SE L, EGPR1_SE L, GPR0_SEL, GPR1_SEL, and CNTS_SEL can be used to configure the desired update values for the registers GPR0, GPR1 and CNTS. These register are updated whenever the edge counter CNT is incremented due to the arrival of a desired edge.

If the preceding data was not consumed by a reader attached to the ARU or by the CPU the TIM channel sets GPROFL status bit and raises a GPROFL[x]_IRQ if it was enabled in TIM[i]_CH[x]_IRQ_EN register and overwrites the old values in GPR0 and GPR1 with the new ones.

If the register CNT produces an overflow during the measurement, the bit CNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_CNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

If the register ECNT produces an overflow during the measurement, the bit ECNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_ECNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

The TIM Input Event Mode does not depend on the bit field CLK_SEL of register TIM[i]_CH[x]_CTRL.

(a) External capture TIM Input Event Mode (TIEM)

If external capture is enabled, capturing is done depending on the DSL, ISL bit and the input signal value defined in next table:

Table 38.147 Capturing Depended on the DSL, ISL and the Input Signal Value, If External Capture Is Enabled

Input signal F_OUTx	External capture	ISL	DSL	Action description
—	Rising edge	1	—	do capture; issue NEWVAL_IRQ; CNT++
—	0	1	—	no
1	Rising edge	0	1	do capture; issue NEWVAL_IRQ; CNT++
0	—	0	1	no
0	Rising edge	0	0	do capture; issue NEWVAL_IRQ; CNT++
1	—	0	0	no

(4) TIM Input Prescaler Mode (TIPM)

In the TIM Input Prescaler Mode the number of edges which should be detected before a TIM[i]_NEWVAL[x]_IRQ is raised is programmable.

In this mode it must be specified in the CNTS register after how many edges the interrupt has to be raised.

A value of 0 in CNTS means that after one edge an interrupt is raised and a value of 1 means that after two edges an interrupt is raised, and so on.

The edges to be counted can be selected by the bit fields DSL and ISL of register TIM[i]_CH[x]_CTRL.

With each triggered interrupt, the registers GPR0 and GPR1 are updated according to bits EGPR0_SEL, EGPR1_SEL, GPR0_SEL and GPR1_SEL.

If the register ECNT produces an overflow during the measurement, the bit ECNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_ECNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

The TIM Input Prescaler Mode does not depend on the bit field CLK_SEL of register TIM[i]_CH[x]_CTRL.

(a) External capture TIM Input Prescaler Mode (TIPM)

If external capture is enabled, the external capture events are counted instead of the input signal edges.

Operation is done depending on the external capture signal, DSL, ISL bit and the input signal value defined in next table:

Table 38.148 Operation Depending on the External Capture Signal, DSL, ISL and the Input Signal Value

Input signal F_OUTx	External capture	ISL	DSL	Action description
—	Rising edge	1	—	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
—	0	1	—	no
1	Rising edge	0	1	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	—	0	1	no
0	Rising edge	0	0	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
1	—	0	0	no

(5) TIM Bit Compression Mode (TBCM)

The TIM Bit Compression Mode can be used to combine all filtered input signals of a TIM sub-module to a parallel m bit data word, which can be routed to the ARU, where m is the number of channels available in the TIM sub-module.

Figure 38.47 gives an overview of the TIM bit compression mode.

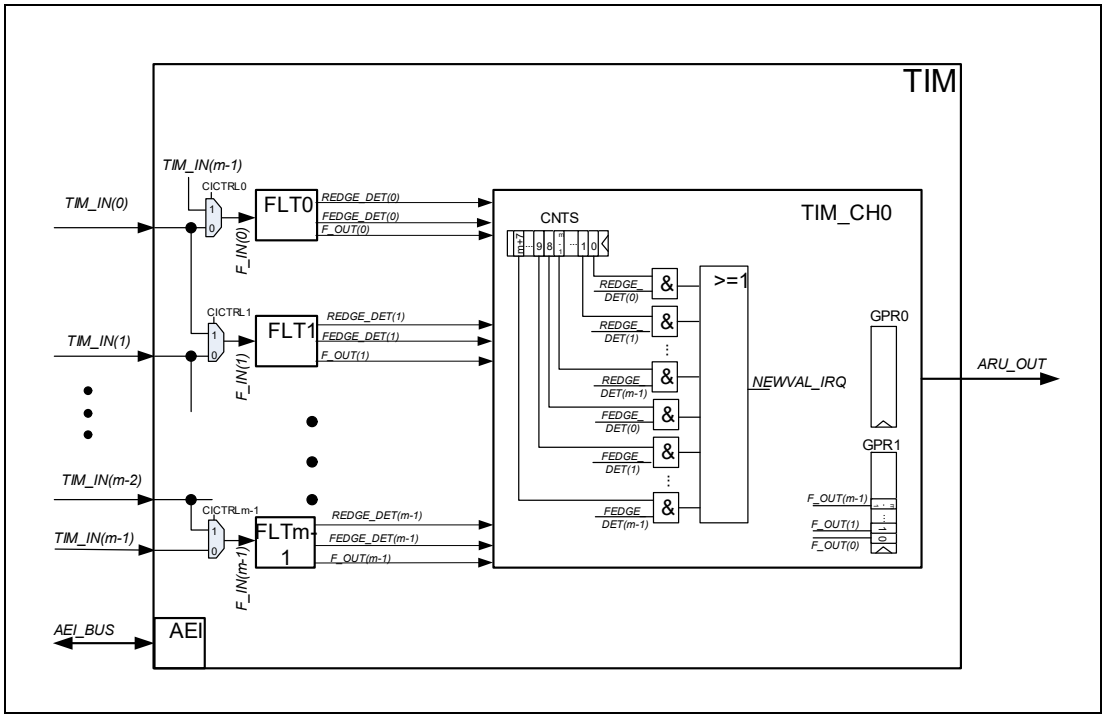


Figure 38.47 TIM Bit Compression Mode

The register CNTS of a channel is used to configure the event that releases the NEWVAL_IRQ and samples the input signals F_IN(0) to F_IN(m-1) in ascending order as a parallel data word in GPR1.

The bits 0 to m-1 of the CNTS register are used to select the REDGE_DET signals of the TIM filters 0 to m-1 as a sampling event, and the bits 8 to (7+ m) are used to select the FEDGE_DET signals of the TIM filters 0 to m-1, respectively. If multiple events are selected, the events are OR-combined (see also **Figure 38.47**).

EGPR0_SE L, GPR0_SEL selects the timestamp value, which is routed through the ARU. GPR1_SEL is not applicable in TBCM mode.

If the bit ARU_EN of register TIM[i]_CH[x]_CTRL is set, the sampled data of register GPR1 is routed together with a time stamp of register GPR0 to the ARU, whenever the NEWVAL_IRQ is released.

In TIM Bit compression mode, the register ECNT increments with each NEWVAL_IRQ, which means that the value of ECNT may depend on all m input signals. Consequently, the LSB of ECNT does not reflect the actual level of the input signal TIM_IN(x).

If the register ECNT produces an overflow during the measurement, the bit ECNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_ECNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

The TIM Bit Compression Mode does not depend on the bit field CLK_SEL of register TIM[i]_CH[x]_CTRL.

(a) External capture Bit Compression Mode (TBCM)

If external capture is enabled, capturing is done depending on the DSL, ISL bit and the input signal value defined in next table:

Table 38.149 Capturing Depended on the DSL, ISL and the Input Signal Value, If External Capture Is Enabled

Input signal F_OUTx	External capture	ISL	DSL	Action description
—	Rising edge	1	—	do capture; issue NEWVAL_IRQ; CNT++
—	0	1	—	no
1	Rising edge	0	1	do capture; issue NEWVAL_IRQ; CNT++
0	—	0	1	no
0	Rising edge	0	0	do capture; issue NEWVAL_IRQ; CNT++
1	—	0	0	no

(6) TIM Gated Periodic Sampling Mode (TGPS)

In the TIM Gated Periodic Sampling Mode the number of CMU clock cycles which should elapse before capturing and raising TIM[i]_NEWVAL[x]_IRQ is programmable. In this mode it must be specified in the CNTS register after how many CMU clock cycles the interrupt has to be raised.

A value of 0 in TIM[i]_CH[x]_CNTS means that after one CLK_SEL edge a trigger/interrupt is raised, and a value of 1 means that after two edges a trigger/interrupt is raised, and so on.

In the TIM[i]_CH[x]_CNT register the elapsed cycles were incremented and compared against TIM[i]_CH[x]_CNTS. If TIM[i]_CH[x]_CNT is greater or equal to TIM[i]_CH[x]_CNTS a trigger will be raised. This allows by writing a value to TIM[i]_CH[x]_CNTS that the actual period time can be changed on the fly.

Operation is done depending on CMU clock, DSL, ISL bit and the input signal value defined in next table:

Table 38.150 Operation depending on CMU clock, DSL, ISL and the input signal value

Input signal F_OUTx	Selected CMU Clock	External capture	ISL	DSL	Action description
—	1	0	1	—	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	0	0	0	1	no
1	1	0	0	1	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	0	—	0	1	no
0	1	0	0	0	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
1	0	0	0	0	no
—	0	0	—	—	no

In this mode the TIM[i]_CH[x]_GPR1 operates as a shadow register for TIM[i]_CH[x]_CNTS. This would allow that the period for the next sampling period could be specified. The update of TIM[i]_CH[x]_CNTS will only take place once on a trigger if the TIM[i]_CH[x]_GPR1 was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from TIM[i]_CH[x]_GPR1 and afterwards the new sampling period for the next sampling period (the one after the actual sampling period) could be written.

With each triggered interrupt, the registers GPR0 and GPR1 are updated according to bits GPR0_SEL, GPR1_SEL, EGPR0_SEL and EGPR1_SEL.

When selecting ECNT as a source for the capture registers, GPRx will show the edge count and the input signal value at point of capture.

Selecting GPR0_SEL = '11' and EGPR0_SEL = '0' for TIM channel 0 all 8 TIM input signals will be captured to GPR0[7:0].

In the TGPS Mode the bit field CLK_SEL of register TIM[i]_CH[x]_CTRL will define the selected CMU clock which will be used.

The behavior of the ECNT counter is configurable by ECNT_RESET. If set to 1 on each interrupt (period expired) the ECNT will be reset.

Otherwise it operates in wrap around mode.

If the register ECNT produces an overflow during the measurement, the bit ECNTOFL is set inside the register TIM[i]_CH[x]_IRQ_NOTIFY and interrupt TIM_ECNTOFL[x]_IRQ is raised depending on corresponding interrupt enable condition.

(a) External capture TIM Gated Periodic Sampling Mode (TGPS)

If external capture is enabled, the external capture events will capture the GPRx, reset the counter CNT and issue a NEWVAL_IRQ.

Operation is done depending on the CMU clock, external capture signal, DSL, ISL bit and the input signal value defined in next table:

Table 38.151 Operation depending on the CMU clock, external capture signal, DSL, ISL and the input signal value

Input signal F_OUTx	Selected CMU Clock	External capture	ISL	DSL	Action description
—	1	0	1	—	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	0	0	0	1	no
1	1	0	0	1	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
0	0	—	0	1	no
0	1	0	0	0	if CNT ≥ CNTS then do capture; issue NEWVAL_IRQ; CNT=0 else CNT++ endif
1	0	0	0	0	no
—	0	0	—	—	no
—	—	Rising edge	—	—	do capture; issue NEWVAL_IRQ; CNT =0

(7) TIM Serial Shift Mode (TSSM)

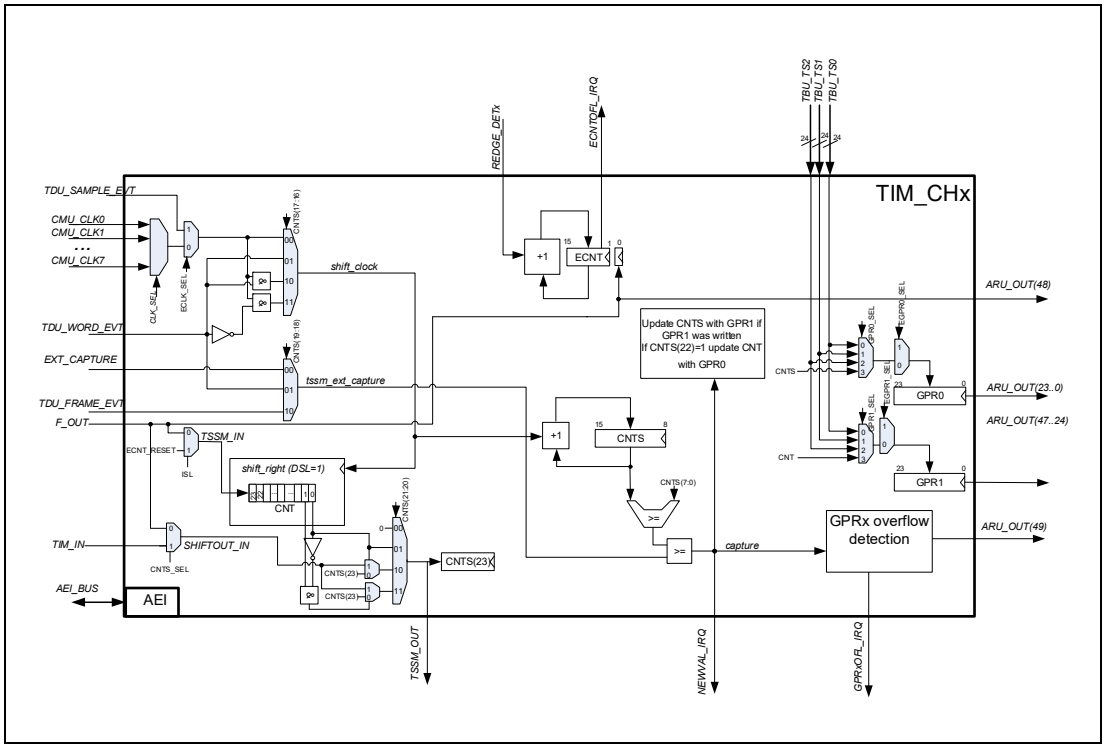


Figure 38.48 TIM Serial Shift Mode

In the TIM Serial Shift Mode on each shift clock event the actual value of the input signal TSSM_INx will be registered in dependence of DSL in the register TIM[i]_CH[x]_CNT. If ISL = 0 is set FOUTx will be used as shift in value TSSM_INx, with ISL = 1 the bit field ECNT_RESET defines the value for TSSM_INx. With DSL = 0 TSSM_INx will be stored in TIM[i]_CH[x]_CNT[0] and TIM[i]_CH[x]_CNT[22:0] will be shifted left.

With DSL = 1 TSSM_OUTx will be stored in TIM[i]_CH[x]_CNT[23] and TIM[i]_CH[x]_CNT[23:1] will be shifted right. Operation is done depending on the shift clock, external capture signal, DSL, ISL bit and the input signal value defined in next table:

Table 38.152 Operation Depending on the Shift Clock, External Capture Signal, DSL, ISL and the Input Signal Value (1/2)

Input signal TSSM_INx	Shift clock	External capture	ISL	DSL	Action description
—	0	0	—	—	no
—	—	1	—	—	if EXT_CAP_EN=1 then see function table in next chapter else no endif
value	1	0	0	0	CNT[23:1]= CNT[22:0]; CNT[0] =value if CNTS[15:8] ≥ CNTS[7:0] then do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 else CNTS[15:8]++ endif

Table 38.152 Operation Depending on the Shift Clock, External Capture Signal, DSL, ISL and the Input Signal Value (2/2)

Input signal TSSM_INx	Shift clock	External capture	ISL	DSL	Action description
value	1	0	0	1	CNT[22:0]= CNT[23:1]; CNT[23]=value if CNTS[15:8] ≥ CNTS[7:0] then do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 else CNTS[15:8]++ endif
value	1	0	1	0	CNT[23:1]= CNT[22:0]; CNT[0]= value if CNTS[15:8] ≥ CNTS[7:0] then do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET else CNTS[15:8]++ endif
value	1	0	1	1	CNT[22:0]= CNT[23:1]; CNT[23]= value if CNTS[15:8] ≥ CNTS[7:0] then do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET else CNTS[15:8]++ endif

The register TIM[i]_CH[x]_CNTS[7:0] define the amount of bits which will be stored inside TIM[i]_CH[x]_CNT.

Each shift clock will increment the register TIM[i]_CH[x]_CNTS[15:8]. If the condition TIM[i]_CH[x]_CNTS[15:8] ≥ TIM[i]_CH[x]_CNTS[7:0] is met a capture event is raised and TIM[i]_NEWVAL[x]_IRQ is asserted.

With each capture event the registers GPR0 and GPR1 are updated according to bits GPR0_SEL, GPR1_SEL, EGPR0_SEL and EGPR1_SEL.

If the bit field ISL is set to 1 the register bits TIM[i]_CH[x]_CNT are set to the value defined by ECNT_RESET in case of a capture event.

In this mode the TIM[i]_CH[x]_GPR1 operates as a shadow register for TIM[i]_CH[x]_CNTS. This allows that the amount of bits to sample can be specified. The update of TIM[i]_CH[x]_CNTS will only take place once on a trigger if the TIM[i]_CH[x]_GPR1 was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from TIM[i]_CH[x]_GPR1 and afterwards the new amount of bits to sample for the next sampling period (the one after the actual sampling period) could be written.

The shift clock which will be in use is selectable by TIM[i]_CH[x]_CNTS[17:16] : 00_B : source selection by USE_TDU_CLK_SRC is in use. It can be set to any CMU_CLK source or to the local TDU sample clock tdu_sample_evt. 01_B : the tdu_word_evt signal will be used as shift clock source. 10_B : the clk source selected by USE_TDU_CLK_SRC is used and gated with tdu_word_evt. If tdu_word_evt=0 then shift clock will be 0. 11_B : the clk source selected by USE_TDU_CLK_SRC is used and gated with tdu_word_evt. If tdu_word_evt=1 then shift clock will be 0.

(a) Signal Generation with TIM Serial Shift Mode

If TIM[i]_CH[x]_CNTS[22] is 1 the TIM[i]_CH[x]_GPR0 operates as a shadow register for TIM[i]_CH[x]_CNT. This allows that the bits for shifting out can be specified. The update of TIM[i]_CH[x]_CNT will only take place once on a trigger if the TIM[i]_CH[x]_GPR0 was written by the CPU. This means that the captured value from the previous trigger can be read by the CPU from TIM[i]_CH[x]_GPR0 and afterwards the new bits to shift out could be written.

In addition the TIM Serial Shift Mode is able to generate a signal TSSM_OUT which can be used internally to the TIM channel. On each system clock the value for TSSM_OUT is generated as defined next. The actual value can be read by the register bit TIM[i]_CH[x]_CNTS[23].

Following functionality for TSSM_OUTx is selectable by TIM[i]_CH[x]_CNTS[21:20]: 00_B: Constant output; TSSM_OUTx = 0. 01_B : Shift output; If DSL = 0 (shift left) then TSSM_OUTx = TIM[i]_CH[x]_CNT[23] else (shift right) TSSM_OUTx = TIM[i]_CH[x]_CNT[0]. 10_B: Latched output; If DSL = 0 and TIM[i]_CH[x]_CNT[23] =1 then TSSM_OUTx = SHIFTOUT_INx elsif DSL = 1 and TIM[i]_CH[x]_CNT[0] =1 then TSSM_OUTx = SHIFTOUT_INx. 10_B: Registered output; If DSL = 0 and TIM[i]_CH[x]_CNT[23:22] = 01_B then TSSM_OUTx = SHIFTOUT_INx elsif DSL =1 and TIM[i]_CH[x]_CNT[1:0] = 10_B then TSSM_OUTx = SHIFTOUT_INx. In case of registered or latched output mode the signal

SHIFTOUT_INx is selectable by CNTS_SEL. If CNTS_SEL = 0 is set FOUTx will be used for SHIFTOUT_INx, with CNTS_SEL = 1 the signal TIM_INx is in us for SHIFTOUT_INx.

(b) External capture TIM Serial Shift Mode (TSSM)

If external capture is enabled (EXT_CAP_EN = 1), the external capture events will capture the GPRx, reset the counter CNT depending on ISL and issue a NEWVAL_IRQ. Functionality from previous table will be applied.

The source which will be used as external capture event for TSSM mode is selectable by TIM[i]_CH[x]_CNTS[19:18] : 00_B : source selection by EXT_CAP_SRC is in use. 01_B : tdu_word_evt signal will be used as source. 10_B : tdu_frame_evt signal will be used as source. 11_B : reserved Operation is done depending on the shift clock, external capture signal, DSL, ISL bit and the input signal value defined in next table:

Table 38.153 Operation depending on the shift clock, external capture signal, DSL, ISL and the input signal value

Input signal F_OUTx	Shift clock	tssm_ext_capture	ISL	DSL	Action description
—	0	1	1	—	do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
—	0	1	0	—	do capture; issue NEWVAL_IRQ; CNTS[15:8]=0
value	1	1	1	0	CNT[23:1]= CNT[22:0]; CNT[0]=value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
value	1	1	1	1	CNT[22:0]= CNT[23:1]; CNT[23]=value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0 CNT[23:0]=ECNT_RESET
value	1	1	0	0	CNT[23:1]= CNT[22:0]; CNT[0]=value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0
value	1	1	0	1	CNT[22:0]= CNT[23:1]; CNT[23]=value do capture; issue NEWVAL_IRQ; CNTS[15:8]=0

38.15.5 MAP Sub-module Interface

The GTM-IP provides one dedicated TIM sub-module TIM0 where channels zero (0) to five (5) are connected to the MAP sub-module described in **Section 38.21**. There, the TIM0 sub-module channels provide the input signal level together with the actual filter value and the annotated time stamp for the edge together in a 49 bit wide signal to the MAP sub-module. This 49 bit wide data signal is marked as valid with a separate valid signal tim0_map_dval[x] (x: 0 to 5).

Table 38.154 Structure of map data

tim0_map_data[x](48)	signal level bit from tim0_ch[x]
tim0_map_data[x](47:24)	actual filter value TIM0_CH[x]_FLT_RE/ TIM0_CH[x]_FLT_FE if corresponding channel x bit field FLT_MODE_RE/ FLT_MODE_FE is 1 else 0 is assigned.
tim0_map_data[x](23:0)	time stamp value selected by TBU0_SEL, GRP0_SEL, EGPR0_SEL, CNTS_SEL of channel x if bit field TIM_EN= 1
tim0_map_dval[x]	mark tim0_map_data[x] valid for one clock cycle

Note: With TIM_EN = 1 the MAP interface starts operation, it is not dependent on the setting of the bit fields TIM_MODE, ISL, DSL.

Note: While the MAP interface is in use the following guidelines have to be fulfilled, otherwise inconsistent filter values can be transferred.
Change TIM0_CH[x]_FLT_RE only between occurrence of rising and falling edge. Change TIM0_CH[x]_FLT_FE only between occurrence of falling and rising edge.

38.15.6 TIM Interrupt Signals

Table 38.155 TIM Interrupt Signals Table

Signal	Description
TIM[i]_NEWVAL[x]_IRQ	New measurement value detected by SMU of channel x (x: 0 to m–1)
TIM[i]_ECNTOFL[x]_IRQ	ECNT counter overflow of channel x (x: 0 to m–1)
TIM[i]_CNTOFL[x]_IRQ	SMU CNT counter overflow of channel x (x: 0 to m–1)
TIM[i]_GPROFL[x]_IRQ	GPR0 and GPR1 data overflow, old data was not read out before new data has arrived at input pin of channel x (x: 0 to m–1)
TIM[i]_TODET[x]_IRQ	Time out reached for input signal of channel x (x: 0 to m–1)
TIM[i]_GLITCHDET[x]_IRQ	A glitch was detected by the TIM filter of channel x (x: 0 to m–1)

38.15.7 TIM Configuration Registers Overview

TIM contains following configuration registers:

Table 38.156 Register list

Symbol	Register Name	Detail in Section
TIM[i]_CH[x]_CTRL	TIMi channel x control register	**38.15.8.1**, **38.15.8.2**
TIM[i]_CH[x]_ECTRL	TIMi channel x extended control register	**38.15.8.19**
TIM[i]_CH[x]_FLT_RE	TIMi channel x filter parameter 0 register	**38.15.8.3**
TIM[i]_CH[x]_FLT_FE	TIMi channel x filter parameter 1 register	**38.15.8.4**
TIM[i]_CH[x]_TDUV	TIMi channel x TDU control register	**38.15.8.16**
TIM[i]_CH[x]_TDUC	TIMi channel x TDU counter register	**38.15.8.17**
TIM[i]_CH[x]_GPR0	TIMi channel x general purpose 0 register	**38.15.8.5**
TIM[i]_CH[x]_GPR1	TIMi channel x general purpose 1 register	**38.15.8.6**
TIM[i]_CH[x]_CNT	TIMi channel x SMU counter register	**38.15.8.7**
TIM[i]_CH[x]_ECNT	TIMi channel x SMU edge counter register	**38.15.8.18**
TIM[i]_CH[x]_CNTS	TIMi channel x SMU shadow counter register	**38.15.8.8**
TIM[i]_CH[x]_IRQ_NOTIFY	TIMi channel x interrupt notification register	**38.15.8.9**
TIM[i]_CH[x]_IRQ_EN	TIMi channel x interrupt enable register	**38.15.8.10**
TIM[i]_CH[x]_EIRQ_EN	TIMi channel x error interrupt enable register	**38.15.8.15**
TIM[i]_CH[x]_IRQ_FORCINT	TIMi channel x force interrupt register	**38.15.8.11**
TIM[i]_CH[x]_IRQ_MODE	TIMi interrupt mode configuration register	**38.15.8.12**
TIM[i]_RST	TIMi global software reset register	**38.15.8.13**
TIM[i]_IN_SRC	TIMi AUX IN source selection register	**38.15.8.14**
TIM[i]_INP_VAL	TIMi input value observation register	**38.15.8.20**

38.15.8 TIM Configuration Registers Description

38.15.8.1 TIM[i]_CH[x]_CTRL (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1024_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOCTRL		EGPR1_SEL	EGPR0_SEL	FR_ECNT_OFL	CLK_SEL			FLT_CTR_FE	FLT_MODE_FE	FLT_CTR_RE	FLT_MODE_RE	EXT_CAP_EN	FLT_CNT_FRQ		FLT_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNT_RESET	ISL	DSL	CNTS_SEL	GPR1_SEL		GPR0_SEL		—	CICTRL	ARU_EN	OSM	TIM_MODE			TIM_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.157 TIM[i]_CH[x]_CTRL Register Contents (1/4)

Bit Position	Bit Name	Function
31, 30	TOCTRL	Timeout control 00_B: Timeout feature disabled 11_B: Timeout feature enabled for both edges 01_B: Timeout feature enabled for rising edge only 10_B: Timeout feature enabled for falling edge only **NOTE** It has to mention that writing of TOCTRL= 0 will every time stop the TDU, independent of the previous state of TOCTRL.
29	EGPR1_SEL	Extension of GPR1_SEL bit field. Details described in GPR1_SEL bit field.
28	EGPR0_SEL	Extension of GPR0_SEL bit field. Details described in GPR0_SEL bit field.
27	FR_ECNT_OFL	Extended Edge counter overflow behavior 0: Overflow will be signalled on ECNT bit width = 8 1: Overflow will be signalled on EECNT bit width (full range)
26 to 24	CLK_SEL	CMU clock source select for channel. If ECLK_SEL = 0: 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected If ECLK_SEL = 1: 000_B: tdu_sample_evt of TDU selected 001_B: Reserved 010_B: Reserved 011_B: Reserved 100_B: Reserved 101_B: Reserved 110_B: Reserved 111_B: Reserved

Table 38.157 TIM[i]_CH[x]_CTRL Register Contents (2/4)

Bit Position	Bit Name	Function
23	FLT_CTR_FE	Filter counter mode for falling edge. If FLT_MODE_FE = 1 / FLT_MODE_FE = 0 EFLT_CTR_FE, FLT_CTR_FE: 00_B = Up-Down Counter individual de-glitch mode / Immediate edge propagation mode 01_B = Hold Counter individual de-glitch mode / Immediate edge propagation mode 10_B = Reset Counter individual de-glitch mode / reserved 11_B = reserved / reserved
22	FLT_MODE_FE	Filter mode for falling edge. 0: Immediate edge propagation mode 1: Individual de-glitch mode
21	FLT_CTR_RE	Filter counter mode for rising edge. If FLT_MODE_RE = 1 / FLT_MODE_RE = 0 EFLT_CTR_RE, FLT_CTR_RE: 00_B = Up-Down Counter individual de-glitch mode / Immediate edge propagation mode 01_B = Hold Counter individual de-glitch mode / Immediate edge propagation mode 10_B = Reset Counter individual de-glitch mode / reserved 11_B = reserved / reserved
20	FLT_MODE_RE	Filter mode for rising edge. 0: Immediate edge propagation mode 1: Individual de-glitch mode
19	EXT_CAP_EN	Enables external capture mode. The selected TIM mode is only sensitive to external capture pulses the input event changes are ignored. 0: External capture disabled 1: External capture enabled
18, 17	FLT_CNT_FRQ	Filter counter frequency select 00_B: FLT_CNT counts with CMU_CLK0 01_B: FLT_CNT counts with CMU_CLK1 10_B: FLT_CNT counts with CMU_CLK6 11_B: FLT_CNT counts with CMU_CLK7
16	FLT_EN	Filter enable for channel x (x = 0 to 7) 0: Filter disabled and internal states are reset 1: Filter enabled **NOTE** If the filter is disabled all filter related units (including CSU) are bypassed, which means that the signal F_IN is directly routed to signal F_OUT.
15	ECNT_RESET	Enables resetting of counter in certain modes If TIM_MODE = 101_B (TGPS) / TIM_MODE = 000_B (TPWM) 0: ECNT counter operating in wrap around mode / ECNT counter operating in wrap around mode, CNT is reset on active input edge defined by DSL 1: ECNT counter is reset with periodic sampling / ECNT counter operating in wrap around mode, CNT is reset on active and inactive input edge else ECNT counter operating in wrap around mode; **NOTE** In TIM_MODE = 110_B (TSSM) the bit field ECNT_RESET defines the initial polarity for the shift register.
14	ISL	Ignore signal level 0: Use DSL bit for selecting active signal level (TIEM) 1: Ignore DSL and treat both edges as active edge (TIEM) This bit is mode dependent and will have different meanings (see details in the TIM Channel mode description).

Table 38.157 TIM[i]_CH[x]_CTRL Register Contents (3/4)

Bit Position	Bit Name	Function
13	DSL	Signal level control 0: Measurement starts with falling edge (low level measurement) 1: Measurement starts with rising edge (high level measurement) **NOTE** In TIM_MODE = 110_B (TSSM) the bit field DSL defines the shift direction. 0 = Shift left 1 = Shift right
12	CNTS_SEL	Selection for CNTS register 0: Use CNT register as input 1: Use TBU_TS0 as input **NOTE** CNTS_SEL in TSSM mode selects the source signal for registered or latched shift out operation. 0 = use F_OUTx 1 = use TIM_INx
11, 10	GPR1_SEL	Selection for GPR1 register If EGPR1_SEL =0: 00_B: Use TBU_TS0 as input 01_B: Use TBU_TS1 as input 10_B: Use TBU_TS2 as input 11_B: Use CNT as input If EGPR1_SEL =1: 00_B: Use ECNT as input 01_B: Use TIM_INP_VAL as input 10_B: Reserved 11_B: Reserved **NOTES** 1. In TBCM mode: EGPR1_SEL = 1, GPR1_SEL = 01 selects TIM_INP_VAL as input, in all other cases TIM Filter F_OUT is used. 2. If a reserved value is written to the EGPR1_SEL, GPR1_SEL bit fields, the hardware will use TBU_TS0 input.
9, 8	GPR0_SEL	Selection for GPR0 register If EGPR0_SEL =0: 00_B: Use TBU_TS0 as input 01_B: Use TBU_TS1 as input 10_B: Use TBU_TS2 as input 11_B: Use CNTS as input; if TGPS mode in channel = 0 is selected use TIM Filter F_OUT as input If EGPR0_SEL =1: 00_B: Use ECNT as input 01_B: Use TIM_INP_VAL as input 10_B: Reserved 11_B: Reserved **NOTE** If a reserved value is written to the EGPR0_SEL, GPR0_SEL bit fields, the hardware will use TBU_TS0 input.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.157 TIM[i]_CH[x]_CTRL Register Contents (4/4)

Bit Position	Bit Name	Function
6	CICTRL	Channel Input Control. 0: Use signal TIM_IN(x) as input for channel x 1: Use signal TIM_IN(x-1) as input for channel x (or TIM_IN(m-1) if x is 0)
5	ARU_EN	GPR0 and GPR1 register values routed to ARU 0: Registers content not routed 1: Registers content routed
4	OSM	One-shot mode 0: Continuous operation mode 1: One-shot mode **NOTE** After finishing the action in one-shot mode the TIM_EN bit is cleared automatically.
3 to 1	TIM_MODE	TIM channel x (x = 0 to 7) mode 000_B: PWM Measurement Mode (TPWM) 001_B: Pulse Integration Mode (TPIM) 010_B: Input Event Mode (TIEM) 011_B: Input Prescaler Mode (TIPM) 100_B: Bit Compression Mode (TBCM) 101_B: Gated Periodic Sampling Mode (TGPS) 110_B: Serial Shift Mode (TSSM) **NOTES** 1. If an undefined value is written to the TIM_MODE register, the hardware switches automatically to TIM_MODE = 000_B(TPWM mode). 2. The TIM_MODE register should not be changed while the TIM channel is enabled. 3. If the TIM channel is enabled and operating in TPWM or TPIM mode after the first valid edge defined by DSL has occurred, a reconfiguration of DSL,ISL,TIM_MODE will not change the channel behavior. Reading these bit fields after reconfiguration will show the newly configured settings but the initial channel behavior will not change. Only a disabling of the TIM channel by setting TIM_EN = 0 and reenabling with TIM_EN = 1 will change the channel operation mode.
0	TIM_EN	TIM channel x (x = 0 to 7) enable 0: Channel disabled 1: Channel enabled **NOTES** 1. Enabling of the channel resets the registers ECNT, TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_GPR0, and TIM[i]_CH[x]_GPR1 to their reset values. 2. After finishing the action in one-shot mode the TIM_EN bit is cleared automatically. Otherwise, the bit must be cleared manually.

38.15.8.2 TIM[i]_CH[x]_CTRL (i = 0, x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1024_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOCTRL		EGPR1_SEL	EGPR0_SEL	FR_ECNT_OFL	CLK_SEL			FLT_CTR_FE	FLT_MODE_FE	FLT_CTR_RE	FLT_MODE_RE	EXT_CAP_EN	FLT_CNT_FRQ		FLT_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNT_RESET	ISL	DSL	CNTS_SEL	GPR1_SEL		GPR0_SEL		TBU0_SEL	CICTRL	ARU_EN	OSM	TIM_MODE			TIM_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.158 TIM[i]_CH[x]_CTRL Register Contents (1/4)

Bit Position	Bit Name	Function
31, 30	TOCTRL	Timeout control 00_B: Timeout feature disabled 01_B: Timeout feature enabled for rising edge only 10_B: Timeout feature enabled for falling edge only 11_B: Timeout feature enabled for both edges **NOTE** It has to mention that writing of TOCTRL = 0 will every time stop the TDU, independent of the previous state of TOCTRL.
29	EGPR1_SEL	Extension of GPR1_SEL bit field. Details described in GPR1_SEL bit field.
28	EGPR0_SEL	Extension of GPR0_SEL bit field. Details described in GPR0_SEL bit field.
27	FR_ECNT_OFL	Extended Edge counter overflow behavior 0: Overflow will be signalled on ECNT bit width = 8 1: Overflow will be signalled on EECNT bit width (full range)
26 to 24	CLK_SEL	CMU clock source select for channel. If ECLK_SEL =0: 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected If ECLK_SEL =1: 000: tdu_sample_evt of TDU selected Other than above: Setting prohibited.
23	FLT_CTR_FE	Filter counter mode for falling edge. If FLT_MODE_FE=1 / FLT_MODE_FE=0 EFLT_CTR_FE, FLT_CTR_FE: 00_B: Up-Down Counter individual de-glitch mode / Immediate edge propagation mode 01_B: Hold Counter individual de-glitch mode / Immediate edge propagation mode 10_B: Reset Counter individual de-glitch mode / reserved 11_B: Setting prohibited

Table 38.158 TIM[i]_CH[x]_CTRL Register Contents (2/4)

Bit Position	Bit Name	Function
22	FLT_MODE_FE	Filter mode for falling edge. 0: Immediate edge propagation mode 1: Individual de-glitch mode
21	FLT_CTR_RE	Filter counter mode for rising edge. If FLT_MODE_RE = 1 / FLT_MODE_RE = 0 EFLT_CTR_RE, FLT_CTR_RE: 00_B: Up-Down Counter individual de-glitch mode / Immediate edge propagation mode 01_B: Hold Counter individual de-glitch mode / Immediate edge propagation mode 10_B: Reset Counter individual de-glitch mode / reserved 11_B: reserved / reserved
20	FLT_MODE_RE	Filter mode for rising edge. 0: Immediate edge propagation mode 1: Individual de-glitch mode
19	EXT_CAP_EN	Enables external capture mode. The selected TIM mode is only sensitive to external capture pulses the input event changes are ignored. 0: External capture disabled 1: External capture enabled
18, 17	FLT_CNT_FRQ	Filter counter frequency select 00_B: FLT_CNT counts with CMU_CLK0 01_B: FLT_CNT counts with CMU_CLK1 10_B: FLT_CNT counts with CMU_CLK6 11_B: FLT_CNT counts with CMU_CLK7
16	FLT_EN	Filter enable for channel x (x = 0 to 7) 0: Filter disabled and internal states are reset 1: Filter enabled **NOTE** If the filter is disabled all filter related units (including CSU) are bypassed, which means that the signal F_IN is directly routed to signal F_OUT.
15	ECNT_RESET	Enables resetting of counter in certain modes If TIM_MODE = 101_B (TGPS) / TIM_MODE = 000_B (TPWM) 0: ECNT counter operating in wrap around mode / ECNT counter operating in wrap around mode, CNT is reset on active input edge defined by DSL 1: ECNT counter is reset with periodic sampling / ECNT counter operating in wrap around mode, CNT is reset on active and inactive input edge else ECNT counter operating in wrap around mode; **NOTE** In TIM_MODE = 110_B (TSSM) the bit field ECNT_RESET defines the initial polarity for the shift register.
14	ISL	Ignore signal level 0: Use DSL bit for selecting active signal level (TIEM) 1: Ignore DSL and treat both edges as active edge (TIEM) This bit is mode dependent and will have different meanings (see details in the TIM Channel mode description).
13	DSL	Signal level control 0: Measurement starts with falling edge (low level measurement) 1: Measurement starts with rising edge (high level measurement) **NOTE** In TIM_MODE = 110_B (TSSM) the bit field DSL defines the shift direction. 0 = Shift left 1 = Shift right

Table 38.158 TIM[i]_CH[x]_CTRL Register Contents (3/4)

Bit Position	Bit Name	Function
12	CNTS_SEL	Selection for CNTS register 0: Use CNT register as input 1: Use TBU_TS0 as input **NOTE** The functionality of the CNTS_SEL is disabled in the modes TIPM and TBCM.
11, 10	GPR1_SEL	Selection for GPR1 register If EGPR1_SEL = 0: 00_B: Use TBU_TS0 as input 01_B: Use TBU_TS1 as input 10_B: Use TBU_TS2 as input 11_B: Use CNT as input If EGPR1_SEL =1: 00_B: Use ECNT as input 01_B: Use TIM_INP_VAL as input Other than above: Setting prohibited. **NOTES** 1. In TBCM mode: EGPR1_SEL=1, GPR1_SEL=01 selects TIM_INP_VAL as input, in all other cases TIM Filter F_OUT is used. 2. If a reserved value is written to the EGPR1_SEL, GPR1_SEL bit fields, the hardware will use TBU_TS0 input.
9, 8	GPR0_SEL	Selection for GPR0 register If EGPR0_SEL = 0: 00_B: Use TBU_TS0 as input 01_B: Use TBU_TS1 as input 10_B: Use TBU_TS2 as input 11_B: Use CNTS as input; if TGPS mode in channel = 0 is selected use TIM Filter F_OUT as input If EGPR0_SEL =1: 00_B: Use TBU_TS0 as input / use ECNT as input 01_B: Use TBU_TS1 as input / use TIM_INP_VAL as input 10_B: Use TBU_TS2 as input / reserved 11_B: Use CNTS as input; if TGPS mode in channel = 0 is selected use TIM Filter F_OUT as input / reserved **NOTE** If a reserved value is written to the EGPR0_SEL, GPR0_SEL bit fields, the hardware will use TBU_TS0 input.
7	TBU0_SEL	TBU_TS0 bits input select for TIM0_CH[x]_GPRz (z: 0, 1) 0: Use TBU_TS0(23 to 0) to store in TIM0_CH[x]_GPR0/TIM0_CH[x]_GPR1 1: Use TBU_TS0(26 to 3) to store in TIM0_CH[x]_GPR0/TIM0_CH[x]_GPR1 **NOTE** This bit is only applicable for TIM0
6	CICTRL	Channel Input Control. 0: Use signal TIM_IN(x) as input for channel x 1: Use signal TIM_IN(x-1) as input for channel x (or TIM_IN(m-1) if x is 0)
5	ARU_EN	GPR0 and GPR1 register values routed to ARU 0: Registers content not routed 1: Registers content routed

Table 38.158 TIM[i]_CH[x]_CTRL Register Contents (4/4)

Bit Position	Bit Name	Function
4	OSM	One-shot mode 0: Continuous operation mode 1: One-shot mode **NOTE** After finishing the action in one-shot mode the TIM_EN bit is cleared automatically.
3 to 1	TIM_MODE	TIM channel x (x = 0 to m-1) mode 000_B: PWM Measurement Mode (TPWM) 001_B: Pulse Integration Mode (TPIM) 010_B: Input Event Mode (TIEM) 011_B: Input Prescaler Mode (TIPM) 100_B: Bit Compression Mode (TBCM) 101_B: Gated Periodic Sampling Mode (TGPS) 110_B: Serial Shift Mode (TSSM) **NOTES** 1. If an undefined value is written to the TIM_MODE register, the hardware switches automatically to TIM_MODE = 000_B (TPWM mode). 2. The TIM_MODE register should not be changed while the TIM channel is enabled. 3. If the TIM channel is enabled and operating in TPWM or TPIM mode after the first valid edge defined by DSL has occurred, a reconfiguration of DSL, ISL, TIM_MODE will not change the channel behavior. Reading these bit fields after reconfiguration will show the newly configured settings but the initial channel behavior will not change. Only a disabling of the TIM channel by setting TIM_EN = 0 and reenabling with TIM_EN = 1 will change the channel operation mode.
0	TIM_EN	TIM channel x (x = 0 to m-1) enable 0: Channel disabled 1: Channel enabled **NOTES** 1. Enabling of the channel resets the registers ECNT, TIM[i]_CH[x]_CNT, TIM[i]_CH[x]_GPR0, and TIM[i]_CH[x]_GPR1 to their reset values. 2. After finishing the action in one-shot mode the TIM_EN bit is cleared automatically. Otherwise, the bit must be cleared manually.

38.15.8.3 TIM[i]_CH[x]_FLT_RE (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $101C_H$ + (800_H × i) + (80_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	FLT_RE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLT_RE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.159 TIM[i]_CH[x]_FLT_RE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	FLT_RE	Filter parameter for rising edge.

Note: FLT_RE has different meanings in the various filter modes. Immediate edge propagation mode = acceptance time for rising edge Individual deglitch time mode = deglitch time for rising edge.

38.15.8.4 TIM[i]_CH[x]_FLT_FE (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1020_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	FLT_FE							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLT_FE															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.160 TIM[i]_CH[x]_FLT_FE Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	FLT_FE	Filter parameter for falling edge.

Note: FLT_FE has different meanings in the various filter modes. Immediate edge propagation mode = acceptance time for falling edge Individual deglitch time mode = deglitch time for falling edge.

38.15.8.5 TIM[i]_CH[x]_GPR0 (x = 0 to 7)

Access: When read, the value after reset is returned. When writing, write the value after reset.

Address: <GTM_base> + 1000_H + (800_H × i) + (80_H × x)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNT								GPR0							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPR0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.161 TIM[i]_CH[x]_GPR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	ECNT	Edge counter. **NOTES** 1. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. 2. The ECNT register is reset to its initial value when the channel is enabled. 3. Bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
23 to 0	GPR0	Input signal characteristic parameter 0. **NOTES** 1. The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit fields EGPR0_SEL, GPR0_SEL of register TIM[i]_CH[x]_CTRL. 2. The content of this register can only be written in TIM channel mode TSSM.

38.15.8.6 TIM[i]_CH[x]_GPR1(x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: GTM_base> + 1004_H + (800_H × i) + (80_H × x)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNT								GPR1							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPR1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.162 TIM[i]_CH[x]_GPR1 Register Contents

Bit Position	Bit Name	Function
31 to 24	ECNT	Edge counter. **NOTES** 1. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. 2. The ECNT register is reset to its initial value when the channel is enabled. 3. Bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
23 to 0	GPR1	Input signal characteristic parameter 1. **NOTES** 1. The content of this register has different meaning for the TIM channels modes. The content directly depends on the bit fields EGPR1_SEL, GPR1_SEL of register TIM[i]_CH[x]_CTRL. 2. In TBCM mode if EGPR1_SEL = 1, GPR1_SEL = 01 then TIM_INP_VAL is used as input in all other cases TIM Filter F_OUT is used as input and Bits GPR1(23:8) = 0 3. The content of this register can only be written in TIM channel mode TGPS and TSSM.

38.15.8.7 TIM[i]_CH[x]_CNT (x = 0 to 7)

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 1008_H + (800_H × i) + (80_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.163 TIM[i]_CH[x]_CNT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	CNT	Actual SMU counter value. **NOTE** The meaning of this value depends on the configured mode: TPWM = actual duration of PWM signal. TPIM = actual duration of all pulses (sum of pulses). TIEM = actual number of received edges. TIPM = actual number of received edges. TGPS = elapsed time for periodic sampling. TSSM = shift data.

38.15.8.8 TIM[i]_CH[x]_CNTS (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1010_H + (800_H × i) + (80_H × x)

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECNT								CNTS							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNTS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.164 TIM[i]_CH[x]_CNTS Register Contents

Bit Position	Bit Name	Function
31 to 24	ECNT	Edge counter. **NOTES** 1. The ECNT counts every incoming filtered edge (rising and falling). The counter value is uneven in case of detected rising, and even in case of detected falling edge. Thus, the input signal level is part of the counter and can be obtained by bit 0 of ECNT. 2. The ECNT register is reset to its initial value when the channel is enabled. 3. Bit 0 depends on the input level coming from the filter unit and defines the reset value immediately.
23 to 0	CNTS	Counter shadow register. **NOTES** 1. The content of this register has different meaning for the TIM channels modes. The content depends directly on the bit field CNTS_SEL of register TIM[i]_CH[x]_CTRL. 2. The register TIM[i]_CH[x]_CNTS is only writable in TIPM, TBCM, TGPS and TSSM mode.

38.15.8.9 TIM[i]_CH[x]_IRQ_NOTIFY (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $102C_H$ + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GLITCHDET	TODET	GPROFL	CNTOFL	ECNTOFL	NEWVAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.165 TIM[i]_CH[x]_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	GLITCHDET	Glitch detected on channel x, (x = 0 to m-1). 0: No glitch detected for last edge 1: Glitch detected for last edge **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
4	TODET	Timeout reached for input signal of channel x, (x = 0 to m-1). See bit 0.
3	GPROFL	GPR0 and GPR1 data overflow, old data not read out before new data has arrived at input pin, (x = 0 to m-1). See bit 0.
2	CNTOFL	SMU CNT counter overflow of channel x, (x = 0 to m-1). See bit 0.
1	ECNTOFL	ECNT counter overflow of channel x, (x = 0 to m-1). See bit 0.
0	NEWVAL	New measurement value detected by in channel x (x =0 to m-1) 0: No event was occurred 1: NEWVAL was occurred on the TIM channel **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

38.15.8.10 TIM[i]_CH[x]_IRQ_EN (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1030_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GLITCHDET_IRQ_EN	TODET_IRQ_EN	GPROFL_IRQ_EN	CNTOFL_IRQ_EN	ECNTOFL_IRQ_EN	NEWVAL_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.166 TIM[i]_CH[x]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	GLITCHDET_ IRQ_ EN	TIM_GLITCHDETx_IRQ interrupt enable, see bit 0.
4	TODET_IRQ_ EN	TIM_TODETx_IRQ interrupt enable, see bit 0.
3	GPROFL_IRQ_EN	TIM_GPROFL_IRQ interrupt enable, see bit 0.
2	CNTOFL_IRQ_EN	TIM_CNTOFLx_IRQ interrupt enable, see bit 0.
1	ECNTOFL_IRQ_EN	TIM_ECNTOFLx_IRQ interrupt enable, see bit 0.
0	NEWVAL_IRQ_EN	TIM_NEWVALx_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

38.15.8.11 TIM[i]_CH[x]_IRQ_FORCINT (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1034_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TRG_GLITCHDET	TRG_TODET	TRG_GPROFL	TRG_CNTOFL	TRG_ECNTOFL	TRG_NEWVAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.167 TIM[i]_CH[x]_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TRG_GLITCHDET	Trigger GLITCHDET bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software, see bit 0.
4	TRG_TODET	Trigger TODET bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software, see bit 0.
3	TRG_GPROFL	Trigger GPROFL bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software, see bit 0.
2	TRG_CNTOFL	Trigger CNTOFL bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software, see bit 0.
1	TRG_ECNTOFL	Trigger ECNTOFL bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software, see bit 0.
0	TRG_NEWVAL	Trigger NEWVAL bit in TIM[i]_CH[x]_IRQ_NOTIFY register by software 0: No interrupt triggering. 1: Assert corresponding field in TIM[i]_CH[x]_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

38.15.8.12 TIM[i]_CH[x]_IRQ_MODE (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1038_H + (800_H × i) + (80_H × x)

Value after reset: 0000 000X_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.168 TIM[i]_CH[x]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.15.8.13 TIM[i]_RST

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $107C_H$ + (800_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.169 TIM[i]_RST Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	RST_CH7	Software reset of channel 7, see bit 0.
6	RST_CH6	Software reset of channel 6, see bit 0.
5	RST_CH5	Software reset of channel 5, see bit 0.
4	RST_CH4	Software reset of channel 4, see bit 0.
3	RST_CH3	Software reset of channel 3, see bit 0.
2	RST_CH2	Software reset of channel 2, see bit 0.
1	RST_CH1	Software reset of channel 1, see bit 0.
0	RST_CH0	Software reset of channel 0 0: No action 1: Reset channel 0 **NOTE** This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately.

38.15.8.14 TIM[i]_IN_SRC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1078_H + (800_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MODE_7		VAL_7		MODE_6		VAL_6		MODE_5		VAL_5		MODE_4		VAL_4	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MODE_3		VAL_3		MODE_2		VAL_2		MODE_1		VAL_1		MODE_0		VAL_0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.170 TIM[i]_IN_SRC Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	MODE_7	Input source to Channel 7, see bits 3, 2.
29, 28	VAL_7	Value to be fed to Channel 7, see bits 1, 0.
27, 26	MODE_6	Input source to Channel 6, see bits 3, 2.
25, 24	VAL_6	Value to be fed to Channel 6, see bits 1, 0.
23, 22	MODE_5	Input source to Channel 5, see bits 3, 2.
21, 20	VAL_5	Value to be fed to Channel 5, see bits 1, 0.
19, 18	MODE_4	Input source to Channel 4, see bits 3, 2.
17, 16	VAL_4	Value to be fed to Channel 4, see bits 1, 0.
15, 14	MODE_3	Input source to Channel 3, see bits 3, 2.
13, 12	VAL_3	Value to be fed to Channel 3, see bits 1, 0.
11, 10	MODE_2	Input source to Channel 2, see bits 3, 2.
9, 8	VAL_2	Value to be fed to Channel 2, see bits 1, 0.
7, 6	MODE_1	Input source to Channel 1, see bits 3, 2.
5, 4	VAL_1	Value to be fed to Channel 1, see bits 1, 0.
3, 2	MODE_0	Input source to Channel 0 Multi-core encoding in use (MODE_x(1) defines the state of the signal) 00_B: State is 0 (ignore write access) 01_B: Change state to 0 10_B: Change state to 1 11_B: State is 1 (ignore write access) Function table: MODE_x(1) = 0, VAL_x(1) = 0: The input signal defined by bit field CICTRL of the TIM channel is used as input source. MODE_x(1) = 0, VAL_x(1) = 1: The signal TIM_AUX_IN of the TIM channel is used as input source. MODE_x(1) = 1: The state VAL_x(1) defines the input level for the TIM channel. **NOTE** Any read access to a MODE_x bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

Table 38.170 TIM[i]_IN_SRC Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	VAL_0	Value to be fed to Channel 0 Multicore encoding in use (VAL_x(1) defines the state of the signal) 00_B: State is 0 (ignore write access) 01_B: Change state to 0 10_B: Change state to 1 11_B: State is 1 (ignore write access) Function depends on the combination of VAL_x(1) and MODE_x(1) see MODE_0 description. **NOTE** Any read access to a VAL_x bit field will always result in a value 00 or 11 indicating current state. A modification of the state is only performed with the values 01 and 10. Writing the values 00 and 11 is always ignored.

38.15.8.15 TIM[i]_CH[x]_EIRQ_EN (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $103C_H$ + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GLITCHDET_EIRQ_EN	TODET_EIRQ_EN	GPROFL_EIRQ_EN	CNTOFL_EIRQ_EN	ECNTOFL_EIRQ_EN	NEWVAL_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.171 TIM[i]_CH[x]_EIRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	GLITCHDET_EIRQ_EN	TIM_GLITCHDETx_IRQ interrupt enable, see bit 0.
4	TODET_EIRQ_EN	TIM_TODETx_IRQ interrupt enable, see bit 0.
3	GPROFL_EIRQ_EN	TIM_GPROFL_IRQ interrupt enable, see bit 0.
2	CNTOFL_EIRQ_EN	TIM_CNTOFLx_IRQ interrupt enable, see bit 0.
1	ECNTOFL_EIRQ_EN	TIM_ECNTOFLx_IRQ interrupt enable, see bit 0.
0	NEWVAL_EIRQ_EN	TIM_NEWVALx_EIRQ error interrupt enable 0: Disable error interrupt, error interrupt is not visible outside GTM-IP 1: Enable error interrupt, error interrupt is visible outside GTM-IP

38.15.8.16 TIM[i]_CH[x]_TDUV (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1018_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TCS			TDU_SAME_CNT_CLK	TCS_USE_SAMPLE_EVT	SLICING		TOV2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOV1								TOV							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.172 TIM[i]_CH[x]_TDUV Register Contents

Bit Position	Bit Name	Function
31	Reserved	This bit is always read as 0. When written, write the initial value.
30 to 28	TCS	Timeout Clock selection 000_B: CMU_CLK0 selected 001_B: CMU_CLK1 selected 010_B: CMU_CLK2 selected 011_B: CMU_CLK3 selected 100_B: CMU_CLK4 selected 101_B: CMU_CLK5 selected 110_B: CMU_CLK6 selected 111_B: CMU_CLK7 selected
27	TDU_SAME_CNT_CLK	TDU_SAME_CNT_CLK: Define clocking of TO_CNT, TO_CNT1. 0: TO_CNT clock selected by (TCS,TCS_USE_SAMPLE_EVT); TO_CNT1 clocked on tdu_word_event 1: TO_CNT1 uses same clock as TO_CNT
26	TCS_USE_SAMPLE_EVT	TCS_USE_SAMPLE_EVT: Use tdu_sample_evt as Timeout Clock 0: CMU_CLK selected by TCS is in use by TO_CNT,TO_CNT2 1: CMU_CLK selected by TCS is in use by TO_CNT2;
25, 24	SLICING	Cascading of counter slices Slicing by use_lut If USE_LUT=00 / USE_LUT!=00 SLICING: 00_B: Combine slice2,slice1,slice0 to 1x24 bit counter / reserved 01_B: Combine slice1,slice0 to 1x16bit counter use slice2 as 1x8 bit counter / combine slice1,slice0 to 1x16bit slice2 not usable 10_B: Use slice2,slice1,slice0 as 3x8 bit counter / use slice1,slice0 as 2x8 bit counter slice2 not usable 11_B: Use slice1,slice0 as 2x8 bit counter / use slice1,slice0 as 2x8 bit counter
23 to 16	TOV2	Time out compare value slice2 channel x (x = 0 to m-1). SLICING! = 11: Compare value for TO_CNT2 SLICING = 11: TOV2 operate as a shadow register for TO_CNT
15 to 8	TOV1	Time out compare value slice1 channel x (x = 0 to m-1). Compare value for TO_CNT1
7 to 0	TOV	Time out compare value slice0 channel x (x = 0 to m-1). Compare value for TO_CNT

38.15.8.17 TIM[i]_CH[x]_TDUC (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1014_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TO_CNT2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TO_CNT1								TO_CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.173 TIM[i]_CH[x]_TDUC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 16	TO_CNT2	Current Timeout value slice2 for channel x (x = 0 to m–1). Counter will be reset to 0X0 on TDU_RESYNC condition
15 to 8	TO_CNT1	Current Timeout value slice1 for channel x (x = 0 to m–1). Counter will be reset to 0X0 on TDU_RESYNC condition
7 to 0	TO_CNT	Current Timeout value slice0 for channel x (x = 0 to m–1). SLICING! = 11_B: counter will be reset to 0X0 on TDU_RESYNC condition SLICING = 11_B: counter will be loaded with TOV2 on TDU_RESYNC condition

Note: The register TIM[i]_CH[x]_TDUC is writable if Timeout unit is disabled (TOCTRL =00_B).

Note: If USE_LUT! = 00_B (input signal generation by lookup table) the bit field TO_CNT2 is writable at any time, TO_CNT, TO_CNT1 will not be changed.

38.15.8.18 TIM[i]_CH[x]_ECNT (x = 0 to 7)

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $100C_H$ + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.174 TIM[i]_CH[x]_ECNT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	ECNT	Edge counter **NOTE** If TIM channel is disabled the content of ECNT gets frozen. A read will auto clear the bits [15:1]. Further read accesses to ECNT will show on Bit 0 the actual input signal value of the channel.

38.15.8.19 TIM[i]_CH[x]_ECTRL (x = 0 to 7)

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1028_H + (800_H × i) + (80_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	USE_PREV_CH_IN	ECLK_SEL	IMM_START	SWAP_CAPTURE	—	—	EFLT_CTR_FE	EFLT_CTR_RE	USE_LUT		—	—	TDU_RESYNC			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TDU_STOP			—	TDU_START			TODET_IRQ_SRC		USE_PREV_TDU_IN	—	EXT_CAP_SRC			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Table 38.175 TIM[i]_CH[x]_ECTRL Register Contents (1/4)

Bit Position	Bit Name	Function
31	USE_PREV_CH_IN	Select input data source for TIM channel. 0: Use input data of local filter unit for channel measurements 1: Use input data of previous channel (after filter unit) for channel measurements
30	ECLK_SEL	Extension of bit field CLK_SEL. Details described in CLK_SEL bit field of register TIM[i]_CH[x]_CTRL.
29	IMM_START	Start immediately the measurement 0: Start with first active edge the measurement 1: Start immediately after enable (TIM_EN=1) the measurement This bit is only applicable in TPWM and TPIM mode. Set to 0 in all other modes.
28	SWAP_CAPTURE	swap point of time of capturing CNTS and GPR1 0: Inactive edge will capture data in CNTS; NEWVAL_IRQ event will capture data in GPR1 1: Swap time of capture: inactive edge will capture data in GPR1; NEWVAL_IRQ event will capture data in CNTS This bit is only applicable in TPWM and TPIM mode. Set to 0 in all other modes.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	EFLT_CTR_FE	Extension of bit field FLT_CTR_FE. Details described in FLT_CTR_FE bit field of register TIM[i]_CH[x]_CTRL.
24	EFLT_CTR_RE	Extension of bit field FLT_CTR_RE. Details described in FLT_CTR_RE bit field of register TIM[i]_CH[x]_CTRL.
23, 22	USE_LUT	Generate Filter input by lookup table 00_B: Lookup table not in use, lut_in0(x) used as filter input 01_B: Use 3 bit lookup table with index = ext_capture(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index]. 10_B: Use 3 bit lookup table with index = fout_prev(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index]. 11_B: Use 3 bit lookup table with index = tssm_out(x) & lut_in1(x) & lut_in0(x). Filter input is defined by TO_CNT2[index].
21, 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.175 TIM[i]_CH[x]_ECTRL Register Contents (2/4)

Bit Position	Bit Name	Function
19 to 16	TDU_RESYNC	Defines condition which will resynchronize the TDU unit. Behavior with SLICING! = 11_B: 0000_B: Reset counter TO_CNT2 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event (see Note); reset counters TO_CNT,TO_CNT1 on tdu_timeout_evt or on tdu_start_000_event (see Note); if SLICING= 10_B and TO_CTRL= –1 then reset TO_CNT on rising input edge; if SLICING= 10_B and TO_CTRL= 1– then reset TO_CNT1 on falling input edge; if SLICING! = 10_B then reset counters TO_CNT,TO_CNT1 on each active edge selected by TOCTRL; 0—1: If SLICING= 10_B and TO_CTRL= x1 then reset TO_CNT on rising input edge; if SLICING= 10_B and TO_CTRL= 1– then reset TO_CNT1 on falling input edge; if SLICING! = 10_B then reset counters TO_CNT, TO_CNT1 on each active edge selected by TOCTRL; if SLICING= 00_B then reset TO_CNT2 on each active edge selected by TOCTRL; 0x1–: Reset counters TO_CNT on tdu_word_evt; 01—: Reset counter TO_CNT1 on tdu_frame_evt; if SLICING= 01_B then reset TO_CNT on tdu_frame_evt; 1000_B: Reset counters TO_CNT,TO_CNT1,TO_CNT2 on event selected by EXT_CAP_SRC; 1—: If SLICING!= 00_B then reset counter TO_CNT2 on tdu_sample_evt; 1—1: Reset counter TO_CNT2 on each active edge selected by TOCTRL; if SLICING= 10_B and TO_CTRL= –1 then reset TO_CNT on rising input edge; if SLICING= 10_B and TO_CTRL= 1– then reset TO_CNT1 on falling input edge; if SLICING! = 10_B then reset counters TO_CNT, TO_CNT1 on each active edge selected by TOCTRL; 1–1–: Reset counters TO_CNT on tdu_word_evt; 11—: Reset counter TO_CNT1 on tdu_frame_evt; if SLICING=01_B then reset TO_CNT on tdu_frame_evt; Behavior with SLICING = 11_B: 0000_B: Load counter TO_CNT with TOV2 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event (see Note); reset counter TO_CNT1 on each active edge selected by TOCTRL or tdu_timeout_evt or on tdu_start_000_event (see Note) —1: Load counter TO_CNT with TOV2 on each active edge selected by TOCTRL; reset counter TO_CNT1 on each active edge selected by TOCTRL 0–1–: Load counter TO_CNT with TOV2 on tdu_word_evt 1–1–: Reset counter TO_CNT on tdu_word_evt –1—: Reset counter TO_CNT1 on tdu_frame_evt 1000_B: Load counter TO_CNT with TOV2; reset counter TO_CNT1 on event selected by EXT_CAP_SRC **NOTE** tdu_start_000_event is defined as: Each writing of TOCTRL!=0 (independent of current TOCTRL) while TDU_START=000_B and TDU is stopped (initially or stopped by TDU_STOP event). This event will last 1 system clock cycle.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.175 TIM[i]_CH[x]_ECTRL Register Contents (3/4)

Bit Position	Bit Name	Function
14 to 12	TDU_STOP	Defines condition which will stop the TDU unit. 000_B: Immediate stop counting of TDU on tdu_toctrl_0_event (see Note) 001_B: stop counting of TDU on tdu_word_evt or on tdu_toctrl_0_event (see Note) 010_B: stop counting of TDU on tdu_frame_evt or on tdu_toctrl_0_event (see Note) 011_B: stop counting of TDU on tdu_timeout_evt or on tdu_toctrl_0_event (see Note) 100_B: stop counting of TDU on external capture event or on tdu_toctrl_0_event (see Note) 101_B: if SLICING =10_B|11_B then stop counting of TO_CNT on tdu_word_evt or on tdu_toctrl_0_event (see Note); stop counting of TO_CNT1 on tdu_frame_evt or on tdu_toctrl_0_event (see Note); stop counting of TO_CNT2 on tdu_toctrl_0_event (see Note); else reserved, no action performed $11–_B$: reserved, no action performed **NOTE** tdu_toctrl_0_event is defined as: Each writing of TOCTRL = 0 (independent of current TOCTRL) while TDU is started. This event will last 1 system clock cycle.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	TDU_START	Defines condition which will start the TDU unit. 000_B: start once immediate on tdu_start_000_event (see Note) 001_B: start once with occurrence of first cmu_clk selected by CLK_SEL when measure unit is enabled by TIM_EN = 1 010_B: start once with occurrence of first active edge selected by TOCTRL; restart on tdu_frame_evt if TDU is stopped 011_B: start once with occurrence of first active edge selected by TOCTRL 100_B: start/restart with occurrence of external capture event (if TDU is stopped, restart again) 101_B: start/restart with occurrence of first cmu_clk selected by CLK_SEL when measure unit is enabled by TIM_EN = 1 (if TDU is stopped, restart again) 110_B: start once with occurrence of external capture event; restart on tdu_frame_evt if TDU is stopped 111_B: start/restart with occurrence of first active edge selected by TOCTRL (if TDU is stopped, restart again) **NOTES** 1. tdu_start_000_event is defined as: Each writing of TOCTRL! = 0 (independent of current TOCTRL) while TDU_START = 000_B and TDU is stopped (initially or stopped by TDU_STOP event). This event will last 1 system clock cycle. 2. In mode SLICING = 11_B every start/restart will load the TO_CNT with value TOV2.
7, 6	TODET_IRQ_SRC	Selection of source for TODET_IRQ 00_B: use tdu_timeout_evt 01_B: use tdu_word_evt 10_B: use tdu_frame_evt 11_B: use tdu_sample_evt **NOTE** With TODET_IRQ_SRC = 00_B the ACB bit 2 will be driven by signal tdu_timeout_evt, if TODET_IRQ_SRC! = 00_B ACB2 will be 0.
5	USE_PREV_TDU_IN	Select input data source for TDU. 0: use input data of local filter for TDU 1: use input data of previous channel (after filter unit) for TDU
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.175 TIM[i]_CH[x]_ECTRL Register Contents (4/4)

Bit Position	Bit Name	Function
3 to 0	EXT_CAP_SRC	Defines selected source for triggering the EXT_CAPTURE functionality. 0000_B: NEW_VAL_IRQ of following channel selected 0001_B: AUX_IN selected 0010_B: CNTOFL_IRQ of following channel selected 0011_B and CICTRL = 1: Use signal TIM_IN(x) as input for channel x 0011_B and CICTRL = 0: Use signal TIM_IN(x–1) as input for channel x (or TIM_IN(m–1) if x is 0) 0100_B: ECNTOFL_IRQ of following channel selected 0101_B:TODET_IRQ of following channel selected 0110_B: GLITCHDET_IRQ of following channel selected 0111_B: GPROFL_IRQ of following channel selected 1000_B: cmu_clk selected by CLK_SEL of following channel 1001_B: REDGE_DET of following channel selected 1010_B: FEDGE_DET of following channel selected 1011_B: Logical or of(FEDGE_DET, REDGE_DET) of following channel selected 1100_B: tdu_sample_evt of local TDU selected 1101_B: tdu_word_evt of local TDU selected 1110_B: tdu_frame_evt of local TDU selected 1111_B: Reserved **NOTE** Undefined values will not be written and AEI_STATUS will signal "10_B"

38.15.8.20 TIM[i]_INP_VAL

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 1074_H + (800_H × i)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TIM_IN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F_IN								F_OUT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.176 TIM[i]_INP_VAL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 16	TIM_IN	Signals after TIM input signal synchronization
15 to 8	F_IN	Signals after INPSRC selection, before TIM FLT unit
7 to 0	F_OUT	Signals after TIM FLT unit

38.16 Timer Output Module (TOM)

38.16.1 Overview

The Timer Output Module (TOM) offers up to 16 independent channels (index x) to generate simple PWM signals at each output pin TOM[i]_CH[x]_OUT.

Additionally, at TOM output TOM[i]_CH15_OUT a pulse count modulated signal can be generated.

The architecture of the TOM sub-module is depicted in **Figure 38.49**.

Indices and their range as used inside this chapter are:

nTOM= number of TOM instances in the device, refer to device specific **38.28, GTM Device 358** for correct value.

i=0..nTOM-1

x=0..15 (number of channels)

y=0,1

z=0..7

The register and bitfield names which are used inside this chapter has the following meaning:

SR0 = TOM[i]_CH[x]_SR0

SR1 = TOM[i]_CH[x]_SR1

CM0 = TOM[i]_CH[x]_CM0

CM1 = TOM[i]_CH[x]_CM1

CN0 = TOM[i]_CH[x]_CN0

CLK_SRC_SR = TOM[i]_CH[x]_CTRL.CLK_SRC_SR

SL = TOM[i]_CH[x]_CTRL.SL

All figures in this chapter which are shown block diagrams or gate schematics are drawn in principal to show specific functionality. They don't have the claim of completeness and do not represent the real implementation of the design.

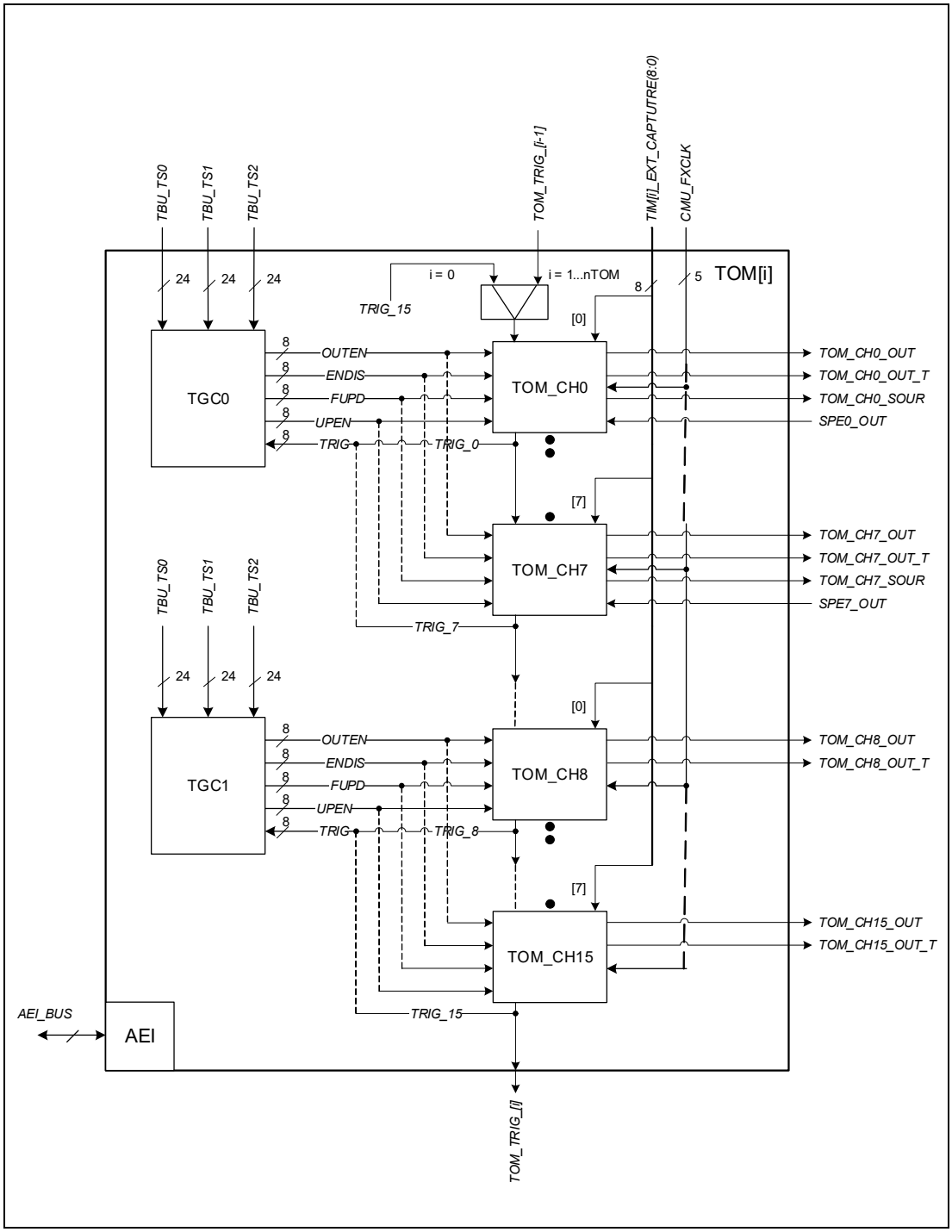


Figure 38.49 TOM block diagram

The two sub-modules TGC0 and TGC1 are global channel control units that control the enabling/ disabling of the channels and their outputs as well as the update of their period and duty cycle register.

The module TOM receives two (three) timestamp values TBU_TS0, TBU_TS1 (and TBU_TS2) in order to realize synchronized output behavior on behalf of a common time base.

The 5 dedicated clock line inputs CMU_FXCLK are providing divided clocks that can be selected to clock the output pins.

The trigger signal TOM_TRIGIN[i-1] of TOM instance i comes from the preceding instance i-1, the trigger TOM_TRIGOUT[i] is routed to succeeding instance i+1. The purpose of the trigger signal is explained later.

NOTE

TOM0 is connected to its own output TOM_TRIG_0, i.e. the last channel of TOM instance 0 can trigger the first channel of TOM instance 0 (this path is registered, which means delayed by one SYS_CLK period).

38.16.2 TOM Global Channel Control (TGC0, TGC1)

38.16.2.1 Overview

There exist two global channel control units (TGC0 and TGC1) to drive a number of individual TOM channels synchronously by external or internal events.

Each TGC[y] can drive up to eight TOM channels where TGC0 controls TOM channels 0 to 7 and TGC1 controls TOM channels 8 to 15.

The TOM sub-module supports four different kinds of signaling mechanisms:

- Global enable/disable mechanism for each TOM channel with control register TOM[i]_TGC[y]_ENDIS_CTRL and status register TOM[i]_TGC[y]_ENDIS_STAT
- Global output enable mechanism for each TOM channel with control register TOM[i]_TGC[y]_OUTEN_CTRL and status register TOM[i]_TGC[y]_OUTEN_STAT
- Global force update mechanism for each TOM channel with control register TOM[i]_TGC[y]_FUPD_CTRL
- Update enable of the register CM0, CM1 and CLK_SRC for each TOM channel with the control bit field UPEN_CTRL[z] of TOM[i]_TGC[y]_GLB_CTRL

38.16.2.2 TGC Sub-unit

Each of the first three individual mechanisms (enable/disable of the channel, output enable and force update) can be driven by three different trigger sources. The three trigger sources are:

- The host CPU (bit HOST_TRIG of register TOM[i]_TGC[y]_GLB_CTRL)
- The TBU time stamp (signal TBU_TS0, TBU_TS1, TBU_TS2 if available)
- The internal trigger signal TRIG (bunch of trigger signals TRIG_[x]) which can be either the trigger TRIG_CCU0 of channel x, the trigger of preceding channel x-1 (i.e. signal TRIG_[x–1]) or the external trigger TIM_EXT_CAPTURE(t) of assigned TIM channel t.

The first way is to trigger the control mechanism by a direct register write access via host CPU (bit HOST_TRIG of register ATOM[i]_TGC[y]_GLB_CTRL).

The second way is provided by a compare match trigger on behalf of a specified time base coming from the module TBU (selected by bits TBU_SEL) and the time stamp compare value defined in the bit field ACT_TB of register TOM[i]_TGC[y]_ACT_TB.

NOTE

A signed compare of ACT_TB and selected TBU_TS[x] with x = 0, 1, 2 is performed.

The third possibility is the input TRIG (bunch of trigger signals TRIGOUT[x]) coming from the TOM channels 0 to 7/8 to 15.

The corresponding trigger signal TRIGOUT[x] coming from channel [x] can be masked by the register TOM[i]_TGC[y]_INT_TRIG.

To enable or disable each individual TOM channel, the register TOM[i]_TGC[y]_ENDIS_CTRL and/or TOM[i]_TGC[y]_ENDIS_STAT have to be used.

The register TOM[i]_TGC[y]_ENDIS_STAT controls directly the signal ENDIS. A write access to this register is possible.

The register TOM[i]_TGC[y]_ENDIS_CTRL is a shadow register that overwrites the value of register TOM[i]_TGC[y]_ENDIS_STAT if signal CTRL_TRIG raises.

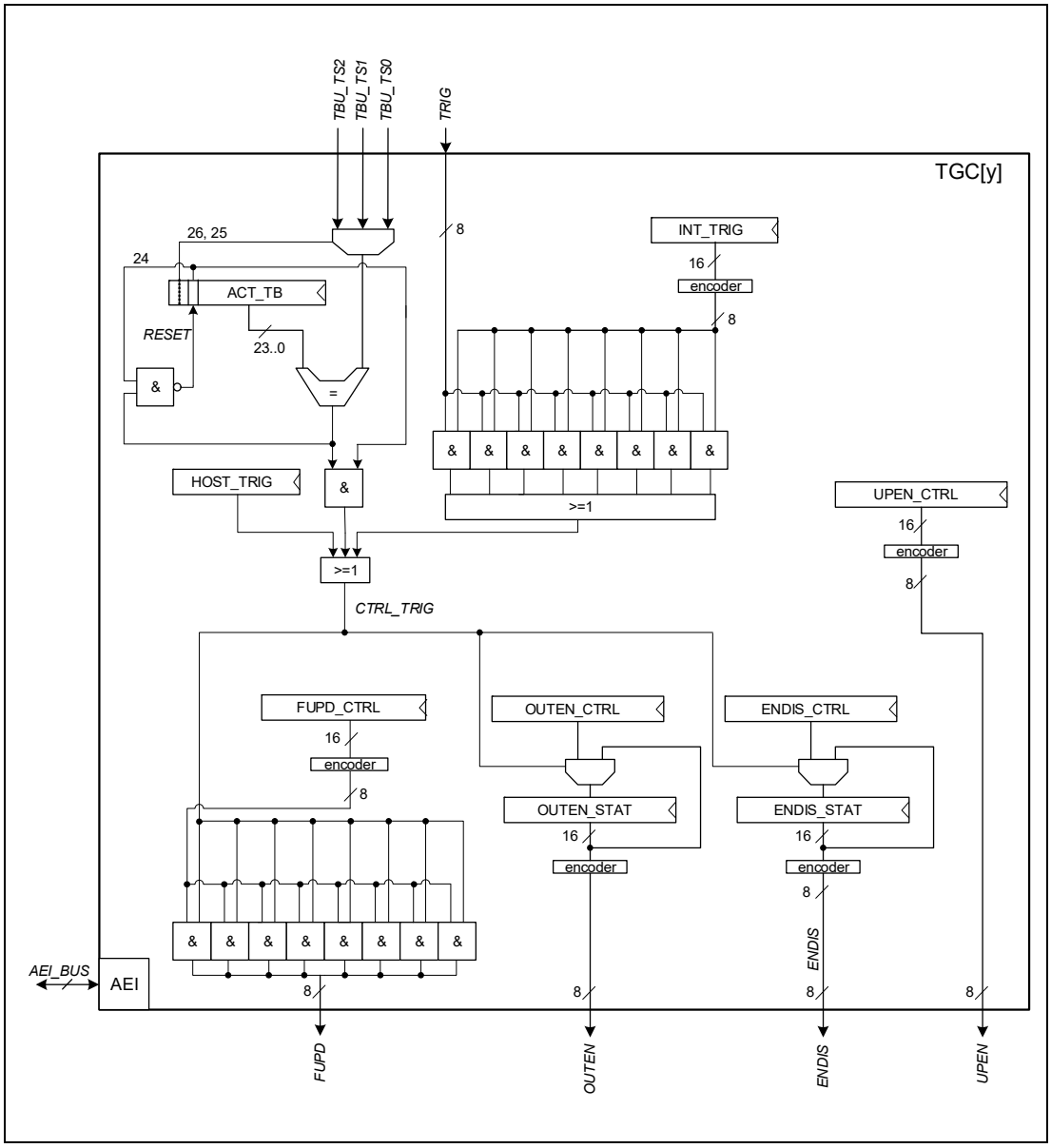


Figure 38.50 TOM Global channel control mechanism

The output of the individual TOM channels can be controlled using the register TOM[i]_TGC[y]_OUTEN_CTRL and TOM[i]_TGC[y]_OUTEN_STAT.

The register TOM[i]_TGC[y]_OUTEN_STAT controls directly the signal OUTEN. A write access to this register is possible.

The register TOM[i]_TGC[y]_OUTEN_CTRL is a shadow register that overwrites the value of register TOM[i]_TGC[y]_OUTEN_STAT if one of the three trigger conditions matches.

If a TOM channel is disabled by the register TOM[i]_TGC[y]_OUTEN_STAT, the actual value of the channel output at TOM_CH[x]_OUT is defined by the signal level bit (SL) defined in the channel control register TOM[i]_CH[x]_CTRL. If the output is enabled, the output at TOM_CH[x]_OUT depends on value of register SOUR (see **Figure 38.50**).

The register TOM[i]_TGC[y]_FUPD_CTRL defines which of the TOM channels receive a FORCE_UPDATE event if the trigger signal CTRL_TRIG is raised.

NOTE

The force update request is stored and executed synchronized to the selected CMU_FXCLK_EN.

The register bits UPEN_CTRL[z] defines for which TOM channel the update of the working register CM0, CM1 and CLK_SRC by the corresponding shadow register SR0, SR1 and CLK_SRC_SR is enabled. If update is enabled, the register CM0, CM1 and CLK_SRC will be updated on reset of counter register CN0 (see **Figure 38.52** and **Figure 38.53**). An exception is the configuration of SR0_TRIG = 1 which enable the trigger generation defined by SR0. Then CM0 is not updated with SR0.

38.16.3 TOM Channel

Each individual TOM channel comprises a Counter Compare Unit 0 (CCU0) which contains the counter register CN0 and the period register CM0, a Counter Compare Unit 1 (CCU1) which contains the duty cycle register CM1 and the Signal Output Generation Unit (SOU) which contains the output register SOUR (see **Figure 38.51**). The architecture is depicted in **Figure 38.51** for channels 0 to 7 and in **Figure 38.51** for channels 8 to 15.

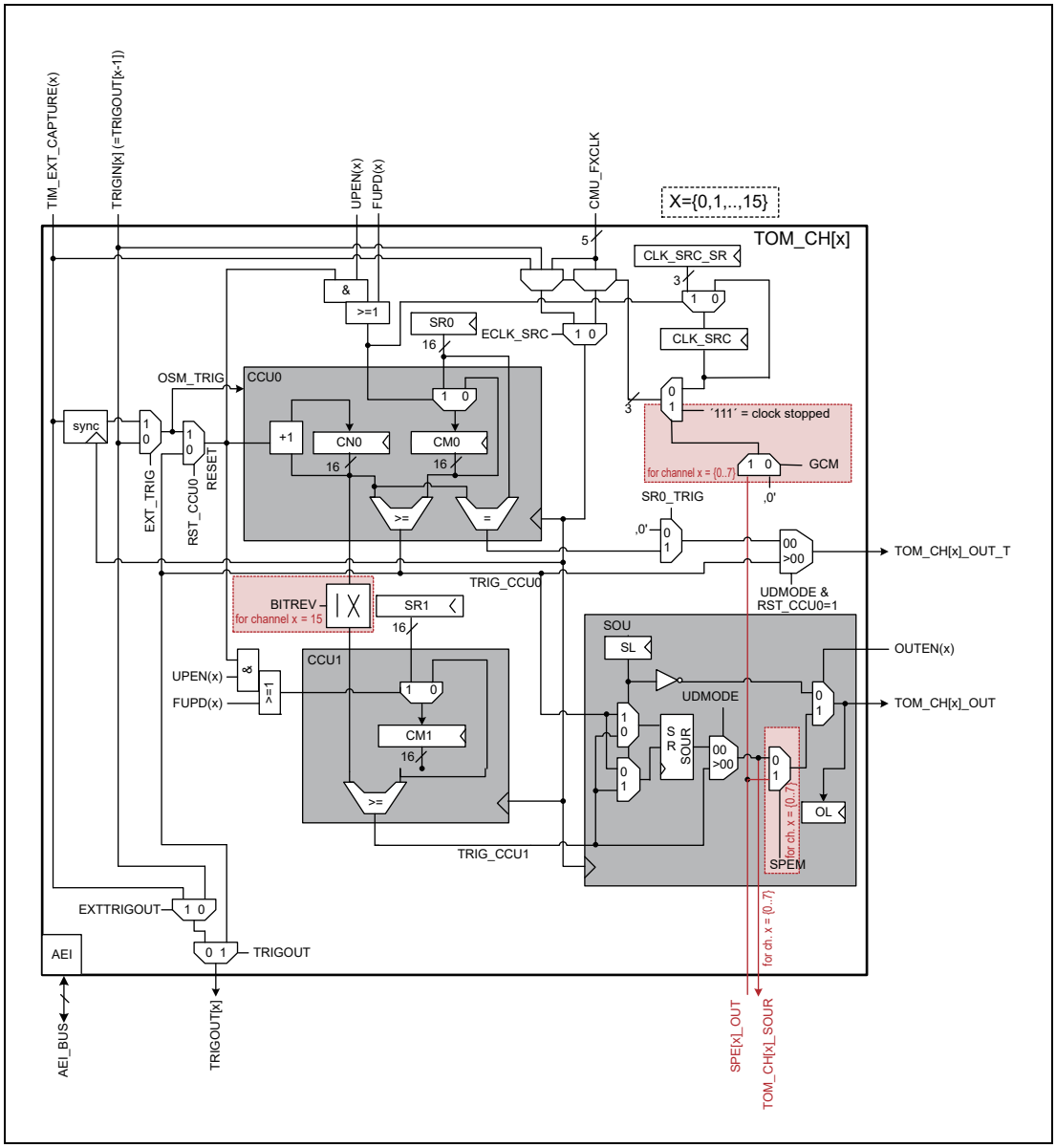


Figure 38.51 TOM Channel 0..15 architecture

The CCU0 contains a counter CN0 which is clocked with the selected clock enable signal from CMU_FXCLK_EN provided from outside of the sub-module.

Depending on configuration bits RST_CCU0 of register TOM[i]_CH[x]_CTRL the counter register CN0 can be reset either when the counter value is equal to the compare value CM0 (i.e. CN0 counts only from 0 to CM0-1 and is then reset to 0) or when signaled by the TOM[i] trigger signal TRIGIN[x] (=TRIGOUT[x-1] of the preceding channel [x-1] (which can also be the last channel of preceding instance TOM[i-1]) or the trigger signal TIM_EXT_CAPTURE(x) of the assigned TIM channel [x].

As an exception, the input TRIG_[0] of instance TOM0 is triggered by its own last channel x (x=15) via signal TRIGOUT[15].

When the counter register CN0 is greater or equal than the value CM0 (in fact CM0 -1), the sub-unit CCU0 triggers the SOU sub-unit and the succeeding TOM sub-module channel (signal TRIG_CCU0).

In the sub-unit CCU1 the counter register CN0 is compared with the value of register CM1. If CN0 is greater or equal than CM1 the sub-unit CCU1 triggers the SOU sub-unit (signal TRIG_CCU1).

If counter CN0 of channel x is reset by its own CCU0 unit (i.e. the compare match of CN0 ≥ CM0 -1 configured by RST_CCU0 = 0), following statements are valid:

1. CN0 counts from 0 to CM0 -1 and is then reset to 0.
2. When CN0 is reset from CM0 to 0, an edge to SL is generated.
3. When CN0 is incrementing and reaches CN0 > CM1, an edge to !SL is generated.
4. If CM0 = 0 or CM0 = 1, the counter CN0 is constant 0.
5. If CM1 = 0, the output is !SL = 0% duty cycle
6. If CM1 ≥ CM0 and CM0 >1, the output is SL = 100% duty cycle

If the counter register CN0 of channel x is reset by the trigger signal coming from another channel or the assigned TIM module (configured by RST_CCU0 = 1), following statements are valid:

1. CN0 counts from 0 to MAX-1 and is then reset to 0 by trigger signal
2. CM0 defines the edge to SL value, CM1 defines the edge to !SL value
3. If CM0 = CM1, the output switches to SL if CN0 = CM0 = CM1 (CM0 has higher priority)
4. If CM0 = 0 and CM1 = MAX, the output is SL = 100% duty cycle
5. If CM0 > MAX, the output is !SL = 0% duty cycle, independent of CM1

The hardware ensures that for both 0% and 100% duty cycle no glitch occurs at the output of the TOM channel.

The SOU sub-unit is responsible for output signal generation. On a trigger TRIG_CCU0 from sub-unit CCU0 or TRIG_CCU1 from sub-unit CCU1 an SR flip-flop of sub-unit SOU is either set or reset. If it is set or reset depends on the configuration bit SL of the control register TOM[i]_CH[x]_CTRL. The initial signal output level for the channel is the reverse value of the bit SL.

Figure 38.54 clarifies the PWM output behavior with respect to the SL bit definition.

The output level on the TOM channel output pin TOM[i]_CH[x]_OUT is captured in bit OL of register TOM[i]_CH[x]_STAT.

38.16.3.1 Duty cycle, Period and Clock Frequency Update Mechanisms

The two action register CM0 and CM1 can be reloaded with the content of the shadow register SR0 and SR1. The register CLK_SRC that determines the clock frequency of the counter register CN0 can be reloaded with its shadow register CLK_SRC_SR (bit field in register TOM[i]_CH[x]_CTRL).

The update of the register CM0, CM1 and CLK_SRC with the content of its shadow register is done when the reset of the counter register CN0 is requested (via signal RESET). This reset of CN0 is done if the comparison of CN0 greater or equal than CM0 is true or when the reset is triggered by another TOM channel [x-1] via the signal TRIGIN[x] (=TRIGOUT[x-1]) or when signaled via the signal TIM_EXT_CAPTURE(x) of the assigned TIM channel [x].

With the update of the register CLK_SRC at the end of a period a new counter CN0 clock frequency can easily be adjusted.

In case of RST_CCU0=1 and update enabled by UPEN_CTRL[z] the register CM0, CM1 and CLK_SRC will be updated when CN0 is reset.

An update of duty cycle, period and counter CN0 clock frequency becoming effective synchronously with start of a new period can easily be reached by performing following steps:

1. Disable the update of the action register with the content of the corresponding shadow register by setting the channel specific configuration bit UPEN_CTRL[z] of register TOM[i]_TGC[y]_GLB_CTRL to '0'.
2. Write new desired values to SR0, SR1, CLK_SRC_SR
3. Enable update of the action register by setting the channel specific configuration bit UPEN_CTRL[z] of register TOM[i]_TGC[y]_GLB_CTRL to '1'.

(1) Synchronous Update Of Duty Cycle Only

A synchronous update of only the duty cycle can be done by simply writing the desired new value to register SR1 without preceding disable of the update mechanism (as described in the chapter above). The new duty cycle is then applied in the period following the period where the update of register SR1 was done.

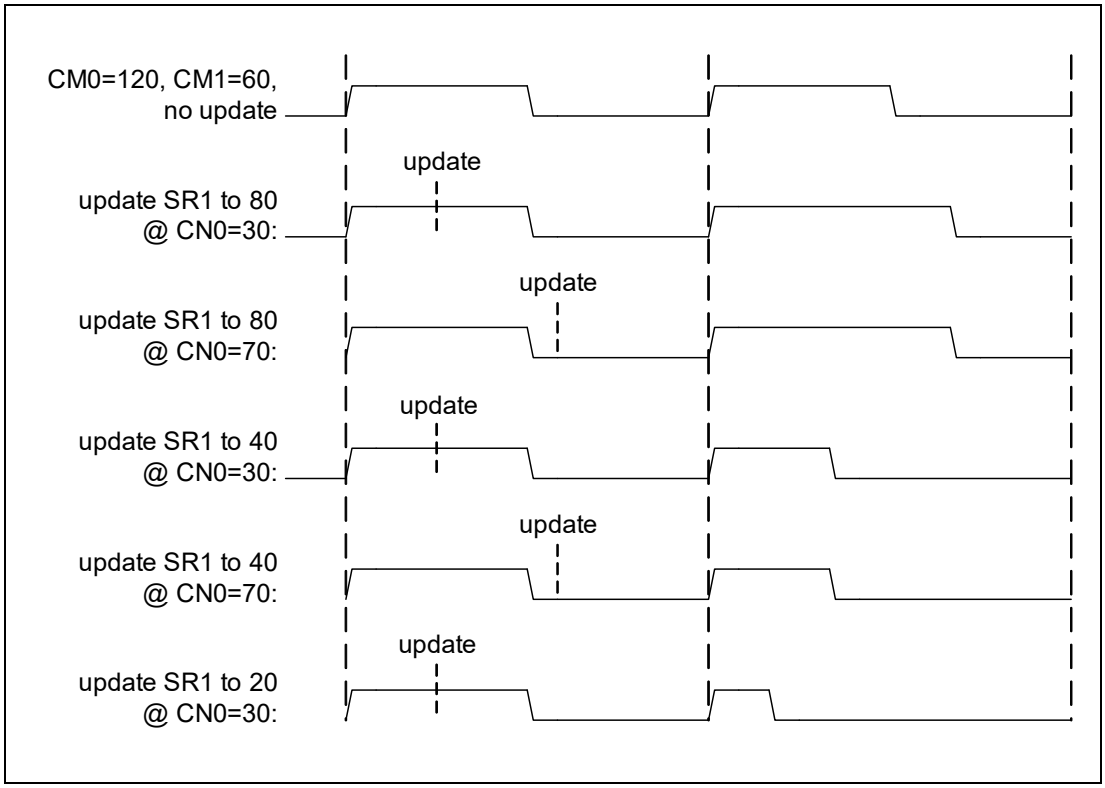


Figure 38.52 Synchronous update of duty cycle

(2) Asynchronous Update Of Duty Cycle Only

If the update of the duty cycle should be performed independent of the start of a new period (asynchronous), the desired new value can be written directly to register CM1. In this case it is recommended to additionally either disable the synchronous update mechanism as a whole (i.e. clearing bits UPEN_CTRL[z] of corresponding channel [x] in register TOM[i]_TGX[y]_GLB_CTRL) or updating SR1 with the same value as CM1 before writing to CM1. Depending on the point of time of the update of CM1 in relation to the actual value of CN0 and CM1, the new duty cycle is applied in the current period or the following period (see **Figure 38.52**). In any case the creation of glitches are avoided. The new duty cycle may jitter from update to update by a maximum of one period (given by CM0). However, the period remains unchanged.

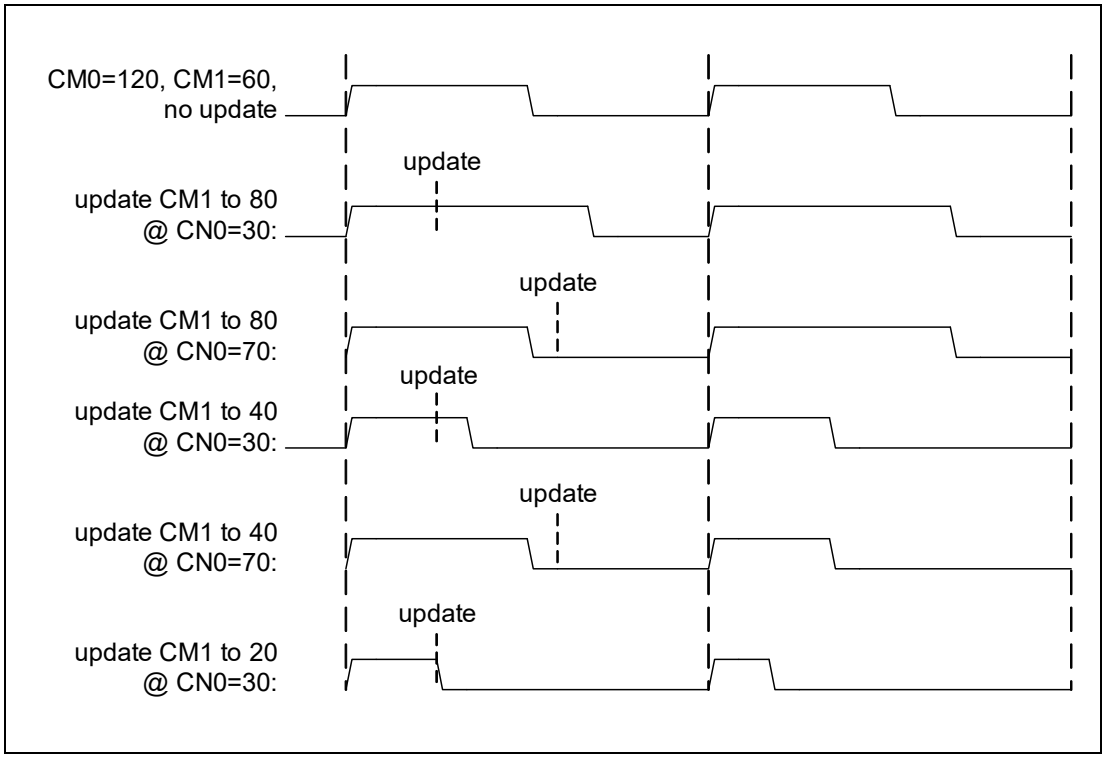


Figure 38.53 Asynchronous update of duty cycle

38.16.3.2 Continuous Counting Up Mode

In continuous mode the TOM channel starts incrementing the counter register CN0 once it is enabled by setting the corresponding bits in register TOM[i]_TGC[y]_ENDIS_STAT (refer to **Section 38.16.2.2** for details of enabling a TOM channel).

The signal level of the generated output signal can be configured with the configuration bit SL of the channel configuration register TOM[i]_CH[x]_CTRL.

If the counter CN0 is reset from CM0 back to zero, the first edge of a period is generated at TOM[i]_CH[x]_OUT.

The second edge of the period is generated if CN0 has reached CM1.

Every time the counter CN0 has reached the value of CM0 it is reset back to zero and proceeds with incrementing.

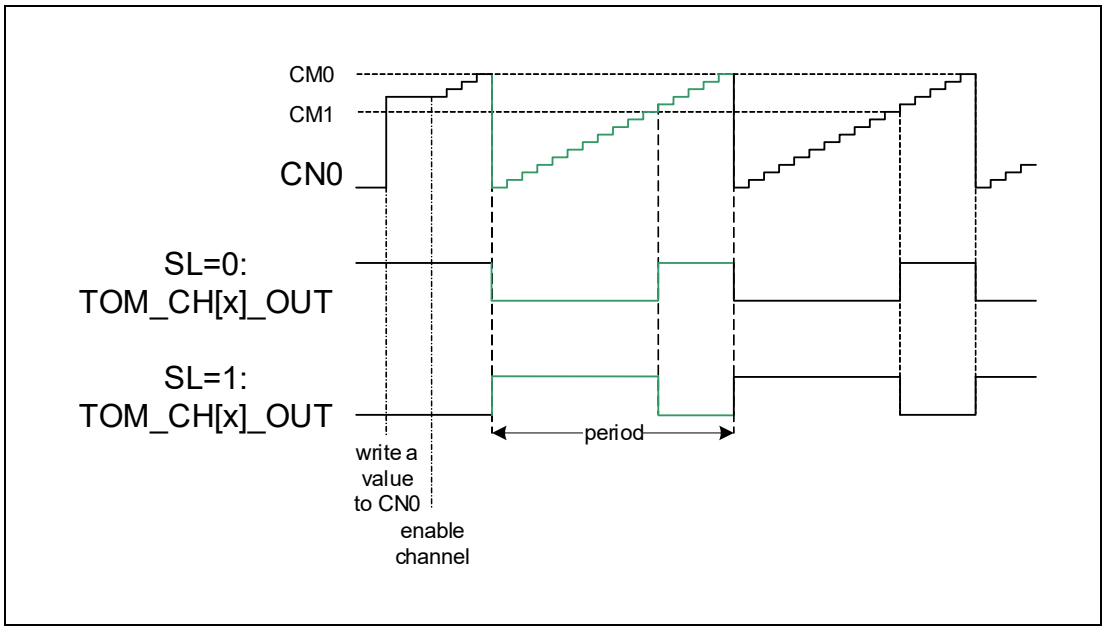


Figure 38.54 PWM Output with respect to configuration bit SL in continuous mode

38.16.3.3 Continuous Counting Up-Down Mode

In continuous mode, if CN0 counts up and down (UDMODE! = 00), depending on configuration bits RST_CCU0 of register TOM[i]_CH[x]_CTRL the counter register CN0 changes direction either when the counter value is equal to the compare value CM0, has counted down to 0 or when triggered by the TOM[i] trigger signal TRIGIN[x] (= TRIGOUT[x-1]) of the preceding channel [x-1] (which can also be the last channel of preceding instance TOM[i-1]) or the trigger signal TIM_EXT_CAPTURE(x) of the assigned TIM channel [x]. In this case, if UPEN_CTRL[x] = 1, also the working register CM0, CM1 and CLK_SRC are updated depending on UDMODE.

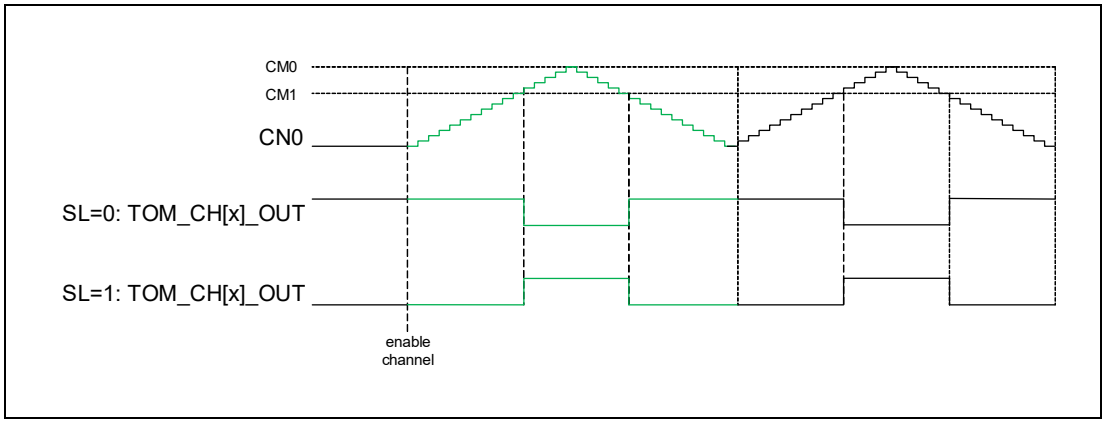


Figure 38.55 PWM Output behavior with respect to the SL bit in the TOM[i]_CH[x]_CTRL register if UDMODE! = 00_B

The clock of the counter register CN0 can be one of the CMU clocks CMU_FXCLK_EN[x]. The clock for CN0 is defined by CLK_SRC_SR value in register TOM[i]_CH[x]_CTRL. The duration of a period in multiples of selected CN0 counter clock ticks is defined by the CM0 configuration value (i.e. CM0 defines half of period in up-down mode). CM1 defines the duty cycle value in clock ticks of selected CN0 counter clock (i.e. CM0 defines half of duty cycle in up-down mode).

If counter CN0 of channel x is reset by its own CCU0 unit (i.e. the compare match of CN0 ≥ CM0 configured by RST_CCU0 = 0), following statements are valid:

1. CN0 counts continuously first up from 0 to CM0-1 and then down to 0.
2. If CN0 ≥ CM1, the output is set to SL
3. If CM1 = 0, the output is SL (i.e. 100% duty cycle)
4. If CM1 ≥ CM0, the output is !SL (i.e. 0% duty cycle)
5. On output TOM[i]_CH[x]_OUT a PWM signal is generated. The period is defined by CM0, the duty cycle is defined by CM1.

This behavior is depicted in **Figure 38.55**.

If the counter register CN0 of channel x is reset by the trigger signal coming from another channel or the assigned TIM module (configured by RST_CCU0 = 1), following statements are valid:

1. CN0 counts continuously first up. On a trigger signal the counter switches to count down mode. If CN0 has reached 0, it switches to count up mode.
2. If CN0 ≥ CM1, the output is set to SL
3. If CM1 = 0, the output is SL (i.e. 100% duty cycle)
4. If CM1 ≥ CM0, the output is !SL (i.e. 0% duty cycle)

5. On output TOM[i]_CHx]_OUT a PWM signal is generated. The period is defined by the CCU0 trigger of triggering channel, the duty cycle is defined by CM1.
6. On output TOM[i]_CHx]_OUT_T a PWM signal is generated. The period is defined by the CCU0 trigger of triggering channel, the duty cycle is defined by CM0.

This behavior is depicted in **Figure 38.56**.

NOTE

In case of up-down counter mode and RST_CCU0 = 1 it is recommended that

1. The triggering channel and the triggered channel are both running in up–down mode
2. The time between two trigger signals is equal to the time needed for CN0 of triggered channel to count back to 0 and again up to the same upper value.

The second recommendation can be reached by synchronizing the start of triggering channel and of triggered channel, i.e. let both channel start with a CN0 value 0.

NOTE

If there is a synchronization register in the trigger chain (indicated by value TOM_TRIG_CHAIN in register CCM[i]_HW_CONF), the additional delay of the trigger by one clock period has to be taken into account by starting at triggering channel with a CN0 value 1 (+1 compared to CN0 of triggered channel).

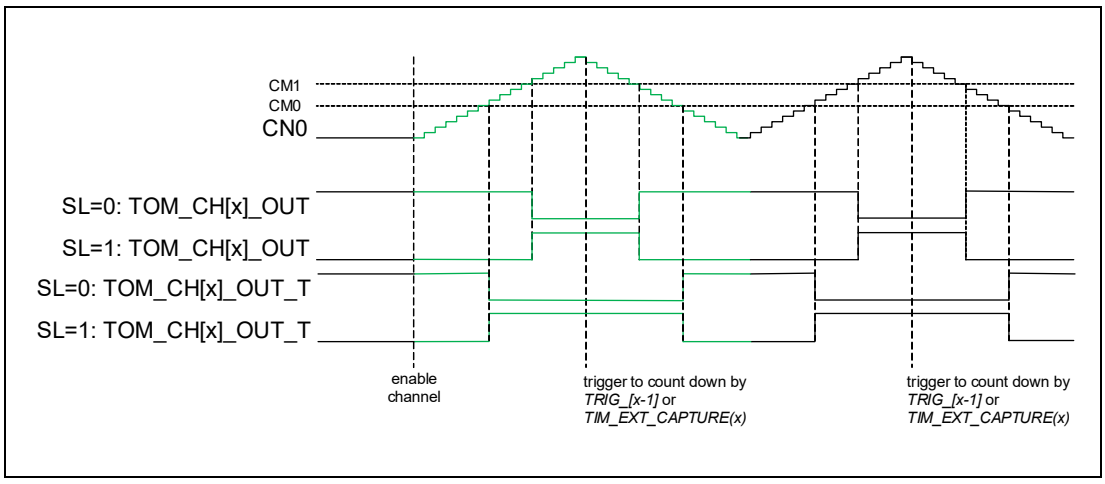


Figure 38.56 PWM Output behavior in case of RST_CCU0 = 1 and UDMODE! = 00

38.16.3.4 One-shot Counting Up Mode

In one-shot mode, the TOM channel generates one pulse with a signal level specified by the configuration bit SL in the channel [x] configuration register TOM[i]_CH[x]_CTRL.

First the channel has to be enabled by setting the corresponding TOM[i]_TGC[y]_ENDIS_STAT value and the one-shot mode has to be enabled by setting bit OSM in register TOM[i]_CH[x]_CTRL.

In one-shot mode the counter CN0 will not be incremented once the channel is enabled.

A write access to the register CN0 triggers the start of pulse generation (i.e. the increment of the counter register CN0).

If SPE mode of TOM[i] channel 2 is enabled (set bit SPEM of register TOM[i]_CH2_CTRL), also the trigger signal SPE[i]_NIPD can trigger the reset of register CN0 to zero and a start of the pulse generation.

The new value of CN0 determines the start delay of the first edge. The delay time of the first edge is given by (CM0-CN0) multiplied with period defined by current value of CLK_SRC.

If the counter CN0 is reset from CM0 back to zero, the first edge at TOM[i]_CH[x]_OUT is generated.

To avoid an update of CMx register with content of SRx register at this point in time, the automatic update should be disabled by setting UPEN_CTRL[x] = 00 (in register TOM[i]_TGC[y]_GLB_CTRL)

The second edge is generated if CN0 is greater or equal than CM1 (i.e. CN0 was incremented until it has reached CM1 or CN0 is greater than CM1 after an update of CM1).

If the counter CN0 has reached the value of CM0 a second time, the counter stops.

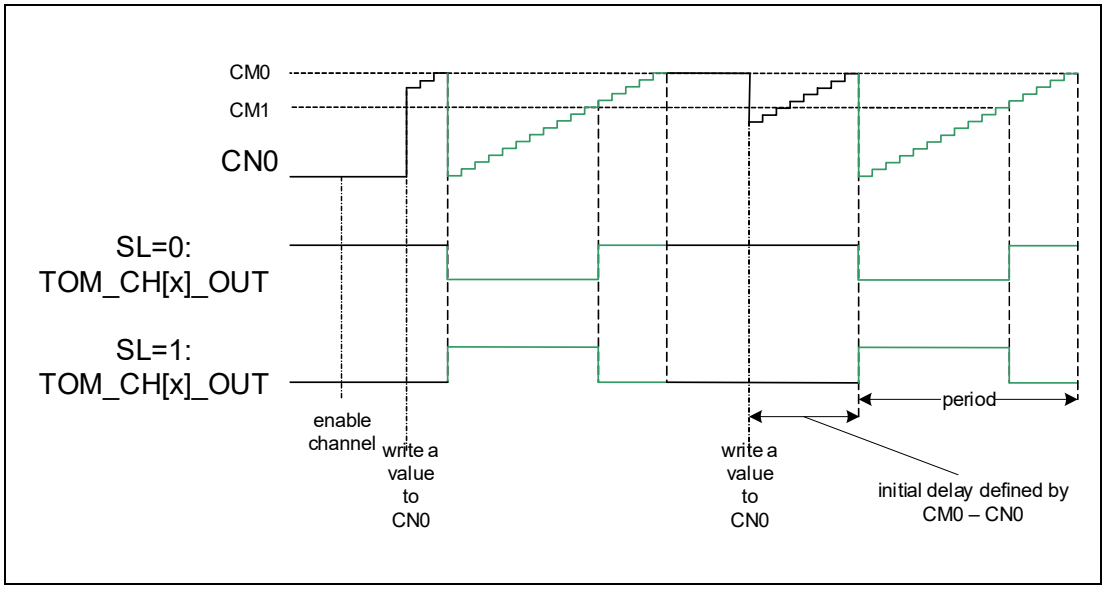


Figure 38.57 PWM Output with respect to configuration bit SL in one-shot mode: trigger by writing to CN0

Further output of single periods can be started by a write access to register CN0.

If CN0 is already incrementing (i.e. started by writing to CN0 a value CN0start < CM0), the affect of a second write access to CN0 depends on the phase of CN0:

phase 1: Update of CN0 before CN0 reaches first time CM0

phase 2: Update of CN0 after CN0 has reached first time CM0 but is less than CM1

phase 3: Update of CN0 after CN0 has reached first time CM0 and CN0 is greater than or equal CM1

In phase 1: Writing to counter CN0 a value CN0new < CM0 leads to a shift of first edge (generated if CN0 reaches CM0 first time) by the time CM0-CN0new.

In phase 2: Writing to incrementing counter CN0 a value CN0new < CM1 while CN0old is below CM1 leads to a lengthening of the pulse. The counter CN0 stops if it reaches CM0.

In phase 3: Writing to incrementing counter CN0 a value CN0new while CN0old is already greater than or equal CM1 leads to an immediate restart of a single pulse generation inclusive the initial delay defined by CM0 - CN0new.

If a channel is configured to one-shot mode and configuration bit OSM_TRIG is set to 1, the trigger signal OSM_TRIG (i.e. TRIGIN[x] or TIM_EXT_CAPTURE(x)) triggers start of one pulse generation.

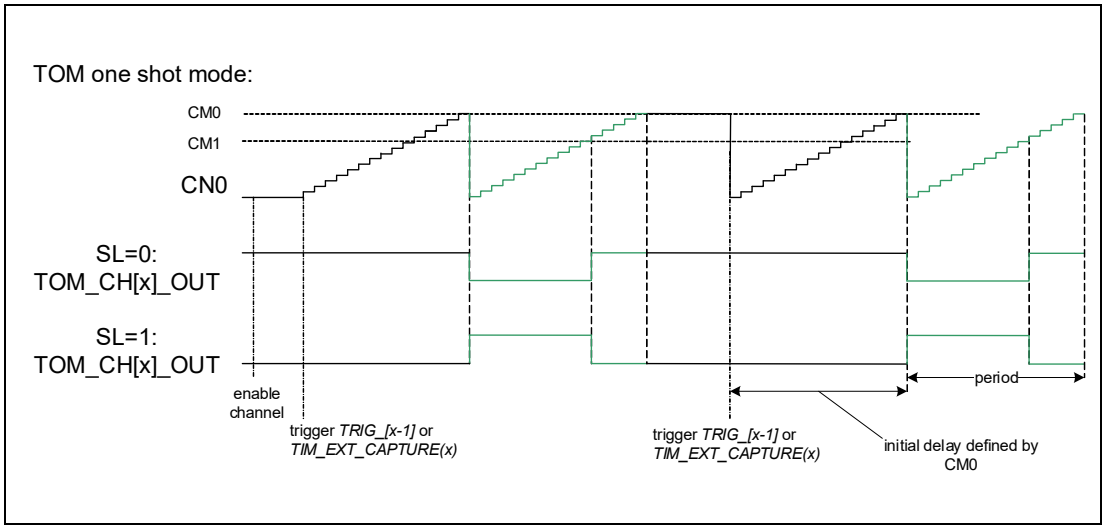


Figure 38.58 PWM Output with respect to configuration bit SL in one-shot mode: trigger by TRIGIN[x]/TRIGOUT[x-1] or TIM_EXT_CAPTURE(x)

38.16.3.5 One-shot Counting Up-Down Mode

The TOM channel can operate in one-shot counting up-down mode when the bit OSM = 1 and the UDMODE! = 00. One-shot mode means that a single pulse with the pulse level defined in bit SL is generated on the output line.

First the channel has to be enabled by setting the corresponding ENDIS_STAT value.

In One-shot mode the counter CN0 will not be incremented once the channel is enabled.

A write access to the register CN0 triggers the start of pulse generation (i.e. the increment of the counter register CN0).

To avoid an update of CMx register with content of SRx register at this point in time, the automatic update should be disabled by setting UPEN_CTRL[x] = 00 (in register TOM[i]_CH[x]_CTRL)

If the counter CN0 is greater or equal than CM1, the output TOM[i]_CH[x]_OUT is set to SL value.

If the counter CN0 is less than CM1, the output TOM[i]_CH[x]_OUT is set to !SL value.

If the counter CN0 has reached the value 0 (by counting down), it stops.

The new value of CN0 determines the start delay of the first edge. The delay time of the first edge is given by (CM1-CN0) multiplied with period defined by current value of CLK_SRC.

Figure 38.59 depicts the pulse generation in one-shot mode.

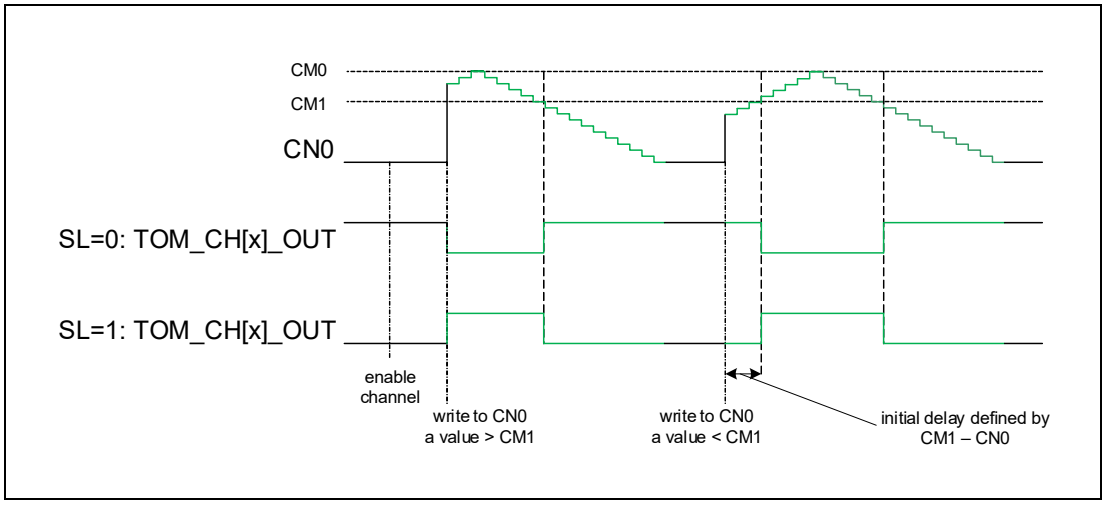


Figure 38.59 PWM Output with respect to configuration bit SL in one-shot counting up-down mode and UDMODE! = 00: trigger by writing to CN0

Further output of single pulses can be started by writing to register CN0.

If a channel is configured to one-shot counting up–down mode and configuration bit OSM_TRIG is set to 1, the trigger signal OSM_TRIG (i.e. TRIGIN[x] or TIM_EXT_CAPTURE(x)) triggers start of one pulse generation.

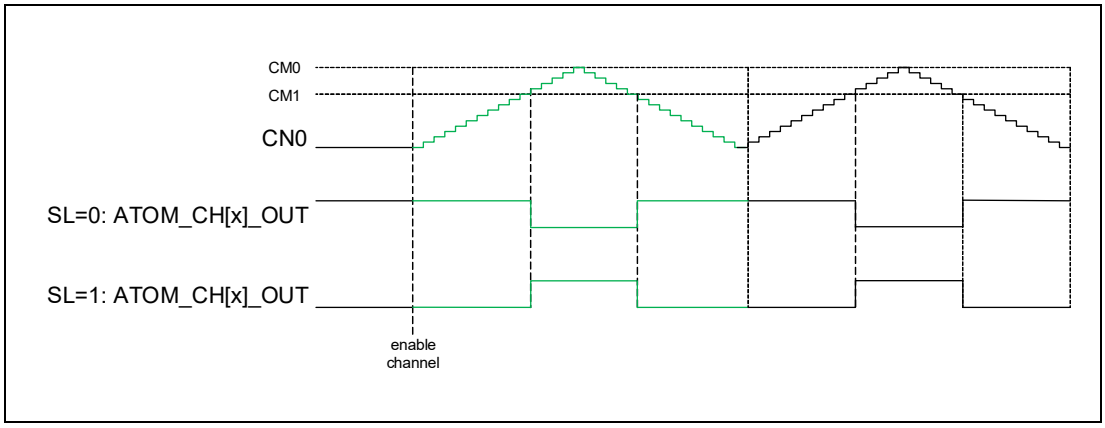


Figure 38.60 PWM Output with respect to configuration bit SL in one-shot counting up-down mode and UDMODE! = 00: trigger by TRIGIN[x-1] or TIM_EXT_CAPTURE(x)

38.16.3.6 Pulse Count Modulation Mode

At the output TOM[i]_CH15_OUT a pulse count modulated signal can be generated instead of the simple PWM output signal.

Figure 38.61 outlines the circuit for Pulse Count Modulation.

The PCM mode is enabled by setting bit BITREV to 1.

With the configuration bit BITREV =1 a bit-reversing of the counter output CN0 is configured. In this case the bits LSB and MSB are swapped, the bits LSB+1 and MSB–1 are swapped, the bits LSB+2 and MSB-2 are swapped and so on.

The effect of bit-reversing of the CN0 register value is shown in the following **Figure 38.61**.

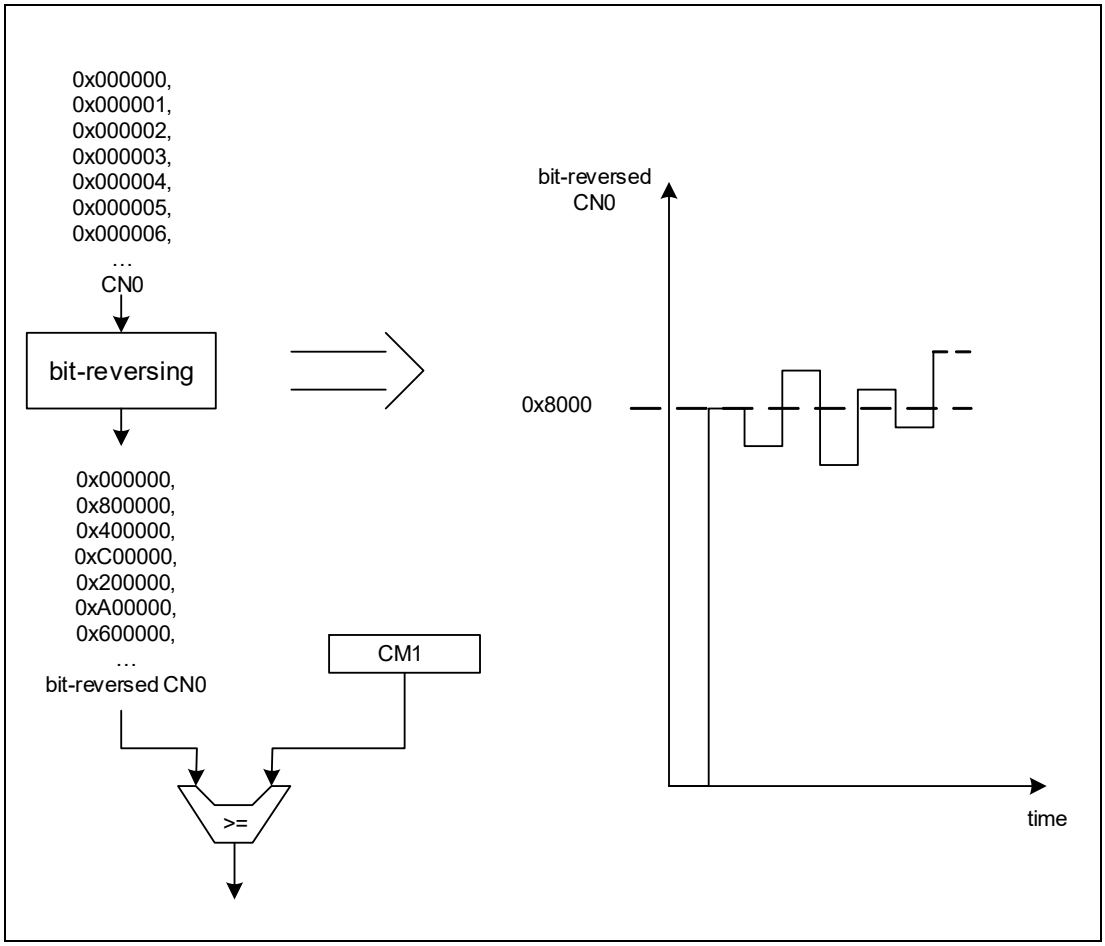


Figure 38.61 Bit reversing of counter CN0 output

In the PCM mode the counter register CN0 is incremented by every clock tick depending on configured CMU clock (CMU_FXCLK).

The output of counter register CN0 is first bit-reversed and then compared with the configured register value CM1.

If the bit-reversed value of register CN0 is greater than CM1, the SR flip-flop of sub-module SOU is set (depending on configuration register SL) otherwise the SR flip-flop is reset. This generates at the output TOM[i]_CH15_OUT a pulse count modulated signal.

In PCM mode the CM0 register - in which the period is defined - normally has to be set to its maximum value $FFFF_H$.

To reduce time period of updating duty cycle value in CM1 register, it is additionally possible to setup period value in CM0 register to smaller values than maximum value as described before. Possible values for CM0 register are each even numbered values to the power of 2 e.g. 8000_H, 4000_H, 2000_H .... In this case the duty cycle has to be configured in the following manner. Depending on how much the period in CM0 register is decreased - means shifted right starting from 10000_H - the duty cycle in CM1 register has to be shifted left (= rotated: shift MSB back into LSB) with same value, e.g. :

Period CM0 = 0100_H → shifted 8 bits right from 10000_H

→ so duty cycle has to be shifted left 8 bit:

e.g. 50% duty cycle = 00080_H → shift 8 bits left→ CM1 = 8000_H

More examples:

period CM0	→	duty cycle	→	shift	→	CM1
$FFFF_H$	→	8000_H	→	no shift	→	8000_H
8000_H	→	4000_H	→	shift 1 bit left	→	8000_H
4000_H	→	1000_H	→	shift 2 bits left	→	4000_H
2000_H	→	$0FFF_H$	→	shift 3 bits left	→	$7FF8_H$
1000_H	→	0333_H	→	shift 4 bits left	→	3330_H
0800_H	→	0055_H	→	shift 5 bits left	→	$0AA0_H$
...						
0020_H	→	0008_H	→	shift 11 bits left	→	4000_H
0010_H	→	0005_H	→	shift 12 bits left	→	5000_H
...						

NOTE

In this mode the interrupt CCU1TC (see register TOM[i]_CH[x]_IRQ_NOTIFY) is set every time if bit reverse value of CN0 is greater or equal than CM1 which may be multiple times during one period. Therefore, from application point of view it is not useful to enable this interrupt.

38.16.3.7 Trigger Generation

For applications with constant PWM period defined by CM0, it is not necessary to update regularly the CM0 register with SR0 register. For these applications the SR0 register can be used to define an additional output signal and interrupt trigger event. If bit SR0_TRIG in register TOM[i]_CH[x]_CTRL is set, the register SR0 is no longer used as a shadow register for register CM0. Instead, SR0 is compared against CN0 and if both are equal, a pulse of signal level '1' is generated at the output TOM[i]_CH[x]_OUT_T. The bit SR0_TRIG should only be set if bit RST_CCU0 of this channel is 0.

If bit SR0_TRIG is set the interrupt notify flag CCU1TC is no longer set on a compare match of CM1 and CN0. Instead, the CCU1TC interrupt notify flag is set in case of a compare equal match of SR0 and CN0.

With configuration bit TRIG_PULSE one can select if the output TOM[i]_CH[x]_OUT_T is high as long as CN0 = SR0 (TRIG_PULSE = 0) or if there will be only one pulse of length one SYS_CLK period when CN0 becomes SR0 (TRIG_PULSE = 1).

The TOM output signal routing to DTM or GTM-IP top level is described in **Section 38.18.9, DTM Configuration Register Description**.

38.16.4 TOM BLDC Support

The TOM sub-module offers in combination with the SPE sub-module a BLDC support. To drive a BLDC engine TOM channels 0 to 7 can be used.

The BLDC support can be configured by setting the SPEM bit inside the TOM[i]_CH[z]_CTRL register. When this bit is set the TOM channel output is controlled through the SPE_OUT(z) signal coming from the SPE sub-module (see **Figure 38.51**). Refer to **Section 38.23, Sensor Pattern Evaluation (SPE)** for a detailed description of the SPE sub-module.

The TOM[i]_CH2, 6, 7, 8 or 9 can be used together with the SPE module to trigger a delayed update of the SPE_OUT_CTRL register (i.e.commutation delay) after new input pattern detected by SPE (signaled by SPE[i]_NIPD). This feature is configured on TOM[i]_CH2, 6, 7, 8 or 9 by setting SPE_TRIG = 1 and OSM = 1. With this configuration the TOM channel i generates one single PWM pulse on trigger by signal SPE_NIPD.

For details, refer to chapter of SPE sub-module description.

38.16.5 TOM Gated Counter Mode

Each TOM - SPE module combination provides also the feature of a gated counter mode. This is reached by using the FSOI input of a TIM module to gate the clock of a CCU0 sub-module. To configure this mode, register of module SPE should be set as following:

1. The SPE should be enabled (bit SPE_EN = 1),
2. All three TIM inputs should be disabled (SIE0 = SIE1 = SIE2 = 0),
3. SPE[i]_OUT_CTRL should be set to 00005555_H (set SPE_OUT() to '0'),
4. Mode FSOM should be enabled (FSOM = 1),
5. Set in bit field FSOL bit c if channel c of module TOM is chosen for gated counter mode

Additionally in module TOM

1. The SPE mode should be disabled (SPEM = 0) and
2. The gated counter mode should be enabled (GCM = 1)

As a result of this configuration, the counter CN0 in sub-module CCU0 of TOM channel c counts as long as input FSOI is '0'.

38.16.6 TOM Interrupt Signals

Signal	Description
TOM_CCU0TCx_IRQ	CCU0 Trigger condition interrupt for channel x
TOM_CCU1TCx_IRQ	CCU1 Trigger condition interrupt for channel x

38.16.7 TOM Configuration Registers Overview

TOM contains following configuration registers:

Table 38.177 Register list

Symbol	Register Name	Detail in Section
TOM[i]_TGC[y]_GLB_CTRL	TOMi TGC y global control register	**38.16.8.1**
TOM[i]_TGC[y]_ENDIS_CTRL	TOMi TGC y enable/disable control register	**38.16.8.2**
TOM[i]_TGC[y]_ENDIS_STAT	TOMi TGC y enable/disable status register	**38.16.8.3**
TOM[i]_TGC[y]_ACT_TB	TOMi TGC y action time base register	**38.16.8.4**
TOM[i]_TGC[y]_OUTEN_CTRL	TOMi TGC y output enable control register	**38.16.8.5**
TOM[i]_TGC[y]_OUTEN_STAT	TOMi TGC y output enable status register	**38.16.8.6**
TOM[i]_TGC[y]_FUPD_CTRL	TOMi TGC y force update control register	**38.16.8.7**
TOM[i]_TGC[y]_INT_TRIG	TOMi TGC y internal trigger control register	**38.16.8.8**
TOM[i]_CH[x]_CTRL	TOMi channel x control register	**38.16.8.9**
TOM[i]_CH[x]_CN0	TOMi channel x CCU0 counter register	**38.16.8.10**
TOM[i]_CH[x]_CM0	TOMi channel x CCU0 compare register	**38.16.8.11**
TOM[i]_CH[x]_SR0	TOMi channel x CCU0 compare shadow register	**38.16.8.12**
TOM[i]_CH[x]_CM1	TOMi channel x CCU1 compare register	**38.16.8.13**
TOM[i]_CH[x]_SR1	TOMi channel x CCU1 compare shadow register	**38.16.8.14**
TOM[i]_CH[x]_STAT	TOMi channel x status register	**38.16.8.15**
TOM[i]_CH[x]_IRQ_NOTIFY	TOMi channel x interrupt notification register	**38.16.8.16**
TOM[i]_CH[x]_IRQ_EN	TOMi channel x interrupt enable register	**38.16.8.17**
TOM[i]_CH[x]_IRQ_FORCINT	TOMi channel x force interrupt register	**38.16.8.18**
TOM[i]_CH[x]_IRQ_MODE	TOMi channel x interrupt mode register	**38.16.8.19**

38.16.8 TOM Configuration Registers Description

38.16.8.1 TOM[i]_TGC[y]_GLB_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8030_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UPEN_CTRL7		UPEN_CTRL6		UPEN_CTRL5		UPEN_CTRL4		UPEN_CTRL3		UPEN_CTRL2		UPEN_CTRL1		UPEN_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0	—	—	—	—	—	—	—	HOST_TRIG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 38.178 TOM[i]_TGC[y]_GLB_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	UPEN_CTRL[7:0]	TOM channel k enable update of register CM0, CM1 and CLK_SRC from SR0, SR1 and CLK_SRC_SR. READ access: 00_B: Update disabled 01_B: — 10_B: — 11_B: Update enabled WRITE access: 00_B: Don't care, bits 1:0 will not be changed 01_B: Disable update 10_B: Enable update 11_B: Don't care, bits 1:0 will not be changed
15 to 8	RST_CH[7:0]	Software reset of channel k 0: No action 1: Reset channel **NOTE** This bit is cleared automatically after write by CPU. The channel register are set to their reset values and channel operation is stopped immediately. The SR flip-flop SOUR is set to '1'.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	HOST_TRIG	Trigger request signal (see TGC0, TGC1) to update the register ENDIS_STAT and OUTEN_STAT 0: no trigger request 1: set trigger request **NOTE** This flag is reset automatically after triggering the update

38.16.8.2 TOM[i]_TGC[y]_ENDIS_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8070_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIS_CTRL7		ENDIS_CTRL6		ENDIS_CTRL5		ENDIS_CTRL4		ENDIS_CTRL3		ENDIS_CTRL2		ENDIS_CTRL1		ENDIS_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.179 TOM[i]_TGC[y]_ENDIS_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ENDIS_CTRL[7:0]	TOM channel k enable/disable update value. If FREEZE = 0: If a TOM channel is disabled, the counter CN0 is stopped and the output register of SOU unit is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If FREEZE = 1: If a TOM channel is disabled, the counter CN0 is stopped. On an enable event, the counter CN0 starts counting from its current value. Write of following double bit values is possible: 00_B: Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger 01_B: Disable channel on an update trigger 10_B: Enable channel on an update trigger 11_B: Don't change bits 1:0 of this register **NOTE** If the output is disabled (OUTEN[0] = 0), the TOM channel 0 output TOM[i]_CH0_OUT is the inverted value of bit SL.

38.16.8.3 TOM[i]_TGC[y]_ENDIS_STAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8074_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIS_STAT7		ENDIS_STAT6		ENDIS_STAT5		ENDIS_STAT4		ENDIS_STAT3		ENDIS_STAT2		ENDIS_STAT1		ENDIS_STAT0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.180 TOM[i]_TGC[y]_ENDIS_STAT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ENDIS_STAT[k]	TOM channel k enable/disable If FREEZE = 0: If a TOM channel is disabled, the counter CN0 is stopped and the output register of SOU unit is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. If FREEZE=1: If a TOM channel is disabled, the counter CN0 is stopped. On an enable event, the counter CN0 starts counting from its current value. Write / Read: 00_B: Don't care, bits 1:0 will not be changed / channel disabled 01_B: Disable channel / — 10_B: Enable channel / — 11_B: Don't care, bits 1:0 will not be changed / channel enabled

38.16.8.4 TOM[i]_TGC[y]_ACT_TB

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8034_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TBU_SEL		TB_TRIG	ACT_TB							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACT_TB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.181 TOM[i]_TGC[y]_ACT_TB Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26, 25	TBU_SEL	Selection of time base used for comparison 00_B: TBU_TS0 selected 01_B: TBU_TS1 selected 10_B: TBU_TS2 selected 11_B: Same as 00_B **NOTE** The bit combination 10 is only applicable if the TBU of the device contains three time base channels. Otherwise, this bit combination is also reserved. Refer to **Figure 38.2, GTM Architecture Block Diagram** to determine the number of channels for TBU of this device.
24	TB_TRIG	Set trigger request 0: No trigger request 1: Set trigger request **NOTE** This flag is reset automatically if the selected time base unit (TBU_TS0 or TBU_TS1 or TBU_TS2 if present) has reached the value ACT_TB and the update of the register were triggered.
23 to 0	ACT_TB	Specifies the signed compare value with selected signal TBU_TS[x], x = 0 to 2. If selected TBU_TS[x] value is in the interval [ACT_TB–$007FFFFF_H$, ACT_TB] the event is in the past and the trigger is generated immediately. Otherwise the event is in the future and the trigger is generated if selected TBU_TS[x] is equal to ACT_TB.

38.16.8.5 TOM[i]_TGC[y]_OUTEN_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8078_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUTEN_CTRL7		OUTEN_CTRL6		OUTEN_CTRL5		OUTEN_CTRL4		OUTEN_CTRL3		OUTEN_CTRL2		OUTEN_CTRL1		OUTEN_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.182 TOM[i]_TGC[y]_OUTEN_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	OUTEN_CTRL [7:0]	Output TOM[i]_CH[k]_OUT enable/disable update value 00_B: Don't care, bits 1:0 of register OUTEN_STAT will not be changed on an update trigger 01_B: Disable channel output on an update trigger 10_B: Enable channel output on an update trigger 11_B: Don't change bits 1:0 of this register **NOTE** If the channel is disabled (ENDIS[0] = 0) or the output is disabled (OUTEN[0] = 0), the TOM channel 0 output TOM[i]_CH0_OUT is the inverted value of bit SL.

38.16.8.6 TOM[i]_TGC[y]_OUTEN_STAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $8078C_H$ + (800_H × i) + (200_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUTEN_STAT7		OUTEN_STAT6		OUTEN_STAT5		OUTEN_STAT4		OUTEN_STAT3		OUTEN_STAT2		OUTEN_STAT1		OUTEN_STAT0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.183 TOM[i]_TGC[y]_OUTEN_STAT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	OUTEN_STAT [7:0]	Control/status of output TOM[i]_CH[7:0]_OUT READ access: 00_B: Output disabled 01_B: — 10_B: — 11_B: Output enabled WRITE access: 00_B: Don't care, bits 1:0 will not be changed 01_B: Disable output 10_B: Enable output 11_B: Don't care, bits 1:0 will not be changed

38.16.8.7 TOM[i]_TGC[y]_FUPD_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8038_H + (800_H × i) + (200_H × y)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSTCN0_CH7		RSTCN0_CH6		RSTCN0_CH5		RSTCN0_CH4		RSTCN0_CH3		RSTCN0_CH2		RSTCN0_CH1		RSTCN0_CH0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FUPD_CTRL7		FUPD_CTRL6		FUPD_CTRL5		FUPD_CTRL4		FUPD_CTRL3		FUPD_CTRL2		FUPD_CTRL1		FUPD_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.184 TOM[i]_TGC[y]_FUPD_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	RSTCN0_CH [7:0]	Reset CN0 of channel [7:0] on force update event READ access: 00_B: CN0 is not reset on forced update 01_B: — 10_B: — 11_B: CN0 is reset on forced update WRITE access: 00_B: Don't care, bits 1:0 will not be changed 01_B: Do not reset CN0 on forced update 10_B: Reset CN0 on forced update 01_B: Don't care, bits 1:0 will not be changed If enabled, reset CN0 triggered by HOST_TRIG, ACT_TB compare match or internal trigger.
15 to 0	FUPD_CTRL [7:0]	Force update of TOM channel k operation register READ access: 00_B: Force update disabled 01_B: — 10_B: — 11_B: Force update enabled WRITE access: 00_B: Don't care, bits 1:0 will not be changed 01_B: Disable force update 10_B: Enable force update 11_B: Don't care, bits 1:0 will not be changed If enabled, force update of register CM0, CM1 and CLK_SRC triggered by HOST_TRIG, ACT_TB compare match or internal trigger. **NOTE** The force update request is stored and executed synchronized to the selected FXCLK.

38.16.8.8 TOM[i]_TGC[y]_INT_TRIG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $803C_H$ + (800_H × i) + (200_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT_TRIG7		INT_TRIG6		INT_TRIG5		INT_TRIG4		INT_TRIG3		INT_TRIG2		INT_TRIG1		INT_TRIG0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.185 TOM[i]_TGC[y]_INT_TRIG Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	INT_TRIG[7:0]	Select input signal TRIG_[7:0] as a trigger source READ access: 00_B: internal trigger from channel 0 (TRIG_0) not used 01_B: — 10_B: — 11_B: internal trigger from channel 0 (TRIG_0) used WRITE access: 00_B: don't care, bits 1:0 will not be changed 01_B: do not use internal trigger from channel 0 (TRIG_0) 10_B: use internal trigger from channel 0 (TRIG_0 11_B: don't care, bits 1:0 will not be changed

38.16.8.9 TOM[i]_CH[x]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8000_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FREEZE	—	GCM	SPEM	BITREV	OSM	SPE_TRIG	TRIGOUT	EXTTRIGOUT	EXT_TRIG	OSM_TRIG	RST_CCU0	UDMODE		TRIG_PULSE	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECLK_SRC	CLK_SRC_SR			SL	—	—	—	SR0_TRIG	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R	R	R	R

Table 38.186 TOM[i]_CH[x]_CTRL Register Contents (1/4)

Bit Position	Bit Name	Function
31	FREEZE	0: A channel disable/enable may change internal register and output register 1: A channel enable/disable does not change an internal or output register but stops counter CN0
30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	GCM	Gated Counter Mode enable 0: Gated Counter mode disabled 1: Gated Counter mode enabled **NOTE** The Gated Counter mode is only available for TOM instances connected to a SPE module and only for channels 0 to 7.
28	SPEM	SPE output mode enable for channel. 0: SPE output mode disabled: TOM[i]_CH[x]_OUT defined by TOM[i] channel x SOUR register 1: SPE output mode enabled: TOM[i]_CH[x]_OUT is defined by SPE[i]_OUT[x] **NOTES** 1. The SPE output mode is only implemented for TOM instances connected to a SPE module and only for TOM channels 0 to 7. 2. On TOM channel 2, 6 and 7 SPEM = 1 has special meaning. If SPEM = 1, the signal SPE_NIPD triggers the reset of CN0. If SPEM = 1 and OSM = 1, the signal SPE_NIPD triggers the start of single pulse generation. 3. For TOM channel 2, 6 and 7 this bit defines in combination with bit SPE_TRIG the source of output pin TOM[i]_CH[x]_OUT and if CN0 can be reset by TOM input signal SPE[i]_NIPD.
27	BITREV	Bit-reversing of output of counter register CN0. **NOTE** This bit enables the PCM mode of channel 15.

Table 38.186 TOM[i]_CH[x]_CTRL Register Contents (2/4)

Bit Position	Bit Name	Function
26	OSM	One-shot mode. In this mode the counter CN0 counts for only one period. The length of period is defined by CM0. A write access to the register CN0 triggers the start of counting. 0: One-shot mode disabled 1: One-shot mode enabled
25	SPE_TRIG	SPE trigger to reset CN0 If SPEM = 0: 0: TOM[i]_CH[x]_OUT defined by TOM[i] channel x SOUR register, CN0 reset is defined by configuration of bit RST_CCU0 1: TOM[i]_CH[x]_OUT defined by TOM[i] channel x SOUR register, CN0 is reset by signal SPE[i]_NIPD If SPEM = 1: 0: TOM[i]_CH[x]_OUT is defined by SPE[i]_OUT[x], CN0 is reset by signal SPE[i]_NIPD 1: TOM[i]_CH[x]_OUT is defined by SPE[i]_OUT[x], CN0 reset is defined by configuration of bit RST_CCU0 For TOM channel 2, 6 and 7 this bit defines in combination with bit SPEM the source of output pin TOM[i]_CH[x]_OUT and if CN0 can be reset by TOM input signal SPE[i]_NIPD. **NOTES** 1. For TOM channel 8 and 9 this bit defines only if CN0 reset is defined by input signal SPE[i]_NIPD or by configuration of RST_CCU0. The output TOM[i]_CH[x]_OUT is not affected. The configuration bit SPEM is not available for these channels and thus assumed to be 0. 2. If a configuration of SPEM | SPE_TRIG = 0 | 1 or 1 | 0 is chosen (i.e. CN0 is reset by signal SPE[i]_NIPD), the one-shot mode in corresponding TOM channel should also be enabled by setting bit OSM = 1 to generate one PWM pulse in case of trigger SPE[i]_NIPD. 3. In SPE module one of the trigger signals TOM[i]_CH2_TRIG_CCU1, TOM[i]_CH6_TRIG_CCU1, TOM[i]_CH7_TRIG_CCU1, TOM[i]_CH8_TRIG_CCU1, or TOM[i]_CH9_TRIG_CCU1 can be used to trigger the update of register SPE[i]_OUT_CTRL.
24	TRIGOUT	Trigger output selection (output signal TRIGOUT[x]) of module TOM_CH[x] 0: TRIGOUT[x] is TRIGIN[x] or TIM_EXT_CAPTURE(x). 1: TRIGOUT[x] is TRIG_CCU0
23	EXTTRIGOUT	Select TIM_EXT_CAPTURE(x) as potential output signal TRIGOUT[x] 0: Signal TRIGIN[x] is selected as output on TRIGOUT[x] (if TRIGOUT= 0) 1: Signal TIM_EXT_CAPTURE(x) is selected as output on TRIGOUT[x] (if TRIGOUT=0)
22	EXT_TRIG	Select TIM_EXT_CAPTURE(x) as trigger signal 0: Signal TIM_[x-1] is selected as trigger to reset CN0 or to start single pulse generation. 1: Signal TIM_EXT_CAPTURE(x) is selected
21	OSM_TRIG	Enable trigger of one-shot pulse by trigger signal OSM_TRIG 0: Signal OSM_TRIG cannot trigger start of single pulse generation 1: Signal OSM_TRIG can trigger start of single pulse generation (only if bit OSM = 1) **NOTE** This bit should only be set if bit OSM = 1 and bit RST_CCU0 = 0.

Table 38.186 TOM[i]_CH[x]_CTRL Register Contents (3/4)

Bit Position	Bit Name	Function
20	RST_CCU0	Reset source of CCU0 0: Reset counter register CN0 to 0 on matching comparison CM0 1: Reset counter register CN0 to 0 on trigger TRIGIN[x] or TIM_EXT_CAPTURE(x). **NOTE** This bit should only be set if bit OSM = 0 (i.e. in continuous mode)
19, 18	UDMODE	Up-down counter mode 00_B: Up-down counter mode disabled: CN0 counts always up 01_B: Up-down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches 0 (i.e. changes from down to up) 10_B: Up-down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches CM0 (i.e. changes from up to down) 11_B: Up-down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches 0 or CM0 (i.e. changes direction)
17	TRIG_PULSE	Trigger output pulse length of one SYS_CLK period 0: Output on TOM[i]_OUT[x]_T is 1 as long as CN0=SR0 (if SR0_TRIG = 1) 1: Output on TOM[i]_OUT[x]_T is 1 for only one SYS_CLK period if CN0 = SR0 (if SR0_TRIG = 1)
16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	ECLK_SRC	Extend CLK_SRC 0: CLK_SRC_SR set 1 selected (see bit CLK_SRC_SR) 1: CLK_SRC_SR set 2 selected (see bit CLK_SRC_SR)
14 to 12	CLK_SRC_SR	Clock source select for channel If ECLK_SRC = 0: 000_B: CMU_FXCLK (0) selected 001_B: CMU_FXCLK (1) selected 010_B: CMU_FXCLK (2) selected 011_B: CMU_FXCLK (3) selected 100_B: CMU_FXCLK (4) selected 101_B: Clock of channel stopped 110_B: Clock of channel stopped 111_B: Clock of channel stopped If ECLK_SRC = 1: 000_B: CMU_FXCLK (0) selected 001_B: CMU_FXCLK (1) selected 010_B: CMU_FXCLK (2) selected 011_B: CMU_FXCLK (3) selected 100_B: CMU_FXCLK (4) selected 101_B: TRIG[x–1] selected 110_B: TIM_EXT_CAPTURE[x] selected 111_B: Reserved The register CLK_SRC is updated with the value of CLK_SRC_SR together with the update of register CM0 and CM1. The input of the FX clock divider depends on the value of FXCLK_SEL (see CMU). **NOTES** 1. This register is a shadow register for the register CLK_SRC. Thus, if the CMU_CLK source for PWM generation should be changed during operation, the old CMU_CLK has to operate until the update of the ATOM channels internal CLK_SRC register by the CLK_SRC_SR content is done either by an end of a period or a forced update. 2. If clock of channel is stopped (i.e. ECLK_SRC = 0 and CLK_SRC = 101/110/111), the channel can only be restarted by resetting CLK_SRC_SR to a value of 000 to 100 and forcing an update via the force update mechanism.

Table 38.186 TOM[i]_CH[x]_CTRL Register Contents (4/4)

Bit Position	Bit Name	Function
11	SL	Signal level for duty cycle 0: Low signal level 1: High signal level If the output is disabled, the output TOM_OUT[x] is set to inverse value of SL. **NOTE** Reset value depends on the hardware configuration chosen by silicon vendor.
10 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SR0_TRIG	SR0 is used to generate a trigger on output TOM[i]_CH[x]_OUT_T if equal to CN0. 0: SR0 is used as a shadow register for register CM0. 1: SR0 is not used as a shadow register for register CM0. SR0 is compared with CN0 and if both are equal, a trigger pulse is generated at output TOM[i]_CH[x]_OUT_T. **NOTE** This bit should only be set if RST_CCU0 of this channel is 0.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

38.16.8.10 TOM[i]_CH[x]_CN0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8014_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CN0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.187 TOM[i]_CH[x]_CN0 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CN0	TOM CCU0 counter register This counter is stopped if the TOM channel is disabled and not reset on an enable event of TOM channel.

38.16.8.11 TOM[i]_CH[x]_CM0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8014_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CN0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.188 TOM[i]_CH[x]_CM0 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CN0	TOM CCU0 compare register Setting CM0 < CM1 configures a duty cycle of 100%

38.16.8.12 TOM[i]_CH[x]_SR0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8004_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SR0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.189 TOM[i]_CH[x]_SR0 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	SR0	TOM channel x shadow register SR0 for update of compare register CM0

38.16.8.13 TOM[i]_CH[x]_CM1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8010_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CM1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.190 TOM[i]_CH[x]_SR0 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CM1	TOM CCU1 compare register Setting CM1 = 0 configures a duty cycle of 0% independent of the configured value of CM0.

38.16.8.14 TOM[i]_CH[x]_SR1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8008_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SR1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.191 TOM[i]_CH[x]_SR1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	SR1	TOM channel x shadow register SR1 for update of compare register CM1

38.16.8.15 TOM[i]_CH[x]_STAT

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 8018_H + (800_H × i) + (40_H × x)

Value after reset: 0000 $000x_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.192 TOM[i]_CH[x]_STAT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	OL	Output level of output TOM_OUT(x) **NOTE** Reset value is the inverted value of SL bit which depends on the hardware configuration chosen by silicon vendor.

38.16.8.16 TOM[i]_CH[x]_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $801C_H$ + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCU1TC	CCU0TC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.193 TOM[i]_CH[x]_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CCU1TC	CCU1 Trigger condition interrupt for channel x If SR0_TRIG = 0: 0: No interrupt occurred 1: The condition CN0 ≥ CM1 was detected If SR0_TRIG = 1: 0: No interrupt occurred 1: The condition SR0 = CN0 was detected **NOTE** The notification of the interrupt is only triggered one time after reaching the condition CN0≥ CM1. To enable re-trigger of the notification first the condition CN0 < CM1 has to be reached.
0	CCU0TC	CCU0 Trigger condition interrupt for channel x 0: No interrupt occurred 1: The condition CN0 ≥ CM0 was detected. The notification of the interrupt is only triggered one time after reaching the condition CN0 ≥ CM0. To enable re-trigger of the notification first the condition CN0 < CM1 has to be reached.

38.16.8.17 TOM[i]_CH[x]_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8020_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCU1TC_IRQ_EN	CCU0TC_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.194 TOM[i]_CH[x]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CCU1TC_IRQ_EN	TOM_CCU1TC_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP
0	CCU0TC_IRQ_EN	TOM_CCU0TC_IRQ interrupt enable 0: Disable interrupt, interrupt is not visible outside GTM-IP 1: Enable interrupt, interrupt is visible outside GTM-IP

38.16.8.18 TOM[i]_CH[x]_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8020_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG_CCU1TC0	TRG_CCU0TC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.195 TOM[i]_CH[x]_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TRG_CCU1TC0	Trigger TOM_CCU1TC0_IRQ interrupt by software 0: No interrupt triggering 1: Assert CCU1TC0_IRQ interrupt for one clock cycle **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
0	TRG_CCU0TC0	Trigger TOM_CCU0TC0_IRQ interrupt by software 0: No interrupt triggering 1: Assert corresponding field in TOM[i]_CH[x]_IRQ_NOTIFY register **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

38.16.8.19 TOM[i]_CH[x]_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 8028_H + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.196 TOM[i]_CH[x]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5**.

38.17 ARU-connected Timer Output Module (ATOM)

38.17.1 Overview

The ARU-connected Timer Output Module (ATOM) is able to generate complex output signals without CPU interaction due to its connectivity to the ARU. Typically, output signal characteristics are provided over the ARU connection through sub-modules connected to ARU like e.g. the MCS, DPLL or PSM. Each ATOM sub-module contains eight output channels which can operate independently from each other in several configurable operation modes. A block diagram of the ATOM sub-module is depicted in **Figure 38.62**.

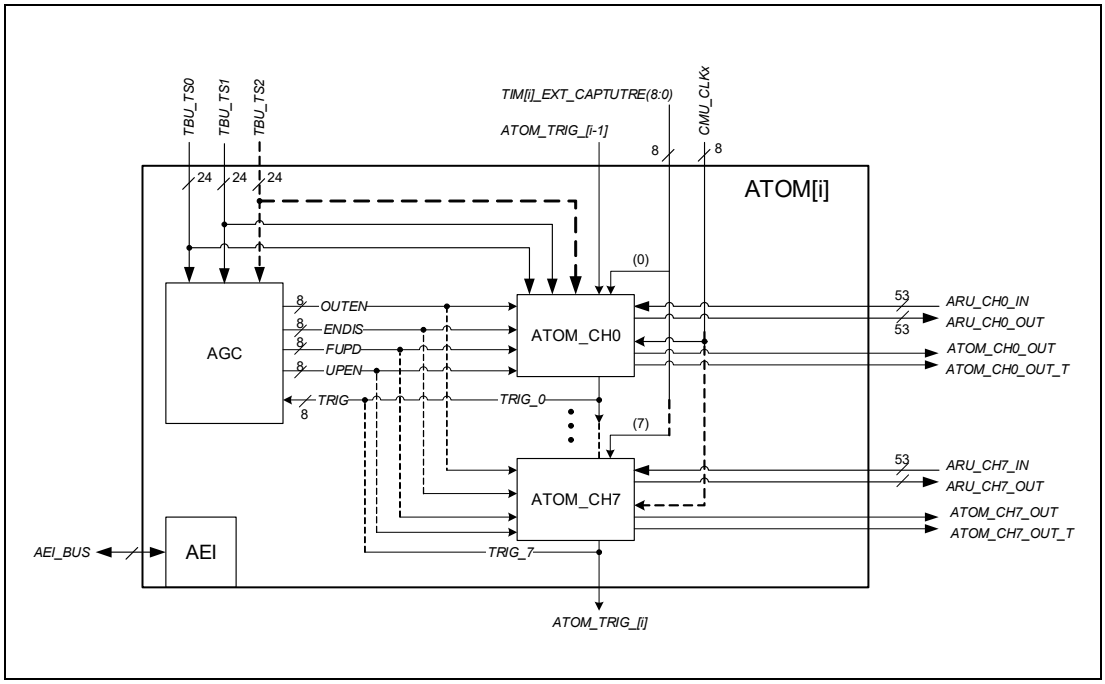


Figure 38.62 ATOM block diagram

The architecture of the ATOM sub-module is similar to the TOM sub-module, but there are some differences. First, the ATOM integrates only eight output channels. Hence, there exists one ATOM Global Control sub-unit (AGC) for the ATOM channels. The ATOM is connected to the ARU and can set up individual read requests from the ARU and write requests to the ARU. Furthermore, the ATOM channels are able to generate signals on behalf of time stamps and the ATOM channels are able to generate a serial output signal on behalf of an internal shift register.

Each ATOM channel provides five modes of operation:

- ATOM Signal Output Mode Immediate (SOMI)
- ATOM Signal Output Mode Compare (SOMC)
- ATOM Signal Output Mode PWM (SOMP)
- ATOM Signal Output Mode Serial (SOMS)
- ATOM Signal Output Mode Buffered Compare (SOMB)

These modes are described in more detail in **Section 38.17.3, ATOM Channel Modes**.

The ATOM channels operation registers (e.g. counter, compare registers) are 24 bit wide. Moreover, the input clocks for the ATOM channels come from the configurable CMU_CLKx signals of the CMU

sub-module. This gives the freedom to select a programmable input clock for the ATOM channel counters. The ATOM channel is able to generate a serial bit stream, which is shifted out at the ATOM[i]_CH[x]_OUT output.

When configured in this serial shift mode (SOMS) the selected CMU clock defines the shift frequency.

Each ATOM channel provides a so called operation and shadow register set. With this architecture it is possible to work with the operation register set, while the shadow register set can be reloaded with new parameters over CPU and/or ARU.

When update via ARU is selected, it is possible to configure for ATOM SOMP mode if both shadow registers are updated via ARU or only one of the shadow registers is updated.

On the other hand, the shadow registers can be used to provide data to the ARU when one or both of the compare units inside an ATOM channel match. This feature is only applicable in SOMC mode.

In TOM channels it is possible to reload the content of the operation registers with the content of the corresponding shadow registers and change the clock input signal for the counter register simultaneously. This simultaneous change of the input clock frequency together with reloading the operation registers is also implemented in the ATOM channels.

In addition to the feature that the CPU can select another CMU_CLKx during operation (i.e. updating the shadow register bit field CLK_SRC_SR of the ATOM[i]_CH[x]_CTRL register), the selection can also be changed via the ARU. Then, for the clock source update, the ACBI register bits of the ATOM[i]_CH[x]_STAT register are used as a shadow register for the new clock source.

In general, the behavior of the compare units CCU0 and CCU1 and the output signal behavior is controlled with the ACB bit field inside the ATOM[i]_CH[x]_CTRL register when the ARU connection is disabled and the behavior is controlled via ARU through the ACBI bit field of the ATOM[i]_CH[x]_STAT register, when the ARU is enabled.

Since the ATOM is connected to the ARU, the shadow registers of an ATOM channel can be reloaded via the ARU connection or via CPU over its AEI interface. When loaded via the ARU interface, the shadow registers act as a buffer between the ARU and the channel operation registers. Thus, a new parameter set for a PWM can be reloaded via ARU into the shadow registers, while the operation registers work on the actual parameter set.

The trigger signal ATOM_TRIG_[i-1] of ATOM instance i comes from the preceding instance i-1, the trigger ATOM_TRIG_[i] is routed to succeeding instance i+1.

NOTE

ATOM0 is connected to its own output ATOM_TRIG_0, i.e. the last channel of ATOM instance 0 can trigger the first channel of ATOM instance 0 (this path is registered, which means delayed by one SYS_CLK period).

38.17.1.1 ATOM Global Control (AGC)

Synchronous start, stop and update of work register of up to 8 channels is possible with the AGC sub-unit. This sub-unit has the same functionality as the TGC sub-unit of the TOM sub-module.

(1) Overview

There exists one global channel control unit (AGC) to drive a number of individual ATOM channels synchronously by external or internal events.

An AGC can drive up to eight ATOM channels.

The ATOM sub-module supports four different kinds of signaling mechanisms:

- Global enable/disable mechanism for each ATOM channel with control register ATOM[i]_AGC_ENDIS_CTRL and status register ATOM[i]_AGC_ENDIS_STAT
- Global output enable mechanism for each ATOM channel with control register ATOM[i]_AGC_OUTEN_CTRL and status register ATOM[i]_AGC_OUTEN_STAT
- Global force update mechanism for each ATOM channel with control register ATOM[i]_AGC_FUPD_CTRL
- Update enable of the register CM0, CM1 and CLK_SRC for each ATOM channel with the control bit field UPEN_CTRL[z] of ATOM[i]_AGC_GLB_CTRL

(2) AGC Sub-unit

Each of the first three individual mechanisms (enable/disable of the channel, output enable and force update) can be driven by three different trigger sources. The three trigger sources are:

- The host CPU (bit HOST_TRIG of register ATOM[i]_AGC_GLB_CTRL)
- The TBU time stamp (signal TBU_TS0 to 2 if available)
- The internal trigger signal TRIG (bunch of trigger signals TRIG_[x]) which can be either the trigger TRIG_CCU0 of channel x, the trigger of preceding channel x–1 (i.e. signal TRIG_[x–1]) or the external trigger TIM_EXT_CAPTURE(x) of assigned TIM channel x.

The first way is to trigger the control mechanism by a direct register write access via host CPU (bit HOST_TRIG of register ATOM[i]_AGC_GLB_CTRL).

The second way is provided by a compare match trigger on behalf of a specified time base coming from the module TBU (selected by bits TBU_SEL) and the time stamp compare value defined in the bit field ACT_TB of register ATOM[i]_AGC_ACT_TB.

NOTE

A cyclic event compare of ACT_TB and selected TBU_TS [x] is performed.

The third possibility is the input TRIG (bunch of trigger signals TRIG_[x]) coming from the ATOM channels 0 to 7.

The corresponding trigger signal TRIG_[x] coming from channel [x] can be masked by the register ATOM[i]_AGC_INT_TRIG.

To enable or disable each individual ATOM channel, the registers ATOM[i]_AGC_ENDIS_CTRL and/or ATOM[i]_AGC_ENDIS_STAT have to be used.

The register ATOM[i]_AGC_ENDIS_STAT controls directly the signal ENDIS. A write access to this register is possible.

The register ATOM[i]_AGC_ENDIS_CTRL is a shadow register that overwrites the value of register ATOM[i]_AGC_ENDIS_STAT if one of the three trigger conditions matches.

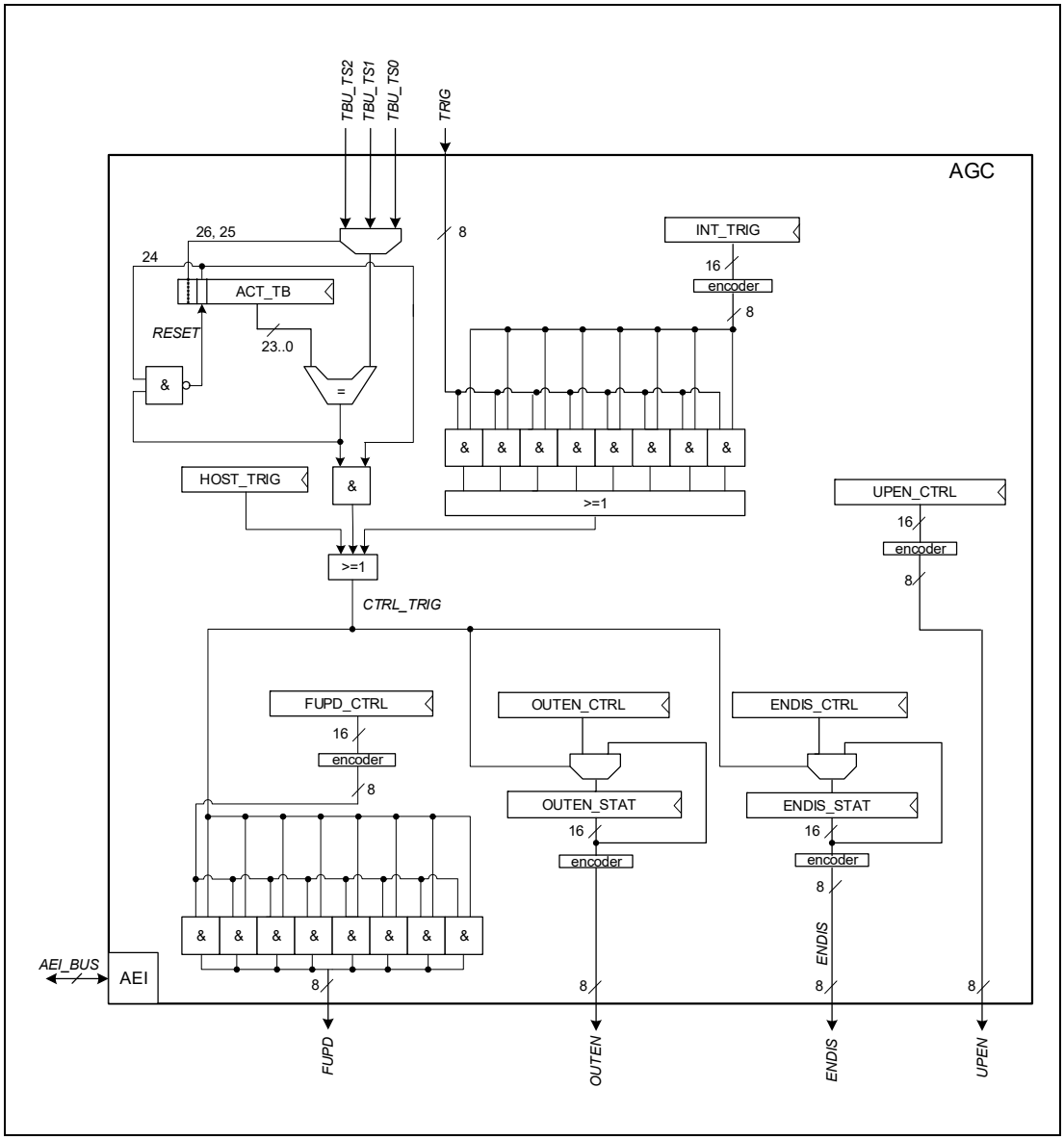


Figure 38.63 ATOM Global channel control mechanism

The output of the individual ATOM channels can be controlled using the register ATOM[i]_AGC_OUTEN_CTRL and ATOM[i]_AGC_OUTEN_STAT.

The register ATOM[i]_AGC_OUTEN_STAT controls directly the signal OUTEN. A write access to this register is possible.

The register ATOM[i]_AGC_OUTEN_CTRL is a shadow register that overwrites the value of register ATOM[i]_AGC_OUTEN_STAT if one of the three trigger conditions matches.

If an ATOM channel is disabled by the register ATOM[i]_AGC_OUTEN_STAT, the actual value of the channel output at ATOM_CH[x]_OUT is defined by the signal level bit (SL) defined in the channel control register ATOM[i]_CH[x]_CTRL. If the output is enabled, the output at ATOM_CH[x]_OUT depends on value of Flip-flop SOUR.

The register ATOM[i]_AGC_FUPD_CTRL defines which of the ATOM channels receive a FORCE_UPDATE event if the trigger signal CTRL_TRIG is raised.

NOTE

In SOMP mode the force update request is stored and executed synchronized to the selected CMU_CLK. In all other modes the force update request is executed immediately.

The register bits UPEN_CTRL[x] defines for which ATOM channel the update of the working register CM0, CM1 and CLK_SRC by the corresponding shadow register SR0, SR1 and CLK_SRC_SR is enabled. If update is enabled, the register CM0, CM1 and CLK_SRC will be updated on reset of counter register CN0 (see **Figure 38.64**).

38.17.1.2 ATOM Channel Mode Overview

Each ATOM channel offers the following different operation modes:

In ATOM Signal Output Mode Immediate (SOMI), the ATOM channels generate an output signal immediately after receiving an ARU word according to the two signal level output bits of the ARU word received through the ACBI bit field. Due to the fact, that the ARU destination channels are served in a round robin order, the output signal can jitter in this mode with a jitter of the ARU round trip time.

In ATOM Signal Output Mode Compare (SOMC), the ATOM channel generates an output signal on behalf of time stamps that are located in the ATOM operation registers. These time stamps are compared with the time stamps, the TBU generates. The ATOM is able to receive new time stamps either by CPU or via the ARU. The new time stamps are directly loaded into the channels operation register. The shadow registers are used as capture registers for two time base values, when a compare match of the channels operation registers occurs.

In ATOM Signal Output Mode PWM (SOMP), the ATOM channel is able to generate simple and complex PWM output signals like the TOM sub-module by comparing its operation registers with a sub-module internal counter. In difference to the TOM, the ATOM shadow registers can be reloaded by the CPU and by the ARU in the background, while the channel operates on the operation registers.

In ATOM Signal Output Mode Serial (SOMS), the ATOM channel generates a serial output bit stream on behalf of a shift register. The number of bits shifted and the shift direction is configurable. The shift frequency is determined by one of the CMU_CLKx clock signals. Refer to **Section 38.17.3.4, ATOM Signal Output Mode Serial (SOMS)** for further details.

In ATOM Signal Output Buffered Compare (SOMB), the ATOM channel generates an output signal on behalf of time stamps that located in the ATOM operation registers. These time stamps are compared with the time stamps, the TBU generates. The ATOM is able to receive new compare values either by CPU or via the ARU. The new compare values received via ARU are stored first in the shadow register and only if previous compare match is occurred, the operation register are updated with the content of the shadow register.

38.17.2 ATOM Channel Architecture

Each ATOM channel is able to generate output signals according to five operation modes. The architecture of the ATOM channels is similar to the architecture of the TOM channels. The general architecture of an ATOM channel is depicted in **Figure 38.64**.

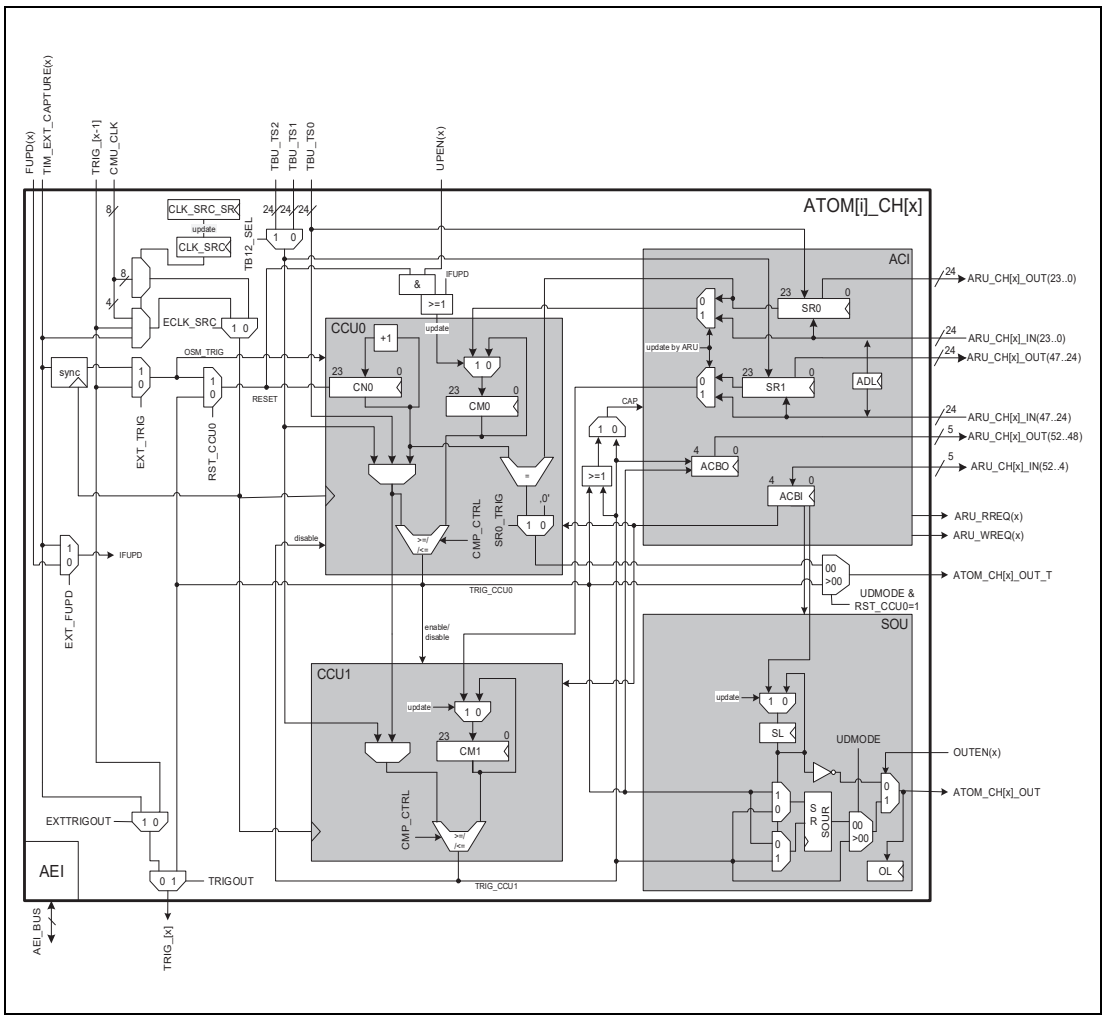


Figure 38.64 ATOM channel architecture

In all ATOM channels the operation registers CN0, CM0 and CM1 and the shadow registers SR0 and SR1 are the 24 bit width. The comparators inside CCU0 and CCU1 provide a selectable signed greater-equal or less-equal comparison to compare against the GTM time bases TBU_TS0, TBU_TS1 and, if available, TBU_TS2. Refer to TBU **Section 38.14, Time Base Unit (TBU)** for further details. The CCU0 and CCU1 units have different tasks for the different ATOM channel modes.

The cyclic event compare is used to detect time base overflows and to guarantee, that a compare match event can be set up for the future even when the time base will first overflow and then reach the compare value.

NOTE

For a correct behavior of this cyclic event compare, the new compare value must not be specified larger/smaller than half of the range of the total time base value ($7FFFFF_H$).

In SOMC/SOMB mode, the two compare units CCUx can be used in combination to each other. When used in combination, the trigger lines TRIG_CCU0 and TRIG_CCU1 can be used to enable/disable the other compare unit on a match event. Refer to **Section 38.17.3.2, ATOM Signal Output Mode Compare (SOMC)** and **Section 38.17.3.5, ATOM Signal Output Mode Buffered Compare (SOMB)** for further details.

The Signal Output Unit (SOU) generates the output signal for each ATOM channel. This output signal level depends on the ATOM channel mode and on the SL bit of the ATOM[i]_CH[x]_CTRL register in combination with the two control bits. These two control bits ACB(1) and ACB(0) can either be received via CPU in the ACB register field of the ATOM[i]_CH[x]_CTRL register or via ARU in the ACBI bit field of the ATOM[i]_CH[x]_STAT register.

The SL bit in the ATOM[i]_CH[x]_CTRL register defines in all modes the operational behavior of the ATOM channel.

When the channel and its output are disabled, the output signal level of the channel is the inverse of the SL bit.

In SOMI, SOMC and SOMB mode the output signal level depends on the SL, ACB0 and ACB1 bits. In SOMP mode the output signal level depends on the two trigger signals TRIG_CCU0 and TRIG_CCU1 since theses two triggers define the PWM timing characteristics and the SL bit defines the level of the duty cycle. In SOMS mode the output signal level is defined by the bit pattern that has to be shifted out by the ATOM channel. The bit pattern is located inside the CM1 register.

The ARU Communication Interface (ACI) sub-unit is responsible for requesting data routed through ARU to the ATOM channel in SOMI, SOMP, SOMB and SOMS modes, and additionally for providing data to the ARU in SOMC mode.

In SOMC mode the ACI shadow registers have a different behavior and are used as output buffer registers for data send to ARU.

38.17.2.1 ARU Communication Interface

The ATOM channels have an ARU Communication Interface (ACI) sub-unit. This sub-unit is responsible for data exchange from and to the ARU. This is done with the two implemented registers SR0, SR1, and the ACBI and ACBO bit fields that are part of the ATOM[i]_CH[x]_STAT register. The ACI architecture is shown in **Figure 38.65**.

If the ARU_EN bit is set inside the ATOM[i]_CH[x]_CTRL register, the ATOM channel is enabled by setting the enable bits inside the ATOM[i]_AGC_ENDIS_STAT register and the CPU hasn't written data not equal to zero into the CM0, CM1, SR0, SR1 register, the ATOM channel will first request data from the ARU before the signal generation starts in SOMP, SOMS, SOMC and SOMB mode.

NOTES

1. If in SOMP mode there is data inside the CM0 or SR0 register not equal to 0 the channel counter CN0 will start counting immediately, regardless whether the channel has received ARU data yet.
2. If in SOMS mode there is data inside the CM0 or SR0 register not equal to 0 the channel will start shifting immediately, regardless whether the channel has received ARU data yet.

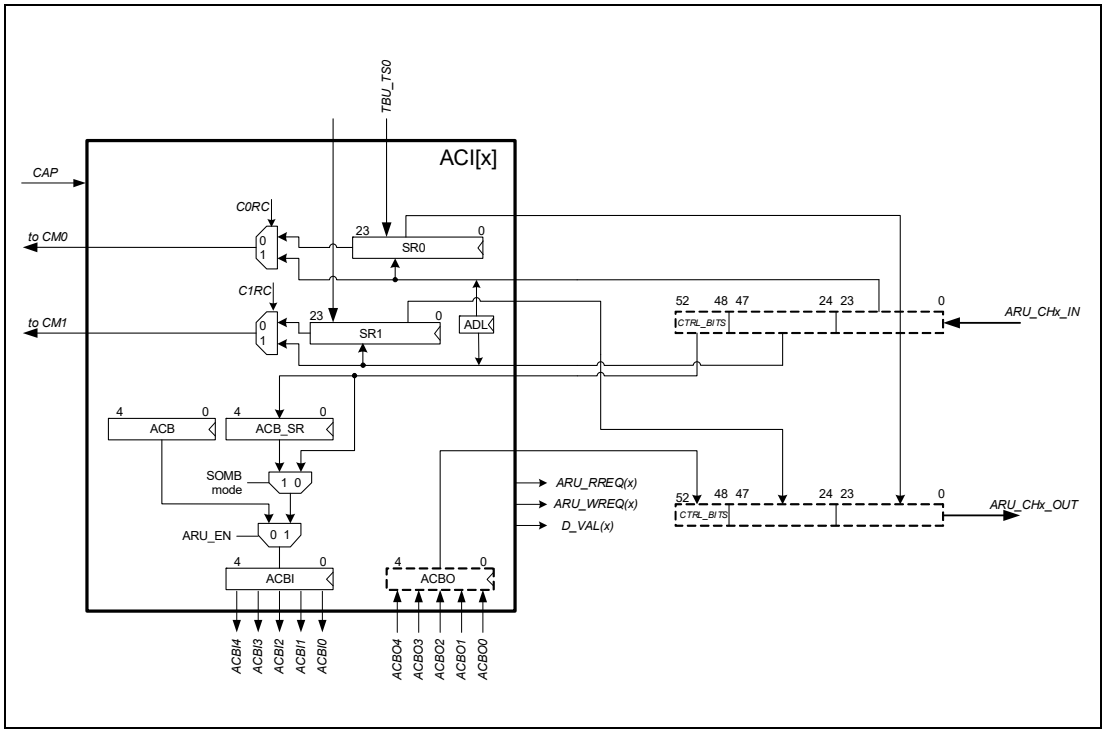


Figure 38.65 ACI architecture overview

Incoming ARU data (53 bit width signal ARU_CHx_IN) is split into three parts by the ACI and communicated to the ATOM channel registers. In SOMI, SOMP, SOMS and SOMB modes incoming ARU data ARU_CHx_IN is split in a way that the lower 24 bits of the ARU data (23 down to 0) are stored in the SR0 register, the upper bits (47 down to 24) are stored in the SR1 register. The bits 52 down to 48 (CTRL_BITS) are stored in SOMI, SOMP and SOMS mode in the ACBI bit field of the register ATOM[i]_CH[x]_STAT, in SOMB mode in the internal ACB_SR register.

The ATOM channel has to ensure, that in a case when the channel operation registers CM0 and CM1 are updated with the SR0 and SR1 register content and an ARU transfer to these shadow registers

happens in parallel that either the old data in both shadow registers is transferred into the operation registers or both new values from the ARU are transferred.

In SOMC mode incoming ARU data ARU_CHx_IN is written directly to the ATOM channel operation register in the way that the lower 24 bits (23 down to 0) are written to CM0, and the bits 47 down to 24 are written to register CM1. The bits 52 down to 48 are stored in the ACBI bit field of the ATOM[i]_CH[x]_STAT register and control the behavior of the compare units and the output signal of the ATOM channel.

In SOMC mode the SR0 and SR1 registers serve as capture registers for the time stamps coming from TBU whenever a compare match event is signaled by the CCU0 and/or CCU1 sub-units via the CAP signal line. These two time stamps are then provided together with actual ATOM channel status information located in the ACBO bit field to the ARU at the dedicated ARU write address of the ATOM channel when the ARU is enabled.

The encoding of the ARU control bits in the different ATOM operation modes is described in more detail in the following chapters.

38.17.3 ATOM Channel Modes

As described above, each ATOM channel can operate independently from each other in one of five dedicated output modes:

- ATOM Signal Output Mode Immediate (SOMI)
- ATOM Signal Output Mode Compare (SOMC)
- ATOM Signal Output Mode PWM (SOMP)
- ATOM Signal Output Mode Serial (SOMS)
- ATOM Signal Output Mode Buffered Compare (SOMB)

The Signal Output Mode PWM (SOMP) is principally the same like the output mode for the TOM sub-module. In addition, it is possible to reload the shadow registers via the ARU without the need of a CPU interaction. The other modes provide additional functionality for signal output control. All operation modes are described in more detail in the following sections.

NOTE

In any output mode, if a channel is enabled, one-shot mode is disabled (OSM = 0; only used in modes SOMP and SOMS) and CM0 ≥ CN0, the counter CN0 is incrementing until it reaches CM0. To avoid unintended counting of CN0 after enabling a channel, it is recommended to reset a channel (or at least CN0 and CM0) before any change on the mode bits MODE, ARU_EN and OSM.

The ACB bitfield of ATOM[i]_CH[x]_CTRL register has a different mapping regarding the ATOM channel mode. Refer to table as below

Table 38.197 Coding of ACB bitfield of ATOM[i]_CH[x]_CTRL register (1/2)

Mode	Name/(Bit)	Description	Coding	Notes
SOMI	ACB0/(4)	ACB0 bit 0	0_B: Set output to inverse bit 1_B: Set output to SL bit	
	--/(8:5)	Not used		
SOMC	ACB10/(5:4)	Signal level Control	00_B: No signal level change at output (exception in tables **Table 38.201, ATOM CCUx Serve first definition ACB42 = 000** and **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001** mode ACB42=001). 01_B: Set output signal level to 1 when SL bit = 0 else output signal level to 0. 10_B: Set output signal level to 0 when SL bit = 0 else output signal level to 1. 11_B: Toggle output signal level (exception in tables **Table 38.201, ATOM CCUx Serve first definition ACB42 = 000** and **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001** mode ACB42=001).	
SOMC	ACB42/(8:6)	Compare strategy	000_B: Compare in CCU0 and CCU1 in parallel, disable the CCUx on a compare match on either of compare units. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 001_B: Compare in CCU0 and CCU1 in parallel, disable the CCUx on a compare match on either compare units. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 010_B: Compare in CCU0 only against TBU_TS0. 011_B: Compare in CCU1 only against TBU_TS1 or TBU_TS2. 100_B: Compare first in CCU0 and then in CCU1. Use TBU_TS0. 101_B: Compare first in CCU0 and then in CCU1. Use TBU_TS1 or TBU_TS2. 110_B: Compare first in CCU0 and then in CCU1. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 111_B: Cancel pending compare events.	**Note:** These bits are only applicable if ARU_EN = '0'. **Note:** Independent of ARU_EN, a writing of 111_B cancels any pending CCU0 or CCU1 compare.
SOMP	ADL/(5:4)	ARU data select	00_B: Load both ARU words into shadow registers. 01_B: Load ARU low word (Bits 23..0) into shadow register SR0. 10_B: Load ARU high word (Bits 47..24) into shadow register SR1. 11_B: Reserved.	
	BITREV/(6)	PCM mode enable	0_B/1_B: Bit-reversing of output of counter register CN0. This bit enables the PCM mode.	**Note:** It is device specific, in which channel the PCM mode is available. Refer to device specific **38.28, GTM Device 358** for this information.
	SR0_TRIG/(7)	SR0 used for ATOM[i]_CH[x]_OUT_T	0_B: SR0 is used as a shadow register for register CM0. 1_B: SR0 is not used as a shadow register for register CM0. SR0 is compared with CN0 and if both are equal, a trigger pulse is generated at output ATOM[i]_CH[x]_OUT_T.	**Note:** This bit may only be set if RST_CCU0 of this channel is 0.

Table 38.197 Coding of ACB bitfield of ATOM[i]_CH[x]_CTRL register (2/2)

Mode	Name/(Bit)	Description	Coding	Notes
SOMS	ACB0/(4)	Shift direction	0_B: Right shift of data is started from bit 0 of CM1. 1_B: Left shift of data is started from bit 23 of CM1.	**Note:** The data that has to be shifted out has to be aligned inside the CM1 register according to the defined shift direction. **Note:** This bit is only applicable if ARU_EN = 0. **Note:** If the direction (ACB0) is changed the output ATOM_OUT[x] switches immediately to the other 'first' bit of CM1 (bit 0 if ACB0 = 0, bit 23 if ACB0 = 1).
	--/(6:5)	Not used		
	DSO/(7)	Double shift output	0_B: CM1 is used as a 24 bit shift register. 1_B: CM1 is split into two 12 bit shift register.	**Note:** If DSO=1, only shift right is possible.
	--/(8)	Not used		
SOMB	ACB10/(5:4)	Signal level control	00_B: No signal level change at output. 01_B: Set output signal level to 1 when SL bit = 0 else output signal level to 0. 10_B: Set output signal level to 0 when SL bit = 0 else output signal level to 1. 11_B: Toggle output signal level. For details: **Table 38.208, ATOM SOMB output control by ACBI[1:0] and SL**	**Note:** These bits are only applicable if ARU_EN = 0.
	ACB42/(8:6)	Compare strategy	000_B: Reserved. Has no effect. 001_B: Reserved. Has no effect. 010_B: Compare in CCU0 only against TBU_TS0. 011_B: Compare in CCU1 only against TBU_TS1 or TBU_TS2. 100_B: Compare first in CCU0 and then in CCU1. Use TBU_TS0. 101_B: Compare first in CCU0 and then in CCU1. Use TBU_TS1 or TBU_TS2. 110_B: Compare first in CCU0 and then in CCU1. Use TBU_TS0 in CCU0 and TBU_TS1 or TBU_TS2 in CCU1. 111_B: Cancel pending comparisons independent on ARU_EN. For details: **Table 38.207, ATOM SOMB compare strategies**	**Note:** These bits are only applicable if ARU_EN = 0.

38.17.3.1 ATOM Signal Output Mode Immediate (SOMI)

In ATOM Signal Output Mode Immediate (SOMI), the ATOM channel generates output signals on the ATOM[i]_CH[x]_OUT output port immediate after update of the bit ACBI(0) of register ATOM[i]_CH[x]_STAT or ACB(0) bit of register ATOM[i]_CH[x]_CTRL.

If ARU access is enabled by setting bit ARU_EN in register ATOM[i]_CH[x]_CTRL, the update of the output ATOM[i]_CH[x]_OUT depends on the bit ACBI(0) of register ATOM[i]_CH[x]_STAT received at the ACI sub-unit and the bit SL bit of register ATOM[i]_CH[x]_CTRL. The remaining 48 ARU bits (47 down to 0) have no meaning in this mode.

If ARU access is disabled, the update of the output ATOM[i]_CH[x]_OUT depends on the bit ACB(0) and the bit SL of register ATOM[i]_CH[x]_CTRL.

The initial ATOM channel port pin ATOM[i]_CH[x]_OUT signal level has to be specified by the SL bit field of the ATOM[i]_CH[x]_CTRL register when OUTEN_CTRL register bit field OUTEN_CTRLx is disabled (see **Section 38.17.6.5, ATOM[i]_AGC_OUTEN_CTRL**) for details.

In SOMI mode the output behavior depends on the SL bit of register ATOM[i]_CH[x]_CTRL and the bit ACBI(0) of the ATOM[i]_CH[x]_STAT register or the bit ACB0 of register ATOM[i]_CH[x]_CTRL:

(1) Output behavior in SOMI mode

SL	ACBI(0)/ACB(0)	Output behavior
0	0	Set output to inverse of SL (1)
0	1	Set output to SL (0)
1	0	Set output to inverse of SL (0)
1	1	Set output to SL (1)

The signal level bit ACBI(0) is transferred to the SOU sub-unit of the ATOM and made visible at the output port according to the table above immediately after the data was received by the ACI. This can introduce a jitter on the output signal since the ARU channels are served in a time multiplexed fashion.

(2) ATOM[i]_CH[x]_CTRL in SOMI mode

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E8004_H$ + (800_H × i) + (40_H × x)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FREEZE	Not used	Not used	Not used	Not used	Not used	Not used	Not used	Not used			Not used	Not used		Not used	Not used
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Not used	Not used			SL	Not used	Not used	Not used				ACB(0)	ARU_EN	Not used	MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 38.198 ATOM[i]_CH[x]_CTRL Registers

Bit Position	Bit Name	Function
31	FREEZE	0: A channel disable/enable may change internal register and output register 1: A channel enable/disable does not change an internal or output register but stops counter CN0
30	Not used	Not used in this mode.
29	Not used	Not used in this mode.
28	Not used	Not used in this mode.
27	Not used	Not used in this mode.
26	Not used	Not used in this mode.
25	Not used	Not used in this mode.
24	Not used	Not used in this mode.
23 to 21	Not used	Not used in this mode.
20	Not used	Not used in this mode.
19, 18	Not used	Not used in this mode.
17	Not used	Not used in this mode.
16	Not used	Not used in this mode.
15	Not used	Not used in this mode.
14 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	SL	Initial signal level after channel is enabled 0: Low signal level 1: High signal level **NOTES** 1. Reset value depends on the hardware configuration chosen by silicon vendor. 2. If the output is disabled, the output ATOM_OUT[x] is set to inverse value of SL. 3. If FREEZE = 0 and the channel is disabled, the output register of SOU unit is set to inverse value of SL. 4. If FREEZE = 1 and the channel is disabled, the output register of SOU unit is not changed and output ATOM_OUT[x] is not changed.
10	Not used	Not used in this mode.
9	Not used	Not used in this mode.
8 to 5	Not used	Not used in this mode.
4	ACB(0)	ACB bit 0 0: Set output to inverse of SL bit 1: Set output to SL bit
3	ARU_EN	ARU Input stream enable 0: ARU Input stream disabled 1: ARU Input stream enabled
2	Not used	Not used in this mode.
1 to 0	MODE	ATOM channel mode select. 00_B: ATOM Signal Output Mode Immediate (SOMI)

38.17.3.2 ATOM Signal Output Mode Compare (SOMC)

(1) Overview

In ATOM Signal Output Mode Compare (SOMC) the output action is performed in dependence of the comparison between input values located in CM0 and/or CM1 registers and the two (three) time base values TBU_TS0 or TBU_TS1 (or TBU_TS2) provided by the TBU. For a description of the time base generation refer to the TBU specification in **Section 38.14**. It is configurable, which of the two (three) time

bases is to be compared with one or both values in CM0 and CM1.

The behavior of the two compare units CCU0 and CCU1 is controlled either with the bits 4 down to 2 of ACB bit field inside the ATOM[i]_CH[x]_CTRL register, when the ARU connection is disabled or with the ACBI bit field of the ATOM[i]_CH[x]_STAT register, when the ARU is enabled. In that case the ACB bit field is updated via the ARU control bits 52 down to 48.

The CCUx trigger signals TRIG_CCU0 and TRIG_CCU1 always create edges, dependent on the predefined signal level in SL bit in combination with two control bits that can be specified by either ARU or CPU within the aforementioned ATOM[i]_CH[x]_CTRL or ATOM[i]_CH[x]_STAT registers.

In SOMC mode the channel is always disabled after the specified compare match event occurred. The shadow registers are used to store two time stamp values at the match time. The channel compare can be re-enabled by first reading the shadow registers, either by CPU or ARU and by providing new data for CMx registers through CPU or ARU. For a detailed description refer to **Section 38.17.3.2, (2) SOMC Mode under CPU control** and **Section 38.17.3.2, (3) SOMC Mode under ARU control**.

If three time bases exist for the GTM-IP there must be a preselection between TBU_TS1 and TBU_TS2 for the ATOM channel. This can be done with TB12_SEL bit in the ATOM[i]_CH[x]_CTRL register.

The comparison in CCU0/1 with time base TBU_TS1 or TBU_TS2 can be done on a greater-equal or less-equal compare according to the CMP_CTRL bit. This control bit has no effect to a compare unit CCU0 or CCU1 that compares against TBU_TS0. In this case always a greater-equal compare is done. The bit CMP_CTRL is part of the ATOM[i]_CH[x]_CTRL register.

When configured in SOMC mode, the channel port pin has to be initialized to an initial signal level. This initial level after enabling the ATOM channel is determined by the SL bit in the ATOM[i]_CH[x]_CTRL register. If the output is disabled, the signal level is set to the inverse level of the SL bit.

If the channel is disabled, the register SOUR is set to the SL bit in the ATOM[i]_CH[x]_CTRL register.

On a compare match event the shadow register SR0 and SR1 are used to capture the TBU time stamp values. SR0 always holds TBU_TS0 and SR1 either holds TBU_TS1 or TBU_TS2 dependent on the TB12_SEL bit in the ATOM[i]_CH[x]_CTRL register.

NOTE

When the channel is disabled and the compare registers are written, the compare registers CMx are loaded with the written value and the channel starts with the comparison on behalf of this values, when the channel is enabled.

(2) SOMC Mode under CPU control

As already mentioned above the ATOM channel can be controlled either by CPU or by ARU. When the channel should be controlled by CPU, the ARU_EN bit inside the ATOM[i]_CH[x]_CTRL register has to be reset.

The output of the ATOM channel is set on a compare match event depending on the ACB10 bit field in combination with the SL bit both located in the ATOM[i]_CH[x]_CTRL register. The output behavior according to the ACB10 bit field in the control register is shown in the following table:

The capture/compare strategy of the two CCUx units can be controlled with the ACB42 bit field inside the ATOM[i]_CH[x]_CTRL register.

The meaning of these bits is shown in the following table:

Table 38.199 Output behavior according to the ACB10 bit field in the control register

SL	ACB10(5)	ACB10(4)	Output Behavior
0	0	0	No signal level change at output (exception in **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**)
0	0	1	Set output signal level to 1
0	1	0	Set output signal level to 0
0	1	1	Toggle output signal level (exception in **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**)
1	0	0	No signal level change at output (exception in **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**)
1	0	1	Set output signal level to 0
1	1	0	Set output signal level to 1
1	1	1	Toggle output signal level (exception in **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**)

The capture/compare strategy of the two CCUx units can be controlled with the ACB42 bit field inside the ATOM[i]_CH[x]_CTRL register. The meaning of these bits is shown in the following table.

Table 38.200 The capture/Compare strategy of the two CCUx (1/2)

ACB42(8)	ACB42(7)	ACB42(6)	CCUx control
0	0	0	Serve First: Compare in CCU0 using TBU_TS0 and in parallel in CCU1 using TBU_TS1 or TBU_TS2. Disable other CCUx on compare match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACB10(5) and ACB10(4). Details see **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**.
0	0	1	Serve First: Compare in CCU0 using TBU_TS0 and in parallel in CCU1 using TBU_TS1 or TBU_TS2. Disable other CCUx on compare match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACB10(5) and ACB10(4). Details see **Table 38.202, ATOM CCUx Serve first definition ACB42 = 001**.
0	1	0	Compare in CCU0 only, use time base TBU_TS0. Output signal level is defined by combination of SL, ACB10(5) and ACB10(4) bits.
0	1	1	Compare in CCU1 only, use time base TBU_TS1 or TBU_TS2. Output signal level is defined by combination of SL, ACB10(5) and ACB10(4) bits.
1	0	0	Serve Last: Compare in CCU0 and then in CCU1 using TBU_TS0. Output signal level when CCU0 matches is defined by combination of SL, ACB10(5) and ACB10(4). On the CCU1 match the output level is toggled.

Table 38.200 The capture/Compare strategy of the two CCUx (2/2)

ACB42(8)	ACB42(7)	ACB42(6)	CCUx control
1	0	1	Serve Last: Compare in CCU0 and then in CCU1 using TBU_TS1 or TBU_TS2. Output signal level when CCU0 matches is defined by combination of SL, ACB10(5) and ACB10(4). On the CCU1 match the output level is toggled.
1	1	0	Serve Last: Compare in CCU0 using TBU_TS0 and then in CCU1 using TBU_TS1 or TBU_TS2. Output signal level when CCU1 matches is defined by combination of SL, ACB10(5) and ACB10(4).
1	1	1	Cancels pending comparison independent on ARU_EN.

The behavior of the ACBI / ACB42 bit combinations 000 and 001 is described in more detail in **Table 38.201** and **Table 38.202**.

Table 38.201 ATOM CCUx Serve first definition ACB42 = 000

ACB4	ACB3	ACB2	ACB1	ACB0	SL	CCU0 match	CCU1 match	Pin level new
0	0	0	0	0	0	0	1	hold
						1	0	hold
						1	1	hold
0	0	0	0	1	0	0	1	1
						1	0	1
						1	1	1
0	0	0	1	0	0	0	1	0
						1	0	0
						1	1	0
0	0	0	1	1	0	0	1	toggle
						1	0	toggle
						1	1	toggle
0	0	0	0	0	1	0	1	hold
						1	0	hold
						1	1	hold
0	0	0	0	1	1	0	1	0
						1	0	0
						1	1	0
0	0	0	1	0	1	0	1	1
						1	0	1
						1	1	1
0	0	0	1	1	1	0	1	toggle
						1	0	toggle
						1	1	toggle

Table 38.202 ATOM CCUx Serve first definition ACB42 = 001 (1/2)

ACB4	ACB3	ACB2	ACB1	ACB0	SL	CCU0 match	CCU1 match	Pin level new
0	0	1	0	0	0	0	1	hold
						1	0	toggle
						1	1	hold
0	0	1	0	1	0	0	1	0
						1	0	1
						1	1	0
0	0	1	1	0	0	0	1	1
						1	0	0
						1	1	1

Table 38.202 ATOM CCUx Serve first definition ACB42 = 001 (2/2)

ACB4	ACB3	ACB2	ACB1	ACB0	SL	CCU0 match	CCU1 match	Pin level new
0	0	1	1	1	0	0	1	toggle
						1	0	hold
						1	1	toggle
0	0	1	0	0	1	0	1	hold
						1	0	toggle
						1	1	hold
0	0	1	0	1	1	0	1	1
						1	0	0
						1	1	1
0	0	1	1	0	1	0	1	0
						1	0	1
						1	1	0
0	0	1	1	1	1	0	1	toggle
						1	0	hold
						1	1	toggle

If the ATOM channel is enabled, the CM0 and/or CM1 registers and the ACB42 bit field of the ATOM[i]_CH[x]_CTRL register can be updated by the CPU as long as the first match event occurs in case of a 'serve last' compare strategy or as long as the overall match event in case of the other compare strategies.

After a compare match event that causes an update of the shadow registers SR0/SR1 and before reading the SR0 and/or SR1 register via ARU, the update of the registers CM0 and/or CM1 is possible but has no effect.

To set up a new compare action, first the SR0 and/or SR1 register containing captured values have to be read and then new compare values have to be written into the register CM0 and/or CM1.

Which CMx register has to be updated depends on the compare strategy defined in the ACB42 bit field of the channel control register. Since the channel immediately starts with the comparison after the CMx register was/were written, the compare strategy has to be updated before the CMx registers are written.

For the 'serve last' compare strategies, if the register CM0 and CM1 are updated, it can happen that one or both compare values are already located in the past. In any way the ATOM channel will first wait until both compare values are written before it starts the time base comparisons to avoid a deadlock.

The CPU can check at any time if at least one of the ATOM channels' capture compare register contains valid data and waits for a compare event to happen. This is signaled by the DV bit inside the ATOM[i]_CH[x]_STAT register.

NOTE

For 'serve last' compare strategies, if DV bit is currently not set, writing to CM0 or CM1 sets immediately the DV bit although the compare is only started if both values are written.

An exception for update of register CM0/CM1 exists in SOMC mode and CCUx control mode 'serve last'. If in this mode the CCU0 compare match event occurred, the update of register CM0/CM1 via CPU is blocked until the CCU1 compare match event.

In the 'serve last' mode (ACB42 = 100 or ACB42 = 101) it is possible to generate very small spikes on the output pin by loading CM0 and CM1 with two time stamp values for TBU_TS0, TBU_TS1 or TBU_TS2 close together. The output pin will then be set or reset dependent on the SL bit and the

specified ACB10(5) and ACB10(4) bits in the ACB10 bit field of the ATOM[i]_CH[x]_CTRL register on the first match event and the output will toggle on the second compare event in the CCU1 compare unit.

NOTE

The bigger (smaller) time stamp has to be loaded into the CM1 register, since the CCU0 will enable the CCU1 once it has reached its comparison time stamp. The order of the comparison time stamps depends on the defined greater-equal or less-equal comparison of the CCUx units.

In addition to storing the captured time stamps in the shadow registers, the ATOM channel provides the result of the compare match event in the ACBO(4) and ACBO(3) bits of the ATOM[i]_CH[x]_STAT register. The meaning of the bits is shown in the following table:

Table 38.203 Compare match event ACBO(4) and ACBO(3) bits of ATOM[i]_CH[x]_STAT

ACBO(4)	ACBO(3)	Indication
0	1	CCU0 compare match occurred
1	0	CCU1 compare match occurred

NOTE

In case of the 'serve last' compare strategy, when the bit SLA in the ATOM[i]_CH[x]_CTRL register is not set, the ACBO(4) bit is always set and the ACBO(3) bit is always reset after the compare match event occurred.

The ACBO bit field is reset, when the DV bit is set.

Depending on the capture compare unit where the time base matched the interrupt CCU0TCx_IRQ or CCU1TCx_IRQ is raised.

NOTE

In case of 'serve first' compare strategy, if both events CCU0 and CCU1 occur at the same point in time, both interrupts will be raised.

The behavior of an ATOM channel in SOMC mode under CPU control is depicted in **Figure 38.66**.

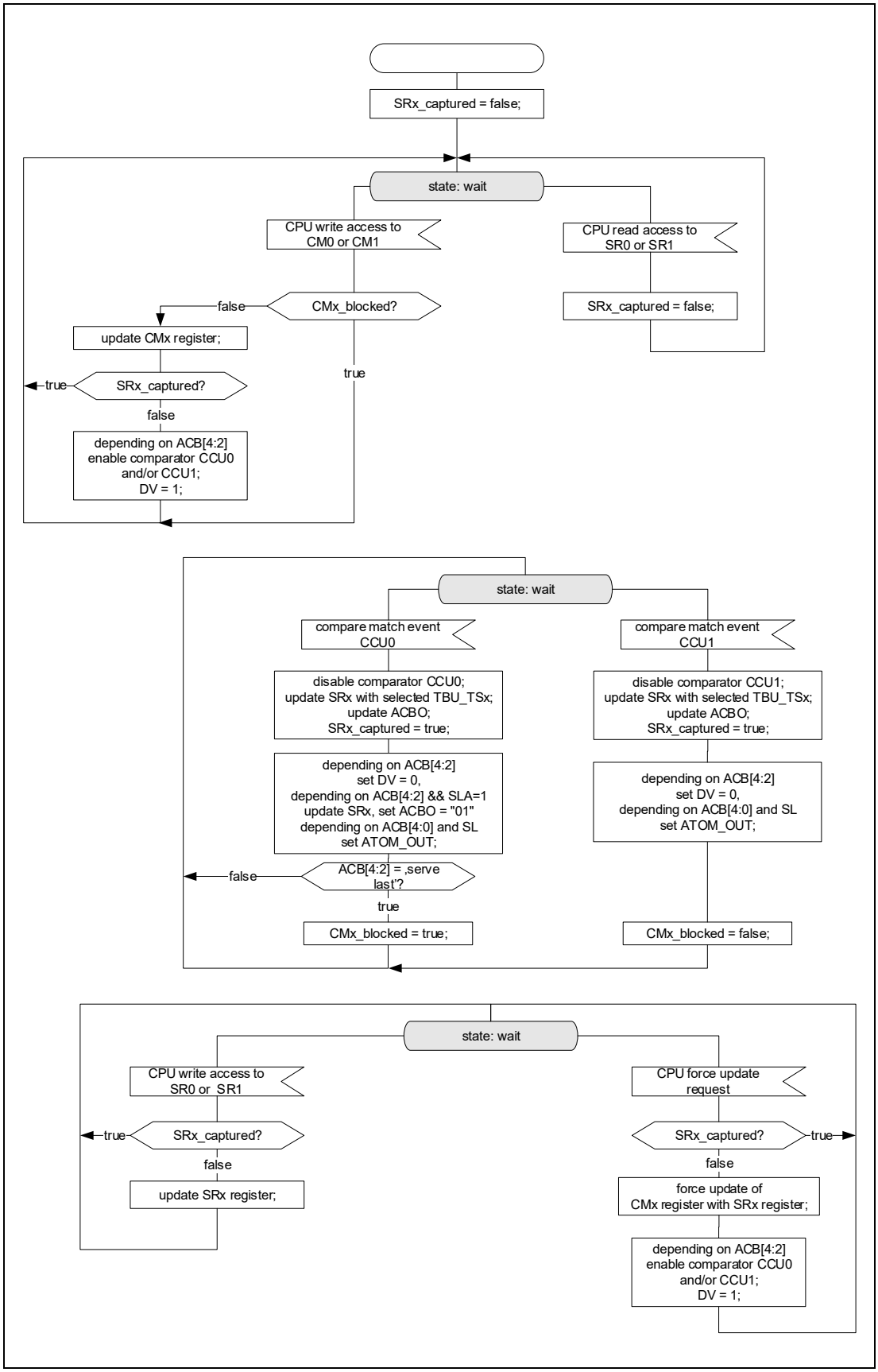


Figure 38.66 SOMC state diagram for channel under CPU control

(3) SOMC Mode under ARU control

When the channel should be controlled by ARU, the ARU_EN bit inside the ATOM[i]_CH[x]_CTRL register has to be set.

In case, the ATOM channel is under ARU control the content for the compare registers CM0 and CM1 as well as the update of the compare strategy can be loaded via the 53 bit ARU word.

The ARU word 23 to 0 is loaded into the CM0 register while the ARU word 47 to 24 is loaded into the CM1 register. The five ARU control bits 52 to 48 are loaded into the ACBI bit field of the ATOM[i]_CH[x]_STAT register and control the channel compare strategy as well as the output behavior in case of compare match events.

For the five ARU control bits 52 to 48 the bits 49 and 48 are loaded into the ACBI bits 1 and 0. The output behavior also depends on the setting of the SL bit inside of the ATOM[i]_CH[x]_CTRL register and is shown in the following table:

Table 38.204 Output behavior depends on SL bit inside of the ATOM[i]_CH[x]_CTRL and ACBI bits 1 and 0

SL	ACBI(1)	ACBI(0)	Output behavior
0	0	0	No signal level change at output (exception in **Table 38.201** and **Table 38.202**)
0	0	1	Set output signal level to 1
0	1	0	Set output signal level to 0
0	1	1	Toggle output signal level (exception in **Table 38.201** and **Table 38.202**)
1	0	0	No signal level change at output (exception in **Table 38.201** and **Table 38.202**)
1	0	1	Set output signal level to 0
1	1	0	Set output signal level to 1
1	1	1	Toggle output signal level (exception in **Table 38.201** and **Table 38.202**)

For the five ARU control bits 52 to 48 the bits 52 to 50 are loaded into the ACBI bits 4 to 2. With these three bits the capture/compare units CCUx can be controlled as shown in the following table:

Table 38.205 Capture/compare units CCUx controlled by ACBI bits 4 to 2 (1/2)

ACBI(4)	ACBI(3)	ACBI(2)	CCUx control
0	0	0	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUx on compare match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACBI(1) and ACBI(0). Details see **Table 38.202**.
0	0	1	Serve First: Compare in CCU0 using *TBU_TS0* and in parallel in CCU1 using *TBU_TS1* or *TBU_TS2*. Disable other CCUx on compare match. Output signal level on the compare match of the matching CCUx unit is defined by combination of SL, ACBI(1) and ACBI(0). Details see **Table 38.201**.
0	1	0	Compare in CCU0 only, use time base *TBU_TS0*. Output signal level is defined by combination of SL, ACBI(1) and ACBI(0) bits.
0	1	1	Compare in CCU1 only, use time base *TBU_TS1* or *TBU_TS2*. Output signal level is defined by combination of SL, ACBI(1) and ACBI(0) bits.

Table 38.205 Capture/compare units CCUx controlled by ACBI bits 4 to 2 (2/2)

ACBI(4)	ACBI(3)	ACBI(2)	CCUx control
1	0	0	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS0*. Output signal level when CCU0 matches is defined by combination of SL, ACBI(1) and ACBI(0). On the CCU1 match the output level is toggled.
1	0	1	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU0 matches is defined by combination of SL, ACBI(1) and ACBI(0). On the CCU1 match the output level is toggled.
1	1	0	Serve Last: Compare in CCU0 using *TBU_TS0* and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU1 matches is defined by combination of SL, ACBI(1) and ACBI(0).
1	1	1	Change ARU read address to ATOM_RDADDR1 DV flag is not set. Neither ACBI(1) nor ACBI(0) is evaluated.

Note: The bit combination 111_B for the ACBI(4), ACBI(3) and ACBI(2) bits forces the channel to request new compare values from another destination read address defined in the ATOM_RDADDR1 bit field of the ATOM[i]_CH[x]_RDADDR register. After data was successfully received and the compare event occurred the ATOM channel switches back to ATOM_RDADDR0 to receive the next data from there.

After the specified compare match event, the captured time stamps are stored in SR0 and SR1 and the compare result is stored in the ACBO bit field of the ATOM[i]_CH[x]_STAT register. The meaning of the ACBO(4) and ACBO(3) bits of the ATOM[i]_CH[x]_STAT is shown in the following table:

Table 38.206 Compare match event ACBO(4) and ACBO(3) bits of ATOM[i]_CH[x]_STAT

ACBO(4)	ACBO(3)	Return value to ARU
0	1	CCU0 compare match occurred
1	0	CCU1 compare match occurred

Note: In case of the “serve last” compare strategy, when the bit SLA in the ATOM[i]_CH[x]_CTRL register is not set, the ACBO(4) bit is always set and the ACBO(3) bit is always reset after the compare match event occurred.

The ACBO bit field is reset, when the DV bit is set.

Depending on the capture compare unit where the time base matched the interrupt CCU0TCx_IRQ or CCU1TCx_IRQ is raised.

When CCU0 and CCU1 is used for comparison it is possible to generate very small spikes on the output pin by loading CM0 and CM1 with two time stamp values for TBU_TS0, TBU_TS1 or TBU_TS2 close together. The output pin will then be set or reset dependent on the SL bit and the specified ACBI(0) and ACBI(1) bits in the ACBI bit field of the ATOM[i]_CH[x]_STAT register on the first match event and the output will toggle on the second match event.

NOTE

The bigger (smaller) time stamp has to be loaded into the CM1 register, since the CCU0 will enable the CCU1 once it has reached its comparison time stamp. The order of the comparison time stamps depends on the defined greater-equal or less-equal comparison of the CCUx units.

For compare strategy 'serve last' the CCU0 and CCU1 compare match may occur sequentially. During different phases of compare match the CPU access rights to register CM0 and CM1 as well as to WR_REQ bit is different. For the case of bit ABM = 0 and EUPM = 0 (register ATOM[i]_CH[x]_CTRL) these access rights by CPU to register CM0 and CM1 and the WR_REQ are depicted in the following figure.

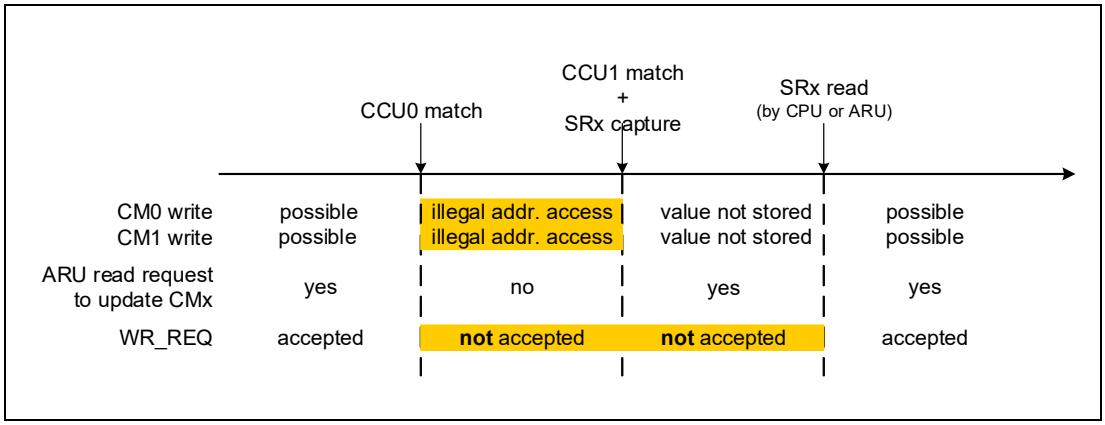


Figure 38.67 CPU access rights in case of compare strategy 'serve last', ABM = 0 and EUPM = 0

For the case of bit ABM = 1 and EUPM = 0 (register ATOM[i]_CH[x]_CTRL) these access rights by CPU to register CM0 and CM1 and the WR_REQ are depicted in the following figure.

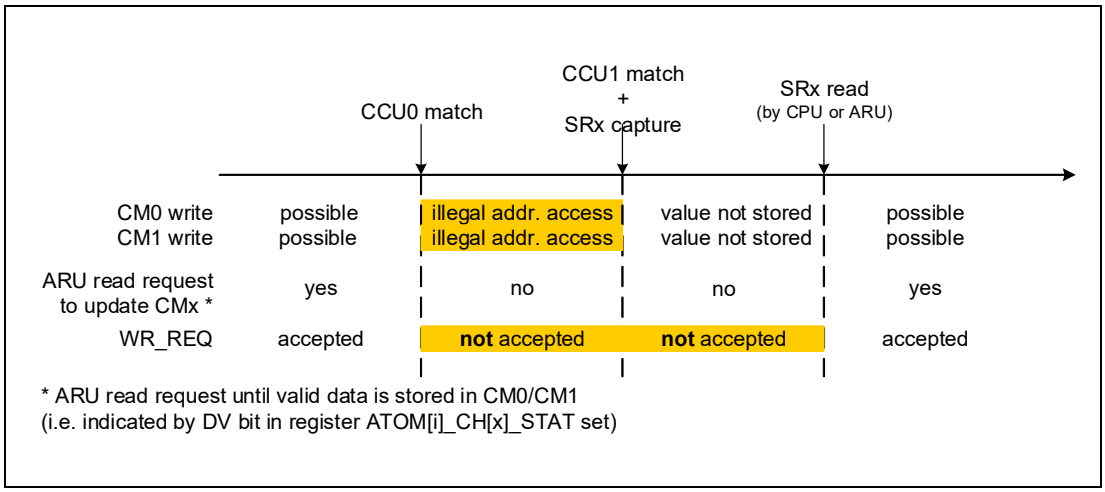


Figure 38.68 CPU access rights in case of compare strategy 'serve last', ABM = 1 and EUPM = 0

For the case of bit EUPM = 1 (register ATOM[i]_CH[x]_CTRL), after CCU0 compare match (and before CCU1 compare match) an update of CM1 as well as a late update via WR_REQ is possible. The value is used for compare. After CCU0 compare match an update of CM0 is not possible, means the value is not stored. The ARU read request is not paused between the compare matches. This behavior is depicted in the following figures.

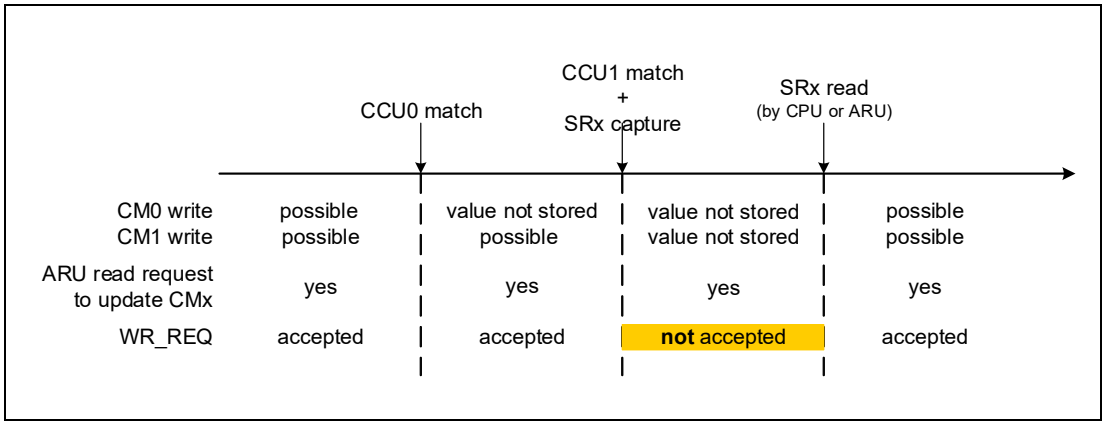


Figure 38.69 CPU access rights in case of compare strategy 'serve last', ABM = 0 and EUPM = 1

The behavior in case of EUPM = 1 and ABM = 1 is depicted in the following figure:

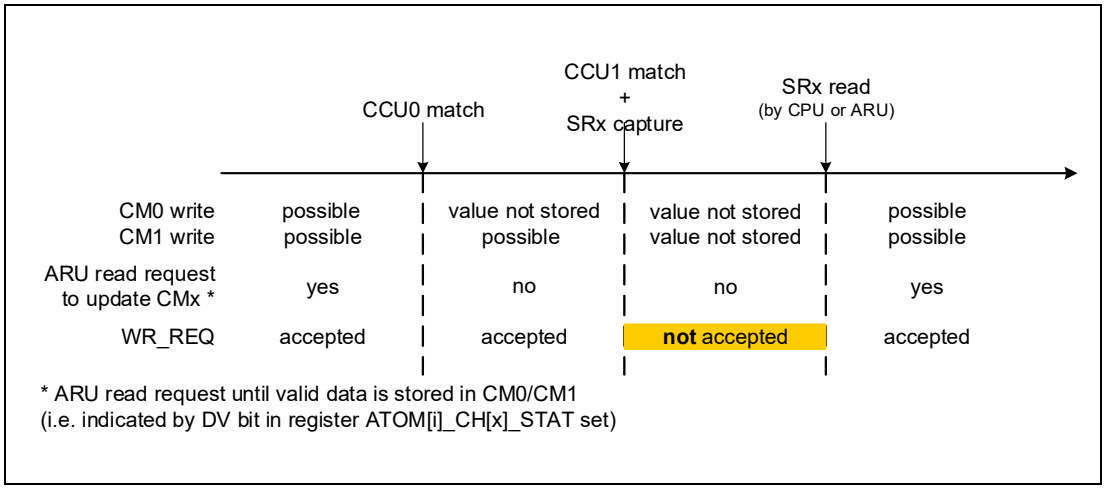


Figure 38.70 CPU access rights in case of compare strategy 'serve last', ABM = 1 and EUPM = 1

In case of EUPM =1 a write access to CM0 or CM1 never causes an AEI write status 10.

(a) ARU Non-Blocking mode

When the compare registers are updated via ARU the update behavior of the channel is configurable with the ABM bit inside the ATOM[i]_CH[x]_CTRL register. When the ABM bit is reset, the ATOM channel is in ARU non-blocking mode.

In the ARU non-blocking mode, data received via ARU is continuously transferred to the registers CM0 and CM1 and the bit field ACBI of register ATOM[i]_CH[x]_STAT as long as no specified compare match event occurs.

After a compare match event that causes an update of the shadow register SR0/SR1 and before reading the SR0 / SR1 register via CPU or ARU, the update of the registers CM0 / CM1 via CPU or ARU is possible but the data is not accepted to be valid (no DV bit is set i register ATOM[i]_CH[x]_CTRL).

To set up a new compare action, first the SR0 / SR1 register containing captured values have to be read and then new compare values have to be written into the register CM0 / CM1. This can be done either by ARU or by CPU.

When the CPU does the register accesses, only one of the shadow registers has to be read. Dependent on the compare strategy, the CPU has to write one or both of the compare registers.

An exception for update of register CM0/CM1 exists in SOMC mode and CCUx control mode 'serve last' if EUPM=0. If in this mode the CCU0 compare match event occurred, the update of register CM0/CM1 via CPU is not possible until the CCU1 compare match event occurs.

NOTE

A write access to either CM0 or CM1 in this case leads to a write status 10.

The CPU can check at any time if the ATOM channel has received valid data from the ARU and waits for a compare event to happen. This is signaled by the DV bit inside the ATOM[i]_CH[x]_STAT register.

The behavior of an ATOM channel in SOMC mode, when ARU is enabled and ARU blocking mode is disabled is shown in **Figure 38.71**.

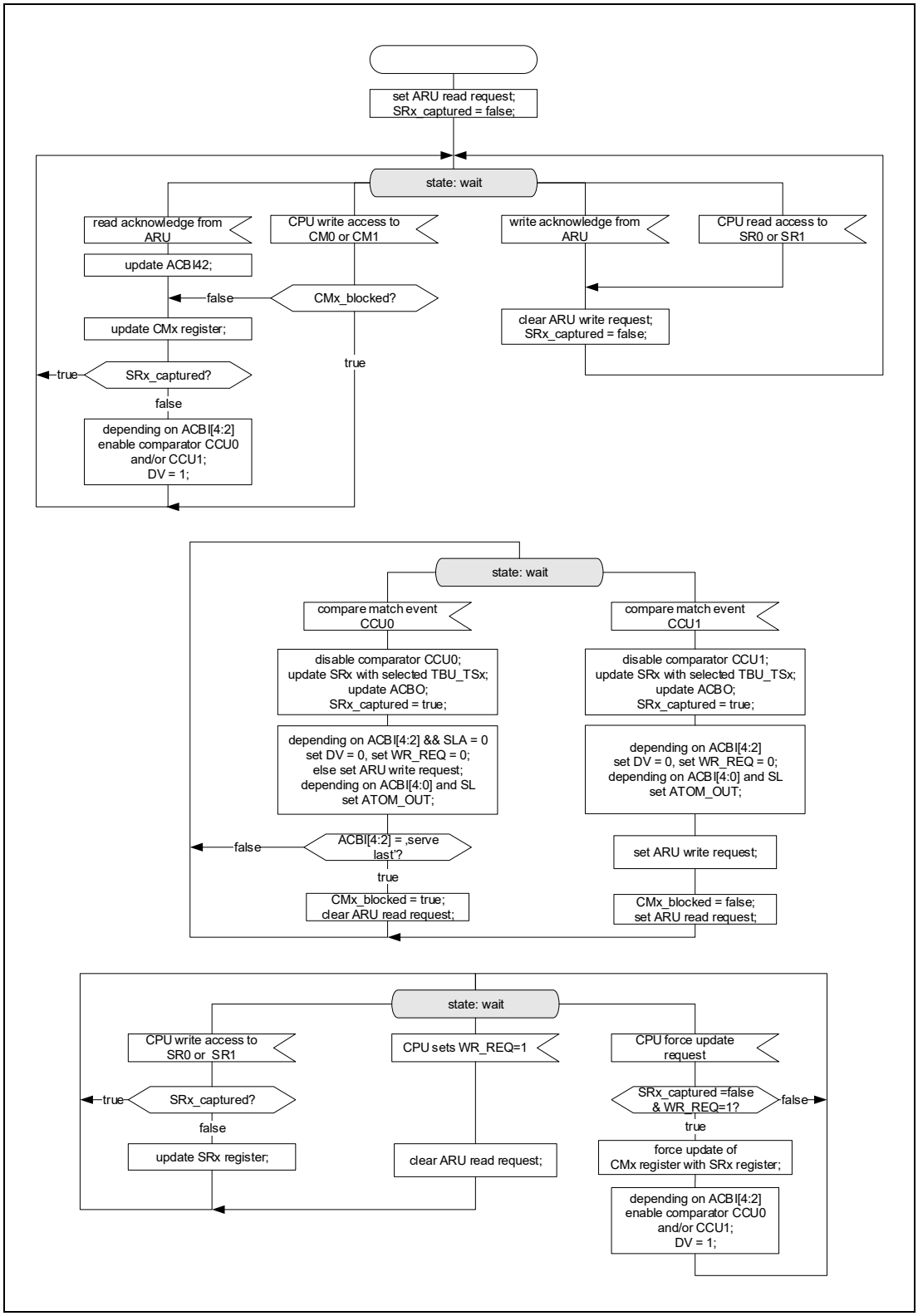


Figure 38.71 SOMC State diagram for SOMC mode, ARU enabled, ABM disabled

(b) ARU Blocking mode

When the compare registers are updated by ARU, the ATOM channel can be configured to receive ARU data in a blocking manner. This can be configured by setting the ABM bit in the ATOM[i]_CH[x]_CTRL register.

If the ABM and ARU_EN bits are set, depending on compare strategy, CM0 and/or CM1 can be updated via ARU with new compare values.

If the compare registers CM0 and/or CM1 are accepting these new data to be valid (indicated by bit DV in register ATOM[i]_CH[x]_STAT), the ATOM channel stops requesting new data via ARU and waits for the compare match event to happen.

When the specified compare match event happens, the shadow registers SR0 and SR1 are updated together with the ACBO bits in the ATOM[i]_CH[x]_STAT register. The data in the shadow registers is marked as valid for the ARU and the DV bit or register ATOM[i]_CH[x]_CTRL is reset.

If the register SR0 and SR1 holding the captured TBU time stamp values are read by either the ARU or the CPU, the next write access to or update of the register CM0 or CM1 via ARU or the CPU enables the new compare match check again.

At least one of the registers SR0 or SR1 has to be read either via ARU or by CPU, before new data is requested via ARU.

NOTE

In case of ABM=1 the application has to handle the situation that the ATOM does not request update of new data for CM0/CM1 until the captured values are read. E.g. if an MCS task starts to write via ARU new data (with AWR(I) command) after capture of data in SR0/SR1, the task sticks in the command until captured data is read by another task via ARU or via the CPU interface.

The CPU can check at any time if the ATOM channel has received valid data from the ARU and waits for a compare event to happen. This is signaled by a set DV bit inside the ATOM[i]_CH[x]_STAT register.

The behavior of an ATOM channel in SOMC mode, when ARU is enabled and ARU blocking mode is enabled is shown in **Figure 38.72**.

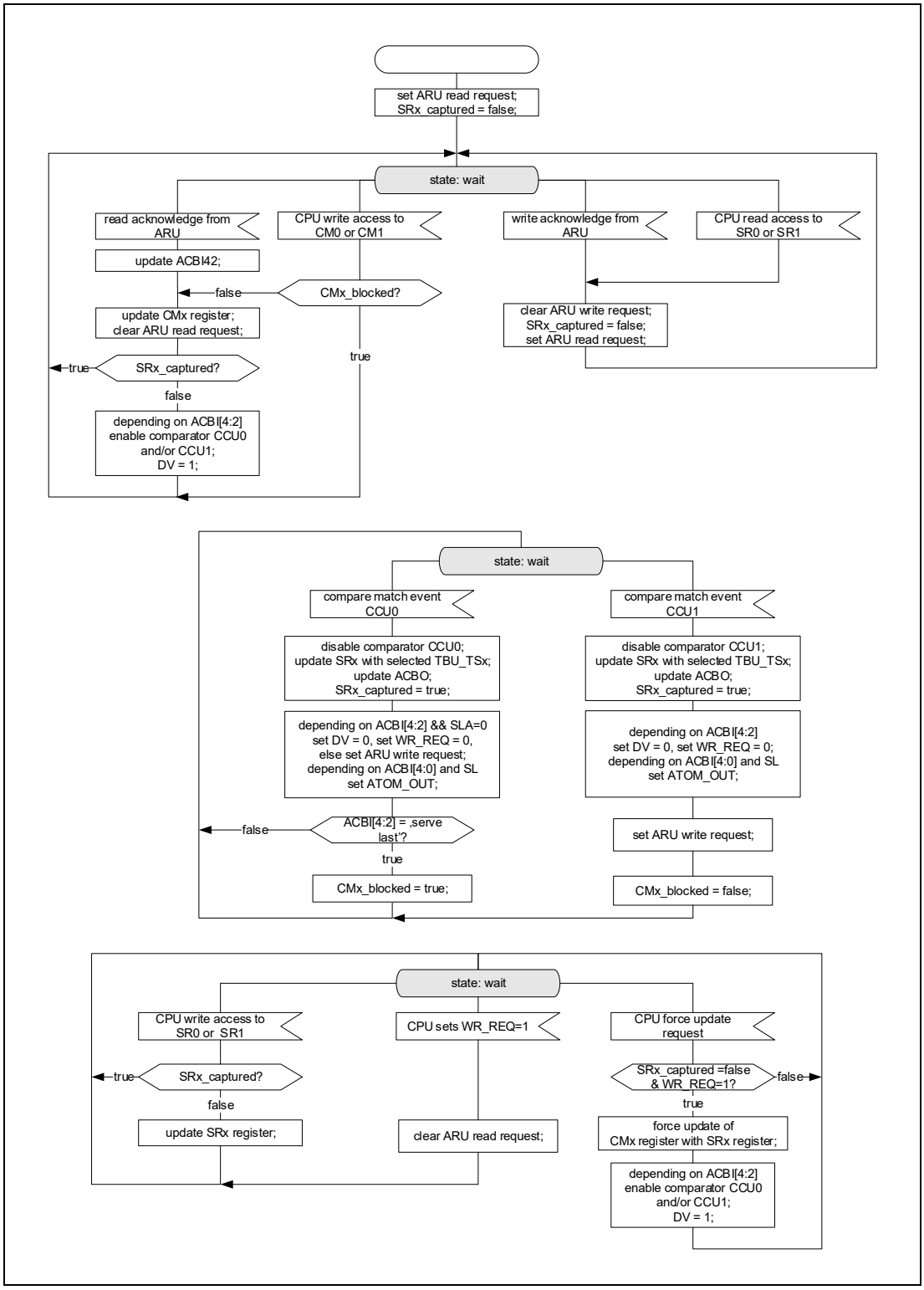


Figure 38.72 SOMC State diagram for SOMC mode, ARU enabled and ABM enabled

(c) ATOM SOMC Late update mechanism

Although, the ATOM channel may be controlled by data received via the ARU, the CPU is able to request at any time a late update of the compare register. This can be initiated by setting the WR_REQ bit inside the ATOM[i]_CH[x]_CTRL register. By doing this, the ATOM will request no further data from ARU (if ARU access was enabled). The channel will in any case continue to compare against the

values stored inside the compare registers (if bit DV was set). The CPU can now update the new compare values until the compare event happens by writing to the shadow registers, and force the ATOM channel to update the compare registers by writing to the force update register bits in the AGC register.

If the WR_REQ bit is set and a compare match event happens, any further access to the shadow registers SR0, SR1 is blocked and the force update of this channel is blocked. In addition, the WRF bit is set in the ATOM[i]_CH[x]_STAT register. Thus, the CPU can determine that the late update failed by reading the WRF bit.

In case of bit EUPM =0 (register ATOM[i]_CH[x]_CTRL) the following statements are true:

If a compare match event already happened, the WR_REQ bit could not be set until the channel is unlocked for a new compare match event by reading the shadow registers. In addition, the WRF bit is set if the CPU tries to write the WR_REQ bit in that case.

In case of bit EUPM = 1 (register ATOM[i]_CH[x]_CTRL) the following statements are true:

If in case of 'serve last' strategy a CCU1 or in any other compare strategy a CCU0 or CCU1 compare match event already happened, the WR_REQ bit could not be set until the channel is unlocked for a new compare match event by reading the shadow registers. In addition, the WRF bit is set if the CPU tries to write the WR_REQ bit in that case.

In general, for a late update the following has to be taken into account:

If between a correct WR_REQ bit set, a correct shadow register write, and before the force update is requested by the AGC a match event occurs on the old compare values, the WRF bit will be set. The force update will be blocked.

The WRF bit will be set in any case if the CPU tries to write to a blocked shadow register.

The WR_REQ bit and the DV bit will be reset on a compare match event.

After a capture event for register SR0 and/or SR1 the force update mechanism will be blocked until a read access to the register SR0 or SR1 by either the ARU or the CPU happens. Writing to SR0 or SR1 after compare match causes an AEI write status 10_B.

The ATOM SOMC late update mechanism from CPU is shown in **Figure 38.73**.

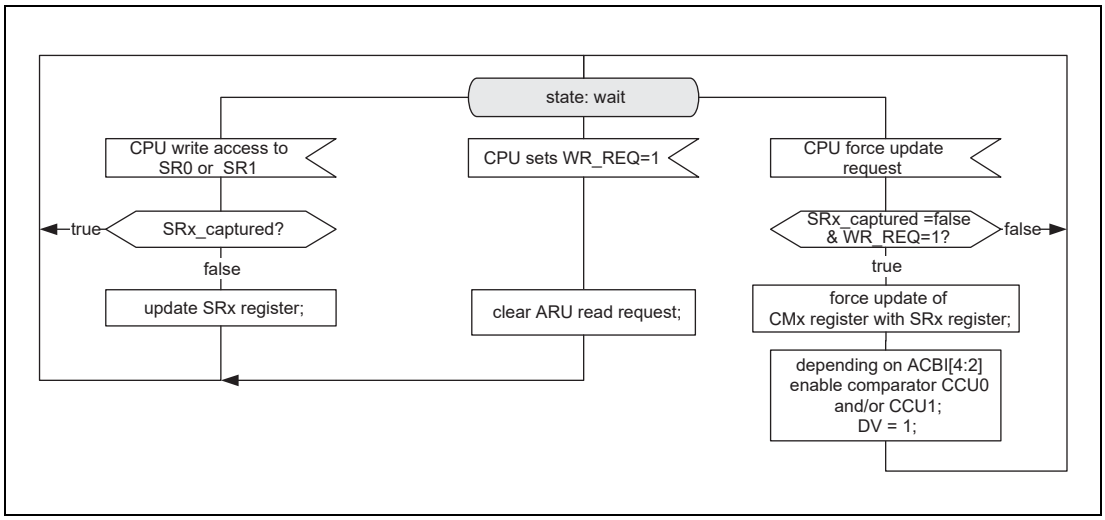


Figure 38.73 SOMC State diagram for late update requests by CPU

38.17.3.3 ATOM Signal Output Mode PWM (SOMP)

In ATOM Signal Output Mode PWM (SOMP) the ATOM sub-module channel is able to generate complex PWM signals with different duty cycles and periods. Duty cycles and periods can be changed synchronously and asynchronously. Synchronous change of the duty cycle and/or period means that the duty cycle or period duration changes after the end of the preceding period. An asynchronous change of period and/or duty cycle means that the duration changes during the actual running PWM period.

The signal level of the pulse generated inside the period can be configured inside the channel control register (SL bit of ATOM[i]_CH[x]_CTRL register). The initial signal output level for the channel is the inverse pulse level defined by the SL bit. **Figure 38.74** depicts this behavior.

The counter CN0 of each channel can run in two different modes depending on configuration of UDMODE in register ATOM[i]_CH[x]_CTRL.

By default the counter counts only up until it reaches CM0 and is then reset to 0. In the up down counter mode CN0 switches between counting up and counting down.

(1) Continuous Counting Up Mode

In SOMP mode with UDMODE = 00_B (i.e. CN0 counts only up), depending on configuration bits RST_CCU0 of register ATOM[i]_CH[x]_CTRL the counter register CN0 can be reset either when the counter value is equal to the compare value CM0 (i.e. CN0 counts only from 0 to CM0 -1 and is then reset to 0) or when signaled by the ATOM[i] trigger signal TRIG_[x-1] of the preceding channel [x-1] (which can also be the last channel of preceding instance TOM[i-1]) or the trigger signal TIM_EXT_CAPTURE(x) of the assigned TIM channel [x]. In this case, if UPEN_CTRL[x] = 1, also the working register CM0, CM1 and CLK_SRC are updated.

NOTE

As an exception, the input TRIG_[0] of instance ATOM0 is triggered by its own last channel 15 via signal TRIG_15

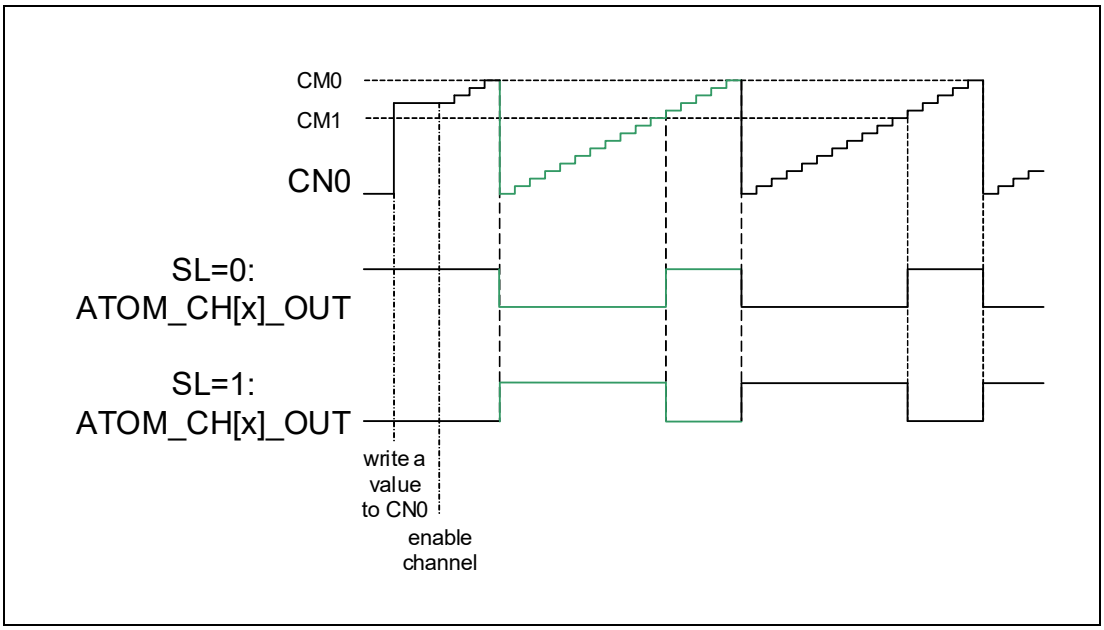


Figure 38.74 PWM Output behavior with respect to the SL bit in the ATOM[i]_CH[x]_CTRL register if UDMODE = 00_B

On an asynchronous update, it is guaranteed, that no spike occurs at the output port of the channel due to a too late update of the operation registers. The behavior of the output signal due to the different possibilities of an asynchronous update during a PWM period is shown in **Figure 38.75**.

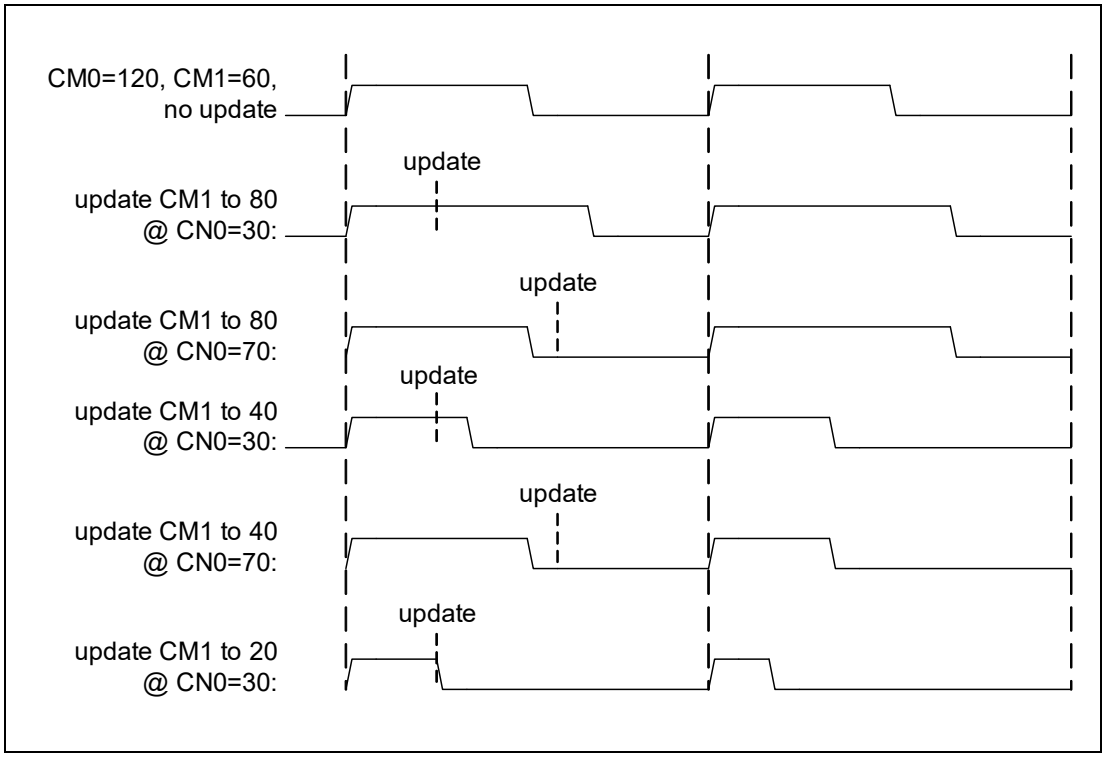


Figure 38.75 PWM Output behavior in case of an asynchronous update of the duty cycle

The duration of the pulse high or low time and period is measured with the counter in sub-unit CCU0. The trigger of the counter is one of the eight CMU clock signals configurable in the channel control register ATOM[i]_CH[x]_CTRL. The register CM0 holds the duration of the period and the register CM1 holds the duration of the duty cycle in clock ticks of the selected CMU clock.

If counter CN0 of channel x is reset by its own CCU0 unit (i.e. the compare match of CN0 ≥ CM0 -1 configured by RST_CCU0 = 0), following statements are valid:

1. CN0 counts from 0 to CM0 -1 and is then reset to 0
2. When CN0 is reset from CM0 to 0, an edge to SL is generated.
3. When CN0 is incrementing and reaches CN0 > CM1, an edge to !SL is generated.
4. If CM0 = 0 or CM0 = 1, the counter CN0 is constant 0.
5. If CM1 = 0, the output is !SL = 0% duty cycle
6. If CM1 ≥ CM0 and CM0 >1, the output is SL = 100% duty cycle

If the counter register CN0 of channel x is reset by the trigger signal coming from another channel or the assigned TIM module (configured by RST_CCU0 = 1), following statements are valid:

- CN0 counts from 0 to MAX–1 and is then reset to 0 by trigger signal – CM0 defines the edge to SL value, CM1 defines the edge to !SL value.
- If CM0 = CM1, the output switches to SL if CN0 = CM0 = CM1 (CM0 has higher priority) – if CM0 =0 and CM1 =MAX, the output is SL = 100% duty cycle – if CM0 > MAX, the output is

!SL = 0% duty cycle, independent of CM1.

In case the counter value CN0 reaches the compare value in register CM0 (in fact CM0 –1) or the channel receives an external update trigger

via the FUPD(x) signal, a synchronous update is performed. A synchronous update means that the registers CM0 and CM1 are updated with the content of the shadow registers SR0 and SR1 and the CLK_SRC register is updated with the value of the CLK_SRC_SR register.

The clock source for the counter can be changed synchronously at the end of a period. If ARU access is disabled, this is done by using the bit field CLK_SRC_SR of register ATOM[i]_CH[x]_CTRL as shadow registers for the next CMU clock source.

(2) Continuous Counting Up-Down Mode

In SOMP mode, if CN0 counts up and down (UDMODE! = 00_B), depending on configuration bit RST_CCU0 of register ATOM[i]_CH[x]_CTRL the counter register CN0 changes the direction either when the counter value is equal to the compare value CM0 (in fact CM0 -1), has counted down to 0 or when triggered by the ATOM[i] trigger signal TRIG_[x-1] of the preceding channel [x-1] (which can also be the last channel of preceding instance ATOM[i-1]) or the trigger signal TIM_EXT_CAPTURE(x) of the assigned TIM channel [x]. In this case, if UPEN_CTRL[x] = 1, also the working register CM0, CM1 and CLK_SRC are updated depending on UDMODE.

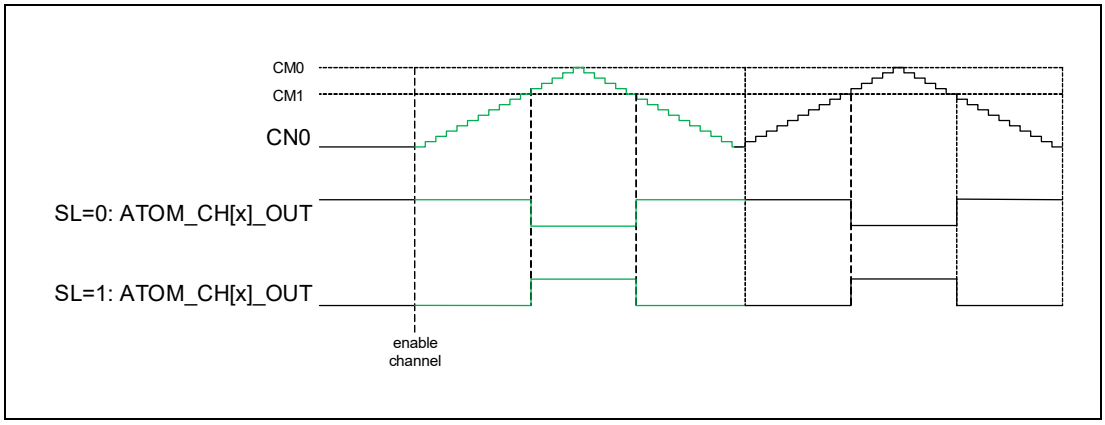


Figure 38.76 PWM Output behavior with respect to the SL bit in the ATOM[i]_CH[x]_CTRL register if UDMODE! = 00_B

The clock of the counter register CN0 can be one of the CMU clocks CMU_CLK[x]. If ARU_EN = 0, the clock for CN0 is defined by CLK_SRC_SR value in register ATOM[i]_CH[x]_CTRL. If ARU_EN = 1, the clock for CN0 is defined by CLK_SRC value received via ARU. The duration of a period in multiples of selected CN0 counter clock ticks is defined by the CM0 configuration value (i.e. CM0 defines half of period in up-down mode). CM1 defines the duty cycle value in clock ticks of selected CN0 counter clock (i.e. CM0 defines half of duty cycle in up-down mode).

If counter CN0 of channel x is reset by its own CCU0 unit (i.e. the compare match of CN0 ≥ CM0 –1 configured by RST_CCU0 = 0), following statements are valid:

1. CN0 counts continuously first up from 0 to CM0 -1 and then down to 0.
2. If CN0 ≥ CM1, the output is set to SL
3. If CM1 = 0, the output is SL (i.e. 100% duty cycle)
4. If CM1 ≥ CM0, the output is !SL (i.e. 0% duty cycle)

5. On output ATOM[i]_CH[x]_OUT a PWM signal is generated. The period is defined by CM0, the duty cycle is defined by CM1.

This behavior is depicted in **Figure 38.76**.

If the counter register CN0 of channel x is reset by the trigger signal coming from another channel or the assigned TIM module (configured by RST_CCU0 = 1), following statements are valid:

1. CN0 counts continuously first up. On a trigger signal the counter switches to count down mode. If CN0 has reached 0, it counts up again.
2. If CN0 ≥ CM1, the output is set to SL
3. If CM1 = 0, the output is SL (i.e. 100% duty cycle)
4. If CM1 ≥ CM0, the output is !SL (i.e. 0% duty cycle)
5. On output ATOM[i]_CH[x]_OUT a PWM signal is generated. The period is defined by the CCU0 trigger of triggering channel, the duty cycle is defined by CM1.
6. On output ATOM[i]_CH[x]_OUT_T a PWM signal is generated. The period is defined by the CCU0 trigger of triggering channel, the duty cycle is defined by CM0.

This behavior is depicted in **Figure 38.77**.

NOTES

1. In case of up-down counter mode and RST_CCU0 = 1 it is recommended that – the triggering channel and the triggered channel are both running in up-down mode and that – the time between two trigger signals is equal to the time needed for CN0 of triggered channel to count back to 0 and again up to the same upper value. The second recommendation can be reached by synchronizing the start of triggering channel and of triggered channel, i.e. let both channel start with a CN0 value 0.
2. If there is a synchronization register in the trigger chain (indicated by value ATOM_TRIG_CHAIN in register CCM[i]_HW_CONF), the additional delay of the trigger by one clock period has to be taken into account by starting at triggering channel with a CN0 value 1 (+1 compared to CN0 of triggered channel).

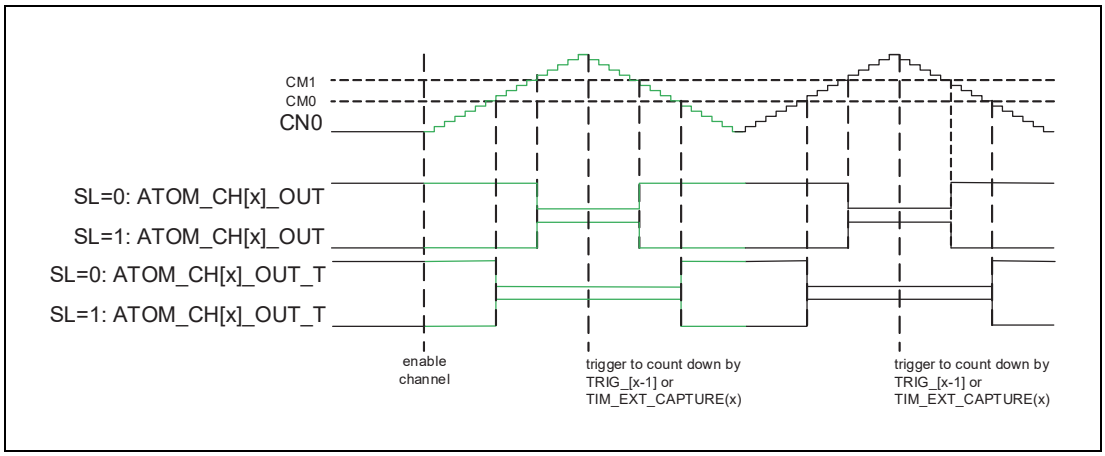


Figure 38.77 PWM Output behavior in case of RST_CCU0 = 1

(3) ARU controlled update

If ARU access is enabled, the bits ACBI(4), ACBI(3) and ACBI(2) received via ARU and stored in register ATOM_[i]_CH[x]_STAT are used as shadow register for the update of the CMU clock source register CLK_SRC.

For the synchronous update mechanism the generation of a complex PWM output waveform is possible without CPU interaction by reloading the shadow registers SR0, SR1 and the ACBI bit field over the ACI sub-unit from the ARU, while the ATOM channel operates on the CM0 and CM1 registers.

This internal update mechanism is established, when the old PWM period ends. The shadow registers are loaded into the operation registers, the counter register is reset, the new clock source according to the CLK_SRC_SR and ACBI(4), ACBI(3) and ACBI(2) bits is selected and the new PWM generation starts.

In parallel, the ATOM channel issues a read request to the ARU to reload the shadow registers with new values while the ATOM channel operates on the operation registers. To guarantee the reloading, the PWM period must not be smaller than the worst case ARU round trip time and source for the PWM characteristic must provide the new data within this time. Otherwise, the old PWM values are used from the shadow registers.

When updated over the ARU the user has to ensure that the new period duration is located in the lower (bits 23 to 0) and the duty cycle duration is located in the upper (bits 47 to 24) ARU data word and the new clock source is specified in the ARU control bits 52 to 50.

This pipelined data stream character is shown in **Figure 38.78**.

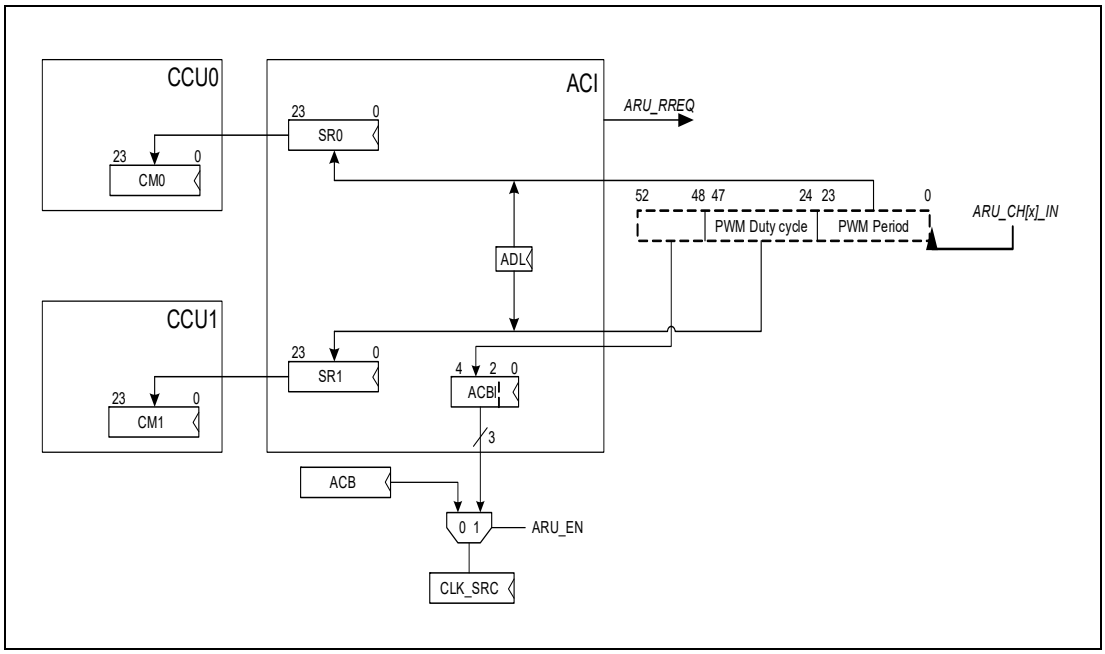


Figure 38.78 ARU Data input stream pipeline structure for SOMP mode

When an ARU transfer is in progress which means the ARU_RREQ is served by the ARU, the ACI locks the update mechanism of CM0, CM1 and CLK_SRC until the read request has finished. The CCU0 and CCU1 operate on the old values when the update mechanism is locked.

(4) CPU controlled update

The shadow registers SR0 and SR1 can also be updated over the AEI bus interface. In this case, ARU_EN hat to be set to 0. When updated via the AEI bus the CM0 and CM1 update mechanism has to be locked via the AGC_GLB_CTRL register with the UPENx signal in the AGC sub-unit. To select the new clock source in this case, the CPU has to write to the CLK_SRC_SR bit field of the ATOM[i]_CH[x]_CTRL register.

For an asynchronous update of the duty cycle and/or period the new values must be written directly into the compare registers CM0 and/or CM1 while the counter CN0 continues counting. This update can be done only via the AEI bus interface immediately by the CPU or by the FUPD(x) trigger signal triggered from the AGC global trigger logic. Values received through the ARU interface are never loaded asynchronously into the operation registers CM0 and CM1. Therefore, the ATOM channel can generate a PWM signal on the output port pin ATOM[i]_CH[x]_OUT on behalf of the content of the CM0 and CM1 registers, while it receives new PWM values via the ARU interface ACI in its shadow registers.

On a compare match of CN0 and CM0 or CM1 the output signal level of ATOM[i]_CH[x]_OUT is toggled according to the signal level output bit SL in the ATOM[i]_CH[x]_CTRL register.

Thus, the duty cycle output level can be changed during runtime by writing the new duty cycle level into the SL bit of the channel configuration register. The new signal level becomes active for the next trigger CCU_TRIGx (since bit SL is written).

Since the ATOM[i]_CH[x]_OUT signal level is defined as the reverse duty cycle output level when the ATOM channel is enabled, a PWM period can be shifted earlier by writing an initial offset value to CN0 register. By doing this, the ATOM channel first counts until CN0 reaches CM0 and then it toggles the output signal at ATOM[i]_CH[x]_OUT.

(5) One-shot Counting Up Mode

The ATOM channel can operate in One-shot mode when the OSM bit is set in the channel control register. One-shot mode means that a single pulse with the pulse level defined in bit SL is generated on the output line.

First the channel has to be enabled by setting the corresponding ENDIS_STAT value.

In one-shot mode the counter CN0 will not be incremented once the channel is enabled.

A write access to the register CN0 triggers the start of pulse generation (i.e. the increment of the counter register CN0).

If the counter CN0 is reset from CM0 –1 back to zero, the first edge at ATOM[i]_CH[x]_OUT is generated.

To avoid an update of CMx register with content of SRx register at this point in time, the automatic update should be disabled by setting UPEN_CTRL[x] = 00_B (in register ATOM[i]_CH[x]_CTRL).

The second edge is generated if CN0 is greater or equal than CM1 (i.e. CN0 was incremented until it has reached CM1 or CN0 is greater than CM1 after an update of CM1).

If the counter CN0 has reached the value of CM0 –1 a second time, the counter stops.

The new value of CN0 determines the start delay of the first edge. The delay time of the first edge is given by (CM0–CN0) multiplied with period defined by current value of CLK_SRC.

Figure 38.79 depicts the pulse generation in SOMP one-shot mode.

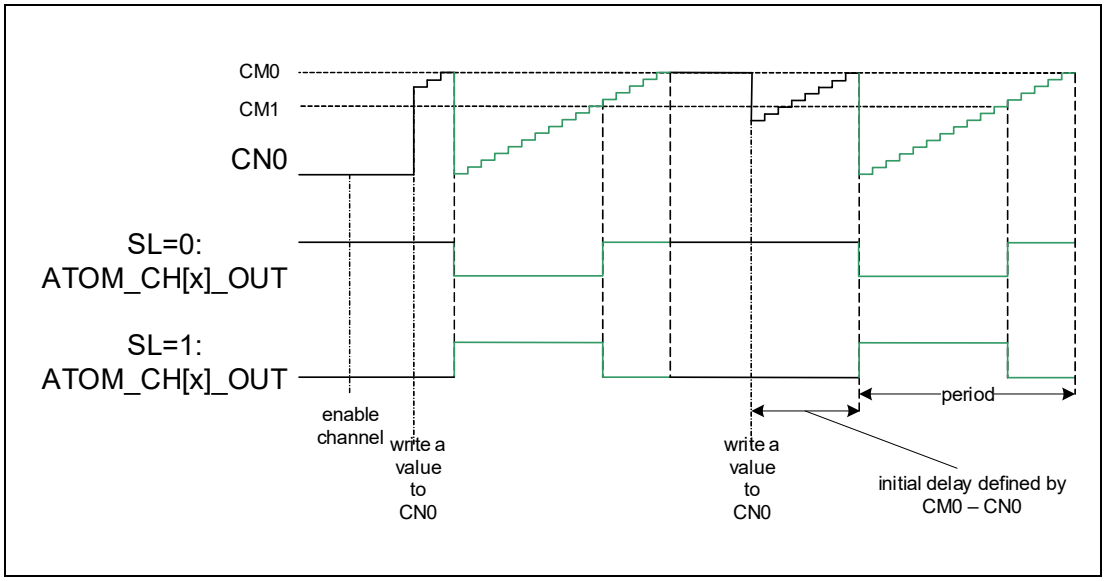


Figure 38.79 PWM Output with respect to configuration bit SL in One-shot counting up mode: trigger by writing to CN0

Further output of single pulses can be started by a write access to register CN0.

If CN0 is already incrementing (i.e. started by writing to CN0 a value CN0start < CM0), the effect of a second write access to CN0 depends on the phase of CN0:

phase 1: update of CN0 before CN0 reaches first time CM0 (in fact CM0 –1)

phase 2: update of CN0 after CN0 has reached first time CM0 (in fact CM0 –1) but is less than CM1

phase 3: update of CN0 after CN0 has reached first time CM0 (in fact CM0 –1) and CN0 is greater than or equal CM1

In phase 1: writing to counter CN0 a value CN0new < CM0 leads to a shift of first edge (generated if CN0 is reset first time from CM0 –1 back to 0) by the time CM0 –CN0new.

In phase 2: writing to incrementing counter CN0 a value CN0new < CM1 while CN0old is below CM1 leads to a lengthening of the pulse. The counter CN0 stops if it reaches CM0.

In phase 3: Writing to incrementing counter CN0 a value CN0new while CN0old is already greater than or equal CM1 leads to an immediate restart of a single pulse generation inclusive the initial delay defined by CM0 – CN0new.

If a channel is configured to one-shot mode and configuration bit OSM_TRIG is set to 1, the trigger signal OSM_TRIG (i.e. TRIG_[x–1] or TIM_EXT_CAPTURE(x)) triggers start of one pulse generation.

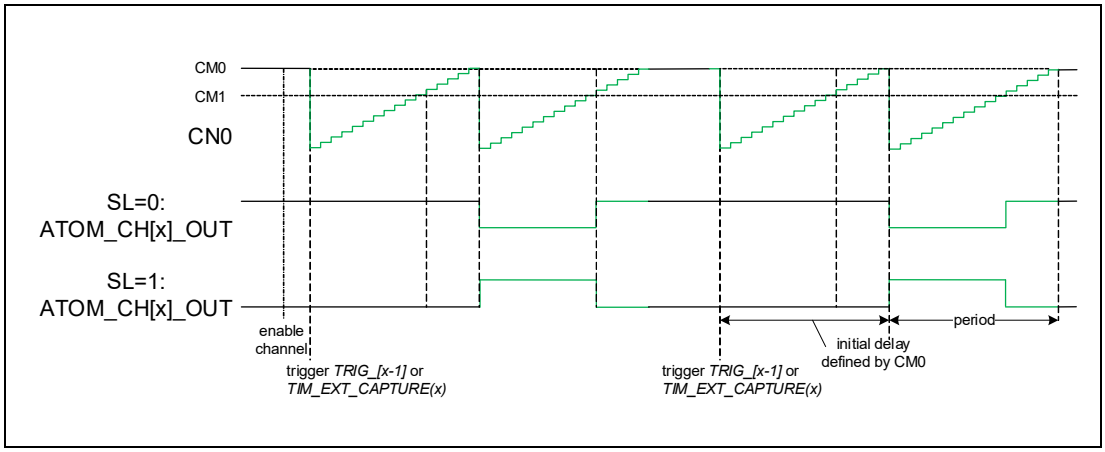


Figure 38.80 PWM Output with respect to configuration bit SL in one-shot mode: trigger by TRIG_[x–1] or TIM_EXT_CAPTURE(x)

(6) One-shot Counting Up-Down Mode

The ATOM channel can operate in one-shot counting up-down mode when the bit OSM = 1 and the UDMODE! = 00_B. One-shot mode means that a single pulse with the pulse level defined in bit SL is generated on the output line.

First the channel has to be enabled by setting the corresponding ENDIS_STAT value.

In one-shot mode the counter CN0 will not be incremented once the channel is enabled.

A write access to the register CN0 triggers the start of pulse generation (i.e. the increment of the counter register CN0).

To avoid an update of CMx register with content of SRx register at this point in time, the automatic update should be disabled by writing UPEN_CTRL[x] = 01_B (see register ATOM[i]_AGC_GLB_CTRL).

If the counter CN0 is greater or equal than CM1, the output ATOM[i]_CH[x]_OUT is set to SL value.

If the counter CN0 is less than CM1, the output ATOM[i]_CH[x]_OUT is set to !SL value.

If the counter CN0 has reached the value 0 (by counting down), it stops.

The new value of CN0 determines the start delay of the first edge. The delay time of the first edge is given by (CM1–CN0) multiplied with period defined by current value of CLK_SRC.

Figure 38.81 depicts the pulse generation in SOMP one-shot mode.

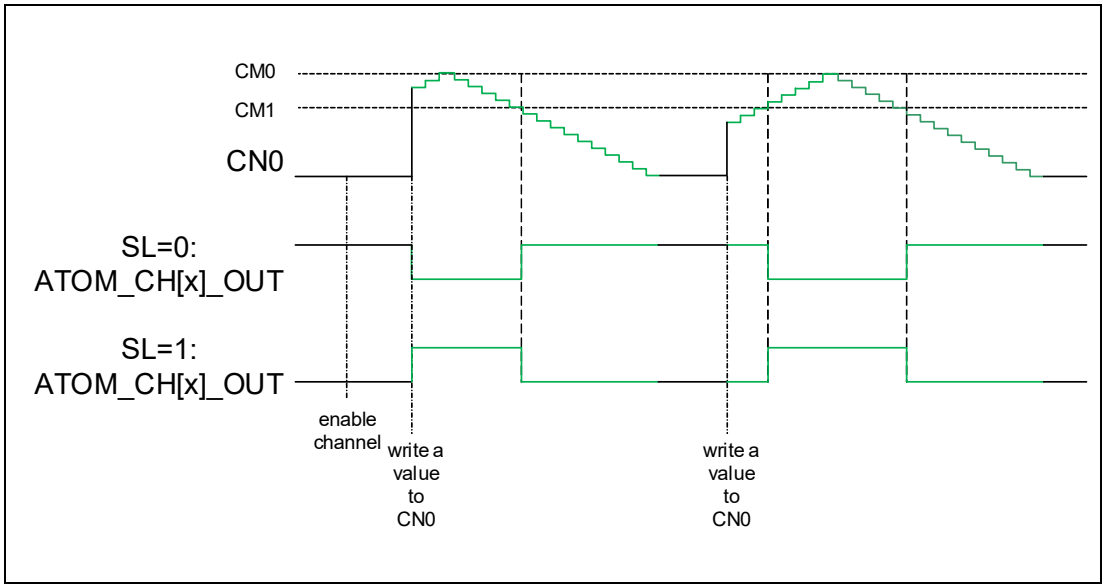


Figure 38.81 PWM Output with respect to configuration bit SL in one-shot counting up-down mode: trigger by writing to CN0

Further output of single pulses can be started by writing to register CN0.

If a channel is configured to one-shot counting up-down mode and configuration bit OSM_TRIG is set to 1, the trigger signal OSM_TRIG (i.e. TRIG_[x–1] or TIM_EXT_CAPTURE(x)) triggers start of one pulse generation.

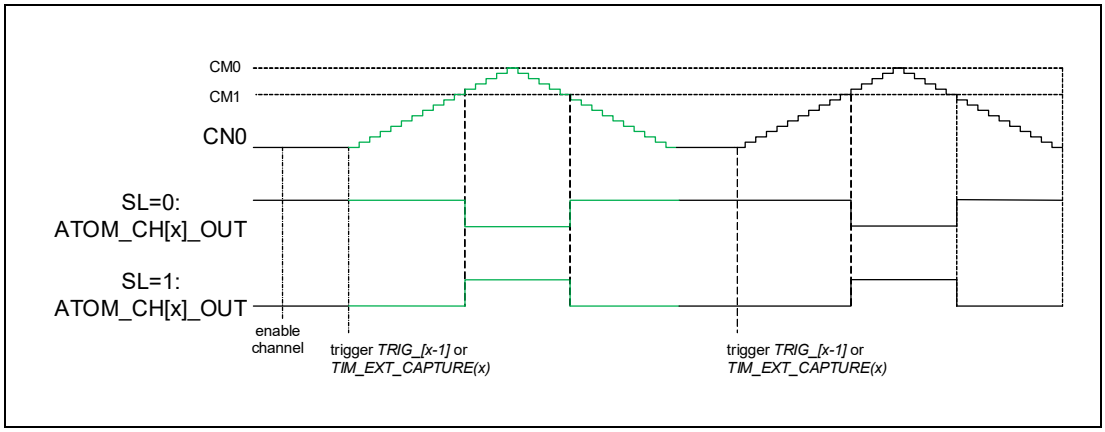


Figure 38.82 PWM Output with respect to configuration bit SL in one-shot counting up-down mode: trigger by TRIG_[x–1] or TIM_EXT_CAPTURE(x)

(7) Pulse Count Modulation Mode

At the output ATOM[i]_CH[x]_OUT a pulse count modulated signal can be generated instead of the simple PWM output signal in SOMP mode.

The PCM mode is enabled by setting bit BITREV to 1 (bit 6 in ATOM[i]_CH[x]_CTRL register).

NOTE

It is device specific, in which channel the PCM mode is available. Refer to device specific **38.28, GTM Device 358** for this information.

With the configuration bit BITREV =1 a bit-reversing of the counter output CN0 is configured. In this case the bits LSB and MSB are swapped, the bits LSB + 1 and MSB–1 are swapped, the bits LSB + 2 and MSB–2 are swapped and so on.

The effect of bit-reversing of the CN0 register value is shown in the following **Figure 38.83**.

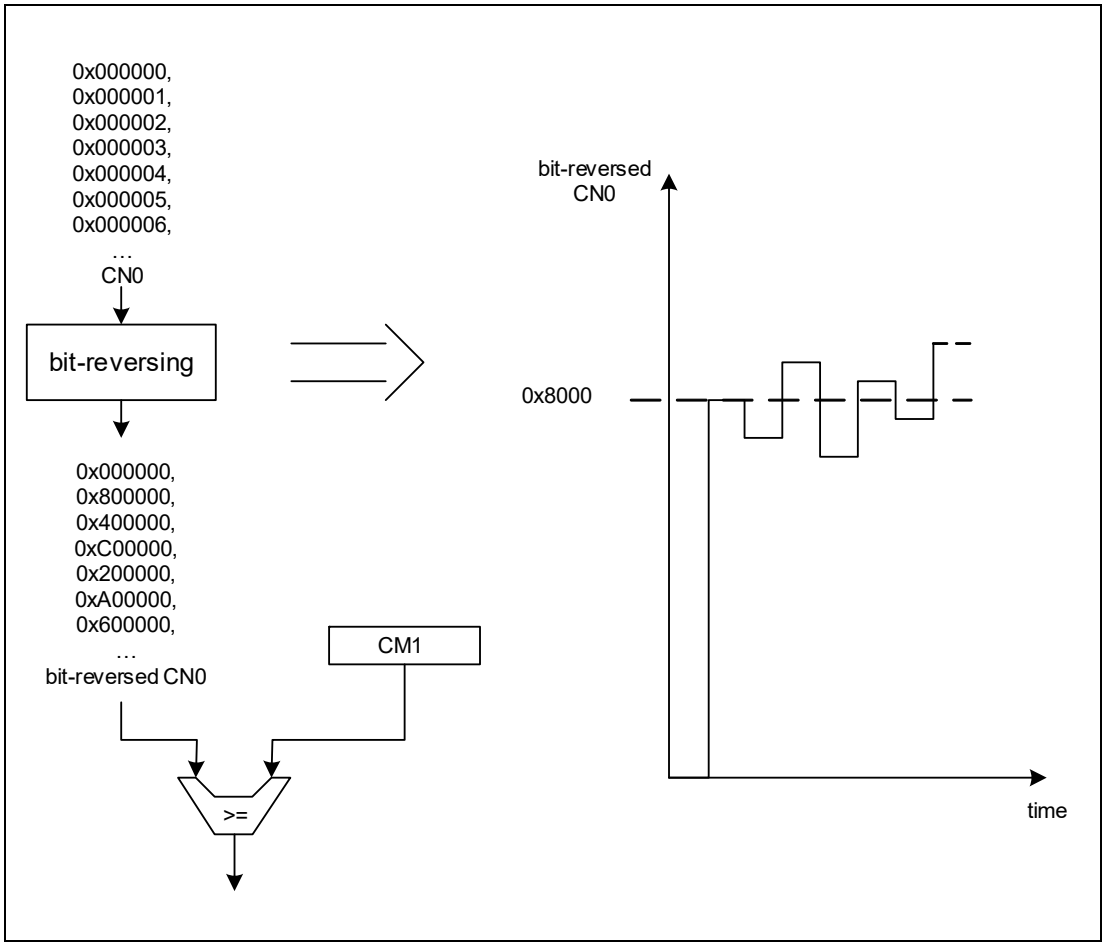


Figure 38.83 Bit reversing of counter CN0 output

In the PCM mode the counter register CN0 is incremented by every clock tick depending on configured CMU clock (CMU_CLK).

The output of counter register CN0 is first bit–reversed and then compared with the configured register value CM1.

If the bit-reversed value of register CN0 is greater or equal than CM1, the SR-FlipFlop of sub-module SOU is set (i.e. set to inverse value of SL) otherwise the SR–FlipFlop is reset (i.e. to the value of SL). This generates at the output ATOM[i]_CH[x]_OUT a pulse count modulated signal.

In PCM mode the CM0 register – in which the period is defined – normally has to be set to its maximum value $FFFFFF_H$.

To reduce time period of updating duty cycle value in CM1 register, it is additionally possible to setup period value in CM0 register to smaller values than maximum value as described before.

Possible values for CM0 register are each even numbered values to the power of 2 e.g. 800000_H, 400000_H, 200000_H ... .

In this case the duty cycle has to be configured in the following manner.

Depending on how much the period in CM0 register is decreased – means shifted right starting from 1000000_H – the duty cycle in CM1 register has to be shifted left (= rotated: shift MSB back into LSB) with same value, e.g. :

period CM0 = 001000_H → shifted 8 bits right from 1000000_H

→ so duty cycle has to be shifted left 8 bit:

e.g. 50% duty cycle = 0008000_H → shift 8 bits left → CM1 = 800000_H

More examples:

period CM0	→	duty cycle	→	shift	→	CM1
$FFFFFF_H$	→	800000_H	→	no shift	→	800000_H
800000_H	→	400000_H	→	shift 1 bit left	→	800000_H
400000_H	→	100000_H	→	shift 2 bits left	→	400000_H
200000_H	→	$0FFFFF_H$	→	shift 3 bits left	→	$7FFFF8_H$
100000_H	→	033333_H	→	shift 4 bits left	→	333330_H
080000_H	→	005555_H	→	shift 5 bits left	→	$0AAAA0_H$
...						
000020_H	→	000008_H	→	shift 19 bits left	→	400000_H
000010_H	→	000005_H	→	shift 20 bits left	→	500000_H
...						

NOTE

In this mode the interrupt CCU1TC (see register ATOM[i]_CH[x]_IRQ_NOTIFY) is set every time if bit reverse value of CN0 is greater or equal than CM1 which may be multiple times during one period. Therefore, from application point of view it is not useful to enable this interrupt.

(8) Trigger generation

For applications with constant PWM period defined by CM0, it is not necessary to update regularly the CM0 register with SR0 register.

For these applications the SR0 register can be used to define an additional output signal and interrupt trigger. If bit SR0_TRIG in register ATOM[i]_CH[x]_CTRL is set, the register SR0 is no longer used as a shadow register for register CM0. Instead, SR0 is compared against CN0 and if both are equal, a pulse of signal level 1 is generated at the output ATOM[i]_CH[x]_OUT_T. The bit SR0_TRIG should only be set if bit RST_CCU0 of this channel is 0.

NOTE

If ARU_EN =1 and both SR0 and SR1 are updated via ARU, the new SR0 value is used immediately after update. Update of SR0 via ARU can be suppressed by ADL configuration in register ATOM[i]_CH[x]_CTRL.

If bit SR0_TRIG is set the interrupt notify flag CCU1TC is no longer set on a compare match of CM1 and CN0. Instead, the CCU1TC interrupt notify flag is set in case of a compare equal match of SR0 and CN0.

With configuration bit TRIG_PULSE one can select if the output ATOM[i]_CH[x]_OUT_T is high as long as CN0 = SR0 (TRIG_PULSE = 0) or if there will be only one pulse of length one SYS_CLK period when CN0 becomes SR0 (TRIG_PULSE = 1).

The ATOM output signal routing to DTM or GTM-IP top level is described in **Section 38.18.7, DTM connections on GTM-IP top level**.

38.17.3.4 ATOM Signal Output Mode Serial (SOMS)

In ATOM Signal Output Mode Serial (SOMS) the ATOM channel acts as a serial output shift register where the content of the CM1 register in the CCU1 unit is shifted out whenever the unit is triggered by the selected CMU_CLK input clock signal. The shift direction is configurable with the ACB(0) bit inside the ATOM[i]_CH[x]_CTRL register when ARU is disabled and the ACBI(0) bit inside the ATOM[i]_CH[x]_STAT register when ARU is enabled.

The data inside the CM1 register has to be aligned according to the selected shift direction in the ACB(0)/ACBI(0) bit. This means that when a right shift is selected, that the data word has to be aligned to bit 0 of the CM1 register and when a left shift is selected, that the data has to be aligned to bit 23 of the CM1 register.

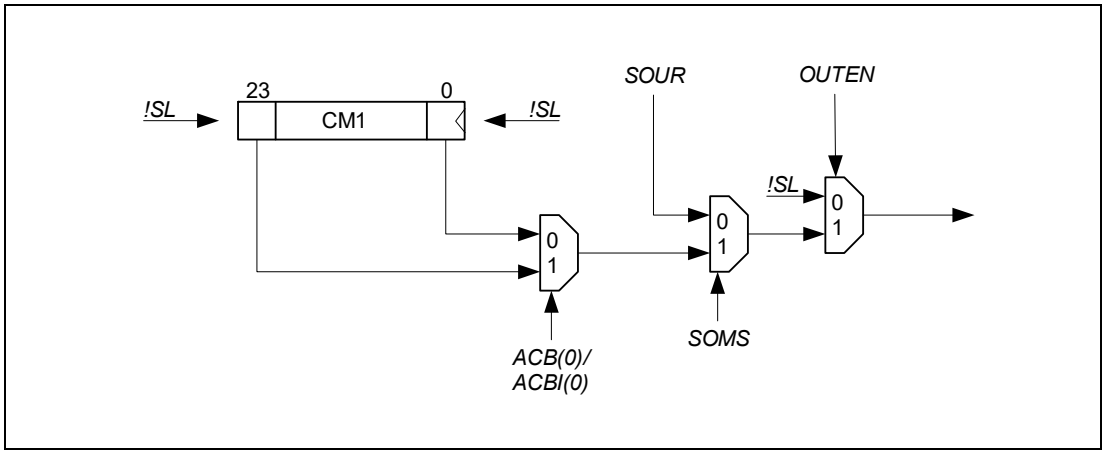


Figure 38.84 SOMS Mode output generation

Figure 38.84 shows the output generation in case of SOMS mode is selected.

In SOMS mode CCU0 runs in counter/compare mode and counts the number of bits shifted out so far. The total number of bits that should be shifted is defined as CM0. The total number of bits that are visible at ATOM_OUT is CM0 + 1.

When the output is disabled the ATOM_OUT is set to the inverse SL bit definition.

When the content of the CM1 register is shifted out, the inverse signal level is shifted into the CM1 register.

When the output is enabled while UPEN_CTRL[x] is disabled, the ATOM_OUT signal level is defined by CM1 bit 0 or 23, dependent on the shift direction defined by ACB(0) or ACBI(0) register setting. **Figure 38.85** should clarify the ATOM channel startup behavior in this case for right shift. For left shift the CM1 bit 0 in **Figure 38.85** has to be replaced by CM1 bit 23.

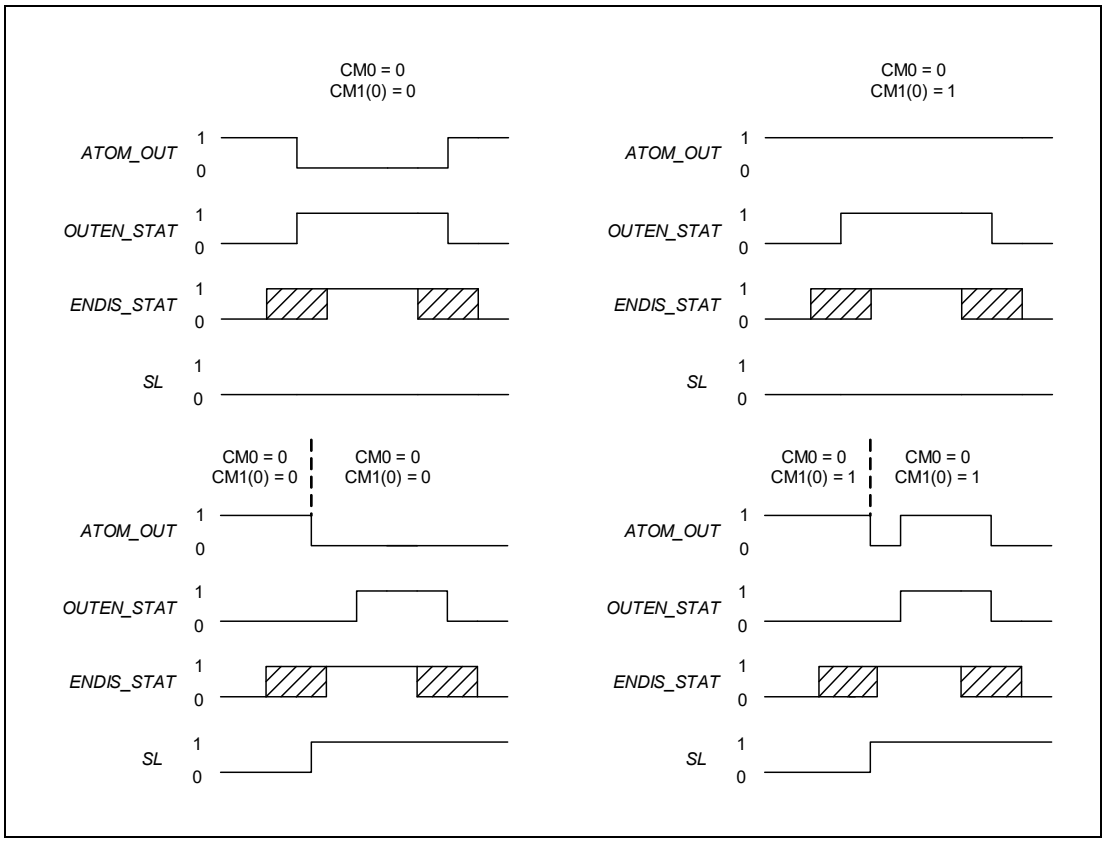


Figure 38.85 SOMS Output signal level at startup, UPEN_CTRL[x] disabled

If UPEN_CTRL[x] is set and the channel is enabled, the output level is defined by bit 0 or 23 of CM1 register dependent on the shift direction.

Figure 38.86 shows the output behavior in that case.

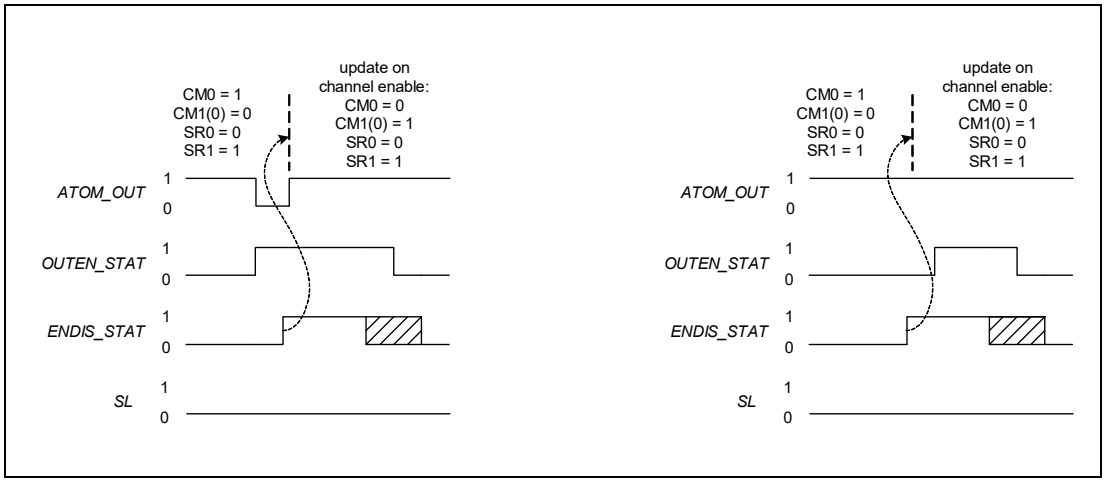


Figure 38.86 SOMS Output signal level at startup, UPEN_CTRL[x] enabled

When the serial data to be shifted is provided via ARU the number of bits that should be shifted has to be defined in the lower 24 bits of the ARU word (23 to 0) and the data that is to be shifted has to be defined in the ARU bits 47 to 24 aligned according to the shift direction. This shift direction has to be defined in the ARU word bit 48 (ACB0 bit).

If bit UPEN_CTRL[x] of a channel x is set, after update of CM0/CM1 register with the content of the SR0 / SR1 register, a new ARU read request is set up.

If bit UPEN_CTRL[x] of a channel x is not set, no (further) ARU read request is set up (because the SR0 / SR1 register are never used for update) and the ATOM may stop shifting after CN0 has reached CM0.

NOTE

In this case also no automatic restart of shifting is possible.

If a channel is enabled with the settings SOMS mode and ARU_EN = 1, the first received values from ARU are stored in register SR0 and SR1. If CN0 and CM0 are 0 (i.e. CN0 is not counting) and the update of channel x is enabled (UPEN_CTRL[x] = 1), an immediate update of the register CM0 and CM1 is also done. This update of CM0 and CM1 triggers the start of shifting.

It is recommended to configure the ATOM channel in One-shot mode when the ARU_EN bit is not set, since the ATOM channel would reload new values from the shadow registers when CN0 reaches CM0.

(1) SOMS mode with ARU_EN = 1 and OSM = 0, UPEN_CTRL[x] = 1:

In case of bit ARU_EN is set and bit OSM is not set, the channel is running in the SOMS continuous mode. Then, if the content of the CM0 register equals the counter CN0, the CM0 and CM1 registers are reloaded with the SR0 and SR1 content and new values are requested from the ARU. If the update of the shadow registers does not happen before CN0 reaches CM0 the old values of SR0 and SR1 are used to reload the operation registers.

In contrast to controlling the channel via AEI, the shift direction defined by ARU word bit 48 has only effect after the update of CMx operation registers from the SRx registers.

(2) SOMS mode with ARU_EN = 1 and OSM = 1, UPEN_CTRL[x] = 1:

In case of bit ARU_EN is set and bit OSM is set, the channel is running in the SOMS one-shot mode. Then, if the content of the CM0 register equals the counter CN0 and if new values are available in SR0 and SR1 (bit DV set), the CM0 and CM1 registers are reloaded with the SR0 and SR1 content and new values are requested from the ARU. If no new values are available in SR0 and SR1, the register CM0 and CM1 will not be updated, the counter CN0 stops and the ATOM channel continues to request new data from ARU. A later reception of new ARU data in SR0 and SR1 will immediately force the update of the register CM0 and CM1 and restart the counter CN0.

(3) SOMS mode with ARU_EN = 0 and OSM = 0, UPEN_CTRL[x] = 1:

In case of bit ARU_EN is not set and bit OSM is not set, the ATOM channel updates its CM0/CM1 register with the content of the SR0/SR1 register and restarts shifting immediately. The first bit of new CM1 register value will be applied at the output without any gap to the last bit of the previous CM1 register value.

(4) SOMS mode with ARU_EN = 0 and OSM = 1, UPEN_CTRL[x] = 1:

In case of bit ARU_EN is not set and bit OSM is set, the ATOM channel stops shifting when CN0 reaches CM0 and no update of CM0 and CM1 is performed.

Then, the shifting of the channel can be restarted again by writing a zero to the CN0 register again.

The writing of a zero to CN0 causes also an immediate update of CM0/CM1 register with the content of SR0/SR1 register.

(5) SOMS mode with double output

If in SOMS mode additionally the mode bit DSO is set (in register ATOM[i]_CH[x]_CTRL) two 12 bit data streams can be shifted out on the outputs ATOM_OUT and ATOM_OUT_T in parallel. This is reached by splitting the register CM1 into two parts. The lower 12 bits are used as a shift register of output ATOM_OUT (i.e. bit 0 is assigned to output ATOM_OUT), the upper 12 bits are used as a shift register of output ATOM_OUT_T (i.e. bit 12 is assigned to output ATOM_OUT_T). On bit 23 and 11 of register CM1 the value !SL is shifted in.

NOTE

In this mode only shift right is possible. Bit ACB0 is ignored.

This behavior is depicted in the following figure:

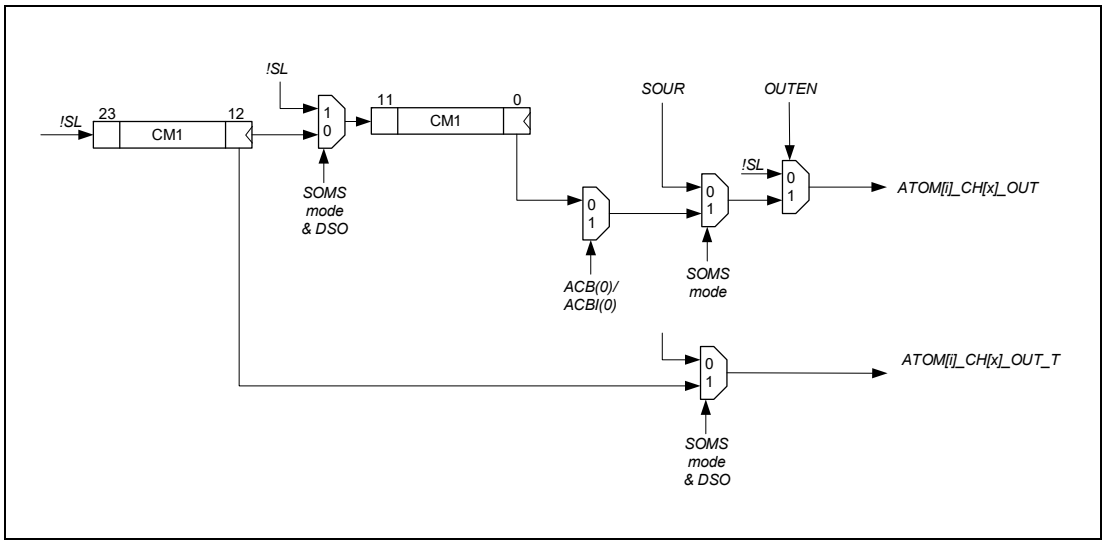


Figure 38.87 Double Output Shift Mode

(6) Interrupts in SOMS mode

In ATOM Signal Output Mode Serial only the interrupt CCU0TC (ATOM[i]_CH[x]_IRQ_NOTIFY) in case of CN0 ≥ CM0 is generated. The interrupt CCU1TC has no meaning and is not generated.

38.17.3.5 ATOM Signal Output Mode Buffered Compare (SOMB)

(1) Overview

In ATOM Signal Output Mode buffered Compare (SOMB) the output action is performed according to the comparison result of the input values located in CM0 and/or CM1 registers and the two (three) time base values TBU_TS0 or TBU_TS1 (or TBU_TS2) provided by the TBU. For a description of the time base generation refer to the TBU specification in **Section 38.14**. It is configurable, which of the two (three) time bases is to be compared with one or both values in CM0 and CM1.

The compare strategy of the two compare units CCU0 and CCU1 is controlled by the value of bit field ACBI of register ATOM[i]_CH[x]_STAT.

Table 38.207 ATOM SOMB compare strategies

ACBI(4)	ACBI(3)	ACBI(2)	CCUx control
0	0	0	Reserved. Has no effect.
0	0	1	Reserved. Has no effect.
0	1	0	Compare in CCU0 only, use time base *TBU_TS0*. Output signal level is defined by combination of SL, ACB10/ACBI(1..0) bits.
0	1	1	Compare in CCU1 only, use time base *TBU_TS1* or *TBU_TS2*. Output signal level is defined by combination of SL, ACBI[1:0] bits.
1	0	0	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS0*. Output signal level when CCU0 matches is defined by combination of SL, ACBI[1:0]. On the CCU1 match the output level is toggled.
1	0	1	Serve Last: Compare in CCU0 and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU0 matches is defined by combination of SL, ACBI[1:0]. On the CCU1 match the output level is toggled.
1	1	0	Serve Last: Compare in CCU0 using *TBU_TS0* and then in CCU1 using *TBU_TS1* or *TBU_TS2*. Output signal level when CCU1 matches is defined by combination of SL, ACBI[1:0]
1	1	1	Cancels pending comparison independent on ARU_EN

This bit field is only readable by CPU. If ARU is disabled, the bit field ACBI can only be updated with the value of bit field ACB of register ATOM[i]_CH[x]_CTRL. If ARU is enabled, the ACBI bit field can be updated with the value of shadow register ACB_SR which contains a value received via ARU or the value of bit field ACB of register ATOM[i]_CH[x]_CTRL.

The table below lists all valid control configurations for bit field ACBI of register ATOM[i]_CH[x]_STAT.

The CCUx trigger signals TRIG_CCU0 and TRIG_CCU1 creates edges depending on the combination of the predefined signal level in SL bit and the two control bits ACBI[1:0].

In SOMB mode, if ARU access is enabled, the new compare values received via ARU are always stored in the shadow register SR0 and SR1 and the ACB bits are stores in an internal register ACB_SR.

If the scheduled compare matches in CCU0 and/or CCU1 are occurred and the SRx register contain new valid values, the register CM0 and CM1 are updated automatically with the content of the corresponding SRx register, the ACBI bit field is updated with the content of internal ACB_SR register and the DV bit of register ATOM[i]_CH[x]_STAT is set. If the SRx register and the CMx register contain no valid value, the compare units are waiting in an idle state.

On a compare match of one of the compare units CCUx units the output ATOM_OUT is set according to combination of ACBI bit 1 down to 0 (in register ATOM[i]_CH[x]_STAT) and the SL bit of register ATOM[i]_CH[x]_CTRL.

Table 38.208 ATOM SOMB output control by ACBI[1:0] and SL

SL	ACBI(1)	ACBI(0)	Output Behavior
0	0	0	No signal level change at output.
0	0	1	Set output signal level to 1.
0	1	0	Set output signal level to 0.
0	1	1	Toggle output signal level.
1	0	0	No signal level change at output.
1	0	1	Set output signal level to 0.
1	1	0	Set output signal level to 1.
1	1	1	Toggle output signal level.

In opposite to SOMC mode no time stamp value of TBU is captured in SRx register.

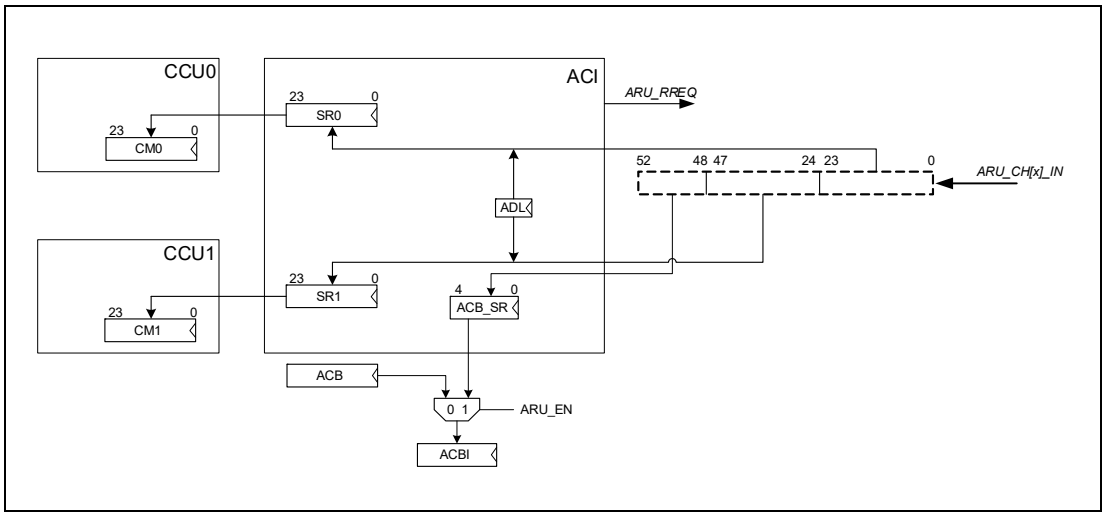


Figure 38.88 ARU interface behavior in SOMB mode

The flag DV of register ATOM[i]_CH[x]_STAT indicates that at least one of the CMx register contains valid data and a compare event may be pending (if channel is enabled). The DV flag is reset if none of the CMx register contains valid data.

(2) SOMB under CPU control

If bit ARU_EN of register ATOM[i]_CH[x]_CTRL is not set, the ATOM channel can only be controlled via CPU.

Writing to one of the CMx register sets automatically the DV bit to validate the new compare value. A comparison depending on value ACBI of register ATOM[i]_CH[x]_STAT is started immediately. Because only the ACB bit of register ATOM[i]_CH[x]_CTRL can be written and this bit field serves as a shadow register for the work register ACBI (bit field of register ATOM[i]_CH[x]_STAT), it is recommended to first update the ACB bit field before updating CMx / SRx register.

The compare strategy is controlled by the value stored in bit field ACBI of register ATOM[i]_CH[x]_STAT. If ARU is disabled, this bit field can only be updated with the value of bit field ACB of register ATOM[i]_CH[x]_CTRL.

The update of bit field ACBI can be triggered by a forced update or the normal update mechanism controlled by bit UPEN_CTRL[x] in register ATOM[i]_AGC_GLB_CTRL.

Writing to one of the SRx register and triggering a forced update, updates the CMx register with the value of SRx register and the ACBI bit field with the content of ACB bit field of register ATOM[i]_CH[x]_CTRL. A new comparison is started.

Writing to one of the SRx register while update of CMx register is disabled (UPEN_CTRL[x] = 0 in ATOM[i]_AGC_GLB_CTRL) and enabling update afterwards, triggers the update of CMx register and the ACBI bit field and starts comparison if previous comparison is finished (DV bit was reset).

If ARU access is disabled (ARU_EN = 0), a force update updates the CMx register with the content of SRx register and the ACBI bit field with the content of ACB bit field of register ATOM[i]_CH[x]_CTRL.

(3) SOMB under ARU control

If both compare units CCU0/CCU1 are finished with previous job (depending on compare strategy) and the SRx register contain no new value, they are waiting until new data was received via ARU and stored in SRx register. Then, an immediately update takes place.

If both compare units are finished with previous job (depending on compare strategy) and there are new data available in SRx register, the update the CMx register with the value of the SRx register and the ACBI bit field with the value of internal ACB_SR register takes place and a new compare job is started immediately.

After an update of the CMx register, a new ARU read request is set.

New compare values received via the ARU are stored in shadow register SRx. The ACB bits received via ARU are stored in the internal register ACB_SR.

If ARU access is enabled (ARU_EN = 1), a force update updates the CMx register with the content of SRx register and the ACBI bit field with the content of internal ACB_SR register.

For compare strategy 'serve last' the CCU0 and CCU1 compare match may occur sequentially. During different phases of compare match the CPU access rights to register CM0 and CM1 as well as to WR_REQ bit is different. These access rights by CPU to register CM0 and CM1 and the WR_REQ are depicted in the following figure for the case of EUPM = 0 (register ATOM[i]_CH[x]_CTRL)

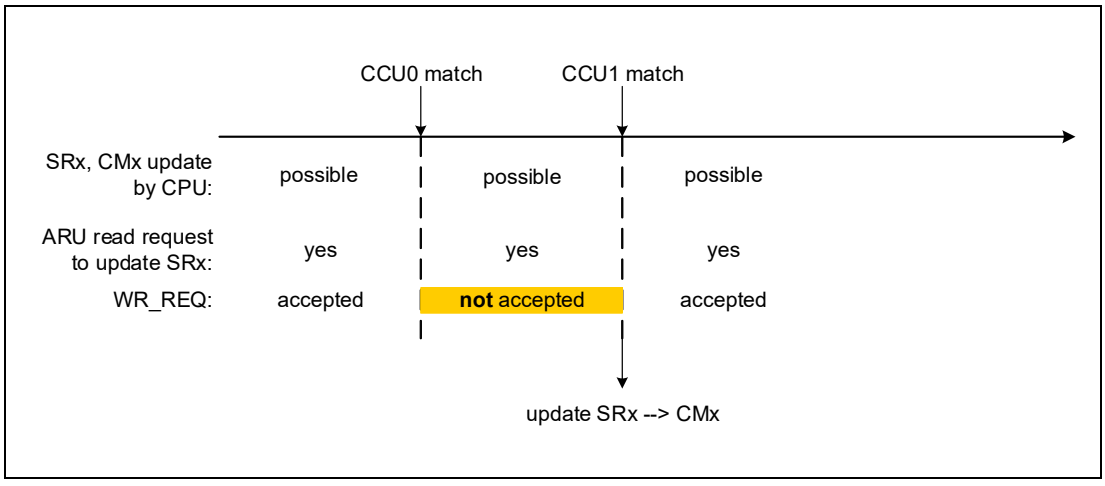


Figure 38.89 CPU access rights in case of compare strategy 'serve last' and EUPM = 0

The access rights by CPU to register CM0 and CM1 and the WR_REQ are depicted in the following figure for the case of EUPM = 1 (register ATOM[i]_CH[x]_CTRL)

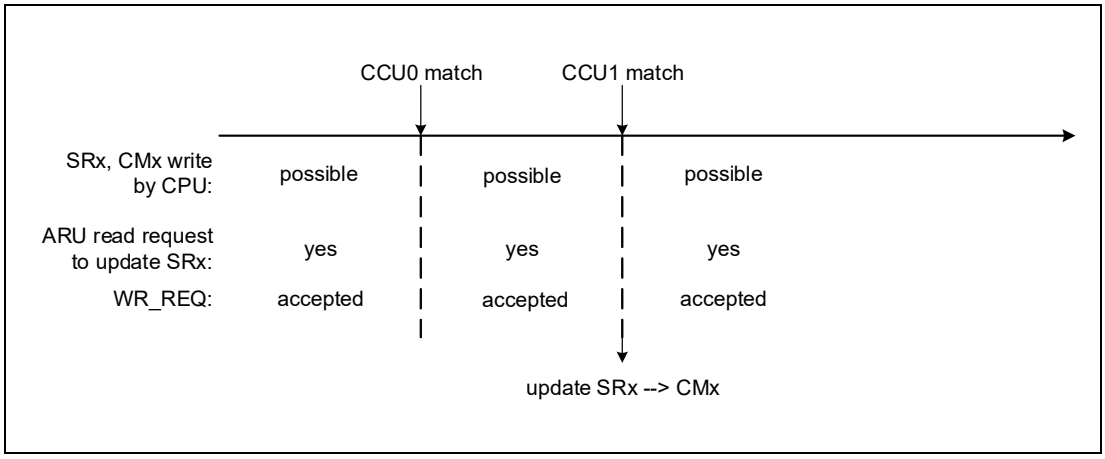


Figure 38.90 CPU access rights in case of compare strategy 'serve last' in case of EUPM = 1

(a) ARU Non-blocking mode

If bit ABM in register ATOM[i]_CH[x]_CTRL is not set, the ARU blocking mode is disabled. In this case the ATOM channel is continuously reading via ARU and storing new values in the SRx register and the ACB shadow register ACB_SR.
If ARU_EN is not set, the bit ABM has no meaning.

(b) ARU Blocking mode

If bit ABM in register ATOM[i]_CH[x]_CTRL is set, the ARU blocking mode is enabled. In this case the ATOM channel stops requesting new SRx values via ARU after reception of a new SRx value and restarts requesting a new value via ARU after compare match on both compare units (depending on compare strategy) followed by the immediate update of the CMx register with content of SRx register and an update of ACBI with the content of ACB_SR.

If ARU_EN is not set, the bit ABM has no meaning.

(c) Late Update by CPU

Although, the ATOM channel may be controlled by data received via the ARU, the CPU is able to request at any time a late update of the compare register. This can be initiated by setting the WR_REQ bit inside the ATOM[i]_CH[x]_CTRL register.

If none of the two compare match event happened, the ATOM channel accepts the setting of WR_REQ bit. In this case, the ATOM will request no further data from ARU (if ARU access was enabled) and will disable the update of CMx register with the content of SRx register on a compare match event.

If at least one of the requested compare match events happened (depending on strategy) the WR_REQ bit is not set and the WRF flag in register ATOM[i]_CH[x]_STAT is set to indicate that the late update was not successful.

The channel will in any case continue to compare against the values stored inside the compare registers (if bit DV was set). The CPU can now update the compare values by writing to the shadow registers and force the ATOM channel to update the compare registers by writing to the force update register bits in the AGC register.

With a force update the WR_REQ bit is reset automatically and the ARU read request is set up again (if ARU access was enabled).

38.17.4 ATOM Interrupt Signals

Signal	Description
CCU0TCx_IRQ	CCU0 Trigger condition interrupt for channel x
CCU1TCx_IRQ	CCU1 Trigger condition interrupt for channel x

38.17.5 ATOM Configuration Registers Overview

ATOM contains following configuration registers:

Table 38.209 Register list

Symbol	Register Name	Detail in Section
ATOM[i]_AGC_GLB_CTRL	ATOMi AGC global control register	**38.17.6.1**
ATOM[i]_AGC_ENDIS_CTRL	ATOMi AGC enable/disable control register	**38.17.6.2**
ATOM[i]_AGC_ENDIS_STAT	ATOMi AGC enable/disable status register	**38.17.6.3**
ATOM[i]_AGC_ACT_TB	ATOMi AGC action time base register	**38.17.6.4**
ATOM[i]_AGC_OUTEN_CTRL	ATOMi AGC output enable control register	**38.17.6.5**
ATOM[i]_AGC_OUTEN_STAT	ATOMi AGC output enable status register	**38.17.6.6**
ATOM[i]_AGC_FUPD_CTRL	ATOMi AGC force update control register	**38.17.6.7**
ATOM[i]_AGC_INT_TRIG	ATOMi AGC internal trigger control register	**38.17.6.8**
ATOM[i]_CH[x]_CTRL	ATOMi channel x control register	**38.17.6.9**
ATOM[i]_CH[x]_STAT	ATOMi channel x status register	**38.17.6.10**
ATOM[i]_CH[x]_RDADDR	ATOMi channel x ARU read address register	**38.17.6.11**
ATOM[i]_CH[x]_CN0	ATOMi channel x CCU0 counter register	**38.17.6.12**
ATOM[i]_CH[x]_CM0	ATOMi channel x CCU0 compare register	**38.17.6.13**
ATOM[i]_CH[x]_SR0	ATOMi channel x CCU0 compare shadow register	**38.17.6.14**
ATOM[i]_CH[x]_CM1	ATOMi channel x CCU1 compare register	**38.17.6.15**
ATOM[i]_CH[x]_SR1	ATOMi channel x CCU1 compare shadow register	**38.17.6.16**
ATOM[i]_CH[x]_IRQ_NOTIFY	ATOMi channel x interrupt notification register	**38.17.6.17**
ATOM[i]_CH[x]_IRQ_EN	ATOMi channel x interrupt enable register	**38.17.6.18**
ATOM[i]_CH[x]_IRQ_FORCINT	ATOMi channel x software interrupt generation	**38.17.6.19**
ATOM[i]_CH[x]_IRQ_MODE	ATOMi channel x interrupt mode configuration register	**38.17.6.20**

38.17.6 ATOM Registers Description

38.17.6.1 ATOM[i]_AGC_GLB_CTRL

Access: This register can be read or written in 32-bit units.

Address: $E8040_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UPEN_CTRL7		UPEN_CTRL6		UPEN_CTRL5		UPEN_CTRL4		UPEN_CTRL3		UPEN_CTRL2		UPEN_CTRL1		UPEN_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RST_CH7	RST_CH6	RST_CH5	RST_CH4	RST_CH3	RST_CH2	RST_CH1	RST_CH0	—	—	—	—	—	—	—	HOST_TRIG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 38.210 ATOM[i]_AGC_GLB_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	UPEN_CTRL [7:0]	ATOM channel 0 enable update of register CM0, CM1 and CLK_SRC from SR0, SR1 and CLK_SRC_SR. READ access: 00_B: Update disabled 01_B: — 10_B: — 11_B: Update enabled WRITE access: 00_B: Don't care, bits will not be change 01_B: Disable update 10_B: Enable update 11_B: Don't care, bits will not be changed
15 to 8	RST_CH[7:0]	Software reset of channel [7:0] 0: No action 1: Reset channel **NOTE** This bit is cleared automatically after write by CPU. The channel registers are set to their reset values and channel operation is stopped immediately. The output register of SOU unit is reset to inverse reset value of SL bit.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	HOST_TRIG	Trigger request signal (see AGC) to update the register ENDIS_STAT and OUTEN_STAT 0: No trigger request 1: Set trigger request **NOTE** This flag is reset automatically after triggering the update

38.17.6.2 ATOM[i]_AGC_ENDIS_CTRL

Access: This register can be read or written in 32-bit units.

Address: $E8044_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIS_CTRL7		ENDIS_CTRL6		ENDIS_CTRL5		ENDIS_CTRL4		ENDIS_CTRL3		ENDIS_CTRL2		ENDIS_CTRL1		ENDIS_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.211 ATOM[i]_AGC_ENDIS_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ENDIS_CTRL [7:0]	ATOM channel k enable/disable update value. 00_B: Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger 01_B: Disable channel on an update trigger 10_B: Enable channel on an update trigger 11_B: Don't change bits 1:0 of this register **NOTES** 1. If FREEZE = 0 and an ATOM channel is disabled, the counter CN0 is stopped and the output register of SOU unit is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. 2. If FREEZE = 1 and an ATOM channel is disabled, the counter CN0 is stopped (SOMP, SOMS mode) and each comparison is stopped (SOMC, SOMB mode). On an enable event, the counter CN0 starts counting from its current value or a comparison is restarted. 3. If the output is disabled (OUTEN[k] = 0), the ATOM channel k output ATOM_OUT[k] is the inverted value of bit SL.

38.17.6.3 ATOM[i]_AGC_ENDIS_STAT

Access: This register can be read or written in 32-bit units.

Address: $E8048_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDIS_STAT		ENDIS_STAT		ENDIS_STAT		ENDIS_STAT		ENDIS_STAT		ENDIS_STAT		ENDIS_STAT		ENDIS_STAT	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.212 ATOM[i]_AGC_ENDIS_STAT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	ENDIS_STAT [7:0]	ATOM channel k enable/disable 00_B: Don't care, bits 1:0 of register ENDIS_STAT will not be changed on an update trigger 01_B: Disable channel on an update trigger 10_B: Enable channel on an update trigger 11_B: Don't change bits 1:0 of this register **NOTES** 1. If FREEZE = 0 and an ATOM channel is disabled, the counter CN0 is stopped and the output register of SOU unit is set to the inverse value of control bit SL. On an enable event, the counter CN0 starts counting from its current value. 2. If FREEZE = 1 and an ATOM channel is disabled, the counter CN0 is stopped (SOMP, SOMS mode) and each comparison is stopped (SOMC, SOMB mode). On an enable event, the counter CN0 starts counting from its current value or a comparison is restarted.

38.17.6.4 ATOM[i]_AGC_ACT_TB

Access: This register can be read or written in 32-bit units.

Address: $E804C_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TBU_SEL		TB_TRIG	ACT_TB							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACT_TB															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.213 ATOM[i]_AGC_ACT_TB Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26, 25	TBU_SEL	Selection of time base used for comparison 00_B: TBU_TS0 selected 01_B: TBU_TS1 selected 10_B: TBU_TS2 selected 11_B: Same as 00 **NOTE** The bit combination 10 is only applicable if the TBU of the device contains three time base channels. Otherwise, this bit combination is also reserved. Refer to **Figure 38.2** to determine the number of channels for TBU of this device.
24	TB_TRIG	Set trigger request 0: No trigger request 1: Set trigger request **NOTE** This flag is reset automatically if the selected time base unit (TBU_TS0 or TBU_TS1 or TBU_TS2 if present) has reached the value ACT_TB and the update of the register were triggered.
23 to 0	ACT_TB	Specifies the signed compare value with selected signal TBU_TS[x], x = 0 to 2. If selected TBU_TS[x] value is in the interval [ACT_TB–$007FFFFF_H$,ACT_TB] the event is in the past and the trigger is generated immediately. Otherwise the event is in the future and the trigger is generated if selected TBU_TS[x] is equal to ACT_TB.

38.17.6.5 ATOM[i]_AGC_OUTEN_CTRL

Access: This register can be read or written in 32-bit units.

Address: $E804C_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUTEN_CTRL7		OUTEN_CTRL6		OUTEN_CTRL5		OUTEN_CTRL4		OUTEN_CTRL3		OUTEN_CTRL2		OUTEN_CTRL1		OUTEN_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.214 ATOM[i]_AGC_OUTEN_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	OUTEN_CTRL [7:0]	Output ATOM_OUT(0) enable/disable update value 00_B: Don't care, bits of register OUTEN_STAT will not be changed on an update trigger 01_B: Disable channel output on an update trigger 10_B: Enable channel output on an update trigger 11_B: Don't change bits of this register **NOTE** If the channel is disabled (ENDIS[0] = 0) or the output is disabled (OUTEN[0] = 0), the TOM channel 0 output ATOM_OUT[0] is the inverted value of bit SL.

38.17.6.6 ATOM[i]_AGC_OUTEN_STAT

Access: This register can be read or written in 32-bit units.

Address: $E8054_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OUTEN_STAT7		OUTEN_STAT6		OUTEN_STAT5		OUTEN_STAT4		OUTEN_STAT3		OUTEN_STAT2		OUTEN_STAT1		OUTEN_STAT0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.215 ATOM[i]_AGC_OUTEN_STAT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	OUTEN_STAT [7:0]	Control/status of output ATOM_OUT[7:0] READ access: 00_B: Output disabled 01_B: — 10_B: — 11_B: Output enabled WRITE access: 00_B: Don't care, bits will not be changed 01_B: Disable output 10_B: Enable output 11_B: Don't care, bits will not be changed

38.17.6.7 ATOM[i]_AGC_FUPD_CTRL

Access: This register can be read or written in 32-bit units.

Address: $E8058_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSTCN0_CH7		RSTCN0_CH6		RSTCN0_CH5		RSTCN0_CH4		RSTCN0_CH3		RSTCN0_CH2		RSTCN0_CH1		RSTCN0_CH0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FUPD_CTRL7		FUPD_CTRL6		FUPD_CTRL5		FUPD_CTRL4		FUPD_CTRL3		FUPD_CTRL2		FUPD_CTRL1		FUPD_CTRL0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.216 ATOM[i]_AGC_FUPD_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 16	RSTCN0_CH [7:0]	RSTCN0_CH[7:0]: Reset CN0 of channel k on force update event READ access: 00_B: CN0 is not reset on forced update 01_B: — 10_B: — 11_B: CN0 is reset on forced update WRITE access: 00_B: Don't care, bits will not be changed 01_B: Do not reset CN0 on forced update 10_B: Reset CN0 on forced update 11_B: Don't care, bits will not be changed If enabled, reset CN0 triggered by HOST_TRIG, ACT_TB compare match or internal trigger.
15 to 0	FUPD_CTRL [7:0]	Force update of ATOM channel [7:0] operation registers READ access: 00_B: Force update disabled 01_B: — 10_B: — 11_B: Force update enabled WRITE access: 00_B: Don't care, bits will not be changed 01_B: Disable force update 10_B: Enable force update 11_B: Don't care, bits will not be changed If enabled, force update of register CM0, CM1 and CLK_SRC triggered by HOST_TRIG, ACT_TB compare match or internal trigger. **NOTES** 1. In SOMP mode the force update request is stored and executed synchronized to the selected CMU_CLK. In all other modes the force update request is executed immediately. 2. In SOMP mode, in case of ECLK_SRC = 1 and CLK_SRC_SR = 011/100/101/110 a force update leads to an immediate update of CM0, CM1 and CLK_SRC.

38.17.6.8 ATOM[i]_AGC_INT_TRIG

Access: This register can be read or written in 32-bit units.

Address: $E805C_H + (800_H \times i)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INT_TRIG7		INT_TRIG6		INT_TRIG5		INT_TRIG4		INT_TRIG3		INT_TRIG2		INT_TRIG1		INT_TRIG0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.217 ATOM[i]_AGC_INT_TRIG Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	INT_TRIG[7:0]	Select input signal TRIG_[7:0] as a trigger source READ access: 00_B: Internal trigger from channel k (TRIG_k) not used 01_B: — 10_B: — 11_B: Internal trigger from channel k (TRIG_k) used WRITE access: 00_B: Don't care, bits will not be changed 01_B: Do not use internal trigger from channel k (TRIG_k) 10_B: Use internal trigger from channel k (TRIG_k) 11_B: Don't care, bits will not be changed

38.17.6.9 ATOM[i]_CH[x]_CTRL

Access: This register can be read or written in 32-bit units.

Address: $E8004_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FREEZE	SOMB	EXT_FUPD	—	ABM	OSM	SLA	TRIGOUT	EXTTRIGOUT	EXT_TRIG	OSM_TRIG	RST_CCU0	UDMODE		TRIG_PULSE	WR_REQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECLK_SRC	CLK_SRC_SR			SL	EUPM	CMP_CTRL	ACB					ARU_EN	TB12_SEL	MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (1/6)

Bit Position	Bit Name	Function
31	FREEZE	0: A channel disable/enable may change internal register and output register 1: A channel enable/disable does not change an internal or output register but stops counter CN0 (in SOMP mode), comparison (in SOMC/SOMB mode) and shifting (in SOMS mode) **NOTE** If channel is disabled and output is enabled, in SOMP mode with UDMODE! = 00_B the output is depending directly on SL bit, independent on FREEZE mode.
30	SOMB	SOMB mode 0: ATOM channel mode defined by bit field MODE 1: ATOM SOMB mode enabled **NOTE** This bit is only applicable in SOMB mode.
29	EXT_FUPD	External forced update 0: Use FUPD(x) signal from AGC to force update 1: Use TIM_EXT_CAPTURE signal to force update **NOTE** This bit is only applicable in SOMP and SOMS mode.
28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (2/6)

Bit Position	Bit Name	Function
27	ABM	ARU blocking mode 0: ARU blocking mode disabled: ATOM reads continuously from ARU and updates CM0,CM1 and ACB bits in case of SOMC mode or SR0, SR1 and ACB bits in case of SOMB mode independent of pending compare match event. 1: ARU blocking mode enabled: after update of CM0, CM1 and ACB bit in case of SOMC mode or SR0, SR1 and ACB bits in case of SOMB mode via ARU, no new data is read via ARU until compare match event occurred and in case of SOMC mode SR0 and/or SR1 are read. **NOTE** This bit is only applicable in SOMC and SOMB mode.
26	OSM	One-shot mode 0: Continuous PWM generation after channel enable 1: A single pulse is generated **NOTE** This bit is only applicable in SOMP and SOMS modes.
25	SLA	'Serve last' ARU communication strategy 0: Capture SRx time stamps after CCU0 match event not provided to ARU 1: Capture SRx time stamps after CCU0 match event provided to ARU **NOTES** 1. This bit is only applicable in SOMC mode. 2. The setting of this bit has only effect, when ACBI (4:2) is configured for 'serve last' compare strategy (100_B, 101_B, or 110_B). 3. When this bit is not set, the captured time stamps in the shadow registers SRx are only provided after the CCU1 match occurred. The ACBO(4:3) bits always return 10_B in that case. 4. By setting this bit, the ATOM channel also provides the captured time stamps after the CCU0 match event to the ARU. The ACBO(4:3) bits are set to 01_B in that case. After the CCU1 match event, the time stamps are captured again in the SRx registers and provided to the ARU. The ACBO(4:3) bits are set to 10_B. When the data in the shadow registers after the CCU0 match was not consumed by an ARU destination and the CCU1 match occurs, the data in the shadow registers is overwritten by the new captured time stamps. The ATOM channel does not request new data from the ARU when the CCU0 match values are read from an ARU destination.
24	TRIGOUT	Trigger output selection (output signal TRIG_CHx) of module ATOM_CHx. 0: TRIG_[x] is TRIG_[x–1] or TIM_EXT_CAPTURE(x). 1: TRIG_[x] is TRIG_CCU0 **NOTE** This bit is only applicable in SOMP, SOMC and SOMB mode.
23	EXTTRIGOUT	Select TIM_EXT_CAPTURE(x) as potential output signal TRIG_[x] 0: Signal TRIG_[x–1] is selected as output on TRIG_[x] (if TRIGOUT = 0) 1: Signal TIM_EXT_CAPTURE(x) is selected as output on TRIG_[x] (if TRIGOUT = 0) **NOTE** This bit is only applicable in SOMP, SOMC and SOMB mode.

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (3/6)

Bit Position	Bit Name	Function
22	EXT_TRIG	Select TIM_EXT_CAPTURE(x) as trigger signal 0: Signal TIM_[x–1] is selected as trigger to reset CN0 or to start single pulse generation. 1: Signal TIM_EXT_CAPTURE(x) is selected **NOTE** This bit is only applicable in SOMP mode.
21	OSM_TRIG	Enable trigger of one-shot pulse by trigger signal OSM_TRIG 0: Signal OSM_TRIG cannot trigger start of single pulse generation 1: Signal OSM_TRIG can trigger start of single pulse generation (only if bit OSM = 1) **NOTES** 1. This bit should only be set if bit OSM = 1 and bit RST_CCU0 = 0. 2. This bit is only applicable in SOMP mode.
20	RST_CCU0	Reset source of CCU0 0: Reset counter register CN0 to 0 on matching comparison with CM0 1: Reset counter register CN0 to 0 on trigger TRIG_[x–1] or TIM_EXT_CAPTURE(x) **NOTES** 1. If RST_CCU0 = 1 and UPEN_CTRLx=1 are set, TRIG_[x–1] or TIM_EXT_CAPTURE(x) triggers also the update of work register (CM0, CM1 and CLK_SRC). 2. This bit is only applicable in SOMP mode. 3. This bit should only be set if bit OSM = 0 (i.e. in continuous mode)
19, 18	UDMODE	Up/down counter mode 00: Up/down counter mode disabled: CN0 counts always up 01: Up/down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches 0 (i.e. changes from down to up) 10: Up/down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches CM0 (i.e. changes from up to down) 11: Up/down counter mode enabled: CN0 counts up and down, CM0, CM1 are updated if CN0 reaches 0 or CM0 (i.e. changes direction) **NOTE** This mode is only applicable in SOMP mode.
17	TRIG_PULSE	Trigger output pulse length of one SYS_CLK period 0: Output on ATOM[i]_OUT[x]_T is '1' as long as CN0_SR0 (if SR0_TRIG = 1) 1: Output on ATOM[i]_OUT[x]_T is '1' for only one SYS_CLK period if CN0 = SR0 (if SR0_TRIG = 1) **NOTE** This bit is only applicable in SOMP mode.

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (4/6)

Bit Position	Bit Name	Function
16	WR_REQ	CPU Write request bit for late compare register update. 0: No late update requested by CPU 1: Late update requested by CPU **NOTES** 1. The CPU can disable subsequent ARU read requests by the channel and can update the shadow registers with new compare values, while the compare units operate on old compare values received by former ARU accesses, if occurred. 2. On a compare match event, the WR_REQ bit will be reset by hardware. 3. At the point of the force update only the shadow registers SR0 and SR1 are transferred into the CM0, CM1 registers. The output action is still defined by the ACBI bit field described by the ARU together with the old compare values for CM0/CM1. 4. This bit is only applicable in SOMC and SOMB mode.
15	ECLK_SRC	Extend CLK_SRC 0: CLK_SRC_SR set 1 selected 1: CLK_SRC_SR set 2 selected **NOTES** 1. See bit CLK_SRC_SR description for details. 2. This bit is only applicable in SOMP and SOMS mode.
14 to 12	CLK_SRC_SR	CMU clock source (SOMS)/shadow register for CMU clock source (SOMP). If ECLK_SRC = 0 000: CMU_CLK0 selected 001: CMU_CLK1 selected 010: CMU_CLK2 selected 011: CMU_CLK3 selected 100: CMU_CLK4 selected 101: CMU_CLK5 selected 110: CMU_CLK6 selected 111: CMU_CLK7 selected If ECLK_SRC = 1 000: CMU_CLK0 selected 001: CMU_CLK1 selected 010: CMU_CLK2 selected 011: Reserved 100: Clock stopped 101: TRIG[x–1] selected 110: TIM_EXT_CAPTURE[x] selected 111: CMU_CLK7 selected **NOTES** 1. This register is a shadow register for the register CLK_SRC. Thus, if the CMU_CLK source for PWM generation should be changed during operation, the old CMU_CLK has to operate until the update of the ATOM channels internal CLK_SRC register by the CLK_SRC_SR content is done either by an end of a period or a forced update. 2. After (channel) reset the selected CLK_SRC value is the SYS_CLK (input of Global Clock Divider). To use in SOMP mode one of the CMU_CLKx, it is recommended to perform a forced update of CLK_SRC with the value of CLK_SRC_SR value before/with enabling the channel. 3. In case of ECLK_SRC=1 and CLK_SRC_SR =011_B/100_B/101_B/110_B a force update leads to an immediate update of CM0, CM1 and CLK_SRC. 4. This bitfield is only applicable in SOMP, SOMS and SOMI mode.

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (5/6)

Bit Position	Bit Name	Function
11	SL	Initial signal level. 0: Low signal level 1: High signal level **NOTES** 1. Reset value depends on the hardware configuration chosen by silicon vendor. 2. If the output is disabled, the output ATOM_OUT[x] is set to inverse SL independent of the ATOM channel mode. 3. In SOMS mode: the inverse value of SL is shifted into the CM1 register. 4. If FREEZE = 0 and the channel is disabled, in SOMP, SOMI, SOMS the internal register SOUR inside ATOM sub unit SOU is set to the inverse value of SL. By enabling the channel the register SOUR is not changed. Thus, if the output is enabled afterwards, the output ATOM_OUT[x] is the inverse value of SL. 5. If FREEZE = 0 and the channel is disabled, in SOMC mode the internal register SOUR inside ATOM sub unit SOU is set to the value of SL. By enabling the channel the register SOUR is not changed. Thus, if the output is enabled and the channel is disabled, the output ATOM_OUT[x] is the value of SL. 6. If FREEZE = 1 and the channel is disabled, the output register of SOU unit is not changed and output ATOM_OUT[x] is not changed.
10	EUPM	Extended update mode 0: No extended update of CM0 and CM1 via CPU or ARU 1: Extended update mode in case of compare strategy 'serve last': update of CM1 after CCU0 compare match possible via ARU or CPU. **NOTES** 1. If EUPM = 1 a write access to CM0 or CM1 never causes an AEI write status 10_B. 2. This bit is only applicable in SOMC and SOMB mode.
9	CMP_CTRL	CMP_CTRL: CCUx compare strategy select. 0: Greater-equal compare against TBU time base values (TBU_TSx ≥ CMx) 1: Less-equal compare against TBU time base values (TBU_TSx ≤ CMx) **NOTES** 1. The compare unit CCU0 or CCU1 that compares against TBU_TS0 (depending on CCUx control mode defined by ACBI(4:2) or ACB42) always performs a greater-equal comparison, independent on CMP_CTRL bit. 2. This bit is only applicable in SOMC and SOMB mode.
8 to 4	ACB	ATOM Mode control bits. **NOTE** These bits have different meaning in the different ATOM channel modes. Refer to **Table 38.197** for further details.
3	ARU_EN	ARU_EN: ARU Input stream enable. 0: ARU Input stream disabled 1: ARU Input stream enabled

Table 38.218 ATOM[i]_CH[x]_CTRL Register Contents (6/6)

Bit Position	Bit Name	Function
2	TB12_SEL	Select time base value TBU_TS1 or TBU_TS2. 0: TBU_TS1 selected for comparison 1: TBU_TS2 selected for comparison • SOMI: Not used • SOMC: Used • SOMB: Used • SOMP: Not used • SOMS: Not used **NOTE** This bit is only applicable if three time bases are present in the GTM-IP. Otherwise, this bit is reserved.
1, 0	MODE	ATOM channel mode select. 00: ATOM Signal Output Mode Immediate (SOMI) 01: ATOM Signal Output Mode Compare (SOMC) 10: ATOM Signal Output Mode PWM (SOMP) 11: ATOM Signal Output Mode Serial (SOMS) **NOTE** Not used in SOMB mode.

38.17.6.10 ATOM[i]_CH[x]_STAT

Access: This register can be read or written in 32-bit units.

Address: $E801C_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 000x_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ACBO					DR	WRF	DV	ACBI				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.219 ATOM[i]_CH[x]_STAT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	ACBO	ATOM Internal status bits. ACBO[3] = 1: CCU0 Compare match occurred ACBO[4] = 1: CCU1 Compare match occurred **NOTES** 1. These bits are only set in SOMC mode. 2. ACBO is reset to 00000_B on an update of register CM0 or CM1 (via ARU or CPU) 3. In SOMC mode these bits are sent as ARU control bits 52 to 48.
23	DR	ARU data rejected flag 0: Received ARU data stored 1: Received ARU data rejected **NOTE** The flag is cleared if valid data is received and stored via ARU.
22	WRF	Write request of CPU failed for late update. 0: Late update was successful, CCUx units wait for comparison 1: Late update failed **NOTES** 1. The bit WRF can be reset by writing a 1 to it. 2. This bit is only applicable in SOMC and SOMB mode.

Table 38.219 ATOM[i]_CH[x]_STAT Register Contents (2/2)

Bit Position	Bit Name	Function
21	DV	Valid ARU Data stored in compare registers. 0: No valid data stored in register CM0 and/or CM1, no comparison is activated 1: Valid data stored in CM0 and/or CM1, comparison activated **NOTE** This bit is only applicable in SOMC and SOMB mode. The CPU can determine the status of the ARU transfers with this bit. After the compare event occurred, the bit is reset by hardware.
20 to 16	ACBI	ATOM Mode control bits. **NOTES** 1. For ATOM SOMI, SOMC, SOMP and SOMS mode this register serves as a mirror for the five ARU control bits received through the ARU interface. The bits are valid, when the DV bit is set. 2. For SOMB mode this bit field serves as the work register of the compare strategy. It can be updated with the value of bit field ACB of register ATOM[i]_CH[x]_CTRL or the value of internal shadow register ACB_SR.
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OL	Actual output signal level of ATOM_CHx_OUT. 0: Output signal level is low 1: Output signal level is high **NOTE** Reset value is the inverted value of bit SL which depends on the hardware configuration chosen by silicon vendor.

38.17.6.11 ATOM[i]_CH[x]_RDADDR

Access: This register can be read or written in 32-bit units.

Address: $E8000_H + (800_H \times i) + (80_H \times x)$

Value after reset: $01FE\ 01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RDADDR1								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RDADDR0								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.220 ATOM[i]_CH[x]_RDADDR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 16	RDADDR1	ARU Read address 1. **NOTES** 1. This read address is only applicable in SOMC mode. 2. The ATOM channel switches to this read address, when requested in the ARU control bits 52 to 48 with the pattern “111——”. The channel switches back to the RDADDR0 after one ARU data package was received on RDADDR1 and the compare match event is occurred. 3. This bit field is only writable if channel is disabled.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	RDADDR0	ARU Read address 0. **NOTES** 1. This read address is used by the ATOM channel to receive data from the ARU immediately after the channel and ARU access is enabled (see ATOM[i]_CH[x]_CTRL register for details). 2. This bit field is only writable if channel is disabled.

38.17.6.12 ATOM[i]_CH[x]_CN0

Access: This register can be read or written in 32-bit units.

Address: $E8018_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CN0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CN0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.221 ATOM[i]_CH[x]_CN0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CN0	ATOM CCU0 counter register.

38.17.6.13 ATOM[i]_CH[x]_CM0

Access: This register can be read or written in 32-bit units.

Address: $E8010_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CM0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CM0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.222 ATOM[i]_CH[x]_CM0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CM0	ATOM CCU0 compare register.

38.17.6.14 ATOM[i]_CH[x]_SR0

Access: This register can be read or written in 32-bit units.

Address: $E8008_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SR0							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SR0															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.223 ATOM[i]_CH[x]_SR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	SR0	ATOM channel x shadow register SR0. **NOTE** The SR0 register is used as shadow register for CM0 in SOMP and SOMS modes and is used as capture register for time base TBU_TS0 in SOMC mode.

38.17.6.15 ATOM[i]_CH[x]_CM1

Access: This register can be read or written in 32-bit units.

Address: $E8014_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CM1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CM1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.224 ATOM[i]_CH[x]_CM1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CM1	ATOM CCU1 compare register.

38.17.6.16 ATOM[i]_CH[x]_SR1

Access: This register can be read or written in 32-bit units.

Address: $E800C_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SR1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SR1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.225 ATOM[i]_CH[x]_CM1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	SR1	ATOM channel x shadow register SR1. **NOTE** The SR1 register is used as shadow register for CM1 in SOMP and SOMS modes and is used as capture register for time base TBU_TS1 or TBU_TS2 (when selected in ATOM[i]_CH[x]_CTRL register) in SOMC mode.

38.17.6.17 ATOM[i]_CH[x]_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: $E8020_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCU1TC	CCU0TC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.226 ATOM[i]_CH[x]_IRQ_NOTIFY Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CCU[1:0]TC	CCU[k] Trigger condition interrupt for channel x. 0: No interrupt occurred. 1: CCU[1:0] Trigger condition interrupt was raised by ATOM channel x. **NOTES** 1. This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged. 2. If bit SR0_TRIG is set to 1 (only valid in SOMP mode), this interrupt notify flag is set in case of SR0 is equal to CN0 and not set in case of CM1 ≥/≤ CN0.

38.17.6.18 ATOM[i]_CH[x]_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: $E8024_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CCU1TC_IRQ_EN	CCU0TC_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.227 ATOM[i]_CH[x]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CCU[1:0]TC_IRQ_EN	ATOM_CCU[1:0]TC_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.

38.17.6.19 ATOM[i]_CH[x]_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: $E8028_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG_CCU1TC	TRG_CCU0TC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.228 ATOM[i]_CH[x]_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TRG_CCU[1:0]TC	Trigger ATOM_CCU[k]TC_IRQ interrupt by software. 0: No interrupt triggering. 1: Assert corresponding field in ATOM[i]_CH[x]_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL.

38.17.6.20 ATOM[i]_CH[x]_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: $E802C_H + (800_H \times i) + (80_H \times x)$

Value after reset: $0000\ 000x_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.229 ATOM[i]_CH[x]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5**.

38.18 Dead Time Module (DTM)

38.18.1 Overview

The following figure gives an overview of the structure of the Dead Time Module (DTM).

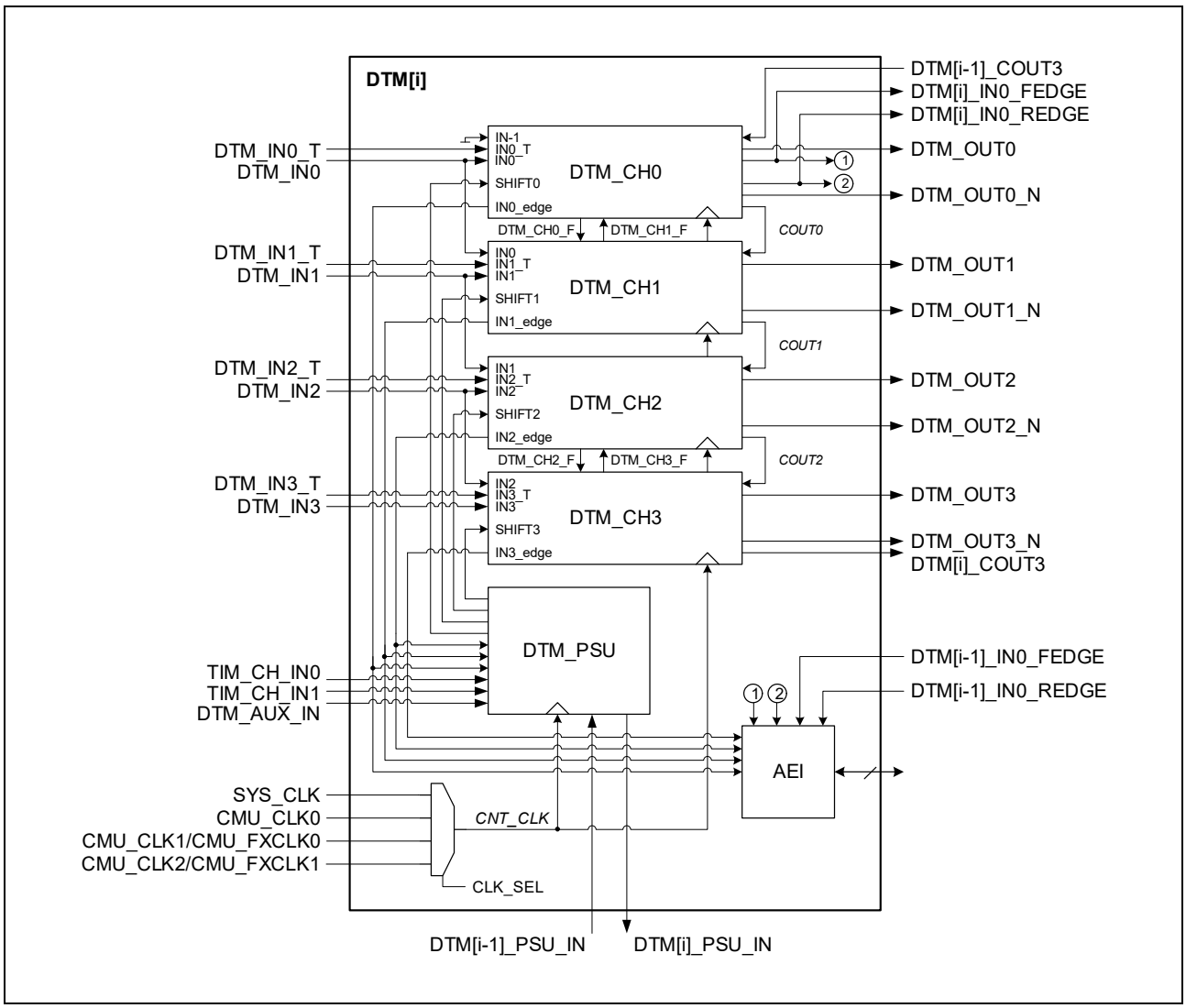


Figure 38.91 DTM block diagram

The main function of the DTM is to derive for each input DTM_IN0 to DTM_IN3 the individual inverse signal (DTM[i]_OUT[x]_N) and to apply an edge specific delay between the edge of the original signal and the edge of the derived inverted signal (i.e., the dead time). This function is mainly used for controlling of half bridges.

A second function provided by DTM is to set the outputs of one channel to the value of the preceding channel if requested by a trigger on input TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN. This feature allows a phase shift on one PWM signal to the phase of the preceding PWM signal up to the next edge on this channel.

The third function provided by DTM is to (N)AND/(N)OR/X(N)OR combine the input DTM_IN[x] signal of one DTM channel with the signal on input TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN (selected inside DTM_PSU and assigned to one of the signals SHIFT[x]) or with the combinational output (signal COUT[x]) of preceding channel. As a result COUT2 may be the combined signal of

DTM_IN0 and TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN and the signal DTM_IN1. For COUT3 this chain can be combined again with signal DTM_IN3.

The outputs of each channel may be swapped individually to provide the function of combining signals on each output of a channel.

In general, the DTM instances are placed behind the TOM and the ATOM instances, i.e., the outputs TOM_OUT[x] and TOM_OUT[x]_T or ATOM_OUT[x] and ATOM_OUT[x]_T are each routed to the DTM instance inputs DTM_IN[y] and DTM_IN[y]_T. Four DTM instances behind a TOM instance i and two DTM instances behind an ATOM instance i are grouped together in a Cluster DTM hierarchy called CDTM[i]. The connections between DTM and the modules TOM and ATOM are depicted in **Figure 38.92, Connections of TIM, TOM and ATOM to DTM**.

Depending on device configuration, not every DTM instance is available. E.g. a device may only have one DTM connected to the first four channels of ATOM. In this case, the other four channels (4 to 7) are connected directly to GTM outputs. For detailed information, which DTM instance is available, refer to corresponding device specific **Section 38.28, GTM Device 358** of this specification.

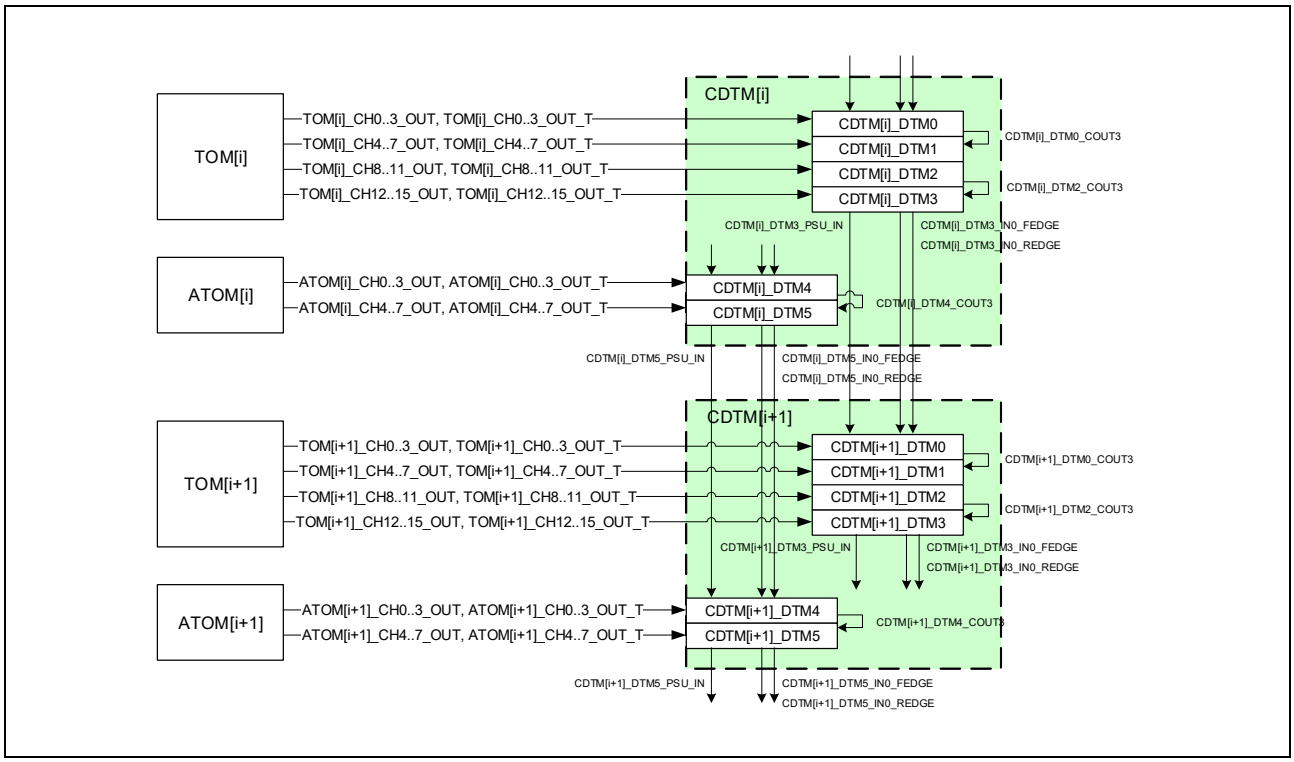


Figure 38.92 Connections of TIM, TOM and ATOM to DTM

Additionally, the DTM instances have inputs TIM_CH_IN0 / TIM_CH_IN1 which are driven by TIM output signals TIM[i]_CH[x]_F_OUT. There are two configurations of TIM to DTM connections possible depending on the DTM channel specific configuration bit TIM_SEL. In case of TIM_SEL = 0 the connected TIM input may not be of the same cluster as the DTM. In case of TIM_SEL = 1 the TIM input is of the same cluster as the DTM.

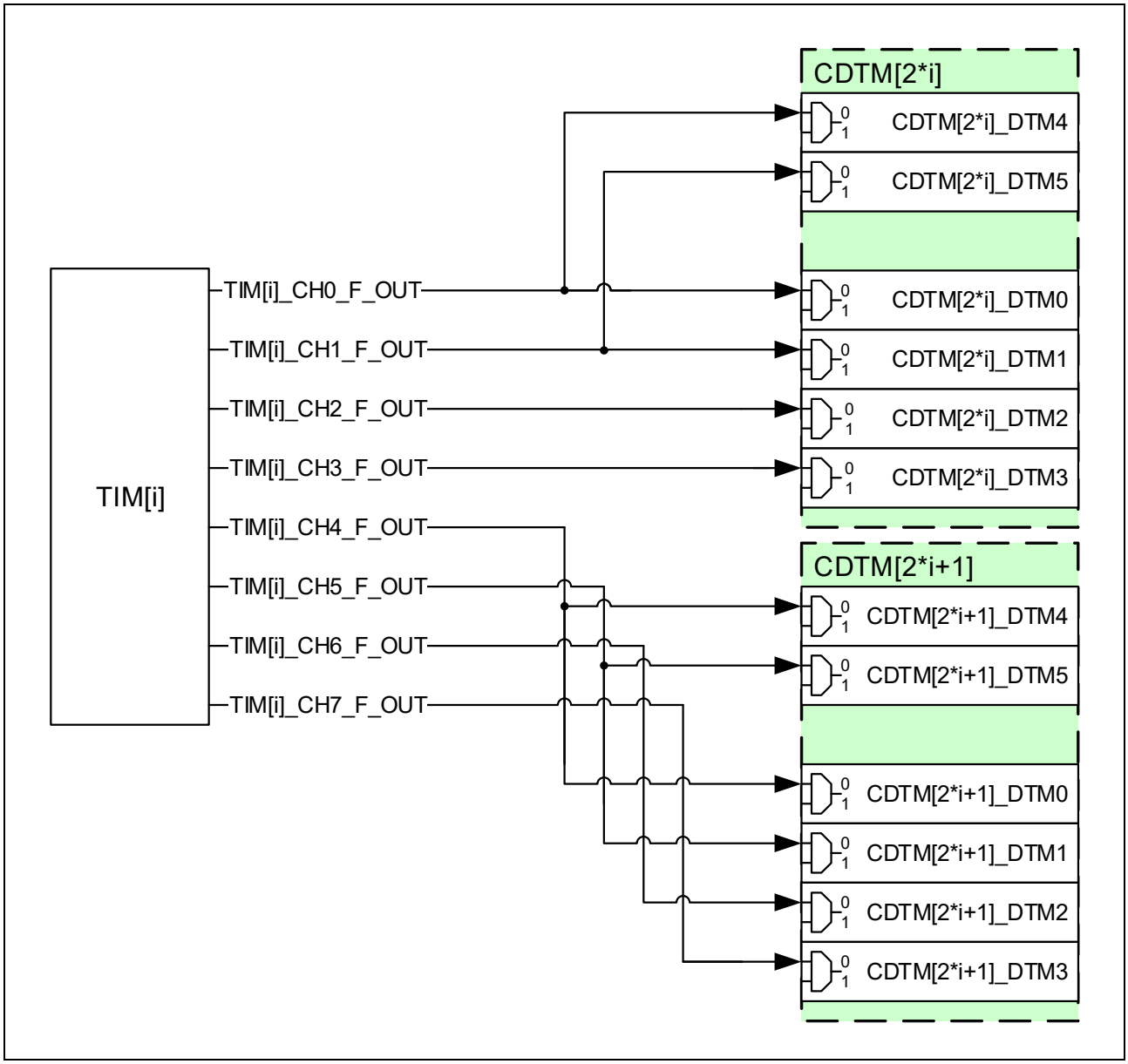


Figure 38.93 Connections of TIM to DTM inputs TIM_CH_IN0/TIM_CH_IN1 for TIM_SEL = 0

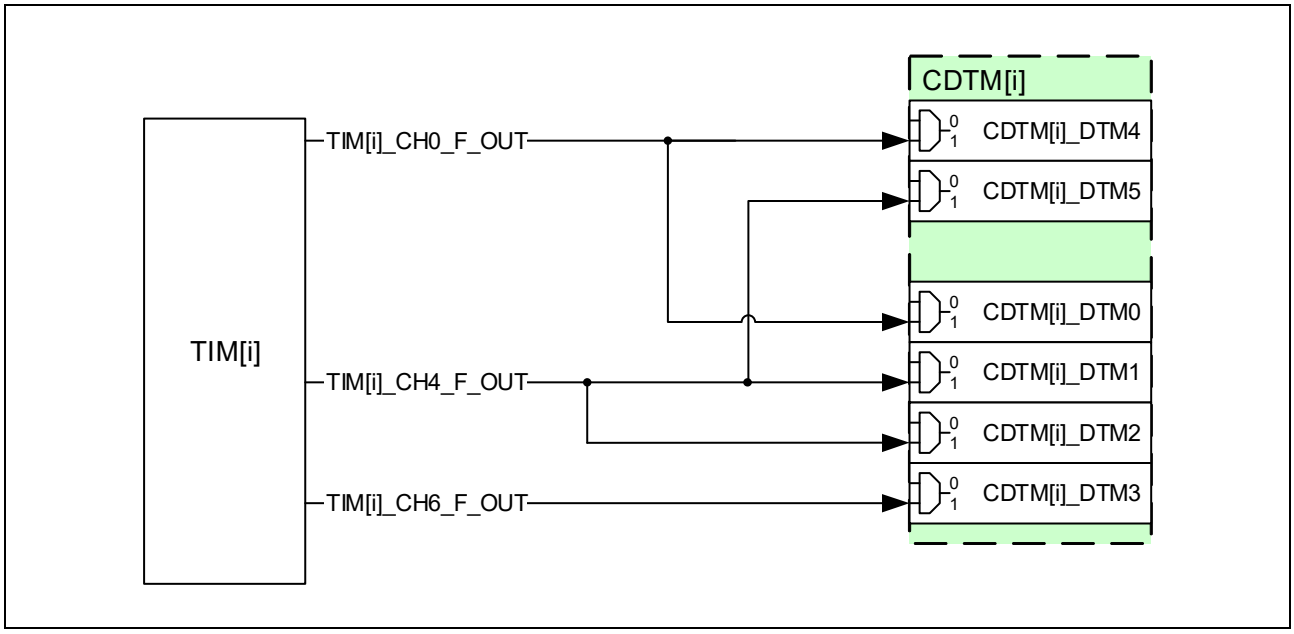


Figure 38.94 Connections of TIM to DTM inputs TIM_CH_IN0/TIM_CH_IN1 for TIM_SEL = 1

There are also connections between DTM instances of same CDTM instance. For each pair of DTM instance 2i and 2i+1 the combinatorial output COUT(3) of DTM[2i] channel 3 is connected to DTM[2i+1] channel 0 COUT(0–1). With this a combinatorial chain over two neighbored DTM instances can be configured. If one of the two neighbored DTM instances is not available (i.e. it is an empty instance) the inputs used for connections between two neighbored DTM instances are left open.

NOTE

For channel x = 0 of DTM instance 2i input signals COUT[x–1] is unused and I1SEL_[x] is defined as 0.

An additional link between DTM instances behind an ATOM is a forwarding of DTM[i]_PSU_IN signal to next available instance of DTM behind an ATOM (e.g. DTM[i+1]_PSU_IN). The same link is available between all available DTM behind a TOM.

NOTE:

For unavailable DTM[i] instances (i.e. the instance DTM[i] is called empty) the signal DTM[i–1]_PSU_IN is passed through empty instance DTM[i] to DTM[i+1]_PSU_IN, DTM[i–1]_IN0_FEDGE and DTM[i–1]_IN0_REDGE are passed through DTM[i] to DTM[i+1]_IN0_FEDGE and DTM[i+1]_IN0_REDGE.

Further connections between neighbored DTM instances are depicted in the following figure:

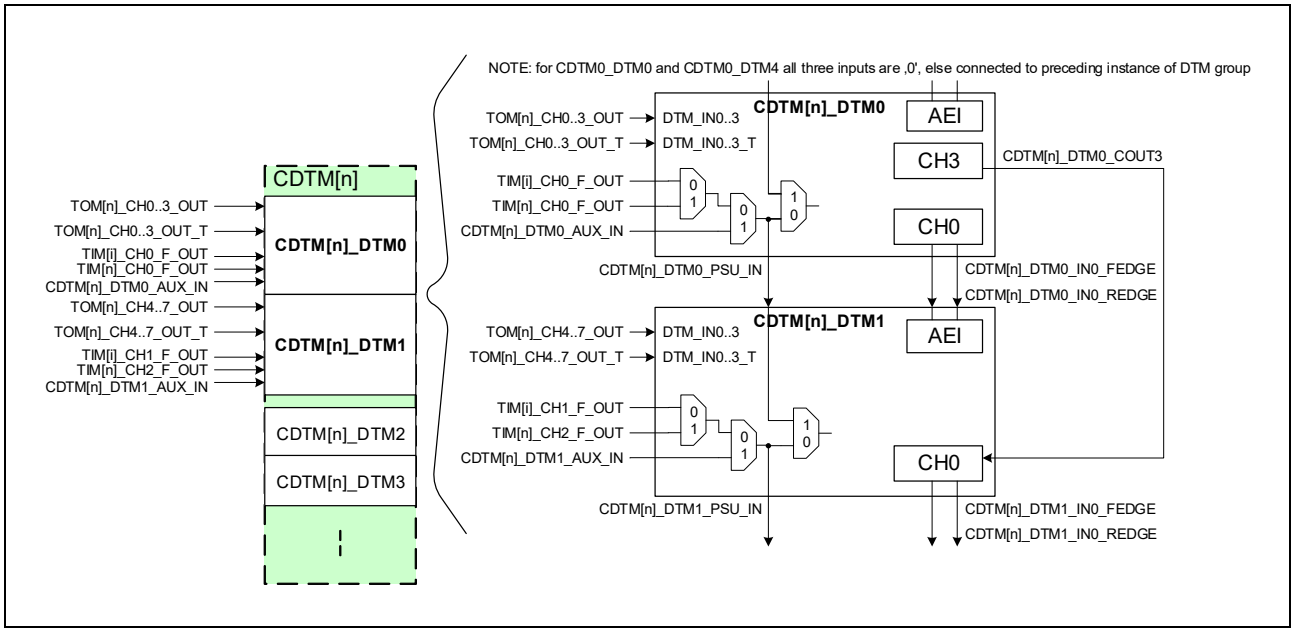


Figure 38.95 Connections between DTM instances

38.18.2 DTM Channel

The following figure depicts the functions of a DTM channel.

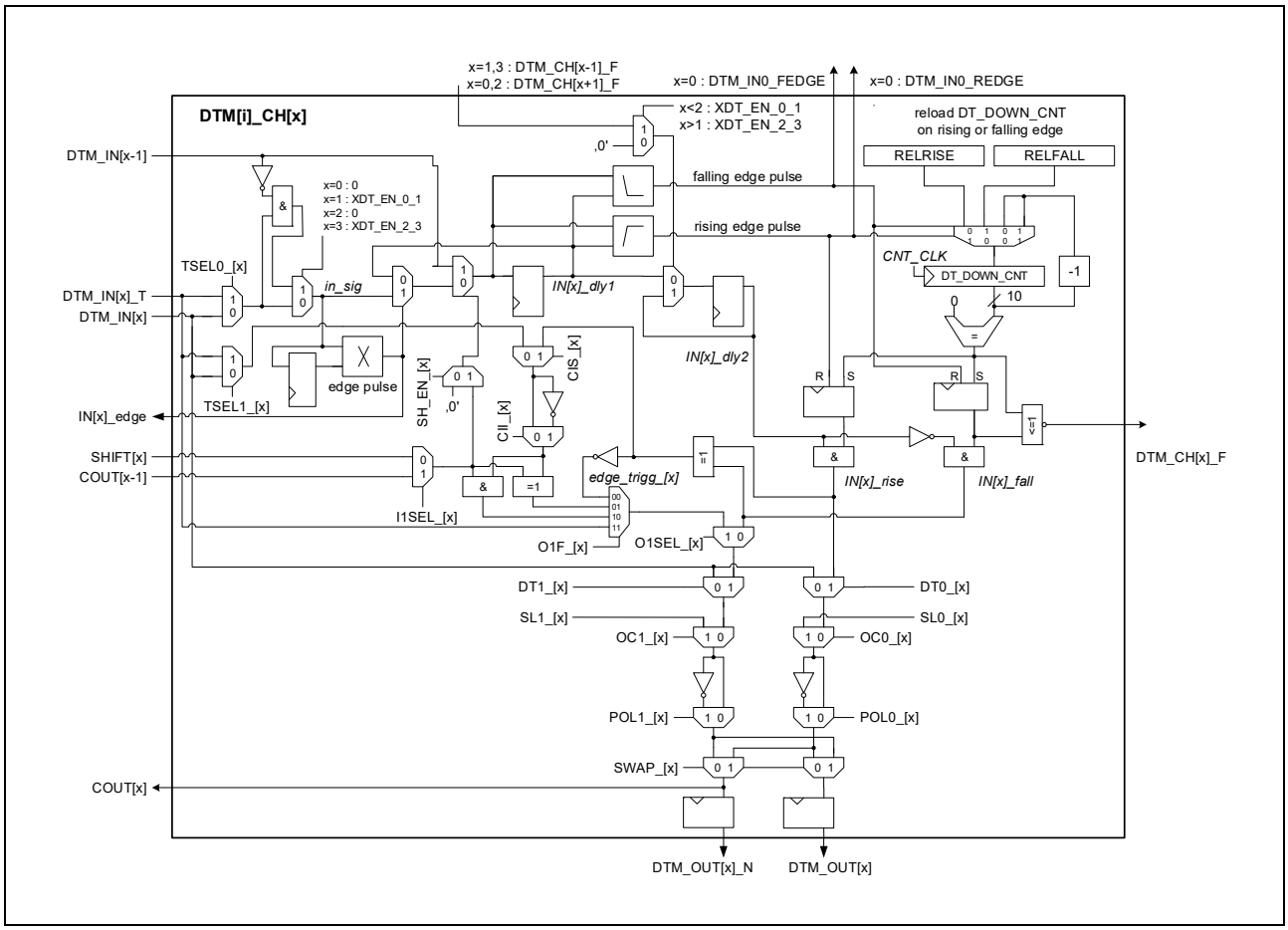


Figure 38.96 DTM channel overview

The main feature of each channel is to derive the inverse signal out of the input signal DTM_IN[x], apply an edge dependent delay on the two resulting signal paths and provide these signals at the outputs DTM[i]_OUT[x] and DTM[i]_OUT[x]_N.

There are two possibilities to apply dead time on GTM output signals. One is to use one DTM channel per TOM/ATOM channel and generate inside the DTM the second inverse signal. This is called the standard dead time generation. The second way is to generate two signals out of two TOM/ATOM channel and to apply inside the DTM only the dead time by using two cross linked DTM channel. This is called the cross dead time generation.

38.18.2.1 Standard dead time generation

The dead time can be configured for each edge individually. The bit field RELRISE in register DTM[i]_CH[x]_DTV contains the reload value for the counter and defines the delay for rising edges in multiples of selected clock ticks. The bit field RELFALL in register DTM[i]_CH[x]_DTV contains the reload value for the counter and defines the delay for falling edges in multiples of selected clock ticks.

The counter is reloaded with the value of RELRISE on a rising edge and reloaded with the value of RELFALL on a falling edge on input DTM_IN[x] (or DTM_IN[x–1] in case of shift enable SH_EN_[x]). On a reload of the counter the flip-flop following the counter output comparator is reset and stays reset until the counter has reached 0. After reload, the counter DT_DOWN_CNT counts down until it reaches 0 and stops at 0.

The signal flow for function of standard dead time signal generation is depicted in following figure

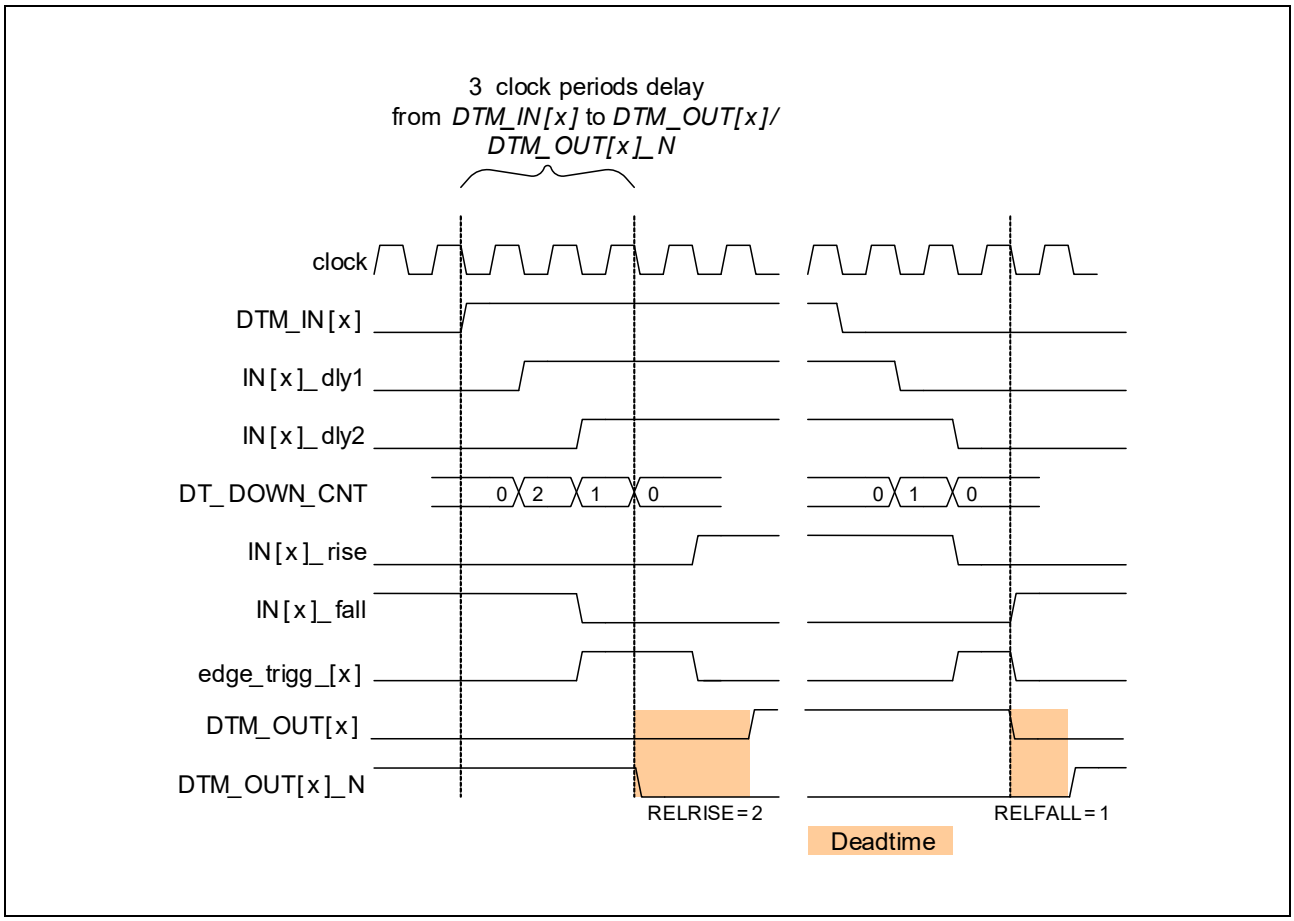


Figure 38.97 Wave signals for function of dead time generation

NOTES

1. The delay from the input signal DTM_IN[x] to the output signals DTM[i]_OUT[x] and DTM[i]_OUT[x]_N is three system clock periods by disabled feed through (see DT0/1_[x] in DTM[i]_CH_CTRL2).
2. The delay from the input signal DTM_IN[x] to the output signals DTM[i]_OUT[x] and DTM[i]_OUT[x]_N is one system clock periods by enabled feed through (see DT0/1_[x] in DTM[i]_CH_CTRL2).
3. The delay from the input signal DTM_IN[x]_T to the output signals DTM[i]_OUT[x] and DTM[i]_OUT[x]_N is three system clock periods in case of disabled feed through (see O1F_[x] and O1SEL_[x] in DTM[i]_CH_CTRL2).
4. The delay from the input signal DTM_IN[x]_T to the output signals DTM[i]_OUT[x] and DTM[i]_OUT[x]_N is one system clock periods in case of enabled feed through (see O1F_[x] and O1SEL_[x] in DTM[i]_CH_CTRL2).
5. The reset level of the output signals DTM[i]_OUT[x] connected from ATOM module depends on the hardware configuration value atom_out_reset_level_c chosen by silicon vendor.
6. The reset level of the output signals DTM[i]_OUT[x]_N connected from ATOM module is defined by the inverse hardware configuration value atom_out_reset_level_c chosen by silicon vendor.
7. The reset level of the output signals DTM[i]_OUT[x] connected from TOM module is defined by the hardware configuration value tom_out_reset_level_c chosen by silicon vendor.
8. The reset level of the output signals DTM[i]_OUT[x]_N connected from TOM module is defined by the inverse hardware configuration value tom_out_reset_level_c chosen by silicon vendor.

38.18.2.2 Cross channel dead time

A second way to apply a dead time value on two output signals is the cross channel dead time. In opposite to the dead time described in **Section 38.18.2.1** the cross channel dead time mode does not generate out of one signal the corresponding inverse signal but tries to apply the dead time on the input signals of two neighbored DTM channel. To do this, two neighbored DTM input signals (on DTM channel (2k) and (2k+1) for k = 0, 1) are cross linked together in the way that a falling edge on one channel leads to a hold phase of current signal value on the cross linked channel.

This behavior is reached by the following: A falling edge on e.g. channel (2k) reloads the DT_DOWN_CNT with the value of RELFALL. While this counter is counting down, the output signal of the cross linked channel (2k+1) keeps its value. If the counter DT_DOWN_CNT has reached 0 again, the channel (2k+1) output is released and can follow the value on its input. The timing of the cross channel dead time is depicted in **Figure 38.98**.

The following **Figure 38.98** shows the behavior in case of input edges at DTM_IN[2k] and DTM_IN[2k+1] occur at the same point in time. Then the falling edge is forwarded immediately (with only two clock cycles delay) and the rising edge is delayed additionally by the number of clock ticks specified by the RELFALL parameter of the cross linked channel.

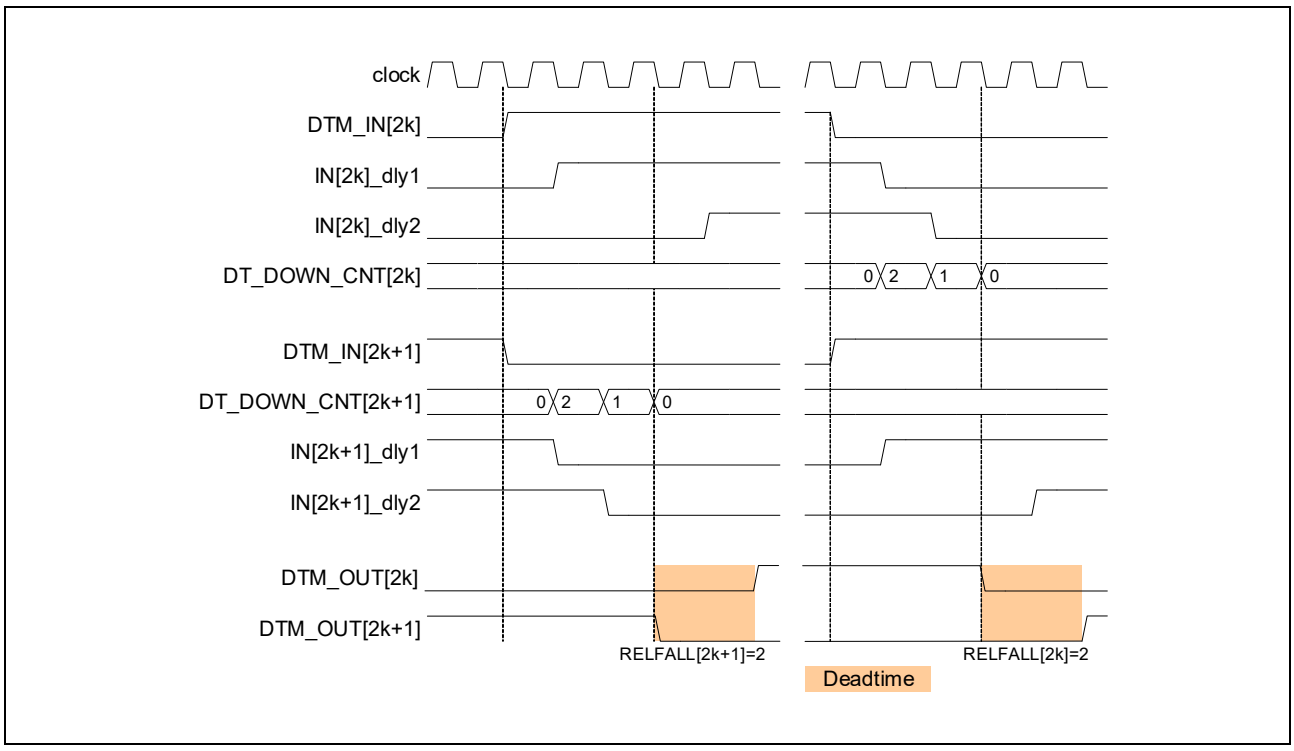


Figure 38.98 Cross channel dead time timing diagram

In case of high level (i.e. 1) at the DTM inputs DTM_IN[2k] and DTM_IN[2k+1] at the same point in time, the channel of (2k) has higher priority than the corresponding channel (2k+1). This means that in this case the input DTM_IN[2k+1] is forced immediately at channel input to low level (i.e. 0). As a result the DTM output of channel DTM_OUT[2k+1] can never be high if the cross linked channel DTM_OUT[2k] is high.

38.18.3 Phase Shift Control Unit

The phase shift unit (DTM_PSU) is depicted in the following figure. It supports the second major function of the DTM module to allow phase shifting of PWM signal on one of the channels.

This feature allows a phase shift on one PWM signal to the phase of the preceding PWM signal up to the next edge on this channel.

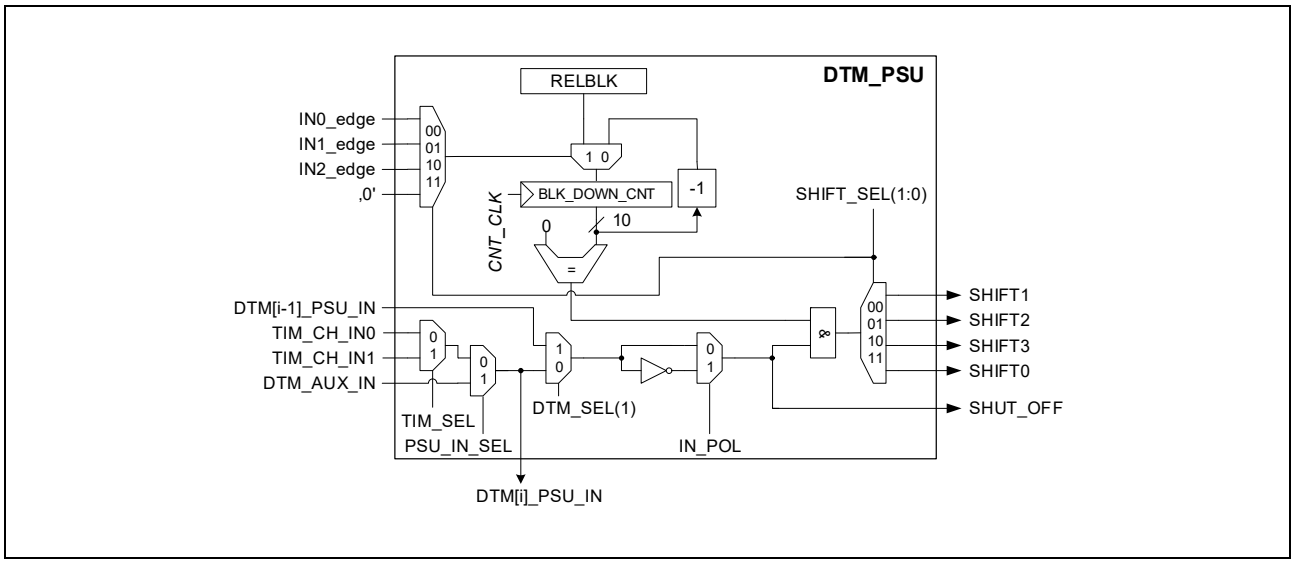


Figure 38.99 Phase Shift Unit overview

This sub-module provides an additional counter BLK_DOWN_CNT and reload register RELBLK (bit field of register DTM[i]_PS_CTRL). The counter is reloaded on an edge detected on one of the selected signals IN0_edge to IN2_edge (selected by bit field SHIFT_SEL in register DTM[i]_PS_CTRL). Then, the counter counts down until it reaches 0. While the counter is counting down, it blocks the trigger (i.e. the selected one of the signals SHIFT[x]) of one of the channels by one of the input signals TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN.

If the counter BLK_DOWN_CNT is not counting, a pulse on the input TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN is forwarded to one of the selected DTM_PSU outputs SHIFT[x]. This signal triggers in the selected channel (if SH_EN_x = 1) the update of the first flip-flop on channel x (i.e. representing IN[x]_DLY) to the input value DTM_IN[x–1] of the preceding channel. If this update leads to an edge, the succeeding part of DTM channel derives the inverse signal and applies the corresponding dead time (i.e. the edge delay) to the output signals of the channel.

NOTE

For channel x = 0 input signals DTM_IN[x–1] is unused and SH_EN_[x] is defined as 0.

The following figure shows an example of phase shifting on channel 1.

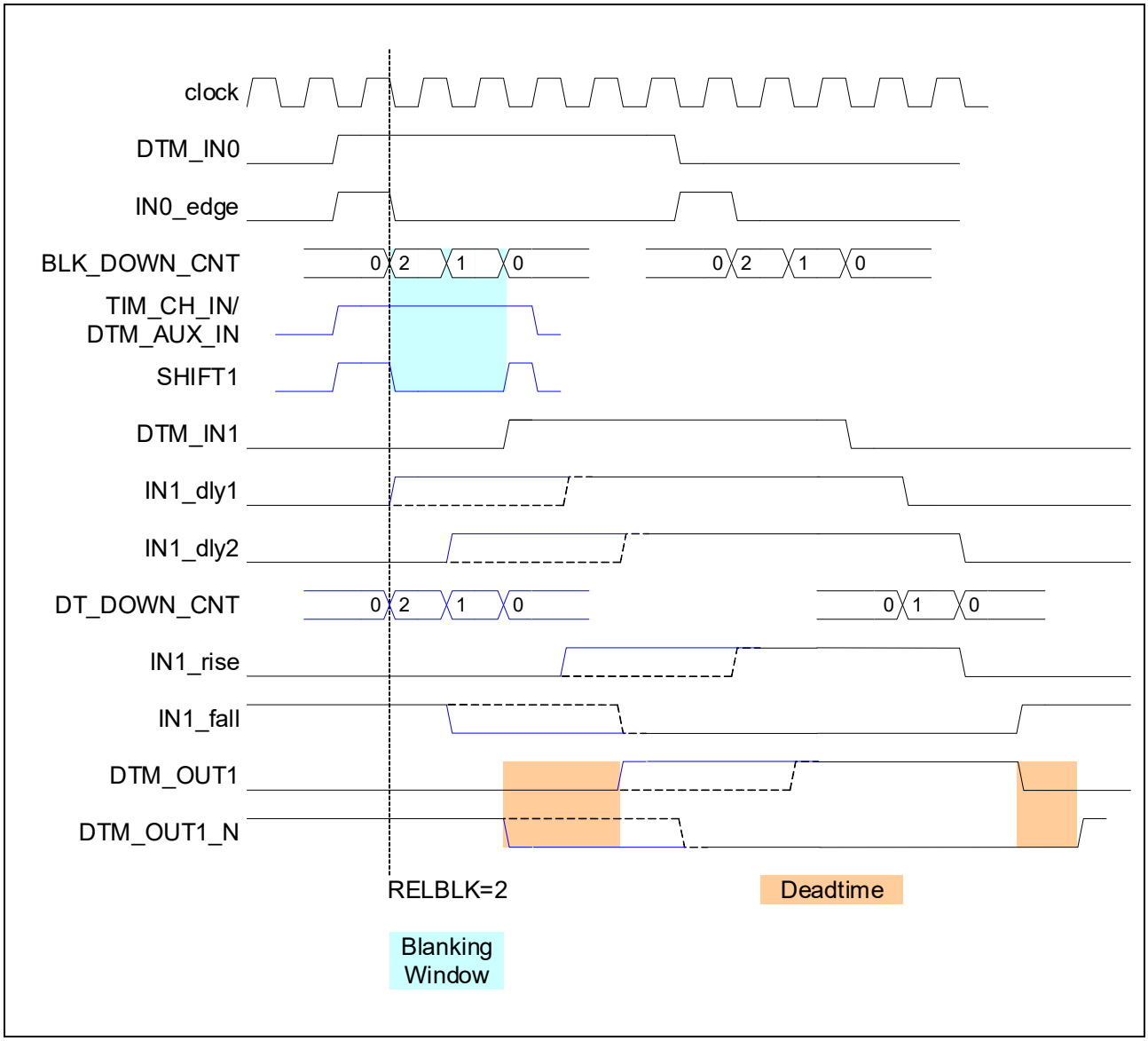


Figure 38.100 Example wave of phase shift on channel 1

38.18.4 Multiple output signal combination

Each channel provides additionally the possibility to combine the channel inputs DTM_IN[x] and SHIFT[x] or COUT[x–1] (selected by I1SEL_[x]) by an AND or an XOR gate (selected by O1F_[x]).

It is recommended to use the combination of signals only if bit field RELBLK of register DTM[i]_PS_CTRL is 0. Otherwise, the signal TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN may be disturbed by the blanking window counter.

Together with the inverter inside sub-module DTM_PSU (selected by IN_POL), the inverter on each output of a channel (selected by POL0_[x] / POL1_[x]) and the possibility to change polarity of DTM_IN[x] inside connected TOM/ATOM channel, a (N)AND, (N)OR or X(N)OR combination of the signals is possible.

38.18.4.1 Combination of input signal TIM_CH_IN/AUX_IN with TOM/ATOM signal

If the input selection I1SEL_[x] of a channel x is set to 0, the output selection O1SEL_[x] is set to 1 and SWAP_[x] is set to 0, depending on PSU_IN_SEL either TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN can be combined with signal DTM_IN[x]. The function of combination on DTM output DTM[i]_OUT[x]_N (and also COUT[x]) is defined by O1F_[x] in the following way:

Table 38.230 The function of combination on DTM output DTM[i]_OUT[x]_N

	O1F_x	POL1_x	IN_POL	(A)TOM output inverted
XOR	01	0	0	no
AND	10	0	0	no
XNOR	01	1	0	no
NAND	10	1	0	no
XNOR	01	1	1	yes
OR	10	1	1	yes
XOR	01	0	1	yes
NOR	10	0	1	yes

Note: The inversion of the (A)TOM output can be reached by switching the SL bit (for TOM and ATOM SOMP/SOMC mode).

38.18.4.2 Combination of multiple TOM/ATOM output signals

If the input selection I1SEL_[x] of a channel x (with x = 1 to 3) is set to 1, the output selection O1SEL_[x] is set to 1 and SWAP_[x] is set to 0, the output of the preceding DTM channel COUT[x–1] can be combined with signal DTM_IN[x]. The function of combination on DTM output DTM[i]_OUT[x]_N (and also COUT[x]) is defined by O1F_[x] in the following way:

Table 38.231 Function of combination on DTM on channel x = 1..3 output DTM[i]_OUT[x]_N (and also COUT[x])

	O1F_x	POL1_x	POL1_x-1	(A)TOM output inverted
XOR	01	0	0	no
AND	10	0	0	no
XNOR	01	1	0	no
NAND	10	1	0	no
XNOR	01	1	1	yes
OR	10	1	1	yes
XOR	01	0	1	yes
NOR	10	0	1	yes

By setting I1SEL_[x] to 1 on all four channel, a combination of all four signals DTM_IN0 to DTM_IN3 can be achieved (combinatorial chain).

To allow also combination of signals generated for output DTM[i]_OUT[x], the outputs 0 and 1 can be swapped by setting bit SWAP_ [x] for channel x.

38.18.4.3 Pulse generation on edge

Another feature of the DTM is to generate on the second output DTM[i]_OUT[x]_N a pulse on every edge of corresponding input signal DTM[i]_IN[x]. This can be reached by configuring O1SEL_[x] to 1, i.e. selecting signal edge_trigg_[x] as the output signal (O1F_[x] has to be 00_B). The signal edge_trigg_[x] is depicted in **Figure 38.97**.

The pulse length can be adjusted individually for each edge type by the configuration value REL_RISE and REL_FALL of register DTM[i]_CH[x]_DV. The parameter REL_RISE defines the pulse length in case of a rising edge on input DTM[i]_IN[x], the parameter REL_FALL define the pulse length in case of a falling edge on input DTM[i]_IN[x].

The generated edge signal edge_trigg_[x] can be combined with the output signal of the preceding DTM channel x–1 at channel input COUT[x–1] (see **Figure 38.96, DTM channel overview**). With the configuration of CIS[x] = 1 and I1SEL_[x] = 1, CII[x] =0 and POL1_[x] = 1, the signal edge_trigg_[x] of channel x is ORed with the inverse signal at channel input COUT[x.1]. The signal at COUT[x.1] can be inverted by changing POL1_[x.1] of channel x.1.

As a result of this configuration one can generate at each edge on DTM input DTM_IN[x] a pulse signal and OR-combine these generated pulse signals with the generated signal of preceding DTM channel. If the combinatorial chain is configured over all four DTM channel the final signal is available at last DTM output DTM_OUT3_N.

38.18.5 Synchronous update of channel control register 2

It is possible to use the shadow register DTM[i]_CH_CTRL2_SR and a selected edge of one of the channel 0 to 3 to update the work register DTM[i]_CH_CTRL2. The update mechanism and its configuration are depicted in the following figure.

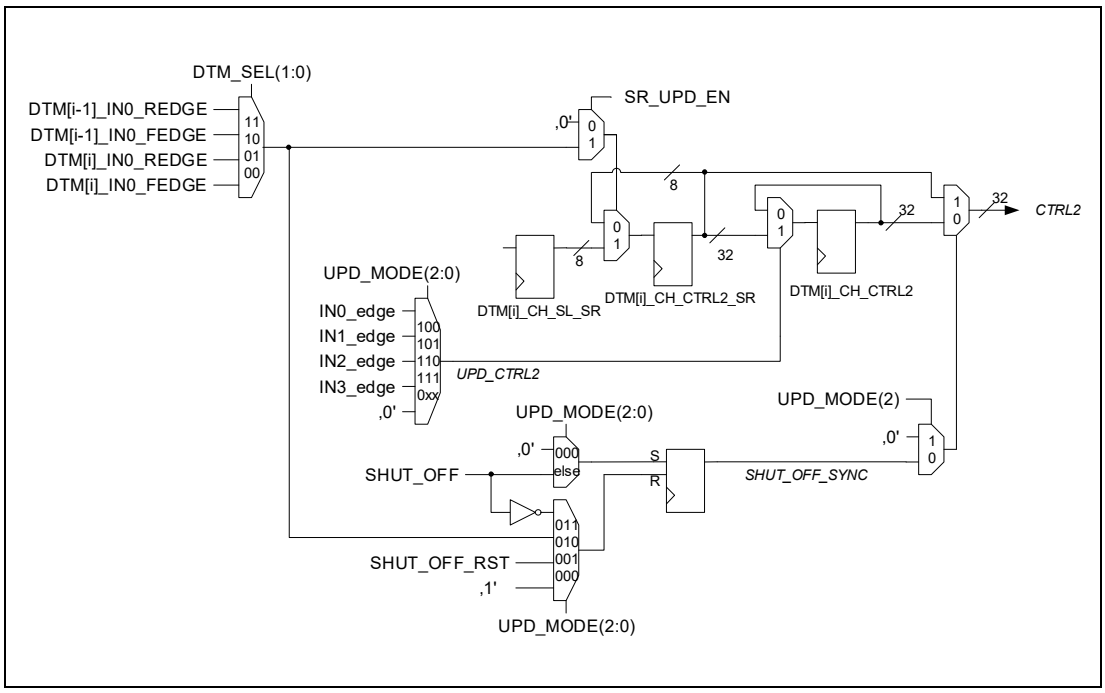


Figure 38.101 Synchronous update mechanism of register DTM[i]_CH_CTRL2

If enabled by the bit field UPD_MODE of register DTM[i]_CTRL (i.e. UPD_MODE = 1xx), the register DTM[i]_CH_CTRL2_SR serves as a shadow register of register DTM[i]_CH_CTRL2. The update is then triggered by an edge on one of the selected inputs DTM_IN0 to DTM_IN3.

The synchronous update allows the user to change output polarity, the selection of constant signal level, the constant signal level itself and the switch to/from feed through path on all four channels in parallel synchronized to one of the input edges on DTM_IN0 to DTM_IN3.

38.18.6 DTM output shut off

A fast shut off for the eight outputs of DTM instance i can be triggered by one of the two assigned inputs TIM[n]_CH_IN or DTM[i]_AUX_IN or the two inputs TIM[m]_CH_IN or DTM[i–1]_AUX_IN of the previous DTM instance i–1. The selection of the trigger signal source is done by the bits PSU_IN_SEL and DTM_SEL(1) (see **Figure 38.99**). The selected trigger signal is named SHUT_OFF. Enabling of the shut off feature is done by setting UPD_MODE(2:0) to one of the values 001_B, 010_B or 011_B.

The shut off behavior of the DTM outputs is defined by the value of register DTM[i]_CH_CTRL2_SR. If the shut off feature is enabled by UPD_MODE, as long as the signal SHUT_OFF_SYNC is 0, the register DTM[i]_CH_CTRL2 defines the output signal behavior. If the signal SHUT_OFF_SYNC is 1, the register DTM[i]_CH_CTRL2_SR defines the output signal behavior. The signal SHUT_OFF_SYNC is set to 1 if signal SHUT_OFF switches to 1. The reset depends on value of UPD_MODE(2:0).

There are three different ways to reset the signal SHUT_OFF_SYNC to 0: – the CPU writes a 1 to bit SHUT_OFF_RST of register DTM_CH_CTRL1 – synchronous to an edge on DTM channel 0 input of this DTM instance i or on an edge on DTM channel 0 input of preceding DTM instance i–1. – asynchronous if signal SHUT_OFF switches back to 0 Additionally, setting UPD_MODE(2:0) to a value 000_B or $1xx_B$ resets also the signal S HUT_OFF_SYNC. **Figure 38.101** depicts the shut off feature and the different shut off release possibilities.

NOTE

The reset of SHUT_OFF_SYNC has lower priority than the set of this signal.

A second shadow register DTM[i]_CH_SR exist for the eight SL bits (SLx_y_SR) of the shadow register DTM[i]_CH_CTRL2_SR. If enabled by configuration bit SR_UPD_EN of register DTM[i]_CTRL, the update of SL bits of register DTM[i]_CH_CTRL2_SR can be triggered by one of the signals selected by bit field DTM_SEL of register DTM[i]_CTRL. This trigger signal is either the rising or the falling edge detected on DTM[i]_IN0 of instance i or the rising or the falling edge on DTM[i–1]_IN0 of preceding instance i–1.

As depicted in **Figure 38.93**, **Figure 38.94** and **Figure 38.99** the DTM input signal TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN can be forwarded to the succeeding instance. Thus, it can be used to trigger shut off in two consecutive DTM instances.

38.18.7 DTM connections on GTM-IP top level

The DTM, if present, is placed behind the outputs of a TOM or ATOM. The outputs of the DTM are routed directly to the top level ports of GTM-IP. If there is a DTM placed behind a TOM or ATOM depends on the GTM-IP device configuration. In case of a DTM behind a TOM or ATOM, the outputs (A)TOM_OUT and (A)TOM_OUT_T are connected to DTM inputs DTM_IN and DTM_IN_T. The outputs of the DTM are routed directly to the top level of GTM-IP. The behavior of DTM after reset is shown in **Figure 38.102**

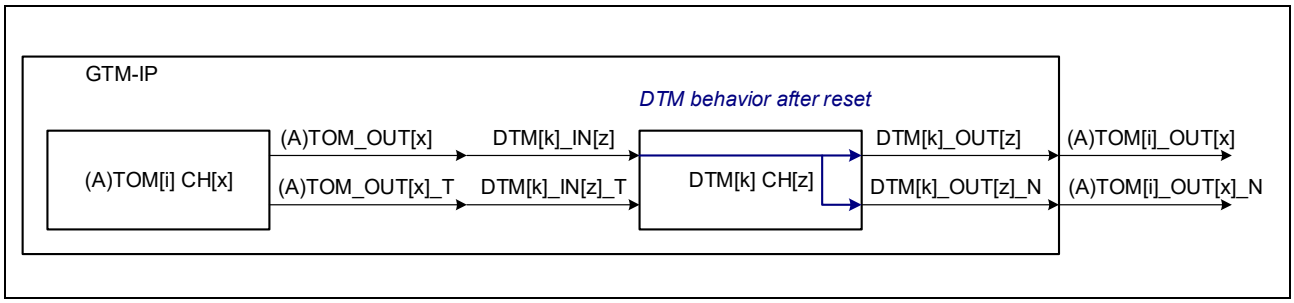


Figure 38.102 DTM behavior after reset

To route the signal DTM[k]_IN[z]_T to the DTM output DTM_[k]_OUT[z]_N, the following DTM channel configuration has to be chosen:

O1F_[x] = 11_B, O1SEL_[x] = 1 and DT1_[x] = 1.

The signals names and the signal routing in the case of no DTM instance is placed behind a TOM or ATOM is shown in **Figure 38.103**.

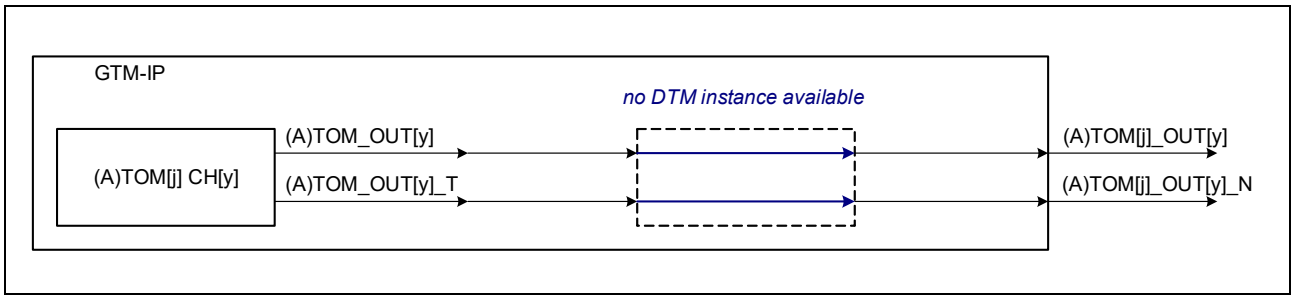


Figure 38.103 (A)TOM output signal routing in case of no DTM instance available

38.18.8 Configuration Register Overview

The following table gives an overview of the DTM configuration register.

Table 38.232 Register list

Symbol	Register Name	Details in Section
CDTM[i]_DTM[j]_CTRL	CDTMi DTMj global configuration and control register	**38.18.9.1**
CDTM[i]_DTM[j]_CH_CTRL1	CDTMi DTMj channel control register 1	**38.18.9.2**
CDTM[i]_DTM[j]_CH_CTRL2	CDTMi DTMj channel control register 2	**38.18.9.3**
CDTM[i]_DTM[j]_CH_CTRL2_SR	CDTMi DTMj channel control register 2 shadow	**38.18.9.4**
CDTM[i]_DTM[j]_CH_CTRL3	CDTMi DTMj channel control register 3	**38.18.9.5**
CDTM[i]_DTM[j]_PS_CTRL	CDTMi DTMj phase shift unit configuration and control register	**38.18.9.6**
CDTM[i]_DTM[j]_CH[z]_DTV	CDTMi DTMj dead time reload values	**38.18.9.7**
CDTM[i]_DTM[j]_CH_SR	CDTM DTMj channel shadow register	**38.18.9.8**

38.18.9 DTM Configuration Register Description

38.18.9.1 CDTM[i]_DTM[j]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4000_H$ + (400_H × i) + (40_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SHUT_OFF_RST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SR_UPD_EN	—	UPD_MODE			DTM_SEL		CLK_SEL	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.233 CDTM[i]_DTM[j]_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	SHUT_OFF_RST	Shut off reset Writing a '1' releases shut off (resets signal SHUT_OFF_SYNC if selected by UPD_MODE(2:0) = "001")
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	SR_UPD_EN	Shadow register update enable 0: No update of SLx_y_SR register bits in register DTM[i]_CH_CTRL2_SR 1: Update of SLx_y_SR register bits in register DTM[i]_CH_CTRL2_SR on trigger
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	UPD_MODE	Update mode 000_B: Asynchronous update – DTM[i]_CH_CTRL2_SR not used for update of DTM[i]_CH_CTRL2 001_B: Shut off release by writing '1' to bit SHUT_OFF_RST of register DTM[i]_CTRL 010_B: Shut off release by an edge on DTM[i]_IN0 or DTM[i-1]_IN0 (defined by bit field DTM_SEL of register DTM[i]_CTRL) 011_B: Shut off release by shut off signal SHUT_OFF (defined by bits PSU_IN_SEL and IN_POL of register DTM[i]_PS_CTRL and DTM_SEL(2) of register DTM[i]_CTRL) 100_B: Signal IN0_edge used to trigger update of DTM[i]_CH_CTRL2 with content of DTM[i]_CH_CTRL2_SR 101_B: Signal IN1_edge used to trigger update of DTM[i]_CH_CTRL2 with content of DTM[i]_CH_CTRL2_SR 110_B: Signal IN2_edge used to trigger update of DTM[i]_CH_CTRL2 with content of DTM[i]_CH_CTRL2_SR 111_B: Signal IN3_edge used to trigger update of DTM[i]_CH_CTRL2 with content of DTM[i]_CH_CTRL2_SR

Table 38.233 CDTM[i]_DTM[j]_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
3, 2	DTM_SEL	Select DTM update and SHUT_OFF reset signal 00_B: Select falling edge on DTM[i] channel 0 input 01_B: Select rising edge on DTM[i] channel 0 input 10_B: Select falling edge on DTM[i-1] channel 0 input 11_B: Select rising edge on DTM[i-1] channel 0 input 0-: Shut off by signal TIM_CH_IN0, TIM_CH_IN1 or DTM_AUX_IN 1-: Shut off by signal DTM[i-1]_PSU_IN
1, 0	CLK_SEL	Clock source select 00_B: SYS_CLK selected 01_B: CMU_CLK0 selected 10_B: CMU_CLK1 selected (if DTM is connected to an ATOM) 11_B: CMU_CLK2 selected (if DTM is connected to an ATOM)

38.18.9.2 CDTM[i]_DTM[j]_CH_CTRL1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4004_H$ + (400_H × i) + (40_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	O1F_3		SWAP_3	SH_EN_3	I1SEL_3	O1SEL_3	—	XDT_EN_2_3	O1F_2		SWAP_2	SH_EN_2	I1SEL_2	O1SEL_2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	O1F_1		SWAP_1	SH_EN_1	I1SEL_1	O1SEL_1	—	XDT_EN_0_1	O1F_0		SWAP_0	—	I1SEL_0	O1SEL_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 38.234 CDTM[i]_DTM[j]_CH_CTRL1 Register Contents (1/3)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	O1F_3	Output 1 function channel 3 00_B: Signal edge_trigg is selected 01_B: XOR of DTM[j]_IN3 and signal SHIFT3/OUT2 10_B: AND of DTM[j]_IN3 and signal SHIFT3/OUT2 11: DTM[j]_IN3_T selected
27	SWAP_3	Swap outputs DTM[i]_CH[3]_OUT0 and DTM[i]_CH[3]_OUT1 (before final output register) 0: Outputs not swapped 1: Swap outputs DTM[i]_OUT3 and DTM[i]_OUT3_N
26	SH_EN_3	Shift enable channel 3 0: DTM[i]_IN2 is not used; no input signal shift 1: Signal selected by I1SEL_3 triggers update of DTM[i]_IN3 with input of DTM[i]_IN2 → input signal shift
25	I1SEL_3	Input 1 select channel 3 0: Signal SHIFT2 selected 1: Signal OUT2 selected
24	O1SEL_3	Output 1 select channel 3 0: Inverse dead time signal selected 1: Special function on output 1 selected (defined by O1F_3)
23	Reserved	This bit is always read as 0. When written, write the initial value.
22	XDT_EN_2_3	Cross dead time enable on channel 0 and 1 0: Cross dead time disabled on channel 2 and 3 1: Cross dead time enabled on channel 2 and 3 **NOTES** 1. When a '1' is written to bit XDT_EN_2_3, the internal register IN[x]_dly1, IN[x]_dly2 and DT_DOWN_CNT is reset to '0' (x:2, 3) 2. TSEL0_[x] and SH_EN_[x] must be '0' for using cross dead time to avoid wrong input signals. (x:2, 3)

Table 38.234 CDTM[i]_DTM[j]_CH_CTRL1 Register Contents (2/3)

Bit Position	Bit Name	Function
21, 20	O1F_2	Output 1 function channel 2 00_B: Signal edge_trigg is selected 01_B: XOR of DTM[j]_IN2 and signal SHIFT2/OUT1 10_B: AND of DTM[j]_IN2 and signal SHIFT2/OUT1 11_B: DTM[j]_IN2_T selected
19	SWAP_2	Swap outputs DTM[i]_CH[2]_OUT0 and DTM[i]_CH[2]_OUT1 (before final output register) 0: Outputs not swapped 1: Swap outputs DTM[i]_OUT2 and DTM[i]_OUT2_N
18	SH_EN_2	Shift enable channel 2 0: DTM[i]_IN1 is not used; no input signal shift 1: Signal selected by I1SEL_2 triggers update of DTM[i]_IN2 with input of DTM[i]_IN1 → input signal shift
17	I1SEL_2	Input 1 select channel 2 0: Signal SHIFT1 selected 1: Signal OUT1 selected
16	O1SEL_2	Output 1 select channel 2 0: Inverse dead time signal selected 1: Special function on output 1 selected (defined by O1F_2)
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	O1F_1	Output 1 function channel 1 00_B: Signal edge_trigg is selected 01_B: XOR of DTM[j]_IN1 and signal SHIFT1/OUT0 10_B: AND of DTM[j]_IN1 and signal SHIFT1/OUT0 11_B: DTM[j]_IN1_T selected
11	SWAP_1	Swap outputs DTM[i]_CH[1]_OUT0 and DTM[i]_CH[1]_OUT1 (before final output register) 0: Outputs not swapped 1: Swap outputs DTM[i]_OUT1 and DTM[i]_OUT1_N
10	SH_EN_1	Shift enable channel 1 0: DTM[i]_IN0 is not used; no input signal shift 1: Signal selected by I1SEL_1 triggers update of DTM[i]_IN1 with input of DTM[i]_IN0 → input signal shift
9	I1SEL_1	Input 1 select channel 1 0: Signal SHIFT1 selected 1: Signal OUT1 selected
8	O1SEL_1	Output 1 select channel 1 0: Inverse dead time signal selected 1: Special function on output 1 selected (defined by O1F_1)
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	XDT_EN_0_1	Cross dead time enable on channel 0 and 1 0: Cross dead time disabled on channel 0 and 1 1: Cross dead time enabled on channel 0 and 1 **NOTES** 1. When a '1' is written to bit XDT_EN_0_1, the internal register IN[x]_dly1, IN[x]_dly2 and DT_DOWN_CNT is reset to '0' (x:0, 1) 2. TSEL0_[x] and SH_EN_1 must be '0' for using cross dead time to avoid wrong input signals. (x:0, 1)
5, 4	O1F_0	Output 1 function channel 0 00_B: Signal edge_trigg is selected 01_B: XOR of DTM[j]_IN0 and signal SHIFT0 10_B: AND of DTM[j]_IN0 and signal SHIFT0 11_B: DTM[j]_IN1_T selected

Table 38.234 CDTM[i]_DTM[j]_CH_CTRL1 Register Contents (3/3)

Bit Position	Bit Name	Function
3	SWAP_0	Swap outputs DTM[i]_CH[0]_OUT0 and DTM[i]_CH[0]_OUT1 (before final output register) 0: Outputs not swapped 1: Swap outputs DTM[i]_OUT0 and DTM[i]_OUT0_N
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	I1SEL_0	Input 1 select channel 0 0: Signal SHIFT0 selected 1: Signal COUT3 from DTM[i–1] selected **NOTE** If it is even I1SEL_0 is not implemented. Then the bit is read as zero and should be written as zero.
0	O1SEL_0	Output 1 select channel 0 0: Inverse dead time signal selected 1: Special function on output 1 selected (defined by O1F_0)

38.18.9.3 CDTM[i]_DTM[j]_CH_CTRL2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4008_H$ + (400_H × i) + (40_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT1_3	SL1_3	OC1_3	POL1_3	DT0_3	SL0_3	OC0_3	POL0_3	DT1_2	SL1_2	OC1_2	POL1_2	DT0_2	SL0_2	OC0_2	POL0_2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT1_1	SL1_1	OC1_1	POL1_1	DT0_1	SL0_1	OC0_1	POL0_1	DT1_0	SL1_0	OC1_0	POL1_0	DT0_0	SL0_0	OC0_0	POL0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.235 CDTM[i]_DTM[j]_CH_CTRL2 Register Contents (1/3)

Bit Position	Bit Name	Function
31	DT1_3	Dead time path enable on output 1 channel 3 0: Feed through from DTM_IN3 to DTM[i]_OUT3_N enabled 1: Dead time path enabled
30	SL1_3	Signal level on output 1 channel 3 0: Signal Level is ‘0’ on output if OC1_3 = 1 1: Signal Level is ‘1’ on output if OC1_3 = 1
29	OC1_3	Output 1 control channel 3 0: Functional output 1: Constant output defined by SL1_3
28	POL1_3	Polarity on output 1 channel 3 0: Output signal not inverted 1: Output signal inverted
27	DT0_3	Dead time path enable on output 0 channel 3 0: Feed through from DTM_IN3 to DTM[i]_OUT3 enabled 1: Dead time path enabled
26	SL0_3	Signal level on output 0 channel 3 0: Signal Level is ‘0’ on output if OC0_3 = 1 1: Signal Level is ‘1’ on output if OC0_3 = 1
25	OC0_3	Output 0 control channel 3 0: Functional output 1: Constant output defined by SL0_3
24	POL0_3	Polarity on output 0 channel 3 0: Output signal not inverted 1: Output signal inverted
23	DT1_2	Dead time path enable on output 1 channel 2 0: Feed through from DTM_IN2 to DTM[i]_OUT2_N enabled 1: Dead time path enabled
22	SL1_2	Signal level on output 1 channel 2 0: Signal Level is '0' on output if OC1_2 = 1 1: Signal Level is '1' on output if OC1_2 = 1
21	OC1_2	Output 1 control channel 2 0: Functional output 1: Constant output defined by SL1_2
20	POL1_2	Polarity on output 1 channel 2 0: Output signal not inverted 1: Output signal inverted

Table 38.235 CDTM[i]_DTM[j]_CH_CTRL2 Register Contents (2/3)

Bit Position	Bit Name	Function
19	DT0_2	Dead time path enable on output 0 channel 2 0: Feed through from DTM_IN2 to DTM[i]_OUT2 enabled 1: Dead time path enabled
18	SL0_2	Signal level on output 0 channel 2 0: Signal Level is '0' on output if OC0_2 = 1 1: Signal Level is '1' on output if OC0_2 = 1
17	OC0_2	Output 0 control channel 2 0: Functional output 1: Constant output defined by SL0_2
16	POL0_2	Polarity on output 0 channel 2 0: Output signal not inverted 1: Output signal inverted
15	DT1_1	Dead time path enable on output 1 channel 1 0: Feed through from DTM_IN1 to DTM[i]_OUT1_N enabled 1: Dead time path enabled
14	SL1_1	Signal level on output 1 channel 1 0: Signal Level is '0' on output if OC1_1 = 1 1: Signal Level is '1' on output if OC1_1 = 1
13	OC1_1	Output 1 control channel 1 0: Functional output 1: Constant output defined by SL1_1
12	POL1_1	Polarity on output 1 channel 1 0: Output signal not inverted 1: Output signal inverted
11	DT0_1	Dead time path enable on output 0 channel 1 0: Feed through from DTM_IN1 to DTM[i]_OUT1 enabled 1: Dead time path enabled
10	SL0_1	Signal level on output 0 channel 1 0: Signal Level is '0' on output if OC0_1 = 1 1: Signal Level is '1' on output if OC0_1 = 1
9	OC0_1	Output 0 control channel 1 0: Functional output 1: Constant output defined by SL0_1
8	POL0_1	Polarity on output 0 channel 1 0: Output signal not inverted 1: Output signal inverted
7	DT1_0	Dead time path enable on output 1 channel 0 0: Feed through from DTM_IN0 to DTM[i]_OUT0_N enabled 1: Dead time path enabled
6	SL1_0	Signal level on output 1 channel 0 0: Signal Level is '0' on output if OC1_0 = 1 1: Signal Level is '1' on output if OC1_0 = 1
5	OC1_0	Output 1 control channel 0 0: Functional output 1: Constant output defined by SL1_0
4	POL1_0	Polarity on output 1 channel 0 0: Output signal not inverted 1: Output signal inverted
3	DT0_0	Dead time path enable on output 0 channel 0 0: Feed through from DTM_IN0 to DTM[i]_OUT0 enabled 1: Dead time path enabled
2	SL0_0	Signal level on output 0 channel 0 0: Signal Level is '0' on output if OC0_0 = 1 1: Signal Level is '1' on output if OC0_0 = 1
1	OC0_0	Output 0 control channel 0 0: Functional output 1: Constant output defined by SL0_0

Table 38.235 CDTM[i]_DTM[j]_CH_CTRL2 Register Contents (3/3)

Bit Position	Bit Name	Function
0	POL0_0	Polarity on output 0 channel 0 0: Output signal not inverted 1: Output signal inverted

38.18.9.4 CDTM[i]_DTM[j]_CH_CTRL2_SR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E400C_H$ + (400_H × i) + (40_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT1_3_SR	SL1_3_SR	OC1_3_SR	POL1_3_SR	DT0_3_SR	SL0_3_SR	OC0_3_SR	POL0_3_SR	DT1_2_SR	SL1_2_SR	OC1_2_SR	POL1_2_SR	DT0_2_SR	SL0_2_SR	OC0_2_SR	POL0_2_SR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT1_1_SR	SL1_1_SR	OC1_1_SR	POL1_1_SR	DT0_1_SR	SL0_1_SR	OC0_1_SR	POL0_1_SR	DT1_0_SR	SL1_0_SR	OC1_0_SR	POL1_0_SR	DT0_0_SR	SL0_0_SR	OC0_0_SR	POL0_0_SR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.236 CDTM[i]_DTM[j]_CH_CTRL2_SR Register Contents (1/3)

Bit Position	Bit Name	Function
31	DT1_3_SR	Dead time path enable on output 1 channel 3 shadow register 0: Feed through from DTM_IN3 to DTM[i]_OUT3_N 1: Dead time path enabled
30	SL1_3_SR	Signal level on output 1 channel 3 shadow register 0: Signal Level is '0' on output if OC1_3 = 1 1: Signal Level is '1' on output if OC1_3 = 1
29	OC1_3_SR	Output 1 control channel 3 shadow register 0: Functional output 1: Constant output defined by SL1_3
28	POL1_3_SR	Polarity on output 1 channel 3 shadow register 0: Output signal not inverted 1: Output signal inverted
27	DT0_3_SR	Dead time path enable on output 0 channel 3 shadow register 0: Feed through from DTM_IN3 to DTM[i]_OUT3 1: Dead time path enabled
26	SL0_3_SR	Signal level on output 0 channel 3 shadow register 0: Signal Level is '0' on output if OC0_3 = 1 1: Signal Level is '1' on output if OC0_3 = 1
25	OC0_3_SR	Output 0 control channel 3 shadow register 0: Functional output 1: Constant output defined by SL0_3
24	POL0_3_SR	Polarity on output 0 channel 3 shadow register 0: Output signal not inverted 1: Output signal inverted
23	DT1_2_SR	Dead time path enable on output 1 channel 2 shadow register 0: Feed through from DTM_IN2 to DTM[i]_OUT2_N 1: Dead time path enabled
22	SL1_2_SR	Signal level on output 1 channel 2 shadow register 0: Signal Level is '0' on output if OC1_2 = 1 1: Signal Level is '1' on output if OC1_2 = 1
21	OC1_2_SR	Output 1 control channel 2 shadow register 0: Functional output 1: Constant output defined by SL1_2
20	POL1_2_SR	Polarity on output 1 channel 2 shadow register 0: Output signal not inverted 1: Output signal inverted

Table 38.236 CDTM[i]_DTM[j]_CH_CTRL2_SR Register Contents (2/3)

Bit Position	Bit Name	Function
19	DT0_2_SR	Dead time path enable on output 0 channel 2 shadow register 0: Feed through from DTM_IN2 to DTM[i]_OUT2 1: Dead time path enabled
18	SL0_2_SR	Signal level on output 0 channel 2 shadow register 0: Signal Level is '0' on output if OC0_2 = 1 1: Signal Level is '1' on output if OC0_2 = 1
17	OC0_2_SR	Output 0 control channel 2 shadow register 0: Functional output 1: Constant output defined by SL0_2
16	POL0_2_SR	Polarity on output 0 channel 2 shadow register 0: Output signal not inverted 1: Output signal inverted
15	DT1_1_SR	Dead time path enable on output 1 channel 1 shadow register 0: Feed through from DTM_IN1 to DTM[i]_OUT1_N 1: Dead time path enabled
14	SL1_1_SR	Signal level on output 1 channel 1 shadow register 0: Signal Level is '0' on output if OC1_1 = 1 1: Signal Level is '1' on output if OC1_1 = 1
13	OC1_1_SR	Output 1 control channel 1 shadow register 0: Functional output 1: Constant output defined by SL1_1
12	POL1_1_SR	Polarity on output 1 channel 1 shadow register 0: Output signal not inverted 1: Output signal inverted
11	DT0_1_SR	Dead time path enable on output 0 channel 1 shadow register 0: Feed through from DTM_IN1 to DTM[i]_OUT1 enabled 1: Dead time path enabled
10	SL0_1_SR	Signal level on output 0 channel 1 shadow register 0: Signal Level is '0' on output if OC0_1 = 1 1: Signal Level is '1' on output if OC0_1 = 1
9	OC0_1_SR	Output 0 control channel 1 shadow register 0: Functional output 1: Constant output defined by SL0_1
8	POL0_1_SR	Polarity on output 0 channel 1 shadow register 0: Output signal not inverted 1: Output signal inverted
7	DT1_0_SR	Dead time path enable on output 1 channel 0 shadow register 0: Feed through from DTM_IN0 to DTM[i]_OUT0_N enabled 1: Dead time path enabled
6	SL1_0_SR	Signal level on output 1 channel 0 shadow register 0: Signal Level is '0' on output if OC1_0 = 1 1: Signal Level is '1' on output if OC1_0 = 1
5	OC1_0_SR	Output 1 control channel 0 shadow register 0: Functional output 1: Constant output defined by SL1_0
4	POL1_0_SR	Polarity on output 1 channel 0 shadow register 0: Output signal not inverted 1: Output signal inverted
3	DT0_0_SR	Dead time path enable on output 0 channel 0 shadow register 0: Feed through from DTM_IN0 to DTM[i]_OUT0 enabled 1: Dead time path enabled
2	SL0_0_SR	Signal level on output 0 channel 0 shadow register 0: Signal Level is '0' on output if OC0_0 = 1 1: Signal Level is '1' on output if OC0_0 = 1
1	OC0_0_SR	Output 0 control channel 0 shadow register 0: Functional output 1: Constant output defined by SL0_0

Table 38.236 CDTM[i]_DTM[j]_CH_CTRL2_SR Register Contents (3/3)

Bit Position	Bit Name	Function
0	POL0_0_SR	Polarity on output 0 channel 0 shadow register 0: Output signal not inverted 1: Output signal inverted

38.18.9.5 CDTM[i]_DTM[j]_CH_CTRL3

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4028_H$ + (400_H × i) + (40_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TSEL1_3	TSEL0_3	CIS3	CII3	—	—	—	—	TSEL1_2	TSEL0_2	CIS2	CII2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	TSEL1_1	TSEL0_1	CIS1	CII1	—	—	—	—	TSEL1_0	TSEL0_0	CIS0	CII0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.237 CDTM[i]_DTM[j]_CH_CTRL3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	TSEL1_3	Input selection combinational logic path 0: Use DTM[i]_IN3 as input for combinational logic path 1: Use DTM[i]_IN3_T as input for combinational logic path
26	TSEL0_3	Input selection for dead time / edge trigger generation 0: Use DTM[i]_IN3 as input for dead time / edge trigger generation 1: Use DTM[i]_IN3_T as input for dead time / edge trigger generation
25	CIS3	Combinational input select channel 3 0: Select input DTM[i]_IN3 1: Select internal signal edge_trigg_3
24	CII3	Combinational input invert channel 3 0: Do not invert input 1: Invert input
23 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19	TSEL1_2	Input selection combinational logic path 0: Use DTM[i]_IN2 as input for combinational logic path 1: Use DTM[i]_IN2_T as input for combinational logic path
18	TSEL0_2	Input selection for dead time / edge trigger generation 0: Use DTM[i]_IN2 as input for dead time / edge trigger generation 1: Use DTM[i]_IN2_T as input for dead time / edge trigger generation
17	CIS2	Combinational input select channel 2 0: Select input DTM[i]_IN2 1: Select internal signal edge_trigg_2
16	CII2	Combinational input invert channel 2 0: Do not invert input 1: Invert input
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	TSEL1_1	Input selection combinational logic path 0: Use DTM[i]_IN1 as input for combinational logic path 1: Use DTM[i]_IN1_T as input for combinational logic path
10	TSEL0_1	Input selection for dead time / edge trigger generation 0: Use DTM[i]_IN1 as input for dead time / edge trigger generation 1: Use DTM[i]_IN1_T as input for dead time / edge trigger generation

Table 38.237 CDTM[i]_DTM[j]_CH_CTRL3 Register Contents (2/2)

Bit Position	Bit Name	Function
9	CIS1	Combinational input select channel 1 0: Select input DTM[i]_IN1 1: Select internal signal edge_trigg_1
8	CII1	Combinational input invert channel 1 0: Do not invert input 1: Invert input
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	TSEL1_0	Input selection combinational logic path 0: Use DTM[i]_IN0 as input for combinational logic path 1: Use DTM[i]_IN0_T as input for combinational logic path
2	TSEL0_0	Input selection for dead time / edge trigger generation 0: Use DTM[i]_IN0 as input for dead time / edge trigger generation 1: Use DTM[i]_IN0_T as input for dead time / edge trigger generation
1	CIS0	Combinational input select channel 0 0: Select input DTM[i]_IN0 1: Select internal signal edge_trigg_0
0	CII0	Combinational input invert channel 0 0: Do not invert input 1: Invert input

38.18.9.6 CDTM[i]_DTM[j]_PS_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4010_H$ + (400_H × i) + (40_H × j)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	SHIFT_SEL		—	TIM_SEL	IN_POL	PSU_IN_SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RELBLK									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.238 CDTM[i]_DTM[j]_PS_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21, 20	SHIFT_SEL	Shift select 00_B: DTM channel 1 is connected via signal SHIFT1 with TIM_CH_IN0, TIM_CH_IN1/DTM_AUX_IN 01_B: DTM channel 2 is connected via signal SHIFT2 with TIM_CH_IN0, TIM_CH_IN1/DTM_AUX_IN 10_B: DTM channel 3 is connected via signal SHIFT3 with TIM_CH_IN0, TIM_CH_IN1/DTM_AUX_IN 11_B: DTM channel 0 is connected via signal SHIFT0 with TIM_CH_IN0, TIM_CH_IN1/DTM_AUX_IN **NOTE** If a channel is not implemented the value is unused. A write with this unused value returns "10" on status.
19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TIM_SEL	TIM input select 0: Select TIM_IN0 1: Select TIM_IN1
17	IN_POL	Input polarity 0: Input signal is not inverted 1: Input signal is inverted
16	PSU_IN_SEL	PSU input select 0: TIM_CH_IN0 or TIM_CH_IN1 selected 1: DTM_AUX_IN selected
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	RELBLK	Reload value blanking window **NOTE** A value of 000_H resets counter BLK_DOWN_CNT

38.18.9.7 CDTM[i]_DTM[j]_CH[z]_DTV

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4014_H$ + (400_H × i) + (40_H × j) + (4_H × z)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	RELFALL									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RELRISE									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.239 CDTM[i]_DTM[j]_CH[z]_DTV Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	RELFALL	Reload value for falling edge dead time
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	RELRISE	Reload value for rising edge dead time

38.18.9.8 CDTM[i]_DTM[j]_CH_SR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $E4024_H$ + (400_H × i) + (40_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SL1_3_SR_SR	SL0_3_SR_SR	SL1_2_SR_SR	SL0_2_SR_SR	SL1_1_SR_SR	SL0_1_SR_SR	SL1_0_SR_SR	SL0_0_SR_SR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.240 CDTM[i]_DTM[j]_CH_SR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SL1_3_SR_SR	Shadow register for bit SL1_3_SR of register DTM[i]_CH_CTRL2_SR
6	SL0_3_SR_SR	Shadow register for bit SL0_3_SR of register DTM[i]_CH_CTRL2_SR
5	SL1_2_SR_SR	Shadow register for bit SL1_2_SR of register DTM[i]_CH_CTRL2_SR
4	SL0_2_SR_SR	Shadow register for bit SL0_2_SR of register DTM[i]_CH_CTRL2_SR
3	SL1_1_SR_SR	Shadow register for bit SL1_1_SR of register DTM[i]_CH_CTRL2_SR
2	SL0_1_SR_SR	Shadow register for bit SL0_1_SR of register DTM[i]_CH_CTRL2_SR
1	SL1_0_SR_SR	Shadow register for bit SL1_0_SR of register DTM[i]_CH_CTRL2_SR
0	SL0_0_SR_SR	Shadow register for bit SL0_0_SR of register DTM[i]_CH_CTRL2_SR

38.19 Multi Channel Sequencer (MCS)

38.19.1 Overview

The Multi Channel Sequencer (MCS) sub module is a generic data processing module that is connected to the ARU. One of its major applications is to calculate complex output sequences that may depend on the time base values of the TBU and are processed in combination with the ATOM sub module. Other applications can use the MCS sub module to perform extended data processing of input data resulting from the TIM sub module. Moreover, some applications may process data provided by the CPU within the MCS sub module, and the calculated results are sent to the outputs using the ATOM sub modules.

Table 38.241 summarizes all available generic design parameters of the MCS hardware structure.:

Table 38.241 Generic Design Parameters

Design Parameter	Description
W	Word width of the data path
T	Number of available MCS channels
RDW	RAM data width of connected RAM
RAW	RAM address width used by the MCS for addressing memory
USR	Use second RAM port (0 – one RAM port available, 1 – two RAM ports available)
BAW	Bus Master Address Width
BDW	Bus Master Data Width
URIP	Use RAM input pipeline registers (0 – no register, 1 – use register)
UROP	Use RAM output pipeline registers (0 – no register, 1 – use register)
UDP	Use Decoder Pipeline register (0 – no register, 1 – use register)
UAP	Use ALU Pipeline register (0 – no register, 1 – use register)
NPS	Total number of pipeline stages (with NPS = 3 + URIP + UROP + UDP + UAP)

All MCS instances in the GTM use the values T=8, W=24, RDW=32, RAW=12, USR=1, BAW = 14, BDW = 32, URIP = 1, UROP = 1, UDP = 1, UAP = 1, and NPS = 7.

38.19.2 Architecture

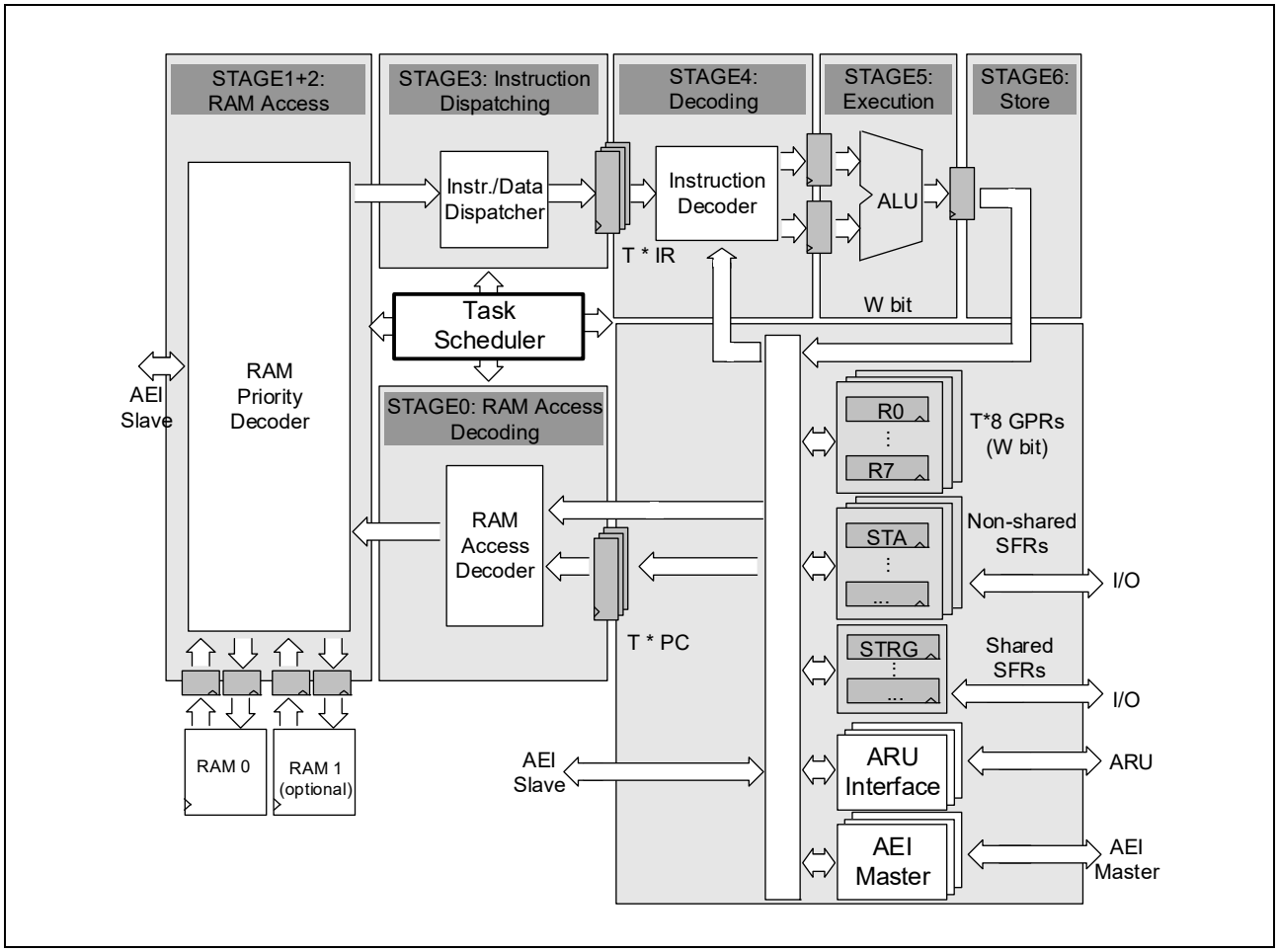


Figure 38.104 MCS Architecture block diagram

Figure 38.104 gives an overview of the MCS architecture assuming that all pipeline registers are implemented.

The data path of the MCS is shared by T so called MCS-channels, whereas each MCS-channel executes a dedicated micro-program that is stored inside the RAM connected to the MCS module.

The connected RAM may contain arbitrary sized code and data sections that are accessible by all MCS-channels and an externally connected master (e.g. a CPU) via AEI Slave interface. More details about the RAM can be found in **Section 38.19.4, Memory Organization**.

An MCS-channel can also be considered as an individual task of a processor that is scheduled to the commonly used data path at a specific point in time. The execution of the different MCS-channels on the different pipeline stages is controlled by a central hardware related task scheduler, which enables immediate task switches in parallel to the program execution. Details about the task scheduler and the available scheduling algorithms can be found in **Section 38.19.3, Scheduling**.

Typically, if data has to be exchanged between different MCS-channels and/or the CPU, the connected RAM, which is accessible by all MCS-channels and the CPU, can be used.

Besides the commonly used data path, each MCS channel has

1. A set of eight General Purpose Registers (GPRs), each W bit wide,
2. A set of non-shared Special Function Registers (SFRs) that are only accessible within a dedicated MCS channels,
3. A set of shared SFRs that are accessible by all MCS channels,
4. A channel specific instruction register (IR),
5. A channel specific program counter register (PC),
6. A dedicated ARU interface for communication with other ARU connected modules,
7. And an AEI Bus Master Interface for controlling other GTM sub modules.

Generally, the GPRs of an MCS channel x are only accessible by its corresponding MCS channel x. However, the MCS provides a configuration that allows an MCS channel x to access the GPRs of its successor MCS channel x+1. This feature can be used to enlarge the number of registers for a specific MCS channel x and/or to exchange data between neighboring channels.

For safety reasons, the register MCS[i]_REG_PROT can be used to define write protections for the neighboring registers of the individual MCS channels.

In order to enable synchronization between different MCS channels and/or the CPU, the MCS provides a common 24 bit wide trigger register that can be accessed as a shared SFR by all MCS channels located in the same module. Writing to STRG sets bits and writing to CTRG clears bits in the common trigger register. To enable triggering of MCS-channels by CPU, the CPU can set bits in the common trigger register by writing to MCS[i]_STRG and clear bits by writing to MCS[i]_CTRG.

Considering the architecture in the figure above and assuming that all available pipeline stages are implemented (the generic parameters URIP, UROP, UDP, and UAP are set to 1), the main actions of the different pipeline stages are as follows:

1. Pipeline stage 0 performs a setup of address, input data, and control signals for the next RAM access of a specific MCS-channel.
2. The actual RAM access of a specific MCS-channel is executed in pipeline stage 1 and 2, assuming an external connection of a synchronous RAM with a latency of one clock cycle.
3. Pipeline stage 3 performs pre-decoding and dispatching of instructions and data resulting from the RAM.
4. In pipeline stage 4 the instructions are decoded and data from the registers are loaded.
5. After that, in pipeline stage 5 the instruction is executed meaning that arithmetic operations are applied.
6. Finally, in pipeline stage 6 the calculated results are stored in the registers.

If any of the pipeline registers is not implemented, the adjacent pipeline stages are merged and thus processed within the same clock cycle.

The RAM priority decoder arbitrates RAM accesses that are requested by the CPU via AEI and by the active MCS-channel. If both, CPU and an MCS-channel request a memory access to the same memory module the MCS-channel is prioritized.

Since the internal registers of the MCS can be updated by different sources (MCS write access by various instructions, CPU write access via AEI slave, MCS write access by neighboring channel) a write conflict occurs if more than one source wants to write to the same register. In this case the result of the register is unpredictable. However, the software should setup its application in a way that such conflicts do not occur.

One exception is the common trigger register, which may be written by multiple sources (different MCS channels and CPU) in order to enable triggering of different MCS channels. Typically, the software should setup its application in a manner that different sources should not write the same bits in the trigger register.

38.19.3 Scheduling

The MCS provides a hardware related task scheduler, which globally controls the execution of the tasks in the different pipeline stages.

The task scheduler implements four different scheduling modes, that can be selected by the SCD_MODE bit field in the MCS[i]_CTRL_STAT register. Depending on the selected scheduling mode, the task scheduler is selecting a dedicated MCS channel that will be executed in pipeline stage 0 in the next clock cycle. Additionally, MCS channels that are already present in the pipeline are shifted to its successor pipeline stage, with each clock cycle. This means, that the execution time of an MCS-channel in a specific pipeline stage is always one clock cycle.

The MCS task scheduler may also schedule an empty cycle to pipeline stage 0, in order to grant a time slice to the CPU for accessing the connected RAM.

NOTE

If the task scheduler assigns an MCS channel to pipeline stage 0, but this channel does not access the RAM, the CPU can access the corresponding RAM, even if the scheduler did not reserve an empty clock cycle.

In the following, the available scheduling modes are described.

38.19.3.1 Round Robin Scheduling

The Round Robin Scheduling Mode implements the simplest scheduling algorithm. This algorithm schedules a predefined set of MCS channels in the range [0; SCD_CH] in ascending order. After the last channel SCD_CH has been assigned to the pipeline, an empty cycle is scheduled in order to enable RAM access for the CPU. The parameter SCD_CH can be controlled by the register MCS[i]_CTRL_STAT. If the value of SCD_CH is greater than T-1, the scheduler assumes a value of T-1 for bit field SCD_CH.

Figure 38.105 shows a timing example of the Round Robin Scheduling with T = 8 MCS-channels (marked as C_0 to C_7) that are scheduled together with a CPU access to a pipeline with and NPS = 7 stages. It is assumed that bit field SCD_CH is set to 7.

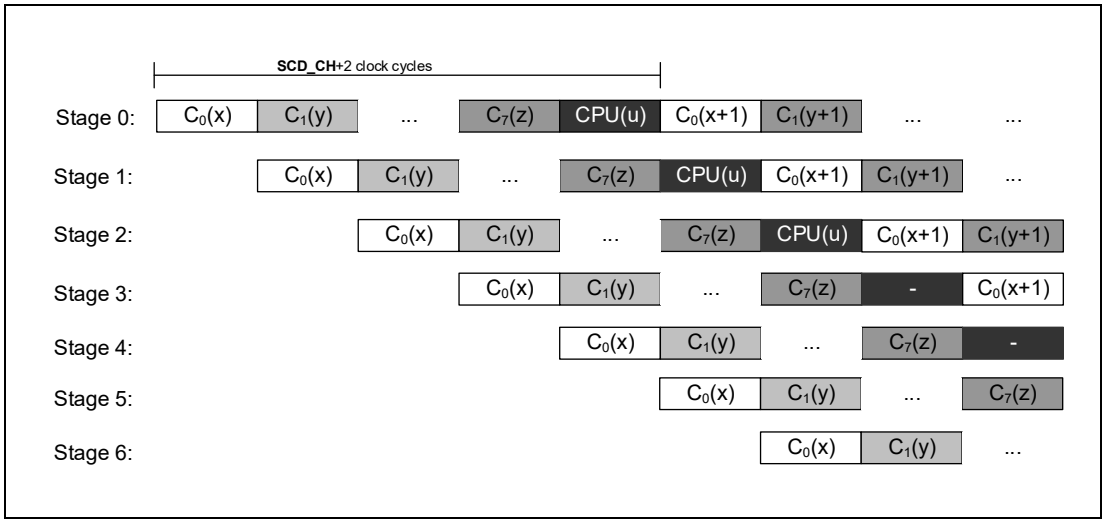


Figure 38.105 Timing of Round Robin Scheduling

The identifier Ci(x) denotes that MCS-channel i is currently executing the instruction or data located in the memory at position x in the corresponding pipeline stage. The figure shows, which MCS-channel is activated in specific pipeline stage at a specific point in time.

Moreover, the figure shows, that the Round Robin scheduling is always repeated after SCD_CH+2 clock cycles, which means that the time duration of an instruction cycle is SCD_CH+2 clock cycles. However, if the value SCD_CH + 2 is less than NPS, the duration of an instruction cycle is limited by the depth of the pipeline to NPS clock cycles. Thus the effective execution time of a single cycle instruction is always MIN(SCD_CH+2, NPS) clock cycles, ignoring the latency of the pipeline.

If NPS is greater than T+1, NPS–T–1 additional empty cycles are inserted at the end of a round trip cycle. In this case the round trip time for the scheduler is determined by NPS, and thus the time duration for an instruction cycle is always NPS clock cycles.

The Round Robin scheduling algorithm has the characteristic that it fairly distributes all time slices to all MCS-channels and the CPU. This means, that the program execution time of a specific task is independent from the activity of any neighboring task or the CPU RAM access, and thus a correct estimation of the actual program execution time is very easy. However, the round-robin scheduling may waste clock cycles by scheduling MCS-channels that are not ready to execute an instruction (e.g. MCS-channel is disabled by CPU). The following scheduling modes overcome this issue.

38.19.3.2 Accelerated Scheduling

In order to improve the computational performance, the accelerated scheduling mode provides two key features. Firstly, the scheduler only selects MCS-channels that are not suspended and thus can actually execute an instruction. Secondly, the scheduler applies instruction prefetching to minimize empty cycles in the pipeline. An MCS-channel is marked as suspended due to one of the reasons:

1. An MCS-channel is executing a read or write request to an ARU connected sub module (instruction ARD, AWR, ARDI, AWRI, NARD, NARDI).
2. An MCS-channel is executing a read or write request at its bus master interface (instruction BRD, BWR, BRDI, BWRI).
3. An MCS-channel waits on a register match event (e.g. instruction WURM), in order to wait on a desired register value (e.g. trigger event from another MCS channel).
4. An MCS-channel is disabled.

In the case of instruction prefetching, the scheduler will assign an MCS-channel C_p to pipeline stage 0, which is already present in another pipeline stage. This means, that the execution of the last instruction of C_p located in the memory MEM(PC/4) is not yet finished completely, whereas PC is the current value of the program counter of MCS-channel C_p. Thus, the newly scheduled MCS-channel C_p will prefetch a successor instruction MEM(PC/4+PFO) under the assumption that there will be no branch and no memory access in the program between the instructions MEM(PC/4) and MEM(PC/4+PFO). The prefetch offset value PFO is determined by counting the number of already scheduled MCS channels C_p in the pipeline. However, if the assumption fails, the pipeline will be flushed by replacing all MCS-channel Cp of the pipeline with an empty cycle, as soon as the instruction decoder detects a branch or a memory access. All other MCS channels unequal to MCS channel C_p within are not affected by the flushing action. The flushing action is always synchronized to the last pipeline stage NPS-1.

Besides the flushing conditions mentioned above, there exist also other conditions that cause a flush of the pipeline for a specific MCS-channel. In the following all possible flushing events are summarized:

1. An MCS-channel is enabled.
2. An MCS-channel is entering a suspended state.
3. An MCS-channel is taking a conditional or unconditional branch (instruction JMP, JBS, JBC, CALL, RET, JMPI, JBSI, JBCI, CALLI).
4. An MCS-channel accessing memory for data transfer (instruction MRD, MWR, MRDI, MWRI, MRDIO, MWRIO, MWRL, MWRIL, PUSH, POP).
5. An MCS-channel is executing a read or write request at its bus master interface (instruction BRD, BWR, BRDI, BWRI).
6. An MCS-channel is modifying the trigger register (write access to CTRG or STRG) and the same channel is reading back this register (read access to CTRG or STRG) while the delay between both accesses is less than UAP+UDP+1 clock cycles.

In general, each MCS-channel can accept instruction prefetching. However, there are some cases in which an upcoming flushing of the pipeline can be easily detected by the MCS hardware due to evaluation of internal states. Therefore, it is defined that an MCS-channel accepts instruction prefetching only under the following conditions:

1. An MCS-channel is currently not in the second cycle of a two-cycle control flow instruction (instruction CALL, RET).
2. An MCS-channel is currently not in the second cycle of a three-cycle memory access instruction (instruction MWRL, MWRIL).

The accelerated scheduling mode guarantees, that the time duration of an instruction cycle varies between 1 and T+1 cycles. Hence, a single cycle instructions has an effective execution time between 1 to T+1 clock cycles, depending on the number of suspended MCS-channels and the actual instruction sequence. The worst case execution time occurs if all channels are active and the CPU also accesses the RAM. The best case occurs e.g. if only one MCS-channel is enabled and the executed program sequence has only linear code without branches and memory access.

The algorithm of the accelerated scheduling mode first, evaluates the state of all available MCS-channels as well as a CPU request to the RAMs and then it decides if a specific MCS-channel or an empty cycle is assigned to pipeline stage 0 in the next clock cycle.

NOTE

The accelerated scheduling mode treats RAM access requests from the CPU in a similar manner as MCS-channels, which means that empty cycles for RAM requests are only inserted into the pipeline if there is an active RAM request from the CPU or no other task can be scheduled.

In order to fairly trade all available MCS-channels as well as CPU RAM requests and to guarantee a worst case execution time of T+1 clock cycles, an additional task prioritization scheme is applied that dynamically prioritizes all MCS-channels and a CPU memory access depending on the history of the scheduler's decisions. The algorithm of the accelerated scheduler mode is executed every clock cycle and it works in the following manner:

1. Try to find an MCS-channel C_r with highest priority that is not suspended and not already scheduled to the pipeline stages 0 to NPS-2. If C_r is found assign C_r to pipeline stage 0 and finish scheduling for current clock cycle.
2. Otherwise, try to find an MCS-channel C_p with highest priority that is not suspended and accepts instruction prefetching. If C_p is found assign C_p to pipeline stage 0 and finish scheduling for current clock cycle.
3. Otherwise, try to find an MCS-channel C_s with highest priority that is suspended and accepts instruction prefetching. If C_s is found assign C_s to pipeline stage 0 and finish scheduling for current clock cycle.
4. Otherwise, assign an empty cycle to pipeline stage 0 and finish scheduling for current clock cycle.

The underlying task prioritization scheme tracks the history of the scheduled MCS-channels in a list consisting of T+1 items. The list is initialized with all MCS-channels followed by a reserved time slot for the CPU RAM access. The position of an MCS-channel within this list implicitly defines the priority, while the back of this list holds the MCS-channel with highest priority. Whenever the

scheduling algorithm described above has found an MCS-channel C_r or C_p to be scheduled in the next clock cycle, it removes this item from the list and put it to the front of the list. In order to fairly prioritize all MCS-channels, the algorithm also moves the item at the back of the list to the second position in the list, after the inserted scheduled front item. Since the list always contains all possible MCS-channels and with each clock cycles each nonscheduled item is moved at least one position towards the end of list, it is obvious that each MCS-channel will have the highest priority not later than T+1 clock cycles.

Figure 38.106 shows a timing example of the accelerated scheduling with NPS=7 pipeline stages.

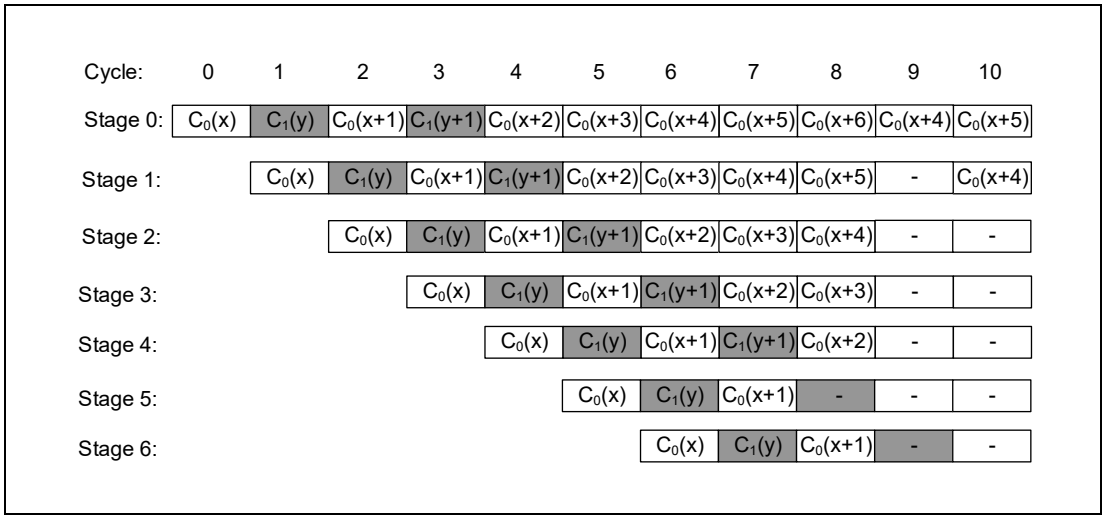


Figure 38.106 Timing of Accelerated Scheduling

The example assumes that initially MCS-channels 0 and 1 are enabled and the program for each MCS-channel is located in the RAM as follows:

Table 38.242 MCS Code example for Accelerated Scheduling

MCS-Channel 0		MCS-Channel 1	
Memory Location	**Instruction**	**Memory Location**	**Instruction**
x+0	ADDL R0, 7	x+0	WURM STRG, R1, $FFFF_H$
x+1	JBC STA, Z, x+4	x+1	ADDL R2, 5
x+2	MOVL R2, 5	x+2	ADD R3, R2

The example assumes that initially MCS-channels 0 and 1 are enabled and the program for each MCS-channel is located in the RAM as shown in **Table 38.242**. Since both channels are ready to run, the scheduler fairly selects the channels in an alternating order, as it can be obtained in stage 0 at the clock cycles before cycle 5. Since MCS-channel 1 is entering suspended state (to wait on a trigger bit) at cycle 7 in stage 6 with the instruction of memory location y, the scheduler will only select MCS-channel 0 in the following by applying instruction prefetching. Moreover, entering the suspended state in channel 1 also flushes the remaining channels 1 out of the pipeline.

NOTE

The scheduler applies instruction prefetching during the whole sequence, due to the fact that the number of enabled channels is always less than the available number of pipeline stages NPS.

The actual state of pipeline in cycle 9 and 10 depends on conditional branch instruction of memory location x + 3 (cycle 8 stage 6). If the branch is not taken, the linear code execution of MCS-channel 0 is continued. However, if the branch to memory location x+4 is taken, as shown in the Figure, the scheduler will fetch the instruction C_0 (x+4) in cycle 9 at stage 0 and flush the stages 1 to NPS-1.

NOTE

The flushing of the pipeline only concerns the prefetched instructions of the MCS-channel that is currently executed in the last stage. If pipeline stage 1 of cycle 9 would belong to another channel than 0, only the stages greater than 2 would have been flushed.

38.19.3.3 Single Prioritization Scheduling

The Single Prioritization Scheduling mode is an extended variant of the Accelerated Scheduling mode, which additionally applies a task prioritization of a single MCS-channel. In this mode, the bit field SCD_CH of register MCS[i]_CTRL_STAT is used to identify a dedicated MCS-channel that is always preferred during scheduling. This means, that the scheduler will assign preferred MCS-channel SCD_CH to pipeline stage 0, as long as this channel is not suspended. If the preferred MCS-channel is entering its suspended state, the scheduling algorithm switches to the accelerated scheduling as previously described in **Section 38.19.3.2, Accelerated Scheduling**. Whenever the MCS-channel SCD_CH is resuming from its suspended state, the scheduler switches back and assign the channel SCD_CH to pipeline stage 0 until the next suspension event occurs. If the bit field SCD_CH contains the value T or higher, the task scheduler will always prioritize CPU access to the RAM. This means, whenever the task scheduler detects that the CPU wants to access an MCS.RAM, the scheduler will assign an empty cycle into pipeline stage 0. If the CPU does not access the RAM any more, it switches back to the accelerated mode, as described previously in **Section 38.19.3.2, Accelerated Scheduling**.

In consequence, the Single Prioritization Scheduling mode cannot guarantee a maximum time duration of an instruction cycle for the overall execution of all MCS-channels, since it strongly depends on the activity of the prioritized MCS-channel SCD_CH. However, the Single Prioritization Scheduling mode provides the fastest possible execution for MCS-channel SCD_CH. Moreover, during the time spawn, in which the prioritized MCS-channel SCD_CH is suspended, this mode guarantees a duration of 1 to T+1 clock cycles of an instruction cycle for all non-prioritized channels.

NOTE

If an MCS program is being executed in Single Prioritization Scheduling mode, the CPU cannot access the MCS RAM from which the prioritized program code is fetched. In this case the CPU RAM access will be blocked until the prioritized MCS channel is entering its suspended state. If data needs to be exchanged between CPU and a prioritized MCS channel, either the general purpose registers or the opposite RAM module (if USR = 1) should be used.

38.19.3.4 Multiple Prioritization Scheduling

The Multiple Prioritization Scheduling mode is an extended variant of the Accelerated Scheduling mode, which additionally applies a task prioritization for multiple MCS-channels. In this mode, the bit field SCD_CH of register MCS[i]_CTRL_STAT is used to identify a set of dedicated MCS-channels, which are always preferred during scheduling. The identifiers of the prioritized MCS-channels are in the range [0; SCD_CH] and the non-prioritized channels are in the range [SCD_CH +1; T.1]. The individual priority for the set of prioritized MCS-channels is applied in descending order, which means that MCS-channel 0 has the highest priority, followed MCS-channel 1, which has the second highest priority, and so on. The non-prioritized MCS-channels do not have any priority. A value of T.1 or higher for the bit field SCD_CH means that all T MCS-channels are prioritized MCS-channels. With each clock cycle, the Multiple Prioritization Scheduling mode will assign the non-suspended MCS-channel with the highest priority from the set of prioritized MCS-channels to pipeline stage 0, as long as there are non-suspended prioritized MCS-channels available. If all prioritized MCS-channels are suspended, the scheduling algorithm switches to the accelerated scheduling as previously described in **Section 38.19.3.2, Accelerated Scheduling** and it schedules the non-prioritized channels. Whenever a prioritized MCS-channel is resuming from its suspended state, the scheduler switches back and applies the described prioritization scheme until the next suspension event of occurs.

In consequence, the Multiple Prioritization Scheduling mode cannot guarantee a maximum time duration of an instruction cycle for the overall execution of all MCS-channels, since it strongly depends on the activity of the prioritized MCS-channels. However, the Multiple Prioritization Scheduling mode provides the fastest possible execution for prioritized MCS-channels. Moreover, during the time spawn, in which all prioritized MCS-channels are suspended, this mode guarantees a duration of 1 to T+1 clock cycles of an instruction cycle for all non-prioritized channels.

NOTE

If an MCS program is being executed in Multiple Prioritization Scheduling mode, the CPU cannot access the MCS RAM from which the prioritized program code is fetched. In this case the CPU RAM access will be blocked until the last prioritized MCS channel is entering its suspended state. If data needs to be exchanged between CPU and a prioritized MCS channel, either the general purpose registers or the opposite RAM module (if USR = 1) should be used.

38.19.4 Memory Organization

The MCS module supports a memory layout of up to 2^(RAW+USR) memory locations each RDW bit wide leading to a maximum byte wise address range from 0 to 2^(RAW+USR+2) –1.

If two RAM ports are used (USR = 1) the entire address space of the MCS is divided into two seamless memory pages.

Further, if the GTM provides a memory configuration sub module (MCFG), memory page 0 begins from (byte wise) address 0 and ranges to address MP0-4 and memory page 1 ranges from MP0 to MP1-4, while MP0 and MP1 are configuration parameters provided by MCFG. If USR is 1 but there is no MCFG module available, the actual parameters MP0 and MP1 can be found in **Section 38.28, GTM Device 358**. The base address for accessing the memory via AEI can also be found in **Section 38.28, GTM Device 358**.

The RAM priority decoder of the MCS will always handle a RAM access from an MCS channel with a higher priority compared to a RAM access from AEI.

However, if a set of active MCS channels are only accessing one common RAM port, the MCS will grant any AEI accesses to the other RAM port in parallel to the related RAM accesses of the running MCS channels, which means that AEI may get the full bandwidth to a dedicated RAM.

Basically, the actual access time to the RAMs via AEI depends on the actual scheduling mode and the activity of tasks. In the modes Round Robin Scheduling and Accelerated Scheduling the scheduler guarantees a maximum write access time of T + 4 clock cycles and a maximum read access time of T + 6 clock cycles. In the scheduling modes Single Prioritization Scheduling and Multiple Prioritization Scheduling, the scheduler cannot guarantee a maximum access time for AEI RAM access.

Depending on the silicon vendor configuration, the connected RAM pages are initialized with zeros in the case of an MCS module reset.

If an ECC Error occurs while an MCS-channel reads data from a memory module, the corresponding MCS-channel is disabled and the ERR bit in register STA is raised.

If the GTM sub module CCM provides several so called address range protectors (ARPs), some code and data sections of the MCS RAM can be write protected. If an MCS channels x writes to such a protected memory region, the MCS channel x is halted, the ERR bit in register STA is set and the bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT is updated.

38.19.5 AEI Bus Master Interface

The MCS module provides an AEI bus master interface, which enables to communicate with externally connected modules. The data width of this interface is BDW bit and the address width is BAW bit leading to a maximum byte wise address ranging from 0 to 2^{BAW+2}-1. The bus master interface is shared among all available MCS channels meaning that each MCS channel may initiate a read or write access on the bus but only one channel can be served at a specific point in time.

However, the AEI bus master interface guarantees, that a bus access is always completed within two instruction cycles and bus access of different MCS channels do not modify the latency of each other. The only exceptions are bus accesses to RAM modules (e.g. accessing memory location in a DPLL RAM or FIFO RAM). AEI bus master accesses to RAM modules cannot be completed within a single clock cycle and thus additional wait cycles have to be inserted into the bus protocol leading to the fact that the MCS channel that is accessing the RAM is entering a suspended state. Moreover, if an MCS channel is accessing a RAM module, the latency of a bus access in another MCS channel can also be modified even if the neighboring channel is accessing only a configuration register.

The AEI bus master interface of an MCS module is connected to AEI slave interface GTM-IP in order to control the sub modules of the GTM within MCS. However, it is not possible to access the entire GTM by a single MCS module. The n-th MCS instance can only access the GTM sub modules that are located within the n-th cluster of the GTM.

Additionally, the address map for accessing GTM with the AEI bus master interface of an MCS differs from the address map for an externally connected CPU that is using the GTM's AEI slave interface. The address map for accessing GTM with the AEI bus master interface of an MCS can be found in **Section 38.28, GTM Device 358**.

Since the sub modules of the GTM can be accessed by the CPU and the AEI bus master of an MCS, the GTM-IP provides an additional arbitration scheme to manage parallel accesses from both master interfaces. If CPU and an MCS want to access a GTM sub module of the same cluster, the arbiter will grant the access to the MCS. However, if the CPU and all the MCS instances want to access GTM sub modules of different clusters, the accesses can be executed in parallel.

The AEI bus master interface can be controlled by the MCS instructions BRD, BRDI, BWR, and BWRI. These instructions are described in **Section 38.19.8, Instruction Set**.

38.19.6 ADC Interface

The clusters of the GTM can provide a dedicated interface for the connection of up to 32 external Analog-Digital-Converter (ADC) channels, which can be mapped arbitrarily to physical instances of single- or multi-channel ADCs. See **Section 38.28, GTM Device 358** for details about the availability of ADC interfaces. An ADC Interface is directly mapped into the address map of the AEI bus master interface of the current cluster's MCS meaning that the available AEI bus master instructions (BRD and BRDI as described in **Section 38.19.8, Instruction Set**) are used to control the connected ADCs. Since the control of the connected ADCs is silicon vendor specific, the GTM specification does not provide a complete specification for controlling connected ADCs. However, to ensure software compatibility at least for the basic features of an ADC, the functionality described in the following are common to all silicon vendors.

38.19.6.1 Basic ADC Functions

The address map of the AEI bus master interface reserves two unique address items for each ADC channel. The address items can be referred by the labels ADC_CH[y]_DATA and ADC_CH[y]_STA for the channel y in the range from 0 to 31. The actual address for these labels can be found in **Section 38.28, GTM Device 358**. The MCS can read from address ADC_CH[y]_DATA in order to get the conversion result of the ADC that is connected to ADC channel y. The conversion result is represented as a signed 24 bit value and it is stored in the register A (A ∈GREG) as referred by the corresponding MCS instruction BRD or BRDI. Additionally, each read access to ADC_CH[y]_DATA triggers the ADC that is connected to channel y. Any read access to ADC_CH[y]_DATA also provides 8 status bits that are stored in register MHB. The bit MHB [7] has always the mnemonic ADC_ACK and the bit MHB [6] has always the mnemonic ADC_NEW_DATA. If bit ADC_ACK is set the result of the data conversion (register A) and the corresponding status bits (bits MHB [6:0]) are validated. If ADC_NEW_DATA is set the current conversion result is new and has never been read by a previous bus read access. The meaning of the bits MHB [5:0] are vendor specific.

Otherwise, if ADC_ACK is cleared the read data is invalid and the MHB [4:0] indicate the channel identifier (with MHB [4:0] ≠ y) that is currently processed by the ADC. A write access to ADC_CH[y]_DATA has no functionality and is always ignored. The MCS can read from address ADC_CH[y]_STA to get additional 31-bit wide vendor specific status information of ADC channel y. The lower 24 bits of the status information is stored in register A (A ∈ GREG) as referred by the corresponding MCS instruction BRD or BRDI. The upper 7 bit of the status information is stored in register MHB [6:0]. The bit MHB [7] has always the mnemonic ADC_ACK. If bit ADC_ACK is set the result of status information in register A and bits MHB [6:0] are validated. Otherwise, if ADC_ACK is cleared the status information is invalid.

A write access to ADC_CH[y]_STA has no functionality and is always ignored. Any read or write access to a register ADC_CH[y]_STA or ADC_CH[y]_DATA updates the AEI status signal that is evaluated in the sub module CCM. The following status information is defined for the AEI status values:

00: No error occurred

01: Optional information register not implemented (only register ADC_CH[y]_STA)

10: Illegal ADC access (e.g. ADC not enabled)

11: Unsupported address (ADC channel y not available)

NOTE

If the received status AEI is unequal to "00" ADC_NEW_DATA is always set and ADC_ACK is always cleared.

CAUTION

ADC_CH[y]_STA has no information and returns the same value as ADC_CH[y]_DATA when read.
Bit 31 of ADC_CH[y]_DATA is always read as 1, and bits 29 to 24 of ADC_CH[y]_DATA are always read as 0.

38.19.7 AXI Bus Master Interface

The GTM provides an AXI bus master interface which allows to initiate bus transactions externally to the GTM. Details about the connection of this bus interface can be found in the specification of the microcontroller. The GTM integrates a central AXI transaction generator module that is controlling and arbitrating the entire AXI master communication of the GTM. This module provides up to 24 communication slots to a cluster of the GTM which implements an MCS instance. In order to control the common AXI master functionality and the functionality of the dedicated AXI communication slots the AEI bus master interface of the MCS provides direct access to the central AXI transaction generator module. This means that the AXI bus master interface is controlled with the standard AEI bus master instructions (instruction BRD, BRDI, BWR, and BWRI).

A detailed explanation about how to use AXI bus master interface can be found in **Section 38.6, AXI Master**.

The actual address map of the AXI bus master interface can be found in **Section 38.28, GTM Device 358**.

38.19.8 Instruction Set

This section describes the entire instruction set of the MCS sub module. First, a brief overview over all available instructions is given and a detailed description of each instruction can be found in **Section 38.19.8, (1) MOVL Instruction** and the following sections.

In general, each instruction is RDW bit wide but the duration of each instruction varies between several instruction cycles. As already described in **Section 38.19.3, Scheduling**, the number of required clock cycles for an instruction cycle can be fixed or variable, depending on the selected scheduling mode. In the case of the Round Robin Scheduling, the duration is fixed with T+1 clock cycles, in the case of the Accelerated Scheduling the duration is variable in the range between 1 and T+1 clock cycles, and in all other Scheduling modes the duration is also variable and may even be more than T+1 clock cycles, depending on the application.

Before the available instructions are described, some commonly used terms, abbreviations and expressions are introduced:

Table 38.243 Instruction set (1/2)

Instruction	Description
OREG	The operation register set OREG = {R0 to R7} ∪ { STA, ACB, CTRG, STRG, TBU_TS0, TBU_TS1, TBU_TS2, MHB} include all MCS accessible internal channel specific GPRs {R0 to R7} and the sub set {STA, ACB, CTRG, STRG, TBU_TS0, TBU_TS1, TDU_TS2, MHB} of SFRs.
XOREG	The extended operation register set XOREG = OREG∪{RS0 to RS7} ∪ {GMI0, GMI1, DSTA, DSTAX} extends the operation registers set OREG by the GPRs of the succeeding MCS channel {RS0 to RS7} and the SFRs {GMI0, GMI1, DSTA, DSTAX, AXIMI}.
WXREG	The extended wait instruction operation register set WXREG = OREG ∪ {GMI0, GMI1, DSTA, DSTAX} extends the operation registers set OREG by the SFRs {GMI0, GMI1, DSTA, DSTAX, AXIMI}.
AREG	The ARU register set AREG = {R0 to R7, ZERO} includes all the registers that can be written by incoming ARU transfers (ARD, ARDI, NARD, and NARDI instructions). These registers include all eight general purpose registers. The dummy register ZERO may be used to discard an incoming 24 bit ARU word.
GREG	The general purpose register set GREG = {R0 to R7} includes all the channel specific GPRs without GPRs of neighboring channels.
BAREG	The base address register set BAREG = OREG∪{RS0 to RS7} extends the register set OREG by the GPRs of neighboring channels. If the extended operation register set XOREG is disabled (bit EN_XOREG of register MCS[i]_CTRL_STAT is cleared) the sets XOREG, WXREG, and BAREG only contains the operation register set OREG. In the following, the register sets OREG, XOREG, GREG, WXREG, BAREG and AREG are referred by the instructions. Typically, an operation announces W data bits. Whenever, a register of a register set implements less than W bits, it is assumed that these register bits only define the LSBs of an operation. The missing MSBs are always read and written as zeros.
WLIT	The set WLIT = {0,1, ..., 2^{W} –1} is a W bit wide literal value used for encoding immediate operands.
ALIT	The set ALIT = {0,1, ..., $2^{RAW+USR}$ –1} is a RAW + USR bit wide literal value used for encoding memory addresses.
AOLIT	The set AOLIT = {–$2^{RAW+USR}$, ..., –1,0,1, ..., $2^{RAW+USR}$ –1} is a RAW + USR bit wide literal value used for encoding relative memory address offsets.
ARDLIT	The set ARDLIT = {0,1, ..., 2^{9} –1} is a 9 bit literal used for ARU read addresses.
AWRLIT	The set AWRLIT = {0,1, ..., 23} is used as ARU write indexes, selecting one of the 24 ARU write address.
BALIT	The set BALIT = {0,1, ..., 2^{BAW} –1} is a BAW bit wide literal used for encoding bus master addresses.
SFTLIT	The set SFTLIT = {0,1, ..., W} is used as literal value for shift instructions.
BWSLIT	The set BWSLIT = {1, ..., W} is used as literal value for multiplication instructions.
BITLIT	The set BITLIT = {0,1, ..., 15} is a 4 bit literal used for bit indexing.
XBITLIT	The set XBITLIT = {0,1, ..., W–1} is a literal used for bit indexing of register bits.
MSKLIT	The set MSKLIT = {0,1, ..., 2^{16} –1} is a 16 bit literal used for bit-masking.
BIT SELECTION	The expression VAR[i] represents the i-th bit of a variable VAR
BIT RANGE SELECTION	The expression VAR[m:n] represents the bit slice of variable VAR that is ranging from bit n to bit m.

Table 38.243 Instruction set (2/2)

Instruction	Description
MEMORY ADDRESSING	The expression MEM(X) represents the RDW bit wide value at location x (x 2 ALIT) of the memory. The expression MEM(x)[m:n] represents the bit slice ranging from bit n to m of the RDW bit wide word at memory location x.
ARU ADDRESSING	In the case of ARU reading, the expression ARU(x) represents the 2*W+5 bit wide ARU word of ARU channel at read address x (x 2 ARDLIT). In the case of ARU writing, the expression ARU(x) represents a 2*W+5 bit wide ARU word that is written to an ARU channel indexed by the index x (x 2 AWRLIT). The index x selects a single ARU write channel from the pool of the MCS sub module's allocated ARU write channels. An MCS sub module has 24 dedicated ARU write channels, indexed by values 0 to 23. The expression ARU(x)[m:n] represents the bit slice ranging from bit n to m of the 2*W+5 bit wide ARU word.
BUS MASTER ADDRESSING	In the case of reading/writing from the bus master interface, the expression BUS(x) represents the BDW bit wide data word that is read/written at address x (x 2 BALIT). The expression BUS(x)[m:n] represents the bit slice ranging from bit n to m of the BDW bit wide data word at the bus.

The individual instructions are decoded by evaluating the bits '0' and '1' at its expected positions, as mentioned in field encoding of the individual instructions below, whereas the leftmost bit indicates the most significant bit and the rightmost bit indicated the lowest significant bit. If the instruction decoder detects an invalid combination of these bits, the corresponding MCS-channel is disabled and the ERR bit in the register STA is set. Bit positions marked as '–' are not relevant for the corresponding instruction and they are ignored completely during instruction decoding.

The bit position 'a_i', 'b_i', and 'c_i' are used for binary encoding of the instruction arguments A, B, and C. The actual decimal value of an argument A, B, and C can be obtained by adding up the products $a_i * 2^i$, $b_i * 2^i$, and $c_i * 2^i$ respectively.

Each instruction can also set the ERR bit of register STA and stop the program execution, if a register write protection of an associated MCS channel is activated by the register MCS[i]_REG_PROT. This behavior is not explicitly mentioned in the instruction descriptions below. If an error occurs due to a write access to a protected register it is ensured that the protected register is not overwritten. However, it is not ensured that other operations (e.g. updating of the PC) of the bad instruction are executed.

Table 38.244 summarizes the entire instruction set of the MCS and **Table 38.249** shows the encoding of the individual instructions.

Table 38.244 Instruction Set Summary (1/4)

Class	Mnemonic	Operation	Instruction cycles	Synopsis
Data transfer	MOVL A, C	A ← C	1	Move Literal, A in OREG, C in WLIT
	MOV A, B	A ← B	1	Move, A in XOREG, B in XOREG
	MRD A, C	A ← MEM(C)[W-1:0]; MHB ← MEM(C)[RDW-1:W]	2*1	Memory Read, A in OREG, C in ALIT
	MWR A, C	MEM(C)[W-1:0] ← A; MEM(C)[RDW-1:W] ← MHB	2*1	Memory Write, A in OREG, C in ALIT
	MRDI A, B [, C]	A ← MEM(B[RAW+USR+1:2]+C)[W-1:0]; MHB ←MEM(B[RAW+USR+1:2]+C)[RDW-1:W]	2*1	Memory Read Indirect, A in OREG, B in OREG, C in AOLIT (default C=0)
	MWRI A, B [, C]	MEM(B[RAW+USR+1:2]+C)[W-1:0] ← A; MEM(B[RAW+USR+1:2]+C)[RDW-1:W] ← MHB	2*1	Memory Write Indirect, A in OREG, B in OREG, C in AOLIT (default C=0)
	MRDIO A, B	A ← MEM(B[RAW+USR+1:2] + R5[RAW+USR+1:2])[W-1:0]; MHB ← MEM(B[RAW+USR+1:2] + R5[RAW+USR+1:2])[RDW-1:W]	2*1	Memory Read Indirect with Offset, A in XOREG, B in BAREG
	MWRIO A, B	MEM(B[RAW+USR+1:2] + R5[RAW+USR+1:2])[W-1:0] ← A; MEM(B[RAW+USR+1:2] + R5[RAW+USR+1:2])[RDW-1:W] ← MHB	2*1	Memory Write Indirect with Offset, A in XOREG, B in BAREG
Data transfer	POP A	A ← MEM(R7[RAW+USR+1:2]); MHB ← MEM(R7[RAW+USR+1:2])[RDW-1:W] R7 ← R7 - 4	2*1	Pop from stack, A in OREG \ {R7}
	PUSH A	R7 ← R7 + 4 MEM(R7[RAW+USR+1:2])[W-1:0] ← A MEM(R7[RAW+USR+1:2])[RDW-1:W] ← MHB	2*1	Push to stack, A in OREG \ {R7}
	MWRL A, C	MEM(C)[W-1:0] ← A	3*2	Memory Write Literal, A in OREG, C in ALIT
	MWRIL A, B	MEM(B[RAW+USR+1:2])[W-1:0] ← A	3*2	Memory Write Indirect Literal, A in OREG, B in OREG
ARU Transfer	ARD A, B, C	A ← ARU(C)[W-1:0] B ← ARU(C)[2*W-1:W] ACB ← ARU(C)[4+2*W:2*W]	≥1	Blocking ARU Read, A in AREG, B in AREG, C in ARDLIT
	AWR A, B, C	ARU(C)[W-1:0] ← A ARU(C)[2*W:W] ← B ARU(C)[4+2*W:2*W] ← ACB	≥1	Blocking ARU Write, A in OREG, B in OREG, C in AWRLIT
	ARDI A, B	A ← ARU(R6[8:0])[W-1:0] B ← ARU(R6[8:0])[2*W-1:W] ACB ← ARU(R6[8:0])[4+2*W:2*W]	≥1	Blocking ARU Read Indirect, A in AREG, B in AREG
	AWRI A, B	ARU(R6[4:0])[W-1:0] ← A ARU(R6[4:0])[2*W-1:W] ← B ARU(R6[4:0])[4+2*W:2*W] ← ACB	≥1	Blocking ARU Write Indirect, A in OREG, B in OREG
	NARD A, B, C	A ← ARU(C)[W-1:0] B ← ARU(C)[2*W:W] ACB ← ARU(C)[4+2*W:2*W]	≥1*7	Non-Blocking ARU Read, A in AREG, B in AREG
	NARDI A, B	A ← ARU(R6[8:0])[W-1:0] B ← ARU(R6[8:0])[2*W-1:W] ACB ← ARU(R6[8:0])[4+2*W:2*W]	≥1*7	Non-Blocking ARU Read Indirect, A in AREG, B in AREG

Table 38.244 Instruction Set Summary (2/4)

Class	Mnemonic	Operation	Instruction cycles	Synopsis
Bus Master	BRD A, C	A ← BUS(C)[W-1:0] MHB ← BUS(C)[BDW-1:W]	≥1*8	Bus Master Read, A in GREG, C in BALIT
	BWR A, C	BUS(C)[W-1:0] ← A BUS(C)[BDW-1:W] ← MHB	≥1*8	Bus Master Write, A in GREG, B C in BALIT
	BRDI A, B	A ← BUS(B[BAW+1:2])[W-1:0] MHB ← BUS(B[BAW+1:2])[BDW-1:W]	≥1*8	Bus Master Read Indirect, A in GREG, B in GREG
	BWRI A, B	BUS(B[BAW+1:2])[W-1:0] ← A BUS(B[BAW+1:2])[BDW-1:W] ← MHB	≥1*8	Bus Master Write Indirect, A in GREG, B in GREG
Arith. / Logic	ADDL A, C	A ← A + C	1	Add Literal, A in OREG, C in W LIT
	ADD A, B	A ← A + B	1	Add, A in XOREG, B in XOREG
	ADDC A, B	A ← A + B + CY	1	Add with carry, A in XOREG, B in XOREG
	SUBL A, C	A ← A - C	1	Subtract Literal, A in OREG, C in W LIT
	SUB A, B	A ← A - B	1	Subtract, A in XOREG, B in XOREG
	SUBC A, B	A ← A - B - CY	1	Subtract with carry, A in XOREG, B in XOREG
	NEG A, B	A ← – B	1	Negate, A in XOREG, B in XOREG
	ANDL A, C	A ← A AND C	1	AND Litral, A in OREG, C in W LIT
	AND A, B	A ← A AND B	1	AND, A in XOREG, B in XOREG
	ORL A, C	A ← A OR C	1	OR Literal, A in OREG, C in W LIT
	OR A, B	A ← A OR B	1	OR, A in XOREG, B in XOREG
	XORL A, C	A ← A XOR C	1	XOR Literal, A in OREG, C in W LIT
	XOR A, B	A ← A XOR B	1	XOR, A in XOREG, B in XOREG
	SETB A, B	A[B[4:0]] ? 1	1	Set Bit, A in XOREG, B in XOREG
	CLRB A, B	A[B[4:0]] ? 0	1	Clear Bit, A in XOREG, B in XOREG
	XCHB A, B	A[B[4:0]] ? CY	1	Exchange Bit with CY, A in XOREG, B in XOREG
	SHR A, C	A ← A >> C	1	Shift Right, A in XOREG, C in SFTLIT
	SHL A, C	A ← A << C	1	Shift Left, A in XOREG, C in SFTLIT
	ASRU A, B	A ← A >> B	1	Shift Right, A in XOREG, B in XOREG
	ASRS A, B	A ← A >> B	1	Shift Right, A in XOREG, B in XOREG
	ASL A, B	A ← A << B	1	Shift Left, A in XOREG, B in XOREG

Table 38.244 Instruction Set Summary (3/4)

Class	Mnemonic	Operation	Instruction cycles	Synopsis
Arith. / Logic	MULU A, B[, C]	[[R4,] A] ← A[(C-1):0] * B[(C-1):0]	1	Multiply Unsigned, A in XOREG, B in XOREG, C in BW SLIT (default C=W)
	MULS A, B[, C]	[[R4,] A] ← A[(C-1):0] * B[(C-1):0]	1	Multiply Signed, A in XOREG, B in XOREG, C in BW SLIT (default C=W)
	DIVU A, B[, C]	R4← A[(C-1):0] - B[(C-1):0] * ⌊A[(C-1):0] / B[(C-1):0]⌋; A ← ⌊A[(C-1):0] / B[(C-1):0]⌋	C*9	Divide Unsigned, A in XOREG, B in XOREG, C in BW SLIT (default C=W)
	DIVS A, B[, C]	R4 ← A[(C-1):0] - B[(C-1):0] * ⌊A[(C-1):0] / B[(C-1):0]⌋; A ← ⌊A[(C-1):0] / B[(C-1):0]⌋	C+4*10	Divide Signed, A in XOREG, B in XOREG, C in BW SLIT (default C=W)
	MINU A, B	A ← MIN(A, B)	1	Minimum Unsigned, A in XOREG, B in XOREG
	MINS A, B	A ← MIN(A, B)	1	Minimum Signed, A in XOREG, B in XOREG
	MAXU A, B	A ← MAX(A, B)	1	Maximum Unsigned, A in XOREG, B in XOREG
	MAXS A, B	A ← MAX(A, B)	1	Maximum Signed, A in XOREG, B in XOREG
Test	ATUL A, C	A < C ⇔ CY is set A = C ⇔ Z is set	1	Arithmetic Test Unsigned Literal, A in OREG, C in W LIT
	ATU A, B	A < B ⇔ CY is set A = C ⇔ Z is set	1	Arithmetic Test Unsigned, A in XOREG, B in XOREG
	ATSL A, C	A < C ⇔ CY is set A = C ⇔ Z is set	1	Arithmetic Test Signed Literal, A in OREG, C in W LIT
	ATS A, B	A < B ⇔ CY is set A = C ⇔ Z is set	1	Arithmetic Test Signed, A in XOREG, B in XOREG
	BTL A, C	A AND C	1	Bit Test Literal, A in OREG, C in W LIT
	BT A, B	A AND B	1	Bit Test, A in XOREG, B in XOREG
Control Flow	JMP C	PC ← C << 2	1*3	Unconditional Jump, C in ALIT
	JBS A, B, C	PC ← C << 2 if A[B] is set	1*4	Jump if Bit Set, A in OREG, B in BITLIT, C in ALIT
	JBC A, B, C	PC ← C << 2 if A[B] is clear	1*4	Jump if Bit Cleared, A in OREG, B in BITLIT, C in ALIT
	CALL C	R7 ← R7 + 4 MEM(R7[RAW+USR+1:2])[RAW+USR+1:0] ← PC+ 4 PC ← C << 2	2*5	Call Subroutine, C in ALIT
	RET	PC ← MEM(R7[RAW+USR+1:2])[RAW+USR+1:0] R7 ← R7 - 4	2*5	Return from Subroutine
	JMPI	PC ← R6[RAW+USR+1:2] << 2	1*3	Unconditional Jump Indirect
	JBSI A, B	PC ← R6[RAW+USR+1:2] << 2 if A[B] is set	1*4	Jump if Bit Set Indirect, A in XOREG, B in XBITLIT
	JBCI A, B	PC ← R6[RAW+USR+1:2] << 2 if A[B] is clear	1*4	Jump if Bit Clear Indirect, A in XOREG, B in XBITLIT
	CALLI	R7 ← R7 + 4 MEM(R7[RAW+USR+1:2])[RAW+USR+1:0] ← PC+ 4 PC ← R6[RAW+USR+1:2] << 2	2*5	Call Subroutine Indirect

Table 38.244 Instruction Set Summary (4/4)

Class	Mnemonic	Operation	Instruction cycles	Synopsis
Others	WURM A, B, C	wait until A = (B AND ((0xFF << 16) + C))	≥1*6	Wait Until Register Match, A in OREG, B in OREG, C in MSKLIT
	WURMX A, B	wait until A = (B AND R6)	≥1*6	Wait Until Register Match, A in OREG, B in WXREG
	WURCX A, B	wait until A ≠ (B AND R6)	≥1*6	Wait Until Register Change, A in OREG, B in WXREG
	WUCE A, B	wait until cyclic event comparison matches	≥1*6	Wait Until Cyclic Event, A in OREG, B in OREG
	NOP		1	No Operation

Note 1. Not faster than 1+NPS clock cycles due to pipeline flushing.
Note 2. Not faster than 1+2*NPS clock cycles due to pipeline flushing.
Note 3. Not faster than NPS clock cycles due to pipeline flushing.
Note 4. If the jump is executed, it is not faster than NPS clock cycles due to pipeline flushing.
Note 5. Not faster than 2*NPS clock cycles due to pipeline flushing.
Note 6. If the MCS is configured in Single Prioritization or Multiple Prioritization Scheduling Mode the worst case latency for reactivating a prioritized MCS-channel is 2+NPS clock cycles.
Note 7. Always faster than one ARU round trip cycle.
Note 8. Suspends current MCS-channel if addressed slave inserts at least one wait cycle otherwise 1 instruction cycle
Note 9. Not faster than C+NPS-1 clock cycles due to pipeline flushing.
Note 10. Not faster than C+3+NPS clock cycles due to pipeline flushing.

(1) MOVL Instruction

Syntax: MOVL A, C

Encoding: $0001a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A ← C

Status: Z

Duration: 1 instruction cycle

Description: Transfer literal value C (C ∈ WLIT) to register A (A∈ OREG).
The zero bit Z of status register STA is set, if the transferred value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(2) MOV Instruction

Syntax: MOV A, B

Encoding: $1010a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0000\text{-}a_4\text{-}b_4$ ----------

Operation: A ← B

Status: Z

Duration: 1 instruction cycle

Description: Transfer register B (B ∈ OREG) to register A (A ∈ OREG).
The zero bit Z of status register STA is set, if the transferred value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(3) MRD Instruction

Syntax: MRD A, C

Encoding: $1010a_3a_2a_1a_0$----$0001c_{15}c_{14}c_{13}c_{12}c_{11}c_{10}c_9c_8c_7c_6c_5c_4c_3c_2$--

Operation: A ← MEM(C)[W-1:0];
MHB ← MEM(C)[RDW-1:W]

Status: Z

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer the lower W bits of memory content at location C (C ∈ ALIT) to register A (A ∈ OREG).
The upper RDW-W bits of the memory content at location C are transferred to the MHB register.
The zero bit Z of status register STA is set, if the lower W bits of the transferred value are zero, otherwise the zero bit is cleared.
If the MHB register is selected as destination register A (A ∈ OREG), the bits 0 to RDW-W-1 of the referred memory location are transferred to MHB.
The program counter PC is incremented by the value 4.

(4) MWR Instruction

Syntax: MWR A, C

Encoding: $1010a_3a_2a_1a_0$----$0010c_{15}c_{14}c_{13}c_{12}c_{11}c_{10}c_9c_8c_7c_6c_5c_4c_3c_2$--

Operation: MEM(C)[W-1:0] ← A;
MEM(C)[RDW-1:W] ← MHB

Status: —

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer W bit value of register A (A ∈ OREG) together with the MHB register to the memory at location C (C ∈ ALIT).
The W bit value of register A is stored in the lower significant bits (bit 0 to W-1) of the memory location.
The MHB register is stored in bits W to RDW-W-1 of the referred memory location.
The program counter PC is incremented by the value 4.

(5) MWRL Instruction

Syntax: MWRL A, C

Encoding: $1010a_3a_2a_1a_0$----$0111c_{15}c_{14}c_{13}c_{12}c_{11}c_{10}c_9c_8c_7c_6c_5c_4c_3c_2$--

Operation: MEM(C)[W-1:0] ← A

Status: —

Duration: 3 instruction cycles but not faster than 1+2*NPS clock cycles due to pipeline flushing.

Description: Transfer W bit value of register A (A ∈ OREG) to memory at location C (C ∈ ALIT).
The W bit value of register A is stored in the lower significant bits (bit 0 to W-1) of the memory location and the bits W to RDW-W are left unchanged.
The program counter PC is incremented by the value 4.
It should be noted that this operation is not an atomic instruction.

(6) MRDI Instruction

Syntax: MRDI A, B [, C]

Encoding: 1010a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0011c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 --

Operation: A←MEM((B[RAW+USR+1:2] + C) AND MASK)[W–1:0]
MHB←MEM((B[RAW+USR+1:2] + C) AND MASK)[RDW–1:W]

Status: Z

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer the bits 0 to W–1 of a memory location to register A (A ∈ OREG) using indirect addressing.
The upper RDW–W bits of this memory location are transferred to MHB register.
The memory location where to read from depends on register B (B ∈ OREG) and literal C (C ∈ AOLIT) and it is defined as ((B[RAW+USR+1:2] + C) AND MASK), where AND is a bitwise AND conjunction and MASK is a bit mask value 2^(RAW+USR)-1.
If the optional operand C is not available in the assembler syntax, the MCS assembler generates code with a default value of 0 for operand C. The zero bit Z of status register STA is set, if the transferred bits 0 to W–1 are zero, otherwise the zero bit is cleared. If the MHB register is selected as destination register A (A ∈ OREG), the bits 0 to RDW–W–1 of the referred memory location are transferred to MHB. The program counter PC is incremented by the value 4.

(7) MRDIO Instruction

Syntax: MRDIO A, B

Encoding: 1010a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1101-a_4 -b_4 ----------

Operation: A←MEM((B[RAW+USR+1:2] + R5[RAW+USR+1:2]) AND MASK)[W–1:0]
MHB←MEM((B[RAW+USR+1:2] + R5[RAW+USR+1:2]) AND MASK)[RDW–1:W]

Status: Z

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer the bits 0 to W–1 of a memory location to register A (A ∈ XOREG) using indirect addressing with offset calculation.
The upper RDW–W bits of this memory location are transferred to MHB register.
The memory location where to read from depends on register B (B ∈ BAREG) and register R5 and it is defined as ((B[RAW+USR+1:2] +R5[RAW+USR+1:2]) AND MASK), where AND is a bitwise AND conjunction and MASK is a bit mask value

2^(RAW+USR)-1.
The zero bit Z of status register STA is set, if the transferred bits 0 to W–1 are zero, otherwise the zero bit is cleared.
If the MHB register is selected as destination register A, the bits 0 to RDW–W–1 of the referred memory location are transferred to MHB.
The program counter PC is incremented by the value 4.

(8) MWRI Instruction

Syntax: MWRI A, B [, C]

Encoding: $1010a_3a_2a_1a_0b_3b_2b_1b_0 0100c_{15}c_{14}c_{13}c_{12}c_{11}c_{10}c_9c_8c_7c_6c_5c_4c_3c_2$--

Operation: MEM(B[RAW+USR+1:2] + C)AND MASK)[W-1:0] ← A;
MEM(B[RAW+USR+1:2] + C)AND MASK)[RDW-1:W] ← MHB

Status: —

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer value of register A (A ∈ OREG) to the least significant bits 0 to W–1 of a memory location using indirect addressing.
The MHB register is moved to the bits W to RDW-1 at the same memory location.
If the optional operand C is not available in the assembler syntax, the MCS assembler generates code with a default value of 0 for operand C.
The memory location where to write to depends on register B (B ∈ OREG) and literal C (C ∈ AOLIT) and it is defined as ((B[RAW+USR+1:2] + C) AND MASK), where AND is a bitwise AND conjunction and MASK is a bit mask value $2^{RAW+USR}$-1.
The program counter PC is incremented by the value 4.

(9) MWRIO Instruction

Syntax: MWRIO A, B

Encoding: $1010a_3a_2a_1a_0b_3b_2b_1b_0 1110\text{-}a_4\text{-}b_4$ ----------

Operation: MEM((B[RAW+USR+1:2] + R5[RAW+USR+1:2]) AND MASK)[W–1:0]←A;
MEM((B[RAW+USR+1:2] + R5[RAW+USR+1:2]) AND MASK)[RDW–1:W]←MHB

Status: —

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description: Transfer value of register A (A ∈ XOREG) to the LSBs 0 to W–1 of a memory location using indirect addressing with offset calculation.
The MHB register is moved to the bits W to RDW–1 at the same memory location.
The memory location where to write to depends on register B (B ∈ BAREG) and register R5 and it is defined as ((B[RAW+USR+1:2] + R5[RAW+USR+1:2]) AND MASK), where AND is a bitwise AND conjunction and MASK is a bit mask value $2^{RAW+USR}$-1.
The program counter PC is incremented by the value 4.

(10) MWRIL Instruction

Syntax: MWRIL A, B

Encoding: $1010a_3a_2a_1a_0b_3b_2b_1b_0 1000$----------

Operation: MEM(B[RAW+USR+1:0])[W-1:0] ← A;

Status: —

Duration: 3 instruction cycles but not faster than 1+2*NPS clock cycles due to pipeline flushing.

Description: Transfer W bit value of A (A ∈ OREG) to memory using indirect addressing.
The memory location where to write to is defined by the bits 2 to RAW+1 of register B (B ∈ OREG).
The W bit value is stored in the LSBs (bit 0 to W-1) of the memory location and the bits W to RDW–1 are left unchanged.
The program counter PC is incremented by the value 4.
This operation is not an atomic instruction.

(11) POP Instruction

Syntax: POP A

Encoding: $1010a_3a_2a_1a_0$----0101----------

Operation:
A ← MEM(R7[RAW+USR+1:2])[W-1:0];
MHB ← MEM(R7[RAW+USR+1:2])[RDW–1:W];
R7 ← R7 – 4;
SP_CNT ← SP_CNT – 1

Status: Z, EN

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description:
Transfer the lower significant bits (bit 0 to W–1) from the top of stack to register A (A ∈ OREG \\ {R7}), followed by decrementing the stack pointer register R7 with the value 4.
The upper bits W to RDW-1 from the top of the stack are transferred to register MHB.
If the MHB register is selected as destination register A (A ∈ OREG \\ {R7}), the bits 0 to RDW–W–1 from the top of the stack are transferred to MHB.
The memory location for the top of the stack is identified by the bits 2 to RAW+1 of the stack pointer register R7.
The zero bit Z of status register STA is set, if the lower W bit of the transferred value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.
The SP_CNT bit field inside the MCS[i]_CH[x]_CTRL register is decremented.
If an underflow on the SP_CNT bit field occurs, the STK_ERR[i]_IRQ is raised.
If an underflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL is set, the current MCS-channel is disabled by clearing the EN bit of STA.

(12) PUSH Instruction

Syntax: PUSH A

Encoding: $1010a_3a_2a_1a_0$----0110----------

Operation:
R7 ← R7 + 4;
MEM(R7[RAW+USR+1:2])[W-1:0] ← A;
MEM(R7[RAW+USR+1:2])[RDW-1:W] ← MHB
SP_CNT ← SP_CNT + 1;

Status: EN

Duration: 2 instruction cycles but not faster than 1+NPS clock cycles due to pipeline flushing.

Description:
Increment the stack pointer register R7 with the value 4, followed by transferring a W bit value of operand A (A ∈ OREG \\ {R7}) together with a MHB register to the new top of the stack. The W bit value of A is stored in the bits 0 to W-1 of the memory location.
The content of the MHB register is stored in the bit W to RDW-1 of the memory location.
The memory location for the top of the stack is referred by the bits 2 to RAW+1 of the stack pointer register.
The program counter PC is incremented by the value 4.
The SP_CNT bit field inside the MCS[i]_CH[x]_CTRL register is incremented.
If an overflow on the SP_CNT bit field occurs, the STK_ERR[i]_IRQ is raised.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL is set, the current MCS-channel is disabled by clearing the EN bit of STA.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL is set, the memory write operation for the A and MHB is discard.

(13) ARD Instruction

Syntax: ARD A, B, C

Encoding: $1011a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0000-------$c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation:
A ← ARU(C)[W-1:0];
B ← ARU(C)[2*W-1:W];
ACB ← ARU(C)[4+2*W:2*W]

Status: CAT, SAT

Duration: Suspends current MCS-channel until ARU transfer finished.

Description: Perform a blocking read access to the ARU and transfer both W bit values received at the ARU port to the registers A and B (A ∈ AREG, B ∈ AREG), whereas A holds the lower W bit ARU word and B holds the upper W bit ARU word.
If A and B see the same register, only the upper W bit ARU word is stored and the lower W bit ARU word is discard.
If any transferred W bit value from the ARU should not stored in a register, the dummy register ZERO ∈ AREG can be selected in A or B to discard the corresponding ARU data. The binary encoding of the address for the dummy register ZERO can be chosen by an arbitrary value within the range 8 to 15.
The received ARU control bits are stored in the register ACB.
The literal C (C ∈ ARDLIT) define the ARU address where to read from.
At the beginning of the instruction execution the CAT bit in the register STA is always cleared. After the execution of the instruction the SAT flag of the register STA is updated in order to show if the transfer was successful (SAT = 1) or if the transfer failed (SAT = 0) due to a cancellation by the CPU.
The program counter PC is incremented by the value 4.

(14) ARDI Instruction

Syntax: ARDI A, B

Encoding: $1011a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0100----------

Operation:
A ← ARU(R6[8:0])[W-1:0];
B ← ARU(R6[8:0])[2*W-1:W];
ACB ← ARU(R6[8:0])[4+2*W:2*W]

Status: CAT, SAT

Duration: Suspends current MCS-channel until ARU transfer finished.

Description: Perform a blocking read access to the ARU and transfer both W bit values received at the ARU port to the registers A and B (A ∈ AREG, B ∈ AREG), whereas A holds the lower W bit ARU word and B holds the upper W bit ARU word.
If A and B refer to the same register, only the upper W bit ARU word is stored and the lower W bit ARU word is discarded.
If any transferred W bit value from the ARU should not be stored in a register, the dummy register ZERO ∈ AREG can be selected in A or B to discard the corresponding ARU data. The binary encoding of the address for the dummy register ZERO can be chosen by an arbitrary value within the range 8 to 15.
The received ARU control bits are stored in the register ACB.
The read address is obtained from the bits 0 to 8 of the channels register R6.
At the beginning of the instruction execution the CAT bit in register STA is always cleared. After the execution of the instruction the SAT flag of the register STA is updated in order to show if the transfer was successful (SAT = 1) or if the transfer failed (SAT = 0) due to a cancellation by the CPU.
The program counter PC is incremented by the value 4.

(15) AWR Instruction

Syntax: AWR A, B, C

Encoding: $1011a_3a_2a_1a_0b_3b_2b_1b_0$ 0001----------$c_4c_3c_2c_1c_0$

Operation:
ARU(C)[W-1:0] ← A;
ARU(C)[2*W-1:W] ← B;
ARU(C)[4+2*W:2*W] ← ACB;

Status: CAT, SAT

Duration: Suspends current MCS-channel until ARU transfer finished.

Description: Perform a blocking write access to the ARU and transfer two W bit values to the ARU port using the registers A and B (A ∈ OREG, B ∈ OREG), whereas A holds the lower W bit ARU word and B holds the upper W bit ARU word.
The ARU control bits are taken from the register ACB.
The literal C (C ∈ AWRLIT) defines an index into the pool of ARU write addresses that are used for writing data. This index is mapped to an ARU write address as shown in column "MCS write index" of table "ARU Write Addresses" in **Section 38.28, GTM Device 358**.
Each MCS sub module has a pool of several write addresses that can be shared between all MCS-channels arbitrarily.
At the beginning of the instruction execution the CAT bit of the register STA is always cleared. After the execution of the instruction the SAT flag of the register STA is updated in order to show if the transfer was successful (SAT = 1) or if the transfer failed (SAT = 0) due to a cancellation by the CPU.
The program counter PC is incremented by the value 4.

(16) AWRI Instruction

Syntax: AWRI A, B

Encoding: $1011a_3a_2a_1a_0b_3b_2b_1b_0$ 0101----------

Operation:
ARU(R6[4:0])[W-1:0] ← A;
ARU(R6[4:0])[2*W-1:W] ← B;
ARU(R6[4:0])[4+2*W:2*W] ← ACB;

Status: CAT, SAT

Duration: Suspends current MCS-channel until ARU transfer finished.

Description: Perform a blocking write access to the ARU and transfer two W bit values to the ARU port using the registers A and B (A ∈ OREG, B ∈ OREG), whereas A holds the lower W bit ARU word and B holds the upper W bit ARU word.
The ARU control bits are taken from the register ACB.
The bits 0 to 4 of the register R6 define an index into the pool of ARU write addresses that are used for writing data. This index is mapped to an ARU write address as shown in column "MCS write index" of table "ARU Write Addresses" in **Section 38.28, GTM Device 358**.
Each MCS sub module has a pool of several write addresses that can be shared between all MCS-channels arbitrarily.
At the beginning of the instruction execution the CAT bit of the register STA is always cleared. After the execution of the instruction the SAT flag of the register STA is updated in order to show if the transfer was successful (SAT = 1) or if the transfer failed (SAT = 0) due to a cancellation by the CPU.
The program counter PC is incremented by the value 4.

(17) NARD Instruction

Syntax: NARD A, B, C

Encoding: $1011 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0010-------$c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation:
A ← ARU(C)[W-1:0];
B ← ARU(C)[2*W:W];
ACB ← ARU(C)[4+2*W:2*W]

Status: SAT

Duration: Suspends current MCS-channel until the ARU is selecting the MCS-channel.

Description: Perform a non-blocking read access to the ARU trying to transfer both W bit values received at the ARU port to the registers A and B (A ∈ AREG, B ∈ AREG), whereas A holds the lower W bit ARU word, B holds the upper W bit ARU word, and the ACB register holds the received ARU control bits. The literal C (C ∈ ARDLIT) define the ARU address where to read from.
Non-blocking ARU read access means that the instruction is suspending the MCS channel until the ARU scheduler is selecting the requesting MCS channel. If the transfer finished successfully, the bit SAT of the register STA is set and the transferred values are stored in the registers A, B, and ACB. If the transfer failed due to missing data at the requested source, the bit SAT of the register STA is cleared and registers A, B, and ACB are not changed.
If A and B refer to the same register, only the upper W bit ARU word is stored and the lower W bit ARU word is discard.
If any transferred W bit value from the ARU should not stored in a register, the dummy register ZERO ∈ AREG can be selected in A or B to discard the corresponding ARU data. The binary encoding of the address for the dummy register ZERO can be chosen by an arbitrary value within the range 8 to 15.
The program counter PC is incremented by the value 4.

(18) NARDI Instruction

Syntax: NARDI A, B

Encoding: $1011a_3a_2a_1a_0b_3b_2b_1b_0$0011----------

Operation:
A ← ARU(R6[8:0])[W-1:0];
B ← ARU(R6[8:0])[2*W-1:W];
ACB ← ARU(R6[8:0])[4+2*W:2*W]

Status: SAT

Duration: Suspends current MCS-channel until the ARU is selecting the MCS-channel.

Description: Perform a non-blocking read access to the ARU trying to transfer both W bit values received at the ARU port to the registers A and B (A ∈ AREG, B ∈ AREG), whereas A holds the lower W bit ARU word, B holds the upper W bit ARU word, and the ACB register holds the received ARU control bits. The read address is obtained from the bits 0 to 8 of the channels register R6.
Non-blocking ARU read access means that the instruction is suspending the MCS channel until the ARU scheduler is selecting the requesting MCS channel. If the transfer finished successfully, the bit SAT of the register STA is set and the transferred values are stored in the registers A, B, and ACB. If the transfer failed due to missing data at the requested source, the bit SAT of the register STA is cleared and registers A, B, and ACB are not changed.
If A and B refer to the same register, only the upper W bit ARU word is stored and the lower 24 bit ARU word is discard.
If any transferred W bit value from the ARU should not stored in a register, the dummy register ZERO ∈ AREG can be selected in A or B to discard the corresponding ARU data. The binary encoding of the address for the dummy register ZERO can be chosen by an arbitrary value within the range 8 to 15.
The program counter PC is incremented by the value 4.

(19) BRD Instruction

Syntax: BRD A, C

Encoding: 1010-a_2 a_1 a_0----1001c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2--

Operation: A ← BUS(C)[W–1:0];
MHB ← BUS(C)[BDW–1:W]

Status: –

Duration: Suspends current MCS-channel if addressed slave inserts at least one wait cycle (e.g. accessing a RAM module) otherwise 1 instruction cycle.

Description: Initiate a read access at the bus master interface using the address C (C ∈ BALIT) and transfer the lower W bits of the received data to register A (A ∈ GREG).
The upper BDW-W bits of the received data are transferred to the MHB register.
If the delay between the a BRD instruction and its successor instruction is one or two system clock cycles (e.g. in accelerated scheduling mode) and the successor instruction is reading data resulting from the BRD instruction, a data hazard in the pipeline occurs resulting in a pipeline flush. This means, if very fast program execution is required (e.g. only one task is activated in accelerated scheduling mode) a program sequence like BRD R1, 0x0288; ADD R3, R1; (9 clock cycles) can be accelerated by reformulating the sequence as BRD R1, 0x0288; NOP; NOP; ADD R3, R1; (4 clock cycles).
Since the MHB register is always transferred via AEI bus master it also figures out another data dependency, which can cause a data hazard resulting in a pipeline flush. Therefore a sequence like BRD R1, 0x0288; BWR R3, 0x304 (9 clock cycles) could also be optimized by the sequence BRD R1, 0x0288; NOP; NOP; BWR R3, 0x304(4 clock cycles).
The program counter PC is incremented by the value 4.

(20) BWR Instruction

Syntax: BWR A, C

Encoding: 1010-a_2 a_1 a_0----1010c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2--

Operation: BUS(C)[W–1:0] ← A;
BUS(C)[BDW–1:W] ← MHB

Status: –

Duration: Suspends current MCS-channel if addressed slave inserts at least one wait cycle (e.g. accessing a RAM module) otherwise 1 instruction cycle.

Description: Initiate a write access at the bus master interface using the address C (C ∈ BALIT) and transfer the content of register A (A ∈ GREG) to the bits 0 to W-1 of the bus.
The content of the MHB register is transferred to the bits W to BDW-W-1 of the bus.
The program counter PC is incremented by the value 4.

(21) BRDI Instruction

Syntax: BRDI A, B

Encoding: 1010-$a_2 a_1 a_0$-$b_2 b_1 b_0$1011----------

Operation: A ← BUS(B[BAW+1:2])[W–1:0];
MHB ← BUS(B[BAW+1:2])[BDW–1:W]

Status: –

Duration: Suspends current MCS-channel if addressed slave inserts at least one wait cycle (e.g. accessing a RAM module) otherwise 1 instruction cycle.

Description: Initiate a read access at the bus master interface using indirect addressing and transfer the lower W bits of the received data to register A (A ∈ GREG).
The upper BDW-W bits of the received data are transferred to the MHB register.
The address for the transfer is identified by the bits 2 to BAW+1 of register B (B ∈ GREG).
If the delay between the a BRDI instruction and its successor instruction is one or two system clock cycles (e.g. in accelerated scheduling mode) and the successor instruction is reading data resulting from the BRDI instruction, a data hazard in the pipeline occurs resulting in a pipeline flush. This means, if very fast program execution is required (e.g. only one task is activated in accelerated scheduling mode) a program sequence like BRDI R1, R6; ADD R3, R1; (9 clock cycles) can be accelerated by reformulating the sequence as BRDI R1, R6; NOP; NOP; ADD R3, R1; (4 clock cycles).
Since the MHB register is always transferred via AEI bus master it also figures out another data dependency, which can cause a data hazard resulting in a pipeline flush. Therefore a sequence like BRDI R1, R2; BWRI R3, R4 (9 clock cycles) could also be optimized by the sequence BRDI R1, R2; NOP; NOP; BWRI R3, R4 (4 clock cycles).
The program counter PC is incremented by the value 4.

(22) BWRI Instruction

Syntax: BWRI A, B

Encoding: 1010-$a_2 a_1 a_0$-$b_2 b_1 b_0$1100----------

Operation: BUS(B[BAW+1:2])[W–1:0] ← A;
BUS(B[BAW+1:2])[BDW–1:W] ← MHB

Status: –

Duration: Suspends current MCS-channel if addressed slave inserts at least one wait cycle (e.g. accessing a RAM module) otherwise 1 instruction cycle.

Description: Initiate a write access at the bus master interface using the indirect addressing and transfer the content of register A (A ∈ GREG) to the bits 0 to W-1 of the bus.
The content of the MHB register is transferred to the bits W to BDW-W-1 of the bus.
The address for the transfer is identified by the bits 2 to BAW+1 of register B (B ∈ GREG).
The program counter PC is incremented by the value 4.

(23) ADDL Instruction

Syntax: ADDL A, C

Encoding: $0010a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A ← A + C

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform addition operation of a register A (A ∈ OREG) with a W bit literal value C (C ∈ WLIT) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned overflow/underflow occurred during addition, otherwise the bit is cleared. An unsigned overflow has occurred when the result of the operation cannot be represented in the interval $[0; 2^W-1]$, assuming that both operands A and C are unsigned values within the interval $[0; 2^W-1]$.
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during addition, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval $[-2^{W-1}; 2^{W-1}-1]$, assuming that both operands A and C are signed values within the interval $[-2^{W-1}; 2^{W-1}-1]$.
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(24) ADD Instruction

Syntax: ADD A, B

Encoding: $1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0000-a_4-b_4 ----------

Operation: A ← A + B

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform addition operation of a register A (A ∈ OREG) with an operand B (B ∈ OREG). The result is stored in the register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned overflow occurred during addition, otherwise the bit is cleared. An unsigned overflow has occurred when the result of the operation cannot be represented in the interval $[0; 2^W-1]$, assuming that both operands A and B are unsigned values within the interval $[0; 2^W-1]$.
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during addition, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval $[-2^{W-1}; 2^{W-1}-1]$, assuming that both operands A and B are signed values within the interval $[-2^{W-1}; 2^{W-1}-1]$.
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(25) ADDC Instruction

Syntax: ADDC A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0110-a_4 -b_4 ----------

Operation: A ← A + B + CY

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform addition operation of a register A (A ∈ XOREG) with an operand B (B ∈ XOREG) and the carry flag CY. The result is stored in the register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned overflow occurred during addition, otherwise the bit is cleared. An unsigned overflow has occurred when the result of the operation cannot be represented in the interval [0; 2^W-1], assuming that both operands A and B are unsigned values within the interval [0; 2^W-1].
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during addition, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], assuming that both operands A and B are signed values within the interval [-2^{W-1}; 2^{W-1}-1].
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(26) SUBL Instruction

Syntax: SUBL A, C

Encoding: 0011a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0

Operation: A ← A - C

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform subtraction operation of a register A (A ∈ OREG) with a W bit literal value C (C ∈ WLIT). The result is stored in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned underflow occurred during subtraction, otherwise the bit is cleared. An unsigned underflow has occurred when the result of the operation cannot be represented in the interval [0; 2^W-1], assuming that both operands A and C are unsigned values within the interval [0; 2^W-1].
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during subtraction, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], assuming that both operands A and C are signed values within the interval [-2^{W-1}; 2^{W-1}-1].
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(27) SUB Instruction

Syntax: SUB A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,0001\text{-}a_4\text{-}b_4$ ----------

Operation: A ← A – B

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform subtraction operation of a register A (A ∈ OREG) with an operand B (B ∈ OREG). The result is stored in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned underflow occurred during subtraction, otherwise the bit is cleared. An unsigned underflow has occurred when the result of the operation cannot be represented in the interval [0; 2^W-1], assuming that both operands A and B are unsigned values within the interval [0; 2^W-1].
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during subtraction, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], assuming that both operands A and B are signed values within the interval [-2^{W-1}; 2^{W-1}-1].
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(28) SUBC Instruction

Syntax: SUBC A, B

Encoding: $1101a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,0111\text{-}a_4\text{-}b_4$ ----------

Operation: A ← A – B – CY

Status: Z, CY, N, V

Duration: 1 instruction cycle

Description: Perform subtraction operation of a register A (A ∈ XOREG) with an operand B (B ∈ XOREG) and the carry flag CY. The result is stored in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is set, if an unsigned underflow occurred during subtraction, otherwise the bit is cleared. An unsigned underflow has occurred when the result of the operation cannot be represented in the interval [0; 2^W-1], assuming that both operands A and B are unsigned values within the interval [0; 2^W-1].
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during subtraction, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], assuming that both operands A and B are signed values within the interval [-2^{W-1}; 2^{W-1}-1].
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(29) NEG Instruction

Syntax: NEG A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,0010\text{-}a_4\text{-}b_4$ ----------

Operation: A ← –B

Status: Z, N, V

Duration: 1 instruction cycle

Description: Perform negation operation (2's Complement) with an operand B (B ∈ OREG) and store the result in a register A (A ∈ OREG).
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during subtraction, otherwise the bit is cleared. A signed overflow/underflow has occurred when the result of the operation cannot be represented in the interval $[-2^{W-1}; 2^{W-1}-1]$, assuming that both operands A and B are signed values within the interval $[-2^{W-1}; 2^{W-1}-1]$.
The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N = 1) or positive (N = 0), assuming that no overflow/underflow occurred.
The program counter PC is incremented by the value 4.

(30) ANDL Instruction

Syntax: ANDL A, C

Encoding: $0100a_3\,a_2\,a_1\,a_0\,c_{23}\,c_{22}\,c_{21}\,c_{20}\,c_{19}\,c_{18}\,c_{17}\,c_{16}\,c_{15}\,c_{14}\,c_{13}\,c_{12}\,c_{11}\,c_{10}\,c_9\,c_8\,c_7\,c_6\,c_5\,c_4\,c_3\,c_2\,c_1\,c_0$

Operation: A ← A AND C

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise AND conjunction of a register A (A ∈ OREG) with a W bit literal value C (C ∈ WLIT) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(31) AND Instruction

Syntax: AND A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,0011\text{-}a_4\text{-}b_4$ ----------

Operation: A ← A AND B

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise AND conjunction of a register A (A ∈ OREG) with an operand B (B ∈ OREG) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(32) ORL Instruction

Syntax: ORL A, C

Encoding: $0101 a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A ← A OR C

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise OR conjunction of a register A (A ∈ OREG) with a W bit literal value C (C ∈ WLIT) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(33) OR Instruction

Syntax: OR A, B

Encoding: $1100 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0100-a_4-b_4 ----------

Operation: A ← A OR B

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise OR conjunction of a register A (A ∈ OREG) with an operand B (B ∈ OREG) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(34) XORL Instruction

Syntax: XORL A, C

Encoding: $0110 a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A ← A XOR C

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise XOR conjunction of a register A (A ∈ OREG) with a W bit literal value C (C ∈ WLIT) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(35) XOR Instruction

Syntax: XOR A, B

Encoding: $1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0101-a_4-b_4 ----------

Operation: A ← A XOR B

Status: Z

Duration: 1 instruction cycle

Description: Perform bitwise XOR conjunction of a register A (A ∈ XOREG) with an operand B (B ∈ XOREG) and store the result in register A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(36) SHR Instruction

Syntax: SHR A, C

Encoding: $1100a_3 a_2 a_1 a_0$ ----0110-a_4 ---------$c_4 c_3 c_2 c_1 c_0$

Operation: A ← A >> C

Status: Z, CY

Duration: 1 instruction cycle

Description: Perform right shift operation C (C ∈ SFTLIT) times of register A (A ∈ XOREG). The MSBs that are shifted into A are cleared.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is updated to the last LSB that is shifted out of the register. If the shift value C is 0 the carry bit CY is cleared.
The program counter PC is incremented by the value 4.

(37) SHL Instruction

Syntax: SHL A, C

Encoding: $1100a_3 a_2 a_1 a_0$ ----0111-a_4 ---------$c_4 c_3 c_2 c_1 c_0$

Operation: A ← A << C

Status: Z, CY

Duration: 1 instruction cycle

Description: Perform left shift operation C (C ∈ SFTLIT) times of register A (A ∈ XOREG). The LSBs that are shifted into A are cleared.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The carry bit CY of status register STA is updated to the previous MSB that is shifted out of the register. If the register A contains less than W bits, or if C is 0 the carry bit CY is always cleared.
The program counter PC is incremented by the value 4.

(38) ASRU Instruction

Syntax: ASRU A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0100-a_4 -b_4 ----------

Operation: A ← A >> B

Status: Z

Duration: 1 instruction cycle

Description: Perform arithmetic unsigned right shift operation, which means that the unsigned operand of register A (A ∈ XOREG) is right shifted B times (B ∈ XOREG). Operand B is also an unsigned type. The MSBs that are shifted into A are cleared.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(39) ASRS Instruction

Syntax: ASRS A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0101-a_4 -b_4 ----------

Operation: A ← A >> B

Status: Z

Duration: 1 instruction cycle

Description: Perform arithmetic signed right shift operation, which means that the signed operand of register A (A ∈ XOREG) is right shifted B times (B ∈ XOREG). Operand B is an unsigned type. The operation also performs a sign extension, which means that value of the MSBs that are shifted into A are determined by the MSB of the original operand A.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(40) ASL Instruction

Syntax: ASL A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0011-a_4 -b_4 ----------

Operation: A ← A << B

Status: Z, CY, V

Duration: 1 instruction cycle

Description: Perform arithmetic left shift operation for signed and unsigned numbers, which means that the operand of register A (A ∈ XOREG) is left shifted B times (B ∈ XOREG). Operand B is always an unsigned type.
The carry bit CY of status register STA is set, if an unsigned overflow occurred during shifting, otherwise the bit is cleared. An unsigned overflow has occurred if the calculated result $A*2^B$ cannot be represented in the interval [0; 2^W-1], assuming that both operands A and B are unsigned values within the interval [0; 2^W-1].
The overflow bit V of status register STA is set, if a signed overflow/underflow occurred during shifting, otherwise the bit is cleared. A signed overflow/underflow has occurred when the calculated result $A*2^B$ cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], assuming that signed operand A is within the interval [-2^{W-1}; 2^{W-1}-1] and the unsigned operand B is within the interval [0; 2^{W-1}-1].
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(41) MULU Instruction

Syntax: MULU A, B[, C]

Encoding: $1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0\ 1000\text{-}a_4\ \text{-}b_4\ \text{-------}c_4 c_3 c_2 c_1 c_0$

Operation: [[R4,] A] ← A[(C–1):0] * B[(C–1):0]

Status: Z

Duration: 1 instruction cycle

Description: Perform an unsigned multiplication operation of an operand A (A ∈ XOREG if C is less than or equal to W/2; A ∈ XOREG \\ {R4} if C is greater than W/2) with an operand B (B ∈ XOREG). The multiplication is only performed with the bits 0 to C–1 (C ∈ BWSLIT) of both operands A and B and the bits C to W–1 are ignored. If C is less than or equal to W/2, the product of the multiplication is stored in register A and register R4 is left unchanged. If C is greater than W/2, the bits 0 to W–1 are stored in A and the bits W to 2*C–1 are stored in R4. The results stored in the registers are always zero extended to W bits. If the optional operand C is not specified in the assembler code, the MCS assembler generates code with a default value of W for operand C. The zero bit Z of status register STA is set, if the calculated product is zero, otherwise the zero bit is cleared.

If the delay between the a MULU instruction and its successor instruction is one system clock cycles (e.g. in accelerated scheduling mode) and the successor instruction is either a WURM, WURMX, WURCX, WUCE, BRDI, BWR, or BWRI instruction that is accessing the multiplication result as argument a data hazard in the pipeline occurs resulting in a pipeline flush. This means, if very fast program execution is required (e.g. only one task is activated in accelerated scheduling mode) a program sequence like MULU R1, R2; BWRI R1, R3; (9 clock cycles) can be accelerated by reformulating the sequence as MULU R1, R2; NOP; BWRI R1, R3; (3 clock cycles).

The program counter PC is incremented by the value 4.

(42) MULS Instruction

Syntax: MULS A, B[, C]

Encoding: 1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1001-a_4 -b_4 -------c_4 c_3 c_2 c_1 c_0

Operation: [[R4,] A] ← A[(C–1):0] * B[(C–1):0]

Status: Z, N

Duration: 1 instruction cycle

Description: Perform a signed multiplication operation of an operand A (A ∈ XOREG if C is less than or equal to W/2; A ∈ XOREG \\ {R4} if C is greater than W/2) with an operand B (B ∈ XOREG). The multiplication is only performed with the bits 0 to C–1 (C ∈ BWSLIT) of both operands A and B, in which bit C–1 is used as sign bit ($–2^{(C–1)}$) and the bits C to W–1 are ignored. If C is less than or equal to W/2, the product of the multiplication is stored in register A and register R4 is left unchanged. If C is greater than W/2, the bits 0 to W–1 are stored in A and the bits W to 2*C–1 are stored in R4. The results stored in the registers are always sign extended to W bits. If the optional operand C is not specified in the assembler code, the MCS assembler generates code with a default value of W for operand C. The zero bit Z of status register STA is set, if the calculated product is zero, otherwise the zero bit is cleared. The negative bit N of status register STA equals the MSB of the operation result, in order to determine if a calculated signed result is negative (N=1) or positive (N=0).
If the delay between the a MULS instruction and its successor instruction is one system clock cycles (e.g. in accelerated scheduling mode) and the successor instruction is either a WURM, WURMX, WURCX, WUCE, BRDI, BWR, or BWRI instruction that is accessing the multiplication result as argument a data hazard in the pipeline occurs resulting in a pipeline flush. This means, if very fast program execution is required (e.g. only one task is activated in accelerated scheduling mode) a program sequence like MULS R1, R2; BWRI R1, R3; (9 clock cycles) can be accelerated by reformulating the sequence as MULS R1, R2; NOP; BWRI R1, R3; (3 clock cycles).
The program counter PC is incremented by the value 4.

(43) DIVU Instruction

Syntax: DIVU A, B[, C]

Encoding: 1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1010-a_4 -b_4 -------c_4 c_3 c_2 c_1 c_0

Operation: R4 ← A[(C–1):0] – B[(C–1):0] * [A[(C–1):0] / B[(C–1):0]];
A ← [A[(C–1):0] / B[(C–1):0]]

Status: CY, Z, ERR

Duration: C instruction cycles but not faster than C+NPS–1 clock cycles due to pipeline flushing.

Description: Perform an unsigned division operation of operand A (A ∈ XOREG \\ {R4, B}) divided by operand B (B ∈ XOREG \\ {R4, A}).
The division is only performed with the bits 0 to C–1 (C ∈ BWSLIT) of the operands and the remaining bits C to W–1 are ignored. This means that the dynamic range of A and B is defined in the interval [0; 2^C-1]. The integral part of the quotient is stored in the register A and the remainder of the division is stored in register R4. The resulting quotient A and remainder R4 are always zero extended to W bits. If the optional operand C is not specified in the assembler code, the MCS assembler generates code with a default value of W for operand C. If the bits 0 to C–1 of operand B are zero, the MCS channel is disabled and the ERR bit in the status register STA is set. The zero bit Z of status register STA is set, if the calculated quotient in A is zero, otherwise the zero bit is cleared. The carry bit CY of status register STA is set, if the calculated remainder in R4 is not zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(44) DIVS Instruction

Syntax: DIVS A, B[, C]

Encoding: 1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1011-a_4 -b_4 -------c_4 c_3 c_2 c_1 c_0

Operation: R4 ← A[(C–1):0] – B[(C–1):0] * [A[(C–1):0] / B[(C–1):0]];
A ← [A[(C–1):0] / B[(C–1):0]]

Status: CY, Z, N, V, ERR

Duration: C + 4 instruction cycles but not faster than C+3+NPS clock cycles due to pipeline flushing.

Description: Perform a signed division operation of operand A (A ∈ XOREG \\ {R4, B}) divided by operand B (B ∈ XOREG \\ {R4, A}). The division is only performed with the bits 0 to C–1 (C ∈ BWSLIT) of both operands, in which bit C–1 is used as sign bit ($–2^{C-1}$) and the bits C to W–1 are ignored. This means that the dynamic range of A[(C–1):0] and B[(C–1):0] is defined in the interval [-2^{C-1}; 2^{C-1}-1]. The integral part of the quotient is stored in the register A and the remainder of the division is stored in register R4. The resulting quotient A and remainder R4 are always sign extended to W bits. The integral part of the quotient is always truncated towards 0. The sign of the remainder is always the same sign as the dividend A[(C–1):0]. The absolute value of the remained is always less than the divisor B[(C–1):0]. If the optional operand C is not specified in the assembler code, the MCS assembler generates code with a default value of W for operand C. If the bits 0 to C–1 of operand B are zero, the MCS channel is disabled and the ERR bit in the status register STA is set. The zero bit Z of status register STA is set, if the calculated quotient in A is zero, otherwise the zero bit is cleared. The carry bit CY of status register STA is set, if the calculated remainder in R4 is not zero, otherwise the zero bit is cleared. The overflow bit V of status register STA is set, if the calculated quotient in A cannot be represented in the interval [-2^{W-1}; 2^{W-1}-1], otherwise the overflow bit is cleared. The negative bit N of status register STA equals the MSB of the quotient, in order to determine if a calculated signed result is negative (N=1) or positive (N=0).
The program counter PC is incremented by the value 4.

(45) MINU Instruction

Syntax: MINU A, B

Encoding: 1100a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1100-a_4 -b_4 ----------

Operation: A ← MIN(A, B)

Status: Z

Duration: 1 instruction cycle

Description: Determine the minimum of an unsigned operand A (A ∈ XOREG) and an unsigned operand B (B ∈ XOREG). If A is less than or equal to B, A is left unchanged. Otherwise, if A is greater than B, the operand B is moved to A.
The zero bit Z of status register STA is set, if the calculated result of A is zero, otherwise the zero bit is cleared.

(46) MINS Instruction

Syntax: MINS A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,1101\text{-}a_4\text{-}b_4$ ----------

Operation: A ← MIN(A, B)

Status: Z

Duration: 1 instruction cycle

Description: Determine the minimum of a signed operand A (A ∈ XOREG) and a signed operand B (B ∈ XOREG). If A is less than or equal to B, A is left unchanged. Otherwise, if A is greater than B, the operand B is moved to A.
The zero bit Z of status register STA is set, if the calculated result of A is zero, otherwise the zero bit is cleared.

(47) MAXU Instruction

Syntax: MAXU A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,1110\text{-}a_4\text{-}b_4$ ----------

Operation: A ← MAX(A, B)

Status: Z

Duration: 1 instruction cycle

Description: Determine the maximum of an unsigned operand A (A ∈ XOREG) and an unsigned operand B (B ∈ XOREG). If A is greater than or equal to B, A is left unchanged. Otherwise, if A is less than B, the operand B is moved to A.
The zero bit Z of status register STA is set, if the calculated result of A is zero, otherwise the zero bit is cleared.

(48) MAXS Instruction

Syntax: MAXS A, B

Encoding: $1100a_3\,a_2\,a_1\,a_0\,b_3\,b_2\,b_1\,b_0\,1111\text{-}a_4\text{-}b_4$ ----------

Operation: A ← MAX(A, B)

Status: Z

Duration: 1 instruction cycle

Description: Determine the maximum of a signed operand A (A ∈ XOREG) and a signed operand B (B ∈ XOREG). If A is greater than or equal to B, A is left unchanged. Otherwise, if A is less than B, the operand B is moved to A.
The zero bit Z of status register STA is set, if the calculated result of A is zero, otherwise the zero bit is cleared.

(49) ATUL Instruction

Syntax: ATUL A, C

Encoding: $0111a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A – C

Status: Z, CY

Duration: 1 instruction cycle

Description: Arithmetic Test with an unsigned operand A (A ∈ OREG) and an unsigned W bit literal value C (C ∈ WLIT).
The carry bit CY of status register STA is set if unsigned operand A is less than unsigned literal C.
Otherwise, the carry bit CY of status register STA is cleared if unsigned operand A is greater than or equal to unsigned literal C.
The zero bit Z of status register STA is set, if A equals to C.
Otherwise, the zero bit Z of status register STA is cleared, if A is unequal to C.
The program counter PC is incremented by the value 4.

(50) ATU Instruction

Syntax: ATU A, B

Encoding: $1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0000-a_4 -b_4 ----------

Operation: A – B

Status: Z, CY

Duration: 1 instruction cycle

Description: Arithmetic Test with an unsigned operand A (A ∈ XOREG) and an unsigned operand B (B ∈ XOREG).
The carry bit CY of status register STA is set if unsigned operand A is less than unsigned operand B.
Otherwise, the carry bit CY of status register STA is cleared if unsigned operand A is greater than or equal to unsigned operand B.
The zero bit Z of status register STA is set, if A equals to B.
Otherwise, the zero bit Z of status register STA is cleared, if A is unequal to B. The program counter PC is incremented by the value 4.

(51) ATSL Instruction

Syntax: ATSL A, C

Encoding: $1000a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: A - C

Status: Z, CY

Duration: 1 instruction cycle

Description: Arithmetic Test with a signed operand A (A ∈ OREG) and a signed W bit literal value C (C ∈WLIT).
The carry bit CY of status register STA is set if signed operand A is less than signed literal C.
Otherwise, the carry bit CY of status register STA is cleared if signed operand A is greater than or equal to signed literal C.
The zero bit Z of status register STA is set, if A equals to C.
Otherwise, the zero bit Z of status register STA is cleared, if A is unequal to C.
The program counter PC is incremented by the value 4.

(52) ATS Instruction

Syntax: ATS A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0001-a_4 -b_4 ----------

Operation: A – B

Status: Z, CY

Duration: 1 instruction cycle

Description: Arithmetic Test with a signed operand A (A ∈ XOREG) and a signed operand B (B ∈ XOREG).
The carry bit CY of status register STA is set if signed operand A is less than signed operand B.
Otherwise, the carry bit CY of status register STA is cleared if signed operand A is greater than or equal to signed operand B.
The zero bit Z of status register STA is set, if A equals to B.
Otherwise, the zero bit Z of status register STA is cleared, if A is unequal to B. The program counter PC is incremented by the value 4.

(53) BTL Instruction

Syntax: BTL A, C

Encoding: 1001a_3 a_2 a_1 a_0 c_{23} c_{22} c_{21} c_{20} c_{19} c_{18} c_{17} c_{16} c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0

Operation: A AND C

Status: Z

Duration: 1 instruction cycle

Description: Bit test of an operand A (A ∈ OREG) with a W bit literal bit mask C (C ∈ WLIT).
The bit test is performed by applying a bitwise logical AND operation with operand A and the bit mask C without storing the result.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(54) BT Instruction

Syntax: BT A, B

Encoding: 1101a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0010-a_4 -b_4 ----------

Operation: A AND B

Status: Z

Duration: 1 instruction cycle

Description: Bit test of an operand A (A ∈ XOREG) with an operand B (B ∈ XOREG), whereas usually one of the operands is a register holding a bit mask.
The bit test is performed by applying a bitwise logical AND operation with register A and register B without storing the result.
The zero bit Z of status register STA is set, if the calculated value is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(55) SETB Instruction

Syntax: SETB A, B

Encoding: 1011a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0110-a_4 -b_4 ----------

Operation: A[B[4:0]] ? 1

Status: Z

Duration: 1 instruction cycle

Description: Set the B[4:0]–th bit (B ∈ XOREG) of an operand A (A ∈ XOREG) to true. Only the bits 0 to 4 of operand B are used as bit index of operand A and the other bits of B are ignored. If the value B[4:0] is greater than or equal to W, the operation of SETB does not modify operand A but the status flag Z is updated.
The instruction SETB performs an implicit read-modify-write operation meaning that the entire content of A is read first and after manipulating the desired bit the entire content of A is written back. This implementation may cause undesired results in special function registers and therefore it should be used carefully.
The zero bit Z of status register STA is set if the modified value of A is zero, otherwise the bit Z is cleared. The Z bit is set e.g. if the B[4:0]-th bit of A is not writable, its value is zero and all other bits of A are cleared.
The program counter PC is incremented by the value 4.

(56) CLRB Instruction

Syntax: CLRB A, B

Encoding: 1011a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0111-a_4 -b_4 ----------

Operation: A[B[4:0]] ? 0

Status: Z

Duration: 1 instruction cycle

Description: Clear the B[4:0].th bit (B ∈ XOREG) of an operand A (A ∈ XOREG). Only the bits 0 to 4 of operand B are used as bit index of operand A and the other bits of B are ignored. If the value B[4:0] is greater than or equal to W, the operation of CLRB does not modify operand A but the status flag Z is updated.
The instruction CLRB performs an implicit read-modify-write operation meaning that the entire content of A is read first and after manipulating the desired bit the entire content of A is written back. This implementation may cause undesired results in special function registers and therefore it should be used carefully.
The zero bit Z of status register STA is set if the modified value of A is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(57) XCHB Instruction

Syntax: XCHB A, B

Encoding: 1010a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 1111-a_4 -b_4 ----------

Operation: A[B[4:0]] ? CY

Status: Z, CY

Duration: 1 instruction cycle

Description: Exchange the B[4:0]-th bit (B ∈ XOREG) of an operand A (A ∈ XOREG) with the CY bit in the status register. Only the bits 0 to 4 of operand B are used as bit index of operand A and the other bits of B are ignored. If the value B[4:0] is greater than or equal to W, the operation of XCHB does not modify operand A but the status flag Z is updated and the bit CY is cleared.
The instruction XCHB performs an implicit read-modify-write operation meaning that the entire content of A is read first and after manipulating the desired bit the entire content of A is written back. This implementation may cause undesired results in special function registers and therefore it should be used carefully.
The zero bit Z of status register STA is set if the modified value of A is zero, otherwise the zero bit is cleared.
The program counter PC is incremented by the value 4.

(58) JMP Instruction

Syntax: JMP C

Encoding: 1110--------0000c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 --

Operation: PC ← C << 2

Status: —

Duration: 1 instruction cycle but not faster than NPS clock cycles due to pipeline flushing.

Description: Execute unconditional jump to the memory location C (C ∈ ALIT).
The program counter PC is loaded with literal C.

(59) JBS Instruction

Syntax: JBS A, B, C

Encoding: 1110a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0001c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 --

Operation: PC ← C << 2 if A[B] is set
PC ← PC + 4 if A[B] is clear

Status: —

Duration: 1 instruction cycle but if the jump is executed, it is not faster than NPS clock cycles due to pipeline flushing.

Description: Execute conditional jump to the memory location C (C ∈ ALIT).
The program counter PC is loaded with literal C, if the bit at position B (B ∈ BITLIT) of operand A (A ∈ OREG) is set.
Otherwise, if the bit is cleared, the program counter PC is incremented by the value 4.

(60) JBC Instruction

Syntax: JBC A, B, C

Encoding: 1110a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0010c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 --

Operation: PC ← C << 2 if A[B] is set
PC ← PC + 4 if A[B] is clear

Status: —

Duration: 1 instruction cycle but if the jump is executed, it is not faster than NPS clock cycles due to pipeline flushing.

Description: Execute conditional jump to the memory location C (C ∈ ALIT).
The program counter PC is loaded with literal C if the bit at position B (B ∈ BITLIT) of operand A (A ∈ OREG) is cleared.
Otherwise, if the bit is set the program counter PC is incremented by the value 4.

(61) CALL Instruction

Syntax: CALL C

Encoding: 1110--------0011c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 --

Operation: R7 ← R7 + 4;
MEM(R7[RAW+USR+1:2])[RAW+USR+1:0] ← PC + 4;
PC ← C << 2;
SP_CNT ← SP_CNT + 1

Status: EN

Duration: 2 instruction cycles but not faster than 2*NPS clock cycles due to pipeline flushing.

Description: Call subprogram at memory location C (C ∈ ALIT).
The stack pointer register R7 is incremented by the value 4.
The memory location for the top of the stack is identified by the bits 2 to RAW+1 of the stack pointer register.
After the stack pointer is incremented, the incremented value of the PC is transferred to the top of the stack.
The program counter PC is loaded with literal C.
The SP_CNT bit field inside the MCS[i]_CH[x]_CTRL register is incremented. If an overflow on the SP_CNT bit field occurs, the STK_ERR[i]_IRQ is raised.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL_STAT is set, the channel current MCS-channel is disabled by clearing the EN bit of STA.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL_STAT is set, the memory write operation of the incremented PC is discarding.

(62) RET Instruction

Syntax: RET

Encoding: 1110--------0100----------

Operation: PC ← MEM(R7[RAW+USR+1:2])[RAW+USR+1:2] << 2;
R7 ← R7 – 4;
SP_CNT ← SP_CNT – 1

Status: EN

Duration: 2 instruction cycles but not faster than 2*NPS clock cycles due to pipeline flushing.

Description: Return from subprogram.
The program counter PC is loaded with current value on the top of the stack.
Finally, the stack pointer register R7 is decremented by the value 4.
The memory location for the top of the stack is identified by the bits 2 to RAW+1 of the stack pointer register.
The SP_CNT bit field inside the MCS[i]_CH[x]_CTRL register is decremented.
If an underflow on the SP_CNT bit field occurs, the STK_ERR[i]_IRQ is raised.
If an underflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL is set, the channel current MCS-channel is disabled by clearing the EN bit of STA.

(63) JMPI Instruction

Syntax: JMPI

Encoding: 1110--------0101----------

Operation: PC ← R6[RAW+USR+1:2] << 2

Status: —

Duration: 1 instruction cycle but not faster than NPS clock cycles due to pipeline flushing.

Description: Execute indirect unconditional jump to the memory location provided in register R6.
The destination address is only defined by the bits 2 to RAW+USR+1 of R6 and the other bits are ignored.
The program counter PC is loaded with (R6[RAW+USR+1:2] << 2).

(64) JBSI Instruction

Syntax: JBSI A, B

Encoding: $1110a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0$ 0110-a_4-b_4 ----------

Operation: PC ← R6[RAW+USR+1:2] << 2 if A[B] is set
PC ← PC + 4 if A[B] is clear

Status: —

Duration: 1 instruction cycle but if the jump is executed it is not faster than NPS clock cycles due to pipeline flushing.

Description: Execute indirect conditional jump to the memory location provided in register R6. The destination address is only defined by the bits 2 to RAW+USR+1 of R6 and the other bits are ignored.
The program counter PC is loaded with (R6[RAW+USR+1:2] << 2) only if the bit at position B (B ∈ XBITLIT) of operand A (A ∈ XOREG) is set.
Otherwise, if the bit is cleared the program counter PC is incremented by the value 4.

(65) JBCI Instruction

Syntax: JBCI A, B

Encoding: 1110a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0111-a_4 -b_4 ----------

Operation: PC ← R6[RAW+USR+1:2] << 2 if A[B] is clear
PC ← PC + 4 if A[B] is set

Status: —

Duration: 1 instruction cycle but if the jump is executed it is not faster than NPS clock cycles due to pipeline flushing.

Description: Execute indirect conditional jump to the memory location provided in register R6. The destination address is only defined by the bits 2 to RAW+USR+1 of R6 and the other bits are ignored.
The program counter PC is loaded with (R6[RAW+USR+1:2] << 2) only if the bit at position B (B ∈ XBITLIT) of operand A (A ∈ XOREG) is cleared.
Otherwise, if the bit is set the program counter PC is incremented by the value 4.

(66) CALLI Instruction

Syntax: CALLI

Encoding: 1110--------0111----------

Operation: R7 ← R7 + 4;
MEM(R7[RAW+USR+1:2])[RAW+USR+1:0] ← PC + 4;
PC ← R6[RAW+USR+1:2] << 2;
SP_CNT ← SP_CNT + 1

Status: EN

Duration: 2 instruction cycles but not faster than 2*NPS clock cycles due to pipeline flushing.

Description: Call subprogram indirectly, where the register R6 is identifying the target memory location. The destination address is only defined by the bits 2 to RAW+USR+1 of R6 and the other bits are ignored.
The stack pointer register R7 is incremented by the value 4.
The memory location for the top of the stack is identified by the bits 2 to RAW+1 of the stack pointer register.
After the stack pointer is incremented, the incremented value of the PC is transferred to the top of the stack.
The program counter PC is loaded with (R6[RAW+USR+1:2] << 2), The SP_CNT bit field inside the MCS[i]_CH[x]_CTRL register is incremented.
If an overflow on the SP_CNT bit field occurs, the STK_ERR[i]_IRQ is raised.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL_STAT is set, the channel current MCS-channel is disabled by clearing the EN bit of STA.
If an overflow on the SP_CNT bit field occurs and the bit HLT_SP_OFL of register MCS[i]_CTRL_STAT is set, the memory write operation of the incremented PC is discarding.

(67) WURM Instruction

Syntax: WURM A, B, C

Encoding: $1111 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0000 c_{15} c_{14} c_{13} c_{12} c_{11} c_{10} c_9 c_8 c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$

Operation: Wait until register match.

Status: CWT

Duration: Suspends current MCS-channel. If the MCS is configured in Single Prioritization or Multiple Prioritization Scheduling Mode, the worst case latency for reactivating a prioritized MCS-channel is 2+NPS clock cycles. This is the delay between the match event of the corresponding WURM instruction and the beginning of the following MCS instruction.

Description: Suspend current MCS-channel until the following register match condition occurs: A = (B AND MASK), whereas A ∈ OREG, B ∈ OREG, AND is a bitwise AND operation with bitmask MASK. The bits 16 to 23 of MASK are set to true and the bits 0 to 15 are copied from the instructions literal C ∈ MSKLIT. If the match condition evaluates to true, the suspended MCS channel is resumed and the program counter PC is incremented by the value 4 meaning that the MCS channel continues its program. However, if the match condition is true at the beginning of the instruction execution, the instruction does not suspend the channel and the program counter PC is incremented by the value 4. At the beginning of the instruction execution the CWT bit in the register STA is always cleared. After the execution of the instruction the CWT bit is updated in order to show if the instruction finished successfully (CWT = 0) or it was canceled by the CPU (CWT = 1). This instruction can be used to wait for one or more trigger events generated by other MCS-channels or the CPU. In this case register B is the trigger register STRG, A is a general purpose register holding the bits with the trigger condition to wait for and C is the bitmask that enables trigger bits of interest. The trigger bits can be set by other MCS channels with a write access (e.g. using a MOVL instruction) to the STRG register or the CPU with a write access to the MCS[i]_STRG register. The trigger bits are not cleared automatically by hardware after resuming an MCS-channel, but they have to be cleared explicitly with a write access to the register CTRG by the MCS-channel or with a write access to the register MCS[i]_CTRG by the CPU.

NOTE

More than one channel can wait for the same trigger bit to continue.

(68) WURMX Instruction

Syntax: WURMX A, B

Encoding: $1111 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0001 \text{---} b_4$ ----------

Operation: Wait until extended register match.

Status: CWT

Duration: Suspends current MCS-channel. If the MCS is configured in Single Prioritization or Multiple Prioritization Scheduling Mode, the worst case latency for reactivating a prioritized MCS-channel is 2+NPS clock cycles. This is the delay between the match event of the corresponding WURMX instruction and the beginning of the following MCS instruction.

Description: Suspend current MCS-channel until the following register match condition occurs: A = B AND R6, whereas A ∈ OREG, B ∈ WXREG, and AND is a bitwise AND operation. If the match condition evaluates to true, the suspended MCS channel is resumed and the program counter PC is incremented by the value 4 meaning that the MCS channel continues its program. However, if the match condition is true at the beginning of the instruction execution, the instruction does not suspend the channel and the program counter PC is incremented by the value 4. At the beginning of the instruction execution the CWT bit in the register STA is always cleared. After the execution of the instruction the CWT bit is updated in order to show if the instruction finished successfully (CWT = 0) or it was canceled by the CPU (CWT = 1).

(69) WURCX Instruction

Syntax: WURCX A, B

Encoding: $1111 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0010 \text{---} b_4$ ----------

Operation: Wait until extended register change.

Status: CWT

Duration: Suspends current MCS-channel. If the MCS is configured in Single Prioritization or Multiple Prioritization Scheduling Mode, the worst case latency for reactivating a prioritized MCS-channel is 2+NPS clock cycles. This is the delay between the match event of the corresponding WURCX instruction and the beginning of the following MCS instruction.

Description: Suspend current MCS-channel until the following register change condition occurs: A 6= B AND R6, whereas A ∈ OREG, B ∈ WXREG, and AND is a bitwise AND operation. If the change condition evaluates to true, the suspended MCS channel is resumed and the program counter PC is incremented by the value 4 meaning that the MCS channel continues its program. However, if the change condition is true at the beginning of the instruction execution, the instruction does not suspend the channel and the program counter PC is incremented by the value 4. At the beginning of the instruction execution the CWT bit in the register STA is always cleared. After the execution of the instruction the CWT bit is updated in order to show if the instruction finished successfully (CWT = 0) or it was canceled by the CPU (CWT = 1).
The WURCX instruction can be used for observation of volatile registers (e.g. register DSTAX) in order to react on status signal changes.

(70) WUCE Instruction

Syntax: WUCE A, B

Encoding: $1111 a_3 a_2 a_1 a_0 b_3 b_2 b_1 b_0 0011$----------

Operation: Wait until cyclic event.

Status: CWT

Duration: Suspends current MCS-channel. If the MCS is configured in Single Prioritization or Multiple Prioritization Scheduling Mode, the worst case latency for reactivating a prioritized MCS-channel is 2+NPS clock cycles. This is the delay between the match event of the corresponding WUCE instruction and the beginning of the following MCS instruction.

Description: Suspend current MCS-channel until a cyclic event compare matches. The meaning of a cyclic event is described in **Section 38.5.4.1, Cyclic Event Compare**. The WUCE instruction can be used to synchronize an MCS program to a cyclic event generated by a TBU channel. If the event is in the future, the MCS channel suspends until the event occurs. If the event is in the past, the WUCE instruction is finished immediately. The cyclic event compare is used to detect time base overflows and to guarantee, that a compare match event can be set up for the future even when the time base will first overflow and then reach the compare value.

NOTE

For a correct behavior of this cyclic event compare, the compare value must not be specified larger/smaller than half of the range of the total time base value ($7FFFFF_H$).

The actual implementation of the WUCE implementation simply performs the subtraction B–A with each clock cycle and it suspends the MCS channel as long as bit W-1 of the subtraction result is set. If the bit W-1 of the subtraction result is cleared the MCS channel is resumed immediately. In order to setup a WUCE instruction correctly, the counting direction of the TBU channel has to be considered. If the TBU channel is counting forward (incrementing), the operand A (A ∈ OREG) must refer the compare value and operand B (B ∈ OREG) must refer the desired TBU counter register (e.g. TBU_TS0). On the other hand, if the TBU channel is counting backward (decrementing), the operand A refers the desired TBU counter register and operand B refers the compare value. If the comparison condition evaluates to true, the suspended MCS channel is resumed and the program counter PC is incremented by the value 4 meaning that the MCS channel continues its program. However, if the condition is true at the beginning of the instruction execution, the instruction does not suspend the channel and the program counter PC is incremented by the value 4. At the beginning of the instruction execution the CWT bit in the register STA is always cleared. After the execution of the instruction the CWT bit is updated in order to show if the instruction finished successfully (CWT = 0) or it was canceled by the CPU (CWT = 1).

(71) NOP Instruction

Syntax: NOP

Encoding: 0000----------

Operation: —

Status: —

Duration: 1 instruction cycle

Description: No operation is performed.
The program counter PC is incremented by the value 4.

38.19.9 MCS Internal Register Overview

Table 38.245 Register list

Symbol	Register Name	Details in Section
R[y]	General Purpose Register y	**38.19.10.1**
RS[y]	Mirror of succeeding channels register R[y]	**38.19.10.2**
STA	Control and status register	**38.19.10.3**
ACB	ARU Control Bit Register	**38.19.10.4**
CTRG	Clear Trigger Bits Register	**38.19.10.5**
STRG	Set Trigger Bits Register	**38.19.10.6**
TBU_TS0	TBU Timestamp TS0 Register	**38.19.10.7**
TBU_TS1	TBU Timestamp TS1 Register	**38.19.10.8**
TBU_TS2	TBU timestamp TS2 register	**38.19.10.9**
MHB	Memory high bits register	**38.19.10.10**
GMI0	GTM module interrupt 0 register	**38.19.10.11**
GMI1	GTM module interrupt 1 register	**38.19.10.12**
DSTA	DPLL status register	**38.19.10.13**
DSTAX	DPLL extended status register	**38.19.10.14**
AXIMI	AXI Master interrupt register	**38.19.10.15**

38.19.10 MCS Internal Register Description

This section describes the MCS internal registers that can be directly addressed with the MCS instruction set. Many of the registers can also be addressed by the CPU but with another Register Label (for details see **Section 38.19.12, MCS Configuration Registers Description**). Some of the internal registers are also shared between neighboring MCS channels.

38.19.10.1 R[y]

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

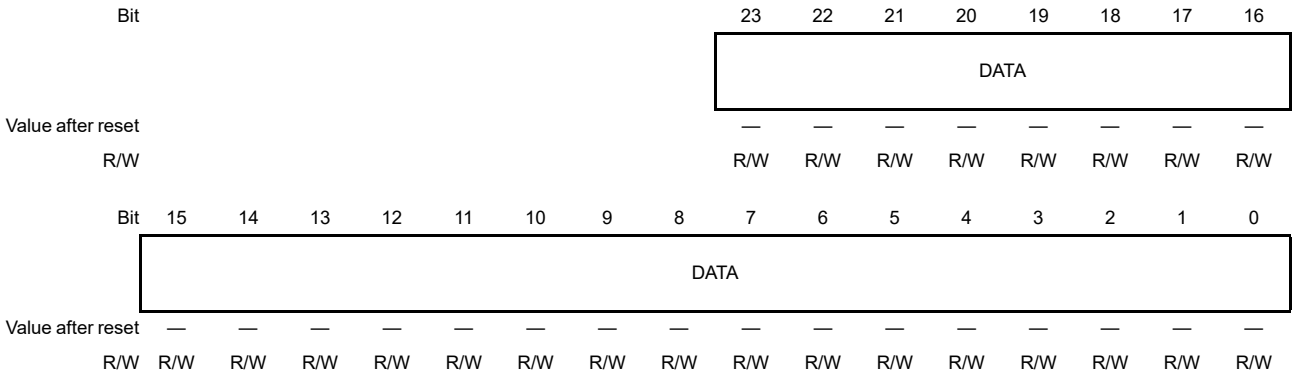


Table 38.246 R[y] Register Contents

Bit Position	Bit Name	Function
23 to 0	DATA	Data field of general purpose register.

Note: Register R4 is also used as destination register of upper multiplication result from instructions MULU and MULS.
Register R5 is also used as offset register for the instructions MRDIO and MWRIO.
Register R6 is also used as a mask register for the instruction WURMX and WURCX.
Register R6 is also used as address destination register for the instructions JMPI, JBSI, JBCI, and CALLI.
Register R6 used also as index/address register for indirect ARU addressing instructions.
Register R7 is also used as stack pointer register, if stack operations are used in the MCS micro program.

38.19.10.2 RS[y]

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	DATA							
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.247 RS[y] Register Contents

Bit Position	Bit Name	Function
23 to 0	DATA	Data field of general purpose register.

Note 1. The register RS[y] (with y = 0 ... 7) mirrors the internal general purpose register R[y] of the succeeding MCS channel. The successor of MCS channel T-1 is MCS channel 0.

Note 2. The registers RS[y] can only be accessed if bit EN_XOREG of register MCS[i]_CTRL_STAT is set.

38.19.10.3 STA

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	—	—	—	—	—	SP_CNT		
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SAT	CWT	CAT	N	V	Z	CY	MCA	ERR	IRQ	EN
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.248 STA Register Contents (1/4)

Bit Position	Bit Name	Function
23 to 19	Reserved	When read, an undefined value is read. When writing, always write 0.
18 to 16	SP_CNT	Stack pointer counter value. **NOTE** Actual stack depth of channel. The bit field is incremented on behalf of a CALL or PUSH instruction and decremented on behalf of a RET or POP instruction. The MCS channel STK_ERR_IRQ is raised, when an overflow or underflow is detected on this bit field.
15 to 11	Reserved	When read, an undefined value is read. When writing, always write 0.
10	SAT	Successful ARU transfer bit. 0: Latest ARU data transfer failed. 1: Latest ARU data transfer finished successfully. **NOTE** This bit is always updated after the execution of ARU instructions in order to show if the ARU data transfer was successful or not. In the case of non-blocking ARU instructions (NARD, NARDI) a cleared SAT flag signalizes that the data source has no data available and in the case of blocking ARU instructions (ARD, ARDI, AWR, AWRI) a cleared SAT flag signalizes that the data transfer was canceled by the CPU.
9	CWT	Cancel WURM instruction bit. 0: No cancellation request for waiting instruction. 1: CPU requested cancellation for waiting instruction. **NOTE** This bit is always cleared at the beginning of the execution of a WURM, WURMX, or WURCX instruction. A cancellation request can be applied by a write request to register MCS[i]_CWT. If the MCS program needs to detect a cancellation request it should evaluate the bit CWT immediately after the cancelled instruction.

Table 38.248 STA Register Contents (2/4)

Bit Position	Bit Name	Function
8	CAT	Cancel ARU transfer bit. 0: No cancellation request for ARU transfer. 1: CPU requested cancellation for ARU transfer. **NOTE** This bit is always cleared at the beginning of the execution of a blocking ARU instruction. A cancellation request can be applied by a write request to register MCS[i]_CAT. If the MCS program needs to detect a cancellation request it should evaluate the bit CAT immediately after the cancelled instruction.
7	N	Negative bit. The negative bit is updated by arithmetic instructions in order to indicate a negative result.
6	V	Overflow bit. The overflow bit is updated by arithmetic instructions in order to indicate a signed under/overflow.
5	Z	Zero bit. The zero bit is updated by several arithmetic, logic and data transfer instructions to indicate a result of zero.
4	CY	Carry bit. The carry bit is updated by several arithmetic and logic instructions. In arithmetic operations, the carry bit indicates an unsigned under/overflow.
3	MCA	MON Activity signalling for MCS channel. 0: No activity signalled to sub module MON. 1: Activity signalled to sub module MON. **NOTE** When this bit is set the corresponding channel in the MON sub module register MON_ACTIVITY is set (see **Section 38.26.8.2, MON_ACTIVITY_0**. This bit is automatically cleared after writing it by the MCS channel program.

Table 38.248 STA Register Contents (3/4)

Bit Position	Bit Name	Function
2	ERR	Set Error Signal. READ access 0: No Error occurred 1: Error occurred. WRITE access 0: No action. 1: Set Error bit. **NOTES** 1. The ERR bit of an MCS-channel reflects an Error status that may be caused by one of the following conditions: – MCS-channel sets the ERR bit by software (e.g. with instruction MOVL STA, 0x4) – ECC RAM Error occurred while accessing the connected RAM (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT) – Decoding an instruction with an invalid opcode (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT) – A memory address range overflow occurred (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT) – Division by zero resulting from a DIVU or DIVS instruction (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT). – MCS channel wants to write to a GPR that is write protected by register MCS[i]_REG_PROT (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT). – MCS channel wants to write to a protected memory range defined by an address range protector (ARP) of the sub module CCM (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT). – MCS channel performs an invalid AEI bus master access while the bit field HLT_AEIM_ERR of register MCS[i]_CTRL_-STAT is set (also disables the MCS-channel by clearing bit EN and the first error occurred updates bit field ERR_SRC_ID of register MCS[i]_CTRL_STAT) 2. If the ERR bit is set due to a memory address range overflow any read or write access to the RAM is blocked. 3. If the GTM includes a MON sub module, the ERR signal is always captured by this module. 4. An MCS-channel can set the error bit by writing value 1 to bit ERR. Writing a value 0 to this bit does not cancel the error signal, and thus has no effect. In Addition, writing a value 1 to ERR always triggers the ERR interrupt, independently from the current state of the error signal. 5. The ERR bit can only be cleared by CPU, by writing a 1 to the MCS[i]_ERR register (see **Section 38.19.12.18, MCS[i]_ERR**). 6. An MCS-channel can read the ERR bit in order to determine the current state of the error signal. The MCS-channel reads a value 1 if an ERR occurred previously, but not cleared by CPU. If an MCS-channel reads a value 0 no error was set or it has been cleared by CPU.

Table 38.248 STA Register Contents (4/4)

Bit Position	Bit Name	Function
1	IRQ	Trigger IRQ. READ access 0: No interrupt pending in MCS-channel x. 1: Interrupt is pending in MCS-channel x. WRITE access 0: No action. 1: Trigger interrupt. **NOTES** 1. An MCS-channel triggers an IRQ by writing value 1 to bit IRQ. Writing a value 0 to this bit does not cancel the IRQ, and thus has no effect. 2. This bit mirrors bit 0 of the register MCS[i]_CH[x]_IRQ_NOTIFY. 3. The IRQ bit can only be cleared by CPU, by writing a 1 to the corresponding MCS[i]_CH[x]_IRQ_NOTIFY register (see **Section 38.19.12.6, MCS[i]_CH[x]_IRQ_NOTIFY**). 4. An MCS-channel can read the IRQ bit in order to determine the current state of the IRQ handling. The MCS-channel reads a value 1 if an IRQ was released but not cleared by CPU. If an MCS-channel reads a value 0 no IRQ was released or it has been cleared by CPU. 5. If NPS > 5 and an MCS program triggers the IRQ (e.g. by MOVL STA, 0x2) the actual interrupt event is delayed by NPS–5 clock cycles, which means that an immediate read of the interrupt notify flag (e.g. by MOV R2, STA) may signalize the state of the IRQ bit before the trigger.
0	EN	Enable current MCS-channel. 0: Disable current MCS-channel. 1: Enable current MCS-channel.

Note: Writing to bits of the register STA with instructions that do implicitly a read−modify−write operation (e.g. "ANDL STA, 0xFFFFFE", "OR STA, R0", or "SETB STA, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results.
A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.4 ACB

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ACB4	ACB3	ACB2	ACB1	ACB0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.249 ACB Register Contents

Bit Position	Bit Name	Function
23 to 5	Reserved	When read, an undefined value is read. When writing, always write 0.
4	ACB4	ARU Control bit 4. **NOTE** This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit 48 of the ARU word.
3	ACB3	ARU Control bit 3. **NOTE** This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit 48 of the ARU word.
2	ACB2	ARU Control bit 2. **NOTE** This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit 48 of the ARU word.
1	ACB1	ARU Control bit 1. **NOTE** This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit 48 of the ARU word.
0	ACB0	ARU Control bit 0. **NOTE** This bit is updated by each ARU read access and its value is sent to ARU by each ARU write access on bit 48 of the ARU word.

38.19.10.5 CTRG

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: 0000 0000_H

Bit	23	22	21	20	19	18	17	16
	TRG23	TRG22	TRG21	TRG20	TRG19	TRG18	TRG17	TRG16
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG15	TRG14	TRG13	TRG12	TRG11	TRG10	TRG9	TRG8	TRG7	TRG6	TRG5	TRG4	TRG3	TRG2	TRG1	TRG0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.250 CTRG Register Contents (1/5)

Bit Position	Bit Name	Function
23	TRG23	Trigger bit 23 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
22	TRG22	Trigger bit 22 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
21	TRG21	Trigger bit 21 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
20	TRG20	Trigger bit 20 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
19	TRG19	Trigger bit 19 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit

Table 38.250 CTRG Register Contents (2/5)

Bit Position	Bit Name	Function
18	TRG18	Trigger bit 18 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
17	TRG17	Trigger bit 17 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
16	TRG16	Trigger bit 16 READ access: 0: Trigger is cleared 1: Trigger is set WRITE access: 0: No action 1: Clear trigger bit
15	TRG15	Trigger bit 15 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
14	TRG14	Trigger bit 14 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
13	TRG13	Trigger bit 13 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit

Table 38.250 CTRG Register Contents (3/5)

Bit Position	Bit Name	Function
12	TRG12	Trigger bit 12 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
11	TRG11	Trigger bit 11 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
10	TRG10	Trigger bit 10 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
9	TRG9	Trigger bit 9 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
8	TRG8	Trigger bit 8 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit

Table 38.250 CTRG Register Contents (4/5)

Bit Position	Bit Name	Function
7	TRG7	Trigger bit 7 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
6	TRG6	Trigger bit 6 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
5	TRG5	Trigger bit 5 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
4	TRG4	Trigger bit 4 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
3	TRG3	Trigger bit 3 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit

Table 38.250 CTRG Register Contents (5/5)

Bit Position	Bit Name	Function
2	TRG2	Trigger bit 2 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
1	TRG1	Trigger bit 1 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit
0	TRG0	Trigger bit 0 READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit

NOTES

1. The trigger bits TRGx are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit.
2. Besides setting the trigger bits with register STRG/MCS[i]_STRG, the k-th trigger bit TRGk (with k < 16) can also be set by the external capture event that is enabled by the k-th bit of register CCM[i]_EXT_CAP_EN. If bit k bit is disabled, the k-th trigger bit TRGk can only be set by MCS or CPU.
3. The result of a read access to this register differs in dependency of the bit field EN_TIM_FOUT of register MCS[i]_CTRL_STAT.

CAUTION

Writing to bits of the register CTRG with instructions that do implicitly a read-modify-write operation (e.g. "OR CTRG, R0" or "SETB CTRG, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.6 STRG

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: 0000 0000$_H$

Bit	23	22	21	20	19	18	17	16
	TRG23	TRG22	TRG21	TRG20	TRG19	TRG18	TRG17	TRG16
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG15	TRG14	TRG13	TRG12	TRG11	TRG10	TRG9	TRG8	TRG7	TRG6	TRG5	TRG4	TRG3	TRG2	TRG1	TRG0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.251 STRG Register Contents (1/4)

Bit Position	Bit Name	Function
23	TRG23	Trigger bit 23 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
22	TRG22	Trigger bit 22 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
21	TRG21	Trigger bit 21 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
20	TRG20	Trigger bit 20 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
19	TRG19	Trigger bit 19 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit

Table 38.251 STRG Register Contents (2/4)

Bit Position	Bit Name	Function
18	TRG18	Trigger bit 18 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
17	TRG17	Trigger bit 17 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
16	TRG16	Trigger bit 16 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
15	TRG15	Trigger bit 15 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
14	TRG14	Trigger bit 14 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
13	TRG13	Trigger bit 13 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
12	TRG12	Trigger bit 12 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
11	TRG11	Trigger bit 11 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit

Table 38.251 STRG Register Contents (3/4)

Bit Position	Bit Name	Function
10	TRG10	Trigger bit 10 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
9	TRG9	Trigger bit 9 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
8	TRG8	Trigger bit 8 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
7	TRG7	Trigger bit 7 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
6	TRG6	Trigger bit 6 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
5	TRG5	Trigger bit 5 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
4	TRG4	Trigger bit 4 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
3	TRG3	Trigger bit 3 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit

Table 38.251 STRG Register Contents (4/4)

Bit Position	Bit Name	Function
2	TRG2	Trigger bit 2 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
1	TRG1	Trigger bit 1 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit
0	TRG0	Trigger bit 0 READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit

NOTES

1. The trigger bits TRGx are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit.
2. Besides setting the trigger bits with register STRG/MCS[i]_STRG, the k-th trigger bit TRGk (with k < 16) can also be set by the external capture event that is enabled by the k-th bit of register CCM[i]_EXT_CAP_EN. If bit k bit is disabled, the k-th trigger bit TRGk can only be set by MCS or CPU.
3. Writing to bits of the register STRG with instructions that do implicitly a read-modify-write operation (e.g. "OR STRG, R0" or "SETB STRG, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.7 TBU_TS0

Access: This register is a read-only register that can be read in 24-bit units.

Address: —

Value after reset: Undefined

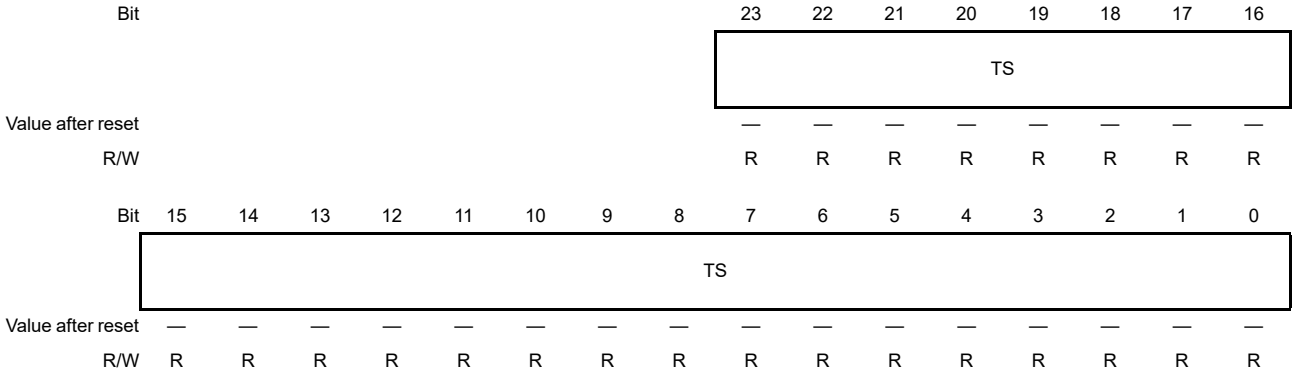


Table 38.252 TBU_TS0 Register Contents

Bit Position	Bit Name	Function
23 to 0	TS	Current TBU time stamp 0.

38.19.10.8 TBU_TS1

Access: This register is a read-only register that can be read in 24-bit units.

Address: —

Value after reset: Undefined

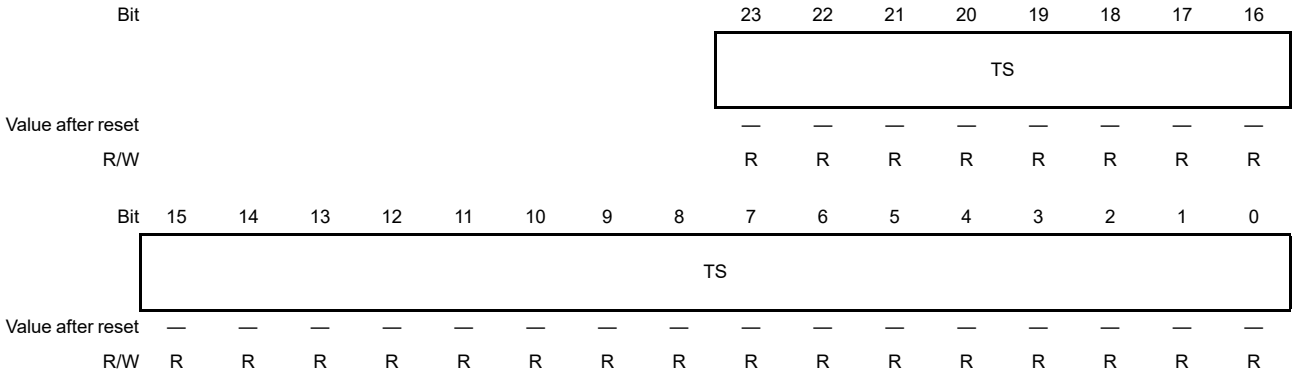


Table 38.253 TBU_TS1 Register Contents

Bit Position	Bit Name	Function
23 to 0	TS	Current TBU time stamp 1.

38.19.10.9 TBU_TS2

Access: This register is a read-only register that can be read in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	TS							
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.254 TBU_TS2 Register Contents

Bit Position	Bit Name	Function
23 to 0	TS	Current TBU time stamp 2.

38.19.10.10 MHB

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.255 MHB Register Contents

Bit Position	Bit Name	Function
23 to 8	Reserved	When read, an undefined value is read. When writing, always write 0.
7 to 0	DATA	High Byte of a memory transfer.

38.19.10.11 GMI0

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	ATOM_CH7_IRQ	ATOM_CH6_IRQ	ATOM_CH5_IRQ	ATOM_CH4_IRQ	ATOM_CH3_IRQ	ATOM_CH2_IRQ	ATOM_CH1_IRQ	ATOM_CH0_IRQ
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM_CH27_IRQ	TOM_CH26_IRQ	TOM_CH25_IRQ	TOM_CH24_IRQ	TOM_CH23_IRQ	TOM_CH22_IRQ	TOM_CH21_IRQ	TOM_CH20_IRQ	TIM_CH7_IRQ	TIM_CH6_IRQ	TIM_CH5_IRQ	TIM_CH4_IRQ	TIM_CH3_IRQ	TIM_CH2_IRQ	TIM_CH1_IRQ	TIM_CH0_IRQ
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.256 GMI0 Register Contents

Bit Position	Bit Name	Function
23 to 16	ATOM_CH[x]_IRQ	IRQ signal ATOM[i]_CH[x]_IRQ. READ access: 0: IRQ signal ATOM[i]_CH[x]_IRQ is cleared 1: IRQ signal ATOM[i]_CH[x]_IRQ is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
15 to 8	TOM_CH[2*x]_IRQ	IRQ signal TOM[i]_CH[2*x]_IRQ or TOM[i]_CH[2*x+1]_IRQ. READ access: 0: Logical OR conjunction of IRQ signals T0M[i]_CH[2*x]_IRQ and T0M[i]_CH[2*x+1]_IRQ is false 1: Logical OR conjunction of IRQ signals T0M[i]_CH[2*x]_IRQ and T0M[i]_CH[2*x+1]_IRQ is true WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
7 to 0	TIM_CH[x]_IRQ	IRQ signal TIM[i]_CH[x]_IRQ. READ access: 0: IRQ signal TIM[i]_CH0_IRQ is cleared 1: IRQ signal TIM[i]_CH0_IRQ is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs

38.19.10.12 GMI1

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	MCS0_CH7_IRQ	MCS0_CH6_IRQ	MCS0_CH5_IRQ	MCS0_CH4_IRQ	MCS0_CH3_IRQ	MCS0_CH2_IRQ	MCS0_CH1_IRQ	MCS0_CH0_IRQ
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTA_IP1_CH7_IRQ	TTA_IP1_CH6_IRQ	TTA_IP1_CH5_IRQ	TTA_IP1_CH4_IRQ	TTA_IP1_CH3_IRQ	TTA_IP1_CH2_IRQ	TTA_IP1_CH1_IRQ	TTA_IP1_CH0_IRQ	MCS_IP1_CH7_IRQ	MCS_IP1_CH6_IRQ	MCS_IP1_CH5_IRQ	MCS_IP1_CH4_IRQ	MCS_IP1_CH3_IRQ	MCS_IP1_CH2_IRQ	MCS_IP1_CH1_IRQ	MCS_IP1_CH0_IRQ
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.257 GMI1 Register Contents

Bit Position	Bit Name	Function
23 to 16	MCS0_CH[x]_IRQ	MCS0_CH0 IRQs. READ access: 0: IRQ signal MCS[i+1]_CH0_IRQ is cleared 1: IRQ signal MCS[i+1]_CH0_IRQ is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
15 to 8	TTA_IP1_CH[x]_IRQ	Neighboring TIM, TOM, ATOM IRQs. READ access: 0: Logical OR conjunction of IRQ signals TIM[i+1]_CH[x]_IRQ, TOM[i+1]_CH[2*x]_IRQ, TOM[i+1]_CH[2*x+1]_IRQ, and ATOM[i+1]_CH[x]_IRQ is false 1: Logical OR conjunction of IRQ signals TIM[i+1]_CH[x]_IRQ, TOM[i+1]_CH[2*x]_IRQ, TOM[i+1]_CH[2*x+1]_IRQ, and ATOM[i+1]_CH[x]_IRQ is true WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
7 to 0	MCS_IP1_CH[x]_IRQ	IRQ signal MCS[i+1]_CH[x]_IRQ. READ access: 0: IRQ signal MCS[i+1]_CH[x]_IRQ is cleared 1: IRQ signal MCS[i+1]_CH[x]_IRQ is set WRITE access: 0: No action 1:Trigger hw_clear signal on the connected IRQs

CAUTION

Writing to bits of the register GMI1 with instructions that do implicitly a read-modify-write operation (e.g. "OR GMI1, R0" or "SETB GMI1, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.13 DSTA

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	—	—	—	CDTI	SISI	SASI	TISI	TASI
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STA_S								STA_T							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.258 DSTA Register Contents (1/2)

Bit Position	Bit Name	Function
23 to 21	Reserved	When read, an undefined value is read. When writing, always write 0.
20	CDTI	Calculation of trigger duration interrupt. READ access: 0: IRQ signal CDTI of DPLL is cleared 1: IRQ signal CDTI of DPLL is set WRITE access: 0: No Action 1:Trigger hw_clear signal on the connected IRQs
19	SISI	State Inactive Slope Interrupt. READ access: 0: IRQ signal SISI of DPLL is cleared 1: IRQ signal SISI of DPLL is set WRITE access: 0: No Action 1:Trigger hw_clear signal on the connected IRQs
18	SASI	State Active Slope Interrupt. READ access: 0: IRQ signal SASI of DPLL is cleared 1: IRQ signal SASI of DPLL is set WRITE access: 0: No Action 1:Trigger hw_clear signal on the connected IRQs
17	TISI	State Inactive Slope Interrupt. READ access: 0: IRQ signal TISI of DPLL is cleared 1: IRQ signal TISI of DPLL is set WRITE access: 0: No Action 1:Trigger hw_clear signal on the connected IRQs
16	TASI	Trigger Active Slope Interrupt. READ access: 0: IRQ signal TASI of DPLL is cleared 1: IRQ signal TASI of DPLL is set WRITE access: 0: No Action 1:Trigger hw_clear signal on the connected IRQs
15 to 8	STA_S	Status State FSM. Actual status of DPLL State FSM. The description of the FSM states can be found in **Section 38.22.18.48, DPLL_STA**.

Table 38.258 DSTA Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 0	STA_T	Status Trigger FSM. Actual status of DPLL Trigger FSM. The description of the FSM states can be found in **Section 38.22.18.48, DPLL_STA**.

NOTES

1. This register is only implemented in MCS instance 0. In other MCS instances, a read access always returns 0 and a write access is always ignored.
2. Writing to bits of the register DSTA with instructions that do implicitly a read-modify-write operation (e.g. "OR DSTA, R0" or "SETB DSTA, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.14 DSTAX

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	—	—	—	CDTI	SISI	SASI	TISI	TASI
Value after reset	—	—	—	—	—	—	—	—
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	INC_CNT2_FLAG	INC_CNT1_FLAG	STA_FLAG_S	STA_FLAG_T
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 38.259 DSTAX Register Contents (1/2)

Bit Position	Bit Name	Function
23 to 21	Reserved	When read, an undefined value is read. When writing, always write 0.
20	CDTI	Calculation of trigger duration interrupt. READ access: 0: IRQ signal CDTI of DPLL is cleared 1: IRQ signal CDTI of DPLL is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
19	SISI	State Inactive Slope Interrupt. READ access: 0: IRQ signal SISI of DPLL is cleared 1: IRQ signal SISI of DPLL is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
18	SASI	State Active Slope Interrupt. READ access: 0: IRQ signal SASI of DPLL is cleared 1: IRQ signal SASI of DPLL is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
17	TISI	State Inactive Slope Interrupt. READ access: 0: IRQ signal TISI of DPLL is cleared 1: IRQ signal TISI of DPLL is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
16	TASI	State Active Slope Interrupt. READ access: 0: IRQ signal TASI of DPLL is cleared 1: IRQ signal TASI of DPLL is set WRITE access: 0: No action 1: Trigger hw_clear signal on the connected IRQs
15 to 4	Reserved	When read, an undefined value is read. When writing, always write 0.

Table 38.259 DSTAX Register Contents (2/2)

Bit Position	Bit Name	Function
3	INC_CNT2_FLAG	DPLL INC_CNT2 Flag. DPLL status state flag as described in bit field definition INC_CNT2_FLAG of register DPLL_STA_FLAG (**Section 38.22.18.54, DPLL_STA_FLAG**).
2	INC_CNT1_FLAG	DPLL INC_CNT1 Flag. DPLL status state flag as described in bit field definition INC_CNT1_FLAG of register DPLL_STA_FLAG (**Section 38.22.18.54, DPLL_STA_FLAG**)
1	STA_FLAG_S	DPLL status state flag. DPLL status state flag as described in bit field definition STA_FLAG_S of register DPLL_STA_FLAG (**Section 38.22.18.54, DPLL_STA_FLAG**).
0	STA_FLAG_T	DPLL status trigger flag. DPLL status trigger flag as described in bit field definition STA_FLAG_T of register DPLL_STA_FLAG (**Section 38.22.18.54, DPLL_STA_FLAG**).

NOTES

1. This register is only implemented in MCS instance 0. In other MCS instances, a read access always returns 0 and a write access is always ignored.
2. Writing to bits of the register DSTAX with instructions that do implicitly a read-modify-write operation (e.g. "OR DSTAX, R0" or "SETB DSTAX, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.10.15 AXIMI

Access: This register can be read or written in 24-bit units.

Address: —

Value after reset: Undefined

Bit	23	22	21	20	19	18	17	16
	AEIM_AXIM_IRQ23	AEIM_AXIM_IRQ22	AEIM_AXIM_IRQ21	AEIM_AXIM_IRQ20	AEIM_AXIM_IRQ19	AEIM_AXIM_IRQ18	AEIM_AXIM_IRQ17	AEIM_AXIM_IRQ16
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEIM_AXIM_IRQ15	AEIM_AXIM_IRQ14	AEIM_AXIM_IRQ13	AEIM_AXIM_IRQ12	AEIM_AXIM_IRQ11	AEIM_AXIM_IRQ10	AEIM_AXIM_IRQ9	AEIM_AXIM_IRQ8	AEIM_AXIM_IRQ7	AEIM_AXIM_IRQ6	AEIM_AXIM_IRQ5	AEIM_AXIM_IRQ4	AEIM_AXIM_IRQ3	AEIM_AXIM_IRQ2	AEIM_AXIM_IRQ1	AEIM_AXIM_IRQ0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.260 AXIMI Register Contents

Bit Position	Bit Name	Function
23 to 0	AEIM_AXIM_IRQ[x]	IRQ signal of AXI bus master slot x. READ access: 0: IRQ signal AEIM_AXIM_IRQ[x] is cleared 1: IRQ signal AEIM_AXIM_IRQ[x] is set WRITE access: 0: No action 1: Trigger signal AEIM_AXIM_IRQ_CLR[i]

CAUTION

Writing to bits of the register AXIMI with instructions that do implicitly a read-modify-write operation (e.g. "OR AXIMI, R0" or "SETB AXIMI, R0") is dangerous, since writing back the possibly modified content of the read access (which reflects status information) may cause undesirable results. A secure way for writing to bits of this register is to use instructions that do not read the content (e.g. use instructions MOVL).

38.19.11 MCS Configuration Registers Overview

The MCS Configuration registers of the MCS module are accessible by the AEI bus interface. Some of these registers simply mirror MCS Internal registers to the AEI. Details can be found in the table below and in the individual register descriptions.

Table 38.261 Register list

Symbol	Register Name	Details in Section
MCS[i]_CH[x]_CTRL	MCSi channel x control register	**38.19.12.1**
MCS[i]_CH[x]_ACB	MCSi channel x ARU control Bit register	**38.19.12.4**
MCS[i]_CH[x]_MHB	MCSi channel x memory high byte register	**38.19.12.5**
MCS[i]_CH[x]_PC	MCSi channel x program counter register	**38.19.12.2**
MCS[i]_CH[x]_R[y]	MCSi channel x general purpose register y	**38.19.12.3**
MCS[i]_CH[x]_IRQ_NOTIFY	MCSi channel x interrupt notification register	**38.19.12.6**
MCS[i]_CH[x]_IRQ_EN	MCSi channel x interrupt enable register	**38.19.12.7**
MCS[i]_CH[x]_IRQ_FORCINT	MCSi channel x force interrupt register	**38.19.12.8**
MCS[i]_CH[x]_IRQ_MODE	MCSi channel x IRQ mode configuration register	**38.19.12.9**
MCS[i]_CH[x]_EIRQ_EN	MCSi channel x error interrupt enable register	**38.19.12.10**
MCS[i]_CTRL_STAT	MCSi control and status register	**38.19.12.11**
MCS[i]_REG_PROT	MCSi write protection register	**38.19.12.12**
MCS[i]_CTRG	MCSi clear trigger control register	**38.19.12.13**
MCS[i]_STRG	MCSi set trigger control register	**38.19.12.14**
MCS[i]_RESET	MCSi reset register	**38.19.12.15**
MCS[i]_CAT	MCSi cancel ARU transfer instruction	**38.19.12.16**
MCS[i]_CWT	MCSi cancel waiting instruction	**38.19.12.17**
MCS[i]_ERR	MCSi error register	**38.19.12.18**
MCS[i]_MEM	MCSi memory region	**38.19.12.19**

38.19.12 MCS Configuration Registers Description

38.19.12.1 MCS[i]_CH[x]_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0020_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SP_CNT		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SAT	CWT	CAT	N	V	Z	CY	—	ERR	IRQ	EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 38.262 MCS[i]_CH[x]_CTRL Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	SP_CNT	Stack pointer counter value. Actual stack depth of channel. The bit field is incremented on behalf of a CALL or PUSH instruction and decremented on behalf of a RET or POP instruction. The MCS channel STK_ERR_IRQ is raised, when an overflow or underflow is detected on this bit field.
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	SAT	Successful ARU transfer bit. **NOTE** This bit is read only and it mirrors the internal state of the ARU transfer status flag SAT.
9	CWT	Cancel WURM instruction state. READ access: 0: No cancellation request for waiting instruction 1: CPU requested cancellation for waiting instruction **NOTES** 1. This bit is always cleared at the beginning of the execution of a WURM, WURMX, or WURCX instruction. A cancellation request can be applied by a write request to register MCS[i]_CWT. If the MCS program needs to detect a cancellation request it should evaluate the bit CWT immediately after the canceled instruction. 2. This bit is read only and it mirrors the internal cancel WURM instruction status flag CWT.

Table 38.262 MCS[i]_CH[x]_CTRL Register Contents (2/3)

Bit Position	Bit Name	Function
8	CAT	Cancel ARU transfer state. 0: No cancellation request for ARU transfer 1: CPU requested cancellation for ARU transfer **NOTES** 1. This bit is always cleared at the beginning of the execution of a blocking ARU instruction. A cancellation request can be applied by a write request to register MCS[i]_CAT. If the MCS program needs to detect a cancellation request it should evaluate the bit CAT immediately after the canceled instruction. 2. This bit is read only and it mirrors the internal state of the ARU transfer status flag CAT.
7	N	Negative bit state. **NOTE** This bit is read only and it mirrors the internal zero flag N.
6	V	Overflow bit state. **NOTE** This bit is read only and it mirrors the internal carry flag V.
5	Z	Zero bit state. **NOTE** This bit is read only and it mirrors the internal zero flag Z.
4	CY	Carry bit state. **NOTE** This bit is read only and it mirrors the internal carry flag CY.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERR	Error state. 0: No error signal pending in MCS-channel x 1: Error signal is pending in MCS-channel x **NOTE** This bit is read only and it mirrors the internal error state.
1	IRQ	Interrupt state. 0: No interrupt pending in MCS-channel x 1: Interrupt is pending in MCS-channel x **NOTE** This bit is read only and it mirrors the internal IRQ state.

Table 38.262 MCS[i]_CH[x]_CTRL Register Contents (3/3)

Bit Position	Bit Name	Function
0	EN	Enable/Disable Request of MCS-channel x. READ access: 0: MCS channel x is disabled 1: MCS channel x is enabled WRITE access: 0: Request for disabling MCS-channel x 1: Request for enabling MCS-channel x **NOTES** 1. Enabling or disabling of an MCS-channel is synchronized to the ending of an instruction and thus it may take several clock cycles, e.g. active memory transfers or pending WURM instruction have to be finished before disabling the MCS-channel. The current internal state of a channel can be obtained by reading the bit EN. 2. To disable an MCS channel reliably the EN bit should be cleared followed by setting the CAT and CWT bit in order to cancel any pending waiting or ARU instructions. 3. The EN bit is write protected during RAM reset phase.

38.19.12.2 MCS[i]_CH[x]_PC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0040_H$

Value after reset: 0000 0000+4*x_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.263 MCS[i]_CH[x]_PC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PC	Current Program Counter. **NOTES** 1. The program counter is only writable if the corresponding MCS-channel is disabled. The bits 0 and 1 are always written as zeros. 2. The actual width of the program counter depends on the MCS configuration. The actual width is RAW+USR+2 bits meaning that only the bits 0 to RAW+USR+1 are available and the other bits (RAW+USR+2 to 31) are reserved.

38.19.12.3 MCS[i]_CH[x]_R[y]

Access: This register can be read or written in 32-bit units.

Address: MCS[i]_CH[x]_R0: <GTM_base> + 1000_H× i + 80_H× x + $F0000_H$
MCS[i]_CH[x]_R1: <GTM_base> + 1000_H× i + 80_H× x + $F0004_H$
MCS[i]_CH[x]_R2: <GTM_base> + 1000_H× i + 80_H× x + $F0008_H$
MCS[i]_CH[x]_R3: <GTM_base> + 1000_H× i + 80_H× x + $F000C_H$
MCS[i]_CH[x]_R4: <GTM_base> + 1000_H× i + 80_H× x + $F0010_H$
MCS[i]_CH[x]_R5: <GTM_base> + 1000_H× i + 80_H× x + $F0014_H$
MCS[i]_CH[x]_R6: <GTM_base> + 1000_H× i + 80_H× x + $F0018_H$
MCS[i]_CH[x]_R7: <GTM_base> + 1000_H× i + 80_H× x + $F001C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.264 MCS[i]_CH[x]_R[y] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DATA	Data of general purpose register R[y]. **NOTES** 1. This register is the same as described in **Section 38.19.10.1**. 2. For the register MCS[i]_CH[x]_R6, **Section 38.19.10.1**, an additional write protection during an active ARDI or NARDI instruction is applied.

38.19.12.4 MCS[i]_CH[x]_ACB

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0024_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	ACB4	ACB3	ACB2	ACB1	ACB0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.265 MCS[i]_CH[x]_ACB Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4	ACB4	ARU Control bit 4. **NOTE** This bit is read only and it mirrors the internal state.
3	ACB3	ARU Control bit 3. **NOTE** This bit is read only and it mirrors the internal state.
2	ACB2	ARU Control bit 2. **NOTE** This bit is read only and it mirrors the internal state.
1	ACB1	ARU Control bit 1. **NOTE** This bit is read only and it mirrors the internal state.
0	ACB0	ARU Control bit 0. **NOTE** This bit is read only and it mirrors the internal state.

38.19.12.5 MCS[i]_CH[x]_MHB

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F003C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DATA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.266 MCS[i]_CH[x]_MHB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	DATA	Data of memory high bit register MHB.

38.19.12.6 MCS[i]_CH[x]_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0044_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR_IRQ	STK_ERR_IRQ	MCS_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.267 MCS[i]_CH[x]_IRQ_NOTIFY Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERR_IRQ	MCS channel x ERR interrupt. READ access: 0: IRQ notify flag is cleared 1: IRQ notify flag is set WRITE access: 0: No action 1: Clear IRQ notify flag **NOTES** 1. If the ERR bit of register STA is triggered the ERR_IRQ will also be set. 2. This bit will be cleared on a CPU write access with a value ‘1’. A read access leaves the bit unchanged.
1	STK_ERR_IRQ	Stack counter overflow/underflow of channel x. READ access: 0: IRQ notify flag is cleared 1: IRQ notify flag is set WRITE access: 0: No action 1: Clear IRQ notify flag **NOTE** This bit will be cleared on a CPU write access with a value ‘1’. A read access leaves the bit unchanged.

Table 38.267 MCS[i]_CH[x]_IRQ_NOTIFY Register Contents (2/2)

Bit Position	Bit Name	Function
0	MCS_IRQ	Interrupt request by MCS-channel x. READ access: 0: IRQ notify flag is cleared 1: IRQ notify flag is set WRITE access: 0: No action 1: Clear IRQ notify flag **NOTES** 1. This bit will be cleared on a CPU write access with a value '1'. A read access leaves the bit unchanged. 2. By writing a '1' to this register, the IRQ flag in the MCS channel status register STA is cleared.

38.19.12.7 MCS[i]_CH[x]_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0048_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR_IRQ_EN	STK_ERR_IRQ_EN	MCS_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.268 MCS[i]_CH[x]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERR_IRQ_EN	MCS channel x ERR_IRQ interrupt enable 0: Disable interrupt 1: Enable interrupt
1	STK_ERR_IRQ_EN	MCS channel x STK_ERR_IRQ interrupt enable 0: Disable interrupt 1: Enable interrupt
0	MCS_IRQ_EN	MCS channel x MCS_IRQ interrupt enable 0: Disable interrupt 1: Enable interrupt

38.19.12.8 MCS[i]_CH[x]_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F004C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG_ERR_IRQ	TRG_STK_ERR_RQ	TRG_MCS_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.269 MCS[i]_CH[x]_IRQ_FORCINT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TRG_ERR_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register by software READ access: 0: No status WRITE access: 0: No action 1: Force setting of IRQ notify flag **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
1	TRG_STK_ERR_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register by software READ access: 0: No status WRITE access: 0: No action 1: Force setting of IRQ notify flag **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
0	TRG_MCS_IRQ	Trigger IRQ bit in MCS_CH_[x]_IRQ_NOTIFY register by software READ access: 0: No status WRITE access: 0: No action 1: Force setting of IRQ notify flag **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

38.19.12.9 MCS[i]_CH[x]_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0050_H$

Value after reset: 0000 000X$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.270 MCS[i]_CH[x]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.19.12.10 MCS[i]_CH[x]_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 80_H× x + $F0054_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR_EIRQ_EN	STK_ERR_EIRQ_EN	MCS_EIRQ_EN0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 38.271 MCS[i]_CH[x]_EIRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERR_EIRQ_EN	MCS channel x ERR_EIRQ error interrupt enable 0: Disable error interrupt 1: Enable error interrupt
1	STK_ERR_EIRQ_EN	MCS channel x STK_ERR_IRQ error interrupt enable 0: Disable error interrupt 1: Enable error interrupt
0	MCS_EIRQ_EN	MCS channel x MCS_EIRQ error interrupt enable 0: Disable error interrupt 1: Enable error interrupt

38.19.12.11 MCS[i]_CTRL_STAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F0064_H$

Value after reset: 000X 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	HLT_AEIM_ERR	EN_XOREG	EN_TIM_FOUT	—	ERR_SRC_ID			—	—	HLT_SP_OFL	RAM_RST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SCD_CH				—	—	—	—	—	—	SCD_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 38.272 MCS[i]_CTRL_STAT Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	HLT_AEIM_ERR	Halt on AEI bus master error. 0: Ignore invalid AEI bus master access 1: Halt MCS-channel on invalid AEI bus master access **NOTES** 1. If the register HLT_AEIM_ERR is set and an MCS channel x is executing an invalid bus master access, the MCS channel x is halted, the ERR bit of its register STA is set and the bit field ERR_SRC_ID of this register is updated. 2. If the bus master is accessing a slave that does not insert wait cycles (e.g. register access) it takes two additional clock cycles until the MCS channel is halted. Within that time spawn the MCS channel can continue with its program execution, depending on the selected scheduling mode. 3. The registers AEIM_XPT_STA and AEIM_XPT_ADDR of the GTM sub module CCM are always updated on the first invalid AEI bus master access, independently of the state of HLT_AEIM_ERR.
25	EN_XOREG	Enable extended register set. 0: Extended operation register sets XOREG, BAREG, and WXREG are disabled 1: Extended operation register sets XOREG, BAREG, and WXREG are enabled **NOTE** If the extended operation register sets are disabled, the MCS instructions can only use the subset OREG of the register set as arguments in the instructions. In this case the upper address bits in the instructions are always read as zeros, which leads to unexpected results of the MCS program if arguments A or B refer a register that is not part or OREG.
24	EN_TIM_FOUT	Enable routing of TIM[i]_CH[x]_F_OUT signal. 0: Read access to register CTRG/MCS[i]_CTRG provides state of the internal trigger registers. 1: Read access to register CTRG/MCS[i]_CTRG provides state of the external signal TIM[i]_CH[x]_F_OUT.

Table 38.272 MCS[i]_CTRL_STAT Register Contents (2/3)

Bit Position	Bit Name	Function
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	ERR_SRC_ID	Error source identifier. 000_B: No HW generated Error occurred 001_B: Detected ECC error 010_B: Detected memory overflow 011_B: Detected invalid opcode 100_B: Divide by zero 101_B: Invalid register write access to GPR from write protected channel 110_B: Invalid memory write access to protected memory region 111_B: Invalid AEI bus master access **NOTE** This register is updated once, if an error was detected by the MCS. The register is set to its initial value 000 after each write access to an existing ERR bit in the register MCS[i]_ERR. If multiple errors occur, ERR_SRC_ID is holding the first type of error which has occurred.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	HLT_SP_OFL	Halt on stack pointer overflow. 0: No halt on MCS-channel stack pointer counter over/underflow. 1: MCS-channel is disabled if a stack pointer counter over/underflow occurs.
16	RAM_RST	RAM reset bit READ access: 0: No RAM reset is active 1: MCS currently resets RAM content WRITE access: 0: No action 1: Initiate RAM reset **NOTES** 1. The RAM reset initializes the memory content with zeros. RAM access and enabling of MCS channels is disabled during active RAM reset. 2. This bit is only writable if the bit RF_PROT in register GTM_CTRL is cleared and all MCS-channels are disabled. 3. The actual reset values of this bit depends on the silicon vendor configuration. The reset value is 1, if the RAM reset is performed together with the sub module reset, otherwise the reset value is 0. If the reset value is 1, the reset value is changed to 0 by hardware, when the RAM reset finished.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	SCD_CH	Channel selection for scheduling algorithm. MCS-channel identifier used by several scheduling modes in different ways. Details about its usage can be found in **Section 38.19.3**. **NOTE** The actual width of the bit field SCD_CH is calculated as $\lceil \log2(T+1) \rceil$. Unused MSBs are reserved and read as zero.
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.272 MCS[i]_CTRL_STAT Register Contents (3/3)

Bit Position	Bit Name	Function
1, 0	SCD_MODE	Select MCS scheduling mode 00_B: Accelerated Scheduling. 01_B: Round Robin Scheduling. 10_B: Single Priority Scheduling. 11_B: Multiple Priority Scheduling.

38.19.12.12 MCS[i]_REG_PROT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F0060_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WPROT7		WPROT6		WPROT5		WPROT4		WPROT3		WPROT2		WPROT1		WPROT0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.273 MCS[i]_REG_PROT Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	WPROT[7:0]	Register Write Protection of MCS-channel k. 00_B: No register write protection activated 01_B: Predecessor MCS channel cannot write to its RS[y] registers 10_B: Current MCS channel cannot write to its R[y] registers 11_B: Reserved

NOTES

1. Only the first T bit fields of this register (bit 0 to 2*T-1) are functionally implemented. The other bits (bit 2*T to 31) are reserved bit fields.
2. The predecessor channel of MCS channel 0 is MCS channel T-1.
3. If an MCS channel x is writing to a general purpose register that is write protected by register MCS[i]_REG_PROT the ERR bit of the register STA is set, the MCS channel x is halted and the ERR_SRC_ID bit field of register MCS[i]_CTRL_STAT is updated.
4. This register is only writable if the bit RF_PROT in register GTM_CTRL is cleared.

38.19.12.13 MCS[i]_CTRG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F0028_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRG23	TRG22	TRG21	TRG20	TRG19	TRG18	TRG17	TRG16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG15	TRG14	TRG13	TRG12	TRG11	TRG10	TRG9	TRG8	TRG7	TRG6	TRG5	TRG4	TRG3	TRG2	TRG1	TRG0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.274 MCS[i]_CTRG Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	TRG[23:16]	Trigger bit [23:16]. READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Clear trigger bit
15 to 0	TRG[15:0]	Trigger bit [7:0]. READ access: if bitfield EN_TIM_FOUT = 0 0: Trigger is cleared 1: Trigger is set Trigger bit i[15:8]. READ access: if bitfield EN_TIM_FOUT = 1 0: Input signal TIM[i]_CH0_F_OUT is cleared 1: Input signal TIM[i]_CH0_F_OUT is set WRITE access: 0: No action 1: Clear trigger bit

NOTES

1. The trigger bits TRGx are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit.
2. Besides setting the trigger bits with register STRG/MCS[i]_STRG, the k-th trigger bit TRGk (with k < 16) can also be set by the external capture event that is enabled by the k-th bit of register CCM[i]_EXT_CAP_EN. If bit k bit is disabled, the k-th trigger bit TRGk can only be set by MCS or CPU.
3. In the scheduling modes Accelerated Scheduling and Round Robin Scheduling, a write access to MCS[i]_CTRG may take up to T + 1 clock cycles, since the write access is scheduled to the next CPU time slot determined by the MCS scheduler. In the modes Single Prioritization Scheduling and Multiple Prioritization Scheduling, no upper limit access time for a write access to MCS[i]_CTRG can be guaranteed. The High Prioritized tasks have to be disabled in order to guarantee fast write access to MCS[i]_CTRG.
4. The result of a read access to this register differs in dependency of the bit field EN_TIM_FOUT of register MCS[i]_CTRL_STAT.

38.19.12.14 MCS[i]_STRG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F002C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRG23	TRG22	TRG21	TRG20	TRG19	TRG18	TRG17	TRG16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG15	TRG14	TRG13	TRG12	TRG11	TRG10	TRG9	TRG8	TRG7	TRG6	TRG5	TRG4	TRG3	TRG2	TRG1	TRG0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.275 MCS[i]_STRG Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TRG[23:0]	Trigger bit [23:0]. READ access: 0: Trigger bit is cleared 1: Trigger bit is set WRITE access: 0: No action 1: Set trigger bit **NOTES** 1. The trigger bits TRGx are accessible by all MCS channels as well as the CPU. Setting a trigger bit can be performed with the STRG register, in the case of an MCS-channel or the MCS[i]_STRG register in the case of the CPU. Clearing a trigger bit can be performed with the CTRG register, in the case of an MCS-channel or the MCS[i]_CTRG register in the case of the CPU. Trigger bits can be used for signalizing specific events to MCS-channels or the CPU. An MCS-channel suspended with a WURM instruction can be resumed by setting the appropriate trigger bit. 2. Besides setting the trigger bits with register STRG/MCS[i]_STRG, the k-th trigger bit TRGk (with k < 16) can also be set by the external capture event that is enabled by the k-th bit of register CCM[i]_EXT_CAP_EN. If bit k bit is disabled, the k-th trigger bit TRGk can only be set by MCS or CPU. 3. In the scheduling modes Accelerated Scheduling and Round Robin Scheduling, a write access to MCS[i]_STRG may take up to T + 1 clock cycles, since the write access is scheduled to the next CPU time slot determined by the MCS scheduler. In the modes Single Prioritization Scheduling and Multiple Prioritization Scheduling, no upper limit access time for a write access to MCS[i]_STRG can be guaranteed. The High Prioritized tasks have to be disabled in order to guarantee fast write access to MCS[i]_STRG.

38.19.12.15 MCS[i]_RESET

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F0068_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	RST8	RST7	RST6	RST5	RST4	RST3	RST2	RST1	RST0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.276 MCS[i]_RESET Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	RST[8:0]	Software reset of channel [8:0] READ access: 0: No status WRITE access: 0: No action 1: Reset channel [8:0] **NOTES** 1. The RSTx (x = 0 ...T-1) bits is cleared automatically after write access of CPU. All channel related registers of channel x are set to their reset values and channel operation is stopped immediately. 2. Channel related registers of channel x are all registers MCS[i]_CH[x]_*, all MCS internal registers accessible by the corresponding channel, with exception of the common trigger register (accessed by MCS[i]_CTRG/ MCS[i]_STRG).

38.19.12.16 MCS[i]_CAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F006C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	--	CAT7	CAT6	CAT5	CAT4	CAT3	CAT2	CAT1	CAT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.277 MCS[i]_CAT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CAT[7:0]	Cancel ARU transfer of channel [7:0]. READ access: 0: No status WRITE access: 0: No action 1: Cancel any pending blocking ARU read or write transfer. **NOTES** 1. The CATx (x = 0 ...T-1) bit inside the STA register of the corresponding MCS-channel is set and any pending blocking ARU read or write request is canceled. The MCS-channel resumes with the instruction after the blocking ARU transfer instruction. 2. The CATx (x = 0 ...T-1) bit is cleared by the corresponding MCS channel, when the channel is entering a blocking ARU read or write instruction.

38.19.12.17 MCS[i]_CWT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F0070_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	--	CWT7	CWT6	CWT5	CWT4	CWT3	CWT2	CWT1	CWT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.278 MCS[i]_CWT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CWT[7:0]	Cancel waiting instruction for channel [7:0]. READ access: 0: No status WRITE access: 0: No action 1: Cancel any pending waiting instruction. **NOTES** 1. The CWTx (x = 0 ...T-1) bit inside the STA register of the corresponding MCS-channel is set and any pending WURM instruction is canceled. The MCS-channel resumes with the instruction after the WURM instruction. 2. The CWTx (x = 0 ...T-1) bit is cleared by the corresponding MCS channel, when the channel reaches a WURM instruction.

38.19.12.18 MCS[i]_ERR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + $F007C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	--	ERR7	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.279 MCS[i]_ERR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ERR[7:0]	Error State of MCS-channel [7:0]. READ access: 0: Error bit is cleared 1: Error bit is set WRITE access: 0: No action 1: Clear error bit **NOTES** 1. The CPU can read the ERRx (x = 0...T-1) bits in order to determine the current error state of the corresponding MCS-channel x. 2. The error state is also evaluated by the sub module MON, if this module is available. 3. Writing a value 1 to this bit resets the corresponding error state and resets the channel internal ERR bit in the STA and channel CTRL registers. Moreover, each write access to this bit also sets the ERR_SRC_ID bit field of register MCS[i]_CTRL_STAT to its reset value.

38.19.12.19 MCS[i]_MEM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 1000_H× i + 38000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.280 MCS[i]_MEM Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA	MCS memory location.

38.20 Memory Configuration (MCFG)

38.20.1 Overview

The Memory Configuration sub-module (MCFG) is an infrastructure module that organizes physical memory blocks and maps them to the RAM ports 0 and 1 of available Multi Channel Sequencer (MCS) modules.

The following parameters are design variables for the MCFG hardware structure that can vary in its range for different devices:

- MAW - Memory address width of a large physical memory block.
- ERM - Enable RAM1 MSB (0 - RAM1 MSB disabled,
  1 - RAM1 MSB enabled,

The actual values for these parameters can be obtained from the device specific **Section 38.28, GTM Device 358**. It should be noted that the actual value of the parameter ERM can be obtained by the bit ERM of the register CCM[i]_HW_CONF.

Depending on the value of parameter ERM, the MCFG module assumes externally connected physical RAM modules with different sizes. If ERM = 0, MCFG assumes that each MCS instance provides a large physical memory block with 2^{MAW} memory locations each 32-bit wide which leads to a RAM module with 2^{MAW+2} memory addresses. Further each MCS instance provides a small physical memory block with 2^{MAW-1} memory locations each 32-bit wide leading to a RAM module with 2^{MAW+1}(byte wise) memory addresses. If ERM = 1, MCFG assumes that each MCS instance provides two large physical memory block each with 2^{MAW} memory locations each 32-bit leading to a RAM module with 2^{MAW+2}(byte wise) memory addresses.

In order to support different memory sizes for different MCS instances, the MCFG module provides three layout configurations for reorganization of memory pages mapped to the RAM ports of neighboring MCS modules. **Table 38.281, Memory Layout Configurations (ERM = 0)** shows all layout configurations for the case that ERM = 0 and **Table 38.283, Memory Layout Configurations (ERM = 1)** shows the layout configurations for the case that ERM = 1. Each box in these pictures represents a physical memory block.

The layout configuration DEFAULT is always assigning a memory block of size $2^{MAW} \times 32$ bits to MCS RAM port 0. Depending on ERM, RAM port 1 of each MCS is whether assigned to a memory block of size $2^{MAW-1} \times 32$ bits (ERM = 0) or a memory block of size $2^{MAW} \times 32$ bits (ERM = 1).

The layout configuration SWAP is swapping the memory block assigned to RAM port 1 of the current MCS instance with the memory block assigned to RAM port 0 of the successive MCS instance. If ERM = 0, this means that the memory of the current MCS instance is increased by $2^{MAW-1} \times 32$ bits but the memory of the successor is decreased by $2^{MAW-1} \times 32$ bits compared to the DEFAULT configuration. If ERM = 1, the SWAP configuration has no effect on the memory sizes of the individual MCS instances.

The layout configuration BORROW is borrowing the memory block assigned to RAM port 0 of the successive MCS instance for the current instance. This means, the memory of the current MCS module is increased by $2^{MAW} \times 32$ bits but the memory of the successor is decreased by $2^{MAW} \times 32$ bits compared to the DEFAULT configuration.

Considering the order the mentioned MCS modules, it should be noted that the successor of the last MCS instance is the first MCS instance MCS0.

The actual sizes of the memory pages mapped to the MCS RAM ports 0 and 1 depends on the layout configuration for of current instance MCS[i] and the layout configuration of the preceding memory

instance MCS[i-1]. The sizes of these memory pages can be obtained by the layout parameters MP0 and MP1, as described in the specification of the MCS.

Table 38.282, Memory Layout Parameters (ERM = 0) and **Table 38.284, Memory Layout Parameters (ERM = 1)** summarize the layout parameters MP0 and MP1 of MCS instance MCS[i] for the case that ERM = 0 and ERM = 1. Note that the predecessor of instance MCS0 is last available MCS instance.

The addressing of memory port 0 ranges from 0 to MP0–4 and the addressing of memory page 1 ranges from MP0 to MP1–4.

This document assumes that the GTM implementation embeds 8 MCS instances. However, the actual number of implemented MCS instances can be obtained from **Section 38.28, GTM Device 358**.

Table 38.281 Memory Layout Configurations (ERM = 0)

	DEFAULT	SWAP	BORROW
Configuration for instance MCS[i]	2^{MAW} × 32 bit 2^{MAW-1} × 32 bit	2^{MAW} × 32 bit 2^{MAW} × 32 bit	2^{MAW} × 32 bit 2^{MAW} × 32 bit 2^{MAW-1} × 32 bit
Configuration for instance MCS[i+1]	2^{MAW} × 32 bit 2^{MAW-1} × 32 bit	2^{MAW-1} × 32 bit 2^{MAW-1} × 32 bit	2^{MAW-1} × 32 bit

Table 38.282 Memory Layout Parameters (ERM = 0)

			Memory Layout Option of preceding MCS instance MCS[i-1]		
			DEFAULT	SWAP	BORROW
Memory Layout Option of current MCS instance MCS[i]	DEFAULT	MP0	2^{MAW+2}	2^{MAW+1}	0
		MP1	$2^{MAW+2}+2^{MAW+1}$	2^{MAW+2}	2^{MAW+1}
	SWAP	MP0	2^{MAW+2}	2^{MAW+1}	0
		MP1	2^{MAW+3}	$2^{MAW+2}+2^{MAW+1}$	2^{MAW+2}
	BORROW	MP0	2^{MAW+2}	2^{MAW+1}	0
		MP1	$2^{MAW+3}+2^{MAW+1}$	2^{MAW+3}	$2^{MAW+2}+2^{MAW+1}$

Table 38.283 Memory Layout Configurations (ERM = 1)

	DEFAULT	SWAP	BORROW
Configuration for instance MCS[i]	2^{MAW} × 32 bit 2^{MAW} × 32 bit	2^{MAW} × 32 bit 2^{MAW} × 32 bit	2^{MAW} × 32 bit 2^{MAW} × 32 bit 2^{MAW} × 32 bit
Configuration for instance MCS[i+1]	2^{MAW} × 32 bit 2^{MAW} × 32 bit	2^{MAW} × 32 bit 2^{MAW} × 32 bit	2^{MAW} × 32 bit

Table 38.284 Memory Layout Parameters (ERM = 1)

			Memory Layout Option of preceding MCS instance MCS[i-1]		
			DEFAULT	**SWAP**	**BORROW**
Memory Layout Option of current MCS instance MCS[i]	DEFAULT	MP0	2^{MAW+2}	2^{MAW+2}	0
		MP1	2^{MAW+3}	2^{MAW+3}	2^{MAW+2}
	SWAP	MP0	2^{MAW+2}	2^{MAW+2}	0
		MP1	2^{MAW+3}	2^{MAW+3}	2^{MAW+2}
	BORROW	MP0	2^{MAW+2}	2^{MAW+2}	0
		MP1	$2^{MAW+2}+2^{MAW+3}$	$2^{MAW+2}+2^{MAW+3}$	2^{MAW+3}

38.20.2 MCFG Configuration Registers Overview

This section describes the configuration registers of the MCFG sub-module.

Table 38.285 Register list

Register Name	Register Name	Details in Section
MCFG_CTRL	Memory layout configuration.	**38.20.3.1**

38.20.3 MCFG Configuration Registers Description

38.20.3.1 MCFG_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $00F40_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MEM9		MEM8	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEM7		MEM6		MEM5		MEM4		MEM3		MEM2		MEM1		MEM0	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.286 MCFG_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19, 18	MEM9	Configure Memory pages for MCS-instance MCS9. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
17,16	MEM8	Configure Memory pages for MCS-instance MCS8. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
15, 14	MEM7	Configure Memory pages for MCS-instance MCS7. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
13, 12	MEM6	Configure Memory pages for MCS-instance MCS6. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
11, 10	MEM5	Configure Memory pages for MCS-instance MCS5. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
9, 8	MEM4	Configure Memory pages for MCS-instance MCS4. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved

Table 38.286 MCFG_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	MEM3	Configure Memory pages for MCS-instance MCS3. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
5, 4	MEM2	Configure Memory pages for MCS-instance MCS2. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
3, 2	MEM1	Configure Memory pages for MCS-instance MCS1. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved
1, 0	MEM0	Configure Memory pages for MCS-instance MCS0. 00_B: DEFAULT configuration 01_B: SWAP configuration 10_B: BORROW configuration 11_B: Reserved

Note 1. It should be noted that the actual GTM-IP implementation may embed less MCS instances than mentioned in this register (see [1]). In this case this register only implements the register bits for available MCS instances.

Note 2. This register is only writable if the bit RF_PROT in register GTM_CTRL is cleared.

38.21 TIM0 Input Mapping Module (MAP)

38.21.1 Overview

The MAP sub-module generates the two input signals TRIGGER and STATE for the sub-module DPLL by evaluating the output signals of the channel 0 up to channel 5 of sub-module TIM0. By using the TIM as input sub-module, the filtering of the input signals can be done inside the TIM channels themselves. The MAP sub-module architecture is depicted in **Figure 38.107**.

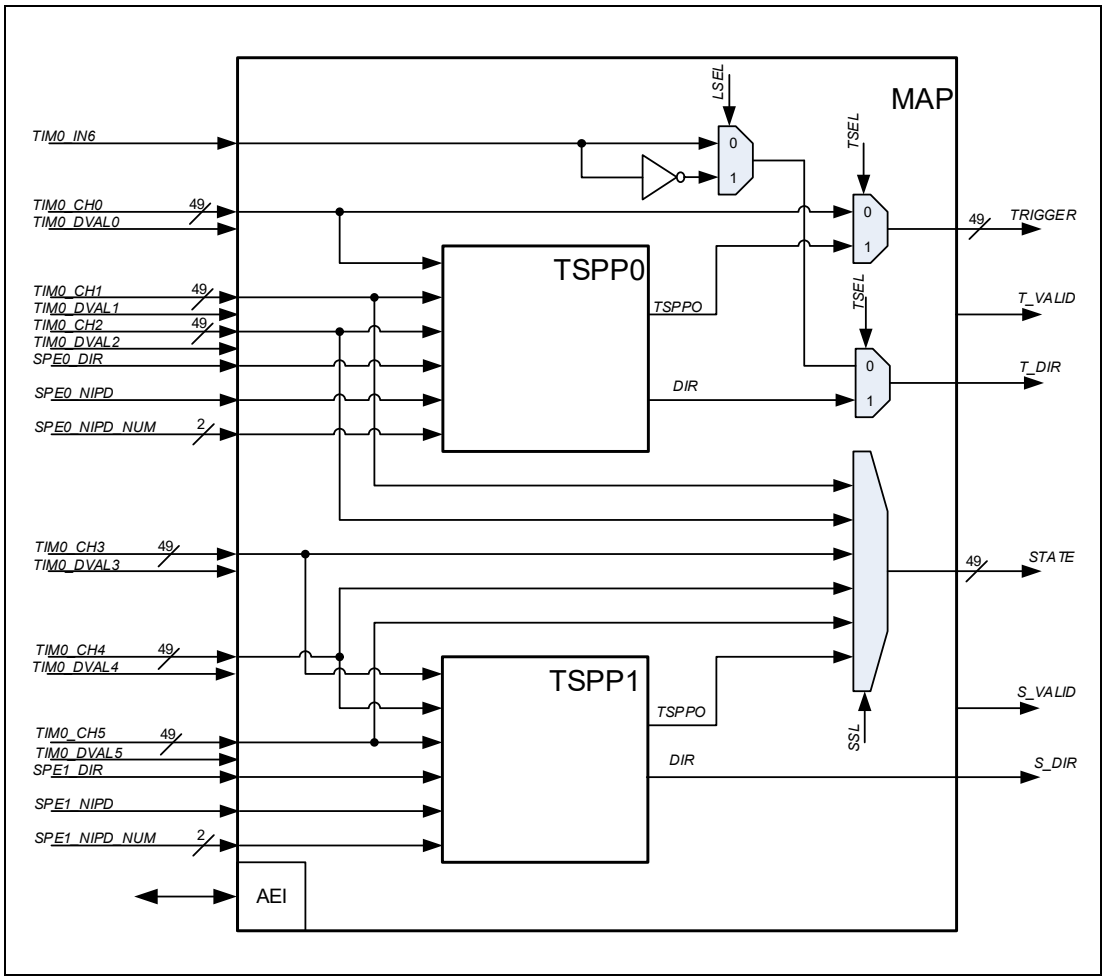


Figure 38.107 MAP sub-module architecture

Generally, the MAP sub-module can route the channel signals coming from TIM0 in three ways. First, it is possible to route the whole 49 bits of data coming from channel 0 of module TIM0 (TIM0_CH0) to the TRIGGER signal which is then provided to the DPLL together with the T_VALID signal.

Second, the MAP module can route one of the five signals coming from the module TIM0 (i.e. the signals coming from channel 1 up to channel 5) to the output signal STATE which is then provided to the module DPLL together with the S_VALID signal.

Third, the TRIGGER, T_VALID, STATE and S_VALID signals can be generated out of the TIM Signal Preprocessing (TSPP) subunits. This is done in combination with the Sensor Pattern Evaluation (SPE) sub-module described in **Section 38.23, Sensor Pattern Evaluation (SPE)**.

There, the signal TRIGGER is generated in subunit TSPP0 out of the TIM0 signals coming from channel 0 up to 2.

The signal STATE is generated in subunit TSPP1 out of the TIM signals coming from channel 3 up to channel 5.

This is only done, when the TSSPx subunits are enabled and when the SPEx_NIPD signal is raised by the SPE sub-module. The SPEx_NIPD_NUM signal encodes, which of the 3 TIMx_CHy input signals has been changed. The SPEx_DIR signal is routed through the TSPPx subunit and implements the T_DIR or S_DIR signal.

A third method to provide a direction signal to DPLL is to use TIM0 channel 6 input (TIM0_IN6) and to route it instead of the DIR signal coming from TSSOP0 to the MAP output T_DIR (set TSEL=0)

38.21.2 TIM Signal Preprocessing (TSPP)

The TSPP combines the three 49-bit input streams coming from the TIM0 sub-module and generates one combined 49-bit output stream

TSPPO. The input stream combination is done in the unit Bit Stream Combination (BSC). The architecture of the TSPP is shown in **Figure 38.108**.

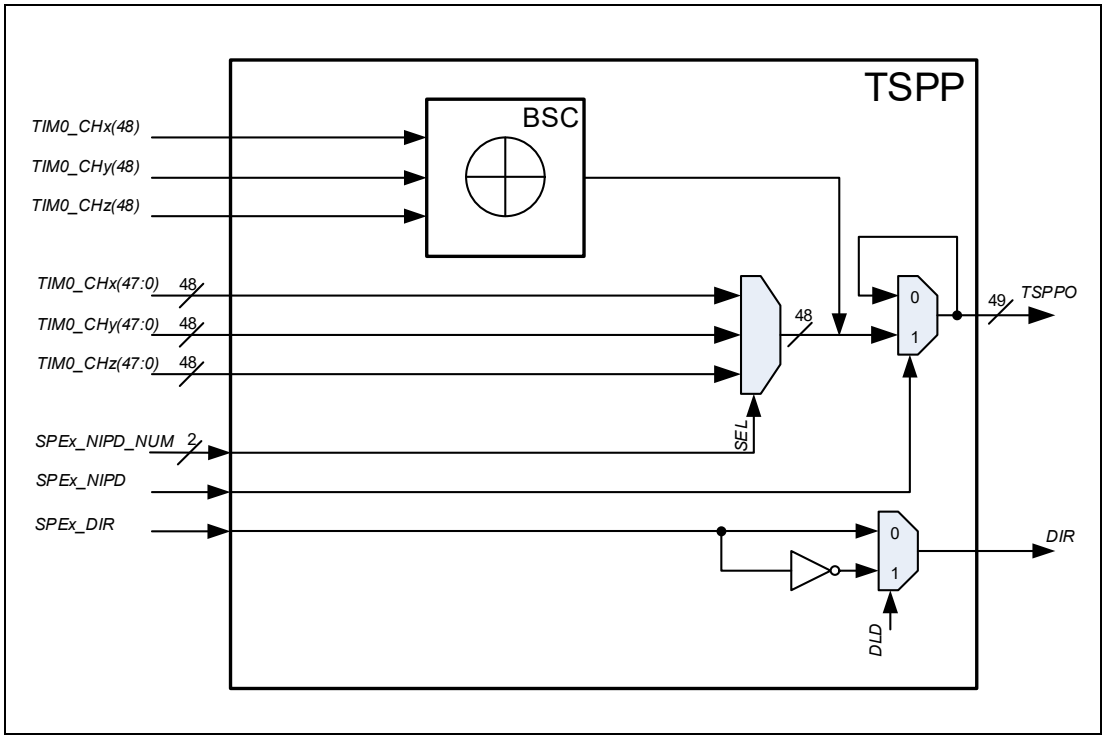


Figure 38.108 TIM Signal Preprocessing (TSPP) subunit architecture

38.21.2.1 Bit Stream Combination

The BSC subunit is used to xor-combine the three most significant bits TIM0_CHx(48), TIM0_CHy(48) and TIM0_CHz(48) of the TIM0 inputs.

The xor-combined signal is merged with the remaining 48 bits of one of the three input signals TIM0_CHx(47 to 0), TIM0_CHy(47 to 0) or TIM0_CHz(47 to 0) the TSPPO signal. The selection is done with the SPEx_NIPD_NUM input signal coming from the SPE sub-module. The action, when the 49 bits are transferred to the TSPPO and the T_VALID or S_VALID signal is raised is determined by the SPEx_NIPD signal coming from the SPE sub-module. The TSPPO output signal generation is shown in the example in **Figure 38.109**.

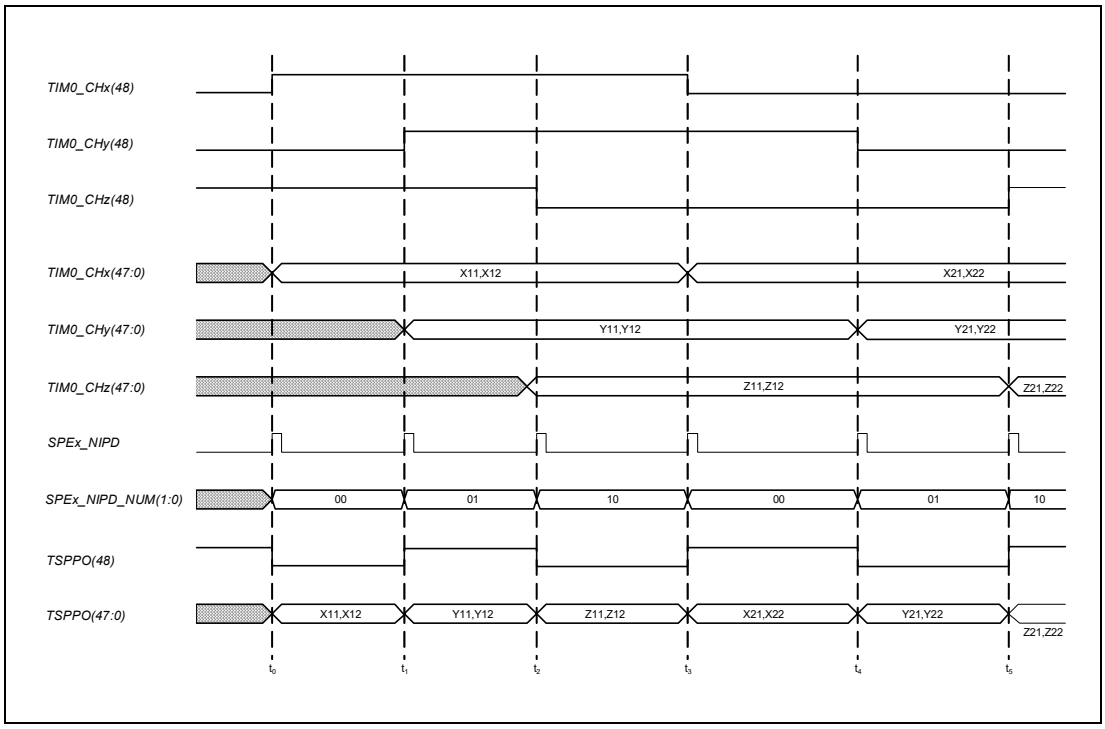


Figure 38.109 TSPP Signal generation for signal TSPPO

The SPEx_NIPD_NUM input signal determines, which data is routed to the TSPPO signal. At the first edge of TIM0_CHx(48) the new data X11 and X12 are routed to TSPPO(47:0). The values X11 and X12 are the two 24-bit values coming from the TIM input channel TIM0_CHx. The next edge is at time t_1 on signal TIM0_CHy(48). Therefore, at time t_1 the TSPPO(48) signal level changes and the TSPPO(47:0) is set to Y11 and Y12 and so forth.

38.21.3 MAP Configuration Registers Overview

MAP contains following configuration registers:

Table 38.287 Register list

Symbol	Register Name	Details in Section
MAP_CTRL	MAP Control register	**38.21.4.1**

38.21.4 MAP Configuration Register Description

38.21.4.1 MAP_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $00F00_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TSPP1_I2V	TSPP1_I1V	TSPP1_I0V	—	—	TSPP1_DLD	TSPP1_EN	—	TSPP0_I2V	TSPP0_I1V	TSPP0_I0V	—	—	TSPP0_DLD	TSPP0_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	LSEL	SSL			TSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.288 MAP_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	TSPP1_I2V	Disable of TSPP1 TIM0_CHz(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP1 is set to 0.
29	TSPP1_I1V	Disable of TSPP1 TIM0_CHy(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP1 is set to 0.
28	TSPP1_I0V	Disable of TSPP1 TIM0_CHy(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP1 is set to 0.
27, 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	TSPP1_DLD	DIR level definition bit. 0: SPEx_DIR signal is routed through as is. 1: SPEx_DIR signal is inverted.
24	TSPP1_EN	Enable of TSPP1 subunit. 0: TSPP1 disabled. 1: TSPP1 enabled.
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	TSPP0_I2V	Disable of TSPP0 TIM0_CHz(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP0 is set to 0.
21	TSPP0_I1V	Disable of TSPP0 TIM0_CHy(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP0 is set to 0.
20	TSPP0_I0V	Disable of TSPP0 TIM0_CHx(48) input line. 0: Input line enabled. 1: Input line disabled; input for TSPP0 is set to 0.
19, 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.288 MAP_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
17	TSPP0_DLD	DIR level definition bit. 0: SPEx_DIR signal is routed through as is. 1: SPEx_DIR signal is inverted.
16	TSPP0_EN	Enable of TSPP0 subunit. 0: TSPP0 disabled. 1: TSPP0 enabled.
15 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	LSEL	TIM0_IN6 input level selection 0: TIM0_IN6 input level '0' encodes TRIGGER in forward direction. 1: TIM0_IN6 input level '1' encodes TRIGGER in forward direction.
3 to 1	SSL	STATE signal output select. 000_B: TIM0_CH1 selected as STATE output signal. 001_B: TIM0_CH2 selected as STATE output signal. 010_B: TIM0_CH3 selected as STATE output signal. 011_B: TIM0_CH4 selected as STATE output signal. 100_B: TIM0_CH5 selected as STATE output signal. 101_B: TSPP1_TSPPO selected as STATE output signal. 110_B: Same as '000' 111_B: Same as '000'
0	TSEL	TRIGGER signal output select. 0: TIM0_CH0 selected as TRIGGER output signal. 1: TSPP0_TSPPO selected as TRIGGER output signal.

38.22 Digital PLL Module (DPLL)

38.22.1 Overview

The digital PLL (DPLL) sub-module is used for frequency multiplication. The purpose of this module is to get a higher precision of position or value information also in the case of applications with rapidly changed input frequencies. There are two input signals TRIGGER and STATE for which periodic events are processed. The time period between two active events is called an increment. Each increment is divided into a given number of sub increments by pulses called SUB_INC. The resolution of the generated pulses is restricted by the period of the CMU_CLK0 clock or the TS_CLK respectively (see description of the modules TBU, CMU). The input signals TRIGGER and STATE can have the meaning of position information of linear or angle motions, mass flow values, temperature, pressure or level of liquids. By means of the DPLL the load of the CPU can be reduced essentially by relieving it from repeated or periodic standard tasks.

The DPLL has to perform the following tasks:

1. Prediction of the duration of the current increment in chapter {REF:DPLL_1338}
2. Generation of SUB_INC1,2 pulses for up to 2 position counters in normal or emergency mode (see chapter {REF:DPLL_1450})
3. Synchronization of the actual position (under CPU control, see chapter {REF:DPLL_1493})
4. Possibility of seamless switch to emergency mode and back under CPU control, see configuration register DPLL_CTRL_0 at chapter {R- EF:DPLL_5737}
5. Prediction of position and time related events in chapter {REF:DPLL_1383}

38.22.2 Requirements and demarcation

The two input signals TRIGGER and STATE can be sensor signals from the same device or from two independent devices. When they come from the same device the TRIGGER input is typically a more frequent signal and STATE is a less frequent signal. In such a case the STATE signal can support an emergency mode, when no TRIGGER signal is available. There are also applications supported when STATE and TRIGGER are independent signals from different devices. Both input signals are combined with a validation signal T_VALID or S_VALID respectively, which shows the appearance of new data and must result in a data fetch and a start of the correspondent state machine to perform the calculations (see explanation below).

When STATE is a redundant signal of the same device only the TRIGGER input is used to generate the SUB_INC1 pulses in normal mode. There is a configuration possible, called emergency mode, for which the SUB_INC1 pulses are generated using the STATE input signal. The decision to switch in the emergency mode and back is made outside the DPLL. The CPU must switch the configuration bit RMO (reference mode) in the DPLL_CTRL_0 register (see **Section 38.22.18, DPLL Register description**). Because a switch in emergency mode can appear suddenly, the information of the last increment duration of the STATE input up to FULL_SCALE should be stored always as a precaution.

The filtering as well as the combination or choice of the input signals is made in the TIM sub-module (see **Section 38.15, Timer Input Module (TIM)**) by use of a configurable filter algorithm for each slope and signal as well as in the MAP module (see **Section 38.21, TIM0 Input Mapping Module (MAP)**) the right TRIGGER or STATE signal is selected by a multiplexer or in the SPE module (see **Section 38.23, Sensor Pattern Evaluation (SPE)**) different signals are combined to a TRIGGER or STATE signal by using an anti valence operation.

The filter delay value of the signal is transmitted from the TIM module in the FT part of the corresponding signal, because the delay conditions of the signals can change during application.

The filter delays depend also on the filter algorithms used. Only the effective filter delay can be considered in the DPLL.

In order to provide the timing conditions to the DPLL the input trigger signals should have a time stamp (and optional in addition a filter value and a signal level value, as stated above) with an appropriate resolution. The resolution of the time stamps can be either the same resolution as the input time base TBU_TS0 (see **Figure 38.111, DPLL Block Diagram**) or 8 times higher, selected by configuration bits in the DPLL_CTRL_1 register (see **Section 38.22.18.2, DPLL_CTRL_1**). The time base TBU_TS0 is used to predict events in the future, called actions.

At the SUB_INCx outputs a predefined number of pulses between each active slope of the TRIGGER/ STATE signal is generated, when the correspondent pulse generator is enabled by the enable bits SGEx=1 in the DPLL_CTRL_1 register (see **Section 38.22.18.2, DPLL_CTRL_1**).

Dependent on configuration different strategies can be used to correct a wrong pulse number.

The FULL_SCALE range is divided into a fix number of nominal increments. Nominal increments do have the same size. The number of nominal increments in HALF_SCALE is specified in the DPLL_CTRL_0 register (see **Section 38.22.18, DPLL Register description**). For synchronization purposes some TRIGGER / STATE input signals can be suppressed in dependency on the current position. Therefore an increment as duration between two active input events can be either a nominal increment or it can consist of more than one nominal increment. While a true nominal increment starts with an active event a virtual increment (of always nominal size) is an increment which starts with a missing event. Each increment which represents a gap (e.g. for synchronization purposes) consists of exactly one true nominal increment and at least one virtual increment, each of them having the same nominal duration (see figure below).

38.22.3 Input signal courses

Typical input signal courses are shown in the figure below.

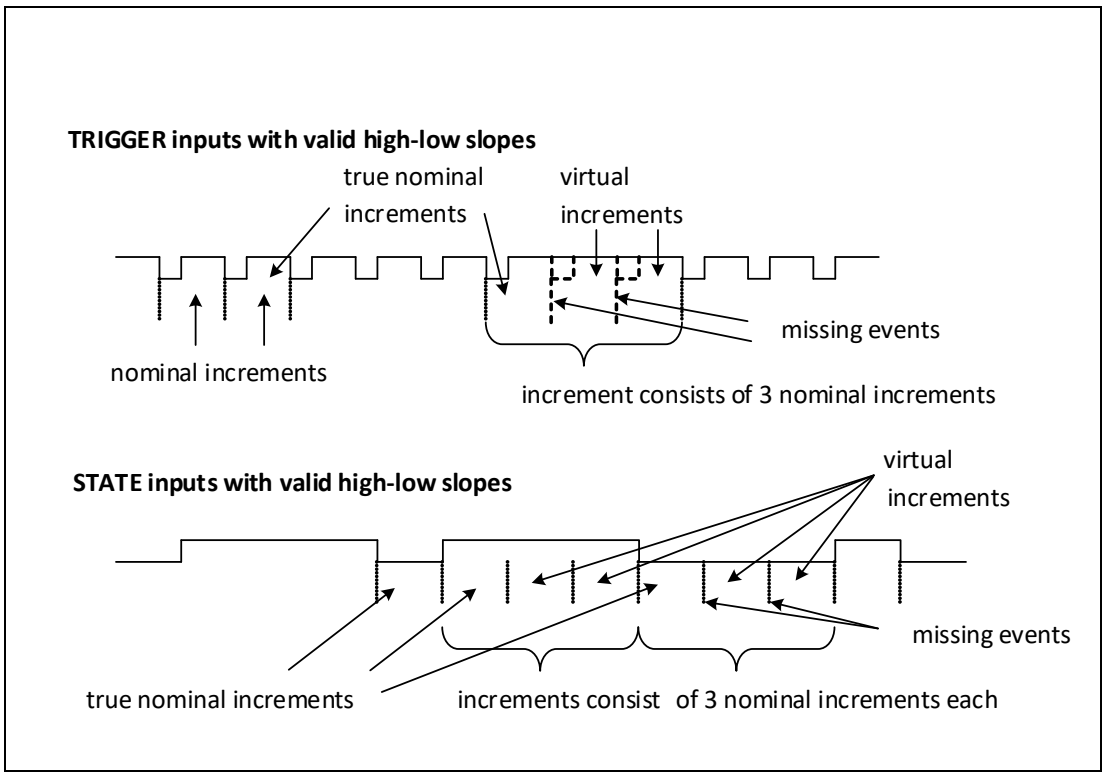


Figure 38.110 Trigger and State Input Signal

38.22.4 Block and interface description

The block description of the DPLL is shown in the following figure.

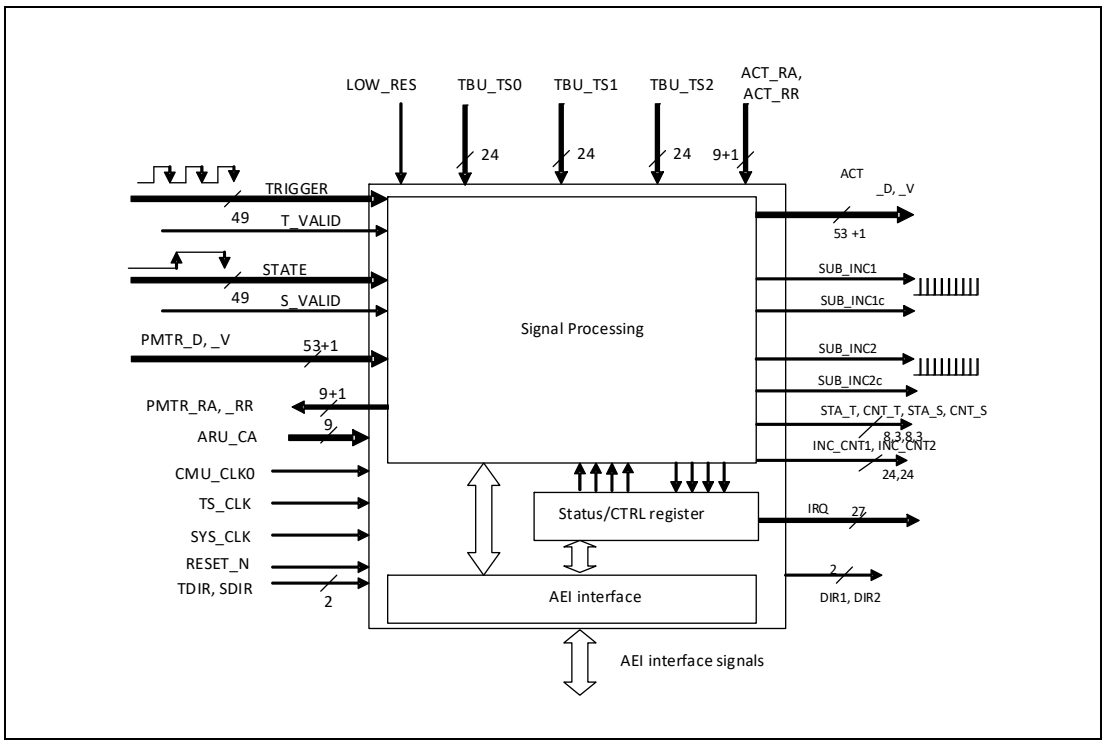


Figure 38.111 DPLL Block Diagram

Table (DPLL_6906) summarizes the interface signals of the DPLL shown by the block diagram above.

Table 38.289 Interface description of DPLL (1/3)

Name	Width	I/O	Description	Comment
TRIGGER	49	I	Normal Signal for triggering DPLL by positions/values Bit(48)= TRIGGER_S Bits(47:24)= TRIGGER_FT Bits(23:0)= TRIGGER_TS	One bit signal value (SV), 24 bits filter delay value info and 24 bits time stamp, filtered in different modes.
T_VALID	1	I	The values of *TRIGGER* are valid	Announces the arrival of a new *TRIGGER* value
STATE	49	I	Assistance signal for synchronization STATE(48)= STATE_S STATE(47:24)= STATE_FT STATE(23:0)= STATE_TS	Replacement of signal *TRIGGER* for emergency situations, or signal from an independent device; bits like above, corresponding
S_VALID	1	I	The values of STATE are valid	Announces the arrival of a new STATE value
PMTR_D	53	I	Position minus time request data, delivered by ARU on request for up to 24 requests PMTR_RR; SV_i=PMTR_D(52:48): ACB bits, directly written to the correspondent DPLL_ACB_j registers PSA[i]=PMTR_D(47:24): position value for action DLA[i]=PMTR_D(23:0) time delay value for action	Data values for calculation of actual Actions; the values are requested by AENi=1[1)] and CAIP=0[2)]; a served request is shown by PMTR_V which signals that valid PMTR data arrived and they are written immediately after that to the corresponding RAM regions and registers; The DLA[i] values must have the same resolution as the TBU_TS0 input.
PMTR_V	1	I	signals a valid PMTR_D value, that means data is delivered on request	when valid: PMTR_D overwrites data in the PSA[i] and DLA[i] registers, also when the corresponding ACT_N[i][3)] bit =1;

Table 38.289 Interface description of DPLL (2/3)

Name	Width	I/O	Description	Comment
ARU_CA	9	I	Channel address; for valid PMTR addresses: demand data by setting PMTR_RR=1 when enabled by AENi=1[1)] and CAIP=0[2)];	counter value of ARU selects PMTR_RA and PMTR_RR when a valid address
PMTR_RA	9	O	read address of PMTR access	reflects ID_PMTR_i according to the selected channel address
PMTR_RR	1	O	read request of PMTR access; suppressed for CAIPi=1 (see DPLL_STATUS register)	reflects the value of the corresponding AENi[1)] bit while the correspondent bit CAIPi=0[2)]
ACT_D	53	O	Output of a time stamp, a position and a control signal for a calculated action; SV_i=ACT_D(52:48): ACB bits, directly written from the correspondent PMTR_D signals; ACT_D(47:24) is the calculated position value PSAC[i] for the action in relation to TBU_TS1 or 2 [6)] and ACT_D(23:0) is the time stamp value TSAC[i] for the action in relation to TBU_TS0 [6)]	Future time stamp with the resolution as TBU_TS0 input, additional position information and additional control bits;
ACT_V	1	O	ACT_D value is available and valid; blocking read access	for a valid action address: ACT_V reflects the shadow value of ACT_N[i][3)] (ACT_N[i] is 1 when new PMTR values are available and the shadow register is updated, when a calculation of the actual PMTR values was done); reset after reading of the ACT_D values
ACT_RA	9	I	ACTION read address;	address bits for selection of all 24 action channels
ACT_RR	1	I	read request of selected action	the action data is demanded from another module
IRQ	27	O	Interrupt request output	Interrupts of DPLL;
SUB_INC1	1	O	Pulse output for *TRIGGER* input filter	sub-position increment provided continuously
SUB_INC2	1	O	Pulse output for *STATE* input filter (when *TRIGGER* and *STATE* are used for 2 independent devices)	sub-position increment provided continuously
SUB_INC1c	1	O	Pulse output for time base unit 1 in compensation mode (can stop in automatic end mode)	sub-position increment related to *TRIGGER* input
SUB_INC2c	1	O	Pulse output for time base unit 2 in compensation mode (can stop in automatic end mode)	sub-position increment related to *STATE* input (when *TRIGGER* and *STATE* are used for 2 independent devices)
TS_CLK	1	I	Time stamp clock	used for generation of the time stamps; this clock is used to generate the SUB_INC1,2 pulses
CMU_CLK0	1	I	CMU clock 0	used for rapid pulse correction of SUB_INC1,2
SYS_CLK	1	I	System clock	High frequency clock
RESET_N	1	I	Asynchronous reset signal	Low active; After Reset he DPLL is available only after performing the RAM reset procedures by the DPLL hardware.
LOW_RES	1	I	low resolution of TBU_TS0 selected; shows which of the 27 bits of TBU_TS0 are connected to the DPLL	LOW_RES=0: TBU_TS0(DPLL)= lower 24 Bits of TBU_TS0(TBU); LOW_RES=1: TBU_TS0(DPLL)= higher 24 Bits of TBU_TS0(TBU); In the case LOW_RES=1 the TS0_HRT and/or TS0_HRS bits can be set [5)]

Table 38.289 Interface description of DPLL (3/3)

Name	Width	I/O	Description	Comment
TBU_TS0	24	I	Actual time stamp from TBU; is needed to decide, if a calculated action is already in the past	24 bit time input, with a resolution of the time stamp clock
TBU_TS1	24	I	Actual position/value stamp 1; for calculation of position stamps (*TRIGGER*/*STATE*)	24 bit pos./val. input, with a resolution of the SUB_INC1 pulses
TBU_TS2	24	I	Actual position/value stamp 2; to be implemented for an additional independent position	ditto for SUB_INC2 for calculation of position stamps (*STATE*) for SMC[5]=RMO[4]=1
TDIR	1	I	Direction of *TRIGGER* input values (TDIR=0 does mean a forward direction and TDIR=1 a backward direction)	direction information from multiple sensors valid only for SMC[5]=1 or IDDS[5]=1
SDIR	1	I	Direction of *STATE* input values (SDIR=0 does mean a forward direction and SDIR=1 a backward direction)	direction information from multiple sensors valid only for SMC[5]=1
DIR1	1	O	Direction information of SUB_INC1 (count forwards for DIR1=0 and backwards for DIR1=1)	count direction of TBU_CH1_BASE; DIR1 changes always after the evaluation of the corresponding valid *TRIGGER* slope and after incrementing/decrementing of the address pointer
DIR2	1	O	Direction information of SUB_INC2 (count forwards for DIR2=0 and backwards for DIR2=1)	count direction of TBU_CH2_BASE; DIR2 changes always after the evaluation of the corresponding valid *STATE* slope and after incrementing/decrementing of the address pointer
STA_T	8	O	Status of state machine TRIGGER	Output to MCS0. Signals accessible via uC interface as well (DPLL_STA)
CNT_T	3	O	Count TRIGGER	Output to MCS0. This reflects the count of active *TRIGGER* slopes (mod8). Signals accessible via uC interface as well (DPLL_STA)
STA_S	8	O	Status of state machine STATE	Output to MCS0. Signals accessible via uC interface as well (DPLL_STA)
CNT_S	3	O	Count STATE	Output to MCS0. This reflects the count of active *STATE* slopes (mod8). Signals accessible via uC interface as well (DPLL_STA)
INC_CNT1	24	O	Increment counter of pulse generator 1 (automatic end mode)	Output to MCS0. Signals accessible via uC interface as well (DPLL_INC_CNT1)
INC_CNT2	24	O	Increment counter of pulse generator 2 (automatic end mode)	Output to MCS0. Signals accessible via uC interface as well (DPLL_INC_CNT2)

38.22.5 DPLL Architecture

38.22.5.1 Purpose of the module

The DPLL generates a predefined number of incremental signal pulses within the period between two events of an input TRIGGER or STATE signal, when the corresponding pulse generator is enabled. The resolution of the pulses is restricted by the frequency of the time stamp clock (TS_CLK). Changes in the period length of the predicted time period of the current increment will result in a change of the pulse frequency in order to get the same number of pulses. This adoption can be performed by DPLL hardware, software or with support of DPLL hardware in different modes. The basic part of a DPLL is to make a prediction of the current period between two TRIGGER and/or STATE signal edges. Disturbances and systematic failures must be considered as well as changes of increment duration caused by acceleration and deceleration of the supervised process. Therefore, a good estimation is to be done using some measuring values from the past. When the process to be predicted takes a steady and differentiable course not only the current increment but also some more increments for the future can be predicted. In utilization of such calculations actions for the future can be predicted.

38.22.5.2 Explanation of the prediction methodology

As already shown in **Section 38.22.1, Overview** the DPLL has to perform different tasks. The basic function for all these tasks is the prediction of the current increment which is based on a relation between increments in the past. Because the relation between two succeeding intervals at a fixed position remains also valid in the case of acceleration or deceleration the prediction of the duration of the current time interval is done by a similarity transformation. Having a good estimation of the current time interval, all the other tasks can be done easily by calculations explained in **Section 38.22.6, Prediction of the current increment duration**.

38.22.5.3 Clock topology

All registers are read using the system clock SYS_CLK. The SUB_INC1,2 pulses generated have in the normal case the highest frequency not higher then CMU_CLK0 or the half of TS_CLK respectively. For individual pulses the frequency can be doubled. All operations can be performed using the system clock.

38.22.5.4 Clock generation

The clock is generated outside the DPLL.

38.22.5.5 Typical frequencies

For the system clock a reasonable clock frequency should be applied to give the DPLL module sufficient computational power to calculate all needed values (prediction of next increment, actions) in time. The typical system clock frequency is in the range from 40 MHz up to 150 MHz.

38.22.5.6 Time stamps and systematic corrections

The time stamps for the input signals TRIGGER and STATE have 24 bits each. These bits represent the value of the 24 bit free running counter running with a clock frequency selected by the configuration of the TBU. Using a typical frequency of 20 MHz the time stamp represents a relative value of time with a resolution of 50 ns.

The input signals have to be filtered. The filter is not part of the DPLL. The time stamps can have a delay caused by the filter algorithm used. There are delayed and undelayed filter algorithms available and the delay value can depend on a time or a position value.

Systematic deviations of TRIGGER inputs can be corrected by a profile, which also considers systematic missing TRIGGER s. The increments containing missing TRIGGER S are divided into the corresponding number of nominal increments whereas the duration of a nominal increment is the greatest divider of all increments duration. For each increment this number of enclosed nominal increments is stored in a profile as NT value for TRIGGER. When the increment is a nominal increment the NT value is 1. For the TRIGGER input the value NT is stored in the ADT_T field in RAM region 2c.

In the case of DPLL_CTRL_0. AMT = 1 the ADT_T[i] values in the RAM region 2c must also contain the adapting information for the TRIGGER signal, which considers for each increment a systematic physical deviation PD from the perfect nominal increment value with a resolution according to the chosen value of MLT+1, which describes the number of SUB_INC1 pulses for a nominal increment.

The value PD for the TRIGGER describes the amount of missing or surplus pulses with a sint13 value, to be added to MLT+1 directly. The correction value is in this way also applicable in the case of missing TRIGGER inputs for the synchronization gaps. In this case the amount of provided SUB_INC1 pulses for a nominal increment (MLT+1) + PD is multiplied by NT. The NT value of the current increment is stored in the variable SYN_T (see NUTC register in **Section 38.22.18.14, DPLL_NUTC**).

In the case of DPLL_CTRL_0.RMO = 1 for DPLL_CTRL_1.SMC =0 (emergency mode) the time stamp of STATE is used to generate the output signal SUB_INC1.

More inaccuracy should be accepted in emergency mode because usually there are only fewer events available for FULL_SCALE according to the value DPLL_CTRL_0.SNU.

For the STATE signal the systematic deviations of the increments can be corrected in the same way as for TRIGGER by profile and adaptation information as described below.

Systematic deviations of STATE inputs can be corrected by a profile, which also considers systematic missing STATE events. The increments containing missing STATEs are divided into the corresponding number of nominal increments whereas the duration of a nominal increment is the greatest divider of all increments duration. For each increment this number of enclosed nominal increments is stored in a profile as NS value for STATE. When the increment is a nominal increment the NS value is 1. For the STATE input the value NS is stored in the ADT_S field in RAM region 1c3.

In the case of DPLL_CTRL_0.AMS = 1 the ADT_S[i] values in the RAM region 1c3 must contain the adapting information for the STATE signal, which considers for each increment a systematic physical deviation PD_S from the perfect nominal increment value with a resolution according to the chosen value of MLS1, which describes the number of SUB_INC1 pulses for a nominal increment (see below).

The number of pulses SUB_INC1 for a nominal STATE increment in emergency mode (for SMC=0) is given by the value of MLS1= (MLT + 1)* (TNU + 1) /(SNU + 1) in order to get the same number of pulses in FULL_SCALE for normal and emergency mode. This value has to be configured by the CPU.

The value PD_S for the STATE describes the amount of missing or surplus pulses with a sint16 value, to be added to MLS1 directly. The correction value is in this way also applicable in the case of missing STATE inputs for the synchronization gaps. In this case the amount of provided SUB_INC1 pulses for a nominal increment MLS1 + PD_S is multiplied by NS. The current NS value is stored in the variable SYN_S (see NUSC register in **Section 38.22.18.15, DPLL_NUSC**).

38.22.5.7 DPLL Architecture overview

As shown in **Figure 38.111, DPLL Block Diagram** the DPLL can process different input signals. The signal TRIGGER is the normal input signal which gives the detailed information of the supervised process. It can be for instance the information of water or other liquid level representing the volume of the liquid, where each millimeter increasing results in a TRIGGER signal generation. In order to get a predefined filling level, without overflow also the inertia of the system must be taken into account. Hence, some delay for closing the inlet valve and also the remaining water amount in the pipe must be considered in order to start the closing action earlier as the filling level will be reached.

A second input signal STATE sends an additional (redundant) information for instance at some centimeters and because of intervals with different distances it gives also information about the system state with the direction of the water flow (in or out), while the TRIGGER signal must not contain information concerning the flow direction. In some applications the inactive slope of TRIGGER can be utilized to transmit direction information. In the case of faults in the TRIGGER signal the STATE signal is to be processed in order to reach the desired value nevertheless, maybe with some loss of accuracy.

The measuring scale can have some systematic failures, because not all millimeter or centimeter distances measured mean the same value. This could be due to changes in the thickness of the measuring cylinder or the inaccurate position of the marks. These systematic failures are well known by the system and for improvement of the prediction the signals ADT_T and ADT_S for the correction of the systematic failures of TRIGGER and STATE respectively are stored in the internal RAM.

The input signals TRIGGER and STATE are represented as a time stamp signal each, which is stored in the 24 bit TS-part of the corresponding signal.

Information concerning the delay of this signal by filtering of disturbances is stored in the 24 bit FT-part of the signal.

In order to establish the relation of time stamps to the actual time the TBU_TS0 ^{2} value is also provided showing the actual time value used for prediction of actions in the future.

After reaching the desired water level the water is filled in a bottle by draining. After that the water filling is repeated. The water level at draining is observed by the same sensor signals (the same number of TRIGGER pulses), but the duration of the draining could be different from the filling time. Both times together form the FULL_SCALE region, while one of them is a HALF_SCALE region, which can differ in time but not in the number of pulses, especially for TRIGGER.

For synchronization purposes some TRIGGER marks can be omitted in order to set the system to a proper synchronization value (maybe before the upper filling value is reached).

In emergency situations, when the TRIGGER signals are missed the STATE signal is used instead of.

The PMTR_i input signals announce the request for a position minus time calculation for up to 24 events.

All 24 events can be activated using the 24 DPLL_CTRL_x.AENi=1 for x=(2,3,4) (action enable) bits. Each of these enable bits are asked by the routing engine for a read access. The corresponding read request is generated by the AENi bit while CAIPx is zero. CAIP1 and CAIP2 are two bits of the DPLL_STATUS register for 12 actions each with the meaning calculation of actions in progress, controlled by the state machine (see **Section 38.22.2, Requirements and demarcation**) for scheduling the operations.

When such a request is serviced by the ARU (in the case CAIPx=0) the values for position and time are written in the corresponding RAM 1a region (0200_H to $025C_H$ for the position value and 0260_H to $02BC_H$ for the delay value), the control bits for the corresponding action are set accordingly. When a

new PMTR value arrives, an old value is overwritten without notice and the shadow bit of ACT_N[i] is cleared while the ACT_N[i] (new action) bit in the DPLL_ACT_STA register is set. The ACT_N[i] is cleared, when the currently calculated action value is in the past. Overwriting of old information is possible without data inconsistency because the read request to ARU is suppressed during action calculations by the CAIP1,2 bits. In this way always the last possible PMTR value is used consistently.

38.22.5.8 DPLL Architecture description

Figure 38.111, DPLL Block Diagram will now be explained in detail in combination with some example configurations of the control registers. There are different configuration bits available which can adopt the DPLL to the use case (see **Section 38.22.18, DPLL Register description**). Let for example in HALF_SCALE the TRIGGER number DPLL_CTRL_0.TNU be $3B_H$ (which is for TNU+1 = 60 decimal that does mean 120 events in FULL_SCALE) and the number of SUB_INC1 pulses between two TRIGGER s DPLL_CTRL_0.MLT be 257_H (this means 600 pulses per TRIGGER event). Than the FULL_SCALE region can be divided into 72000 parts each of them associated with its own SUB_INC1 pulse. For a run through FULL_SCALE all 72000 pulses should appear but maybe with a different pulse frequency between two TRIGGER events. For this example after each 600 pulses at the SUB_INC1 output the next TRIGGER event is to be expected with the corresponding new time stamp.

Missing SUB_INC1 pulses due to acceleration have to be taken into account within the next increment. Not one pulse has to be missed or added because of calculation inaccuracy in average for a sufficient number of FULL_SCALE periods. This means that not one pulse is sent in addition and all missing pulses are to be caught up on afterwards.

For the systematic arrangement of TRIGGER inputs the profile (as already mentioned in **Section 38.22.5.6, Time stamps and systematic corrections** is stored in the RAM region 2c (see **Section 38.22.24.4, DPLL_ADT_T[i]**). In this field the relative position of gaps can be stored in the NT value and also physical deviations in the PD value.

For the consideration of systematic missing TRIGGER s the actual NT value of the profile is stored in the SYN_T bits of the NUTC register (see **Section 38.22.18.14, DPLL_NUTC**).

In normal mode the physical deviation values PD in the ADT_T field could be used to balance the local systematic inaccuracy of the TRIGGER signal. The value of PD (see **Section 38.22.24.4, DPLL_ADT_T[i]**) is the pulse difference in the corresponding nominal increment and does mean the number of sub pulses to be added to the nominal number of pulses. PD is a signed integer value using 13 bits: up to +/–4096 pulses can be added for each increment.

The NT value of the profile ADT_T has the value 1, when a nominal increment is assumed. An integer number greater than 1 shows the number of nominal increments to be considered for a gap. For the actual increment after synchronization the corresponding NT value is stored in SYN_T of the NUTC register.

Using the STATE input there are similar configuration bits available (see **Section 38.22.18, DPLL Register description**). Let for example in HALF_SCALE the STATE number DPLL_CTRL_0.SNU be 0xB (which is for DPLL_CTRL_0.SNU+1 =12 decimal and while DPLL_CTRL_0.SYSF =0 that does mean 24 events in FULL_SCALE). In order to get the same number of SUB_INC1 pulses for FULL_SCALE as above for TRIGGER s the value (DPLL_CTRL_0.ML- T+1)=600 is divided by 2*(DPLL_CTRL_0.SNU+1)=24 and multiplied with 2*(DPLL_CTRL_0.TNU+1)=120. The result 3000 must be stored in MLS1 by the CPU (see **Section 38.22.22.39, DPLL_MLS1**).

For the systematic arrangement of STATE inputs the profile (as already mentioned in **Section 38.22.5.6, Time stamps and systematic corrections** is stored in the RAM region 1c3 (see

Section 38.22.22.54, DPLL_ADT_S[i]). In this field the relative position of gaps can be stored in the NS value and also physical deviations in the PD_S value.

For the consideration of systematic missing TRIGGER s the actual NS value of the profile is stored in the SYN_S bits of the NUSC register (see **Section 38.22.18.15, DPLL_NUSC**).

In emergency mode the physical deviation values PD_S in the ADT_S field could be used to balance the local systematic inaccuracy of the STATE signal. The value of PD_S (see **Section 38.22.22.54, DPLL_ADT_S[i]**) is the pulse difference in the corresponding increment and does mean the number of sub pulses to be added to the nominal number of pulses per increment. PD_S is a signed integer value using 16 bits: up to +/–32768 pulses can be added for each increment.

In emergency mode the physical deviation values PD_S in the ADT_S field could be used to balance the local systematic inaccuracy of the STATE signal. The value of PD_S (see **Section 38.22.22.54, DPLL_ADT_S[i]**) is the pulse difference in the corresponding nominal increment and does mean the number of sub pulses to be added to the nominal number of pulses. PD_S is a signed integer value using 16 bits: up to +/–32768 pulses can be added for each increment.

The NS value of the profile ADT_S has the value 1, when a nominal increment is assumed. An integer number greater than 1 shows the number of nominal increments to be considered for a gap. For the actual increment after synchronization the corresponding NS value is stored in SYN_S of the NUSC register.

38.22.5.9 Block diagrams of time stamp processing.

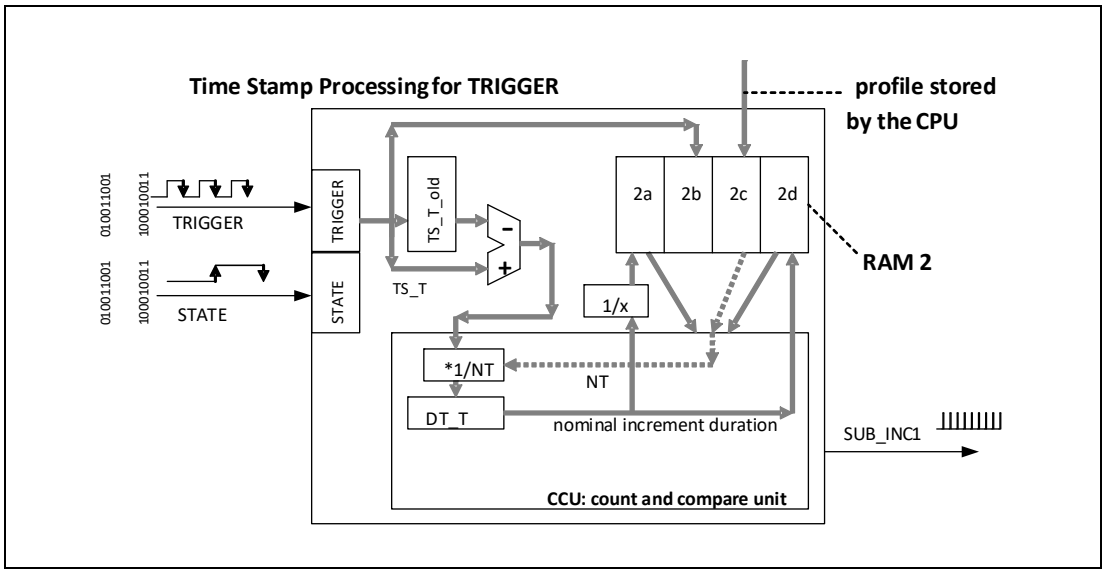


Figure 38.112 Time Stamp Processing Trigger

As shown in the block diagram above the time stamp difference of two succeeding input events is calculated. For the prediction of the current increment duration such values from the past are used. For this purpose the measured and calculated values of the last FULL_SCALE period are stored in the RAM. For the TRIGGER input there are 4 different RAM parts in the RAM region 2: 2a stores the reciprocals of each nominal increment duration RDT_T 2b stores the time stamps of each active input event TSF_T 2c is used for the profile ADT_T and 2d for the nominal increment duration DT_T. Because the prediction is based on the relations of increments in the past this relation can be calculated easily by the multiplication of increment duration values with the reciprocal value of another increment. In order not to be forced to distinguish between gaps and "normal" increments duration also for gaps only the nominal duration and the correspondent reciprocal values are stored in the RAM field. This is possible by consideration of the NT value in the profile: the measured increment duration is divided by NT.

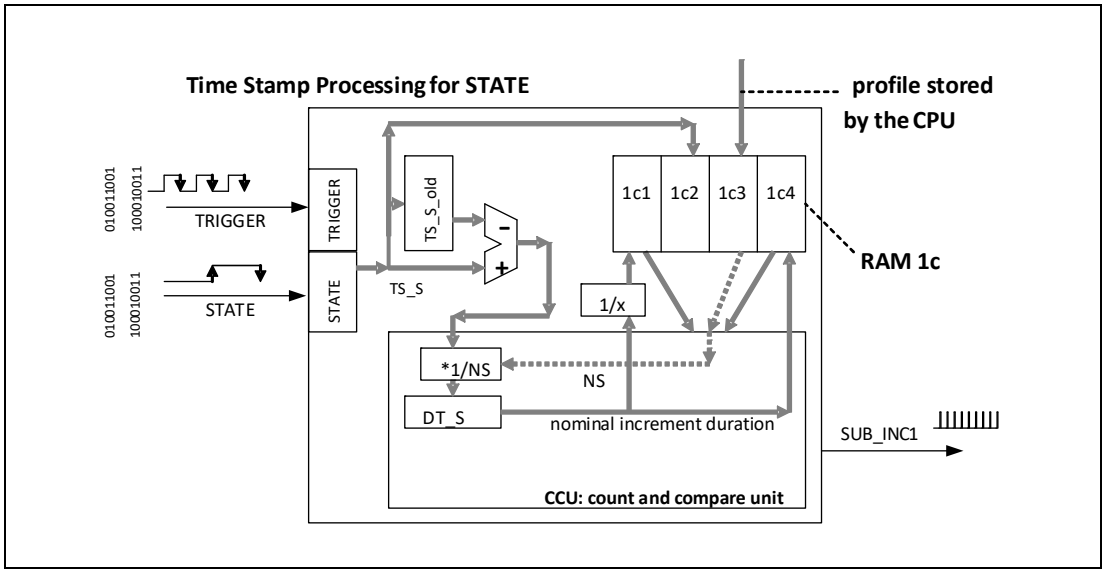


Figure 38.113 Time Stamp Processing State

For the *STATE* input there are also 4 different RAM parts in the RAM region 1c:

1. 1c1 stores the reciprocals of each nominal increment duration RDT_S
2. 1c2 stores the time stamps of each active input event TSF_S
3. 1c3 is used for the profile ADT_S and
4. 1c4 for the nominal increment duration DT_S.

The calculations are performed similar as for the TRIGGER input. The NS value in the profile shows the appearance of a gap.

38.22.5.10 Register and RAM address overview

The address map of the DPLL is divided into register and memory regions as defined in Table (DPLL_1052)[XREF TARGET " (DPLL_1052)] NOT EXIST. The addresses from 0000_H to $00FC_H$ are reserved for registers, from 0100_H to $01FC_H$ is reserved for action registers to serve the ARU at immediately read request. The RAM is divided into 3 independent accessible parts 1a, 1b+c and 2. The part 1a from 0200_H to $037C_H$ is used for PMTR values got from ARU and intermediate calculation values; there is no write access from the CPU possible, while the DPLL is enabled. The RAM 1b part from 0400_H to $05FC_H$ is reserved for RAM variables and the RAM part 1c from 0600_H to $09FC_H$ is used for the STATE signal values. The RAM region 2 from 4000_H to $7FFC_H$ is reserved for the TRIGGER signal values. RAM region 1a has a size of 288 bytes, Ram 1b+c uses 1,125 Kbytes while RAM region 2 is configurable from 1,5 to 12 Kbytes, depending on the number of TRIGGER events in FULL_SCALE. The AOSV_2 register is used to determine the beginning of each part.

The table in (DPLL_1052)[XREF TARGET " (DPLL_1052)] NOT EXIST gives the DPLL Address map overview

Registers are used to control the DPLL and to show its status. Also parameters are stored in registers when useful. The table below shows the addresses for status and control registers as well as values stored in additional registers. The register meaning explained in the register overview (**Section 38.22.11, DPLL Register Memory overview**) while the bit positions of the status and control registers are described in detail in **Section 38.22.18, DPLL Register description**.

Time stamps for TRIGGER and STATE can have either the same resolution as the TBU_TS0 input or 8 times higher. This is configured in the DPLL_CTRL_1 register (see **Section 38.22.18.2, DPLL_CTRL_1**). While the TBU_TS0 is used for action predictions the higher resolution of TRIGGER and STATE inputs can be used for a more accurate pulse generation.

The time stamp fields of TRIGGER and STATE are stored in the corresponding RAM regions in such a way, that for a gap also entries for the virtual increments are provided. This is due to the necessity to calculate time differences between a given number of (real and virtual) input events independent of a gap. Therefore the gap is extended in the RAM fields 2b and 1c2. For all other RAM regions in RAM 2 and RAM 1c the gap is considered as one increment.

For the access to the RAM fields there must be address pointers. When the device starts all address pointers have a zero value and the first measured and calculated values are stored in the beginning of the corresponding RAM field. Because the position of the device is usually unknown at the beginning no profile information can be used. The profile regions must have their own address pointers each which are set by the CPU as soon as the position is known. By setting the appropriate value to the address pointer APT_2C of the TRIGGER profile or APS_1C3 of the STATE profile respectively the synchronization bits in the DPLL_STATUS register SYT or SYS are set respectively. In the following the gap information can be used.

Because the time stamp fields are extended at the gaps there must be additional address pointers for these regions: APT_2B for TRIGGER time stamps and APS_1C2 for STATE time stamps. These address pointers must be incremented by NT or NS respectively when a gap appears.

Table 38.290 Register and RAM address map (1/2)

Addr. range Start	Addr. range End	Value number	Byte #	Content	Indication	Region	RAM size
Addr. range Start	Addr. range End	Value number	Byte #	Content	Indication	Region	RAM size
0x0000	0x0FC	64	256	Register	used/reserved	0	no RAM
0x100	0x1FC	64	192	ACTION registers	direct read from ARU	0	no RAM

Table 38.290 Register and RAM address map (2/2)

Addr. range Start	Addr. range End	Value number	Byte #	Content	Indication	Region	RAM size
0x0200	0x03FC	128	384	PMTR values RAM 1a	CPU R/Pw access, when DPLL disabled; ARU has highest priority	1a with own ports	RAM part 1a: 384 bytes
0x0400	0x05FC	128	384	Variables RAM 1b	R and monitored W access by the CPU	1b	RAM part 1b+c: 1,125 Kbytes
0x0600	0x09FC	256	768	*STATE* data	R and monitored W access by the CPU	1c	
0x0600	0x06FC	64	192	RDT_S[i]	*STATE* reciprocal values	1c1	
0x0700	0x07FC	64	192	TSF_S[i]	*STATE* TS values	1c2	
0x0800	0x08FC	64	192	ADT_S[i]	adapted values of *STATE*	1c3	
0x0900	0x09FC	64	192	DT_S[i]	nom. STATE inc.	1c4	
0x4000	0x47FC ... 0x7FFC	512 ... 4096	1536 ... 12288	*TRIGGER* data	R and monitored W access of CPU	2	RAM part 2: 1,5... 12 Kbytes
0x4000	0x41FC...4FFC	128... 1024	384...3072	RDT_T[i]	TRIGGER reciprocal values	2a	
0x4200...5000	0x43FC...5FFC	128... 1024	384...3072	TSF_T[i]	*TRIGGER* TS values	2b	
0x4400...6000	0x45FC...6FFC	128... 1024	384...3072	ADT_T[i]	adapted values of *TRIGGER*	2c	
0x4600...7000	0x47FC...7FFC	128... 1024	384...3072	DT_T[i]	nom. *TRIGGER* increments	2d	

(1) RAM Region 1

RAM region 1 has a size of 1,5 Kbytes and is used to store variables and parameters as well as the measured and calculated values for increments of STATE. The RAM 1 region is divided into two independent accessible RAM parts (a and b+c) with own ports. The address information is shown in the table above and the detailed description is performed in the following chapters. The RAM 1a is used to store the PMTR values got from ARU and in addition some intermediate calculation results of actions. RAM region 1b is used for variables needed for the prediction of increments, while RAM 1c is used to store time stamps, profile and duration of all the STATE inputs of the last FULL_SCALE region. All variables and values of RAM 1b+c part use a data width of up to 24 bits.

The RAM is to be initialized by the DPLL after HW-reset. All RAM cells must have a zero value after performing the initialize procedure. This is performed when setting The Init_RAM bit in the DPLL_RAM_INI register. The DPLL is only available after finishing this procedure. The initialization progress is shown in the status bits of the same register.

RAM Region 1a:

Used for storage of PMTR values got from ARU; read and write access by the CPU is only possible, when the DPLL is disabled.

The CPU Address range: 0200_H to $03FC_H$

RAM Region 1b:

Usable for intermediate calculations and auxiliary values, data width of 3 bytes used for 24 bit values; A write access to this region results in an interrupt to the CPU, when enabled.

Address range: 0400_H to $05FC_H$

RAM Region 1c :

Values of all STATE increments in FULL_SCALE, data width of 3 bytes used for 24 bit values; A write access to this region results in an interrupt to the CPU, when enabled.

Address range: 0600_H to $09FC_H$

In RAM region 1c there is a difference in the amount of data. While for the RAM regions 1c1, 1c3 and 1c4 there are 2*(SNU+1–SYN_NS) entries for SYSF=0 or 2*(SNU+1) –SYN_NS entries for SYSF=1, for the RAM region 1c2 there are 2*(SNU+1) entries (see DPLL_CTRL_0 and _1 registers). For the latter also the virtual events are considered, that means the gap is divided into equidistant parts each having the same position share as increments without a gap. For that reason the CPU must extend the stored TSF_S[i] values in the RAM region 1c2 before the APS_1C3 is written. The write access to APS_1C3 sets the SYS bit in the DPLL_Status register in order to show the end of the synchronization process. Only when the SYS bit is set the PMTR values can consider more than the last increment duration for the action prediction by setting NUSE to a corresponding value.

NOTE

RAM regions 1b and 1c have a common port.

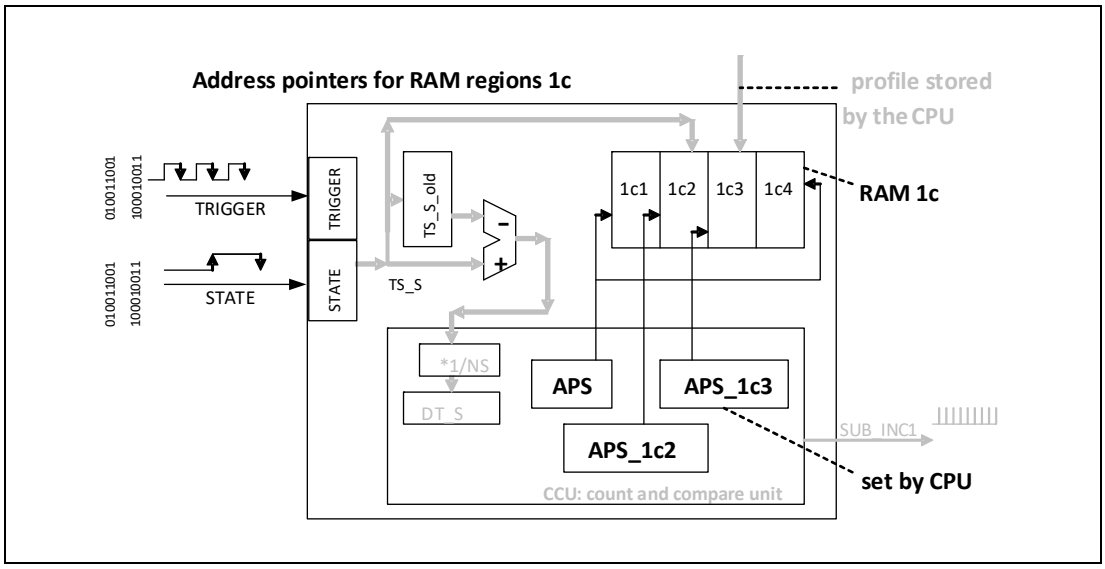


Figure 38.114 Address Pointer for RAM 1c

The address pointers for RAM region 1c are shown in the diagram above. While the address pointer APS points to the RAM regions 1c1 and 1c4, the address pointer APS_1C2 points to the time stamp field in the region 1c2. This is necessary, because in the time stamp field the gaps are extended to the number of nominal increments (see explanation above and also to the synchronization procedure explained in **Section 38.22.8.6, (1) Synchronization description**). The address pointer APS_1C3 is set by the CPU when the position is known and therefore the relation to the other address

pointers is calculated. This setting of this profile address pointer synchronizes the RAM fields to one another. The synchronization is shown in the DPLL_STATUS register (see **Section 38.22.18.30, DPLL_STATUS**) by the SYS bit.

(2) RAM Region 2

The RAM region 2 has a configurable size of 1,5 to 12 Kbytes and is used to store measured and calculated values for increments of TRIGGER. The address information is explained in **Section 38.13, Cluster Configuration Module (CCM)** while the meaning is explained in this chapter.

Because of up to 512 TRIGGER events in HALF_SCALE the fields 2a, b c and d must have up to 1024 storage places each. For 3 Bytes word size this does mean up to 12 Kbyte of RAM region 2.

In order to save RAM size for configurations with less TRIGGER events the RAM is configurable by the offset switch Register OSW ($001C_H$) and the address offset value register of RAM region 2 AOSV_2 (0020_H). The RAM is to be initialized by the DPLL after HW-reset. All RAM cells must have a zero value after performing the initialize procedure. The DPLL is only available after finishing this procedure.

In RAM region 2 there is a difference in the amount of data. While for the RAM regions 2a, 2c and 2d there are 2*(TNU+1–SYN_NT) entries, for the RAM region 2b there are 2*(TNU+1) entries (see **Section 38.22.18.1, DPLL_CTRL_0** and **Section 38.22.18.2, DPLL_CTRL_1**). For the latter also the virtual events are considered, that means the gap is divided into equidistant parts each having the same position share as increments without a gap. For that reason the CPU must extend the stored TSF_T[i] values in the RAM region 2b before the APT_2C is written.

The write access to APT_2C sets the SYT bit in the DPLL_Status register in order to show the end of the synchronization process. Only when the SYT bit is set the PMTR values can consider more than the last increment duration for the action prediction by setting NUTE to a value greater than one.

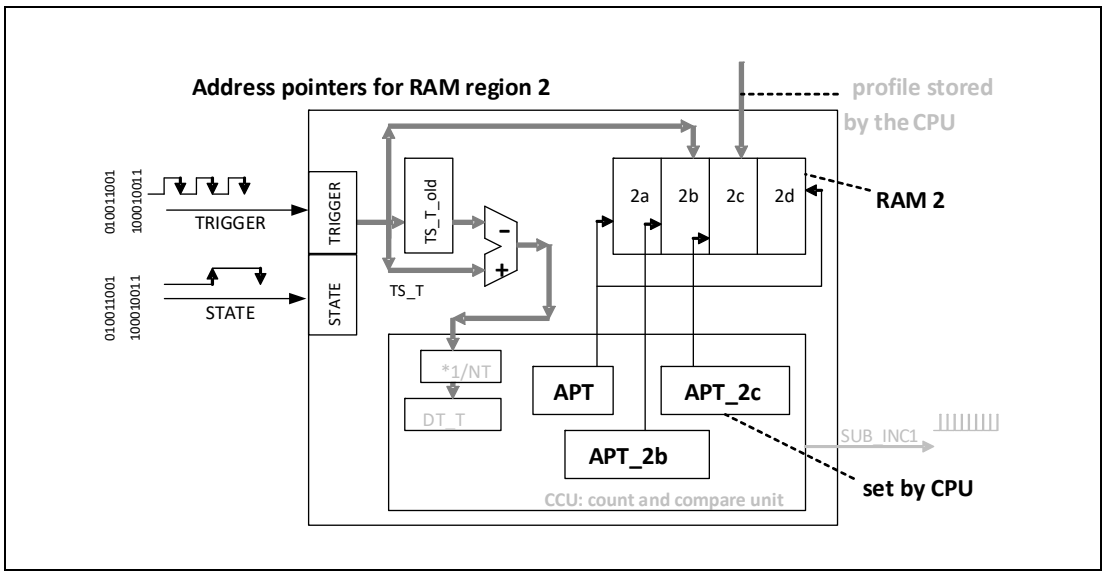


Figure 38.115 Address Pointer for RAM 2

The address pointers for RAM region 2 are shown in the diagram above. While the address pointer APT points to the RAM regions 2a and 2d, the address pointer APT_2B points to the time stamp field in the region 2b. This is necessary, because in the time stamp field the gaps are extended to the number of nominal increments (see explanation above and also to the synchronization procedure explained in **Section 38.22.8.6, (1) Synchronization description**). The address pointer APT_2C is set by the CPU when the position is known and therefore the relation to the other address pointers is calculated. This setting of this profile address pointer synchronizes the RAM fields to one another. The synchronization is shown in the DPLL_STATUS register (see **Section 38.22.18.30, DPLL_STATUS**) by the SYT bit.

38.22.5.11 Software reset and DPLL deactivation

The DPLL module allows different options of deactivation and/or reset. To stop the operation of the DPLL module it is possible to deactivate the DPLL by setting of DPLL_CTRL_1.DEN = 0. This stops the calculations for the generation of the sub increments and the actions. Some control register areas are only configurable in this mode, some but not all register signals are set into an initial state. The RAM memory is not affected by DPLL deactivation at all. The behavior of the DPLL output signals and registers when deactivated is described in this document. The deeper option to reconfigure the DPLL is the use of the software reset. When the DPLL module is deactivated setting DPLL_CTRL_1.SWR

= 1 performs a reset of all DPLL registers and state controllers. The RAM memory is not affected by the software reset at all. After the software reset the DPLL module remains in deactivated state and the control registers must be configured again before operation (activation by DEN = 1) can start again. The RAM modules can be reset (written to all zero) by activation of the memory initialization control bit (INIT_RAM) of the register DPLL_RAM_INI. If the RAM initialization is automatically done after power on reset or not depends on the GTM implementation. A special case is the configuration of the control bit DPLL_CTRL_11.STATE_EXT. If this bit shall be modified during operation a software reset of the DPLL module is strongly recommended. A RAM initialization should also be considered depending on the given application case.

38.22.6 Prediction of the current increment duration

38.22.6.1 The use of increments in the past

Past values to be considered for the prediction of TRIGGER

In order to take into account values of increments for TRIGGER s in the past, the NUTE value is configured to determine the number of past values. In addition the VTN has a value according to the number of virtual increments in the NUTE region. Because gaps come in to the NUTE region or leave it the VTN value must be updated by the CPU until NUTE is set to HALF_SCALE or FULL_SCALE. For the RAM regions 2a and 2d the value NUTE–VTN is to be considered while for the RAM region 2b only the NUTE value is to be considered. This is due to the fact that the time stamp entries in a gap are extended to the number of nominal increments, but duration entries not.

Past values to be considered for the prediction of STATE

In order to take into account values of increments for STATE in the past, the NUSE value is configured to determine the number of past values. In addition the VSN has a value according to the number of virtual increments in the NUSE region. Because gaps come in to the NUSE region or leave it the VSN value must be updated by the CPU until NUSE is set to HALF_SCALE or FULL_SCALE. For the RAM regions 1c1 and 1c4 in the past the value NUSE–VSN is to be considered while for the RAM region 1c2 only the NUSE value is to be considered. This is due to the fact that time stamp entries in a gap are extended to the number of nominal increments, but duration entries not.

38.22.6.2 Increment prediction in Normal Mode and for first PMSM forwards

For the prediction of increments and actions in normal mode the values are calculated as described in the following equations.

The ascending order of calculation must be hold in order not to lose results still needed. It is important for TRIGGER values to calculate and store in the RAM region 2 all values according to equations up to DPLL-14 before DPLL-1a4 to 7, DPLL-1b1 and DPLL-1c1, while the last one overwrites DT_T[i] when NUTE (see **Section 38.22.18.14, DPLL_NUTC**) is set to the FULL_SCALE range. Because the old value of DT_T[i] is also needed for equation DPLL-10 and DPLL-11 this value is stored temporarily at DT_T_ACT as shown by equation **Section 38.22.6.2, (1) Equations DPLL-1a to calculate TRIGGER time stamps** or **Section 38.22.6.2, (2) Equation DPLL-1b to calculate DT_T_ACT (nominal value)** respectively until all prediction calculations are done and after that equation DPLL-1a4 to 7 and DPLL-1b1 updates DT_T[i]: update DT_T[i] after calculations of equation DPLL-14. For p=APT calculates in normal mode.

When using filter information of TRIGGER_FT, selected by IDT=1, it must be distinguished by IFP, if this filter information is time or position related. In order to make possible to perform the automatic resolution corrections of equations (DPLL_10988)[XREF TARGET " (DPLL_10988)] NOT EXIST and (DPLL_10989)[XREF TARGET " (DPLL_10989)] NOT EXIST the filter unit in TIM module must operate using the time stamp clock.

(1) Equations DPLL-1a to calculate TRIGGER time stamps

For calculation of time stamps use the filter delay information and an additional TRIGGER input delay value stored in register TIDEL (initial zero)

TS_T_1 = TRIGGER_TS – TIDEL (DPLL-1a0)

$TS\_T = TS\_T_1$ – FTV_Tx (for IDT=1 and IFP=0) (DPLL-1a1) with

FTV_Tx = FTV_T/8 (for LOW_RES = 1 and TS0_HRT = 0) (DPLL-1a1a)

FTV_Tx = FTV_T (for LOW_RES = 0 or TS0_HRT = 1) (DPLL-1a1b) and

$TS\_T = TS\_T_1$ - FTV_T *(CDT_TX/NMB_T)_old (for IDT=1 and IFP=1) (DPLL-1a2) this can be also calculated using the value of ADD_IN_CALN:

$TS\_T = TS\_T_1$ – FTV_T *(1/ADD_IN_CALN_old) (for IDT=1 and IFP=1) (DPLL-1a3)

For (CDT_TX/NMB_T)_old and (1/ADD_IN_CALN_old) consider values, calculated for the last increment; position related filter values are only considered up to at least 1 ms time between two TRIGGER events. The reciprocal value is stored using a 32 bit fractional part, while only the 24 lower bits are used – for explanation see note (DPLL_10980)[XREF TARGET " (DPLL_10980)] NOT EXIST at DPLL_CTRL_0 register. The value of 1/ADD_IN_CALN_old or (CDT_TX/NMB_T)_old is set to $FFFFFF_H$ in the case of an overflow.

CDT_TX is the predicted duration of the last TRIGGER increment and NMB_T the calculated number of SUB_INC1 events in the last increment, because the new calculations are done by equations DPLL-5 and DPLL-21 for the current increment after that. Therefore in equation (DPL- L_10991)[XREF TARGET " (DPLL_10991)] NOT EXIST the value ADD_IN of the last increment is used (see equation DPLL-25). SYN_T_OLD is the number of TRIGGER events including missing TRIGGERs as specified in the NUTC register for the last increment, with the initial value of 1.

For storage of time stamps in the RAM see also equations DPLL-1a4 ff. after calculation of actions, **Section 38.22.7.5, (1) Equation DPLL-1a4 to update the time stamp values for TRIGGER**.

(2) Equation DPLL-1b to calculate DT_T_ACT (nominal value)

DT_T_ACT= (TS_T – TS_T_OLD)/SYN_T_OLD (DPLL-1b)

For the case SYT=0 (still no synchronization to the profile) the values SYN_T and SYN_T_OLD are still assumed as having the value 1.

Correct the current increment duration value got by equation (DPLL_6911)[XREF TARGET " (DPLL_6911)] NOT EXIST in the case of physical deviations (ADT=1) by DT_T_ACT = DT_T_ACT * (1 – PDC_T + PDC_T –PDC_T) (DPLL-1b1)

with the relative correction value for the last increment

PDC_T = PD_OLD / (MLT+1) (DPLL-1b2) (for SMC=0)

PDC_T = PD_OLD / (MLS1) (for SMC=1) (DPLL-1b3) The term (1 – PDC_T + PDC_T –PDC_T) of equation (DPLL_10998)[XREF TARGET " (DPLL_10998)]NOT EXIST is representing the third order Taylor series of the term 1/(1+PDC_T), which is chosen to reduce the additional time delay due to the more complex computation.

(3) Equation DPLL-1c to calculate RDT_T_ACT (nominal value)

RDT_T_ACT = 1 / DT_T_ACT (DPLL-1c)

(4) Equation DPLL-2a1 to calculate QDT_T_ACT

Relation of the recent last two increment values for APT=p in forward direction (DIR1=0)
QDT_T_ACT = DT_T_ACT * RDT_T[p–1] (DPLL-2a1)

QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

NOTE

QDT_T_ACT uses a 6 bit integer part and an 18 bit fractional part.

(5) Equation DPLL-3 to calculate the error of last prediction

When q = NUTE–VTN considers for the error calculation only the last valid prediction values for DIR1=0: Calculate the error of the last prediction when using only RDT_T_FS1, DT_T[p–q], DT_T[p–q–1] and DT_T[p–1] for the prediction of DT_T[p]:

EDT_T = DT_T_ACT – (DT_T[p–1] * QDT_T[p–q]) (DPLL-3) with

QDT_T[p–q] = DT_T[p–q] * RDT_T[p–q–1] (for FST= 0) (DPLL-2b1)

QDT_T[p–q] = DT_T[p–q] * RDT_T_FS1 (for FST=1) (DPLL-2b2)

FST = 1 has the meaning: NUTE=FULL_SCALE (see **Section 38.22.18.14, DPLL_NUTC**) while QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

NOTE

QDT_T[p – q] uses a 6 bit integer part and an 18 bit fractional part.

(6) Equation DPLL-4 to calculate the weighted average error

for SYT=1 calculate:

MEDT_T := (EDT_T + MEDT_T) / 2(DPLL-4)

(7) Equations DPLL-5 to calculate the current increment value

nominal increment value (for ADT=0):

CDT_TX _nom = (DT_T_ACT + MEDT_T) * QDT_T[p–q+1] (DPLL-5a1)

nominal increment value (for ADT=1):

CDT_TX _nom_corr = CDT_TX _nom * (1 + CDC_T) (DPLL-5a2) with:

CDC_T = PD / (MLT+1) (for SMC=0) (DPLL-5a3)

CDC_T = PD / (MLS1) (for SMC=1) (DPLL-5a4) and

QDT_T[p–q+1] = DT_T[p–q+1] * RDT_T[p–q] (DPLL-2c) (for q > 1) and for q=1 use equation **Section 38.22.6.2, (4) Equation DPLL-2a1 to calculate QDT_T_ACT**.

while QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part. The CDT_TX_nom value is limited by the relation

CTN_MIN < CDT_TX_nom < CTN_MAX When the calculated value exceeds one of the limits, it is replaced by the corresponding limit value. The expected duration to the next TRIGGER event is:

CDT_TX = CDT_TX _nom * SYN_T (DPLL-5b)(for ADT=0) CDT_TX = CDT_TX _nom_corr * SYN_T
(for ADT=1)

QDT_T[p–q+1] uses a 6 bit integer part and an 18 bit fractional part.

NOTE

In the case of an overflow in equations (DPLL_11005)[XREF TARGET '' (DPLL_11005)] NOT EXIST or (DPLL_11016)[XREF TARGET '' (DPLL_11016)] NOT EXIST set the value to $FFFFFF_H$ and the corresponding CTO bit in the DPLL_STATUS register. In the case of negative values set CDT_TX to 0_H without any effect to the CTO bit.

38.22.6.3 Increment prediction in Emergency Mode and for second PMSM forwards

Note that the ascending order of calculations for STATE and storage of the values in the RAM region 1c must be hold in order not to lose results still needed. The same considerations as done for DT_T_ACT are valid for DT_S_ACT (equation DPLL-6a4...7, DPLL-6b1 and DPLL-6b1): update TD_S[i] only after calculations of equation DPLL-14.

When using filter information of STATE_FT, selected by IDS=1, it must be distinguished by IFP, if this filter information is time or position related. In order to make possible to perform the automatic resolution corrections of equation (DPLL_11019)[XREF TARGET '' (DPLL_11019)] NOT EXIST the filter unit in TIM must operate using the time stamp clock.

(1) Equations DPLL-6a to calculate STATE time stamps

For calculation of time stamps use the filter delay information, the additional STATE input delay value stored in the register SIDEL (initial zero) and use p=APS while DIR2=0:

TS_S_1= STATE_TS-SIDEL (DPLL-6a0)

TS_S = TS_S_1-FTV_Sx (for IDS=1 and IFP=0) (DPLL-6a1) with

FTV_Sx = FTV_S / 8 (for LOW_RES = 1 and TS0_HRS = 0) (DPLL-6a1a)

FTV_Sx = FTV_S (for LOW_RES = 0 or TS0_HRS = 1) (DPLL-6a1b) and

TS_S = TS_S_1- FTV_S * (CDT_SX / NMB_S)_old (for IDS=1 and IFP=1) (DPLL-6a2) this can be also calculated using the value of ADD_IN_CALE:

TS_S = TS_S_1-FTV_S * (1 / ADD_IN_CALE)_old (for IDS=1 and IFP=1) (DPLL-6a3) with see also equations DPLL.6a4 ff. at **Section 38.22.6.2, Increment prediction in Normal Mode and for first PMSM forwards** for TRIGGER.

For (CDT_SX / NMB_S)_old and (1/ADD_IN_CALE)_old consider values, calculated for the last increment; position related filter values are only considered up to at least 1 ms time between two STATE events. The reciprocal value is stored using a 32 bit fractional part, while only the 24 lower bits are used – for explanation see note (DPLL_10980)[XREF TARGET '' (DPLL_10980)] NOT EXIST. The value of 1/ADD_IN_CALE_old or (CDT_SX/NMB_S)_old is set to 0xFFFFFF in the case of an overflow.

NOTE

CDT_SX is the predicted duration of the last STATE increment and NMB_S the calculated number of SUB_INC1 events in the last increment, because the new calculations are done by equations DPLL-10 and DPLL-22 respectively for the current increment after that. Therefore in equation (DPLL_11022)[XREF TARGET '' (DPLL_11022)] NOT EXIST the value ADD_IN of the last increment is used (see equation DPLL-26). SYN_S_OLD is the number of increments including missing STATE s as specified in the NUSC register for the last increment with the initial value of 1. The update to the RAM region 1c4 is done after all related calculations (see equation DPLL-6b1 for this reason).

(2) Equation DPLL-6b to calculate DT_S_ACT (nominal value)

DT_S_ACT= (TS_S – TS_S_OLD) / SYN_S_OLD (DPLL-6b)

For the case SYS=0 (still no synchronization to the profile) the values SYN_S and SYN_S_OLD are still assumed as having the value 1.

Correct the current increment duration value got by equation (DPLL_6917)[XREF TARGET " (DPLL_6917)] NOT EXIST in the case of physical deviations (ADS=1) by

DT_S_ACT = DT_S_ACT * (1 – PDC_S + PDC_S – PDC_S) (DPLL-6b1)

with the relative correction value for the last increment

PDC_S = PD_S_OLD/(MLS1) (DPLL-6b2)(for SMC=0)

PDC_S = PD_S_OLD/(MLS2)(DPLL-6b3) (for SMC=1)

The term (1 – PDC_S + PDC_S – PDC_S) of equation (DPLL_11024)[XREF TARGET " (DPLL_11024)] NOT EXIST is representing the third order Taylor series of the term 1/(1+PDC_S), which is chosen to reduce the additional time delay due to the more complex computation.

(3) Equation DPLL-6c to calculate RDT_S_ACT (nominal value)

RDT_S_ACT = 1 / DT_S_ACT(DPLL-6c)

(4) Equation DPLL-7a1 to calculate QDT_S_ACT

Relation of the last two increment values for APS=p in forward direction (DIR2=0)

QDT_S_ACT = DT_S_ACT * RDT_S[p–1](DPLL-7a1)

QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

NOTE

QDT_S_ACT uses a 6 bit integer part and an 18 bit fractional part.

(5) Equation DPLL-8 to calculate the error of last prediction

With q= NUSE–VSN when using QDT_S[p–q] and DT_S[p–1] for the prediction of DT_S[p]

EDT_S = DT_S_ACT – (DT_S[p–1] * QDT_S[p–q]) (DPLL-8) and with

QDT_S[p–q] = DT_S[p–q] * RDT_S[p–q–1] (for FSS=0) (DPLL-7b1)

QDT_S[p–q] = DT_S[p–q] * RDT_S_FS1 (for FSS=1) (DPLL-7b2)

FSS=1 has the meaning: NUSE=FULL_SCALE (see NUSC register). QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

NOTE

QDT_S[p–q] uses a 6 bit integer part and an 18 bit fractional part.

(6) Equation DPLL-9 to calculate the weighted average error

for SYS=1 calculate:

MEDT_S := (EDT_S + MEDT_S) / 2

(7) Equations DPLL-10 to calculate the current increment (nominal value)

nominal increment value (for ADS=0):

CDT_SX _nom = (DT_S_ACT + MEDT_S) * QDT_S[p–q+1] (DPLL-10a1)

or nominal increment value (for ADS=1):

CDT_SX _nom_corr = CDT_SX _nom * (1+CDC_S) (DPLL-10a2) with: CDC_S = PD / (MLS1) (DPLL-10a3) (for SMC=0)

CDC_S = PD / (MLS2) (DPLL-10a4) (for SMC=1) and

QDT_S[p–q+1] = DT_S[p–q+1] * RDT_S[p–q] (for q>1) (DPLL-7c) and for q=1 use equation (DPLL_11029)[XREF TARGET '' (DPLL_11029)] NOT EXIST. While QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

The CDT_SX_nom value is limited by the relation

CSN_MIN < CDT_SX_nom < CSN_MAX (DPLL-10c) When the calculated value exceeds one of the limits, it is replaced by the corresponding limit value.

The expected duration to the next STATE event is (for ADT=0): CDT_SX = CDT_SX _nom * SYN_T (for ADT=1): CDT_SX = CDT_SX _nom_corr * SYN_S (DPLL-10b)

NOTES

- QDT_S[p–q+1] uses a 6 bit integer part and an 18 bit fractional part.
- In the case of an overflow in equations (DPLL_11036)[XREF TARGET '' (DPLL_11036)] NOT EXIST or (DPLL_11045)[XREF TARGET '' (DPLL_11045)] NOT EXIST set the value to $FFFFFF_H$ and the corresponding CSO bit in the DPLL_STATUS register. In the case of negative values set CDT_SX to 0_H without any effect to the CSO bit. All 5 steps above (DPLL-6 to DPLL-10) are only needed in emergency mode. For the normal mode the calculations of equations DPLL-6 and DPLL-7 are done solely in order to get the values needed for a sudden switch to emergency mode.

38.22.6.4 Increment prediction in Normal Mode and for first PMSM backwards

(1) Equations DPLL-2a2 to calculate QDT_T_ACT backwards

QDT_T_ACT = DT_T_ACT * RDT_T[p+1] (DPLL-2a2)

QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

(2) Equation DPLL-3a to calculate of the error of last prediction

When q = NUTE–VTN and DIR1=1 using only QT_T[p+q] and DT_T[p+1] for the prediction of DT_T[p].

EDT_T = DT_T_ACT – (DT_T[p+1] * QDT_T[p+q] (DPLL-3a) with

QDT_T[p+q] = DT_T[p+q] * RDT_T[p+q+1] for FST=0 (DPLL-2b3)

QDT_T[p+q] = DT_T[p+q] * RDT_T_FS1 for FST=1 (DPLL-2b4)

FST=1 has the meaning: NUTE=FULL_SCALE (see NUTC register). QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

(3) Equation DPLL-4 to calculate the weighted average error

For SYT=1 calculate:
MEDT_T := (EDT_T + MEDT_T) / 2 (DPLL-4)

(4) Equation DPLL-5 to calculate the current increment value

nominal increment value (for ADT=0):
CDT_TX _nom = (DT_T_ACT + MEDT_T) * QDT_T[p+q–1](DPLL-5a5)

nominal increment value (for ADT=1):
CDT_TX _nom_corr = CDT_TX _nom * (1 + CDC_T) (DPLL-5a6)with

CDC_T = PD / (MLT + 1)(DPLL-5a3) (for SMC=0) see (DPLL_11008)[XREF TARGET " (DPLL_11008)] NOT EXIST

CDC_T = PD / (MLS1) (DPLL-5a4)(for SMC=1) see (DPLL_11009)[XREF TARGET " (DPLL_11009)] NOT EXIST and QDT_T[p+q–1] = DT_T[p+q–1] * RDT_T[p+q] (for q>1) (DPLL-2c1) for q=1 use **Section 38.22.6.2, (4) Equation DPLL-2a1 to calculate QDT_T_ACT**

while QDT_T_ACT as well as QDT_T[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part. The CDT_TX_nom value is limited by the relation

CTN_MIN < CDT_TX_nom < CTN_MAX (DPLL-5c) When the calculated value exceeds one of the limits, it is replaced by the corresponding limit value. The expected duration to the next TRIGGER event

CDT_TX = CDT_TX _nom * SYN_T (for ADT=0) see (DPLL_11005)[XREF TARGET " (DPLL_11005)] NOT EXIST

CDT_TX = CDT_TX _nom_corr * SYN_T (DPLL-5b) (for ADT=1) see (DPLL_11016)[XREF TARGET " (DPLL_11016)] NOT EXIST

NOTE

In the case of an overflow in equations (DPLL_11005)[XREF TARGET '' (DPLL_11005)] NOT EXIST or (DPLL_11016)[XREF TARGET '' (DPLL_11016)] NOT EXIST set the value to $FFFFFF_H$ and the corresponding CTO bit in the DPLL_STATUS register. In the case of negative values set CDT_TX(_nom) to 0_H.

38.22.6.5 Increment prediction in Emergency Mode and for second PMSM backwards

(1) Equation DPLL-7a2 to calculate QDT_S_ACT backwards

QDT_S_ACT = DT_S_ACT * RDT_S[p+1]

QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

(2) Equation DPLL-8a to calculate the error of the last prediction

While q= NUSE–VSN, use only QDT_S[p+q] and DT_S[p+1] for the prediction of DT_S[p]

EDT_S = DT_S_ACT – (DT_S[p+1] * QDT_S[p+q]) with

QDT_S[p–q] = DT_S[p+q] * RDT_S[p+q+1] (for FSS=0)

QDT_S[p–q] = DT_T[p+q] * RDT_S_FS1 (for FSS=1)

FSS=1 has the meaning: NUSE=FULL_SCALE (see NUSC register). QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part.

(3) Equation DPLL-9 to calculate the weighted average error

For SYS=1 calculate:

MEDT_S := (EDT_S + MEDT_S) / 2

(4) Equations DPLL-10 to calculate the current increment value

nominal increment value (for ADS=0):

CDT_SX _nom = (DT_S_ACT + MEDT_S) * QDT_S[p+q–1]

or nominal increment value (for ADS=1):

CDT_SX _nom_corr = CDT_SX _nom * (1+CDC_S) with:

CDC_S = PD / (MLS1) (for SMC=0) compare to (DPLL_11039)[XREF TARGET " (DPLL_11039)] NOT EXIST

CDC_S = PD / (MLS2) (for SMC=1) compare to (DPLL_11040)[XREF TARGET " (DPLL_11040)] NOT EXIST.

With QDT_S[p+q–1] = DT_S[p+q–1] * RDT_S[p+q] (for q>1) for q=1 use **Section 38.22.6.2, (4) Equation DPLL-2a1 to calculate QDT_T_ACT**.

While QDT_S_ACT as well as QDT_S[i] have a 24 bit value using a 6 bit integer part and an 18 bit fractional part. The CDT_SX_nom value is limited by the relation

CSN_MIN < CDT_SX_nom < CSN_MAX (compare to (DPLL_11044)[XREF TARGET " (DPLL_11044)] NOT EXIST) When the calculated value exceeds one of the limits, it is replaced by the corresponding limit value.

Then calculate the expected duration to the next STATE event CDT_SX = CDT_SX _nom * SYN_T (for ADT=0) CDT_SX = CDT_SX _nom_corr * SYN_S (for ADT=1) compare to (DPLL_11045)[XREF TARGET " (DPLL_11045)] NOT EXIST

NOTE

In the case of an overflow in equations (DPLL_11036)[XREF TARGET '' (DPLL_11036)] NOT EXIST, (DPLL_11045)[XREF TARGET '' (DPLL_11045)] NOT EXIST set the value to $FFFFFF_H$ and the corresponding CSO bit in the DPLL_STATUS register. In the case of negative values set CDT_SX(_nom) to 0x0. All 5 steps above (DPLL-6 to DPLL-10) are only needed in emergency mode. For the normal mode the calculations of equations DPLL-6 and DPLL-7 are done solely in order to get the values needed for a sudden switch to emergency mode.

38.22.7 Calculations for actions

As already shown for the calculation of the current interval by equations DPLL-1 to DPLL-10 for the prediction of actions a similar calculation is to be done, as shown by the equations DPLL-11. to DPLL-14. The calculation of actions is also needed when the DPLL is used for synchronous motor control applications (SMC=1, see DPLL_CTRL_1 register). For action prediction purposes the measured time periods of the past (one FULL_SCALE back, when the corresponding NUTE or NUSE values are set properly by the CPU) are used. The calculation can be explained by the following assumptions, which are considerably simple:

Take the corresponding increments for prediction in the past and put the sum of it in relation to the increment (DT_T[k], DT_S[k], with ≥ 0, which is represented by the time stamp difference) which is exactly one FULL_SCALE period in the past (DPLL-11 or DPLL-13 respectively). Make a prediction for the coming sum of increments using the current measured increment (DT_T_ACT or DT_S_ACT respectively, that means DPLL-1 or DPLL-6 respectively) and add a weighted average error (DPLL_1347 and DPLL_1554 or DPLL_1361 and DPLL_1362 respectively, calculated for one increment prediction) before multiplication with the relation of equation DPLL-11 or DPLL-13 respectively in order to get the result as described by equations DPLL_6935 or DPLL-14 respectively.

In order to avoid division operations instead of the increment (DT_T[k], DT_S[k], with k > 0) in the past its reciprocal value (RDT_T[k], RDT_S[k], with k > 0) is used, which is stored also in RAM. For the calculation of actions perform always a new refined calculation as long as the resulting time stamp is not in the past. In the other case the TSAC/PSAC values (time/position stamp of action calculated) is set to the time/position stamp of the last input event (TRIGGER/STATE), the ACT_N[i] bit in the DPLL_ACT_STA register is reset, while the corresponding ACT_N[i] bit in the DPLL_ACT_STA_shadow register is set. Each new PMTR_i value will set this ACT_N[i] bit again and reset the correspondent shadow bit until a new calculation is performed.

Make sure that the prediction parameters are chosen such that under all conditions (acceleration/ deceleration) the values of PDT_T, PDT_S and DTA respectively do not exceed the value $FFFFFF_H$. This requirement can limit the predicted position range in the case of very low speed.

Action updates at highest speed

Up to 32 action values can be calculated. For the shortest increment duration (23,4 µs) not all of them can be updated with each active input event. Notice the following conditions and parameters for an estimation of possible results. All time estimation values are given for a system clock frequency of 100 MHz and the assumption, that the calculation of the DPLL is not impeded by a remote read or write access to the DPLL RAMs. Each RAM access is to be considered by an additional delay of about 40 ns ($t_{remote\_RAM\_acces}$, to be described later in detail). When using a different system clock frequency the calculation duration is extended accordingly.

1. Typical time needed for basic operations (RAM update, pointer calculation and SUB_INC generation for normal, emergency mode or one PMSM:
   t_{basic_0} = 9,9 µs.
2. Typical time needed basic operations (RAM update, pointer calculation and SUB_INC generation for two PMSMs:
   t_{basic_1} = 11,0 µs.
3. Typical time needed to calculate one action
   t_{action_i} = 3,7 µs.

NOTE

The above mentioned values are observed worst case values, when the two state machines of TRIGGER and STATE are both in operation.

These values allow the calculation of at least 3 action values for each input event for all specified increments duration. The complete time needed for the basic operation, n action calculations and k remote RAM access operations can be calculated as follows

$t_{complete} = t_{basic\_0/1} + n*t_{action\_i} + k*t_{remote\_RAM\_access}$

Typical applications

Normal and emergency mode For a typical application with the shortest increment duration of 100 µs in normal or emergency mode the calculation of up to 24 action values can be performed for each active input event.

One PMSM For one PMSM and a typical shortest increment time of 39 µs there is the calculation of up to 7 action values possible for each input event.

Two PMSMs with restricted action calculations When only one PMSM uses the action calculation service and the shortest increment duration is 39 µs, there can up to 7 actions served for each active input event.

Two PMSMs with unrestricted action calculations When 2 PMSMs are used and both use the action calculation service at a minimal increment duration of 39 µs there are up to 7 action calculations possible for each of the two engines – that means up to 14 action calculations per increment in average.

38.22.7.1 Action calculations for TRIGGER forwards

valid for RMO=0 or for SMC=1 with p=APT_2B, t=APT, m=NA[i] (part w), mb=NA[i](part b)/1024, NUTE–VTN=q, NUTE=n

NOTE

All 5 steps in equations **Section 38.22.7.1, (1) Equation DPLL-11a1 to calculate the time prediction for an action** (DPLL-11) to (DPLL_6935)[XREF TARGET '' (DPLL_6935)] NOT EXIST are only calculated in normal mode.

(1) Equation DPLL-11a1 to calculate the time prediction for an action

For DIR1=0 and q>m calculate:

PDT_T[i] = (TSF_T[p+m–n] – TSF_T[p–n] + mb* DT_Tx[t–q+1]) * RDT_T[t–q] (DPLL-11a1)with

DT_Tx[t–q+1] = DT_T[t–q+1] for TS0_HRT=0 (DPLL-11b2) compare to (DPLL_11086)[XREF TARGET " (DPLL_11086)] NOT EXIST, (DPLL_11089)[XREF TARGET " (DPLL_11089)] NOT EXIST or DT_Tx[t–q+1] = DT_T[t–q+1]/8 (for TS0_HRT=1) (DPLL-11b3) compare to (DPLL_11087)[XREF TARGET " (DPLL_11087)] NOT EXIST, (DPLL_11090)[XREF TARGET " (DPLL_11090)] NOT EXIST and while the multiplication with mb does mean the fractional part of NA[i].

For SMC=0 and RMO=0 calculate for DIR1=0 all 32 actions in forward direction, if requested; in the case SMC=1 calculate up to 16 actions 0 to 15 in dependence of the TRIGGER input.

(2) Equation DPLL-11a2 to calculate the time prediction for an action

For SYT=1, NUTE= 2*(TNU+1), q>m and DIR1=0 calculate:

PDT_T[i] = (TSF_T[p+m] – TSF_T[p]+ mb* DT_Tx[t–q+1]) * RDT_T[t] (DPLL-11a2) with

DT_Tx[t–q+1] = DT_T[t–q+1] for TS0_HRT=0 (DPLL-11b2)compare to (DPLL_11083)[XREF TARGET " (DPLL_11083)] NOT EXIST, (DPLL_11089)[XREF TARGET " (DPLL_11089)] NOT EXIST or

DT_Tx[t–q+1] = DT_T[t–q+1]/8 (for TS0_HRT=1) (DPLL-11b3)compare to (DPLL_11084)[XREF TARGET " (DPLL_11084)] NOT EXIST, (DPLL_11090)[XREF TARGET " (DPLL_11090)] NOT EXIST

(3) Equation DPLL-11b to calculate the time prediction for an action

for DIR1=0, NUTE–VTN=q, q (< or =) m, n>1 and t=APT:

PDT_T[i] = (m+mb)*DT_Tx[t–q+1] * RDT_T[t–q](DPLL-11b) with

DT_Tx[t–q+1] = DT_T[t–q+1] for TS0_HRT=0 (DPLL-11b2) compare to (DPLL_11086)[XREF TARGET " (DPLL_11086)] NOT EXIST, (DPLL_11083)[XREF TARGET " (DPLL_11083)] NOT EXIST or

DT_Tx[t–q+1] = DT_T[t–q+1]/8 (for TS0_HRT=1)(DPLL-11b3) compare to (DPLL_11084)[XREF TARGET " (DPLL_11084)] NOT EXIST, (DPLL_11087)[XREF TARGET " (DPLL_11087)] NOT EXIST

NOTE

Make the calculations above before updating the TSF_T[i] values according to equations DPLL-1c3 ff.

(4) Equation DPLL-11c to calculate the time prediction for an action

for n=1 (this is always valid for SYT=0)

PDT_T[i] = (m+mb)* DT_T_ax* RDT_T[t–1] (DPLL-11c)with

DT_T_ax = DT_T_ACT for TS0_HRT=0 (DPLL-1a4a) compare to (DPLL_11095)[XREF TARGET " (DPLL_11095)] NOT EXIST, (DPLL_11097)[XREF TARGET " (DPLL_11097)] NOT EXIST or

DT_T_ax = DT_T_ACT/8 (for TS0_HRT=1) (DPLL-1a4b) compare to (DPLL_11098)[XREF TARGET " (DPLL_11098)] NOT EXIST, (DPLL_11102)[XREF TARGET " (DPLL_11102)] NOT EXIST

NOTE

For the relevant last increment add the fractional part of DT_T_ACT as described in NA[i].

(5) Equation DPLL-12 to calculate the duration value until action

DTA[i] = (DT_T_ACT + MEDT_T)* PDT_T[i] (DPLL-12) compare to (DPLL_6941)[XREF TARGET " (DPLL_6941)] NOT EXIST

38.22.7.2 Action calculations for TRIGGER backwards

Valid for RMO=0 or for SMC=1 with p=APT_2B, t=APT, m=NA[i] (part w), mb=NA[i](part b)/1024, q= NUTE–VTN and n=NUTE

For SMC=0 and RMO=0 calculate for DIR1=1 all 32 actions in backward direction for special purposes; in the case SMC=1 calculate up to 16 actions 0 to 15 in dependence of the TRIGGER input.

NOTE

All 5 steps in equations DPLL-11 to (DPLL_6935)[XREF TARGET '' (DPLL_6935)] NOT EXIST are only calculated in normal mode or when SMC=1.

(1) Equation DPLL-11a3 to calculate the time prediction for an action

For DIR1=1 and q>m calculate:

PDT_T[i] = (TSF_T[p–m+n] – TSF_T[p+n] + mb* DT_Tx[t+q–1]) * RDT_T[t+q] (DPLL-11a3) with DT_Tx[t+q–1] = DT_T[t+q–1] for TS0_HRT=0 (DPLL-11b4) or

DT_Tx[t+q–1] = DT_T[t+q–1]/8 for TS0_HRT=1 (DPLL-11b5)

(2) Equation DPLL-11a4 to calculate the time prediction for an action

For SYT=1 and NUTE = 2*(TNU+1), q>m, VTN=2*SYN_NT and hence NUTE–VTN = 2*(TNU+1–SYN_NT) for DIR1=1 PDT_T[i] is:

PDT_T[i] = (TSF_T[p–m] – TSF_T[p]+ mb* DT_Tx[t+q–1]) * RDT_T[t] (DPLL-11a4) with
DT_Tx[t+q–1] = DT_T[t+q–1] for TS0_HRT=0 (DPLL-11b4) or
DT_Tx[t+q–1] = DT_T[t+q–1]/8 for TS0_HRT=1 (DPLL-11b5)

NOTE

Make the calculations above before updating the TSF_T[i] values according to equations DPLL-1c3 ff.

(3) Equation DPLL-11b1 to calculate the time prediction for an action

For NUTE–VTN =q, q (< or =) m the following equation is valid for n>1 and t=APT:

PDT_T[i] = (m+mb)*DT_Tx[t+q–1] * RDT_T[t+q] (DPLL-11b1) with

DT_Tx[t+q–1] = DT_T[t+q–1] for TS0_HRT=0 (DPLL-11b4) or
DT_Tx[t+q–1] = DT_T[t+q–1]/8 for TS0_HRT=1 (DPLL-11b5)

(4) Equation DPLL-11c1 to calculate the time prediction for an action

for n=1 (this is always valid for SYT=0)

PDT_T[i] = (m+mb)* DT_T_ax * RDT_T[t+1] (DPLL-11c1) with

DT_T_ax = DT_T_ACT for TS0_HRT=0 compare to (DPLL_11095)[XREF TARGET '' (DPLL_11095)] NOT EXIST, (DPLL_11092)[XREF TARGET '' (DPLL_11092)] NOT EXIST or

DT_T_ax = DT_T_ACT/8 (for TS0_HRT=1) compare to (DPLL_11093)[XREF TARGET '' (DPLL_11093)] NOT EXIST, (DPLL_11102)[XREF TARGET '' (DPLL_11102)] NOT EXIST

NOTE

For the relevant last increment add the fractional part of DT_T_ACT as described in NA[i].

(5) Equation DPLL-12 to calculate the duration value for an action

DTA[i] = (DT_T_ACT + MEDT_T)* PDT_T[i] compare to (DPLL_6935)[XREF TARGET " (DPLL_6935)] NOT EXIST

Use the results of equations **Section 38.22.6.2, (1) Equations DPLL-1a to calculate TRIGGER time stamps**, **Section 38.22.6.2, (2) Equation DPLL-1b to calculate DT_T_ACT (nominal value)**, **Section 38.22.6.2, (5) Equation DPLL-3 to calculate the error of last prediction** and **Section 38.22.6.2, (6) Equation DPLL-4 to calculate the weighted average error** for the above calculation.

38.22.7.3 Action calculations for STATE forwards

valid for RMO=1 with p=APS_1C2, t=APS, m=NA[i](part w) mb=NA[i](part b)/1024, NUSE–VSN = q and NUSE=n>m

For SMC=0 and RMO=1 calculate for DIR2=0 all 32 actions in forward direction, if requested; in the case SMC=1 and RMO=1 calculate up to 16 actions 16 to 31 in dependence of the STATE input.

NOTE

All 5 steps of equations DPLL-13 to DPLL-14 are only calculated in emergency mode or for SMC=1 in combination with RMO=1.

(1) Equation DPLL-13a1 to calculate the time prediction for an action

For DIR2=0 and q>m calculate:

PDT_S[i] = (TSF_S[p+m–n] – TSF_S[p–n] + mb* DT_Sx[t–q+1] * RDT_S[t–q] (DPLL-13a1) with DT_Sx[t–q+1]= DT_S[t–q+1] for TS0_HRS=0 (DPLL-13b2) or

DT_Sx[t–q+1] = DT_S[t–q+1]/8 for TS0_HRS=1 (DPLL-13b3)

(2) Equation DPLL-13a2 to calculate the time prediction for an action

For SYS=1 and NUSE=2*(SNU+1), q>m, SYSF=0, VSN=2*SYN_NS and hence NUSE–VSN = 2*(SNU+1–SYN_NS) PDT_S[i] is equal to:

PDT_S[i] = (TSF_S[p+m] – TSF_S[p]+ mb*DT_Sx[t–q+1]) * RDT_S[t] (DPLL-13a2) with

DT_Sx[t–q+1] = DT_S[t–q+1] for TS0_HRS=0 (DPLL-13b2) or
DT_Sx[t–q+1] = DT_S[t–q+1]/8 for TS0_HRS=1 (DPLL-13b3)

(3) Equation DPLL-13b to calculate the time prediction for an action

For NUSE –VTN=q, q (< or =) m and n>1:

PDT_S[i] = (m+mb)*DT_Sx[t–q+1] * RDT_S[t–q] (DPLL-13b) with
DT_Sx[t–q+1] = DT_S[t–q+1] for TS0_HRS=0 (DPLL-13b2) or
DT_Sx[t–q+1] = DT_S[t–q+1]/8 for TS0_HRS=1 (DPLL-13b3)

(4) Equation DPLL-13c to calculate the time prediction for an action

for n=1

PDT_S[i] = (m+mb)* DT_S_ax * RDT_S[t–1] (DPLL-13c) with
DT_S_ax = DT_S_ACT for TS0_HRS=0 (DPLL-6a4a) or
DT_S_ax = DT_S_ACT/8 for TS0_HRS=1 (DPLL-6a4b)

(5) Equation DPLL-14 to calculate the duration value for an action

DTA[i] = (DT_S_ACT + MEDT_S)* PDT_S[i] (DPLL-14)

Use the results of DPLL-7, **Section 38.22.6.3, (5) Equation DPLL-8 to calculate the error of last prediction** and **Section 38.22.6.3, (6) Equation DPLL-9 to calculate the weighted average error** for the above calculation

38.22.7.4 Action calculations for STATE backwards

valid for RMO=1 with p=APS_1C2, t=APS, m=NA[i](part w) mb=NA[i](part b)/1024, NUSE–VSN = q and NUSE=n

For SMC=0 and RMO=1 calculate for DIR1=1 all 32 actions in backwards mode for special purposes; in the case SMC=1 and RMO=1 calculate up to 16 actions 16 to 31 in dependence of the STATE input.

NOTE

All 5 steps of equations DPLL-13 to DPLL-14 are only calculated in emergency mode or for SMC=1 in combination with RMO=1.

(1) Equation DPLL-13a3 to calculate the time prediction for an action

For (DIR2= 1 (SMC=1) or DIR1=1 (SMC=0)) and q>m calculate

PDT_S[i] = (TSF_S[p–m+n] – TSF_S[p+n] + mb* DT_Sx[t+q–1]) * RDT_S[t+q] (DPLL-13a3) with
DT_Sx[t+q–1]= DT_S[t+q–1] for TS0_HRS=0 (DPLL-13b4) or
DT_Sx[t+q–1] = DT_S[t+q–1]/8 for TS0_HRS=1 (DPLL-13b5)

(2) Equation DPLL-13a4 to calculate the time prediction for an action

For SYS=1, NUSE=2*(SNU+1), q>m, SYSF=0, VSN=2*SYN_NS and hence NUSE–VSN = 2*(SNU+1–SYN_NS) calculate:

PDT_S[i] = (TSF_S[p–m] – TSF_S[p]+ mb* DT_Sx[t+q–1]) * RDT_S[t] (DPLL-13a4) with
DT_Sx[t+q–1]= DT_S[t+q–1] for TS0_HRS=0 (DPLL-13b4) or

DT_Sx[t+q–1] = DT_S[t+q–1]/8 for TS0_HRS=1 (DPLL-13b5)

(3) Equation DPLL-13b1 to calculate the time prediction for an action

For NUSE–VSN =q, q (< or =) m, NUSE=n and n>1:

PDT_S[i] = m*DT_Sx[t+q–1] * RDT_S[t+q] (DPLL-13b1) with DT_Sx[t+q–1] = DT_S[t+q–1] for TS0_HRS=0 (DPLL-13b4) or DT_Sx[t+q–1] = DT_S[t+q–1]/8 for TS0_HRS=1 (DPLL-13b5)

(4) Equation DPLL-13c1 to calculate the time prediction for an action

for n=1:

PDT_S[i] = (m+mb)* DT_S_ax * RDT_S[t+1] (DPLL-13c1) with
DT_S_ax = DT_S_ACT (for TS0_HRS=0) (DPLL-6a4a) or
DT_S_ax = DT_S_ACT/8 (for TS0_HRS=1) (DPLL-6a4b)

(5) Equation DPLL-14 to calculate the duration value until action

DTA[i] = (DT_S_ACT + MEDT_S)* PDT_S[i] (DPLL-14)

Use the results of DPLL-7, **Section 38.22.6.3, (5) Equation DPLL-8 to calculate the error of last prediction** and **(6), Equation DPLL-9 to calculate the weighted average error** or (DPLL_6929)[XREF TARGET " (DPLL_6929)] NOT EXIST for the above calculation.

38.22.7.5 Update of RAM in Normal and Emergency Mode

After considering the calculations for up to all 24 actions according to equations (DPLL-11, (DPLL_6935)[XREF TARGET " (DPLL_6935)] NOT EXIST), only when going back to state 1 or 21 (because of a new TRIGGER or STATE event, that means when no further PMTR values are to be considered) set time stamp values and duration of increments in the RAM.

(1) Equation DPLL-1a4 to update the time stamp values for TRIGGER

TSF_T[s]=TS_Tx (DPLL-1a4) using the following equations for the determination of TS_Tx

For TS0_HRT=0:

TS_Tx=TS_T (DPLL-1a4w)

DT_T_ax = DT_T_ACT compare to (DPLL_11092)[XREF TARGET " (DPLL_11092)] NOT EXIST, (DPLL_11097)[XREF TARGET " (DPLL_11097)] NOT EXIST

For TS0_HRT=1:

TS_Tx(20:0)=TS_T/8 (DPLL-1a4x)

TS_Tx(23:21)=TBU_TS0_T(23:21) (DPLL-1a4y) for TBU_TS0_T(20:0) > or = TS_Tx(20:0)
TS_Tx(23:21)=TBU_TS0_T(23:21) –1 (DPLL-1a4z) for TBU_TS0_T(20:0) < TS_Tx(20:0)

DT_T_ax = DT_T_ACT/8 compare to (DPLL_11093)[XREF TARGET " (DPLL_11093)] NOT EXIST, (DPLL_11098)[XREF TARGET " (DPLL_11098)] NOT EXIST

NOTE

The combination of values LOW_RES=0 and TS0_HRT=1 is not possible.

Store the time stamp values in the time stamp field according to the address pointer APT_2B=s, but make this update only after the calculation of actions **Section 38.22.7, Calculations for actions** because the old TSF_T[i] values are still needed for these calculations. Note that the address pointer after a gap is still incremented by SYN_T_OLD in that case (see state machine step 1 in **Section 38.22.8.6, Scheduling of the Calculation**).

(2) Equation DPLL-1a5-7 to extend the time stamp values for TRIGGER in forward direction

when SYT=1 and SYN_T_OLD=r>1 and DIR1=0:
TSF_T[s–1] = TSF_T[s] –DT_T_ax (DPLL-1a5)

TSF_T[s–2] = TSF_T[s–1] –DT_T_ax (DPLL-1a6) until

TSF_T[s– r+1] = TSF_T[s– r+2] –DT_T_ax (DPLL-1a7) after the incrementation of the pointer APT_2B by SYN_T_OLD

(3) Equations DPLL-1a5-7 for backward direction

When SYT=1 and SYN_T_OLD=r>1 and DIR1=1

TSF_T[s+1] = TSF_T[s] –DT_T_ax (DPLL-1a5)

TSF_T[s+2] = TSF_T[s+1] –DT_T_ax (DPLL-1a6) until

TSF_T[s+r–1] = TSF_T[s+r–2] –DT_T_ax (DPLL-1a7) after the decrementation of the pointer APT_2B by SYN_T_OLD

(4) Equations DPLL-1b1 and DPLL-1c1 to update the RAM after calculation

DT_T[p] = DT_T_ACT (DPLL-1b1)

save old reciprocal value from RAM before overwriting:

RDT_T_FS1= RDT_T[p] (DPLL-1c1)

after that store new value in RAM

RDT_T[p]= RDT_T_ACT (DPLL-1c2)

Store increment duration and reciprocal value in RAM region 2 in normal mode after calculation of actions only when a new active TRIGG- ER slope is detected and in emergency mode directly after calculation of DT_T_ACT or RDT_T_ACT respectively.

(5) Equation DPLL-6a4 to update the time stamp values for STATE

TSF_S[s]=TS_Sx (DPLL-6a4) using the following equations for the determination of TS_Sx

For TS0_HRS=0:

TS_Sx=TS_S (DPLL-6a4)

DT_S_ax = DT_S_ACT (DPLL-6a4a)

For TS0_HRS=1: TS_Sx(20:0)=TS_S/8 (DPLL-6a4x)

TS_Sx(23:21)=TBU_TS0_S(23:21) (DPLL-6a4y) for TBU_TS0_S(20:0) > or = TS_Sx(20:0)
TS_Sx(23:21)=TBU_TS0_S(23:21) –1 (DPLL-6a4z) for TBU_TS0_S(20:0) < TS_Sx(20:0)

DT_S_ax = DT_S_ACT/8 (DPLL-6a4b)

NOTE

The combination of values LOW_RES=0 and TS0_HRS=1 is not possible.

Store the time stamp value in the time stamp field according to the address pointer APS_1C2=s, but make this update only after the calculation of actions (equations DPLL-13a2, **Section 38.22.7.3, (2) Equation DPLL-13a2 to calculate the time prediction for an action** or DPLL-13a4 **Section 38.22.7.4, (2) Equation DPLL-13a4 to calculate the time prediction for an action**, if applicable) because the old TSF_S[i] values are still needed for these calculations. Note that the address pointer after a gap is still incremented by SYN_S_OLD in that case (see state machine step 21 in **Section 38.22.8.6, Scheduling of the Calculation**).

(6) Equations DPLL-6a5-7 to extend the time stamp values for STATE

When SYS=1 and SYN_S_OLD=r>1 and DIR2=0 or DIR1=0 respectively calculate

TSF_S[s–1] = TSF_S[s] – DT_S_ax (DPLL-6a5)

TSF_S[s–2] = TSF_S[s–1] – DT_S_ax (DPLL-6a6) until TSF_S[s– r+1] = TSF_S[s– r+2] – DT_S_ax (DPLL-6a7)

after incrementation of the pointer APS_2b by SYN_S_OLD

(7) Equations DPLL-6a5-7 for backward direction

When SYS=1 and SYN_S_OLD=r>1 and DIR2=1 or DIR1=1 respectively calculate TSF_S[s+1] = TSF_S[s] – DT_S_ax (DPLL-6a5)

TSF_S[s+2] = TSF_S[s+1] – DT_S_ax (DPLL-6a6) until

TSF_S[s+r–1] = TSF_S[s+r–2] – DT_S_ax (DPLL-6a7)

after the incrementation of the pointer APS_1C2 by SYN_S_OLD

(8) Equations DPLL-6b1 and DPLL-6c2 to update the RAM after calculation

DT_S[p] = DT_S_ACT (DPLL-6b1)

save old reciprocal value from RAM before overwriting:

RDT_S_FS1= RDT_S[p] (DPLL-6c1)

after that store new value in RAM

RDT_S[p] = RDT_S_ACT (DPLL-6c2)

when a new active STATE slope is detected in emergency mode or in normal mode (SMC=RMO=0) directly after calculation of the values above.

Store increment duration and reciprocal value in RAM region 1c in emergency mode after calculation of actions only when a new active STATE slope is detected and in normal mode directly after calculation of DT_S_ACT or RDT_S_ACT respectively.

38.22.7.6 Time and position stamps for actions in Normal Mode

(1) Equation DPLL-15 to calculate the action time stamp

TSAC[i]= DTA[i] – DLA[i] + TS_Tx (for DTA[i] > DLA[i] and DTA[i] – DLA[i] < 800000_H) (DPLL-15a)

TSAC[i]= TS_Tx (for DTA[i] < DLA[i]) (DPLL-15b)

TSAC[i]= $7FFFFF_H$ + TS_Tx (for DTA[i] > DLA[i] and DTA[i] – DLA[i] > $0x7FFFFF_H$) (DPLL-15c)

NOTE

For TS_Tx see equations (DPLL-1a4 and following), **Section 38.22.7.5, (1) Equation DPLL-1a4 to update the time stamp values for TRIGGER**.

The calculation is done after the calculation of the current expected duration value according to equation (DPLL_6941)[XREF TARGET " (DPLL_6941)] NOT EXIST at **Section 38.22.7.2, (5) Equation DPLL-12 to calculate the duration value for an action**. The time stamp of the action can be calculated as shown above in equation DPLL-15 using the delay value of the action and the current time stamp.

(2) Equations DPLL-17 to calculate the position stamp forwards

for DIR1=0 and TS0_HRT=0:

PSAC[i] = PSA[i] – (DLA[i]*RCDT_TX_NOM)*(MLT+1) (DPLL-17) with

RCDT_TX_NOM= (1/CDT_TX_NOM) * SYN_T (DPLL-17a) and
RCDT_TX= 1/CDT_TX (DPLL-17b)

for DIR1=0 and TS0_HRT=1:

PSAC[i] = PSA[i] – (8*DLA[i]*RCDT_TX_NOM)*(MLT+1) (DPLL-17d) with
RCDT_TX_NOM= (1/CDT_TX_NOM) * SYN_T (DPLL-17a) and

RCDT_TX= 1/CDT_TX (DPLL-17b)

replace (MLT+1) in equations (DPLL-17) and (DPLL-17d) by MLS1 for SMC=1

use the calculated value of (DPLL-17b) also for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i. The action is to be updated for each new TRIGGER event until the calculated time stamp is in the past. In this case the values of TSAC[i] and PSAC[i] depend on the DPLL_CTRL_11.ACBU signal. When DPLL_CTRL_11.ACBU = '0': Use the time stamp of the last input event instead of the calculated value and the calculated position stamp of the actual increment as target position value. When DPLL_CTRL_11.ACBU = '1': For ACB_[z][1]= '1': is used as input signal to control if "action in past" shall be checked based on position information. If the position has reached "past" use the calculated position stamp of the actual increment as target position value. For ACB_[z][1]= '0': In this case the PSAC[i] is used as calculated by the DPLL. For ACB_[z][0]= '1': is used as input signal to control if "action in past" shall be checked based on time information. If the time has reached "past" use the time stamp of the last input event instead of the calculated TSAC[i] value. For ACB_[z][0]= '0': In this case the TSAC[i] is used as calculated by the DPLL. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation the ACT_D values can be read only once.

(3) Equations DPLL-17 to calculate the position stamp backwards

For DIR1=1 and TS0_HRT=0:

PSAC[i] = PSA[i] + (DLA[i]*RCDT_TX_NOM)*(MLT+1) (DPLL-17c) with

RCDT_TX_NOM= (1/CDT_TX_NOM) * SYN_T (DPLL-17a) and

RCDT_TX= 1/CDT_TX (DPLL-17b)

For DIR1=1 and TS0_HRT=1:

PSAC[i] = PSA[i] + (8*DLA[i]*RCDT_TX_NOM)*(MLT+1) (DPLL-17e) with
RCDT_TX_NOM= (1/CDT_TX_NOM) * SYN_T (DPLL-17a) and

RCDT_TX= 1/CDT_TX (DPLL-17b)

replace (MLT+1) in equations (DPLL-17c) and (DPLL-17e) by MLS1 for SMC=1

use the calculated value of (DPLL-17b) also for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i. The action is to be updated for each new TRIGGER event until the calculated time stamp is in the past. In this case the values of TSAC[i] and PSAC[i] depend on the DPLL_CTRL_11.ACBU signal. When DPLL_CTRL_11.ACBU = '0': Use the time stamp of the last input event instead of the calculated value and the calculated position stamp of the actual increment as target position value. When DPLL_CTRL_11.ACBU

= '1': For ACB_[z][1]= '1': is used as input signal to control if "action in past" shall be checked based on position information. If the position has reached "past" use the calculated position stamp of the actual increment as target position value. For ACB_[z][1]= '0': In this case the PSAC[i] is used as calculated by the DPLL. For ACB_[z][0]= '1': is used as input signal to control if "action in past" shall be checked based on time information. If the time has reached "past" use the time stamp of the last input event instead of the calculated TSAC[i] value. For ACB_[z][0]= '0': In this case the TSAC[i] is used as calculated by the DPLL. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation the ACT_D values can be read only once.

38.22.7.7 The use of the RAM

The RAM is used to store the data of the last FULL_SCALE period. The use of single port RAMs is recommended. The data width of the RAM is usual 3 bytes, but could be extended to 4 bytes in future applications. There are 3 different RAMs, each with separate access ports. The RAM 1a is used to store the position minus time requests, got from the ARU. No CPU access is possible to this RAM during operation (when the DPLL is enabled). Ram 1b is used for configuration parameters and variables needed for calculations. Within RAM 1c the values of the STATE events are stored. RAM 1b and RAM 1c do have a common access port and are also marked as RAM 1bc in order to clarify this fact. RAM 2 is used for values of the TRIGGER events. Because of the access of the DPLL internal state machine at the one side and the CPU at the other side the access priority has to be controlled for both RAMs 1bc and 2. The access priority is defined as stated below. The CPU access procedure via AE-interface goes in a wait state (waiting for data valid) while it needs a colliding RAM access during serving a corresponding state machine RAM access. In order not to provoke unexpected behavior of the algorithms the writing of the CPU to the RAM regions 1b, 1c or 2 will be monitored and results in interrupt requests when enabled.

CPU access is specified at follows:

1. CPU has highest priority for a single read/write access. The DPLL algorithm is stalled during external bus RAM accesses.
2. After serving the CPU access to the RAM the DPLL gets the highest RAM access priority for 8 clock cycles. Afterwards continue with 1.

The RAM address space has to be implemented in the address space of the CPU.

38.22.7.8 Time and position stamps for actions in Emergency Mode

(1) Equation DPLL-18 to calculate the action time stamp

TSAC[i]= DTA[i] – DLA[i] + TS_Sx (for DTA[i] > DLA[i] and DTA[i] – DLA[i] < 0x800000) (DPLL-18a) TSAC[i]= TS_Sx (for DTA[i] < DLA[i]) (DPLL-18b)

TSAC[i]= $FFFFF_H$ + TS_Sx (for DTA[i] > DLA[i] and DTA[i] – DLA[i] > $7FFFFF_H$) (DPLL-18c)

NOTE

For TS_Sx see equations (DPLL-6a4 and following), **Section 38.22.7.5, (5) Equation DPLL-6a4 to update the time stamp values for STATE**.

The calculation is done after the calculation of the current expected duration value according to equation DPLL-14 at **Section 38.22.7.3, (5) Equation DPLL-14 to calculate the duration value for an action**. The time stamp of the action can be calculated as shown in equation DPLL-18 using the delay value of the action and the current time stamp.

(2) Equations DPLL-20 to calculate the position stamp forwards

for DIR2=0 or DIR1=0 respectively and TS0_HRS=0: PSAC[i] = PSA[i] – (DLA[i]*RCDT_SX_NOM)*MLS1 (DPLL-20) with
RCDT_SX_NOM= (1/CDT_SX_NOM) * SYN_S (DPLL-20a) and
RCDT_SX= 1/CDT_SX (DPLL-20b)

for DIR2=0 or DIR1=0 respectively and TS0_HRS=1:

PSAC[i] = PSA[i] – (8*DLA[i]*RCDT_SX_NOM)*MLS1 (DPLL-20d) with
RCDT_SX_NOM= (1/CDT_SX_NOM) * SYN_S (DPLL-20a) and
RCDT_SX= 1/CDT_SX (DPLL-20b)

replace MLS1 in equations (DPLL-20) and (DPLL-20d) by MLS2 for (SMC=1 and RMO=1)

use the calculated value of (DPLL-17b) also for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i. The action is to be updated for each new *STATE* event until the calculated time stamp is in the past. In this case the values of TSAC[i] and PSAC[i] depend on the DPLL_CTRL_11.ACBU signal. When DPLL_CTRL_11.ACBU = '0': Use the time stamp of the last input event instead of the calculated value and the calculated position stamp of the actual increment as target position value. When DPLL_CTRL_11.ACBU = '1': For ACB_[z][1]= '1': is used as input signal to control if "action in past" shall be checked based on position information. If the position has reached "past" use the calculated position stamp of the actual increment as target position value. For ACB_[z][1]= '0': In this case the PSAC[i] is used as calculated by the DPLL. For ACB_[z][0]= '1': is used as input signal to control if "action in past" shall be checked based on time information. If the time has reached "past" use the time stamp of the last input event instead of the calculated TSAC[i] value. For ACB_[z][0]= '0': In this case the TSAC[i] is used as calculated by the DPLL. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation the ACT_D values can be read only once.

(3) Equations DPLL-20 to calculate the position stamp backwards

For DIR2=1 or DIR1=1 respectively and TS0_HRS=0:

PSAC[i] = PSA[i] + (DLA[i]*RCDT_SX_NOM)*MLS1 (DPLL-20c) with RCDT_SX_NOM= (1/CDT_SX_NOM) * SYN_S (DPLL-20a) and RCDT_SX= 1/CDT_SX (DPLL-20b)

For DIR2=1 or DIR1=1 respectively and TS0_HRS=1:

PSAC[i] = PSA[i] + (8*DLA[i]*RCDT_SX_NOM)*MLS1 (DPLL-20e) with RCDT_SX_NOM= (1/CDT_SX_NOM) * SYN_S (DPLL-20a) and RCDT_SX= 1/CDT_SX (DPLL-20b)

replace MLS1 in equations (DPLL-20c) and (DPLL-20e) by MLS2 for (SMC=1 and RMO=1)

Use the calculated value of (DPLL-20b) also for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D. The action is to be updated for each new STATE event until the event is in the past. In this case use the time stamp of the last input event instead of the calculated value and the calculated position stamp of the actual increment as target position value. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation the ACT_D values can be read only once.

Use the calculated value of (DPLL-17b) also for the generation of SUB_INCi and serve the action by transmission of TSAC[i] and PSAC[i] to ACT_D_i. The action is to be updated for each new STATE event until the calculated time stamp is in the past. In this case the values of TSAC[i] and PSAC[i] depend on the DPLL_CTRL_11.ACBU signal. When DPLL_CTRL_11.ACBU = '0': Use the time stamp of the last input event instead of the calculated value and the calculated position stamp of the actual increment as target position value. When DPLL_CTRL_11.ACBU = '1': For ACB_[z][1]= '1': is used as input signal to control if "action in past" shall be checked based on position information. If the position has reached "past" use the calculated position stamp of the actual increment as target position value. For ACB_[z][1]= '0': In this case the PSAC[i] is used as calculated by the DPLL. For ACB_[z][0]= '1': is used as input signal to control if "action in past" shall be checked based on time information. If the time has reached "past" use the time stamp of the last input event instead of the calculated TSAC[i] value. For ACB_[z][0]= '0': In this case the TSAC[i] is used as calculated by the DPLL. Set the corresponding shadow bit in the DPLL_ACT_STA register. Because of the blocking read operation the ACT_D values can be read only once.

38.22.8 Signal processing

38.22.8.1 Time stamp processing

Signal processing does mean the computation of the time stamps in order to calculate at which time the outputs have to appear. For such purposes the time stamp values have to be stored in the RAM and by calculating the difference between old and new values the duration of the last time interval is determined simply. This difference should be also stored in the RAM in order to see the changes between the intervals by changing the conditions and the speed of the observed process.

38.22.8.2 Count and compare unit

The count and compare unit processes all input signals taking into account the configuration values. It uses a state machine and provides the output signals as described above.

38.22.8.3 Sub pulse generation for SMC=0

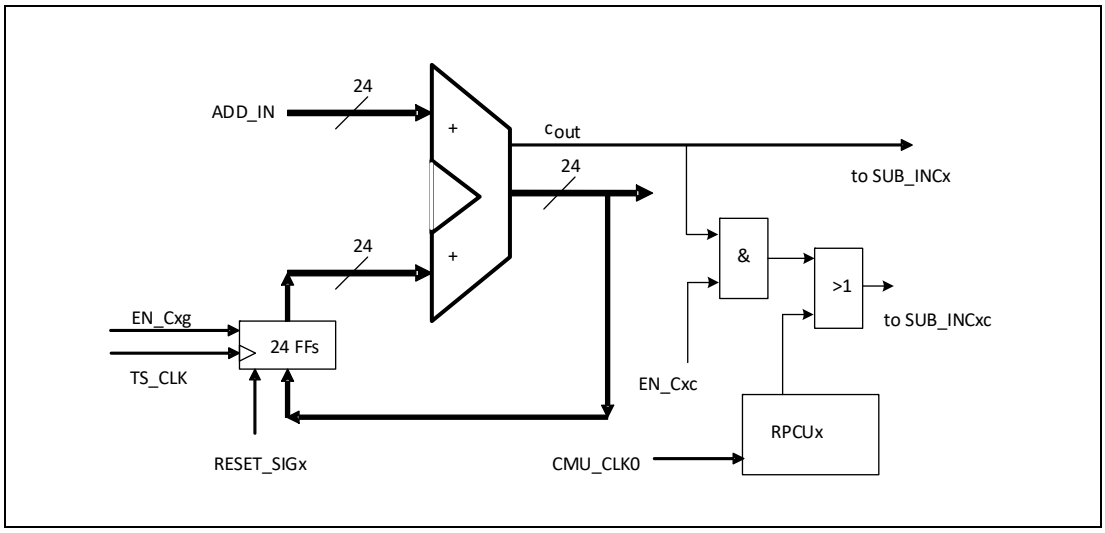


Figure 38.116 Adder for generation of SUB_INCx by the carry c_{out}

NOTE

The SUB_INC generation by the circuit above has the advantage, that the resolution for higher speed values is better as for a simple down counter. After RESET and after EN_Cxg=0 the flip-flops (FFs) should have a zero value. EN_Cxg has to be zero until reliable ADD_IN values are available and the pulse generation starts. This is controlled by the configuration bits SGE1,2 in the DPLL_CONTROL_1 register. The calculated values for the increment prediction using equations (DPLL_11010)[XREF TARGET " (DPLL_11010)] NOT EXIST in **Section 38.22.6.2, (7) Equations DPLL-5 to calculate the current increment value**, (DPLL_11057)[XREF TARGET " (DPLL_11057)] NOT EXIST in **Section 38.22.6.4, (4) Equation DPLL-5 to calculate the current increment value**, (DPLL_11041)[XREF TARGET " (DPLL_11041)] NOT EXIST in **Section 38.22.6.3, (7) Equations DPLL-10 to calculate the current increment (nominal value)** or (DPLL_11077)[XREF TARGET " (DPLL_11077)] NOT EXIST in **Section 38.22.6.5, (4) Equations DPLL-10 to calculate the current increment value** respectively are valid only when at least NUTE>1 TRIGGER values or at least NUSE>1 STATE values are available. For NUTE =1 or NUSE=1 respectively the equations DPLL-25 in **Section 38.22.8.3, (3) Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0** and DPLL-26 in **Section 38.22.8.3, (5) Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC = 0** use the actual increment value subtracted by the weighted average error.

The generation of SUB_INC1 pulses depends on the configuration of the DPLL. In automatic end mode the counter INC_CNT1 resets the enable signal EN_C1 when the number of pulses desired is reached. In this case only the uncompensated output SUB_INC1 remains active in order to provide pulses for the input filter unit. In the case of acceleration missing pulses can be determined at the next TRIGGER/STATE event in normal/emergency mode easily. For the correction strategy COA = 0 those missing pulses are sent out with CMU_CLK0 frequency as soon they are determined. During this time period the EN_Cxg remains cleared. After calculation or providing of a new ADD_IN value the FFs are enabled by EN_Cxg. In this way no pulse is lost. The new pulses are sent out afterwards, when INC_CNT1 is set to the desired value, maybe by adding MLT+1 or MLS1 respectively for the new TRIGGER/STATE event.

Because the used DIV procedure of the algorithms results only in integer values, a systematic failure could appear. The pulse generation at SUB_INC1 will stop in automatic end mode when the INC_CNT1 register reaches zero or all remaining pulses at a new increment will be considered in the next calculation. In this way the loss of pulses can be avoided. When a new TRIGGER/STATE appears the value of SYN_T*(MLT+1) or SYN_S*MLS1 respectively is added to INC_CNT1, when SGE1=1. Therefore for FULL_SCALE 2*(TNU+1)*(MLT+1) pulses SUB_INC1 generated, when INC_CNT1 reaches the zero value. The generation of SUB_INC1 pulses has to be done as fast as possible. The calculations for the ADD_IN value must be done first. Therefore all values needed for calculation are to be fetched in a forecast.

(1) Equation DPLL-21 to calculate the number of pulses to be sent in normal mode using the automatic end mode condition

For RMO=0, SMC=0 and DMO=0

NMB_T = ((MLT+1) + PD_store) *SYN_T + MP + MPVAL1 (DPLL-21) with

PD_store = ADT_T[12:0], prefetched during last increment

SYN_T = ADT_T[18:16], prefetched during last increment

MPVAL1 = pulse correction value for PCM1_SHADOW_TRIGGER=1

while the value for PD_store is zero for AMT=0 and the value of MP is zero for COA=0.

In order to get a higher resolution for higher speed a generator for the sub-pulses is chosen using an adder. All missing pulses MP are considered using equation DPLL-21 and are determined by counting the number of pulses of the last increment. The value SYN_T is stored from the last increment using NT of the ADT_T[i] value at RAM region 2c.

(2) Equations DPLL-22–24 to calculate the number of pulses to be sent in emergency mode using the automatic end mode condition for SMC = 0

For RMO=1, SMC=0 and DMO=0; the value for PD_S_store is zero for AMS=0

NMB_S = (MLS1 + PD_S_store) *SYN_S+ MP (DPLL-22) with

MLS1= (MLT+1) * (TNU+1) / (SNU+1) (DPLL-23)

PD_S_store = ADT_S[15:0], prefetched during last increment

SYN_S = ADT_S[21:16], prefetched during last increment MPVAL1 = pulse correction value for PCM1_SHADOW_STATE=1

while the value for PD_S_store is zero for AMS=0 and the value of MP is zero for COA=0

Note that these calculations above in equations DPLL-21 and DPLL-22 are only valid for an automatic end mode (DMO = 0). For calculation of the number of generated pulses a value of 0.5 is added as shown in equations DPLL-25 or DPLL-26 respectively in order to compensate rounding down errors at the succeeding arithmetic operations. Because in automatic end mode the number of pulses is limited by INC_CNT1 it is guaranteed, that not more pulses as needed are generated and in the same way missing pulses are caught up for the next increment.

(3) Equation DPLL-25 to calculate ADD_IN in normal mode for SMC = 0

In normal mode (for RMO=0) calculate in the case LOW_RES=TS0_HRT ADD_IN_CALN= (NMB_T+0.5) *RCDT_TX (DPLL-25) with

RCDT_TX is the 2 time value of the quotient in equation DPLL-17b **Section 38.22.7.6, (3) Equations DPLL-17 to calculate the position stamp backwards**.

In normal mode (for RMO=0) calculate in the case LOW_RES=1 and TS0_HRT=0

ADD_IN_CALN= (NMB_T+0.5) *(RCDT_TX /8) (DPLL-25a) with

RCDT_TX is the 2 time value of the quotient in equation DPLL-17b **Section 38.22.7.6, (3) Equations DPLL-17 to calculate the position stamp backwards**.

For RMO=0 and SMC=0:

ADD_IN_CAL1 = ADD_IN_CALN (DPLL-25b)

LOW_RES=0 and TS0_HRT=1 is not possible. For such a configuration the RCT bit in the DPLL_STATUS register is set together with the ERR bit.

In the automatic end mode (DMO=0) missing pulses should be sent to the input RPCUx (rapid pulse catch up on) in **Figure 38.116**, to be caught up on with CMU_CLK0 (for COA=0). When normal and rapid pulses are generated simultaneously, the SUB_INCx frequency is doubled at this moment in order to count two pulses at the TBU_CHx_BASE register. In order to make the frequency doubling possible, the CMU_CLK0 should be having a frequency which does not exceed half the frequency of TS_CLK. In addition the ADD_IN value should never exceed the value 800000_H. This limitation is only necessary for DMO=0 and COA=0 (see **Section 38.22.18.2, DPLL_CTRL_1**). For the normal mode replace ADD_IN of the ADDER (see **Figure 38.116**) by ADD_IN_CAL1 (when calculated,

DLM=0) or ADD_IN_LD1 (when provided by the CPU, DLM=1). The sub-pulse generation in this case is done by the following calculations using a 24 bit adder with a carry out c_{out} and the following inputs: – ADD_IN the second input is the output of the adder, stored one time stamp clock before.

In order not to complicate the calculation procedure use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

(4) Enabling of the compensated output for pulses

The c_{out} of the adder influences directly the SUB_INC1 output of the DPLL (see **Figure 38.116**). The compensated output SUB_INCxc is in automatic end mode only enabled by EN_Cxc when INC_CNTx >0.

(5) Equation DPLL-26 to calculate ADD_IN in emergency mode for SMC = 0

In emergency mode (RMO=1) calculate in the case LOW_RES=TS0_HRS ADD_IN_CALE= (NMB_S+0.5)* RCDT_SX (DPLL-26) while

RCDT_SX is the 2 time value of the quotient in equation DPLL-20b **Section 38.22.7.8, (2) Equations DPLL-20 to calculate the position stamp forwards**.

In emergency mode (RMO=1) calculate in the case LOW_RES=1 and TS0_HRS=0 ADD_IN_CALE= (NMB_S+0.5)* RCDT_SX /8 (DPLL-26a) while

RCDT_SX is the 2 time value of the quotient in equation DPLL-20b **Section 38.22.7.8, (2) Equations DPLL-20 to calculate the position stamp forwards**.

For RMO=1 and SMC=0:

ADD_IN_CAL1 = ADD_IN_CALE (DPLL-26b)

LOW_RES=0 and TS0_HRS=1 is not possible. For such a configuration the RCS bit in the DPLL_STATUS register is set together with the ERR bit.

In the automatic end mode (DMO=0) missing pulses should be sent to the input RPCUx (rapid pulse catch up on) in **Figure 38.116**, to be caught up on with CMU_CLK0 (for COA=0). When normal and rapid pulses are generated simultaneously, the SUB_INCx frequency is doubled at this moment in order to count two pulses at the TBU_CHx_BASE register. In order to make the frequency doubling possible, the CMU_CLK0 should be having a frequency which does not exceed half the frequency of the system clock. In addition the ADD_IN value should never exceed the value 0x800000 when the TS_CLK frequency exceeds half the frequency of the system clock. This limitation is only necessary for DMO=0 and COA=0 (see DPLL_CTRL_1 register). For the emergency mode replace ADD_IN of the ADDER (see **Figure 38.116**) by ADD_IN_CAL1 (when calculated, DLM=0) or ADD_IN_LD1 (when provided by the CPU, DLM=1). The sub-pulse generation in this case is done by the following calculations using a 24 bit adder with a carry out c_{one} and the following inputs: – ADD_IN – the second input is the output of the adder, stored one time stamp clock before.

In order not to complicate the calculation procedure use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

38.22.8.4 Sub pulse generation for SMC = 1

(1) Necessity of two pulse generators

The Adder of **Figure 38.116** must be implemented twice in the case of SMC=1: one for SUB_INC1 controlled by the TRIGGER input and (while RMO=1) one for SUB_INC2, controlled by the STATE input. In the case described in the chapter above for SMC=0 only one Adder is used to generate SUB_INC1 controlled by the TRIGGER in normal mode or by STATE in emergency mode.

(2) Equation DPLL-27 to calculate the number of pulses to be sent for the first device using the automatic end mode condition

For SMC=1 and DMO=0 NMB_T = (MLS1 + PD_store) *SYN_T + MP + MPVAL1 (DPLL-27) with PD_store = ADT_T[12:0],prefetched during last increment SYN_T = ADT_T[18:16], prefetched during last increment MPVAL1 = pulse correction value for PCM1_SHADOW_TRIGGER=1 while the value for PD_store is zero for AMT=0 and for COA=0 use zero instead of the value of MP

(3) Equation DPLL-28 to calculate the number of pulses to be sent for the second device using the automatic end mode condition

for RMO=1, SMC=1 and DMO=0 NMB_S = (MLS2 + PD_S_store) *SYN_S+ MP + MPVAL2 (DPLL-28) with PD_S_store = ADT_S[15:0], prefetched during last increment SYN_S = ADT_S[21:16], prefetched during last increment MPVAL2 = pulse correction value for PCM2_SHADOW_STATE=1 while the value for PD_S_store is zero for AMS=0 and for COA=0 use zero instead of the value of MP

Note that these calculations above in equations DPLL-27 and DPLL-28 are only valid for an automatic end mode (DMO = 0). In addition the number of generated pulses is added by 0.5 as shown in equations DPLL-30 or DPLL-31 respectively in order to compensate rounding down errors at the succeeding division operation. Because in automatic end mode the number of pulses is limited by INC_CNTx it is guaranteed, that not more pulses as needed are generated and in the same way missing pulses are made up for the next increment.

(4) Equation DPLL-30 to calculate ADD_IN for the first device for SMC = 1

The sub-pulse generation in this case is done by the following calculations using a 24 bit adder with a carry out c_{out} and the following inputs:

– ADD_IN – the second input is the (delayed) output of the adder, stored with each time stamp clock. Replace ADD_IN by ADD_IN_CAL1 (when calculated, DLM1=0) or ADD_IN_LD1 (when provided by the CPU, DLM1=1) respectively while:

For SMC=1 and LOW_RES=TS0_HRT ADD_IN_CAL1= (NMB_T+0.5) * RCDT_TX (DPLL-30) When RCDT_TX is the 2 time value of the quotient in equation DPLL-17b **Section 38.22.7.6, (3) Equations DPLL-17 to calculate the position stamp backwards**.

For SMC=1,LOW_RES= 1 and TS0_HRT=0 ADD_IN_CAL1= (NMB_T+0.5) * (RCDT_TX /8) (DPLL-30a) When RCDT_TX is the 2 time value of the quotient in equation DPLL-17b **Section 38.22.7.6, (3) Equations DPLL-17 to calculate the position stamp backwards**.

In order not to complicate the calculation procedure use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

ADD_IN_CAL1 is a 24 bit integer value. The CDT_TX is the expected duration of current TRIGGER increment. The c_{out} of the adder influences directly the SUB_INC1 output of the DPLL (see **Figure**

38.116). The SUB_INC1 output is in automatic end mode only enabled by EN_C1 when INC_CNT1 >0.

(5) Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1

Replace ADD_IN by ADD_IN_CAL2 (when calculated, DLM2=0) or ADD_IN_LD2 (when provided by the CPU, DLM2=1) respectively while:

for SMC=1, RMO=1 and LOW_RES=TS0_HRS: ADD_IN_CAL2= (NMB_S+0.5)* RCDT_SX (DPLL-31) When RCDT_SX is the 2 time value of the quotient in equation DPLL-20b **Section 38.22.7.8, (2) Equations DPLL-20 to calculate the position stamp forwards**.

for SMC=1, RMO=1, LOW_RES=1 and TS0_HRS=0: ADD_IN_CAL2= (NMB_S+0.5)* (RCDT_SX /8) (DPLL-31a) When RCDT_SX is the 2 time value of the quotient in equation DPLL-20b **Section 38.22.7.8, (2) Equations DPLL-20 to calculate the position stamp forwards**.

In order not to complicate the calculation procedure use a Multiplier with a sufficient bit width at the output and use the corresponding shifted output bits.

The c_{out} of the adder2 influences directly the SUB_INC2 output of the DPLL (see **Figure 38.116**). The SUB_INC2 output is in automatic end mode only enabled by EN_C2 when **INC_CNT2** >0.

NOTE

Note that after RESET and after EN_Cxc=0 (after stopping in automatic end mode) the flip-flops (FFs) have a zero value and also EN_Cxg has to be zero until reliable ADD_IN values are available and the pulse generation starts. The calculated values for the increment prediction using equations DPLL-2c **Section 38.22.6.2, (7) Equations DPLL-5 to calculate the current increment value**, (DPLL_11057)[XREF TARGET '' (DPLL_11057)] NOT EXIST in chapter **Section 38.22.6.4, (4) Equation DPLL-5 to calculate the current increment value**, (DPLL_11041)[XREF TARGET '' (DPLL_11041)] NOT EXIST in **Section 38.22.6.3, (7) Equations DPLL-10 to calculate the current increment (nominal value)** or (DPLL_11077)[XREF TARGET'' (DPLL_11077)] NOT EXIST in **Section 38.22.6.5, (4) Equations DPLL-10 to calculate the current increment value** respectively are valid only when NUTE>1 or NUSE>1 respectively. For NUTE=1 or NUSE=1 respectively the equations DPLL-30 (see **Section 38.22.8.4, (4) Equation DPLL-30 to calculate ADD_IN for the first device for SMC = 1**) and DPLL-31 (see **Section 38.22.8.4, (5) Equation DPLL-31 to calculate ADD_IN for the second device for SMC = 1**) use the actual increment value subtracted by the weighted average error.

The generation of SUB_INCx pulses depends on the configuration of the DPLL. In automatic end mode the counter INC_CNTx resets the enable signal EN_Cxcu when the number of pulses desired is reached. In this case only the uncompensated outputs SUB_INCx remain active in order to provide pulses for the input filter units. A new TRIGGER or STATE input respectively can reset the FFs and also ADD_IN, especially when EN_Cxc was zero before. In the case of acceleration missing pulses can be determined at the next TRIGGER/STATE event easily. For the correction strategy COA = 0 those missing pulses are sent out with CMU_CLK0 frequency as soon they are determined. After that the pulse counter INC_CNTx should be always zero and the new pulses are sent out afterwards, when INC_CNTx is set to the desired value by adding MLS1 or MLS2 for the new TRIGGER or STATE event respectively.

Because the used DIV procedure of the algorithms results only in integer values, a systematic failure could appear. The pulse generation will stop when the INC_CNTx register reaches zero or all

remaining pulses at a new increment will be considered in the next calculation. In this way the loss of pulses can be avoided.

When a new TRIGGER appears the value of SYN_T*MLS1 is added to INC_CNT1. Therefore for FULL_SCALE 2*(TNU+1)*MLS1 pulses SUB_INC1 generated, when INC_CNT1 reaches the zero value. The generation of SUB_INC1 pulses has to be done as fast as possible.

When a new STATE appears the value of SYN_S*MLS2 is added to INC_CNT2. Therefore for FULL_SCALE 2*(SNU+1)*MLS2 pulses SUB_I- NC2 generated, when INC_CNT2 reaches the zero value. The generation of SUB_INC2 pulses has to be done as fast as possible.

38.22.8.5 Calculation of the Accurate Position Values

All appearing TRIGGER and STATE signals do have a time stamp and a position stamp assigned after the input filter procedure. For the calculation of the exact time stamp the filter values are considered in the calculations of equations **Section 38.22.6.2, (1) Equations DPLL-1a to calculate TRIGGER time stamps** or DPLL-6a **Section 38.22.6.3, (1) Equations DPLL-6a to calculate STATE time stamps** respectively. A corresponding calculation is to be performed for the calculation of position values. The PSTC and PSSC values can be corrected by the CPU, when needed.

After reset, while FTD=0 and no active TRIGGER slope is detected: PSTC = 0 (DPLL-32a)

Calculate the new Position value for each active TRIGGER event: PSTC= PSTC_old + NMB_T_TAR_OLD (DPLL-32b) when FTD=1 and SGE1=1 with PSTC_old is the last PSTC value and NMB_T_old is the number of pulses which are calculated and provided for sending out in the last increment.

After reset, while FSD=0 and no active STATE slope is detected: PSSC= 0 (DPLL-33a)

Calculate the new Position value for each STATE event: PSSC= PSSC_old + NMB_S_TAR_OLD (DPLL-32b) when FSD=1 and SGE1=1 (SMC=0) or SGE2=1 (SMC=1) respectively with PSSC_old is the last PSSC value and NMB_S_old is the number of pulses which are calculated and provided for sending out in the last increment.

38.22.8.6 Scheduling of the Calculation

After enabling the DPLL with each active TRIGGER or STATE event respectively a cycle of operations is performed to calculate all the results shown in detail in the table below **Section 38.22.2, Requirements and demarcation**. A state machine controls this procedure and consists of two parts, the first is triggered by an active slope of the signal TRIGGER, begins at step 1 and ends at step 20 (in normal mode and for SMC=1). The second state machine is controlled by an active slope of the signal STATE, begins at step 21 and ends at step 40 (in emergency mode and also for SMC=RMO=1). Depending on the mode used all 20 steps are executed or already after 2 steps the jump into the initial state is performed, as shown in the state machine descriptions below. For each new extended cycle (without this jump) all prediction values for actions in the case SMC=0 are calculated once more (with maybe improved accuracy because of better parameters) and all pending decisions are made using these new values when transmitted to the decision device.

In (DPLL_6908)[XREF TARGET " (DPLL_6908)] NOT EXIST the steps of the state machine are described. Note that the elaboration of the steps depends on the configuration bits described in the comments. The steps 4 to 17 are only calculated in normal mode (in the state machine explanation below marked yellow in **Section 38.22.2, Requirements and demarcation**), but steps 24 to 37 are only calculated in emergency mode (in the state machine explanation below marked cyan in **Section 38.22.2, Requirements and demarcation**) when SMC=0.

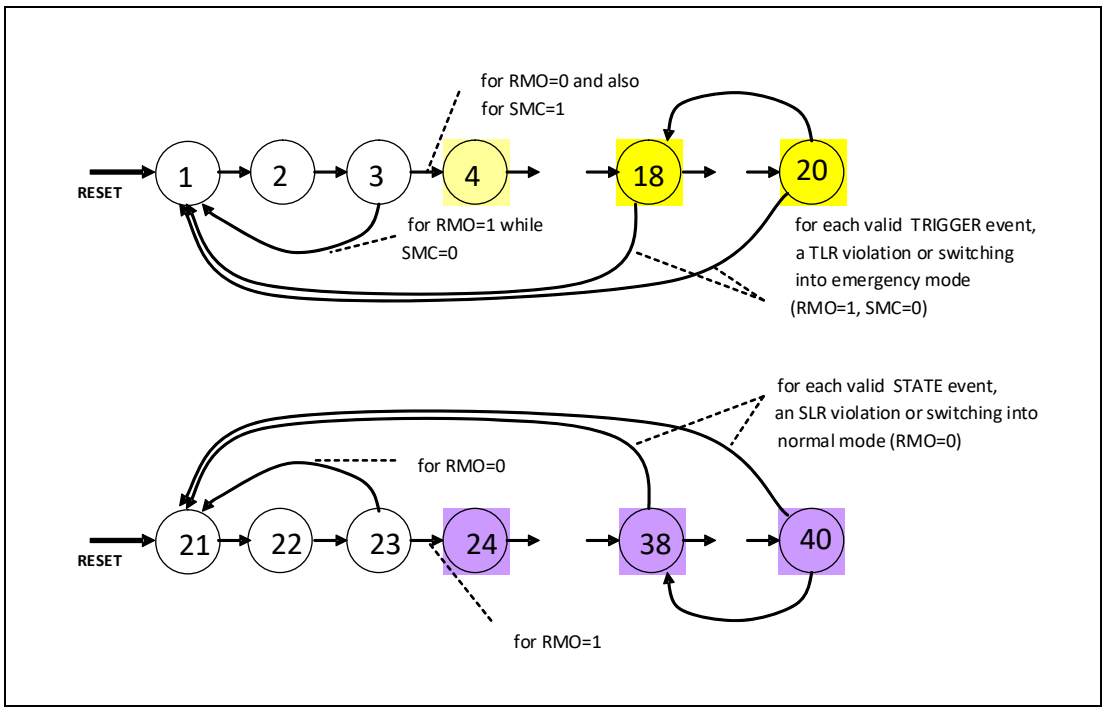


Figure 38.117 State machine partitioning for normal and emergency mode.

(1) Synchronization description

TRIGGER:

The APT (address pointer for duration and reciprocal duration values of TRIGGER increments) is initially set to zero and incremented with each active TRIGGER event. Therefore data are stored in the RAM beginning from the first available value. The actual duration of the last increment is stored at DT_T_ACT. For the prediction of the next increment it is assumed, that the same value is valid as long as NUTE is one.

A missing TRIGGER is assumed, when at least after TOV* DT_T_ACT no active TRIGGER event appears.

The data of equations DPLL-1b1 and DPLL-1c2 **Section 38.22.7.5, (4) Equations DPLL-1b1 and DPLL-1c1 to update the RAM after calculation** are written in the corresponding RAM regions and APT is incremented accordingly up to 2*TNU–2*SYN_NT+1.

The APT_2B (address pointer for the time stamp field of TRIGGER) is initially set to zero and incremented with each active TRIGGER event. When no gap is detected because of the incomplete synchronization process at the beginning, for all TRIGGER events the time stamp values are written in the RAM up to 2*(TNU+1) entries, although only 2*(TNU+1–SYN_NT) events in FULL_SCALE appear. When the current position is detected, the synchronization procedure can be performed as described below: Before the CPU sets the APT_2C address pointer in order to synchronize to the profile, it writes the corresponding increment value for the necessary extension of the RAM region 2b value APT_2B_EXT into the register APT_2B_sync and sets the status bit APT_2C_status. This value can be e.g. 2*SYN_NT, when all gaps in FULL_SCALE already passed the input data stream of TRIGGER, or less than this value, when up to now e.g. only a single gap is to be considered in the data stream stored already in the RAM region 2b. The number of virtual increments to be considered depends on the number of inputs already got. After writing APT_2C by the CPU, with the next TRIGGER event the APT_2B address pointer is incremented (as usual) and then the additional offset value APT_2B_EXT is added to it once (while APT_2B_STATUS=1 and for forward direction). For that reason the APT_2B_STATUS bit is reset after it. The old APT_2B value before adding the offset is stored in the APT_2B_OLD register as information for the CPU where to start the extension procedure. In the following the CPU fills in the time stamp field around the APT_2B_OLD position taking into account the corresponding number of virtual entries stored in the APT_2B_EXT value and the corresponding NT values in the profile. The extension procedure ends when all gaps considered in the APT_2B_EXT value are treated once. In the consequence all storage locations of RAM region 2b up to now do have the corresponding entries. Future gaps are treated by the DPLL. For a backward direction the APT_2C_ext value is subtracted accordingly.

When the CPU writes the APT_2C address pointer the SYT bit is set simultaneously. For SYT=1 in normal mode (SMC=0) the LOCK1 bit is set with the system clock, when the right number of increments between two synchronization gaps is detected by the DPLL. An unexpected missing TRIGGER or an additional TRIGGER between two synchronization gaps does reset the LOCK1 bit in normal mode. In that case the CPU must correct the SUB_INC pulse number and maybe correct the APT_2C pointer. For this purpose the LL1I interrupt can be used.

When SYT is set the calculations of equations DPLL-1 to DPLL-5 are performed accordingly and the values are stored in (and distributed to) the right RAM positions. This includes the multiple time stamp storage by the DPLL for a gap according to equations DPLL-1a5 to 7 forwards **Section 38.22.7.5, (2) Equation DPLL-1a5-7 to extend the time stamp values for TRIGGER in forward direction** or backwards **Section 38.22.7.5, (3) Equations DPLL-1a5-7 for backward direction**. The APT_2B pointer is for that reason incremented or decremented before this operation considering the virtual increments in addition. Note that for the APT and APT_2C pointers the gap is considered as a single increment.

STATE:

The APS (address pointer for duration and reciprocal duration values of STATE) is initially set to zero and incremented with each active STATE event. Therefore data are stored in the RAM field beginning at the first location. The actual duration of the last increment is stored at DT_S_ACT. For the prediction of the next increment it is assumed, that the same value is valid as long as NUSE is one.

A missing STATE is assumed, when at least after SOV* DT_S_ACT no active STATE event appears.

The data of equations DPLL-6b1 and DPLL-6c2 **Section 38.22.7.5, (8) Equations DPLL-6b1 and DPLL-6c2 to update the RAM after calculation** is written in the corresponding RAM regions and APS is incremented accordingly up to 2*SNU–2*SYN_NS+1 (for SYSF=0).

The APS_1C2 (address pointer for the time stamp field of STATE) is initially set to zero and incremented with each active STATE event. When no gap is detected because of the incomplete synchronization process at the beginning, for all STATE events the time stamp values are written in the RAM up to 2*(SNU+1) entries, although (e.g. for SYSF=0) only 2*(SNU+1–SYN_NS) events in FULL_SCALE appear. When the current position is detected, the synchronization procedure can be performed as described below: Before the CPU sets the APS_1C3 address pointer in order to synchronize to the profile, it writes the corresponding increment value APS_1C2_EXT for the necessary extension of the RAM region 1c2 into the register DPLL_APS_SYNC and sets the APS_1C2_STATUS bit there. This value can be e.g. 2*SYN_NS (for SYSF=0) or SYN_NS (for SYSF=1), when all gaps in FULL_SCALE already passed the input data stream of STATE. Also less than this value can be considered, when up to now only a single gap is to be considered in the data stream stored already in the RAM region 1c2. The number of increments to be considered depends on the number of inputs already got. After writing APS_1C3 by the CPU, with the next active STATE slope the APS_1C2 address pointer is incremented (as usual) and then the additional offset value APS_1C2_EXT is added to it once (while APS_1C2_STATUS=1 and forward direction). For that reason the APS_1C2_STATUS bit is reset after it. The old APS_1C2 value is stored in the APS_1C2_OLD register as information for the CPU where to start the extension procedure. In the following the CPU extends the time stamp field beginning from the APS_1C2_OLD position taking into account the corresponding number of virtual entries according to the APS_1C2_EXT value and also the correspondent NS values in the profile. The extension procedure ends when all gaps considered in the APS_1C2_EXT value are treated once. In the consequence all storage locations of RAM region 1c2 up to now do have the corresponding entries. Future gaps are treated by the DPLL. For a backward direction the APS_1C2_EXT value is subtracted accordingly.

When the CPU writes the APS_1C3 address pointer the SYS bit is set simultaneously. For SYS=1 in emergency mode (SMC=0 and DMO=1) the LOCK1 bit is set with the system clock, when the right number of increments between two synchronization gaps is detected by the DPLL. An unexpected missing STATE or an additional STATE between two synchronization gaps does reset the LOCK1 bit in emergency mode. In that case the CPU must correct the SUB_INC1 pulse number and maybe correct the APS_1C3 pointer. For this purpose the LL1I interrupt can be used.

When SYS is set the calculations of equations DPLL-5 to DPLL-10 are performed accordingly and the values are stored in (and distributed to) the right RAM positions. This includes the multiple time stamp storage by the DPLL for a gap according to equations DPLL-6a5 to 7 forwards **Section 38.22.7.5, (6) Equations DPLL-6a5-7 to extend the time stamp values for STATE** or backwards **Section 38.22.7.5, (7) Equations DPLL-6a5-7 for backward direction**. The APS_1C2 pointer is for that reason incremented or decremented before this operation considering the virtual increments in addition. Note that for the APS and APS_2c pointers the gap is considered as a single increment.

SMC=1:

For SMC=1 it is assumed, that the starting position is known by measuring the characteristic of the device. In this way the APT and APT_2C as well the APS and APS_1C3 values are set properly, maybe with an unknown repetition rate. When no gap is to be considered for TRIGGER or STATE signals the APT_2B and APS_1C2 address pointers are set equal to APT or APS respectively. It is assumed, that all missing TRIGGER s and missing STATE s can be also considered from the beginning, when a valid profile with the corresponding adapted values is written in the RAM regions 1c3 and 2c respectively. In that case the TSF_T[i] and TSF_S[i] must be extended by the DPLL according to the profile. Thus the SYT and SYS bits could be set from the beginning and the LOCK1 and LOCK2 bits are set after recognition of the corresponding gaps accordingly. When no gap exists (SYN_NT=0 or SYN_NS=0), the LOCK bits are set immediately. The CPU can correct the APT_2C and APS_1C3 pointer according to the recognized repetition rate later once more without the loss of Lock1,2.

(2) Operation for direction change in normal and emergency mode (SMC = 0)

When for SMC=0 in normal mode a backwards condition is detected for the TRIGGER input signal (e.g. when THMI is not violated), the LOCK1 bit in the DPLL_STATUS register is reset, the NUTE value in NUTC register is set to 1 (the same for NUSE in NUSC). The address pointers APT_2C as described below (and after that decremented for each following active slope of TRIGGER as long as the DIR1 bit shows the backward direction). Notice that in the case of the change of the direction the ITN and ISN bit in the DPLL_STATUS register are reset.

For this transition to the backward direction no change of address pointer APT and APT_2B is necessary.

profile update for TRIGGER when changing direction The profile address pointer APT_2C is changed step by step in order to update the profile information in SYN_T, SYN_T old and PD_store:

1. decrement APT_2C, load SYN_T
2. decrement APT_2C, load SYN_T
3. decrement APT_2C, load SYN_T, PD_store, update SYN_T_OLD
4. decrement APT_2C, make calculations, load SYN_T and PD_store, update SYN_T_OLD and PD_store_old and wait for a new TRIGGER event.

NOTE

The update of SYN_T_OLD and the loading of PD_store can be performed in all steps above. The value of APT_2B needs not to be corrected. For a direction change from backwards to forwards make the same corrections by incrementing APT_2C.

Make calculations does mean: the operation of the state machine starts with the calculations of NMB_T and INC_CNT1 using the actual APT_2C address pointer value, see **Section 38.22.2, Requirements and demarcation**.

The TBU_TB1 value is to be corrected by the number of pulses sent out in the wrong direction mode during the last and current increment. This correction is done by sending out SUB_INC1 pulses for decrementing TBU_TB1 (while DIR1=1).

Save inc_cnt1 value at direction change to inc_cnt1_save. Calculate the new inc_cnt1 value as follows:

1. Stop sending pulses and save inc_cnt1 at the moment of direction change as inc_cnt1_save.

2. Set inc_cnt1 to the target value of the last increment
   nmb_t_tar_old
3. Add the target number of trigger which were calculated for the current increment when this value was already added to inc_cnt1 before the direction change is detected
   + nmb_t_tar
4. Subtract the value of still not sent pulses (remaining value at inc_cnt1_save)
   – inc_cnt1_save
5. Calculate the new target pulses to be sent considering the new values of SYN_T and PD_store and add them:
   + nmb_t_tar_new

This does mean the following equation: inc_cnt1 = nmb_t_tar_old + nmb_t_tar – inc_cnt1_save + nmb_t_tar_new All pulses summarized at inc_cnt1 are sent out by the maximum possible frequency, because no speed information is available for the first increment after changing the direction. Notice that no pulse correction using PCM1 of DPLL_CTRL1 is possible during direction change. When PSTC was incremented/decremented at the active slope and after that the direction change was detected at the same input event, correct PSTC once by – nmb_t_tar_old when changed to backwards + nmb_t_tar_old when changed to forwards in order to compensate the former operation. When the direction information is known before an intended change of PSTC, do not change them. Store the new calculated value nmb_t_tar_new at nmb_t_tar for the correct calculation of PSTC at the next input event.

consequences for STATE With the next active STATE event the direction information is already given. The profile pointer APS_1C3 is to be corrected by a two times decrement in order to point to the profile of the next following increment. In the following it is decremented with each STATE event while DIR1=1. The SYN_S and PD_S_store values must be updated accordingly, including SYN_S_OLD and PD_S_store_old.

Because the right direction is already known when an input event appears, make the following corrections: – decrement APS_1C3, load SYN_S and PD_S_store, update SYN_S_OLD and PD_S_store_old – decrement APS_1C3, make calculations, load SYN_S and PD_S_store, update SYN_S_OLD and PD_S_store_old and wait for a new STATE event. Note: The update of SYN_S_OLD and the loading of PD_S_store can be performed in all steps above. The value of APS_1C2 needs not to be corrected.

When a new STATE event occurs, all address pointers are decremented accordingly as long as DIR1=1.

In emergency mode the pulses are corrected as follows: Save inc_cnt1 value at direction change to inc_cnt1_save. Calculate the new inc_cnt1 value as follows:

1. Stop sending pulses and save inc_cnt1 at the moment of direction change as inc_cnt1_save.
2. Set inc_cnt1 to the target value of the current increment nmb_s_tar Note: In contrast to the normal mode, nmb_s_tar is to be used instead of nmb_s_tar_old, because direction information in emergency mode is only given from the TRIGGER input and occurs of a STATE event independently. That means: The calculations at the last STATE event were done for the correct former direction. In addition still no pulse calculations are performed for the current increment, because the direction change is known at the moment of the recent STATE event. Later direction changes are considered at the next STATE event.
3. Do not add the calculated number of state pulses because no new STATE event occurred.
4. Subtract the value of still not sent target pulses (remaining value at inc_cnt1_save) – inc_cnt1_save

5. Add the new calculated target pulses for the current increment + nmb_s_tar_new when for the calculation all new conditions of PD_S_store and SYN_S are considered. inc_cnt1 = nmb_s_tar_old – inc_cnt1_save + nmb_s_tar_new All pulses summarized at inc_cnt1 are sent out by the maximum possible frequency, because no speed information is available for the first increment after changing the direction. Notice that no pulse correction using PCM1 of DPLL_CTRL1 is possible during direction change. Do not change PSSC and suppress incrementing/decrementing of PSSC at the event directly following to the direction change information. Store the new calculated value nmb_s_tar_new at nmb_s_tar for the correct calculation of PSTC at the next input event.

repeated change to forward direction for TRIGGER The DIR1 bit remains set as long as the THMI value remains none violated for the following TRIGGER events and is reset when for an inactive TRIGGER slope the THMI is violated.

Resetting the DIR1 to 0 results (after repeated reset of LOCK1, ITN, ISN) the opposite correction of the profile address pointer considered.

This does mean two increment operations of the address pointer APS_1C3 including the update of SYN_S and PD_S_store with the automatic update of SYN_S_OLD and PD_S_store_old for STATE and four increment operations of the address pointer APT_2C including the update of SYN_T and PD_store with the automatic update of SYN_T_OLD and PD_store_old for TRIGGER.

The correction of TBU_CH1 is done by sending out the correction pulses with the highest possible frequency at SUB_INC1 while DIR1=0. The number of pulses is calculated as shown above.

consequences for STATE see corrections above. After that the address pointers are incremented again with each following active STATE event as long as DIR1 = 0.

(3) Operation for direction change for TRIGGER (SMC = 1)

When for SMC=1 a backwards condition is detected for the TRIGGER input signal (TDIR=1, resulting in DIR1=1), the LOCK1 bit in the DPLL_STATUS register is reset, the NUTE value in NUTC register is set to 1. The address pointers APT and APT_2C as well as APT_2B are decremented for each active slope of TRIGGER as long as the DIR1 bit shows the backward direction. Notice that in the case of the change of the direction the ITN bit in the DPLL_STATUS register is reset.

profile update for TRIGGER Make the same update steps for the profile address pointer as shown in **Section 38.22.8.6, (2) Operation for direction change in normal and emergency mode (SMC = 0)** : Decrement APT_2C for 2 times with the update of the SYN_T and PD_store values at each step with an automatic update of SYN_T_OLD and PD_store_old:

1. decrement APT_2C, load SYN_T, PD_store, update SYN_T_OLD
2. decrement APT_2C, make calculations, load SYN_T and PD_store, update SYN_T_OLD and PD_store_old and wait for a new TRIGGER event.

In the normal case no correction of wrong pulses sent is necessary, because the direction change is detected by the pattern immediately. Nevertheless a correction is necessary as shown below. In the other case: see treatment of pulses TBU_CH1_BASE in normal mode at **Section 38.22.8.6, (2) Operation for direction change in normal and emergency mode (SMC = 0)**.

Save inc_cntx value at direction change to inc_cnt1_save. Calculate the new inc_cnt1 value as follows: 1. Clear inc_cnt1. 2. Set inc_cnt1 to the target value of the last increment nmb_t_tar Note: In contrast to the normal mode, nmb_t_tar is to be used instead of nmb_t_tar_old, because the direction information

is known before the calculation takes place. 3. Do not add the calculated number of trigger pulses because it is not calculated yet before the direction change information is known. 4. Subtract the value of still not sent pulses (remaining value at inc_cnt1_save) – inc_cnt1_save 5. Add the new calculated target pulses for the current increment + nmb_t_tar_new when for the calculation all new conditions of PD_S_store and SYN_S are considered. inc_cnt1 = nmb_t_tar_old – inc_cnt1_save + nmb_t_tar_new All pulses summarized at inc_cnt1 are sent out by the maximum possible frequency, because no speed information is available for the first increment after changing the direction. Notice that no pulse correction using PCM1 of DPLL_CTRL1 is possible during direction change.

Suppress changing of PSTC for the TRIGGER event when a direction change is detected. Store the new calculated value nmb_t_tar_new at nmb_t_tar for the correct calculation of PSTC at the next input event.

repeated change to forward direction for TRIGGER The DIR1 bit remains set as long as the TDIR bit is set for the following TRIGGER events and is reset when for an active TRIGGER slope the TDIR is zero.

Resetting the DIR1 to 0 results (after repeated reset of LOCK1 and ITN) the opposite correction of the address pointer use. This does mean two increment operations of the address pointer including the update of SYN_T and PD_store.

A complex correction of TBU_CH1_BASE and INC_CNT1 is in the normal case not necessary, when all increments are equal (SYN_NT=0) and no adapt information is used. In this case only the MLS1 value is added to INC_CNT1 in order to back count the value for the last increment. In the other case: see treatment of pulses TBU_CH1_BASE and ICN_CNT1 in normal mode at **Section 38.22.8.6, (2) Operation for direction change in normal and emergency mode (SMC = 0)**.

(4) Operation for direction change for STATE (SMC=1)

When for SMC=1 a backwards condition is detected for the STATE input signal (SDIR=1, resulting in DIR2=1), the LOCK2 bit in the DPLL_STA- TUS register is reset, the NUSE value in NUSC register is set to 1 and the address pointers APS and APS_1C3_f and APS_1C2 are decremented for each active slope of *STATE* as long as the DIR2 bit shows the backward direction.

NOTE

In case of the change of the direction the ISN bit in the DPLL_STATUS register is reset.

For this transition to the backward direction no change of address pointer APS and APS_1C2 is necessary.

profile update for STATE Make the same update steps for the profile address pointer as shown in **Section 38.22.8.6, (2) Operation for direction change in normal and emergency mode (SMC = 0)**: Decrement APS_1C3 for 2 times with the update of the SYN_S, SYN_S_OLD, PD_S_store and PD_S_store_old values at each step:

1. decrement APT_1c3, load SYN_S, PD_S_store, update SYN_S_OLD
2. decrement APT_1c3, make calculations, load SYN_S and PD_S_store, update SYN_S_OLD and PD_S_store_old and wait for a new STATE event.

A complex correction of TBU_CH2_BASE and INC_CNT2 is in the normal case not necessary, when all increments are equal (SYN_NS=0) and no adapt information is used. In this case only the MLS2 value is added to INC_CNT2 in order to back count the value for the last increment. In the other case:

see treatment of pulses TBU_CH1_BASE and ICN_CNT1 in normal mode at **Section 38.22.8.6, (2) Operation for direction change in normal and emergency mode (SMC = 0)**.

For the second PMSM the pulses are corrected as follows: Save inc_cnt2 value at direction change to inc_cnt2_save. Calculate the new inc_cnt2 value as follows:

1. Clear inc_cnt2.
2. Set inc_cnt2 to the target value of the last increment nmb_s_tar Note: In contrast to the normal mode, nmb_s_tar is to be used instead of nmb_s_tar_old, because no new calculation is performed so far.
3. Do not add the calculated number of state pulses because it is not calculated yet before the direction change information is known.
4. Subtract the value of still not sent pulses (remaining value at inc_cnt2_save) – inc_cnt2_save
5. Add the new calculated target pulses for the current increment + nmb_s_tar_new when for the calculation all new conditions of PD_S_store and SYN_S are considered. inc_cnt2 = nmb_s_tar_old– inc_cnt2_save + nmb_s_tar_new All pulses summarized at inc_cnt2 are sent out by the maximum possible frequency, because no speed information is available for the first increment after changing the direction. Notice that no pulse correction using PCM2 of DPLL_CTRL1 is possible during direction change. Do not change PSSC for a STATE event when a direction change is detected. Store the new calculated value nmb_s_tar_new at nmb_s_tar for the correct calculation of PSTC at the next input event.

repeated change to forward direction for STATE The DIR2 bit remains set as long as the SDIR bit is set for the following STATE events and is reset when for an active STATE slope SDIR is zero.

Resetting the DIR2 to 0 results (after repeated reset of LOCK2 and FSD) in the opposite correction of the address pointer use.

After a last decrementing of all address pointers the APS_1C3 is incremented 2 times with a repeated update of SYN_S, SYN_S_OLD and PD_S_store after each increment.

(5) DPLL reaction in the case of non plausible input signals

When the DPLL is synchronized concerning the TRIGGER signal by setting the FTD, SYT and LOCK1 bits in the DPLL_STATUS register, the number of active TRIGGER events between the gaps is to be checked continuously.

When additional events appear while a gap is expected, the LOCK1 bit is reset and the ITN bit in the DPLL_STATUS register is set.

When an unexpected gap appears (missing TRIGGER S), the NUTE value in the NUTC register is set to 1, the LOCK1 bit is reset and the ITN bit in the DPLL_STATUS register is set. The address pointers are incremented with the next active TRIGGER slope accordingly.

The TOR Bit in the DPLL_STATUS register is set, when the time from the last valid TRIGGER event to the next active TRIGGER slope exceeds the value of the last nominal TRIGGER duration multiplied with the value of the TLR register (see **Section 38.22.22.36, DPLL_TLR**). In this case also the TORI interrupt is generated, when enabled.

When in the following the direction DIR1 changes as described in the chapters above the ITN bit in the DPLL_STATUS register is reset, the use of the address pointers APT_2C is switched and the pulse correction takes place as described above.

In all other cases the CPU can interact to leave the unstable state. This can be done by setting the APT_2C address pointer which results in a reset of the ITN bit. In the following NUTE can also be set to higher values.

When the DPLL is synchronized concerning the STATE signal by setting the FSD, SYS and LOCK1 (for SMC=0) or LOCK2 (for SMC=1) bits in the DPLL_STATUS register, the number of active STATE events between the gaps is to be checked continuously.

When additional events appear while a gap is expected or while an unexpected missing STATE event appears, the LOCK1,2 bit is reset and the ISN bit in the DPLL_STATUS register is set.

When an unexpected gap appears for RMO=SMC=1 (missing STATE s for synchronous motor control), the NUSE value in the NUSC register is set to 1, the LOCK2 bit is reset and the ISN bit in the DPLL_STATUS register is set. The address pointers are incremented with the next active STATE slope accordingly.

When the STATE locking range SLR is violated, the state machine 2 will remain in state 21 and the address pointer APS, APS_1C2 and APS_1C3 will remain unchanged until the CPU sets the APS_1C3 accordingly. In this case also the NUSE value in the NUSC register is set to 1. The DPLL stops the generation of the SUB_INC1,2 pulses respectively and will perform no other actions – remaining in step21 of the second state machine. The SOR Bit in the DPLL_STATUS register is set, when the time to the next active STATE slope exceeds the value of the last nominal STATE duration multiplied with the value of the SLR register.

In this case also the SORI interrupt is generated, when enabled.

When in the following the direction DIR2 changes as described in the chapters above the ISN bit in the DPLL_STATUS register is reset, the use of the address pointers APS_1C3 is switched and the pulse correction takes place as described above. In all other cases the CPU must interact to leave the unstable state. This can be done by setting the APS_1C3 address pointers which results in a reset of the ISN bit. In the following NUSE can also be set to higher values.

(6) State description of the State Machine.

Table 38.291 State description of the State Machine Table (1/7)

Step	Description	Comments

Always for DEN=1	for each inactive TRIGGER slope with TEN=1:
	check, if the last active TRIGGER slope was passing the PVT check; only in this case perform the following tasks:
	calculate the time stamp difference ?T to the last active event, store this value at THVAL;
	when THMI >0 is violated (?T < THMI):
	generate TINI interrupt,
	set DIR1=0 (forwards)
	set BWD1=0 (see DPLL_STATUS register)
	else (only for THMI >0):
	set DIR1= 1 (backwards);
	set BWD1=1 (see DPLL_STATUS register)
	after changing the direction correct the pulses WP sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency:
	WP=NMB_T-DPLL_INC_CNT1;
	correct INC_CNT1 by addition of 2*WP before sending the correction pulses;
	generate the TISI interrupt;
	
	check THMA, when THMA is violated, generate the TAXI interrupt; go to step 1
	for each inactive STATE slope with SEN=1:
	set DIR2=DIR1
		for SMC=0;
		set DIR1 always after inc./ decr. the address pointers APT, APT_x;
		go to step 1;
		stop output of SUB_INC1 and correct pulses after changing DIR1 after incr./ decr. of APS_x
		set DIR2 always after incr./decr. the address pointers APS, APS_x;
		go to step 1

Always for DEN=1 and (TEN=1 or SEN=1, respectively)	set DIR1=BWD1=TDIR, set DIR2=BWD2=SDIR; for each change of TDIR go to step 1 after performing the following calculations: correct INC_CNT1 correct the pulses (WP, see above) sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency. For each change of SDIR go to step 21 after performing the following calculations: update of SYN_S, PD_S_store according to chapter correct INC_CNT1,2 correct the pulses sent with wrong direction information and send the pulses for the actual increment in addition with highest possible frequency.	**for SMC=1**; set the direction bits always after incr./decr. the corresponding address pointers;

Table 38.291 State description of the State Machine Table (2/7)

Step	Description	Comments
1	When DEN = 0 or TEN=0: stay in step 1 until DEN=1, TEN=1 and at least one active *TRIGGER* has been detected (FTD=1); the following steps are performed always (not necessarily in step 1, but also in steps 18 to 20 (when waiting for new PMTR values to be calculated): compare TRIGGER_S with TSL (active slope); **When no active TRIGGER slope appears** and when TS_T_CHECK time is reached: send missing *TRIGGER* INT, also when a gap is expected according to the profile; set MT=1 (missing *TRIGGER* bit) in the DPLL_STATUS register; do not leave the active step, until a valid active *TRIGGER* appears. When an active TRIGGER slope appears check PVT - when the PVT value is violated: generate the PWI interrupt, ignore the *TRIGGER* input and wait for the next active TRIGGER slope (ignore each inactive slope); do not store any value - When the PVT value is fulfilled: store the actual position stamp at PSTM (value at the TRIGGER event) update the RAM region 2 by equation DPLL-1a-c (see chapter) **store the actual INC_CNT1 value at MP1 as missing pulses** *(instead of calculation in step 5)* store all relevant configuration bits **X** of the DPLL_CTRL(0,1) Registers in **shadow** registers and consider them for all corresponding calculations of steps 2 to 20 accordingly; the relevant bits are explained in the registers itself generate the TASI interrupt;	Depending on TSL, TEN, DEN step one is leaving with the next *TRIGGER* input; **Note:** Step 1 is also left in emergency mode when an active *TRIGGER* event appears in order to make a switch back to normal mode possible; _old - values are values valid at the last but one active *TRIGGER* event; for the whole table: use always MLS1 instead of (MLT+1) for the case SMC=1; **dir_crement** does mean: increment for DIR1=0 decrement for DIR1=1 *)replace (MLT+1) by MLS1 for SMC=1 **) NMB_T_TAR is the target value of NMB_T of the last increment (see step 5 ff.) ***) add MPVAL1 once to INC_CNT1, that means only when PCM1=1 ****) **SGE1_delay** is the value of SGE1 delayed by one active TRIGGER event *****) PD_store = 0 for AMT=0 (see DPLL_CTRL_0 register)

	for FTD=0: set PSTC=PSTM
	set FTD (first *TRIGGER* detected)
	do not change PSTC,APT, APT_2B
	for (RMO=0 or SMC=1)
	and SGE1=1: increment INC_CNT1 by (MLT+1)*) +MPVAL1***)
	send SUB_INC1 pulses with highest possible frequency when SGE1=1 and DPLL_CTRL_11.SIP1 = 0.
	for SYT=0 and FTD =1:
	dir_crement APT and APT_2B by one;
	dir_crement for **SGE1_delay******)**=1:** PSTC by NMB_T_TAR**)
	for (RMO=0 or SMC=1) **and SGE1=1**: increment INC_CNT1 by (MLT+1)*) +MPVAL1***)
	
	for SYT=1:
	dir_crement APT, APT_2C**, dir_crement** APT_2B by SYN_T_OLD
	dir_crement for **SGE1_delay******)**=1** PSTC by NMB_T_TAR**)
	for (RMO=0 or SMC=1) **and SGE1=1**: increment INC_CNT1 by SYN_T*((MLT+1)*)+ PD_store)*****) + MPVAL1***)
	PD_store is 0 for AMT=0
	within the DPLL_STATUS register:
	set LOCK1 bit accordingly;	

2	calculate TS_T according to equations DPLL-1a; calculate DT_T_ACT = TS_T - TS_T_OLD calculate RDT_T_ACT calculate QDT_TX according to equation DPLL-2	

Table 38.291 State description of the State Machine Table (3/7)

Step	Description	Comments
3	send CDTI interrupt when NTI_CNT is zero or decrement NTI_CNT when not zero; calculate EDT_T and MEDT_T according to equations DPLL-3 and DPLL-4 for (RMO=1 and SMC=0): update SYN_T, PD_store and go back to step 1	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_T and PD_store are not updated until a new active TRIGGER slope occurs.
4	calculate CDT_TX according to equation DPLL-5a and b;	for RMO=0 or SMC=1;

5	calculate missing pulses:
	MP1 = INC_CNT1(at the moment of an active TRIGGER slope)
	
	calculate target pulses:
	NMB_T_TAR = ((MLT+1)*) *+ PD_store) *SYN_T + MPVAL1
	(instead of PD_store use zero in the case AMT=0)
		for RMO=0 or SMC=1;
		
		*)replace (MLT+1) by MLS1 for SMC=1;
		
		add MPVAL1 only for PCM=1 and reset PCM1 after that;

6	sent MP with highest possible frequency and set NMB_T = NMB_T_TAR	for RMO=0 or SMC=1, DMO=0 and COA=0
7	calculate the number of pulses to be sent NMB_T = NMB_T_TAR + MP (see equations DPLL-21 or DPLL-27 respectively)	for RMO=0 or SMC=1, DMO=0 and COA=1
8	NMB_T = SYN_T*CNT_NUM_1	for RMO=0 or SMC=1, DMO=1
9	update SYN_T and PD_store;	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_T and PD_store are not updated until a new active TRIGGER slope occurs.
10	calculate ADD_IN_CAL1 according to equation DPLL-25 and DPLL-25b or DPLL-31 and store this value in RAM use ADD_IN_CAL1 as ADD_IN value for the case DLM=0 use ADD_IN_LD1 as ADD_IN for the case DLM=1, but do this update immediately (without waiting for this step 10); for DMO=DLM=0 and EN_C1u=0: reset the flip-flops in the SUB_INC1 generator; start sending SUB_INC1;	for RMO=0 or SMC=1 for DLM=0 for DLM=1
11	calculate TS_T_CHECK = TS_T + DT_T_ACT *(TOV);	for RMO=0 or SMC=1;
12	automatic setting of actions masking bits in the DPLL_STATUS register: for SMC=0: set CAIP1=CAIP2=1 for SMC=1: set only CAIP1=1	steps 12 to 16 are not valid for the combination: (SMC=0 and RMO=1)
13	for all correspondent actions with ACT_N[i]=1 calculate: NA[i] = (PSA[i] - PSTC)/(MLT+1)*)for forward direction with w= integer part and b = remainder of the division (fractional part); for backward direction use NA[i] = (PSTC - PSA[i])/(MLT+1)*) and consider in both cases the time base overflow in order to get a positive difference	actions 0...11 for SMC=1 actions 0...23 for SMC=0 depending on ACT_N[i] in **DPLL_ACT_STA** register; replace MLT+1 by MLS1 for SMC=1
14	calculate PDT_T[i] and DTA[i] for up to 24 action values according to equations DPLL-11 and DPLL-12;	actions 0...11 for SMC=1 actions 0...23 for SMC=0
15	calculate TSAC[i] according to equation DPLL-15 and PSAC[i] according to equation DPLL-17	actions 0...11 for SMC=1 actions 0...23 for SMC=0
16	automatic resetting of actions masking bits in the DPLL_STATUS register: for SMC=0: set CAIP1=CAIP2=0 for SMC=1: set only CAIP1=0; set the corresponding ACT_N[i] bits in the DPLL_ACT_STA register	Set ACT_N[i] for all enabled actions concerned: 0...11 for SMC=1 0...23 for SMC=0

Table 38.291 State description of the State Machine Table (4/7)

Step	Description	Comments
17	check the relation of the last increment to its predecessor according to the profile and taking into account TOV: set the ITN status bit and reset the corresponding LOCK bit, when not plausible; go to step 18, when no active *TRIGGER* appears for all following steps 18 to 20: go immediately back to step 1, when an active TRIGGER event occurs, interrupt all calculations there and reset all CAIP in that case; when going back to step 1: store TS_T in RAM 2b according to APT_2B; update RAM 2a and RAM 2d	for all conditions
18	wait for a new PMTR value; set the corresponding CAIPx values and go to step 19 in that case	go immediately to step 1 and update the RAM according to step 17 when an active *TRIGGER* event occurs
19	make the requested action calculation according to new PMTR values	go immediately to step 1 and update the RAM according to step 17 when an active *TRIGGER* event occurs
20	reset CAIPx and go back to step 18	go immediately to step 1 and update the RAM according to step 17 when an active *TRIGGER* event occurs
21	When DEN = 0 or SEN=0: make sure that the first active slope of STATE is detected; stay in step 1 until DEN=1, SEN=1 and at least one active *STATE* has been detected (FSD=1); the following steps are performed always (not necessarily in step 21, but also in steps 38 to 40 (when waiting for new PMTR values to be calculated): compare STATE_S with SSL (active slope); for each inactive slope: generate a SISI interrupt; send missing *STATE* INT when TS_S_CHECK time is reached and set MS=1 (missing *STATE* bits) in that case; do not leave step 21 while no active *STATE* appears. When an active *STATE* slope appears: store the actual position stamp at PSSM (value at the STATE event) update RAM by equation DPLL-6a-c (see chapter); **store the actual INC_CNT1/2 at MP1/MP2 respectively as missing pulses** *(instead of calculations in step 25)* store all relevant configuration bits **X** of the DPLL_CTRL(0,1) Registers in **shadow** registers and consider them for all corresponding calculations of steps 22 to 37 accordingly; the relevant bits are explained in the registers itself for FSD=0:	Depending on SSL, SEN, DEN step 21 is leaving with the next *STATE* input; for the steps 22-37: for SMC=1 replace: MLS1 by MLS2, LOCK1 by LOCK2; SUB_INC1 by SUB_INC2; CNT_NUM_1 by CNT_NUM_2; MPVAL1 by MPVAL2; EN_C1u by EN_C2u; **dir_crement** does mean: increment for DIR2=0 decrement for DIR2=1 or DIR1 respectively *) target number of pulses of the last increment (see step 25 ff.) **) add MPVAL1 or MPVAL2 only once, that means as long as PCM1 or PCM2 is set respectively ***) **SGE1_delay** is the value of SGE1 delayed by one active STATE event ****) **SGE2_delay** is the value of SGE2 delayed by one active STATE event *****) PD_S_store = 0 for AMS=0 (see DPLL_CTRL_0 register)

Table 38.291 State description of the State Machine Table (5/7)

Step	Description	Comments

	set PSSC=PSSM
	set FSD (first
	STATE detected)
	do not increment PSSC
	for (RMO=1 and SMC=0)
	and SGE1=1: increment INC_CNT1 by MLS1+MPVAL1**)
	for (RMO=1 and SMC=1)
	and SGE2=1: increment INC_CNT2 by MLS2+MPVAL2**)
	
	for SYS=0, FSD =1:
	dir_crement PSSC by NMB_S_TAR*) for (SMC=0 **and SGE1_delay*****)**=1**) or (SMC=1 and **SGE2_delay******)**=1**)
	increment INC_CNT1 by MLS1+MPVAL1
	) (for SMC=0, **SGE1=1 and RMO=1);
	increment INC_CNT2 by MLS2+MPVAL2
	) (for SMC=1, **SGE2=1 and RMO=1);
	dir_crement APS and APS_1C2
	for SYS=1:
	dir_crement APS and APS_1C3
	dir_crement APS_1C2 by SYN_S_OLD
	for RMO=1 and SMC=0: for
	SGE1_delay***)**=1 dir_crement** PSSC by NMB_S_TAR*);
	for **SGE1=1** increment INC_CNT1 by SYN_S*(MLS1 + PD_S_store) + MPVAL1**)
	for RMO=1 and SMC=1: for
	SGE2_delay****)**=1 dir_crement** PSSC by NMB_S_TAR*);
	for **SGE2=1** increment INC_CNT2 by SYN_S*(MLS2 + PD_S_store)*****) +MPVAL2**)
	within the DPLL_STATUS register:
	set LOCK1 or 2 bit accordingly;	

22	calculate TS_S according to equations DPLL-6a; calculate DT_S_ACT = TS_S - TS_S_OLD calculate RDT_S_ACT calculate QDT_SX	
23	send CDSI interrupt; calculate EDT_S and MEDT_S according to equations DPLL-8 and DPLL-9 for RMO=0: go back to step 21 for RMO=0 and update SYN_S and PD_S_store using the current ADT_S[i] values in that case;	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_S and PD_S_store are not updated until a new active STATE slope occurs.
24	calculate CDT_SX according to equation DPLL-10a and b;	only for RMO=1
25	calculate missing pulses - for TBU_CH1: MP1 = INC_CNT1(active STATE slope) - for TBU_CH2: MP2 = INC_CNT2(active STATE slope) calculate target number of pulses: NMB_S_TAR = (MLS1 + PD_S_store)*SYN_S + PD_S_store +MPVAL1 (for SMC=0) NMB_S_TAR = MLS2* (SYN_S + PD_S_store) + MPVAL2 (for SMC=1) (instead of PD_S_store use zero in the case AMS=0)	only for RMO=1 for SMC=0 instead of MPVAL1 use zero for PCM1=0 for SMC=1 instead of MPVAL2 use zero for PCM2=0; add MPVAL1/2 once to INC_CNT1/2 and reset PCM1/2 after that
26	sent MPx with highest possible frequency and set NMB_S = NMB_S_TAR	only for RMO=1, DMO=0 and COA=0
27	calculate number of pulses to be sent according to DPLL-22 or NMB_S = NMB_S_TAR + MPx	only for RMO=1, DMO=0 and COA=1
28	NMB_S = SYN_S*CNT_NUM_1 (SMC=0) NMB_S = SYN_S*CNT_NUM_2 (SMC=1)	only for RMO=1, DMO=1

Table 38.291 State description of the State Machine Table (6/7)

Step	Description	Comments
29	update SYN_S and PD_S_store;	**Note:** There are different behaviors of RM and HW-IP: For the HW-IP the values of SYN_S and PD_S_store are not updated until a new active STATE slope occurs.
30	calculate ADD_IN_CAL2 according to equation DPLL-26 and DPLL-26b or DPLL-31 respectively and store this value in RAM use ADD_IN_CAL2 as ADD_IN value for the case DLM=0 use ADD_IN_LD2 as ADD_IN for the case DLM=1, but do this update immediately (without waiting for this step 30); for RMO=1, DMO=DLM=0 and EN_C1u=0 (EN_C1u=0): reset the flip-flops in the SUB_INC1 or SUB_INC2 generator respectively; start sending SUB_INC1 / SUB_INC2;	only for RMO=1 for DLM=0 for DLM=1
31	calculate TS_S_CHECK = TS_S + DT_S _ACT * (SOV);	only for RMO=1;
32	automatic setting of actions masking bits in the DPLL_STATUS register: CAIP1 and CAIP2 for SMC=0 only CAIP2 for SMC=1	for RMO=1
33	for all actions with ACT_N[i]=0 calculate: NA[i] = (PSA[i] - PSSC)/MLS1 for forward direction with w = integer part and b = remainder of the division (fractional part) for backward direction use NA[i] = (PSSC - PSA[i])/(MLS1) and consider in both cases the time base overflow in order to get a positive difference use MLS2 as divider in the case of SMC=1	for SMC=0: 24 actions, for SMC=1: 12 actions; depending on ACT_N[i] in **DPLL_ACT_STA** register
34	calculate PDT_S[i] and DTA[i] for up to 24 action values according to equations DPLL-13 and DPLL-14;	only for RMO=1; for SMC=0 actions 0...23 for SMC=1 actions 12...23
35	calculate TSAC[i] according to equation DPLL-18 and PSAC[i] according to equation DPLL-20	for the relevant actions (see above) and RMO=1
36	automatic reset of the actions masking bit CAIP in the DPLL_STATUS register: CAIP1=CAIP2=0 for SMC=0 and only CAIP2=0 for SMC=1 set the corresponding ACT_N[i] bits in the DPLL_ACT_STA register	for the relevant actions (see above) and RMO=1 Set ACT_N[i] and reset ACT_WRi for all enabled actions
37	check the duration of the last increment to its predecessor according to the profile and taking into account SOV: set the ISN status bit and reset the corresponding LOCK bit, when not plausible; go to step 38, when no active *STATE* appears for all following steps 38 to 40: go immediately back to step 21, when an active *STATE* event occurs, interrupt all calculations there and reset all CAIPx in that case; when going back to step 21: store TS_S in RAM 1c2 according to APS_1C2; update RAM 1c1 and RAM 1c4	for all conditions

Table 38.291 State description of the State Machine Table (7/7)

Step	Description	Comments
38	wait for a new PMTR value; set the corresponding CAIPx values and go to step 39 in that case	go immediately to step 21 and update the RAM according to step 37 when an active *STATE* event occurs
39	make the requested action calculation according to new PMTR values	go immediately to step 21 and update the RAM according to step 37 when an active *STATE* event occurs
40	reset CAIP and go back to step 38	go immediately to step 21 and update the RAM according to step 37 when an active *STATE* event occurs

38.22.9 DPLL Interrupt signals

The DPLL provides 27 interrupt lines. These interrupts are shown below.

Table 38.292 DPLL Interrupt Signals

Signal	Description
DPLL_DCGI_IRQ	Direction change
DPLL_SORI_IRQ	*STATE* is out of range
DPLL_TORI_IRQ	*TRIGGER* is out of range
DPLL_CDSI_IRQ	*STATE* duration calculated for last increment
DPLL_CDTI_IRQ	*TRIGGER* duration calculated for last increment
DPLL_TE4_IRQ	*TRIGGER* event interrupt 4 request[3)]
DPLL_TE3_IRQ	*TRIGGER* event interrupt 3 request[3)]
DPLL_TE2_IRQ	*TRIGGER* event interrupt 2 request[3)]
DPLL_TE1_IRQ	*TRIGGER* event interrupt 1 request[3)]
DPLL_TE0_IRQ	*TRIGGER* event interrupt 0 request[3)]
DPLL_LL2_IRQ	Loss of lock interrupt for SUB_INC2 request
DPLL_GL2_IRQ	Get of lock interrupt for SUB_INC2 request
DPLL_E_IRQ	Error interrupt request
DPLL_LL1_IRQ	Loss of lock interrupt for SUB_INC1 request
DPLL_GL1_IRQ	Get of lock interrupt for SUB_INC1 request
DPLL_W1_IRQ	Write access to RAM region 1b or 1c interrupt request
DPLL_W2_IRQ	Write access to RAM region 2 interrupt request
DPLL_PW_IRQ	Plausibility window violation interrupt of *TRIGGER* request
DPLL_TAS_IRQ	*TRIGGER* active slope while NTI_CNT is zero interrupt request
DPLL_SAS_IRQ	*STATE* active slope interrupt request
DPLL_MT_IRQ	Missing *TRIGGER* interrupt request
DPLL_MS_IRQ	Missing *STATE* interrupt request
DPLL_TIS_IRQ	*TRIGGER* inactive slope interrupt request
DPLL_SIS_IRQ	*STATE* inactive slope interrupt request
DPLL_TAX_IRQ	*TRIGGER* maximum hold time violation interrupt request
DPLL_TIN_IRQ	*TRIGGER* minimum hold time violation interrupt request
DPLL_PE_IRQ	DPLL enable interrupt request
DPLL_PD_IRQ	DPLL disable interrupt request

NOTE

TEi_IRQ depends on the TINT value in ADT_T[i]1) and is only active when SYT2) =1.

1. see RAM region 2 explanations; see 19.14
2. see DPLL STATUS register; see 19.12.30
3. see TINT value in the corresponding ADT_T[i] section of RAM region 2; see 19.14.3

38.22.10 MCS to DPLL interface

A reduced AEI interface is implemented in the DPLL, which can only be accessed by the MCS Bus Master interface in the same cluster. The purpose of this interface is to enable a faster interchange of data between the MCS and the DPLL, while enabling a certain control over the DPLL internal state machine.

38.22.10.1 Architecture and organization

The implemented interface has an address width of 4 bits, while the size of the data interface is 24 bits.

The following table shows the implemented AEI addresses from the MCS side. Label RD refers to the label used for the address when reading from the MCS or writing from the DPLL, whereas Label WR refers to the label used for the address when writing from the MCS or reading from the DPLL.

38.22.10.2 General functionality

In order to have a better understanding of the implications when this interface is used, the following working concepts are informally defined here. They refer to the STATE engine operation when DPLL_CTRL_11.STATE_EXT is set.

Update of ram: Operation which stores TSF_S, DT_S_ACTUAL and RDT_S_ACTUAL back to the RAM and reads the profile. Calculation of sub-increments: Calculation of DT_S_ACTUAL, RDT_S_ACTUAL, NMB_S and ADD_IN. Change of direction: Update of profile and its increment or decrement (only in the STATE processor). Calculation of actions (PMT): Where the calculation of PSAC and TSAC is performed (only in state processor).

If DPLL_CTRL_11.STATE_EXT is not set, the DPLL will ignore the data written to this interface from the MCS. The DPLL will not update the interface either and a read done to this interface from the MCS can obtain out-of-date information.

If DPLL_CTRL_11.STATE_EXT is set, some modifications are done to the way that the DPLL module works when using the STATE engine. Up to 128 STATE events can be handled. RAM1c is not used anymore. Instead, the data needed to perform each of the already described operations is fetched from registers in this interface. The data that would have to be written back to RAM1c is also written to this interface. In each of the procedures described above, the DPLL will enter in one or more stalled states, in which it will wait for one or more words to be written to MCS2DPLL_DEB15(STATUS_INFO)

(1) Correspondence between STA_S values and their unlocking keywords

Operation	STA_S value	First Keyword	Second Keyword
Update of ram	00001_001_B	0xE	0x1
Calculation of sub-increments	00010_000_B	0xD	0x2
Calculation of actions	01110_000_B	0xC	0x3
Change of direction	00000_100_B	0xB	0x4
Change of direction	00000_110_B	0xB	0x4

The stalled STATE in the DPLL is freed by writing the upper keywords to MCS2DPLL_DEB15. Note the requirements on DPLL level (described in **Section 38.22.25, MCS to DPLL Configuration Registers Overview**) regarding the data on the interface that has to be written by the MCS program. If this data is incorrectly delivered or the STATE state machine is unlocked before delivery, the proper signal processing of the DPLL cannot be assured.

For the particular case of an update of ram after a virtual increment, the data field TSF_S is not calculated completely by the DPLL STATE processing unit. Instead, the values needed in order to fill this data field are provided (**Section 38.22.7.5, (6) Equations DPLL-6a5-7 to extend the time**

stamp values for STATE and **Section 38.22.7.5, (7) Equations DPLL-6a5-7 for backward direction**)

38.22.10.3 MCS to DPLL Register overview

Table 38.293 MCS to DPLL register overview

Address offset (see Section 38.28)	Common Label	Label RD	Label WR	Details in section
0x0	MCS2DPLL_DEB0	DT_S_P	DT_S_P1	
0x4	MCS2DPLL_DEB1	not used	RDT_S_P1	
0x8	MCS2DPLL_DEB2	TS_SX	RDT_S_PQ1	
0xC	MCS2DPLL_DEB3	DT_SX	DT_S_PQ	
0x10	MCS2DPLL_DEB4	SYN_S_OLD	RDT_S_PQ	
0x14	MCS2DPLL_DEB5	M_DW	DT_S_PQ1	
0x18	MCS2DPLL_DEB6	not used	ADT_S_P	
0x1C	MCS2DPLL_DEB7	RDT_S_P_RD	S_P_RD	
0x20	MCS2DPLL_DEB8	not used	TSF_S_P	
0x24	MCS2DPLL_DEB9	not used	TSF_S_P_MQ	
0x28	MCS2DPLL_DEB10	not used	TSF_S_P_PM_MQ	
0x2C	MCS2DPLL_DEB11	not used	TSF_S_P_PM	
0x30	MCS2DPLL_DEB12	not used	ADT_S_P1	
0x34	MCS2DPLL_DEB13	not used	not used	
0x38	MCS2DPLL_DEB14	not used	not used	
0x3C	MCS2DPLL_DEB15	not used	STATUS_INFO	

38.22.11 DPLL Register Memory overview

The available registers and the size of the RAM area 2 depends on the chosen device. Refer to **Section 38.28, GTM Device 358**.

NOTES

- The registers DPLL_ID_PMTR 24–31 are not available for all devices. Refer to **Section 38.28, GTM Device 358**.
- The values PSA24–31, DLA24–31, NA24–31 and DTA24–31 in RAM 1a are not available for all devices. Refer to **Section 38.28, GTM Device 358**.
- The values PDT_24 to PDT_31 in RAM1b are not available for all devices. Refer to **Section 38.28, GTM Device 358**.
- The registers DPLL_NUSC_EXT1, DPLL_NUSC_EXT2, DPLL_APS_EXT, DPLL_APS_1C3_EXT, DPLL_APS_SYNC_EXT, and DPLL_CTRL_EXT will return AEI_STATUS = b#10 if DPLL_CTRL_11.STATE_EXT is not set.
- For each of the regions, the maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

The description of registers is beginning at the register DPLL_CTRL_0.

The description of RAM regions is beginning at RAM 1a (see below): Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

The description of the memory region RAM 1a begins with memory element PSA[i]: The RAM region 1a is writable only for DEN=0 (see DPLL_CTRL_1 register).

The description of memory region RAM1b begins with memory element TS_T. The description of memory region RAM1c begins with memory element RDT_S.

The description of register region EXT begins with the register DPLL_TSAC[z]: This is an extension of the normal register region above in order to allow up to 32 action calculations and later specification modifications.

The description of the memory region RAM 2 begins with memory element RDT_T.

38.22.12 DPLL Configuration Registers Overview

DPLL contains following configuration registers:

Table 38.294 Register list (1/2)

Symbol	Register Name	Details in Section
DPLL_CTRL_0	Control Register 0	**38.22.18.1**
DPLL_CTRL_1	Control Register 1	**38.22.18.2**
DPLL_CTRL_2	DPLL Control Register 2 (actions 0-7 enable)	**38.22.18.3**
DPLL_CTRL_3	DPLL Control Register 3 (actions 8-15 enable)	**38.22.18.4**
DPLL_CTRL_4	DPLL Control Register 4 (actions 16-23 enable)	**38.22.18.5**
DPLL_CTRL_5	DPLL Control Register 5 (actions 24-31 enable)	**38.22.18.6**
DPLL_ACT_STA	Action Status Register including Shadow Register	**38.22.18.7**
DPLL_OSW	Offset and Switch old/new Address Register	**38.22.18.8**
DPLL_AOSV_2	Address Offset Register of RAM 2 Regions	**38.22.18.9**
DPLL_APT	DPLL Actual RAM pointer to RAM regions 2a, b and d	**38.22.18.10**
DPLL_APS	DPLL Actual RAM pointer to regions 1c1, 1c2 and 1c4	**38.22.18.11**
DPLL_APT_2C	Actual RAM Pointer Address for Region 2c	**38.22.18.12**
DPLL_APS_1C3	Actual RAM Pointer Address for RAM region 1c3	**38.22.18.13**
DPLL_NUTC	Number of Recent TRIGGER Events used for Calculations	**38.22.18.14**
DPLL_NUSC	Number of Recent STATE Events used for Calculations	**38.22.18.15**
DPLL_NTI_CNT	Number of Active TRIGGER Events to Interrupt	**38.22.18.16**
DPLL_IRQ_NOTIFY	Interrupt Register	**38.22.18.17**
DPLL_IRQ_EN	Interrupt Enable Register	**38.22.18.18**
DPLL_IRQ_FORCINT	Force Interrupt Register	**38.22.18.19**
DPLL_IRQ_MODE	Interrupt Request Mode	**38.22.18.20**
DPLL_EIRQ_EN	Error Interrupt Enable Register	**38.22.18.21**
DPLL_INC_CNT1	DPLL Counter for pulses for TBU_CH1_BASE to be sent in automatic end mode	**38.22.18.22**
DPLL_INC_CNT2	DPLL Counter for pulses for TBU_CH2_BASE to be sent in automatic end mode when SMC=RMO=1	**38.22.18.23**
DPLL_APT_SYNC	DPLL old RAM pointer and offset value for TRIGGER	**38.22.18.24**
DPLL_APS_SYNC	DPLL old RAM pointer and offset value for STATE	**38.22.18.25**
DPLL_TBU_TS0_T	DPLL TBU_CH0_BASE value at last TRIGGER event	**Note:**
DPLL_TBU_TS0_S	DPLL TBU_CH0_BASE value at last STATE event	**38.22.18.27**
DPLL_ADD_IN_LD1	DPLL direct load input value for SUB_INC1	**38.22.18.28**
DPLL_ADD_IN_LD2	DPLL direct load input value for SUB_INC2	**38.22.18.29**
DPLL_STATUS	Status Register	**38.22.18.30**
DPLL_ID_PMTR_[z]	DPLL 9 bit ID information for input signals PMT z 3)	**38.22.18.31**
DPLL_CTRL_0_SHADOW_TRIGGER	DPLL shadow register of DPLL_CTRL_0	**38.22.18.32**
DPLL_CTRL_0_SHADOW_STATE	DPLL shadow register of DPLL_CTRL_0	**38.22.18.33**
DPLL_CTRL_1_SHADOW_TRIGGER	DPLL shadow register of DPLL_CTRL_1	**38.22.18.34**

Table 38.294 Register list (2/2)

Symbol	Register Name	Details in Section
DPLL_CTRL_1_SHADOW_STATE	DPLL shadow register of DPLL_CTRL_1	**38.22.18.35**
DPLL_RAM_INI	Register to control the RAM Initialization	**38.22.18.36**
DPLL_TSAC[z]	Calculated Time Value to start Action z	**38.22.18.37**
DPLL_PSAC[z]	Calculated Position Value to start Action z	**38.22.18.38**
DPLL_ACB_[z]	Control Bits for up to 32 Actions	**38.22.18.39**
DPLL_CTRL_11	Control Register 11	**38.22.18.40**
DPLL_THVAL2	Measured last pulse time from active to inactive slope of TRIGGER after correction of input slope filter delays	**38.22.18.41**
DPLL_TIDEL	TRIGGER input delay	**38.22.18.42**
DPLL_SIDEL	STATE input delay	**38.22.18.43**
DPLL_CTN_MIN	CDT_T_NOM min value	**38.22.18.44**
DPLL_CTN_MAX	CDT_T_NOM max value	**38.22.18.45**
DPLL_CSN_MIN	CDT_S_NOM min value	**38.22.18.46**
DPLL_CSN_MAX	CDT_S_NOM max value	**38.22.18.47**
DPLL_STA	Status of the state machine states	**38.22.18.48**
DPLL_INCF1_OFFSET	Start value of the ADD_IN_ADDER1	**38.22.18.49**
DPLL_INCF2_OFFSET	Start value of the ADD_IN_ADDER2	**38.22.18.50**
DPLL_DT_T_START	Start value of DPLL_DT_T_ACT	**38.22.18.51**
DPLL_DT_S_START	Start value of DPLL_DT_S_ACT	**38.22.18.52**
DPLL_STA_MASK	Notify value for STA_S of register DPLL_STA.	**38.22.18.53**
DPLL_STA_FLAG	Flag according to DPLL_MASK.STA_NOTIFY	**38.22.18.54**
DPLL_INC_CNT1_MASK	Notify value for INC_CNT1 of register DPLL_INC_CNT1	**38.22.18.55**
DPLL_INC_CNT2_MASK	Notify value for INC_CNT2 of register DPLL_INC_CNT2	**38.22.18.56**
DPLL_NUSC_EXT1	Number of Recent STATE Events used for Calculations	**38.22.18.57**
DPLL_NUSC_EXT2	Number of Recent STATE Events used for Calculations	**38.22.18.58**
DPLL_APS_EXT	Actual RAM Pointer Address for STATE	**38.22.18.59**
DPLL_APS_1C3_EXT	Address pointer STATE for RAM region 1c3; Actual RAM pointer address value for ADT_S[i] Initial value: zero (00_H). Actual RAM pointer and synchronization	**38.22.18.60**
DPLL_APS_SYNC_EXT	STATE Time Stamp Field Offset at Synchronization Time	**38.22.18.61**
DPLL_CTRL_EXT	STATE Time Stamp Field Offset at Synchronization Time	**38.22.18.62**

Note 1. This register is not available for all devices. Refer to **Section 38.28, GTM Device 358**.

Note 2. The registers DPLL_ID_PMTR 24-31 are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.13 RAM Region 1a map description

Memory name	Description	Details in section
PSA[i] (i:0...NOAC-1)	Position/Value request for action i	
DLA[i] (i:0...NOAC-1)	Time to react before PSAi	
NA[i] (i:0...NOAC-1)	Number of TRIGGER/STATE increments to ACTION i	
DTA[i] (i:0...NOAC-1)	Calculated relative time to ACTION i	

Note: The values PSA24-31, DLA24-31, NA24-31 and DTA24-31 in RAM 1a are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.14 RAM Region 1b map description

Table 38.295 RAM region 1b (1/3)

Memory name	Description	Details in section
TS_T	Actual signal TRIGGER time stamp register TRIGGER_TS	
TS_T_OLD	Previous signal TRIGGER time stamp register TRIGGER_TS_OLD	
FTV_T	Actual signal TRIGGER filter value	
TS_S	Actual signal STATE time stamp register STATE_TS	
TS_S_OLD	Previous signal STATE time stamp register STATE_TS_OLD	
FTV_S	Actual signal STATE filter value	
THMI	TRIGGER hold time min. value	
THMA	TRIGGER hold time max. value	
THVAL	measured last pulse time from active to inactive TRIGGER slope	
TOV	Time out value of TRIGGER, according to the last nominal increment for a missing TRIGGER	
TOV_S	Time out value of STATE, according to the last nominal increment for a missing STATE	
ADD_IN_CAL1	calculated ADD_IN value for SUB_INC1 generation	
ADD_IN_CAL2	calculated ADD_IN value for SUB_INC2 generation	
MPVAL1	missing pulses to be added/subtracted directly to SUB_INC1 and INC_CNT1 once	
MPVAL2	missing pulses to be added/subtracted directly to SUB_INC2 and INC_CNT2 once	
NMB_T_TAR	target number of TRIGGER pulses	
NMB_T_TAR_OLD	target number of TRIGGER pulses	
NMB_S_TAR	target number of STATE pulses	
NMB_S_TAR_OLD	target number of STATE pulses	
RCDT_TX	reciprocal value of expected increment duration (T)	
RCDT_SX	reciprocal value of expected increment duration (S)	
RCDT_TX_NOM	reciprocal value of the expected nominal increment duration (T)	

Table 38.295 RAM region 1b (2/3)

Memory name	Description	Details in section
RCDT_SX_NOM	reciprocal value of the expected nominal increment duration (S)	
RDT_T_ACT	actual reciprocal value of TRIGGER	
RDT_S_ACT	actual reciprocal value of STATE	
DT_T_ACT	Duration of last TRIGGER increment	
DT_S_ACT	Duration of last STATE increment	
EDT_T	Absolute error of prediction for last TRIGGER increment	
MEDT_T	Average absolute error of prediction up to the last TRIGGER increment	
EDT_S	absolute error of prediction for last STATE increment	
MEDT_S	Average absolute error of prediction up to the last STATE increment	
CDT_TX	Expected duration of current TRIGGER increment	
CDT_SX	Expected duration of current STATE increment	
CDT_TX_NOM	Expected nominal duration of current TRIGGER increment (without consideration of missing events)	
CDT_SX_NOM	Expected nominal duration of current STATE increment (without consideration of missing events)	
TLR	TRIGGER locking range value; the TOR bit in the DPLL_STATUS register is set when violated	
SLR	STATE locking range value; the SOR bit is set when violated	
PDT_[i] (i:0...NOAC-1)	predicted time to ACTION i	
MLS1	Calculated number of sub-pulses between two STATE events (to be set by CPU)	
MLS2	Calculated number of sub-pulses between two STATE events (to be set by CPU) for the use when SMC=RMO=1	
CNT_NUM_1	number of sub-pulses of SUB_INC1 in continuous mode, updated by the host only	
CNT_NUM_2	number of sub-pulses of SUB_INC2 in continuous mode, updated by the host only	
PVT	Plausibility value of next active TRIGGER slope	
PSTC	Accurate calculated position stamp of last TRIGGER input;	
PSSC	Accurate calculated position stamp of last STATE input;	
PSTM	Measured position stamp at last active TRIGGER input	
PSTM_OLD	Measured position stamp at last but one active TRIGGER input	
PSSM	Measured position stamp at last active STATE input	

Table 38.295 RAM region 1b (3/3)

Memory name	Description	Details in section
PSSM_OLD	Measured position stamp at last but one active STATE input	
NMB_T	Number of pulses of current increment in normal mode for SUB_INC1 (see equation DPLL-21 or for SMC=1 equation DPLL-27 respectively)	
NMB_S	Number of pulses of current increment in emergency mod for SUB_INC1 (see equation DPLL-22) or in the case SMC=1 for SUB_INC2 (see equation DPLL-28)	

Note: The values PDT_24 to PDT_31 in RAM1b are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.15 RAM Region 1c map description

Memory name	Description	Details in section
RDT_S[i] (i:0...63)	Part of RAM1c1. Reciprocal value of the corresponding successive increment i, for each true nominal increment.	
TSF_S[i] (i:0...63)	Part of RAM1c2. Time stamp field for state events, for each true nominal increment plus each virtual increment.	
ADT_S[i] (i:0...63)	Part of RAM1c3. Adapt values for the current STATE increment, for each true nominal increment.	
DT_S[i] (i:0...63)	Part of RAM1c4. Uncorrected last increment value of STATE for full scale, for each true nominal increment.	

38.22.16 Register Region EXT description

Register name	Description	Details in
DPLL_TSAC[z] (z:0...NOAC-1)	DPLL calculated action time stamps for action z	
DPLL_PSAC[z] (z:0...NOAC-1)	DPLL calculated action position stamps for action z	
DPLL_ACB_[z] (z:0...(NOAC/4)-1)	DPLL control bits for actions ((4*z)...(4*z)+3)	
DPLL_CTRL_11	DPLL control register	
DPLL_THVAL2	DPLL immediate THVAL value	
DPLL_TIDEL	DPLL additional TRIGGER input delay	
DPLL_SIDEL	DPLL additional STATE input delay	
DPLL_CTN_MIN	CDT_T_NOM minimum value	
DPLL_CTN_MAX	CDT_T_NOM maximum value	
DPLL_CSN_MIN	CDT_S_NOM minimum value	
DPLL_CSN_MAX	CDT_S_NOM maximum value	
DPLL_STA	DPLL state machine status information	
DPLL_INCF1_OFFSET	DPLL ADD_IN_ADDER1 offset for fast pulse generation	
DPLL_INCF2_OFFSET	DPLL ADD_IN_ADDER2 offset for fast pulse generation	
DPLL_DT_T_START	DPLL first value of DPLL_DT_T_ACT for the first increment after setting SIP1 from 0 to 1.	
DPLL_DT_S_START	DPLL first value of DPLL_DT_S_ACT for the first increment after setting SIP2 from 0 to 1.	
DPLL_STA_MASK	DPLL trigger masks for signals DPLL_STA_T and DPLL_STA_S	
DPLL_STA_FLAG	DPLL STA_T/S and INC_CNT1/2 flags	
DPLL_INC_CNT1_MASK	DPLL INC_CNT1 trigger mask	
DPLL_INC_CNT2_MASK	DPLL INC_CNT2 trigger mask	
DPLL_NUSC_EXT1	Extension register number 1 for DPLL_NUSC [4)]	
DPLL_NUSC_EXT2	Extension register number 2 for DPLL_NUSC [4)]	
DPLL_APS_EXT	Extension register for DPLL_APS [4)]	
DPLL_APS_1C3_EXT	Extension register for DPLL_APS_1C3 [4)]	
DPLL_APS_SYNC_EXT	Extension register for DPLL_APS_SYNC [4)]	
DPLL_CTRL_EXT	Extension register for DPLL_CTRL [4)]	

38.22.17 RAM Region 2 map description

Memory name	Description	Details in section
RDT_T[i] (i:0...AOSV_2B/4-1)	Region 2a. Reciprocal value of the corresponding successive increment i, for each true nominal increment.	
TSF_T[i] (i:0...AOSV_2B/4-1)	Region 2b. Time Stamp Field for TRIGGER event i, for each true nominal increment plus each virtual increment.	
ADT_T[i] (i:0...AOSV_2B/4-1)	Region 2c. Adapt values for the current TRIGGER increment i, for each true nominal increment.	
DT_T[i] (i:0...AOSV_2B/4-1)	Region 2d. Uncorrected last increment value of TRIGGER i, for each true nominal increment.	

Note: Note: For each of the regions, the maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

The description of registers is beginning at the register DPLL_CTRL_0.

The description of RAM regions is beginning at RAM 1a (see below):

Bits 31 down to 24 in each RAM region are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Reserved address regions are not protected against writing.

The description of the memory region RAM 1a begins with memory element PSA[i]:

The RAM region 1a is writeable only for DEN=0 (see DPLL_CTRL_1 register).

The description of memory region RAM1b begins with memory element TS_T.

The description of memory region RAM1c begins with memory element RDT_S.

The description of register region EXT begins with the register DPLL_TSAC[z]:

This is an extension of the normal register region above in order to allow up to 32 action calculations and later specification modifications.

The description of the memory region RAM 2 begins with memory element RDT_T.

38.22.18 DPLL Register description

38.22.18.1 DPLL_CTRL_0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28000_H

Value after reset: $003B\ BA57_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMO	TEN	SEN	IDT	IDS	AMT	AMS	TNU								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SNU					IFP	MLT									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.296 DPLL_CTRL_0 Register Contents (1/3)

Bit Position	Bit Name	Function
31	RMO *1*2	Reference mode; selection of the relevant the input signal for generation of SUB_INC1. See notes {REF:DPLL_5764}, {REF:DPLL_10978}. 0: Normal mode; the signal TRIGGER is used to generate the SUB_INC1 signals 1: Emergency mode for SMC=0; signal STATE is used to generate the SUB_INC1 signals; Double synchronous mode for SMC=1: signal TRIGGER is used to generate the SUB_INC1 signals and STATE is used to generate the SUB_INC2 signals **NOTE** For SMC=0: TRIGGER and STATE are prepared to calculate SUB_INC1. The RMO bit gives a decision only, which of them is used. For changing from normal mode to emergency mode at the following STATE slope (according to the RMO value in the shadow register, see note (DPLL_5764)*[XREF TARGET " (DPLL_5764)] NOT EXIST*) the PSSC value is calculated by PSSC = PSTC. For changing from emergency mode to normal mode at the following TRIGGER slope (according to the RMO value in the shadow register, see note (DPLL_10978)*[XREF TARGET " (DPLL_10978)] NOT EXIST*) the PSTC value is calculated by PSTC = PSSC. In case of no further TRIGGER or STATE events the CPU has to perform the above corrections.
30	TEN	TRIGGER enable. 0: TRIGGER signal is not enabled (no signal considered) 1: TRIGGER signal is enabled
29	SEN	STATE enable. 0: STATE signal is not enabled (no signal considered) 1: STATE signal is enabled
28	IDT *1	Input delay TRIGGER; use of input delay information transmitted in FT part of the TRIGGER signal. See note {REF:DPLL_5764}. 0: Delay information is not used 1: Up to 24 bits of the FT part contain the delay value of the input signal, concerning the corresponding edge

Table 38.296 DPLL_CTRL_0 Register Contents (2/3)

Bit Position	Bit Name	Function
27	IDS *2	Input delay STATE; Use of input delay information transmitted in FT part of the STATE signal. See note {REF:DPLL_10978}. 0: Delay information is not used 1: Up to 24 bits of the FT part contain the delay value of the input signal, concerning the corresponding edge
36	AMT *2	Adapt mode TRIGGER; Use of adaptation information of TRIGGER. See note {REF:DPLL_5764}. 0: No adaptation information for TRIGGER is used 1: Immediate adapting mode; the values for physical deviation PD of ADT_T[i] are considered to calculate the SUB_INC1 pulses in normal mode and for SMC=1
25	AMS	Adapt mode STATE; Use of adaptation information of STATE. See note {REF:DPLL_10978}. 0: No adaptation information is used for STATE 1: Immediate adapting mode; the values for physical deviation PD_S of ADT_S[i] are considered to calculate SUB_INC1 pulses in emergency mode (SMC=0) or SUB_INC2 pulses for SMC=1
24 to 16	TNU*3	TRIGGER number; TNU+1 is number of nominal TRIGGER events in HALF_SCALE (1 to 512). See note {REF:DPLL_10979}. **NOTE** The number of nominal TRIGGER events is the decimal value plus 1. This value can only be written when (RMO=1 and SMC=0) or DEN=0. To make sure that this signal is not changed during a mode change RMO=0 means that the status of RMO=0 must be given before and during writing to the register. Set TSL=00 before changing this value and set RMO=0 only after FULL_SCALE with TSL>0.
15 to 11	SUM*3	STATE number; SNU+1 is number of nominal STATE events in HALF_SCALE (1 to 32). See note {REF:DPLL_10979}. **NOTES** 1. The number of nominal STATE events is the decimal value plus 1. This value can only be written when (RMO=0 and SMC=0) or DEN=0. To make sure that this signal is not changed during a mode change RMO=0 means that the status of RMO=0 must be given before and during writing to the register. Set SSL=00 before changing this value and set RMO=1 only after FULL_SCALE with SSL>0. 2. This register can only be written when DPLL_CTRL_11.STATE_EXT is not set. If DPLL_CTRL_11.STATE_EXT is set, the signal cannot be written, the read value is zero.
10	IFP *1*2*4	Input filter position; value contains position or time related information. See notes {REF:DPLL_5764}, {REF:DPLL_10978}, {REF:DPLL_10980}. 0: TRIGGER_FT and STATE_FT mean time related values, that means the number of time stamp clocks 1: TRIGGER_FT and STATE_FT mean position related values, that means the number of SUB_INC1 (or SUB_INC2 in the case SMC=1) pulses respectively

Table 38.296 DPLL_CTRL_0 Register Contents (3/3)

Bit Position	Bit Name	Function
9 to 0	MLT *1	Multiplier for TRIGGER; MLT+1 is number of SUB_INC1 pulses between two TRIGGER events in normal mode (1 to 1024); See note {REF:DPLL_5764} **NOTES** • For emergency mode the number of SUB_INC1 pulses between two STATE events is calculated by the CPU using the formula MLS1=(MLT+1)* (TNU+1) / (SNU+1) in order to get the same number of SUB_INC1 pulses for FULL_SCALE. This value is stored in RAM at $05C0_H$. Change of MLT by the CPU must result in the corresponding change of MLS1 by the CPU for SMC=0. • The number of MLT events is the binary value plus 1. The value MLT+1 is replaced by MLS1 in the case of SMC=1 (see DPLL_CTRL_1 register) for all relevant calculations.

Note 1. Stored in an independent shadow register for an active TRIGGER event and for DEN = 1.

Note 2. Stored in an independent shadow register for an active STATE event and for DEN = 1.

Note 3. The time between two active STATE or TRIGGER events must be always greater then 23,4 µs; in addition the TS_CLK and the resolution must be chosen such that for each nominal increment the time stamps at the beginning and the end of the increment differ at least in the value of 257

Note 4. For IFP=1 the time between two active TRIGGER or STATE events must be always greater then 2,34 ms and the value x of MLT, MLS1 or MLS2 must be chosen such that the number of time stamp pulses between two SUB_INC events must be less than 65536. This is fulfilled when x is greater than 256.

38.22.18.2 DPLL_CTRL_1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSL		SSL		SMC	TS0_HRT	TS0_HRS	SYSF	SWR	LCD	SYN_NT					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SYN_NS					PCM2	DLM2	SGE2	PCM1	DLM1	SGE1	PIT	COA	IDDS	DEN	DMO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.297 DPLL_CTRL_1 Register Contents (1/5)

Bit Position	Bit Name	Function
31, 30	TSL	TRIGGER slope select; Definition of active slope for signal TRIGGER each active slope is an event defined by TNU. Set by DEN=0 only. No slope of TRIGGER will be used; 00_B: No slope of TRIGGER will be used; this value makes only sense in emergency mode 01_B: Low high slope will be used as active slope, only inputs with a signal value of "1" will be considered 10_B: High low slope will be used as active slope, only inputs with a signal value of "0" will be considered 11_B: Both slopes will be used as active slopes If DPLL_STATUS.FTD = '0' "level sensitive for first TRIGGER input signal edge" No input signal of TRIGGER will be used; 00_B: No input signal of TRIGGER will be used; this value makes only sense in normal mode 01_B: "high" input signal level will be used as active slope, only inputs with a signal value of "1" will be considered 10_B: "low" input signal level will be used as active slope, only inputs with a signal value of "0" will be considered 11_B: Both input signal levels will be used as active slopes **NOTES** 1. This value can only be written when (RMO=1 and SMC=0) or DEN=0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register. 2. Stored in an independent shadow register for an active TRIGGER event and for DEN = 1. 3. Stored in an independent shadow register for an active STATE event and for DEN = 1. 4. Bit is cleared, when transmitted to shadow register

Table 38.297 DPLL_CTRL_1 Register Contents (2/5)

Bit Position	Bit Name	Function
29, 28	SSL	STATE slope select; Definition of active slope for signal STATE each active slope is an event defined by SNU. Set by DEN=0 only. If DPLL_STATUS.FSD = '1' "slope sensitive after detection of first STATE input signal" No slope of STATE will be used; 00_B: No slope of STATE will be used; this value makes only sense in normal mode 01_B: Low high slope will be used as active slope, only inputs with a signal value of "1" will be considered 10_B: High low slope will be used as active slope, only inputs with a signal value of "0" will be considered 11_B: Both slopes will be used as active slopes If DPLL_STATUS.FSD = '0' "level sensitive for first STATE input signal edge" No input signal of STATE will be used; 00_B: No input signal of STATE will be used; this value makes only sense in normal mode 01_B: "high" input signal level will be used as active slope, only inputs with a signal value of "1" will be considered 10_B: "low" input signal level will be used as active slope, only inputs with a signal value of "0" will be considered 11_B: Both input signal levels will be used as active slopes **NOTE** This value can only be written when (RMO=0 and SMC=0) or DEN=0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register.
27	SMC	Synchronous Motor Control 0: TRIGGER and STATE inputs are used for a control different to SMC. 1: The TRIGGER input reflects a combined sensor signal for SMC and in the case of RMO=1 also STATE reflects a different combined sensor signal **NOTE** This bit can only be written when the DPLL is disabled.
26	TS0_HRT	Time stamp high resolution TRIGGER 0: The resolution of the used DPLL input TBU_TS0 bits is equal to the TRIGGER input time stamp resolution 1: The TRIGGER input time stamps have a 8 times higher resolution as the TBU_TS0 input **NOTE** This bit can only be written when the DPLL is disabled.
25	TS0_HRS	Time stamp high resolution STATE 0: The resolution of the used DPLL input TBU_TS0 bits is equal to the STATE input time stamp resolution 1: The STATE input time stamps have a 8 times higher resolution as the TBU_TS0 DPLL input **NOTE** This bit can only be written when the DPLL is disabled.

Table 38.297 DPLL_CTRL_1 Register Contents (3/5)

Bit Position	Bit Name	Function
24	SYSF	SYN_NS for FULL_SCALE 0: The SYN_NS value is valid for HALF_SCALE 1: The SYN_NS value is valid for FULL_SCALE The value SYN_NS does mean the sum of all systematic missing STATE events in HALF_SCALE (for SYSF=0) or FULL SCALE (for SYSF=1). **NOTE** This value can only be written when (RMO=0 and SMC=0) or DEN=0. Set SSL=00 before changing this value and set RMO=1 only after FULL_SCALE with SSL>0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register.
23	SWR	Software reset 0: No software reset enabled 1: Software reset enabled Resets all register and internal states of the DPLL **NOTE** Setting the SWR bit results only in a software reset when the DPLL is not enabled (DEN=0).
22	LCD	Locking condition definition 0: Locking condition definition is one times missing TRIGGERs as expected by the profile in HALF_SCALE (one gap). 1: Locking condition definition is n–1 times missing TRIGGERs as expected by the profile in HALF_SCALE (one additional tooth) **NOTE** This bit can only be written when the DPLL is disabled and be fixed to zero, when not needed for an implementation.
21 to 16	SYN_NT	Synchronization number of TRIGGER; summarized number of virtual increments in HALF_SCALE. Sum of all systematic missing TRIGGER events in HALF_SCALE; the SYN_NT missing TRIGGER can be divided up to an arbitrary number of blocks. The pattern of events and missing events in FULL_SCALE is shown in RAM region 2c as value NT in addition to the adapted values. The number of stored increments in FULL_SCALE must be equal to 2*(TNU–SYN_NT). This pattern is written by the CPU beginning from a fixed reference point (maybe beginning of the FULL_SCALE region). The relation to the actual increment is established by setting of the profile RAM pointer APT_2C in an appropriate relation to the RAM pointer APT of the actual increment by the CPU. **NOTE** This value can only be written when (RMO=1 and SMC=0) or DEN=0. Set TSL=00 before changing this value and set RMO=0 only after FULL_SCALE with TSL>0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register.

Table 38.297 DPLL_CTRL_1 Register Contents (4/5)

Bit Position	Bit Name	Function
15 to 11	SYN_NS	Synchronization number of STATE; summarized number of virtual increments in HALF_SCALE. Sum of all systematic missing STATE events in HALF_SCALE (for SYSF=0) or FULL SCALE (for SYSF=1); the SYN_NS missing STATES can be divided up to an arbitrary number of blocks. The pattern of events and missing events in FULL_SCALE is shown in RAM region 1c3 as value NS in addition to the adapted values. The number of stored increments in FULL_SCALE must be equal to 2*(SNU+1–SYN_NS) for SYSF=0 or 2*(SNU+1)–SYN_NS for SYSF=1. This pattern is written by the CPU beginning from a fixed reference point (maybe beginning of the FULL_SCALE region). The relation to the actual increment is established by setting of the profile RAM pointer APS_1C3 in an appropriate relation to the RAM pointer APS of the actual increment by the CPU. **NOTES** • This value can only be written when (RMO=0 and SMC=0) or DEN=0. Set SSL=00 before changing this value and set RMO=1 only after FULL_SCALE with SSL>0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register. • This register can only be written when DPLL_CTRL_11.STATE_EXT is not set. If DPLL_CTRL_11.STATE_EXT is set, the signal cannot be written, the read value is zero.
10	PCM*2, *3	Pulse Correction Mode for SUB_INC2 generation. See notes {REF:DPLL_10981}, {REF:DPLL_10982}. 0: The DPLL does not use the correction value stored in MPVAL2. 1: The DPLL uses the correction value stored in MPVAL2
9	DLM*2	Direct Load Mode for SUB_INC2 generation. See note {REF:DPLL_10981}. 0: The DPLL uses the calculated ADD_IN_CAL value for the SUB_INC2 generation. 1: The ADD_IN_LD value is used for the SUB_INC2 generation and is provided by the CPU; the value remains valid until the CPU writes a new one; the calculated ADD_IN values are stored as ADD_IN_CAL in the RAM at different locations for normal and emergency mode
8	SGE*2	SUB_INC2 generator enable. See note {REF:DPLL_10981}. 0: The SUB_INC2 generator is not enabled. 1: The SUB_INC2 generator is enabled
7	PCM*1, *2, *3	Pulse Correction Mode for SUB_INC1 generation. See notes {REF:DPLL_5804},{REF:DPLL_10981}, {REF:DPLL_10982}. 0: The DPLL does not use the correction value stored in MPVAL1 1: The DPLL uses the correction value stored in MPVAL1 in normal and emergency mode
6	DLM*1, *2	Direct Load Mode for SUB_INC1 generation. See notes {REF:DPLL_5804},{REF:DPLL_10981}. 0: The DPLL uses the calculated ADD_IN_CAL value for the SUB_INC1 generation 1: The ADD_IN_LD value is used for the SUB_INC1 generation and is provided by the CPU; the value remains valid until the CPU writes a new one; the calculated ADD_IN values are stored as ADD_IN_CAL in the RAM at different locations for normal and emergency mode
5	SGE1*1, *2	SUB_INC1 generator enable. See notes {REF:DPLL_5804},{REF:DPLL_10981}. 0: The SUB_INC1 generator is not enabled 1: The SUB_INC1 generator is enabled

Table 38.297 DPLL_CTRL_1 Register Contents (5/5)

Bit Position	Bit Name	Function
4	PIT*1	Plausibility value PVT to next active TRIGGER is time related. See note {REF:DPLL_5804}. 0: The plausibility value is position related (PVT contains the number of SUB_INC1 pulses) 1: The plausibility value is time related (the PVT value is to be multiplied with the duration of the last increment DT_T_ACT and divided by 1024)
3	COA*1, *2	Correction strategy in automatic end mode (DMO=0). See notes {REF:DPLL_5804},{REF:DPLL_10981}. 0: The pulse frequency of the CMU_CLK0 will be used to make up for missing pulses from last increment; the output of the calculated new pulses will start after resetting the FFs in the pulse generation unit. The frequency of CMU_CLK0 should not exceed half the frequency of the system clock (see {REF:DPLL_9427}) 1: Missing pulses of the last increment are distributed evenly to the next increment, calculations are done when the next active input event appears. The number of missing sub-pulses will be determined by the pulse counter difference between the last two active TRIGGER / STATE events respectively; the FFs in the pulse generation unit are not reset before sending new pulses. **NOTE** For SMC=RMO=1: COA is used for SUB_INC1 and SUB_INC2
2	IDDS	Input direction detection strategy in the case of SMC=0. 0: The input direction is detected comparing the THMI value with the duration between active and inactive slope of TRIGGER. 1: The input direction is detected using TDIR input signal also in the case SMC=0. **NOTE** This bit can only be written when the DPLL is disabled and be fixed to zero, when not needed for an implementation. Independent of the value of IDDS is the direction information for TRIGGER in the case SMC=0 always considered at the moment when the inactive slope appears.
1	DEN	DPLL enable. 0: The DPLL is not enabled; Disabling the DPLL will result in a reset state of the DPLL_STATUS register which remains in this state until DEN=1. No DPLL related interrupt will be generated in that case. 1: The DPLL is enabled. **NOTE** The bits 31 down to 0 of the DPLL_STATUS register are cleared, when the DPLL is disabled. Some bits of the control registers can be set only when DEN=0. The protected bits in the DPLL_CTRL_1 register cannot be written when simultaneously DEN is set to 1.
0	DMO*1, *2	DPLL mode select. See notes {REF:DPLL_5804},{REF:DPLL_10981}. 0: Automatic end mode; if the number of pulses for an increment is reached, no further pulse is generated until the next active TRIGGER / STATE is received; in the case of getting a new active TRIGGER / STATE before the defined number of pulses is reached, the pulse frequency is changed according to the conditions described below (COA). 1: Continuous mode; in this mode a difference between the predefined number of pulses and the actual number of generated pulses can influence the pulse frequency by writing a corresponding pulse number into CNT_NUM_1 or CNT_NUM_2 respectively in RAM region 1b.

Note: This value can only be written when (RMO=1 and SMC=0) or DEN=0. To make sure that this signal is not changed during a mode change SMC=0 means that the status of SMC=0 must be given before and during writing to the register.

Note 1. Stored in an independent shadow register for an active TRIGGER event and for DEN = 1.
Note 2. Stored in an independent shadow register for an active STATE event and for DEN = 1.
Note 3. Bit is cleared, when transmitted to shadow register

38.22.18.3 DPLL_CTRL_2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WAD7	WAD6	WAD5	WAD4	WAD3	WAD2	WAD1	WAD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN7	AEN6	AEN5	AEN4	AEN3	AEN2	AEN1	AEN0	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.298 DPLL_CTRL_2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	WAD[7:0]	Write control bit of Action_[7:0]. 0: The corresponding AENi bit is not writable 1: The corresponding AENi bit is writable
15 to 8	AEN*[1][7:0]	ACTION_[7:0] enable. 0: The corresponding action is not enabled 1: The corresponding action is enabled
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. This bit can be written only if the correspondent WADi Bit is set in the same access. It can be set for debug purposes by CPU also, when DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Note: For WADi =1 only the corresponding AENi bits are writable. The AENi bits remain unchanged when the corresponding WADi=0.

38.22.18.4 DPLL_CTRL_3

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2800C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WAD15	WAD14	WAD13	WAD12	WAD11	WAD10	WAD9	WAD8
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN15	AEN14	AEN13	AEN12	AEN11	AEN10	AEN9	AEN8	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.299 DPLL_CTRL_3 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	WAD[15:8]	Write control bit of Action_[15:8]. 0: The corresponding AENi bit is not writable 1: The corresponding AENi bit is writable
15 to 8	AEN*[1][15:8]	ACTION_[15:8] enable. 0: The corresponding action is not enabled 1: The corresponding action is enabled
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. This bit can be written only if the correspondent WADi Bit is set in the same access. It can be set for debug purposes by CPU also, when DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Note: For WADi =1 only the corresponding AENi bits are writable. The AENi bits remain unchanged when the corresponding WADi=0.

38.22.18.5 DPLL_CTRL_4

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WAD23	WAD22	WAD21	WAD20	WAD19	WAD18	WAD17	WAD16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN23	AEN22	AEN21	AEN20	AEN19	AEN18	AEN17	AEN16	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.300 DPLL_CTRL_4 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	WAD[23:16]	Write control bit of Action_[23:16]. 0: The corresponding AENi bit is not writable 1: The corresponding AENi bit is writable
15 to 8	AEN*[1][23:16]	ACTION_[23:16] enable. 0: The corresponding action is not enabled 1: The corresponding action is enabled
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. This bit can be written only if the correspondent WADi Bit is set in the same access. It can be set for debug purposes by CPU also, when DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Note: For WADi =1 only the corresponding AENi bits are writable. The AENi bits remain unchanged when the corresponding WADi=0.

38.22.18.6 DPLL_CTRL_5

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WAD31	WAD30	WAD29	WAD28	WAD27	WAD26	WAD25	WAD24
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEN31	AEN30	AEN29	AEN28	AEN27	AEN26	AEN25	AEN24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.301 DPLL_CTRL_5 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	WAD[31:24]	Write control bit of Action_[31:24]. 0: The corresponding AENi bit is not writable 1: The corresponding AENi bit is writable
15 to 8	AEN*[1][31:24]	ACTION_[31:24] enable. 0: The corresponding action is not enabled 1: The corresponding action is enabled
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. This bit can be written only if the correspondent WADi Bit is set in the same access. It can be set for debug purposes by CPU also, when DPLL is disabled. The enable bit becomes active only when the DPLL is in operation (DEN=1).

Note: For WADi =1 only the corresponding AENi bits are writable. The AENi bits remain unchanged when the corresponding WADi=0

38.22.18.7 DPLL_ACT_STA

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ACT_N															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACT_N															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.302 DPLL_ACT_STA Register Contents

Bit Position	Bit Name	Function
31 to 0	ACT_N	New output data values concerning to action i provided 0: No new output data available after a recent PMT request or actual event value is in the past or invalid. 1: Either new PMTR data received or calculation is repeated to be more precise by taking into account new TRIGGER or STATE values **NOTES** 1. ACT_N[i] is set (for AEN[k]=1 and a new valid PMTR), that means when new action data are to be calculated for the correspondent action. After each calculation of the new actions values the ACT_N[i] bit updates the corresponding bit in the connected shadow register. The status of the ACT_N[i] bits in the shadow register is reflected by the corresponding DPLL output signal ACT_V (valid bit). reset together with the corresponding shadow register bit for AEN[k]=0; reset without the corresponding shadow register bit when the calculated event is in the past (the shadow register bit is set, when it was not set before in that case) the corresponding shadow register bit is reset, when new PMTR data are written or when the provided action data are read (blocking read) writable for debugging purposes together with the corresponding shadow register when DEN=0 2. These bits can only be written for test purposes when the DPLL is disabled.

38.22.18.8 DPLL_OSW

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2801C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OSS		—	—	—	—	—	—	SWON_T	SWON_S
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.303 DPLL_OSW Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9, 8	OSS	Offset size of RAM region 2 00_B: Offset size 128 of RAM region 2. 01_B: Offset size 256 of RAM region 2. 10_B: Offset size 512 of RAM region 2. 11_B: Offset size 1024 of RAM region 2. **NOTES** 1. At least 128 and at most 1024 values can be stored in each of the RAM 2 regions a to d accordingly. The value can be set only for DEN=0. The change of the OSS value results in an automatic change of the offset values in the DPLL_AOSV_2 register 2. This value can only be written when the DPLL is disabled.
7 to 2	Reserved	When read, the value after reset is returned.
1	SWON_T	Switch of new TRIGGER; Switch bit for LSB address of TRIGGER. This bit is changed for each write access to TS_T/TS_T_OLD. Using this unchanged address bit SWON_T for any access to TS_T results always in an access to TS_T_OLD. For writing to this address the former old (TS_T_OLD_old) value is overwritten by the new one while the SWON_T bit changes. Thus the former new one is now the old one and the next access is after changing SWON_T directed to this place. Therefore write to TS_T first and after that immediately to FTV_T and PSTM, always before a new TS_T value is to be written. **NOTE** After writing TS_T, FTV_T and PSTM in this order the address pointer AP with LSB(AP)=SWON_T shows for the corresponding address to TS_T_OLD, FTV_T and PSTM while LSB(AP)=/SWON_T results in an access to TS_T, FTV_T_old and PSTM_OLD respectively. The value can be read only. This bit is reset when disabling the DPLL (DEN=0).

Table 38.303 DPLL_OSW Register Contents (2/2)

Bit Position	Bit Name	Function
0	SWON_S	Switch of new STATE; Switch bit for LSB address of STATE. This bit is changed for each write access to TS_S/TS_S_OLD. Using this unchanged address bit SWON_S for any access to TS_S results always in an access to TS_S_OLD. For writing to this address the former old (TS_S_OLD_old) value is overwritten by the new one while the SWON_S bit changes. Thus the former new one is now the old one and the next access is after changing SWON_S directed to this place. Therefore write to TS_S first and after that immediately to FTV_S and PSSM, always before a new TS_S value is to be written. **NOTE** After writing TS_S, FTV_S and PSSM in this order the address pointer AP with LSB(AP)=SWON_S shows for the corresponding address to TS_S_OLD, FTV_S and PSSM while LSB(AP)=/SWON_S results in an access to TS_S, FTV_S_old and PSSM_OLD respectively. The value can be read only. This bit is reset when disabling the DPLL (DEN=0).

38.22.18.9 DPLL_AOSV_2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 28020_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AOSV_2D								AOSV_2C							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AOSV_2B								AOSV_2A							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.304 DPLL_AOSV_2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	AOSV_2D	Address offset value of the RAM 2D region. OS– 0x0: AOSV_2D= 0x06 OS– 0x1: AOSV_2D= 0x0C OS– 0x2: AOSV_2D= 0x18 OS– 0x3: AOSV_2D= 0x30 The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2d. When the APT value is added to this start address, the current RAM cell DT_Tx is addressed. value is set automatically when OSS in the PPLL_OSW register is set: **NOTES** The offset values are needed to support a scalable RAM size of region 2 from 1,5 Kbytes to 12 Kbytes. The values above must be in correlation with the offset size defined in the OSW register. All offset values are set automatically in accordance to the OSS value in the DPLL_OSW register. This value can be set only for DEN=0.
23 to 16	AOSV_2C	Address offset value of the RAM 2C region. OS– 0x0: AOSV_2C= 0x04 OS– 0x1: AOSV_2C= 0x08 OS– 0x2: AOSV_2C= 0x10 OS– 0x3: AOSV_2C= 0x20 The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2c. When the APT value is added to this start address, the current RAM cell ADT_Tx is addressed. value is set automatically when OSS in the PPLL_OSW register is set:

Table 38.304 DPLL_AOSV_2 Register Contents (2/2)

Bit Position	Bit Name	Function
15 to 8	AOSV_2B	Address offset value of the RAM 2B region. OS– 0x0: AOSV_2B= 0x02 OS– 0x1: AOSV_2B= 0x04 OS– 0x2: AOSV_2B= 0x08 OS– 0x3: AOSV_2B= 0x10 The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2b. When the APT value is added to this start address, the current RAM cell TSF_Tx is addressed. value is set automatically when OSS in the PPLL_OSW register is set:
7 to 0	AOSV_2A	Address offset value of the RAM 2A region. OS– 0x0: AOSV_2A= 0x00 OS– 0x1: AOSV_2A= 0x00 OS– 0x2: AOSV_2A= 0x00 OS– 0x3: AOSV_2A= 0x00 The value in this field is to be multiplied by 256 (shift left 8 Bits) and added with the start address of the RAM in order to get the start address of RAM region 2a. When the APT value is added to this start address, the current RAM cell RDT_Tx is addressed. value is set automatically when OSS in the PPLL_OSW register is set:

38.22.18.10 DPLL_APT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	APT_2B							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APT_2B		WAPT_2B	—	APT										WAPT	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 38.305 DPLL_APT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 14	APT_2B	Address pointer TRIGGER for RAM region 2b; Actual RAM pointer address value for TSF_T[i] Actual RAM pointer address of TRIGGER events in FULL_SCALE for 2*(TNU+1) TRIGGER periods; this pointer is used for the RAM region 2b. The RAM pointer is initially set to zero. For SYT=1: The pointer APT_2B is incremented by SYN_T_OLD for each active TRIGGER event (simultaneously with APT and APT_2C) for DIR1=0 when an active TRIGGER input appears. For DIR1=1 (backwards) the APT is decremented by SYN_T_OLD. For SYT=0: APT_2B is incremented or decremented by 1. In addition when the APT_2C value is written by the CPU – in order to synchronize the DPLL- with the next active TRIGGER event the APT_2B_EXT value is added/subtracted (while APT_2B_STATUS is one; see DPLL_APT_SYNC register at **Section 38.22.18.24, DPLL_APT_SYNC**). **NOTE** This value can only be written when the WAPT_2B bit is set.
13	WAPT_2B	Write bit for address pointer APT_2B, read as zero. 0: The APT_2B is not writable 1: The APT_2B is writable
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.305 DPLL_APT Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 2	APT	Address pointer TRIGGER; Actual RAM pointer address value offset for DT_T[i] and RDT_T[i] in FULL_SCALE for 2*(TNU+1–SYN_NT) TRIGGER events. This pointer is used for the RAM region 2 subsections 2a and 2d. The pointer APT is incremented for each active TRIGGER event (simultaneously with APT_2B, APT_2C) for DIR1=0. For DIR1=1 the APT is decremented. The APT offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region. **NOTES** 1. The APT pointer value is directed to the RAM position, in which the data values are to be written, which corresponds to the last increment. The APT value is not to be changed, when the direction (shown by DIR1) changes, because it points always to a storage place after the considered increment. Changing of DIR1 takes place always after an active TRIGGER event and the resulting increment/decrement. 2. This value can only be written when the WAPT bit is set.
1	WAPT	Write bit for address pointer APT, read as zero. 0: The APT is not writable 1: The APT is writable
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

38.22.18.11 DPLL_APS

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28028_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	APS_1C2			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APS_1C2		WAPS_1C2	—	—	—	—	—	APS						WAPS	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 38.306 DPLL_APS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 14	APS_1C2	Address pointer STATE for RAM region 1c2; Actual RAM pointer address value for TSF_S[i]. Initial value: zero (0x00). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1) in normal and emergency mode; this pointer is used for the RAM region 1c2. For SYS=1: APS_1C2 is incremented (decremented) by SYN_S_OLD for each active STATE event and DIR2=0 (DIR2=1). For SYS=0: APT_1c2 is incremented or decremented by 1 respectively. The APS_1C2 offset value is added in the above shown bit position with the subsection offset of the RAM region. In addition when the APS_1C3 value is written by the CPU – in order to synchronize the DPLL- with the next active STATE event the APS_1C2_EXT value is added/subtracted (while APS_1C2_STATUS is one; see DPLL_APT_SYNC register at **Section 38.22.18.25, DPLL_APS_SYNC**). **NOTE** This value can only be written when the WAPS_1C2 bit is set
13	WAPS_1C2	Write bit for address pointer APS_1C2, read as zero. 0: The APS_1C2 is not writable 1: The APS_1C2 is writable
12 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.306 DPLL_APS Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 2	APS	Address pointer STATE; Actual RAM pointer address value for DT_S[i] and RDT_S[i] Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1–SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)–SYN_NS for SYSF=1 respectively; this pointer is used for the RAM region 1c1 and 1c4. APS is incremented (decremented) by one for each active STATE event and DIR2=0 DIR2=1). The APS offset value is added in the above shown bit position with the subsection offset of the RAM region. **NOTES** 1. The APS pointer value is directed to the RAM position, in which the data values are to be written, which correspond to the last increment. The APS value is not to be changed, when the direction (shown by DIR2) changes, because it points always to a storage place after the considered increment. Changing of DIR2 takes place always after an active STATE event and the resulting increment/decrement. 2. This value can only be written when the WAPS bit is set.
1	WAPS	Write bit for address pointer APS, read as zero. 0: The APS is not writable 1: The APS is writable
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

38.22.18.12 DPLL_APT_2C

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2802C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	APT_2C										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 38.307 DPLL_APT_2C Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset. **NOTES** 1. The APT_2C pointer values are directed to the RAM position of the profile element in RAM region 2c, which correspond to the current increment. For DIR1=0 (DIR1=1) the pointers APT_2C_x are incremented (decremented) by one simultaneously with APT. For SMC=0 the change of DIR1 takes place always after an active TRIGGER event (by evaluation of the inactive slope) and the resulting increment/decrement. In the case SMC=1 the direction change is known before the input event is processed. 2. The correction of the APT_2C pointer differs: for SMC=0 correct 4 times and for SMC=1 correct only 2 times. 3. The APT_2C_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region.
11 to 2	APT_2C	Address pointer TRIGGER for RAM region 2c; Actual RAM pointer address value for ADT_T[i]. Actual RAM pointer address value of TRIGGER adapt events in FULL_SCALE for 2*(TNU+1–SYN_NT) TRIGGER periods depending on the size of the used RAM 2; this pointer is used for the RAM region 2 for the subsection 2c only. The RAM pointer is initially set to zero. The APT_2C value is set by the CPU when the synchronization condition was detected. Within the RAM region 2c initially the conditions for synchronization gaps and adapted values are stored by the CPU.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

38.22.18.13 DPLL_APS_1C3

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	APS_1C3						—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 38.308 DPLL_APS_1C3 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset. **NOTES** 1. The APS_1C3 pointer value is directed to the RAM position of the profile element in RAM region 1c2, which corresponds to the current increment. When changing the direction DIR1 or DIR2 respectively, this is always known before an active STATE event is processed. This is because of the pattern recognition in SPE (for PMSM) or because of the direction change recognition by TRIGGER. This direction change results in an automatic increment (forwards) or decrement (backwards) when the input event occurs in addition with a 2 times correction. 2. The APS_1C3_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region. 3. This register is only used when DPLL_CTRL_11.STATE_EXT is not set. If DPLL_CTRL_11.STATE_EXT is set any read/write access to this register will return AEI_STATUS = 10_B.
7 to 2	APS_1C3	Address pointer STATE for RAM region 1c3; Actual RAM pointer address value for ADT_S[i] Initial value: zero (00_H). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 64 STATE events but limited to 2*(SNU+1–SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)–SYN_NS for SYSF=1 respectively; this pointer is used for the RAM region 1c3. The RAM pointer is set by the CPU accordingly, when the synchronization condition was detected.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note: Note: The APT_2C pointer values are directed to the RAM position of the profile element in RAM region 2c, which correspond to the current increment. For DIR1=0 (DIR1=1) the pointers APT_2C_x are incremented (decremented) by one simultaneously with APT. For SMC=0 the change of DIR1 takes place always after an active TRIGGER event (by evaluation of the inactive slope) and the resulting increment/decrement. In the case SMC=1 the direction change is known before the input event is processed.

The correction of the APT_2C pointer differs: for SMC=0 correct 4 times and for SMC=1 correct only 2 times.

The APT_2C_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region.

38.22.18.14 DPLL_NUTC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WVTN	WSYN	WNUT	—	—	—	—	VTN						SYN_T_OLD		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SYN_T			—	—	FST	NUTE									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.309 DPLL_NUTC Register Contents (1/3)

Bit Position	Bit Name	Function
31	WVTN	Write control bit for VTN; read as zero. 0: The VTN value is not writable 1: The VTN value is writable
30	WSYN	Write control bit for SYN_T and SYN_T_OLD; read as zero. 0: The SYN_T value is not writable 1: The SYN_T value is writable
29	WNUT	Write control bit for NUTE and FST; read as zero. 0: The NUTE value is not writable 1: The NUTE value is writable
28 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 19	VTN	Virtual TRIGGER number; number of virtual increments in the current NUTE region This value reflects the number of virtual increments in the current NUTE region; for NUTE=1 this value is zero, when the CPU sets NUTE to a value > 1, it must also set VTN to the correspondent value; for NUTE is set to FULL_SCALE including NUTE=zero (2 modulo 2) the VTN is to be set to 2* SYN_NT. The VTN value is subtracted from the NUTE value in order to get the corresponding APT value for the past; the VTN value is not used for the APT_2B pointer. VTN is to be updated by the CPU when a new gap is to be considered for NUTE or a gap is leaving the NUTE region; for this purpose the TINT values in the profile can be used to generate an interrupt for the CPU at the corresponding positions; no further update of VTN is necessary when NUTE is set to FULL_SCALE. **NOTE** This value can only be written when the WVTN bit is set.

Table 38.309 DPLL_NUTC Register Contents (2/3)

Bit Position	Bit Name	Function
18 to 16	SYN_T_OLD	Number of real and virtual events to be considered for the last increment. This value reflects the NT value of the last but one valid increment, stored in ADT_T[i]; is updated automatically when writing SYN_T. **NOTE** This value is updated by the SYN_T value when the WSYN bit in this register is set.
15 to 13	SYN_T	Number of real and virtual events to be considered for the current increment. This value reflects the NT value of the last valid increment, stored in ADT_T[i]; to be updated after all calculations in step 17 of Table (DPLL_6908)*[XREF TARGET " (DPLL_6908)] NOT EXIST*. **NOTE** This value can only be written when the WSYN bit in this register is set.
12, 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	FST	FULL_SCALE of TRIGGER; this value is to be set, when NUTE is set to FULL_SCALE. **NOTE** This value can only be written when the WNUT bit is set.

Table 38.309 DPLL_NUTC Register Contents (3/3)

Bit Position	Bit Name	Function
9 to 0	NUTE	Number of recent TRIGGER events used for SUB_INC1 and action calculations modulo 2*(TNUmax+1). 0: The NUTE value is less than FULL_SCALE 1: The NUTE value is equal to FULL_SCALE Number of last nominal increments to be considered for the calculations. No gap is considered in that case for this value, but in the VTN value (see below): This value is set by the CPU, but reset automatically to 1 by a change of direction or loss of LOCK. Each other value can be set by the CPU, maybe Full_SCALE, HALF_SCALE or parts of them. For FULL_SCALE set NUTE= 2*(TNU+1) and for HALF_SCALE NUTE= TNU+1. The relation values QDT_Tx are calculated using NUTE values in the past with its maximum value of 2*(TNU+1). The value zero (in combination with the value FST=1) does mean 2 values in the past. **NOTES** 1. To prevent that inconsistencies between internal pointer in which NUTE is used and the case decision of different prediction method's for prediction of the next event and PMT (position minus time) occur, the NUTE value is stored internally at that point of time when the internal pointers are calculated for the next event cycle. 0 = the NUTE value is less then FULL_SCALE 1 = the NUTE value is equal to FULL_SCALE This value is set by the CPU, but reset automatically to “0” by a change of direction or loss of LOCK. 2. To prevent that inconsistencies between internal pointer in which NUTE is used and the case decision of different prediction method's for prediction of the next event and PMT (position minus time) occur, the NUTE value is stored internally at that point of time when the internal pointers are calculated for the next event cycle 0 = the NUTE value is less then FULL_SCALE 1 = the NUTE value is equal to FULL_SCALE This value is set by the CPU, but reset automatically to “0” by a change of direction or loss of LOCK.

Note: DPLL Number of recent TRIGGER events used for calculations (mod 2*(TNU +1-SYN_NT)).

38.22.18.15 DPLL_NUSC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28038_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WVSN	WSYN	WNUS	—	—	—	—	VSN						SYN_S_OLD		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SYN_S_OLD			SYN_S						FSS	NUSE					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.310 DPLL_NUSC Register Contents (1/2)

Bit Position	Bit Name	Function
31	WVSN	Write control bit for VSN; read as zero. 0: The VSN value is not writable 1: The VSN value is writable **NOTE** This register is only used when DPLL_CTRL_11.STATE_EXT is not set. If DPLL_CTRL_11.STATE_EXT is set any read/write access to this register will return AEI_STATUS = 10_B.
30	WSYN	Write control bit for SYN_S and SYN_S_OLD; read as zero. 0: The SYN_S value is not writable 1: The SYN_S value is writable
29	WNUS	Write control bit for NUSE; read as zero. 0: The NUSE value is not writable 1: The NUSE value is writable
28 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 19	VSN	Virtual STATE number; number of virtual state increments in the current NUSE region. This value reflects the number of virtual increments in the current NUSE region; for NUSE=1 this value is zero, when the CPU sets NUSE to a value > 1 or zero (2 modulo 2), it must also set VSN to the correspondent value; the VSN value is subtracted from the NUSE value in order to get the corresponding APS value for the past; the VSN value is not used for the APS_1C2 pointer. VSN is to be updated by the CPU when a new gap is to be considered for NUSE or a gap is leaving the NUSE region; for this purpose the SASI interrupt can be used; no further update of VSN is necessary when NUSE is set to FULL_SCALE. **NOTE** This value can only be written when the WVSN bit is set.

Table 38.310 DPLL_NUSC Register Contents (2/2)

Bit Position	Bit Name	Function
18 to 13	SYN_S_OLD	Number of real and virtual events to be considered for the last increment. This value reflects the NS value of the last but one valid increment, stored in ADT_S[i]; is updated automatically when writing SYN_S. **NOTE** This value is updated by the SYN_S value when the WSYN bit in this register is set.
12 to 7	SYN_S	Number of real and virtual events to be considered for the current increment. This value reflects the NS value of the last valid increment, stored in ADT_S[i]; to be updated after all calculations in step 37 of Table (DPLL_6908)*[XREF TARGET " (DPLL_6908)] NOT EXIST*. **NOTE** This value can only be written when the WSYN bit in this register is set.
6	FSS	FULL_SCALE of STATE; this value is to be set, when NUSE is set to FULL_SCALE 0: The NUSE value is less than FULL_SCALE 1: The NUSE value is equal to FULL_SCALE This value is set by the CPU, but reset automatically to 0 by a change of direction or loss of LOCK. **NOTE** This value can only be written when the WNUS bit is set.
5 to 0	NUSE	Number of recent STATE events used for SUB_INCx calculations modulo 2*(SNUmax+1). No gap is considered in that case for this value, but in the VSN value (see below): This register is set by the CPU but reset automatically to 1 by a change of direction or loss of LOCK. Each other value can be set by the CPU, maybe Full_SCALE, HALF_SCALE or parts of them. The relation values QDT_Sx are calculated using NUSE values in the past with its maximum value of 2*SNU+1. **NOTES** 1. To prevent that inconsistencies between internal pointer in which NUSE is used and the case decision of different prediction method's for prediction of the next event and PMT (position minus time) occur, the NUSE value is stored internally at that point of time when the internal pointers are calculated for the next event cycle. 2. This value can only be written when the WNUS bit is set.

38.22.18.16 DPLL_NTI_CNT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2803C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	NTI_CNT									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.311 DPLL_NTI_CNT Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	NTI_CNT	Number of TRIGGERs to interrupt; Number of active TRIGGER events to the next DPLL_CDTI interrupt. This value shows the remaining TRIGGER events until an active TRIGGER slope results in a DPLL_CDTI interrupt; the value is to be count down for each active TRIGGER event.

38.22.18.17 DPLL_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DCGI	SORI	TORI	CDSI	CDTI	TE4I	TE3I	TE2I	TE1I	TE0I	LL2I	GL2I
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EI	LL1I	GL1I	W1I	W2I	PWI	TASI	SASI	MTI	MSI	TISI	SISI	TAXI	TINI	PEI	PDI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.312 DPLL_IRQ_NOTIFY Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	DCGI	Direction change interrupt 0: No direction change of TRIGGER is detected 1: Direction change of TRIGGER is detected **NOTE** The interrupt occurs at line number 0.
26	SORI	STATE out of range 0: STATE is not out of range 1: STATE is out of range, the SOR bit in the DPLL_STATUS register is set to 1
25	TORI	TRIGGER out of range interrupt 0: TRIGGER is not out of range 1: TRIGGER is out of range, the TOR bit in the DPLL_STATUS register is set to 1
24	CDSI	Calculation of STATE duration done 0: No Interrupt on calculated STATE duration requested 1: Interrupt on calculated STATE duration requested
23	CDTI	Calculation of TRIGGER duration done, only while NTI_CNT is zero. 0: No Interrupt on calculated TRIGGER duration requested or NTI_CNT is not zero 1: Interrupt on calculated TRIGGER duration requested while NTI_CNT is zero
22	TE4I	TRIGGER event interrupt 4. 0: No Interrupt on TRIGGER event 4 requested 1: Interrupt on TRIGGER event 4 requested
21	TE3I	TRIGGER event interrupt 3. 0: No Interrupt on TRIGGER event 3 requested 1: Interrupt on TRIGGER event 3 requested
20	TE2I	TRIGGER event interrupt 2. 0: No Interrupt on TRIGGER event 2 requested 1: Interrupt on TRIGGER event 2 requested

Table 38.312 DPLL_IRQ_NOTIFY Register Contents (2/3)

Bit Position	Bit Name	Function
19	TE1I	TRIGGER event interrupt 1. 0: No Interrupt on TRIGGER event 1 requested 1: Interrupt on TRIGGER event 1 requested
18	TE0I	TRIGGER event interrupt 0. 0: No Interrupt on TRIGGER event 0 requested 1: Interrupt on TRIGGER event 0 requested
17	LL2I	Loss of lock interrupt for SUB_INC2. 0: The lock loss interrupt is not requested 1: The lock loss interrupt is requested
16	GL2I	Get of lock interrupt, for SUB_INC2. 0: The lock getting interrupt is not requested 1: The lock getting interrupt is requested
15	EI	Error interrupt (see status register bit 31). 0: The error interrupt is not requested 1: The error interrupt is requested
14	LL1I	Loss of lock interrupt for SUB_INC1. 0: The lock loss interrupt is not requested 1: The lock loss interrupt is requested
13	GL1I	Get of lock interrupt, for SUB_INC1. 0: The lock getting interrupt is not requested 1: The lock getting interrupt is requested
12	W1I	Write access to RAM region 1b or 1c interrupt. 0: The RAM write access interrupt is not requested 1: The RAM write access interrupt is requested
11	W2I	RAM write access to RAM region 2 interrupt. 0: The RAM write access interrupt is not requested 1: The RAM write access interrupt is requested
10	PWI	Plausibility window (PVT) violation interrupt of TRIGGER. 0: The plausibility window is not violated 1: The plausibility window is violated
9	TASI	TRIGGER active slope interrupt. 0: No active slope of TRIGGER is detected 1: An active slope of TRIGGER is detected
8	SASI	STATE active slope interrupt. 0: No active slope of STATE is detected 1: An active slope of STATE is detected
7	MTI	Missing TRIGGER interrupt. 0: The missing TRIGGER interrupt is not requested 1: The missing TRIGGER interrupt is requested
6	MSI	Missing STATE interrupt. 0: The missing STATE interrupt is not requested 1: The missing STATE interrupt is requested
5	TISI	TRIGGER inactive slope interrupt. 0: No inactive slope of TRIGGER is detected 1: An inactive slope of TRIGGER is detected **NOTE** The TISI bit is only set for an inactive slope when the preceding active slope was accepted. In the case of suppression of the last active slope by the plausibility check the next inactive slope is to be ignored. No set of TISI is performed in this case.
4	SISI	STATE inactive slope interrupt. 0: No inactive slope of STATE is detected 1: An inactive slope of STATE is detected

Table 38.312 DPLL_IRQ_NOTIFY Register Contents (3/3)

Bit Position	Bit Name	Function
3	TAXI	TRIGGER maximum hold time violation interrupt (dt > THMA > 0). 0: No violation of maximum hold time of TRIGGER is detected 1: A violation of maximum hold time of TRIGGER is detected
2	TINI	TRIGGER minimum hold time violation interrupt (dt ≤ THMI > 0). 0: No violation of minimum hold time of TRIGGER is detected 1: A violation of minimum hold time of TRIGGER is detected
1	PEI	DPLL enable interrupt; announces the switch on of the DEN bit. 0: The DPLL enable interrupt is not requested 1: The DPLL enable interrupt is requested **NOTE** This event is combined with the PDI interrupt to the common PDI + PEI interrupt line number 1.
0	PDI	DPLL disable interrupt; announces the switch off of the DEN bit. 0: The DPLL disable interrupt is not requested 1: The DPLL disable interrupt is requested **NOTE** This event is combined with the PEI interrupt to the common PDI + PEI interrupt line number 1.

Note: All bits in the DPLL_IRQ_NOTIFY register are set permanently until writing a one bit value is performed to the corresponding bit.

38.22.18.18 DPLL_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DCGI_IRQ_EN	SORI_IRQ_EN	TORI_IRQ_EN	CDSI_IRQ_EN	CDTI_IRQ_EN	TE4I_IRQ_EN	TE3I_IRQ_EN	TE2I_IRQ_EN	TE1I_IRQ_EN	TE0I_IRQ_EN	LL2I_IRQ_EN	GL2I_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EI_IRQ_EN	LL1I_IRQ_EN	GL1I_IRQ_EN	W1I_IRQ_EN	W2I_IRQ_EN	PWI_IRQ_EN	TASI_IRQ_EN	SASI_IRQ_EN	MTI_IRQ_EN	MSI_IRQ_EN	TISI_IRQ_EN	SISI_IRQ_EN	TAXI_IRQ_EN	TINI_IRQ_EN	PEI_IRQ_EN	PDI_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.313 DPLL_IRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	DCGI_IRQ_EN	Direction change interrupt 0: No Interrupt when a direction change of TRIGGER is detected 1: Interrupt when a direction change of TRIGGER is detected
26	SORI_IRQ_EN	STATE out of range 0: No Interrupt when STATE is out of range enabled 1: Interrupt when STATE is out of range enabled
25	TORI_IRQ_EN	TRIGGER out of range interrupt 0: No Interrupt when TRIGGER is out of range enabled 1: Interrupt when TRIGGER is out of range enabled1:
24	CDSI_IRQ_EN	Enable interrupt when calculation of TRIGGER duration done 0: No Interrupt on calculated STATE duration enabled 1: Interrupt on calculated STATE duration enabled
23	CDTI_IRQ_EN	Enable interrupt when calculation of TRIGGER duration done 0: No Interrupt on calculated TRIGGER duration enabled 1: Interrupt on calculated TRIGGER duration enabled
22	TE4I_IRQ_EN	TRIGGER event interrupt 4 enable. 0: No Interrupt on TRIGGER event 4 enabled 1: Interrupt on TRIGGER event 4 enabled
21	TE3I_IRQ_EN	TRIGGER event interrupt 3 enable. 0: No Interrupt on TRIGGER event 3 enabled 1: Interrupt on TRIGGER event 3 enabled
20	TE2I_IRQ_EN	TRIGGER event interrupt 2 enable. 0: No Interrupt on TRIGGER event 2 enabled 1: Interrupt on TRIGGER event 2 enabled
19	TE1I_IRQ_EN	TRIGGER event interrupt 1 enable. 0: No Interrupt on TRIGGER event 1 enabled 1: Interrupt on TRIGGER event 1 enabled
18	TE0I_IRQ_EN	TRIGGER event interrupt 0 enable. 0: No Interrupt on TRIGGER event 0 enabled 1: Interrupt on TRIGGER event 0 enabled
17	LL2I_IRQ_EN	Loss of lock interrupt enable for SUB_INC2. 0: The lock loss interrupt is not requested 1: The lock loss interrupt is requested

Table 38.313 DPLL_IRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
16	GL2I_IRQ_EN	Get of lock interrupt enable for SUB_INC2. 0: The lock getting interrupt is not requested 1: The lock getting interrupt is requested
15	EI_IRQ_EN	Error interrupt enable (see status register). 0: The error interrupt is not enabled 1: The error interrupt is enabled
14	LL1I_IRQ_EN	Loss of lock interrupt enable. 0: The lock loss interrupt is not enabled 1: The lock loss interrupt is enabled
13	GL1I_IRQ_EN	Get of lock interrupt enable, when lock arises. 0: The lock getting interrupt is not enabled 1: The lock getting interrupt is enabled
12	W1I_IRQ_EN	Write access to RAM region 1b or 1c interrupt. 0: The RAM write access interrupt is not enabled 1: The RAM write access interrupt is enabled.
11	W2I_IRQ_EN	RAM write access to RAM region 2 interrupt enable. 0: The RAM write access interrupt is not enabled 1: The RAM write access interrupt is enabled
10	PWI_IRQ_EN	Plausibility window (PVT) violation interrupt of TRIGGER enable. 0: The plausibility violation interrupt is not enabled 1: The plausibility violation interrupt is enabled
9	TASI_IRQ_EN	TRIGGER active slope interrupt enable. 0: The active slope TRIGGER interrupt is not enabled 1: The active slope TRIGGER interrupt is enabled
8	SASI_IRQ_EN	STATE active slope interrupt enable. 0: The active slope STATE interrupt is not enabled. 1: The active slope STATE interrupt is enabled
7	MTI_IRQ_EN	Missing TRIGGER interrupt enable. 0: The missing TRIGGER interrupt is not enabled 1: The missing TRIGGER interrupt is enabled
6	MSI_IRQ_EN	Missing STATE interrupt enable. 0: The missing STATE interrupt is not enabled 1: The missing STATE interrupt is enabled
5	TISI_IRQ_EN	TRIGGER inactive slope interrupt enable bit. 0: The interrupt at the inactive slope of TRIGGER is not enabled 1: The interrupt at the inactive slope of TRIGGER is enabled
4	SISI_IRQ_EN	STATE inactive slope interrupt enable bit. 0: The interrupt at the inactive slope of STATE is not enabled 1: The interrupt at the inactive slope of STATE is enabled
3	TAXI_IRQ_EN	TRIGGER maximum hold time violation interrupt enable bit. 0: Maximum hold time violation of TRIGGER interrupt is not enabled 1: The maximum hold time violation of TRIGGER interrupt is enabled
2	TINI_IRQ_EN	TRIGGER minimum hold time violation interrupt enable bit. 0: Minimum hold time violation of TRIGGER interrupt is not enabled 1: The minimum hold time violation of TRIGGER interrupt is enabled
1	PEI_IRQ_EN	DPLL enable interrupt enable, when switch on of the DEN bit. 0: The DPLL enable interrupt is not enabled 1: The DPLL enable interrupt is enabled
0	PDI_IRQ_EN	DPLL disable interrupt enable, when switch off of the DEN bit. 0: The DPLL disable interrupt is not enabled 1: The DPLL disable interrupt is enabled

38.22.18.19 DPLL_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28048_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	TRG_DCGI	TRG_SORI	TRG_TORI	TRG_CDSI	TRG_CDTI	TRG_TE4I	TRG_TE3I	TRG_TE2I	TRG_TE1I	TRG_TE0I	TRG_LL2I	TRG_GL2I
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG_EI	TRG_LL1I	TRG_GL1I	TRG_W1I	TRG_W2I	TRG_PWI	TRG_TASI	TRG_SASI	TRG_MTI	TRG_MSI	TRG_TISI	TRG_SISI	TRG_TAXI	TRG_TINI	TRG_PEI	TRG_PDI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.314 DPLL_IRQ_FORCINT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	TRG_DCGI	Force interrupt DCGI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
26	TRG_SORI	Force Interrupt SORI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
25	TRG_TORI	Force Interrupt TORI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
24	TRG_CDSI	Force Interrupt CDSI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
23	TRG_CDTI	Force Interrupt CDTI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
22	TRG_TE4I	Force Interrupt TE4I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
21	TRG_TE3I	Force Interrupt TE3I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
20	TRG_TE2I	Force Interrupt TE2I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
19	TRG_TE1I	Force Interrupt TE1I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
18	TRG_TE0I	Force Interrupt TE0I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
17	TRG_LL2I	Force Interrupt LL2I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock

Table 38.314 DPLL_IRQ_FORCINT Register Contents (2/2)

Bit Position	Bit Name	Function
16	TRG_GL2I	Force Interrupt GL2I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
15	TRG_EI	Force Interrupt EI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
14	TRG_LL1I	Force Interrupt LL1I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
13	TRG_GL1I	Force Interrupt GL1I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
12	TRG_W1I	Force Interrupt W1I 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
11	TRG_W2I	Force Interrupt W2IF 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
10	TRG_PWI	Force Interrupt PWI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
9	TRG_TASI	Force Interrupt TASI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
8	TRG_SASI	Force Interrupt SASI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
7	TRG_MTI	Force Interrupt MTI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
6	TRG_MSI	Force Interrupt MSI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
5	TRG_TISI	Force Interrupt TISI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
4	TRG_SISI	Force Interrupt SISI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
3	TRG_TAXI	Force Interrupt TAXI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
2	TRG_TINI	Force Interrupt TINI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
1	TRG_PEI	Force Interrupt PEI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock
0	TRG_PDI	Force Interrupt PDI 0: The corresponding interrupt is not forced 1: The corresponding interrupt is forced for one clock **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

38.22.18.20 DPLL_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2804C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.315 DPLL_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.22.18.21 DPLL_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28050_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DCGI_EIRQ_EN	SORI_EIRQ_EN	TORI_EIRQ_EN	CDSI_EIRQ_EN	CDTI_EIRQ_EN	TE4I_EIRQ_EN	TE3I_EIRQ_EN	TE2I_EIRQ_EN	TE1I_EIRQ_EN	TE0I_EIRQ_EN	LL2I_EIRQ_EN	GL2I_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EI_EIRQ_EN	LL1I_EIRQ_EN	GL1I_EIRQ_EN	W1I_EIRQ_EN	W2I_EIRQ_EN	PWI_EIRQ_EN	TASI_EIRQ_EN	SASI_EIRQ_EN	MTI_EIRQ_EN	MSI_EIRQ_EN	TISI_EIRQ_EN	SISI_EIRQ_EN	TAXI_EIRQ_EN	TINI_EIRQ_EN	PEI_EIRQ_EN	PDI_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.316 DPLL_EIRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	DCGI_EIRQ_EN	Direction change interrupt 0: No Interrupt when a direction change of TRIGGER is detected 1: Interrupt when a direction change of TRIGGER is detected
26	SORI_EIRQ_EN	STATE out of range 0: No Interrupt when STATE is out of range enabled 1: Interrupt when STATE is out of range enabled
25	TORI_EIRQ_EN	TRIGGER out of range interrupt 0: No Interrupt when TRIGGER is out of range enabled 1: Interrupt when TRIGGER is out of range enabled
24	CDSI_EIRQ_EN	Enable interrupt when calculation of TRIGGER duration done 0: No Interrupt on calculated STATE duration enabled 1: Interrupt on calculated STATE duration enabled
23	CDTI_EIRQ_EN	Enable interrupt when calculation of TRIGGER duration done 0: No Interrupt on calculated TRIGGER duration enabled 1: Interrupt on calculated TRIGGER duration enabled
22	TE4I_EIRQ_EN	TRIGGER event interrupt 4 enable. 0: No Interrupt on TRIGGER event 4 enabled 1: Interrupt on TRIGGER event 4 enabled
21	TE3I_EIRQ_EN	TRIGGER event interrupt 3 enable. 0: No Interrupt on TRIGGER event 3 enabled 1: Interrupt on TRIGGER event 3 enabled
20	TE2I_EIRQ_EN	TRIGGER event interrupt 2 enable. 0: No Interrupt on TRIGGER event 2 enabled 1: Interrupt on TRIGGER event 2 enabled
19	TE1I_EIRQ_EN	TRIGGER event interrupt 1 enable. 0: No Interrupt on TRIGGER event 1 enabled 1: Interrupt on TRIGGER event 1 enabled
18	TE0I_EIRQ_EN	TRIGGER event interrupt 0 enable. 0: No Interrupt on TRIGGER event 0 enabled 1: Interrupt on TRIGGER event 0 enabled
17	LL2I_EIRQ_EN	Loss of lock interrupt enable for SUB_INC2. 0: The lock loss interrupt is not requested 1: The lock loss interrupt is requested

Table 38.316 DPLL_EIRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
16	GL2I_EIRQ_EN	Get of lock interrupt enable for SUB_INC2. 0: The lock getting interrupt is not requested 1: The lock getting interrupt is requested
15	EI_EIRQ_EN	Error interrupt enable (see status register). 0: The error interrupt is not enabled 1: The error interrupt is enabled
14	LL1I_EIRQ_EN	Loss of lock interrupt enable. 0: The lock loss interrupt is not enabled 1: The lock loss interrupt is enabled
13	GL1I_EIRQ_EN	Get of lock interrupt enable, when lock arises. 0: The lock getting interrupt is not enabled 1: The lock getting interrupt is enabled
12	W1I_EIRQ_EN	Write access to RAM region 1b or 1c interrupt. 0: The RAM write access interrupt is not enabled 1: The RAM write access interrupt is enabled.
11	W2I_EIRQ_EN	RAM write access to RAM region 2 interrupt enable. 0: The RAM write access interrupt is not enabled 1: The RAM write access interrupt is enabled
10	PWI_EIRQ_EN	Plausibility window (PVT) violation interrupt of TRIGGER enable. 0: The plausibility violation interrupt is not enabled 1: The plausibility violation interrupt is enabled
9	TASI_EIRQ_EN	TRIGGER active slope interrupt enable. 0: The active slope TRIGGER interrupt is not enabled 1: The active slope TRIGGER interrupt is enabled
8	SASI_EIRQ_EN	STATE active slope interrupt enable. 0: The active slope STATE interrupt is not enabled. 1: The active slope STATE interrupt is enabled
7	MTI_EIRQ_EN	Missing TRIGGER interrupt enable. 0: The missing TRIGGER interrupt is not enabled 1: The missing TRIGGER interrupt is enabled
6	MSI_EIRQ_EN	Missing STATE interrupt enable. 0: The missing STATE interrupt is not enabled 1: The missing STATE interrupt is enabled
5	TISI_EIRQ_EN	TRIGGER inactive slope interrupt enable bit. 0: The interrupt at the inactive slope of TRIGGER is not enabled 1: The interrupt at the inactive slope of TRIGGER is enabled
4	SISI_EIRQ_EN	STATE inactive slope interrupt enable bit. 0: The interrupt at the inactive slope of STATE is not enabled 1: The interrupt at the inactive slope of STATE is enabled
3	TAXI_EIRQ_EN	TRIGGER maximum hold time violation interrupt enable bit. 0: Maximum hold time violation of TRIGGER interrupt is not enabled 1: The maximum hold time violation of TRIGGER interrupt is enabled
2	TINI_EIRQ_EN	TRIGGER minimum hold time violation interrupt enable bit. 0: Minimum hold time violation of TRIGGER interrupt is not enabled 1: The minimum hold time violation of TRIGGER interrupt is enabled
1	PEI_EIRQ_EN	DPLL enable interrupt enable, when switch on of the DEN bit. 0: The DPLL enable interrupt is not enabled 1: The DPLL enable interrupt is enabled
0	PDI_EIRQ_EN	DPLL disable interrupt enable, when switch off of the DEN bit. 0: The DPLL disable interrupt is not enabled 1: The DPLL disable interrupt is enabled

38.22.18.22 DPLL_INC_CNT1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280B0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	INC_CNT1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INC_CNT1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.317 DPLL_INC_CNT1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	INC_CNT1	Actual number of pulses to be still sent out at the current increment until the next active input signal in automatic end mode; Automatic addition of the number of demanded pulses MLT/MLS1 when getting an active TRIGGER / STATE input in normal or emergency mode respectively when SGE1=1; writable only for test purposes when DEN=0 In the case of a change of the direction the wrong number of pulses is corrected twice: Add the difference between NMB_T and INC_CNT1 twice to INC_CNT1 before sending out the correction pulses. **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.23 DPLL_INC_CNT2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	INC_CNT2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INC_CNT2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.318 DPLL_INC_CNT2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	INC_CNT2	Actual number of pulses to be still sent out at the current increment until the next active input signal in automatic end mode; Automatic addition of the number of demanded pulses MLS2 when getting an active TRIGGER / STATE input in normal or emergency mode respectively when SGE2=1; writable only for test purposes when DEN=0 In the case of a change of the direction the wrong number of pulses is corrected twice: Add the difference between NMB_S and INC_CNT2 twice to INC_CNT2 before sending out the correction pulses. **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.24 DPLL_APT_SYNC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280B8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	APT_2B_OLD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APT_2B_OLD		—	—	—	—	—	—	—	APT_2B_STATUS	APT_2B_EXT					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.319 DPLL_APT_SYNC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 14	APT_2B_OLD	Address pointer TRIGGER for RAM region 2b at synchronization time; this value is set by the current APT_2B value when the synchronization takes place for the first active TRIGGER event after writing APT_2C but before adding the offset value APT_2B_EXT (that means: when APT_2B_STATUS=1). Address pointer APT_2B value at the moment of synchronization, before the offset value is added, that means the pointer with this value points to the last value before the additional inserted gap
13 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	APT_2B_STATUS	Address pointer 2b status; set by CPU before the synchronization is performed. The value is cleared when the APT_2B_OLD value is written. 0: APT_2B_EXT is not to be considered. 1: APT_2B_EXT has to be considered for time stamp field extension.

Table 38.319 DPLL_APT_SYNC Register Contents (2/2)

Bit Position	Bit Name	Function
5 to 0	APT_2B_EXT	Address pointer 2b extension; this offset value determines, by which value the APT_2B is changed at the synchronization time; set by CPU before the synchronization is performed. This offset value is the number of virtual increments to be inserted in the TSF for an imminent intended synchronization; the CPU sets its value dependent on the gaps until the synchronization time taking into account the considered NUTE value to be set and including the next future increment (when SYN_T_OLD is still 1). When the synchronization takes place, this value is to be added to the APT_2B address pointer (for forward direction, DIR1=0) and the APT_2B_STATUS bit is cleared after it. For backward direction subtract APT_2B_EXT accordingly. This correction is done after updating the RAM TSF with the last TS_T value. **NOTE** When the synchronization is intended and the NUTE value is to be set to FULL_SCALE after it, the APT_2B_EXT value must be set to 2*SYN_NT in order to be able to fill all gaps in the extended TSF_T with the corresponding values by the CPU. When still not all values for FULL_SCALE are available, the APT_2B_EXT value considers only a share according to the corresponding NUTE value to be set after the synchronization.

38.22.18.25 DPLL_APS_SYNC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280BC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	APS_1C2_OLD			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APS_1C2_OLD		—	—	—	—	—	—	—	APS_1C2_STATUS	APS_1C2_EXT					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.320 DPLL_APS_SYNC Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 14	APS_1C2_OLD	Address pointer STATE for RAM region 1c2 at synchronization time; this value is set by the current APS_1C2 value when the synchronization takes place for the first active STATE event after writing APS_1C3 but before adding the offset value APS_1C2_EXT (that means: when APS_1C2_STATUS=1). Address pointer APS_1C2 value at the moment of synchronization, before the offset value is added, that means the pointer with this value points to the last value before the additional inserted gap
13 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	APS_1C2_STATUS	Address pointer 1c2 status; set by CPU before the synchronization is performed. The value is cleared automatically when the APS_1C2_OLD value is written. 0: APS_1C2_EXT is not to be considered. 1: APS_1C2_EXT has to be considered for time stamp field extension.
5 to 0	APS_1C2_EXT	Address pointer 1c2 extension; this offset value determines, by which value the APS_1C2 is changed at the synchronization time; set by CPU before the synchronization is performed. This offset value is the number of virtual increments to be inserted in the TSF for an imminent intended synchronization; the CPU sets its value dependent on the gaps until the synchronization time taking into account the considered NUSE value to be set and including the next future increment (when SYN_S_OLD is still 1). When the synchronization takes place, this value is to be added to the APS_1C2 address pointer (for forward direction, DIR2=0) and the APT_1c2_status bit is cleared after it. For backward direction subtract APS_1C2_EXT accordingly. **NOTE** When the synchronization is intended and the NUSE value is to be set to FULL_SCALE after it, the APS_1C2_EXT value must be set to SYN_NS (for SYSF=1) or 2*SYN_NS (for SYSF=0) in order to be able to fill all gaps in the extended TSF_S with the corresponding values by the CPU. When still not all values for FULL_SCALE are available, the APS_1C2_EXT value considers only a share according to the NUSE value to be set after the synchronization.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is not set. If DPLL_CTRL_11.STATE_EXT is set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.26 DPLL_TBU_TS0_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBU_TS0_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBU_TS0_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.321 DPLL_TBU_TS0_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TBU_TS0_T	Value of TBU_TS0 at the last TRIGGER event; for each T_valid the value of TBU_TS0 is stored in this register; the register is writable only for test purposes when DEN=0. **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.27 DPLL_TBU_TS0_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBU_TS0_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBU_TS0_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.322 DPLL_TBU_TS0_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TBU_TS0_S	Value of TBU_TS0 at the last STATE event; for each S_VALID the value of TBU_TS0 is stored in this register; the register is writable only for test purposes when DEN=0. **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.28 DPLL_ADD_IN_LD1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADD_IN_LD1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD_IN_LD1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.323 DPLL_ADD_IN_LD1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ADD_IN_LD1	Input value for SUB_INC1 generation, given by CPU. This value can be used in normal and emergency mode (SMC=0) as well as for SMC=1. For DLM1 = 1: **NOTES** 1. The value is loaded by the CPU but used by the DPLL only for DLM1=1 (see DPLL_CTRL_1 register). When switching DLM1 to 1, the value in the register is used for the SUB_INC1 generation beginning from the next active TRIGGER or STATE event respectively independently if new values are written by the CPU or not. 2. When a new value is written the output frequency changes according to the given value beginning immediately from the moment of writing. Do not wait for performing step 10 in the state machine for ADD_IN calculations. 3. If the ADD_IN_LD1 value is zero all pulses are sent with the highest possible frequency. For DLM1 = 0: **NOTES** 1. The value loaded by the CPU is stored directly in the internal add_in register which is used to control the sub increment pulse generator directly (see DPLL_CTRL_1 register, DLM1 = 0). 2. When a new ADD_IN_LD1 value is written the output frequency is immediately changed from the moment of writing. The ADD_IN values calculated internally of the DPLL are written to the internal ADD_IN register as well. In the moment when the internal calculation of the ADD_IN values is writing the results into the internal ADD_IN register of the pulse generator the internally calculated ADD_IN values does always have higher priority compared to the values written via the ADD_IN_LD1 register. 3. If the ADD_IN_LD1 value is zero all pulses are sent with the highest possible frequency.

38.22.18.29 DPLL_ADD_IN_LD2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280CC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADD_IN_LD2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD_IN_LD2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.324 DPLL_ADD_IN_LD2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ADD_IN_LD2	Input value for SUB_INC2 generation, given by CPU. This value can be used for SMC=1 while RMO=1. For DLM2 = 1: **NOTES** 1. The value is loaded by the CPU but used by the DPLL only for DLM2=1 (see DPLL_CTRL_1 register). When switching DLM2 to 1, the value in the register is used for the SUB_INC2 generation beginning from the next STATE event respectively independently if new values are written by the CPU or not. 2. When a new value is written the output frequency changes according to the given value beginning immediately from the moment of writing. Do not wait for performing step 30 in the state machine for ADD_IN calculations. 3. If the ADD_IN_LD2 value is zero all pulses are sent with the highest possible frequency. For DLM2 = 0: **NOTES** 1. The value loaded by the CPU is stored directly in the internal add_in register which is used to control the sub increment pulse generator directly (see DPLL_CTRL_1 register, DLM2 = 0). 2. When a new ADD_IN_LD2 value is written the output frequency is immediately changed from the moment of writing. The ADD_IN values calculated internally of the DPLL are written to the internal ADD_IN register as well. In the moment when the internal calculation of the ADD_IN values is writing the results into the internal ADD_IN register of the pulse generator the internally calculated ADD_IN values does always have higher priority compared to the values written via the ADD_IN_LD2 register. 3. If the ADD_IN_LD2 value is zero all pulses are sent with the highest possible frequency.

38.22.18.30 DPLL_STATUS

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $280FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ERR	LOCK1	FTD	FSD	SYT	SYS	LOCK2	—	BWD1	BWD2	ITN	ISN	CAIP1	CAIP2	CSVT	CSVS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOW_RES	—	—	RAM2_ERR	MT	TOR	MS	SOR	PSE	RCT	RCS	CRO	CTO	—	CSO	FPCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Table 38.325 DPLL_STATUS Register Contents (1/5)

Bit Position	Bit Name	Function
31	ERR	Error during configuration or operation resulting in unexpected values. 0: when all bits in position 8 to 0 and 10 and 12 are zero 1: when at least one bit in position 8 to 0 or 10 or 12 is one **NOTES** 1. The SOR bit is set, when the time from the last valid STATE event to the next active STATE slope exceeds the value of the last nominal STATE duration multiplied with the value of the SLR register (see **Section 38.22.22.37, DPLL_SLR**) and is reset, when at the current or last active input event a direction change was detected. The SYS bit is not influenced by setting the SOR bit. 2. The TOR bit is set, when the time from the last valid TRIGGER event to the next active TRIGGER slope exceeds the value of the last nominal TRIGGER duration multiplied with the value of the TLR register (see **Section 38.22.22.36, DPLL_TLR**) and is reset, when at the current or last active input event a direction change was detected. The SYT bit is not influenced by setting the TOR bit 3. The DPLL_STATUS register is reset, when the DPLL is disabled (switching DEN from 1 to 0).

Table 38.325 DPLL_STATUS Register Contents (2/5)

Bit Position	Bit Name	Function
30	LOCK1	DPLL Lock status concerning SUB_INC1 0: The DPLL is not locked for TRIGGER M– 0 or SMC=1) or SM– 1) and RM– 1) 1: The DPLL is locked for TRIGGER M– 0 or SMC=1) or SM– 1) and RM– 1) **NOTES** LOCK1 is set: 1. In normal mode (for RMO=SMC=0, LCD=0): Bit is set for an active TRIGGER event when SYT is set and the number of events between two gaps is as expected by the profile (NT values in the ADT_T[i] field) or when SYN_NT=0 and SYT=1. 2. In normal mode (for RMO=SMC=0, LCD=1): Bit is set for an active TRIGGER event when SYT is set and the number of events between two increments without missing TRIGGER (no gap) is as expected by the profile (NT values in the ADT_T[i] field). 3. In emergency mode (for RMO=1 and SMC=0): Bit is set for an active STATE event, when SYS is set and the received events are in correspondence to the profile (NS values in the ADT_S[i] field) for at least two (four in case of direction change) expected missing STATE events or when SYN_NS=0. 4. For SMC=1: Bit is set for an active TRIGGER even when SYT is set and SYN_NT=0 or when SYT is set and the profile stored in the ADT_T[i] field matches once between two gaps. LOCK1 is reset for RMO=SMC=0: 5. When a corresponding missing TRIGGER event occurs while SYN_T=1. This does mean an unexpected missing TRIGGER. 6. When the corresponding input signal TRIGGER is out of locking range TLR, 7. When a corresponding direction change is detected for RMO=1 and SMC=0: 8. When a corresponding missing STATE event occurs while SYN_S=1. This does mean an unexpected missing STATE. 9. When the corresponding input signal STATE is out of locking range TLR for SMC=1: 10. When a corresponding missing TRIGGER event occurs while SYN_T=1. This does mean an unexpected missing TRIGGER. 11. When the corresponding input signal TRIGGER is out of locking range TLR. 12. When a corresponding direction change is detected
29	FTD	First TRIGGER detected. 0: No active TRIGGER event was detected after enabling DPLL 1: At least one active TRIGGER event was detected after enabling DPLL **NOTE** No change of FTD for switching from normal to emergency mode or vice versa.
28	FSD	First STATE detected. 0: Still no active STATE event was detected after enabling DPLL 1: At least one active STATE event was detected after enabling DPLL **NOTE** No change of FSD for switching from normal to emergency mode or vice versa.

Table 38.325 DPLL_STATUS Register Contents (3/5)

Bit Position	Bit Name	Function
27	SYT	Synchronization condition of TRIGGER fixed. This bit is set when the CPU writes to the APT_2C address pointer.
26	SYS	Synchronization condition of STATE fixed. This bit is set when the CPU writes to the APS_1C3 address pointer.
25	LOCK2	DPLL Lock status concerning SUB_INC2 0: The DPLL is not locked concerning STATE for SMC=1 1: The DPLL is locked concerning STATE for SMC=1 **NOTES** 1. Locking of SUB_INC2 appears for RMO=SMC=1: Bit is set, when SYS is set and the number of events between two missing STATE s is as expected by the SYN_S values. 2. LOCK2 is set for SMC=RMO=1: for an active STATE event when SYS is set and SYN_NS=0 or when SYS is set and the profile stored in the ADT_Si field matches once between two gaps. LOCK2 is reset: for SMC=RMO=1 – when a missing STATE event occurs while SYN_S=1. This does mean an unexpected missing STATE. – when the corresponding input signal STATE is out of locking range SLR
24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	BWD1	Backwards drive of SUB_INC1 0: Forward direction 1: Backward direction
22	BWD2	Backwards drive of SUB_INC2 0: Forward direction 1: Backward direction
21	ITN	Increment number of TRIGGER is not plausible; Bit is set when the number of TRIGGERS is different to profile 0: The number of TRIGGER events between synchronization gaps is plausible, a direction change is detected or the address pointer APT_2C is written 1: After setting LOCK1 in normal mode (for SMC=0 or SMC=1) or in emergency mode (only for SMC=0) for missing or additional TRIGGER signals detected; bit is cleared when a direction change is detected or the APT_2C is written
20	ISN	Increment number of STATE is not plausible; Bit is set when the number of STATES is different to profile 0: The number of STATE events between synchronization gaps is plausible, a direction change is detected or the APS_1C3 pointer is written 1: After setting LOCK1 in emergency mode (SMC=0 and RMO=1) or LOCK2 for SMC=RMO=1 missing or additional STATE signals detected; bit is cleared when a direction change is detected or the APS_1C3 is written
19	CAIP1	Calculation of lower half actions in progress 0: Currently no action calculation, new data requests possible 1: Action calculation in progress, no new data requests possible
18	CAIP2	Calculation of upper half actions in progress 0: Currently no action calculation, new data requests possible 1: Action calculation in progress, no new data requests possible
17	CSVT	Current signal value TRIGGER 0: The last TRIGGER_S value was 0 1: The last TRIGGER_S value was 1
16	CSVS	Current signal value STATE 0: The last STATE_S value was 0 1: The last STATE_S value was 1

Table 38.325 DPLL_STATUS Register Contents (4/5)

Bit Position	Bit Name	Function
15	LOW_RES	low resolution of TBU_TS0 is used for DPLL input; this value reflects the input signal LOW_RES 0: The lower 24 Bits of TBU_TS0 are used as input for the DPLL 1: The higher 24 Bits of TBU_TS0 are used as input for the DPLL
14, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	RAM2_ERR	DPLL internal access to not configured RAM2 memory space 0: No access to not configured RAM2 memory space 1: Access to not configured RAM2 memory space
11	MT	Missing TRIGGER detected according to TOV 0: No missing TRIGGER detected or a new active TRIGGER slope occurred 1: At least one missing TRIGGER detected after the last active slope
10	TOR	TRIGGER out of range 0: All TRIGGER signal events appear within TLR interval or a direction change was detected 1: At least one TRIGGER signal event is out of TLR;
9	MS	Missing STATE detected according to SOV. 0: No missing STATE detected or a new active STATE slope occurred 1: At least one missing STATE detected after the last active slope
8	SOR	STATE out of range 0: All STATE signal events appear within SLR interval or a direction change was detected 1: At least one STATE signal event is out of SLR;
7	PSE	Prediction space configuration error 0: No prediction space error detected 1: Configured offset value of RAM2 is too small in order to store all TNU+1 values twice in FULL_SCALE
6	RCT	Resolution conflict TRIGGER. Bit is reset, when condition is changed accordingly 0: No resolution conflict detected 1: the TS0_HRT value is set to 1 while LOW_RES=0
5	RCS	Resolution conflict STATE. Bit is reset, when condition is changed accordingly 0: No resolution conflict detected 1: the TS0_HRS value is set to 1 while LOW_RES=0
4	CRO	Calculated Reciprocal value overflow; Bit is set when the calculation of RDT_T_ACT or RDT_S_ACT leads to an overflow 0: No overflow at any reciprocal calculation 1: overflow for at least one reciprocal calculation **NOTE** An overflow in calculation of reciprocal values can occur, when the condition of (DPLL_10979)*[XREF TARGET " (DPLL_10979)] NOT EXIST* is violated. Such an overflow can occur according to the calculations in equations (DPLL_6912)*[XREF TARGET " (DPLL_6912)] NOT EXIST* or (DPLL_6918)*[XREF TARGET " (DPLL_6918)] NOT EXIST*. The overflow is detected when after the calculation and shifting left 32 bits at least one of the bits 31 to 24 is not zero. In that case the corresponding register is set to $FFFFFF_H$.

Table 38.325 DPLL_STATUS Register Contents (5/5)

Bit Position	Bit Name	Function
3	CTO	Calculated TRIGGER duration overflow; Bit is set when equations {REF:DPLL_11005} or {REF:11016} lead to an overflow 0: No overflow at equation {REF:DPLL_11005} or {REF:DPLL_11016} 1: Overflow at equation {REF:DPLL_11005} or {REF:DPLL_11016} **NOTE** When one of the above bits is set the corresponding register contains the maximum value $FFFFFF_H$.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSO	Calculated STATE duration overflow; Bit is set when equations {REF:DPLL_11036} or {REF:DPLL_11045} lead to an overflow 0: No overflow at equations {REF:DPLL_11036} or {REF:DPLL_11045} 1: Overflow at equations {REF:DPLL_11036} or {REF:DPLL_11045}
0	FPCE	Fast pulse correction error 0: No error at fast pulse correction detected 1: Negative value of MPVAL1/2 used for fast pulse correction mode

Note: Note: Only the values characterized by 1) are stored for an active TRIGGER slope. All other values remain 0. When DEN=0 the relevant bit values of the original register DPLL_CTRL_0 are transferred without any input event at the next system clock. This results in the above reset value.

38.22.18.31 DPLL_ID_PMTR_[z]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times z + 28100_H$

Value after reset: 0000 $01FE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ID_PMTR_X								
Value after reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.326 DPLL_ID_PMTR_[z] Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	ID_PMTR_X	ID information to the input signal PMTR[i] from the ARU. **NOTE** This value can only be written when the action [i] is disabled by the correspondent bit AENi=0 of the registers DPLL_CTRL_2, ...5 respectively or when the DPLL is disabled (DEN=0).

38.22.18.32 DPLL_CTRL_0_SHADOW_TRIGGER

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $281E0_H$

Value after reset: $0000\ 0257_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMO	—	—	IDT	—	AMT	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IFP	MLT									
Value after reset	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.327 DPLL_CTRL_0_SHADOW_TRIGGER Register Contents

Bit Position	Bit Name	Function
31	RMO [1)]	Reference mode; selection of the relevant the input signal for generation of SUB_INC1.
30, 29	Reserved	When read, the value after reset is returned.
28	IDT [1)]	Input delay TRIGGER; use of input delay information transmitted in FT part of the TRIGGER signal.
27	Reserved	When read, the value after reset is returned.
26	AMT [1)]	Adapt mode TRIGGER; Use of adaptation information of TRIGGER.
25 to 11	Reserved	When read, the value after reset is returned.
10	IFP [1)]	Input filter position; value contains position or time related information.
9 to 0	MLT [1)]	Multiplier for TRIGGER; MLT+1 is number of SUB_INC1 pulses between two TRIGGER events in normal mode (1...1024);

Note: Only the values characterized by 1) are stored for an active TRIGGER slope. All other values remain 0. When DEN=0 the relevant bit values of the original register DPLL_CTRL_0 are transferred without any input event at the next system clock. This results in the above reset value.

38.22.18.33 DPLL_CTRL_0_SHADOW_STATE

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $281E4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMO	—	—	—	IDS	—	AMS	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IFP	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.328 DPLL_CTRL_0_SHADOW_STATE Register Contents

Bit Position	Bit Name	Function
31	RMO [1)]	Reference mode; selection of the relevant the input signal for generation of SUB_INC1.
30 to 28	Reserved	When read, the value after reset is returned.
27	IDS [1)]	Input delay STATE; Use of input delay information transmitted in FT part of the STATE signal.
26	Reserved	When read, the value after reset is returned.
25	AMS [1)]	Adapt mode STATE; Use of adaptation information of STATE.
24 to 11	Reserved	When read, the value after reset is returned.
10	IFP [1)]	Input filter position; value contains position or time related information.
9 to 0	Reserved	When read, the value after reset is returned.

Note: Only the values characterized by 2) are stored for an active STATE slope. All other values remain 0. When DEN=0 the relevant bit values of the original register DPLL_CTRL_0 are transferred without any input event at the next system clock.

38.22.18.34 DPLL_CTRL_1_SHADOW_TRIGGER

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $281E8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCM1	DLM1	SGE1	PIT	COA	—	—	DMO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.329 DPLL_CTRL_1_SHADOW_TRIGGER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	PCM1	Pulse Correction Mode for SUB_INC1 generation.
6	DLM1	Direct Load Mode for SUB_INC1 generation
5	SGE1	SUB_INC1 generator enable.
4	PIT	Plausibility value PVT to next active TRIGGER is time related
3	COA	Correction strategy in automatic end mode (DMO=0).
2, 1	Reserved	When read, the value after reset is returned.
0	DMO	DPLL mode select.

Note: Only the values characterized by DMO, COA, PIT, SGE1, DLM1 and PCM1 are stored for an active TRIGGER slope. All other values remain 0. When DEN=0 the relevant bit values of the original register DPLL_CTRL_1 are transferred without any input event at the next system clock.

38.22.18.35 DPLL_CTRL_1_SHADOW_STATE

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $281EC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PCM2	DLM2	SGE2	PCM1	DLM1	SGE1	—	COA	—	—	DMO
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.330 DPLL_CTRL_1_SHADOW_STATE Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	PCM2	Pulse Correction Mode for SUB_INC2 generation.
9	DLM2	Direct Load Mode for SUB_INC2 generation
8	SGE2	SUB_INC2 generator enable.
7	PCM1	Pulse Correction Mode for SUB_INC1 generation.
6	DLM1	Direct Load Mode for SUB_INC1 generation
5	SGE1	SUB_INC1 generator enable.
4	Reserved	When read, the value after reset is returned.
3	COA	Correction strategy in automatic end mode (DMO=0).
2, 1	Reserved	When read, the value after reset is returned.
0	DMO	DPLL mode select.

Note: Only the values characterized by DMO, COA, SGE1, DLM1, PCM1, SGE2, DLM2 and PCM2 are stored for an active STATE slope. All other values remain 0. When DEN=0 the relevant bit values of the original register DPLL_CTRL_1 are transferred without any input event at the next system clock.

38.22.18.36 DPLL_RAM_INI

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $281FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	INIT_RAM	—	INIT_2	INIT_1BC	INIT_1A
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Table 38.331 DPLL_RAM_INI Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	INIT_RAM	RAM regions 1a, 1b and 2 are to be initialized. 0: Do not start initialization of all RAM regions 1: Start initialization of all RAM regions **NOTES** 1. Setting the INIT_RAM bit results only in a RAM reset when the DPLL is not enabled (DEN=0). 2. Depending on the vendor configuration the connected RAM regions are initialized to zero in the case of a module HW reset or for setting the RST bit in the GTM_RST register. 3. In the case of no RAM initialization it must be ensured that all relevant parameters are configured correctly. Otherwise there is no guarantee to get a predictable behavior.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	INIT_2	RAM region 2 initialization in progress 0: No initialization of considered RAM region in progress 1: Initialization of considered RAM region in progress
1	INIT_1BC	RAM region 1b and 1c initialization in progress 0: No initialization of considered RAM region in progress 1: Initialization of considered RAM region in progress
0	INIT_1A	RAM region 1a initialization in progress 0: No initialization of considered RAM region in progress 1: Initialization of considered RAM region in progress

38.22.18.37 DPLL_TSAC[z]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28E00_H$

Value after reset: $007F\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSAC							
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSAC															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.332 DPLL_TSAC[z] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TSAC	Calculated time stamp for ACTION z (z = 0...NOAC–1)1) **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.38 DPLL_PSAC[z]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times z + 28E80_H$

Value after reset: $007F\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSAC							
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSAC															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.333 DPLL_PSAC[z] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSAC	Calculated position value for the start of ACTION z in normal or emergency mode according to equations DPLL-17 or DPLL-20 respectively (z = 0...NOAC–1). **NOTE** This value can only be written when the DPLL is disabled.

38.22.18.39 DPLL_ACB_[z]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times z + 28F00_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ACB_3					—	—	—	ACB_2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	ACB_1					—	—	—	ACB_0				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.334 DPLL_ACB_[z] Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 24	ACB_3	Action Control Bits of ACTION_i, reflects ACT_D[i](52:48), i=4*z **NOTES** 1. When DPLL_CTRL_11.ACBU = '0': ACB_3[4:0] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. When DPLL_CTRL_11.ACBU = '1': ACB_3[4:2] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. ACB_3[1]= '1' is used as input signal to control if "action in past" shall be checked based on position information. ACB_3[1] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. ACB_3[0] is used as input signal to control if "action in past" shall be checked based on time information. ACB_3[0] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. 2. This value can only be written via AEI-interface when the DPLL is disabled.
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.334 DPLL_ACB_[z] Register Contents (2/2)

Bit Position	Bit Name	Function
20 to 16	ACB_2	Action Control Bits of ACTION_i, reflects ACT_D[i](52:48), i=4*z **NOTES** 1. When DPLL_CTRL_11.ACBU = '0': ACB_2[4:0] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. When DPLL_CTRL_11.ACBU = '1': ACB_2[4:2] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. ACB_2[1]= '1' is used as input signal to control if "action in past" shall be checked based on position information. ACB_2[1] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. ACB_2[0] is used as input signal to control if "action in past" shall be checked based on time information. ACB_2[0] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. 2. This value can only be written via AEI-interface when the DPLL is disabled.
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	ACB_1	Action Control Bits of ACTION_i, reflects ACT_D[i](52:48), i=4*z **NOTES** 1. When DPLL_CTRL_11.ACBU = '0': ACB_1[4:0] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. When DPLL_CTRL_11.ACBU = '1': ACB_1[4:2] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. ACB_1[1]= '1' is used as input signal to control if "action in past" shall be checked based on position information. ACB_1[1] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. ACB_1[0] is used as input signal to control if "action in past" shall be checked based on time information. ACB_1[0] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. 2. This value can only be written via AEI-interface when the DPLL is disabled.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	ACB_0	Action Control Bits of ACTION_i, reflects ACT_D[i](52:48), i=4*z

Note: When DPLL_CTRL_11.ACBU = '0': ACB_0[4:0] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. When DPLL_CTRL_11.ACBU = '1': ACB_0[4:2] are taken as received by ARU interface and are transmitted unchanged as result of action (PMT) calculation. ACB_0[1]= '1' is used as input signal to control if "action in past" shall be checked based on position information. ACB_0[1] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far. ACB_0[0] is used as input signal to control if "action in past" shall be checked based on time information. ACB_0[0] is written to '1' if action channel has reached "action in past" condition after action has been calculated, written to '0' if action has not reached "past" so far.

This value can only be written via AEI-interface when the DPLL is disabled.

38.22.18.40 DPLL_CTRL_11

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F20_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WACBU	WSTATE_EXT	WPCMF2_INCCNT_B	WINCF2	WFSYL2	WPCMF2	WERZ2	WSIP2	WADS	WADT	WPCMF1_INCCNT_B	WINCF1	WFSYL1	WPCMF1	WERZ1	WSIP1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACBU	STATE_EXT	PCMF2_INCCNT_B	INCF2	FSYL2	PCMF2	ERZ2	SIP2	ADS	ADT	PCMF1_INCCNT_B	INCF1	FSYL1	PCMF1	ERZ1	SIP1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.335 DPLL_CTRL_11 Register Contents (1/7)

Bit Position	Bit Name	Function
31	WACBU	Write enable for ACB use; the ACB values of PMTR are used to decide if an action is in the past 0: Writing to ACBU is not enabled. 1: Writing to ACBU is enabled. **Note:** Enable writing.
30	WSTATE_EXT	Write enable of STATE_EXT 0: Writing to STATE_EXT is not enabled. 1: Writing to STATE_EXT is enabled.
29	WPCMF2_INCCNT_B	Write enable of PCMF2_INCCNT_B 0: Writing to PCMF2_INCCNT_B is not enabled. 1: Writing to PCMF2_INCCNT_B is enabled.
28	WINCF2	Write enable for INC_CNT2 fast 0: Writing to INCF2 is not enabled 1: Writing to INCF2 is enabled. **Note:** Enable writing.
27	WFSYL2	Write enable for Force Synchronization Loss 2. 0: Writing to FSYL2 is not enabled. 1: Writing to FSYL2 is enabled **Note:** Enable writing.
26	WPCMF2	Write enable for pulse correction mode fast 2 0: Writing to PCMF2 is not enabled 1: Writing to PCMF2 is enabled **Note:** Enable writing.
25	WERZ2	Write enable for error zero 2. 0: Writing to ERZ2 is not enabled 1: Writing to ERZ2 is enabled. **Note:** Enable writing.
24	WSIP2	Write enable for simplified increment prediction 2. 0: Writing to SIP2 is not enabled. 1: Writing to SIP2 is enabled. **Note:** Enable writing.
23	WADS	Write enable of ADS 0: Writing to ADS is not enabled. 1: Writing to ADS is enabled.

Table 38.335 DPLL_CTRL_11 Register Contents (2/7)

Bit Position	Bit Name	Function
22	WADT	Write enable of ADT 0: Writing to ADT is not enabled. 1: Writing to ADT is enabled.
21	WPCMF1_INCCNT_B	Write enable of PCMF1_INCCNT_B 0: Writing to PCMF1_INCCNT_B is not enabled. 1: Writing to PCMF1_INCCNT_B is enabled.
20	WINCF1	Write enable for INC_CNT1 fast 0: Writing to INCF1 is not enabled. 1: Writing to INCF1 is enabled. **Note:** Enable writing
19	WFSYL1	Write enable for Force Synchronization Loss 1. 0: Writing to FSYL1 is not enabled. 1: Writing to FSYL1 is enabled. **Note:** Enable writing.
18	WPCMF1	Write enable for pulse correction mode fast 1 0: Writing to PCM1 is not enabled. 1: Writing to PCM1 is enabled. **Note:** Enable writing.
17	WERZ1	Write enable for error zero 1. 0: Writing to ERZ1 is not enabled. 1: Writing to ERZ1 is enabled. **Note:** Enable writing.
16	WSIP1	Write enable for simplified increment prediction 1. 0: Writing to SIP1 is not enabled. 1: Writing to SIP1 is enabled. **Note:** Enable writing.
15	ACBU	ACB use; the ACB values of PMTR are used to decide if an action is in the past 0: ACB values of PMTR are not considered in DPLL; the decision if an action is in the past is made considering the calculated time value. 1: ACB values of PMTR are considered in DPLL as follows: if ACB[1] = 1, consider if the calculated position value of the corresponding action is in the past. if ACB[0] = 1, consider if the calculated time value of the corresponding action is in the past. ACB[1] and ACB[0] can be set also simultaneously to 1 **NOTE** Return ACB values together with actions as zero, when the actions are in future; set ACB[1]=1, when calculated position value is in the past and the ACB[1] of PMTR was 1. Set ACB[0]=1, when calculated time value is in the past and the ACB[0] of PMTR was 1. The value of ACBU can be only written when WACBU=1.
14	STATE_EXT	Use of STATE engine extension 0: STATE extension is not considered. 1: STATE extension is enabled up to 128 STATE events **NOTE** The STATE_EXT value can be only written when WSTATE_EXT=1 and the DPLL is disabled. See **Section 38.22.10** for a further explanation. If this bit shall be modified during operation a software reset of the DPLL module is strongly recommended. A RAM initialization should also be considered depending on the given application case.

Table 38.335 DPLL_CTRL_11 Register Contents (3/7)

Bit Position	Bit Name	Function
13	PCMF2_INCCNT_B	No increment of INC_CNT2 when PCMF2 active (automatic end mode). 0: Add MPVAL2 value is as well added to the INC_CNT2 when fast pulse correction is done by PCM2 or PCMF2. 1: Do not add MPVAL2 value to the INC_CNT2 register when fast pulse correction is done by PCM2 or PCMF2. This means that just fast pulses are done by decrementing current content of INC_CNT2 register as long as INC_CNT2 is not zero (automatic end mode). The number of pulses (MPVAL2) shall be sufficiently smaller than INC_CNT2 when MPVAL2 is written. **NOTE** The PCMF2_INCCNT_B value can be only written when WPCMF2_INCCNT_B=1.
12	INCF2	INC_CNT2 fast 0: The calculation of a new INC_CNT2 is performed after an active slope was detected and the plausibility check was performed. 1: The calculation of a new INC_CNT2 is prepared before an active slope is detected; the plausibility check is supported by an additional HW checker in order to get the decision earlier and after this decision the pulse generator for SUB_INC2 starts immediately sending out pulses. The calculation of ADD_IN for the SUB_INC generation is performed without adding the 0.5 value to NMB_S in equations DPLL_25 ff. The Signal RESET_SIGx of the pulse generator (see chapter 18.8.3.6) is activated for each new active input slope in order to reset the register values. **NOTE** The INCF2 value can be only written when WINCF2=1.
11	FSYL2	Force Synchronization Loss of LOCK2. 0: No force of synchronization loss. 1: Reset LOCK2, and reset SYS (in emergency mode and for SMC=1). **NOTE** The synchronization loss resets SYS and prevents the use of profiles respectively. The above described effect for FSYL2=1 is only active when WFSYL2=1 simultaneously

Table 38.335 DPLL_CTRL_11 Register Contents (4/7)

Bit Position	Bit Name	Function
10	PCMF2	Pulse correction mode fast for INC_CNT2 0: No fast update of pulses, provided by MPVAL2. 1: When PCM2 is set while PCMF2=1, the pulses provided by MPVAL2 are sent using the rapid pulse generator RPCUx without waiting for a new input event. **NOTE** The fast pulse generation is performed immediately within the current increment. MPVAL2 must be positive integers for the fast pulse correction mode – in the case of negative values the correction is suppressed and the FPCE (fast pulse correction error) bit in the DPLL_STATUS register is set, causing the EI (error interrupt) when enabled. The setting of PCMF2 prevents the transfer of control bits PCM2 to the corresponding shadow registers with an active input event and prevents therefore the distribution of the MPVAL1 values over the current or next increment. The MPVAL2 pulses are sent with the fast clock CMU_CLK0 by the rapid pulse generator RPCUx (see chapter 18.8.3.6 of specification v3.0) triggered in the state 6/26 or 18/38 of the state machines (see chapters 18.8.6.1 and 18.8.6.7 of specification v2.1.0) respectively. The INC_CNT2 is incremented by MPVAL2 respectively. When taken the MPVAL2 value to RPCUx and INC_CNT2 the PCM2 bit is reset immediately and after that also the PCMF2 bit. The value of PCMF2 can be only written when WPCMF2=1. Be careful when using the fast pulse correction during a direction change. Because of sending the correction pulses before, during or after the direction change recognition the result is typically unpredictable. No automatic correction of the fast correction pulses is provided. The necessary corrections must be performed on responsibility of the user.
9	ERZ2	Error is assumed as zero in emergency mode and for the second engine for SMC=1. 0: The MEDT_S value is considered as provided in the corresponding equations. 1: Instead of using MEDT_S the value "0" is used in the corresponding equations. **NOTE** The calculation of EDT_S and MEDT_S is performed independently from the ERZ2 value in all modes without any influence to the MEDT_S value itself. The ERZ2 value influences the use of MEDT_S in emergency mode and for SMC=1 with RMO=1. The value of ERZ2 can be only written when WERZ2=1.
8	SIP2	Simplified increment prediction in emergency mode and for the second engine in the case RMO=1. 0: Increment prediction calculation; the current increment duration CDT_SX is calculated using the relation between increments duration in the past like explained by the corresponding equations. 1: Increment prediction continuation; in this mode for the increment prediction value calculation CDT_SX the value of QDT_S is replaced by 1 for all calculations when NUSE?VSN=1; in the other case the value of SIP2 is ignored and the calculation is performed like for SIP2=0. For the first increment after setting SIP2 from 0 to 1 the value of DT_S_ACT is replaced by the value of the DT_S_START register. This results in a CDT_SX value which is equal to DT_S_START. Notice that this DT_S_START value must be always > 256. **NOTE** The value of SIP2 influences only the increment prediction and error accumulation when NUSE-VSN=1. The calculation of QDT_S itself is not influenced by the SIP2 bit. The value of SIP2 can be only written when WSIP2=1.

Table 38.335 DPLL_CTRL_11 Register Contents (5/7)

Bit Position	Bit Name	Function
7	ADS	Correction of DT_S_ACTUAL, CDT_SX_nom_corr by PD_S 0: No correction of DT_S_ACTUAL, CDT_SX_nom_corr by physical deviation (PD_S) defined in profile of STATE processing unit. 1: Correction of DT_S_ACTUAL, CDT_SX_nom_corr by physical deviation (PD_S) defined in profile of STATE processing unit.
6	ADT	Correction of DT_T_ACTUAL, CDT_TX_nom_corr by PD_T 0: No correction of DT_T_ACTUAL, CDT_TX_nom_corr by physical deviation (PD_T) defined in profile of TRIGGER processing unit. 1: Correction of DT_T_ACTUAL, CDT_TX_nom_corr by physical deviation (PD_T) defined in profile of TRIGGER processing unit.
5	PCMF1_INCCNT_B	No increment of INC_CNT1 when PCMF1 active (automatic end mode). 0: Add MPVAL1 value is as well to the INC_CNT1 when fast pulse correction is done by PCM1 or PCMF1. 1: Do not add MPVAL1 value to the INC_CNT1 register when fast pulse correction is done by PCM1 or PCMF1. This means that just fast pulses are done by decrementing current content of INC_CNT1 register as long as INC_CNT1 is not zero (automatic end mode). The number of pulses (MPVAL1) shall be sufficiently smaller than INC_CNT1 when MPVAL1 is written. **NOTE** The PCMF1_INCCNT_B value can be only written when WPCMF1_INCCNT_B=1.
4	INCF1	INC_CNT1 fast correction 0: The calculation of a new INC_CNT1 is performed after an active slope was detected and the plausibility check was performed. 1: The calculation of a new INC_CNT1 is prepared before an active slope is detected; the plausibility check is supported by an additional HW checker in order to get the decision earlier and after this decision the pulse generator for SUB_INC1 starts immediately sending out pulses The calculation of ADD_IN for the SUB_INC generation is performed without adding the 0.5 value to NMB_T/S in equations DPLL_25 ff. The Signal RESET_SIGx of the pulse generator (see chapter 18.8.3.6) is activated for each new active input slope in order to reset the register values. **NOTE** The INCF1 value can be only written when WINCF1=1. The INCF1 bit should only be written when DPLL_CTRL_1.DEN = '0' (DPLL disabled) to prevent generation of wrong number of sub increments.
3	FSYL1	Force Synchronization Loss of LOCK1. 0: No force of synchronization loss. 1: Reset LOCK1, and reset SYT (in normal mode and for SMC=1) or reset SYS (in emergency mode) **NOTE** The synchronization loss resets SYT/SYS and prevents the use of profiles respectively. The above described effect for FSYL1=1 is only active when WFSYL1=1 simultaneously.

Table 38.335 DPLL_CTRL_11 Register Contents (6/7)

Bit Position	Bit Name	Function
2	PCMF1	Pulse correction mode fast for INC_CNT1 0: No fast update of pulses, provided by MPVAL1. 1: When PCM1 is set while PCMF1=1, the pulses provided by MPVAL1 are sent using the rapid pulse generator RPCUx without waiting for a new input event. **NOTE** The fast pulse generation is performed immediately within the current increment. MPVAL1 must be positive integers for the fast pulse correction mode – in the case of negative values the correction is suppressed and the FPCE (fast pulse correction error) bit in the DPLL_STATUS register is set, causing the EI (error interrupt) when enabled. The setting of PCMF1 prevents the transfer of control bits PCM1 to the corresponding shadow registers with an active input event and prevents therefore the distribution of the MPVAL1 values over the current or next increment. The MPVAL1 pulses are sent with the fast clock CMU_CLK0 by the rapid pulse generator RPCUx (see chapter 18.8.3.6 of specification v3.0) triggered in the state 6/26 or 18/38 of the state machines (see chapters 18.8.6.1 and 18.8.6.7 of specification v2.1.0) respectively. The INC_CNT1 is incremented by MPVAL1 respectively. When taken the MPVAL1 value to RPCUx and INC_CNT1 the PCM1 bit is reset immediately and after that also the PCMF1 bit. The value of PCMF1 can be only written when WPCMF1=1. Be careful when using the fast pulse correction during a direction change. Because of sending the correction pulses before, during or after the direction change recognition the result is typically unpredictable. No automatic correction of the fast correction pulses is provided. The necessary corrections must be performed on responsibility of the user.
1	ERZ1	Error is assumed as zero in normal mode and for the first engine for SMC=1. 0: The MEDT_T value is considered as provided in the corresponding equations. 1: Instead of using MEDT_T the value "0" is used in the corresponding equations. **NOTE** The calculation of EDT_T and MEDT_T is performed independently from the ERZ1 value in all modes without any influence to the MEDT_T value itself. The ERZ1 value influences the use of MEDT_T in normal mode and for SMC=1. The value of ERZ1 can be only written when WERZ1=1.

Table 38.335 DPLL_CTRL_11 Register Contents (7/7)

Bit Position	Bit Name	Function
0	SIP1	Simplified increment prediction in normal mode and for the first engine in the case SMC=1. 0: Increment prediction calculation; the current increment duration CDT_TX is calculated using the relation between increment duration in the past like explained by the corresponding equations. 1: Increment prediction continuation; in this mode for the increment prediction value calculation of CDT_TX the value of QDT_T is replaced by 1 for all calculations when NUTE–VTN=1; in the other case the value of SIP1 is ignored and the calculation is performed like for SIP1=0. For the first increment after setting SIP1 from 0 to 1 the value of DT_T_ACT is replaced by the value of the DT_T_START register. This results in a CDT_TX value which is equal to DT_T_START. Notice that this DT_T_Start value must be always > 256. **NOTES** 1. The value of SIP1 influences only the increment prediction CDT_TX and when NUTE–VTN=1. The calculation of QDT_T itself is not influenced by the SIP1 bit. The value of SIP1 can be only written when WSIP1=1. When SIP1=1 is set the first pulses of the subincrement generator are not generated with highest frequency for the first increment (DPLL_STATUS.ftd = 0, DPLL_CTRL_1.SGE1=1). 2. When SIP1=1 is set the first pulses of the subincrement generator are not generated with highest frequency for the first increment (DPLL_STATUS.ftd=0, DPLL_CTRL_1.SGE1=1).

38.22.18.41 DPLL_THVAL2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $28F24_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	THVAL							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THVAL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.336 DPLL_THVAL2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	THVAL	Measured last pulse time from active to inactive slope of TRIGGER after correction of input slope filter delays. **NOTE** This value is available immediately after the inactive slope of TRIGGER. The measured value considers all input slope filter delays. From the received input the corresponding filter delays are subtracted before the time stamp difference of active and inactive slope is calculated.

38.22.18.42 DPLL_TIDEL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F28_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TIDEL							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIDEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.337 DPLL_TIDEL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TIDEL	TRIGGER input delay Transmit this value with each active TRIGGER slope into a shadow register. Subtract this shadow register value from each TRIGGER time stamp (active and inactive slope). This feature is always active and cannot be disabled by a control bit

38.22.18.43 DPLL_SIDEL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F2C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SIDEL							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SIDEL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.338 DPLL_SIDEL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	SIDEL	STATE input delay Transmit this value with each active STATE slope into a shadow register. Subtract this shadow register value from each STATE time stamp (active and inactive slope). This feature is always active and cannot be disabled by a control bit.

38.22.18.44 DPLL_CTN_MIN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F6C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CTN_MIN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTN_MIN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.339 DPLL_CTN_MIN Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CTN_MIN	CDT_T_NOM min value Use this register value as CDT_TX_NOM value when the calculated value for the nominal increment prediction of TRIGGER is less than the register value

38.22.18.45 DPLL_CTN_MAX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F70_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CTN_MAX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTN_MAX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.340 DPLL_CTN_MAX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CTN_MAX	CDT_T_NOM max value Use this register value as CDT_TX_NOM value when the calculated value for the nominal increment prediction of TRIGGER is greater than the register value

38.22.18.46 DPLL_CSN_MIN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F74_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CSN_MIN							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSN_MIN															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.341 DPLL_CSN_MIN Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CSN_MIN	CDT_S_NOM min value Use this register value as CDT_SX_NOM value when the calculated value for the nominal increment prediction of STATE is less than the register value

38.22.18.47 DPLL_CSN_MAX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F78_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CSN_MAX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSN_MAX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.342 DPLL_CSN_MAX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CSN_MAX	CDT_S_NOM max value Use this register value as CDT_SX_NOM value when the calculated value for the nominal increment prediction of STATE is greater than the register value

38.22.18.48 DPLL_STA

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $28F40_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CNT_S			—	STA_S			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STA_S				CNT_T			—	STA_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.343 DPLL_STA Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 21	CNT_S	Count STATE; this reflects the count of active STATE slopes (mod8). This value shows the number of active STATE slopes (mod8) **NOTE** This value allows distinguishing if the above state machine status is consistent to other status values read before or after it.
20	Reserved	When read, the value after reset is returned.
19 to 12	STA_S	Status of STATE state machine; state binary coded This bit field reflects the status of the STATE state machine See **Table 38.345, STA_S bit**. **NOTE** The decimal step number 21 to 40 of the state machine is binary coded from 0x01 to 0x14 respectively using the upper 5 bits (19:15) after subtraction of 20 to the decimal value. The lower 3 bits (14:12) show substates of the corresponding state machine. When the DPLL is disabled this field is 0x000.
11 to 9	CNT_T	Count TRIGGER; this reflects the count of active TRIGGER slopes (mod8). This value shows the number of active TRIGGER slopes (mod8) **NOTE** This value allows distinguishing if the above state machine status is consistent to other status values read before or after it.
8	Reserved	When read, the value after reset is returned.

Table 38.343 DPLL_STA Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 0	STA_T	Status of TRIGGER state machine; state binary coded This bit field reflects the status of the TRIGGER state machine See **Table 38.344, STA_T bit**. **NOTE** The decimal step number 1 to 20 of the state machine is binary coded from 0x01 to 0x14 respectively using the upper 5 bits (7:3). The lower 3 bit (2:0) show substates of the corresponding state machine. When the DPLL is disabled this field is 0x000.

Table 38.344 STA_T bit (1/2)

STA_T(7:3)	STA_T(2:0)	Description/ Monitored action
0	0	Reset state
0	1	Wait, DEN=0
0	2	Calculation of 1/mlt+1, mls1, mls2.
0	3	calculation of direction change issues (pointers and profile update)
0	4	APT_2C is being incremented in normal mode
0	5	APT_2C was incremented 3 times in normal mode, 1 in emergency or SMC=1
0	6	APT_2C was incremented 4 times in normal mode, 2 in emergency or SMC=1
0	7	Update of pointers is finished, perform change of direction operations
1	0	pvt-check
1	1	update of RAM: write RDT_T; DT_T; TSF_T
1	2	loading of profile (syn_t, update syn_t_old) from ADT_T
1	3	TASI-irq, store FTV into RAM1b
1	4	Write PSTC; modify apt, apt_2b; apt_2c (if synchronized); Start fast pulse updates if necessary; Update inc_cnt1;
2	0	Write TS_T to ram1b, calculate dt_t_actual
3	0	update nti_cnt, cdti-irq if nti_cnt=0;
3	1	calculated EDT_T, MEDT_T, RDT_T_actual
4	0	calculate cdt_tx_nom, cdt_tx
5	0	calculate PSTM, rcdt_t, nmb_t_tar, start fast correction of missing pulses (if necessary) for rmo=0 or smc=1.
6	0	calculate nmb_t for rmo=0 or smc=1, dmo=0, coa=0.
7	0	calculate nmb_t for rmo=0 or smc=1, dmo=0, coa=1.
8	0	calculate nmb_t for rmo=0 or smc=1, dmo=1.
9	0	
10	0	calculate add_in_cal1
10	1	write of add_in_cal1 finished, all subincrement calculations done for last active input event
11	0	calculate ts_t_check (MTI-irq), r_add_caln (prepare time stamp calculation (TS_T)) for IDT=IFP=1.
12	0	set caip1,2, action masking bits, action calculation loop control.
13	0	calculate NA(i),
14	0	calculate PDT_T(i)
14	1	calculate DTA(i)

Table 38.344 STA_T bit (2/2)

STA_T(7:3)	STA_T(2:0)	Description/ Monitored action
15	0	calculate TSAC(i)
15	1	calculate PSAC(i)
15	2	action(i) in past condition occurred: assignment of output data.
15	3	action loop control
16	0	wait for new action calculation

Table 38.345 STA_S bit (1/2)

STA_S(7:3)	STA_S(2:0)	Description/ Monitored action
0	0	Reset state
0	1	Wait, DEN=0
0	2	Calculation of 1/mlt+1, mls1, mls2.
0	3	Calculation of direction change issues (pointers and profile update)
0	4	--
0	5	APS_1c3 was incremented once
0	6	APS_1c3 was incremented twice
0	7	Update of pointers is finished, perform change of direction operations
1	0	pvt-check
1	1	update of RAM: write RDT_S; DT_S; TSF_S
1	2	loading of profile (syn_s, update syn_s_old) from ADT_S
1	3	SASI-irq, store FTV into RAM1b
1	4	Write PSSC; modify aps, aps_1c2; aps_1c3 (if synchronized); Start fast pulse updates if necessary; Update inc_cnt1/2;
2	0	Write TS_S to ram1b, calculate dt_s_actual
3	0	update cdsi-irq
3	1	calculate EDT_S, MEDT_S, RDT_S_actual
4	0	calculate cdt_sx_nom, cdt_sx
5	0	calculate PSSM, rcdt_s, nmb_s_tar, start fast correction of missing pulses (if necessary).
6	0	calculate nmb_s for rmo=1 or smc=1, dmo=0, coa=0.
7	0	calculate nmb_s for rmo=1 or smc=1, dmo=0, coa=1.
8	0	calculate nmb_t for rmo=1 or smc=1, dmo=1.
9	0	
10	0	calculate add_in_cal1
10	1	write of add_in_cal1 finished, all subincrement calulations done for last active input event
11	0	calculate ts_s_check (MSI-irq), r_add_caln (prepare time stamp calculation(TS_S)) for IDT=IFP=1.
12	0	set caip1,2, action masking bits, action calculation loop
13	0	calculate NA(i),
14	0	calculate PDT_S(i)
14	1	calculate DTA(i)
15	0	calculate TSAC(i)
15	1	calculate PSAC(i)

Table 38.345 STA_S bit (2/2)

STA_S(7:3)	STA_S(2:0)	Description/ Monitored action
15	2	action(i) in past condition occurred: assignment of output data.
15	3	action loop control
16	0	wait for new action calculation

38.22.18.49 DPLL_INCF1_OFFSET

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F44_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DPLL_INCF1_OFFSET							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPLL_INCF1_OFFSET															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.346 DPLL_INCF1_OFFSET Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DPLL_INCF1_OFFSET	Start value of the ADD_IN_ADDER1 In the case of set DPLL_CTRL_11–INCF1 the ADD_IN_ADDER1 starts always after an active new input event (TRIGGER in normal mode or STATE in emergency mode respectively) with this offset value. In the case of choosing DPLL_INCF1_OFFSET=$FFFFFF_H$ the generation of the first SUB_INC1 pulse is performed with the next TS_CLK. In the case of DPLL_INCF1_OFFSET=000000_H the first pulse is delayed by a full SUB_INC1 period and in the case of DPLL_INCF1_OFFSET=$7FFFFF_H$ the first pulse is delayed by a half SUB_INC1 period. Any other value is possible.

38.22.18.50 DPLL_INCF2_OFFSET

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F48_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DPLL_INCF2_OFFSET							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPLL_INCF2_OFFSET															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.347 DPLL_INCF2_OFFSET Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DPLL_INCF2_OFFSET	Start value of the ADD_IN_ADDER2 In the case of set DPLL_CTRL_11–INCF2 the ADD_IN_ADDER2 starts always after an active new input event (STATE) with this offset value. In the case of choosing DPLL_INCF2_OFFSET=$FFFFFF_H$ the generation of the first SUB_INC2 pulse is performed with the next TS_CLK. In the case of DPLL_INCF2_OFFSET=000000_H the first pulse is delayed by a full SUB_INC2 period and in the case of DPLL_INCF2_OFFSET=$7FFFFF_H$ the first pulse is delayed by a half SUB_INC2 period. Any other value is possible.

38.22.18.51 DPLL_DT_T_START

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F4C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DPLL_DT_T_START							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPLL_DT_T_START															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.348 DPLL_DT_T_START Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DPLL_DT_T_START	Start value of DPLL_DT_T_ACT for the first increment after SIP1 is set to 1 For the first increment after setting SIP1 from 0 to 1 the value of DPLL_DT_T_START is taken instead of the calculated DPLL_DT_T_ACT for the current increment duration. This value should be always > 256 in order to avoid an overflow during the calculation of DPLL_RDT_T_ACT.

38.22.18.52 DPLL_DT_S_START

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F50_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DPLL_DT_S_START							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPLL_DT_S_START															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.349 DPLL_DT_S_START Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DPLL_DT_S_START	Start value of DPLL_DT_S_ACT for the first increment after SIP2 is set to 1 For the first increment after setting SIP2 from 0 to 1 the value of DPLL_DT_S_START is taken instead of the calculated DPLL_DT_S_ACT for the current increment duration. This value should be always > 256 in order to avoid an overflow during the calculation of DPLL_RDT_S_ACT.

38.22.18.53 DPLL_STA_MASK

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F54_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STA_NOTIFY_S								STA_NOTIFY_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.350 DPLL_STA_MASK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	STA_NOTIFY_S	Notify value for STA_S of register DPLL_STA. The STA_NOTIFY_S is representing a trigger mask of DPLL_STA.STA_S. When DPLL_STA.STA_S reaches the value of STA_NOTIFY_S the flag DPLL_STA_FLAG.STA_FLAG_S is set to '1' when DPLL_STA.STA_S is leaving the state STA_NOTIFY_S.
7 to 0	STA_NOTIFY_T	Notify value for STA_T of register DPLL_STA. The STA_NOTIFY_T is representing a trigger mask of DPLL_STA.STA_T. When DPLL_STA.STA_T reaches the value of STA_NOTIFY_T the flag DPLL_STA_FLAG.STA_FLAG_T is set to '1' when DPLL_STA.STA_T is leaving the state STA_NOTIFY_T. The signal is visible to MCS0 sub module as part of the special function register.

38.22.18.54 DPLL_STA_FLAG

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F58_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	INC_CNT2_FLAG	INC_CNT1_FLAG	STA_FLAG_S	—	—	—	—	—	—	—	STA_FLAG_T
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 38.351 DPLL_STA_FLAG Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	INC_CNT2_FLAG	Flag according to DPLL_INC_CNT2_MASK.INC_CNT2_NOTIFY The INC_CNT2_FLAG is set to '1' indicating that the signal DPLL_INC_CNT2.INC_CNT2 has left the state defined by the trigger mask of DPLL_INC_CNT2_MASK.INC_CNT2_NOTIFY. The Flag is reset when this bit of the register is written to '1'. The signal is visible to MCS0 sub module as part of the special function register.
9	INC_CNT1_FLAG	Flag according to DPLL_INC_CNT1_MASK.INC_CNT1_NOTIFY The INC_CNT1_FLAG is set to '1' indicating that the signal DPLL_INC_CNT1.INC_CNT1 has left the state defined by the trigger mask of DPLL_INC_CNT1_MASK.INC_CNT1_NOTIFY. The Flag is reset when this bit of the register is written to '1'. The signal is visible to MCS0 sub module as part of the special function register.
8	STA_FLAG_S	Flag according to DPLL_STA_MASK.STA_NOTIFY_S The STA_FLAG_S is set to '1' indicating that the signal DPLL_STA.STA_S has left the state defined by the trigger mask of DPLL_STA_MASK.STA_NOTIFY_S. The Flag is reset when this bit of the register is written to '1'. The signal is visible to MCS0 sub module as part of the special function register.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STA_FLAG_T	Flag according to DPLL_MASK.STA_NOTIFY_T The STA_FLAG_T is set to '1' indicating that the signal DPLL_STA.STA_T has left the state defined by the trigger mask of DPLL_STA_MASK.STA_NOTIFY_T. The Flag is reset when this bit of the register is written to '1'. The signal is visible to MCS0 sub module as part of the special function register.

38.22.18.55 DPLL_INC_CNT1_MASK

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F5C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	INC_CNT1_NOTIFY							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INC_CNT1_NOTIFY															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.352 DPLL_INC_CNT1_MASK Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	INC_CNT1_NOTIFY	Notify value for INC_CNT1 of register DPLL_INC_CNT1. The INC_CNT1_NOTIFY is representing a trigger mask of DPLL_INC_CNT1.INC_CNT1. When DPLL_INC_CNT1.INC_CNT1 reaches the value of INC_CNT1_NOTIFY the flag DPLL_STA_FLAG.INC_CNT1_FLAG is set to '1' when DPLL_INC_CNT1.INC_CNT1 is leaving the state INC_CNT1_NOTIFY.

38.22.18.56 DPLL_INC_CNT2_MASK

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F60_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	INC_CNT2_NOTIFY							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INC_CNT2_NOTIFY															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.353 DPLL_INC_CNT2_MASK Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	INC_CNT2_NOTIFY	Notify value for INC_CNT2 of register DPLL_INC_CNT2. The INC_CNT2_NOTIFY is representing a trigger mask of DPLL_INC_CNT2.INC_CNT2. When DPLL_INC_CNT2.INC_CNT2 reaches the value of INC_CNT2_NOTIFY the flag DPLL_STA_FLAG.INC_CNT2_FLAG is set to '1' when DPLL_INC_CNT2.INC_CNT2 is leaving the state INC_CNT2_NOTIFY.

38.22.18.57 DPLL_NUSC_EXT1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F64_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	WSYN	—	—	—	—	—	—	—	SYN_S_OLD						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SYN_S						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.354 DPLL_NUSC_EXT1 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	WSYN	Write control bit for SYN_S and SYN_S_OLD; read as zero. 0: The SYN_S value is not writable 1: The SYN_S value is writable
29 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	SYN_S_OLD	Number of real and virtual events to be considered for the last increment. This value reflects the NS value of the last but one valid increment, stored in ADT_S[i]; is updated automatically when writing SYN_S. **NOTE** This value is updated by the SYN_S value when the WSYN bit in this register is set.
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	SYN_S	Number of real and virtual events to be considered for the current increment. This value reflects the NS value of the last valid increment, stored in ADT_S[i]; to be updated after all calculations in step 37 of Table (DPLL_6908)*[XREF TARGET " (DPLL_6908)] NOT EXIST* **NOTE** This value can only be written when the WSYN bit in this register is set.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.58 DPLL_NUSC_EXT2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F68_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WVSN	—	WNUS	—	—	—	—	—	—	VSN						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSS	—	—	—	—	—	—	—	—	NUSE						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.355 DPLL_NUSC_EXT2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	WVSN	Write control bit for VSN; read as zero. 0: The VSN value is not writable 1: The VSN value is writable
30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	WNUS	Write control bit for NUSE; read as zero. 0 = the NUSE value is not writeable 1 = the NUSE value is writeable
28 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	VSN	Virtual STATE number; number of virtual state increments in the current NUSE region. This value reflects the number of virtual increments in the current NUSE region; for NUSE=1 this value is zero, when the CPU sets NUSE to a value > 1 or zero (2 modulo 2), it must also set VSN to the correspondent value; the VSN value is subtracted from the NUSE value in order to get the corresponding APS value for the past; the VSN value is not used for the APS_1C2 pointer. VSN is to be updated by the CPU when a new gap is to be considered for NUSE or a gap is leaving the NUSE region; for this purpose the SASI interrupt can be used; no further update of VSN is necessary when NUSE is set to FULL_SCALE **NOTE** This value can only be written when the WVSN bit is set.
15	FSS	FULL_SCALE of STATE; this value is to be set, when NUSE is set to FULL_SCALE 0: The NUSE value is less than FULL_SCALE 1: The NUSE value is equal to FULL_SCALE This value is set by the CPU, but reset automatically to 0 by a change of direction or loss of LOCK. **NOTE** This value can only be written when the WNUS bit is set.

Table 38.355 DPLL_NUSC_EXT2 Register Contents (2/2)

Bit Position	Bit Name	Function
14 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	NUSE	Number of recent STATE events used for SUB_INCx calculations modulo 2*(SNUmax+1). No gap is considered in that case for this value, but in the VSN value (see below): This register is set by the CPU but reset automatically to 1 by a change of direction or loss of LOCK. Each other value can be set by the CPU, maybe Full_SCALE, HALF_SCALE or parts of them. The relation values QDT_Sx are calculated using NUSE values in the past with its maximum value of 2*SNU+1. **NOTE** This value can only be written when the WNUS bit is set.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.59 DPLL_APS_EXT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F38_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	APS_1C2				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APS_1C2		WAPS_1C2	—	—	—	—	APS							WAPS	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 38.356 DPLL_APS_EXT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 14	APS_1C2	Address pointer STATE for RAM region 1c2; Actual RAM pointer address value for TSF_S[i]. Initial value: zero (00_H). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 128 STATE events but limited to 2*(SNU+1) in normal and emergency mode; this pointer is used for the RAM region 1c2. For SYS=1: APS_1C2 is incremented (decremented) by SYN_S_OLD for each active STATE event and DIR2=0 (DIR2=1). For SYS=0: APT_1c2 is incremented or decremented by 1 respectively. The APS_1C2 offset value is added in the above shown bit position with the subsection offset of the RAM region. In addition when the APS_1C3 value is written by the CPU – in order to synchronize the DPLL- with the next active STATE event the APS_1C2_EXT value is added/subtracted (while APS_1C2_STATUS is one; see DPLL_APT_SYNC register at **Section 38.22.18.25, DPLL_APS_SYNC**). **NOTE** This value can only be written when the WAPS_1C2 bit is set
13	WAPS_1C2	Write bit for address pointer APS_1C2, read as zero. 0: The APS_1C2 is not writable 1: The APS_1C2 is writable
12 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 38.356 DPLL_APS_EXT Register Contents (2/2)

Bit Position	Bit Name	Function
8 to 2	APS	Address pointer STATE; Actual RAM pointer address value for DT_S[i] and RDT_S[i] Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 128 STATE events but limited to 2*(SNU+1–SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)–SYN_NS for SYSF=1 respectively; See **Section 38.22.10, MCS to DPLL interface**. APS is incremented (decremented) by one for each active STATE event and DIR2=0 DIR2=1). The APS offset value is added in the above shown bit position with the subsection offset of the RAM region. **NOTES** 1. The APS pointer value is directed to the RAM position, in which the data values are to be written, which correspond to the last increment. The APS value is not to be changed, when the direction (shown by DIR2) changes, because it points always to a storage place after the considered increment. Changing of DIR2 takes place always after an active STATE event and the resulting increment/decrement. 2. This value can only be written when the WAPS bit is set.
1	WAPS	Write bit for address pointer APS, read as zero. 0: The APS is not writable 1: The APS is writable
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.60 DPLL_APS_1C3_EXT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F3C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	APS_1C3							—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 38.357 DPLL_APS_1C3_EXT Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 2	APS_1C3	Address pointer STATE for RAM region 1c3; Actual RAM pointer address value for ADT_S[i] Initial value: zero (0x00). Actual RAM pointer and synchronization position/value of STATE events in FULL_SCALE for up to 128 STATE events but limited to 2*(SNU+1–SYN_NS) in normal and emergency mode for SYSF=0 or to 2*(SNU+1)–SYN_NS for SYSF=1 respectively; this pointer is used for the RAM region 1c3. See **Section 38.22.10, MCS to DPLL interface**. The RAM pointer is set by the CPU accordingly, when the synchronization condition was detected.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. The APS_1C3 pointer value is directed to the RAM position of the profile element in RAM region 1c2, which corresponds to the current increment. When changing the direction DIR1 or DIR2 respectively, this is always known before an active STATE event is processed. This is because of the pattern recognition in SPE (for PMSM) or because of the direction change recognition by TRIGGER. This direction change results in an automatic increment (forwards) or decrement (backwards) when the input event occurs in addition with a 2 times correction.

Note 2. The APS_1C3_x offset value is added in the above shown bit position with the subsection address offset of the corresponding RAM region.

Note 3. This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.61 DPLL_APS_SYNC_EXT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F30_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	APS_1C2_OLD						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	APS_1C2_STATUS	—	—	—	—	—	—	—	—	APS_1C2_EXT						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.358 DPLL_APS_SYNC_EXT Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	APS_1C2_OLD	Address pointer STATE for RAM region 1c2 at synchronization time; this value is set by the current APS_1C2 value when the synchronization takes place for the first active STATE event after writing APS_1C3 but before adding the offset value APS_1C2_EXT (that means: when APS_1C2_STATUS=1). Address pointer APS_1C2 value at the moment of synchronization, before the offset value is added, that means the pointer with this value points to the last value before the additional inserted gap
15	APS_1C2_STATUS	Address pointer 1c2 status; set by CPU before the synchronization is performed. The value is cleared automatically when the APS_1C2_OLD value is written. 0: APS_1C2_EXT is not to be considered. 1: APS_1C2_EXT has to be considered for time stamp field extension.
14 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	APS_1C2_EXT	Address pointer 1c2 extension; this offset value determines, by which value the APS_1C2 is changed at the synchronization time; set by CPU before the synchronization is performed. This offset value is the number of virtual increments to be inserted in the TSF for an imminent intended synchronization; the CPU sets its value dependent on the gaps until the synchronization time taking into account the considered NUSE value to be set and including the next future increment (when SYN_S_OLD is still 1). When the synchronization takes place, this value is to be added to the APS_1C2 address pointer (for forward direction, DIR2=0) and the APT_1c2_status bit is cleared after it. For backward direction subtract APS_1C2_EXT accordingly. **NOTE** When the synchronization is intended and the NUSE value is to be set to FULL_SCALE after it, the APS_1C2_EXT value must be set to SYN_NS (for SYSF=1) or 2*SYN_NS (for SYSF=0) in order to be able to fill all gaps in the extended TSF_S with the corresponding values by the CPU. When still not all values for FULL_SCALE are available, the APS_1C2_EXT value considers only a share according to the NUSE value to be set after the synchronization.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.18.62 DPLL_CTRL_EXT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $28F34_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	SYN_NS					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SNU					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.359 DPLL_CTRL_EXT Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	SYN_NS	Synchronization number of STATE; summarized number of virtual increments in HALF_SCALE Sum of all systematic missing STATE events in HALF_SCALE (for SYSF=0) or FULL SCALE (for SYSF=1); the SYN_NS missing STATES can be divided up to an arbitrary number of blocks. The pattern of events and missing events in FULL_SCALE is shown in RAM region 1c3 as value NS in addition to the adapted values. The number of stored increments in FULL_SCALE must be equal to 2*(SNU+1–SYN_NS) for SYSF=0 or 2*(SNU+1)–SYN_NS for SYSF=1. This pattern is written by the CPU beginning from a fixed reference point (maybe beginning of the FULL_SCALE region). The relation to the actual increment is established by setting of the profile RAM pointer APS_1C3 in an appropriate relation to the RAM pointer APS of the actual increment by the CPU. **NOTES** 1. This value can only be written when the DPLL is disabled. 2. This value can only be written when (RMO=0 and SMC=0) or DEN=0. Set SSL=00 before changing this value and set RMO=1 only after FULL_SCALE with SSL>0.
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	SNU	STATE number; SNU+1 is number of nominal STATE events in HALF_SCALE (1 to 32). **NOTES** 1. This bit can only be written when the DPLL is disabled. 2. The number of nominal STATE events is the decimal value plus 1. This value can only be written when (RMO=0 and SMC=0) or DEN=0. Set SSL=00 before changing this value and set RMO=1 only after FULL_SCALE with SSL>0.

Note: This register is only used when DPLL_CTRL_11.STATE_EXT is set. If DPLL_CTRL_11.STATE_EXT is not set any read/write access to this register will return AEI_STATUS = 10_B.

38.22.19 DPLL RAM Region 1a Value Configuration Registers Overview

DPLL RAM Region 1a value contains following configuration registers:

Table 38.360 Register list

Symbol	Register Name	Details in Section
DPLL_RR1A	DPLL_RR1A	**38.22.20.1**
DPLL_PSA[i]	Position Request for Action i	**38.22.20.2**
DPLL_DLA[i]	Time to React for Action i	**38.22.20.3**
DPLL_NA[i]	Calculated Relative Time to Action i	**38.22.20.4**
DPLL_DTA[i]	Calculated Relative Time to Action i	**38.22.20.5**

38.22.20 DPLL RAM Region 1a value description

NOTE

Bits 31 to 24 of RAM region 1a are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Unused address regions are not protected against writing when implemented.

38.22.20.1 DPLL_RR1A

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28200_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.361 DPLL_RR1A Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA	DATA

38.22.20.2 DPLL_PSA[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.362 DPLL_PSA[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSA	Position information of a desired action (i=0...NOAC–1). **NOTE** This value can only be written when the DPLL is disabled.

Note: The PSA values for actions 24...31 are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.20.3 DPLL_DLA[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DLA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DLA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.363 DPLL_DLA[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DLA	Time to react before the corresponding position value of a desired action is reached (x=0...NOAC–1). In the case of LOW_RES=1 (see chapter {REF:DPLL_6906}) this delay value must be also given as low resolution value. **NOTE** This value can only be written when the DPLL is disabled.

Note: The DLA values for actions 24...31 are only available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.20.4 DPLL_NA[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED				DW			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DW						DB									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.364 DPLL_NA[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	NOT_USED	Not used **NOTE** Must be written to zero.
19 to 10	DW	Number of events to Action_i (integer part, i=0...NOAC–1). **NOTES** 1. Use the maximum value for NA_DW=0x3FF in the case of a calculated value which exceeds the represent able value. 2. This value can only be written when the DPLL is disabled.
9 to 0	DB	Number of events to Action_i (fractional part, i=0...NOAC–1). **NOTE** This value can only be written when the DPLL is disabled.

Note: The NA values for actions 24...31 are not available for devices. Refer to **Section 38.28, GTM Device 358**.

38.22.20.5 DPLL_DTA[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DTA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.365 DPLL_DTA[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DTA	Calculated relative time to ACTION_i (i=0...NOAC–1) **NOTE** This value can only be written when the DPLL is disabled. The DTA value is a positive integer value. When calculations using equations (DPLL_6935)*[XREF TARGET " (DPLL_6935)] NOT EXIST* or DPLL-14 result in a negative value, it is replaced by zero.

38.22.21 DPLL RAM Region 1b and 1c Value Configuration Registers Overview

DPLL RAM Region 1b and 1c value contains following configuration registers:

Table 38.366 Register list (1/2)

Symbol	Register Name	Details in Section
DPLL_TS_T	Actual TRIGGER Time Stamp Value	**38.22.22.1**
DPLL_TS_T_OLD	Previous TRIGGER Time Stamp Value	**38.22.22.2**
DPLL_FTV_T	Actual TRIGGER Filter value	**38.22.22.3**
DPLL_TS_S	Actual STATE Time Stamp Register	**38.22.22.4**
DPLL_TS_S_OLD	Previous STATE Time Stamp Register	**38.22.22.5**
DPLL_FTV_S	Actual STATE Filter Value	**38.22.22.6**
DPLL_THMI	TRIGGER Hold Time Min. Value	**38.22.22.7**
DPLL_THMA	TRIGGER Hold Time Max. Value	**38.22.22.8**
DPLL_THVAL	Measured TRIGGER Hold Time Value	**38.22.22.9**
DPLL_TOV	Time Out Value of Active TRIGGER Slope	**38.22.22.10**
DPLL_TOV_S	Time Out Value of active STATE Slope	**38.22.22.11**
DPLL_ADD_IN_CAL1	Calculated ADD_IN Value for SUB_INC1 Generation	**38.22.22.12**
DPLL_ADD_IN_CAL2	Calculated ADD_IN Value for SUB_INC2 Generation	**38.22.22.13**
DPLL_MPVAL1	Missing Pulses to be Added or Subtracted Directly	**38.22.22.14**
DPLL_MPVAL2	Missing Pulses to be Added or Subtracted Directly	**38.22.22.15**
DPLL_NMB_T_TAR	Target Number of Pulses to be sent in Normal Mode	**38.22.22.16**
DPLL_NMB_T_TAR_OLD	Last but one Target Number of Pulses to be sent in Normal Mode	**38.22.22.17**
DPLL_NMB_S_TAR	Target Number of Pulses to be sent in Emergency Mode	**38.22.22.18**
DPLL_NMB_S_TAR_OLD	Last but one Target Number of Pulses to be sent in Emergency Mode	**38.22.22.19**
DPLL_RCDT_TX	Reciprocal Value of the Expected Increment Duration of TRIGGER	**38.22.22.20**
DPLL_RCDT_SX	Reciprocal Value of the Expected Increment Duration of STATE	**38.22.22.21**
DPLL_RCDT_TX_NOM	Reciprocal Value of the Expected Nominal Increment Duration of TRIGGER	**38.22.22.22**
DPLL_RCDT_SX_NOM	Reciprocal Value of the Expected Nominal Increment Duration of STATE	**38.22.22.23**
DPLL_RDT_T_ACT	Reciprocal Value of the Last Increment of TRIGGER	**38.22.22.24**
DPLL_RDT_S_ACT	Reciprocal Value of the Last Increment of STATE	**38.22.22.25**
DPLL_DT_T_ACT	Duration of the Last TRIGGER Increment	**38.22.22.26**
DPLL_DT_S_ACT	Duration of the Last STATE Increment	**38.22.22.27**
DPLL_EDT_T	Difference of Prediction to Actual Value of the Last TRIGGER Increment	**38.22.22.28**
DPLL_MEDT_T	Weighted Difference of Prediction Errors of TRIGGER	**38.22.22.29**
DPLL_EDT_S	Difference of Prediction to Actual Value of the Last STATE Increment	**38.22.22.30**
DPLL_MEDT_S	Weighted Difference of Prediction Errors of STATE	**38.22.22.31**
DPLL_CDT_TX	Prediction of the Actual TRIGGER Increment Duration	**38.22.22.32**
DPLL_CDT_SX	Prediction of the Actual STATE Increment Duration	**38.22.22.33**

Table 38.366 Register list (2/2)

Symbol	Register Name	Details in Section
DPLL_CDT_TX_NOM	Prediction of the Nominal TRIGGER Increment Duration	**38.22.22.34**
DPLL_CDT_SX_NOM	Prediction of the Nominal STATE Increment Duration	**38.22.22.35**
DPLL_TLR	TRIGGER Locking Range	**38.22.22.36**
DPLL_SLR	STATE Locking Range	**38.22.22.37**
DPLL_PDT_[z]	Projected Increment Sum Relations for Action [z]	**38.22.22.38**
DPLL_MLS1	Calculated Number of Sub-Pulses between two nominal STATE Events for SMC = 0	**38.22.22.39**
DPLL_MLS2	Description: Calculated Number of Sub-Pulses between two nominal STATE Events for SMC=1 and RMO=1	**38.22.22.40**
DPLL_CNT_NUM_1	Counter for number of SUB_INC1 pulses	**38.22.22.41**
DPLL_CNT_NUM_2	Counter for number of SUB_INC2 pulses	**38.22.22.42**
DPLL_PVT	Plausibility Value of Next TRIGGER Slope	**38.22.22.43**
DPLL_PSTC	Actual Calculated Position Stamp of TRIGGER	**38.22.22.44**
DPLL_PSSC	Actual Calculated Position Stamp of STATE	**38.22.22.45**
DPLL_PSTM	Measured Position Stamp at Last TRIGGER Input	**38.22.22.46**
DPLL_PSTM_OLD	Measured Position Stamp at Last but one TRIGGER Input	**38.22.22.47**
DPLL_PSSM	Measured Position Stamp at Last STATE Input	**38.22.22.48**
DPLL_PSSM_OLD	Measured Position Stamp at Last but one STATE Input	**38.22.22.49**
DPLL_NMB_T	Number of Pulses to be sent in Normal Mode	**38.22.22.50**
DPLL_NMB_S	Number of Pulses to be sent in Emergency Mode	**38.22.22.51**
DPLL_RDT_S[i]	Reciprocal Values of the Nominal STATE Increment Duration in FULL_SCALE	**38.22.22.52**
DPLL_TSF_S[i]	Time Stamp Values of the Nominal STATE Events in FULL_SCALE	**38.22.22.53**
DPLL_ADT_S[i]	Adapt and Profile Values of the STATE Increments in FULL_SCALE	**38.22.22.54**
DPLL_DT_S[i]	Nominal STATE Increment Duration in FULL_SCALE	**38.22.22.55**

38.22.22 DPLL RAM Region 1b and 1c value description

NOTE

Bits 31 to 24 of RAM region 1b and 1c are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Unused address regions are not protected against writing when implemented.

38.22.22.1 DPLL_TS_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28400_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRIGGER_TS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRIGGER_TS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.367 DPLL_TS_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TRIGGER_TS	Time stamp value of the last active TRIGGER input. Measured TRIGGER time stamp **NOTE** The LSB address is determined using the SWON_T value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.2 DPLL_TS_T_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28404_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRIGGER_TS_OLD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRIGGER_TS_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.368 DPLL_TS_T_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TRIGGER_TS_OLD	Time stamp value of the last but one active TRIGGER input. Previous measured TRIGGER time stamp **NOTE** The LSB address is determined using the SWON_T value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.3 DPLL_FTV_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28408_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRIGGER_FT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRIGGER_FT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.369 DPLL_FTV_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TRIGGER_FT	Filter value of the last active TRIGGER input. Transmitted filter value **NOTE** The LSB address is determined using the SWON_T value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.4 DPLL_TS_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28410_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STATE_TS							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STATE_TS															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.370 DPLL_TS_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	STATE_TS	Time stamp value of the last active STATE input. **NOTE** The LSB address is determined using the SWON_S value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.5 DPLL_TS_S_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28414_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STATE_TS_OLD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STATE_TS_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.371 DPLL_TS_S_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	STATE_TS_OLD	Time stamp value of the last active STATE input. **NOTE** The LSB address is determined using the SWON_S value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.6 DPLL_FTV_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28418_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	STATE_FT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STATE_FT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.372 DPLL_FTV_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	STATE_FT	Filter value of the last active STATE input. Transmitted filter value **NOTE** The LSB address is determined using the SWON_S value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.7 DPLL_THMI

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28420_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THMI															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.373 DPLL_THMI Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 0	THMI	Minimal time between active and inactive TRIGGER slope (uint16); the time value corresponds to the time stamp clock counts: this does mean the clock selected for the TBU_CH0_BASE (see **Section 38.14.4.2, TBU_CH0_CTRL** register) Set min. value; generate the TINI interrupt in the case of a violation for THMI>0. **NOTE** Typical retention time values after an active slope can be e.g. between 45 µs (forwards) and 90 µs (backwards). When THMI is zero, consider always a THMI violation (forwards).

38.22.22.8 DPLL_THMA

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28424_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THMA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.374 DPLL_THMA Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 0	THMA	Maximal time between active and inactive TRIGGER slope (uint16); the time value corresponds to the time stamp clock counts: this does mean the clock selected for the TBU_CH0_BASE (see **Section 38.14.4.2, TBU_CH0_CTRL** register) Max. value to be set; generate the TAX interrupt in the case of a violation for THMA>0.

38.22.22.9 DPLL_THVAL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28428_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	THVAL							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	THVAL															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.375 DPLL_THVAL Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	THVAL	Measured time from the last active slope to the next inactive TRIGGER slope in time stamp clock counts: this does mean the clock selected for the TBU_CH0_BASE (uint16); The measured value considers all input slope filter delays. From the received input the corresponding filter delays are subtracted before the time stamp difference of active and inactive slope is calculated.

Note: In the case of LOW_RES=1 and TBU_HRT=0 the difference between the time stamps of active and inactive slope is multiplied by 8. The register contains this value.

38.22.22.10 DPLL_TOV

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28430_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOV_DW						TOV_DB									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.376 DPLL_TOV Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 10	TOV_DW	Decision value (integer part) for missing TRIGGER interrupt. TOV(15:0) is to be multiplied with the duration of the last increment and divided by 1024 in order to get the time-out time value for a missing TRIGGER event. **NOTE** For the case of LOW_RES=1 (see **Section 38.22.18.30, DPLL_STATUS** register) consider for the calculation of the time out value the following cases: LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=1: multiply the TBU_TS0 value by 8 LOW_RES=1 and DPLL_CTRL_1/TS0_HRT=0: multiply the TBU_TS0 value by 8 multiply the estimated time point value (using TS_T, dt_t_ACT and TOV) by 8 LOW_RES=0 and DPLL_CTRL_1/TS0_HRT=0: use TBU_TS0 and the estimated time point value unchanged.
9 to 0	TOV_DB	Decision value (fractional part) for missing TRIGGER interrupt.

38.22.22.11 DPLL_TOV_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28434_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DW						DB									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.377 DPLL_TOV_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 10	DW	Decision value (integer part) for missing STATE interrupt. TOV_S (15:0) is to be multiplied with the duration of the last increment and divided by 1024 in order to get the time-out time value for a missing STATE event. **NOTE** For the case of LOW_RES=1 (see **Section 38.22.18.30, DPLL_STATUS**) consider for the calculation of the time out value the following cases: LOW_RES=1 and DPLL_CTRL_1/TS0_HRS=1: multiply the TBU_TS0 value by 8 LOW_RES=1 and DPLL_CTRL_1/TS0_HRS=0: multiply the TBU_TS0 value by 8 multiply the estimated time point value (using TS_T, dt_s_ACT and SOV) by 8 LOW_RES=0 and DPLL_CTRL_1/TS0_HRS=0: use TBU_TS0 and the estimated time point value unchanged.
9 to 0	DB	Decision value (fractional part) for missing STATE interrupt.

38.22.22.12 DPLL_ADD_IN_CAL1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28438_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADD_IN_CAL1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD_IN_CAL1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.378 DPLL_ADD_IN_CAL1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ADD_IN_CAL1	Calculated input value for SUB_INC1 generation, calculated by the DPLL. Calculated value The update of the ADD_IN value by the new calculated value ADD_IN_CAL1 is suppressed for one increment when an unexpected missing TRIGGER (SMC=1 or RMO=0) or an unexpected STATE (RMO=1 and SMC=0) is detected.

38.22.22.13 DPLL_ADD_IN_CAL2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2843C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADD_IN_CAL2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADD_IN_CAL2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.379 DPLL_ADD_IN_CAL2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	ADD_IN_CAL2	Input value for SUB_INC2 generation, calculated by the DPLL for SMC=RMO=1. Calculated value The update of the ADD_IN value by the calculated value ADD_IN_CAL2 is suppressed for one increment when an unexpected missing STATE (RMO=SMC=1) is detected.

38.22.22.14 DPLL_MPVAL1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28440_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SIX1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPVAL1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.380 DPLL_MPVAL1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	SIX1	Sign extension for MPVAL1 00_H: MPVAL1 is a positive number FF_H: MPVAL1 is a negative number **NOTE** All bits must be written to either all zeros or all ones.
15 to 0	MPVAL1	Missing pulses for direct correction of SUB_INC1 pulses by the CPU (sint16); Used only for RMO=0 or SMC=1 for the case PCM1=1. Add MPVAL1 once to INC_CNT1 and reset PCM1 after applying once

Note: Do not provide negative values which exceed the amount of NT*(MLT+1) or MLS1 respectively; when considered negative PD values the sum of both (MPVAL1 + NT*PD) should not exceed the amount of NT*(MLT+1) or MLS1 respectively.

38.22.22.15 DPLL_MPVAL2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28444_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SIX2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPVAL2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.381 DPLL_MPVAL2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	SIX2	Sign extension for MPVAL2 00_H: MPVAL2 is a positive number FF_H: MPVAL2 is a negative number **NOTE** All bits must be written to either all zeros or all ones.
15 to 0	MPVAL2	Missing pulses for direct correction of SUB_INC2 pulses by the CPU (sint16); Used only for SMC=RMO=1 for the case PCM2=1. Add MPVAL2 once to INC_CNT2 and reset PCM2 after applying once **NOTE** Do not provide negative values which exceed the amount of MLS2; when considered negative PD_S values the sum of both (MPVAL2 + NS *PD_S) should not exceed the amount of MLS2.

38.22.22.16 DPLL_NMB_T_TAR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28448_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_T_TAR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.382 DPLL_NMB_T_TAR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 0	NMB_T_TAR	Target number of pulses for TRIGGER; Calculated target number of pulses in normal mode for the current TRIGGER increment without missing pulses. **NOTE** The LSB address is determined using the SWON_T value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.17 DPLL_NMB_T_TAR_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2844C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_T_TAR_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.383 DPLL_NMB_T_TAR_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used **NOTE** Must be written to zero.
15 to 0	NMB_T_TAR_OLD	Target Number of pulses for TRIGGER; Calculated number of pulses in normal mode for the current TRIGGER increment without missing pulses. Calculated target pulse number **NOTE** The LSB address is determined using the SWON_T value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.18 DPLL_NMB_S_TAR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28450_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED				NMB_S_TAR			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_S_TAR															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.384 DPLL_NMB_S_TAR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	NOT_USED	Not used **NOTE** Must be written to zero.
19 to 0	NMB_S_TAR	Target Number of pulses for STATE; Calculated number of pulses in emergency mode for the current STATE increment without missing pulses. Calculated target pulse number **NOTE** The LSB address is determined using the SWON_S value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.19 DPLL_NMB_S_TAR_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28454_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED				NMB_S_TAR_OLD			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_S_TAR_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.385 DPLL_NMB_S_TAR_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	NOT_USED	Not used **NOTE** Must be written to zero.
19 to 0	NMB_S_TAR_OLD	Target Number of pulses for STATE; Calculated number of pulses in emergency mode for the current STATE increment without missing pulses. Calculated target pulse number **NOTE** The LSB address is determined using the SWON_S value in the OSW register (see **Section 38.22.18.8, DPLL_OSW**).

38.22.22.20 DPLL_RCDT_TX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 028460_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RCDT_TX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCDT_TX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.386 DPLL_RCDT_TX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RCDT_TX	Reciprocal value of expected increment duration *232 while only the lower 24 bits are used. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$.

38.22.22.21 DPLL_RCDT_SX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28464_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RCDT_SX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCDT_SX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.387 DPLL_RCDT_SX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RCDT_SX	Reciprocal value of expected increment duration *232 while only the lower 24 bits are used. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$.

38.22.22.22 DPLL_RCDT_TX_NOM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28468_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RCDT_TX_NOM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCDT_TX_NOM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.388 DPLL_RCDT_TX_NOM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RCDT_TX_NOM	Reciprocal value of nominal increment duration *232 while only the lower 24 bits are used. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$.

38.22.22.23 DPLL_RCDT_SX_NOM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2846C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RCDT_SX_NOM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCDT_SX_NOM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.389 DPLL_RCDT_SX_NOM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RCDT_SX_NOM	Reciprocal value of nominal increment duration *232 while only the lower 24 bits are used. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$.

Note: RCDT_TX_NOM and RCDT_SX_NOM are calculated by the values RCDT_TX and RCDT_SX to be multiplied with SYN_T or SYN_S respectively.

38.22.22.24 DPLL_RDT_T_ACT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28470_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RDT_T_ACT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDT_T_ACT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.390 DPLL_RDT_T_ACT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RDT_T_ACT	Reciprocal value of last TRIGGER increment *232, only the lower 24 bits are used; the LSB is rounded up when the next truncated bit is 1. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$ and the CRO bit in the DPLL_STATUS register is set (see **Section 38.22.18.30, DPLL_STATUS**).

38.22.22.25 DPLL_RDT_S_ACT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28474_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RDT_S_ACT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDT_S_ACT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.391 DPLL_RDT_S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RDT_S_ACT	Reciprocal value of last STATE increment *232, only the lower 24 bits are used; the LSB is rounded up when the next truncated bit is 1. Calculated value; when an overflow occurs in calculation the value is set to $FFFFFF_H$ and the CRO bit in the DPLL_STATUS register is set (see **Section 38.22.18.30, DPLL_STATUS**).

38.22.22.26 DPLL_DT_T_ACT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28478_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DT_T_ACT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT_T_ACT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.392 DPLL_DT_T_ACT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DT_T_ACT	Calculated duration of the last TRIGGER increment. Calculated duration of the last increment; Value will be written into the corresponding RAM field, when all calculations for the considered increment are done and APT is valid.

38.22.22.27 DPLL_DT_S_ACT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2847C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DT_S_ACT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT_S_ACT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.393 DPLL_DT_S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DT_S_ACT	Calculated duration of the last STATE increment. Calculated increment duration Value will be written into the corresponding RAM field, when all calculations for the considered increment are done and APS is valid.

38.22.22.28 DPLL_EDT_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28480_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EDT_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EDT_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.394 DPLL_EDT_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	EDT_T	Signed difference between actual value and a simple prediction of the last TRIGGER increment: sint24 Calculated error value

38.22.22.29 DPLL_MEDT_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28484_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MEDT_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEDT_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.395 DPLL_MEDT_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	MEDT_T	Signed middle weighted difference between actual value and prediction of the last TRIGGER increments: sint24; only calculated for SYT=1 Calculated medium error value, see **Section 38.22.6.2, (6) Equation DPLL-4 to calculate the weighted average error** The value is calculated only after synchronization (SYT=1) and the update is suppressed for one increment when an unexpected missing TRIGGER is detected.

38.22.22.30 DPLL_EDT_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28488_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EDT_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EDT_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.396 DPLL_EDT_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	EDT_S	Signed difference between actual value and prediction of the last STATE increment: sint24 Calculated error value, see **Section 38.22.6.3, (5) Equation DPLL-8 to calculate the error of last prediction**

38.22.22.31 DPLL_MEDT_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2848C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MEDT_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MEDT_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.397 DPLL_MEDT_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	MEDT_S	Signed middle weighted difference between actual value and prediction of the last STATE increments: sint24; only calculated for SYS=1 Calculated medium error value, see **Section 38.22.6.3, (6) Equation DPLL-9 to calculate the weighted average error**. The value is calculated only after synchronization (SYS=1) and the update is suppressed for one increment when an unexpected missing STATE is detected.

38.22.22.32 DPLL_CDT_TX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28490_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CDT_TX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDT_TX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.398 DPLL_CDT_TX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CDT_TX	Calculated duration of the current TRIGGER increment. Calculated value

38.22.22.33 DPLL_CDT_SX

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28494_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CDT_SX							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDT_SX															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.399 DPLL_CDT_SX Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CDT_SX	Calculated duration of the current STATE increment. Calculated value

38.22.22.34 DPLL_CDT_TX_NOM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 28498_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CDT_TX_NOM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDT_TX_NOM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.400 DPLL_CDT_TX_NOM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CDT_TX_NOM	Calculated duration of the current nominal TRIGGER event. Calculated value

38.22.22.35 DPLL_CDT_SX_NOM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2849C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CDT_SX_NOM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDT_SX_NOM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.401 DPLL_CDT_SX_NOM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CDT_SX_NOM	Calculated duration of the current nominal STATE event. Calculated value

38.22.22.36 DPLL_TLR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $284A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NOT_USED								TLR							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.402 DPLL_TLR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 8	NOT_USED	Not used. **NOTE** Must be written to zero.
7 to 0	TLR	Value is to be multiplied with the last nominal TRIGGER duration in order to get the range for the next TRIGGER event without setting TOR in the DPLL_STATUS register. Multiply value with the last nominal increment duration and check violation; when TLR = 0 don't perform the check.

38.22.22.37 DPLL_SLR

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $284A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NOT_USED								SLR							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.403 DPLL_SLR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 8	NOT_USED	Not used. **NOTE** Must be written to zero.
7 to 0	SLR	Value is to be multiplied with the last nominal STATE duration in order to get the range for the next STATE event without setting SOR in the DPLL_STATUS register. Multiply value with the last nominal increment duration and check violation; when SLR = 0 don't perform the check.

38.22.22.38 DPLL_PDT_[z]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times z + 28500_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DW							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DW		DB													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.404 DPLL_PDT_[z] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 14	DW	Integer part of relation between TRIGGER and STATE increments. Definition of relation values between TRIGGER and STATE increments PDT[i] according to Equations DPLL-11 or DPLL-13 (z:0...NOAC–1).
13 to 0	DB	Fractional part of relation between TRIGGER and STATE increments.

Note: The PDT[z] values for actions i=24...31 are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.22.39 DPLL_MLS1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285C0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED						MLS1	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLS1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.405 DPLL_MLS1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 18	NOT_USED	Not used. **NOTE** Must be written to zero.
17 to 0	MLS1	Number of pulses between two STATE events (to be set and updated by the CPU). For SMC = 0 the value of MLS1 is calculated once by the CPU for fixed values in the DPLL_CTRL_0 register by the formula MLS1 = ((MLT+1) x (TNU+1)/(SNU+1)) and set accordingly. For SMC = 1 the value of MLS1 represents the number of pulses between two nominal TRIGGER events (to be set and updated by the CPU).

38.22.22.40 DPLL_MLS2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285C4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED						MLS2	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MLS2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.406 DPLL_MLS2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 18	NOT_USED	Not used. **NOTE** Must be written to zero.
17 to 0	MLS2	Number of pulses between two STATE events (to be set and updated by the CPU). Using adapt information and the missing STATE event information SYN_S, this value can be corrected for each increment automatically.

38.22.22.41 DPLL_CNT_NUM_1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285C8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CNT_NUM_1							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT_NUM_1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.407 DPLL_CNT_NUM_1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CNT_NUM_1	Counter for number of SUB_INC1 pulses; Number of pulses in continuous mode for a nominal increment in normal and emergency mode for SUB_INC1, given and updated by CPU only. Count value for continuous mode.

38.22.22.42 DPLL_CNT_NUM_2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285CC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CNT_NUM_2							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT_NUM_2															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.408 DPLL_CNT_NUM_2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	CNT_NUM_2	Counter for number of SUB_INC2 pulses; Number of pulses in continuous mode for a nominal increment in normal and emergency mode for SUB_INC2, given and updated by CPU only. Count value for continuous mode.

38.22.22.43 DPLL_PVT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285D0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PVT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PVT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.409 DPLL_PVT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PVT	Plausibility value of next active TRIGGER slope. The meaning of the value depends on the value of the PIT value in the DPLL_CTRL_1 register. For PIT = 0: the number of SUB_INC1 pulses to be waited for until a next active TRIGGER event is accepted. For PIT = 1: PVT is to be multiplied with the last nominal increment time DT_T_ACT and divided by 1024 and reduced to a 24 bit value in order to get the time to be waited for until the next active TRIGGER event is accepted. The wait time must be exceeded for an active slope. **NOTE** When an active TRIGGER slope is detected while the wait condition is not fulfilled the interrupt PWI is generated. Note that the SGE1 must be set, when PIT = 0 in order to provide the necessary SUB_INC1 pulses for checking. After an unexpected missing TRIGGER the plausibility check is suppressed for the following increment. In case of direction change the PVT value is automatically set to zero in order to deactivate the check.

38.22.22.44 DPLL_PSTC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285E0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSTC							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSTC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.410 DPLL_PSTC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSTC	Calculated position stamp of last TRIGGER input; Value is set by the DPLL and can be updated by the CPU when filter values are to be considered for the exact position (see **Section 38.22.18.30, DPLL_STATUS** and DPLL_CTRL registers for explanation of the status and control bits used). For each active slope of TRIGGER in normal mode when FTD = 0: PSTC is set from actual position value, for the first active TRIGGER event (no filter delay considered) the CPU must update the value once, taking into account the filter value when FTD = 1: PSTC is incremented at each TRIGGER event by SMC = 0: ((MLT+1)+PD) x SYN_T; while PD = 0 for AMT = 0, SMC = 1: (MLS1+PD) x (SYN_T); while PD = 0 for AMT = 0.

38.22.22.45 DPLL_PSSC

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285E4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSSC							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSSC															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.411 DPLL_PSSC Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSSC	Calculated position stamp for the last STATE input; First value is set by the DPLL and can be updated by the CPU when the filter delay is to be considered. For each active slope of STATE in emergency mode when FSD = 0: PSSC is set from actual position value (no filter delay considered), the CPU must update the value once, taking into account the filter value when FSD = 1: at each active slope of STATE (PD_S_store = 0 for AMS = 0): SMC = 0: add (MLS1+PD_S_store) x (SYN_S); SMC = 1: add (MLS2+PD_S_store) x (SYN_S);

38.22.22.46 DPLL_PSTM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285E8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSTM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSTM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.412 DPLL_PSTM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSTM	Position stamp of TRIGGER, measured; Measured position stamp of last active TRIGGER input. Store the value TBU_TS1 when an active TRIGGER event occurs. The value of PSTM is invalid for (RMO = 1 and SMC = 0).

Note: The LSB address is determined using the SWON_T value in the OSW register.

38.22.22.47 DPLL_PSTM_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285EC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSTM_OLD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSTM_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.413 DPLL_PSTM_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSTM_OLD	Last but one position stamp of TRIGGER, measured; Measured position stamp of last but one active TRIGGER input. Last PSTM value: see explanation of PSTM.

Note: The LSB address is determined using the SWON_T value in the OSW register.

38.22.22.48 DPLL_PSSM

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSSM							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSSM															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.414 DPLL_PSSM Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSSM	Position stamp of STATE, measured; Measured position stamp of last active STATE input. Store the value TBU_TS1 or TBU_TS2 respectively at the moment when an active STATE event occurs. The value of PSSM is invalid for (RMO = 0 and SMC = 0).

Note: The LSB address is determined using the SWON_T value in the OSW register.

38.22.22.49 DPLL_PSSM_OLD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285F4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSSM_OLD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSSM_OLD															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.415 DPLL_PSSM_OLD Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	PSSM_OLD	Last but one position stamp of STATE, measured; Measured position stamp of last but one active STATE input. Last PSSM value: see explanation of PSSM.

Note: The LSB address is determined using the SWON_T value in the OSW register.

38.22.22.50 DPLL_NMB_T

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285F8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.416 DPLL_NMB_T Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	NOT_USED	Not used. **NOTE** Must be written to zero.
15 to 0	NMB_T	Number of pulses for TRIGGER; Calculated number of pulses in normal mode for the current TRIGGER increment. Calculated pulse number.

38.22.22.51 DPLL_NMB_S

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $285FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED				NMB_S			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMB_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.417 DPLL_NMB_S Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 20	NOT_USED	Not used. **NOTE** Must be written to zero.
19 to 0	NMB_S	Number of pulses for STATE; Calculated number of pulses in emergency mode for the current STATE increment. Calculated pulse number.

38.22.22.52 DPLL_RDT_S[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 28600_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RDT_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDT_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.418 DPLL_RDT_S[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RDT_S	Reciprocal difference time of STATE; nominal reciprocal value of the number of time stamp clocks measured in the corresponding increment x232 while only the lower 24 bits are used; no gap considered. The LSB is rounded up when the next truncated bit is 1. **NOTE** There are 2 x (SNU+1–SYN_NS) entries for SYSF = 0 or 2 x (SNU+1)–SYN_NS entries for SYSF = 1 respectively.

Note: If DPLL_CTRL_11.STATE_EXT is set, this memory range is not used by the DPLL, but emulated outside the DPLL. The DPLL will access the MCS to DPLL interface 19.15 and will expect data to be correctly stored there. This means in fact, that the handling of the RDT_S values has to be done outside, in the MCS integrated in the same cluster.

38.22.22.53 DPLL_TSF_S[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 28700_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSF_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSF_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.419 DPLL_TSF_S[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TSF_S	Time stamp field of STATE; Time stamp value of each active STATE event. **NOTE** There are 2 x (SNU+1) entries.

Note: If DPLL_CTRL_11.STATE_EXT is set, this memory range is not used by the DPLL, but emulated outside the DPLL. The DPLL will access the MCS to DPLL interface 19.15 and will expect data to be correctly stored there. This means in fact, that the handling of the RDT_S values has to be done outside, in the MCS integrated in the same cluster.

38.22.22.54 DPLL_ADT_S[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 28800_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED		NS					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.420 DPLL_ADT_S[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23, 22	NOT_USED	Not used. **NOTE** Must be written to zero.
21 to 16	NS	Number of STATEs; number of nominal STATE parts in the corresponding increment. **NOTE** There are 2 x (SNU+1–SYN_NS) entries for SYSF = 0 or 2 x (SNU+1)–SYN_NS entries for SYSF = 1 respectively.
15 to 0	PD_S	Physical deviation of STATE; Adapt values for each nominal STATE increment in FULL_SCALE (sint16); This value represents the number of pulses to be added to the correspondent nominal increment. The absolute value of a negative PD_S must not exceed MLS1 or MLS2 respectively. The PD value does mean the number of SUB_INC1 pulses per nominal tooth to be added to NS x ((MLS1/2+1)+PD_S);

Note: If DPLL_CTRL_11.STATE_EXT is set, this memory range is not used by the DPLL, but emulated outside the DPLL. The DPLL will access the MCS to DPLL interface 19.15 and will expect data to be correctly stored there. This means in fact, that the handling of the RDT_S values has to be done outside, in the MCS integrated in the same cluster.

38.22.22.55 DPLL_DT_S[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 28900_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DT_S							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT_S															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.421 DPLL_DT_S[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DT_S	Difference time of STATE; nominal increment duration values for each STATE increment in FULL_SCALE (considering no gap). **NOTE** There are 2 x (SNU+1–SYN_NS) entries for SYSF = 0 or 2 x (SNU+1)–SYN_NS entries for SYSF = 1 respectively.

Note: If DPLL_CTRL_11.STATE_EXT is set, this memory range is not used by the DPLL, but emulated outside the DPLL. The DPLL will access the MCS to DPLL interface 19.15 and will expect data to be correctly stored there. This means in fact, that the handling of the RDT_S values has to be done outside, in the MCS integrated in the same cluster.

38.22.23 DPLL RAM Region 2 Value Configuration Registers Overview

DPLL RAM Region 2 value contains following configuration registers:

Table 38.422 Register list

Symbol	Register Name	Details in Section
DPLL_RR2	DPLL_RR2	**38.22.24.1**
DPLL_RDT_T[i]	Reciprocal Values of the Nominal TRIGGER Increments Duration in FULL_SCALE	**38.22.24.2**
DPLL_TSF_T[i]	Time Stamp Values of the Nominal TRIGGER Increments in FULL_SCALE	**38.22.24.3**
DPLL_ADT_T[i]	Adapt and Profile Values of the TRIGGER Increments in FULL_SCALE	**38.22.24.4**
DPLL_DT_T[i]	Nominal TRIGGER Increments Duration in FULL_SCALE	**38.22.24.5**

38.22.24 DPLL RAM Region 2 value description

NOTE

Bits 31 to 24 of RAM region 2 are not implemented and therefore always read as zero (reserved). Other bits which are declared as reserved are not protected against writing. Unused address regions are not protected against writing when implemented.

38.22.24.1 DPLL_RR2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $2C000_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.423 DPLL_RR2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DATA	DATA

Note: The PSA values for actions 24...31 are not available for all devices. Refer to **Section 38.28, GTM Device 358**.

38.22.24.2 DPLL_RDT_T[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 00000_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	RDT_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDT_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.424 DPLL_RDT_T[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	RDT_T	Reciprocal difference time of TRIGGER; 2 x (TNU+1– SYN_NT) stored values nominal reciprocal value of the number of time stamp clocks measured in the corresponding increment (which is divided by the number of nominal increments); multiplied by x232 while only the lower 24 bits are used; the LSB is rounded up, when the next truncated bit is 1. **NOTE** There are 2 x (TNU+1– SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

Note: The starting index for Memory DPLL_RDT_T[i] in RAM2 is defined by the parameter AOSV_2A in DPLL_AOSV2 Register

38.22.24.3 DPLL_TSF_T[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 00000_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TSF_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSF_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.425 DPLL_TSF_T[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	TSF_T	Time stamp field of active TRIGGER slopes. **NOTE** There are 2 x (TNU+1) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

Note: The starting index for Memory DPLL_TSF_T[i] in RAM2 is defined by the parameter AOSV_2C in DPLL_AOSV2 Register

38.22.24.4 DPLL_ADT_T[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $4_H \times i + 00000_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	NOT_USED					NT		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TINT			PD												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.426 DPLL_ADT_T[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 19	NOT_USED	Not used. **NOTE** Must be written to zero.
18 to 16	NT	Number of TRIGGERs; number of nominal TRIGGER parts in the corresponding increment. **NOTE** There are 2 x (TNU+1– SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.
15 to 13	TINT	TRIGGER Interrupt information; Depending on the value up to 7 different interrupts can be generated. In the current version the 5 interrupts TE0_IRQ to TE4_IRQ are supported by TINT = "001", "010", "011", "100", "101" respectively. For the values "000", "110" and "111" no interrupt is generated and no other reaction is performed. The corresponding interrupt is activated, when the TINT value is read by the DPLL together with the other values (PD, NT) according to the profile.
12 to 0	PD	Physical deviation; Adapt values for each nominal TRIGGER increment in FULL_SCALE (sint13); The PD value does mean the number of SUB_INC1 pulses to be added to NT x ((MLT+1)+PD); the absolute value of a negative PD must not exceed (MLT+1) or MLS1 respectively; systematic missing TRIGGER events must be considered for the value of PD;

Note: The starting index for Memory DPLL_ADT_T[i] in RAM2 is defined by the parameter AOSV_2C in DPLL_AOSV2 Register

38.22.24.5 DPLL_DT_T[i]

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> 4_H × i + 00000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DT_T							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT_T															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.427 DPLL_DT_T[i] Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	DT_T	Difference time of TRIGGER; increment duration values for each TRIGGER increment in FULL_SCALE divided by the number of nominal increments (nominal value). **NOTE** There are 2 x (TNU+1– SYN_NT) entries. The maximum number of entries is restricted to a value corresponding to the OSS value in the DPLL_OSW register.

Note: The starting index for Memory DPLL_DT_T[i] in RAM2 is defined by the parameter AOSV_2D in DPLL_AOSV2 Register

38.22.25 MCS to DPLL Configuration Registers Overview

MCS to DPLL contains following configuration registers:

Table 38.428 Register list

Symbol	Register Name	Details in Section
MCS2DPLL_DEB0	Data exchange buffer 0	**38.22.26.1**
MCS2DPLL_DEB1	Data exchange buffer 1	**38.22.26.2**
MCS2DPLL_DEB2	Data exchange buffer 2	**38.22.26.3**
MCS2DPLL_DEB3	Data exchange buffer 3	**38.22.26.4**
MCS2DPLL_DEB4	Data exchange buffer 4	**38.22.26.5**
MCS2DPLL_DEB5	Data exchange buffer 5	**38.22.26.6**
MCS2DPLL_DEB6	Data exchange buffer 6	**38.22.26.7**
MCS2DPLL_DEB7	Data exchange buffer 7	**38.22.26.8**
MCS2DPLL_DEB8	Data exchange buffer 8	**38.22.26.9**
MCS2DPLL_DEB9	Data exchange buffer 9	**38.22.26.10**
MCS2DPLL_DEB10	Data exchange buffer 10	**38.22.26.11**
MCS2DPLL_DEB11	Data exchange buffer 11	**38.22.26.12**
MCS2DPLL_DEB12	Data exchange buffer 12	**38.22.26.13**
MCS2DPLL_DEB13	Data exchange buffer 13	**38.22.26.14**
MCS2DPLL_DEB14	Data exchange buffer 14	**38.22.26.15**
MCS2DPLL_DEB15	Data exchange buffer 15	**38.22.26.16**

38.22.26 MCS to DPLL Registers Description

As already mentioned in **Section 38.22.10**, the following registers of the MCS2DPLL interface are exclusively accessible by the MCS instance of cluster 0 and cannot be accessed directly via CPU.

38.22.26.1 MCS2DPLL_DEB0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.429 MCS2DPLL_DEB0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	DATA: Data exchange buffer 0. Actual content depends on whether it is a Read or Write operation from MCS. READ access from MCS: Duration of the last increment DT_S_ACT (19.7.5). This value is updated by the DPLL during the update of ram (STA_S = 0000_1001_B) and is ready to be read when STA_S is modified to 0000_1010_B. WRITE access from MCS: The DPLL expects DT_S[p-1] (19.6.3.5) or DT_S[p+1](19.6.5.2) during the increment prediction (STA_S = 0001_0000_B).

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.2 MCS2DPLL_DEB1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.430 MCS2DPLL_DEB1 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	DATA: Data exchange buffer 1. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects RDT_S[p-1] (19.6.3.4) or RDT_S[p+1](19.6.5.1) during the increment prediction (STA_S = 0001_0000_B) and RDT_S[t-1] (19.7.3) or RDT_S[t+1](19.7.4) during the action calculation (STA_S = 0111_0000_B)

Note: In both cases, the data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

38.22.26.3 MCS2DPLL_DEB2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.431 MCS2DPLL_DEB2 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	DATA: Data exchange buffer 2. Actual content depends on whether it is a Read or Write operation from MCS. READ access from MCS: TS_Sx. Use to compute/update the time stamp values for STATE during the update of ram (STA_S = 0000_1001_B) WRITE access from MCS: The DPLL expects RDT_S[p-q-1] (19.6.3.5) or RDT_S[p+q+1] during the increment prediction (STA_S = 0001_0000_B).

Note 1. The data read from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

Note 2. The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.4 MCS2DPLL_DEB3

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.432 MCS2DPLL_DEB3 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 3. Actual content depends on whether it is a Read or Write operation from MCS. READ access from MCS: DT_Sx. Use to compute/update the time stamp values for STATE during the update of ram (STA_S = 0000_1001_B) WRITE access from MCS: The DPLL expects DT_S[p-q] (19.6.3.5) or DT_S[p+q] during the increment prediction (STA_S = 0001_0000_B).

Note: The data read from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

38.22.26.5 MCS2DPLL_DEB4

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.433 MCS2DPLL_DEB4 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 4. Actual content depends on whether it is a Read or Write operation from MCS. READ access from MCS: SYN_S_OLD. Use to compute/update the time stamp values for STATE during the update of ram (STA_S = 0000_1001_B) WRITE access from MCS: The DPLL expects RDT_S[p-q] or RDT_S[p+q](19.6.5.4) during the increment prediction (STA_S = 0001_0000_B) and RDT_S[t-q] or RDT_S[t+q] during the action calculation (STA_S = 0111_0000_B)

Note 1. The data read from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

Note 2. The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.6 MCS2DPLL_DEB5

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.434 MCS2DPLL_DEB5 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 5. Actual content depends on whether it is a Read or Write operation from MCS. READ. Use to provide the proper Time Stamp Field value during the action calculation (STA_S = 0111_0000_B) WRITE access from MCS: The DPLL expects DT_S[p-q+1] or DT_S[p+q-1]during the increment prediction (STA_S = 0001_0000_B) and DT_S[t-q+1] or DT_S[t+q-1] during the action calculation (STA_S = 0111_0000_B)

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

38.22.26.7 MCS2DPLL_DEB6

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.435 MCS2DPLL_DEB6 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 6. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects ADT_S[APS_1C2] during the update of RAM (STA_S = 0000_1001_B) and change of direction (STA_S = 0000_0100_B and STA_S = 0000_0110_B)

Note 1. In both cases, the current ADT_S[APS_1C2] value should be stored in the register before unlocking the state machine the second time, i.e.: between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

Note 2. The format of ADT_S should match the defined in **Section 38.22.22.54, DPLL_ADT_S[i]**.

38.22.26.8 MCS2DPLL_DEB7

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.436 MCS2DPLL_DEB7 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 7. Actual content depends on whether it is a Read or Write operation from MCS. READ access from MCS: Duration of the reciprocal of the last increment RDT_S_ACT. This value is written by the DPLL during the update of ram (STA_S = 0000_1001_B) and is ready to be read when STA_S is modified to 0000_1010_B. WRITE access from MCS: The DPLL expects the reciprocal of the last increment RDT_S[APS] before it is overwritten with RDT_S_ACT during the update of ram (STA_S = 0000_1001_B). For the action calculation, this value is needed as well as RDT_S[t]

Note: During an update of ram, perform the write before unlocking the state machine (STA_S = 0000_1001_B), i.e.: after the first write to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO) but before the second write. During the action calculation, the data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

38.22.26.9 MCS2DPLL_DEB8

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.437 MCS2DPLL_DEB8 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 8. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects TSF_S[p] during the action calculation (STA_S = 0111_0000_B)

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.10 MCS2DPLL_DEB9

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.438 MCS2DPLL_DEB9 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 9. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects TSF_S[p-n] or TSF_S[p+n] during the action calculation (STA_S = 0111_0000_B)

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.11 MCS2DPLL_DEB10

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.439 MCS2DPLL_DEB10 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 10. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects TSF_S[p+m-n] or TSF_S[p-m+n] during the action calculation (STA_S = 0111_0000_B)

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.12 MCS2DPLL_DEB11

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.440 MCS2DPLL_DEB11 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 11. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects TSF_S[p+m] or TSF_S[p-m] during the action calculation (STA_S = 0111_0000_B)

Note: The data write from MCS should be performed between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO)

38.22.26.13 MCS2DPLL_DEB12

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

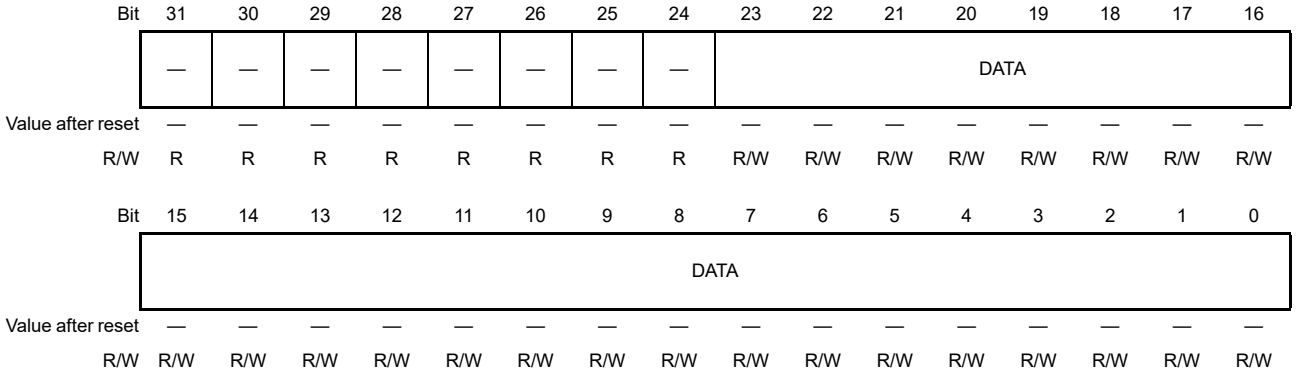


Table 38.441 MCS2DPLL_DEB12 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 12. READ access from MCS: Reads as 0. WRITE access from MCS: The DPLL expects the future adapt information ADT_S[APS_1C2+1] (when in forwards) or ADT_S[APS_1C2-1] (when in backwards) during the update of RAM (STA_S = 0000_1001_B) and change of direction (STA_S = 0000_0100_B and STA_S = 0000_0110_B)

Note: In both cases, the current ADT_S[APS_1C2+1] or ADT_S[APS_1C2-1] value should be stored in the register before unlocking the state machine the second time, i.e.: between the two writes to MCS2DPLL_DEB15 (MCS2DPLL_STATUS_INFO).

38.22.26.14 MCS2DPLL_DEB13

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.442 MCS2DPLL_DEB13 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned.
23 to 0	DATA	Data exchange buffer 13. READ access from MCS: Reads as 0. WRITE access from MCS: Ignored during DPLL processing

38.22.26.15 MCS2DPLL_DEB14

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.443 MCS2DPLL_DEB14 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned.
23 to 0	DATA	Data exchange buffer 14. READ access from MCS: Reads as 0. WRITE access from MCS: Ignored during DPLL processing.

38.22.26.16 MCS2DPLL_DEB15

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00000_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	DATA							
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.444 MCS2DPLL_DEB15 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the undefined value is returned. When writing, write "X".
23 to 0	DATA	Data exchange buffer 15. READ access from MCS: Reads as 0. WRITE access from MCS: Unlocks the DPLL STATE state machine.

38.23 Sensor Pattern Evaluation (SPE)

38.23.1 Overview

The Sensor Pattern Evaluation (SPE) sub-module can be used to evaluate three hall sensor inputs and together with the TOM module to support the drive of BLDC engines. Thus, the input signals are filtered already in the connected TIM channels. In addition, the SPE sub-module can be used as an input stage to the MAP sub-module if the DPLL should be used to calculate the rotation speed of one or two electric engine(s). The integration of the SPE sub-module into the overall GTM-IP architecture concept is shown in **Figure 38.118**.

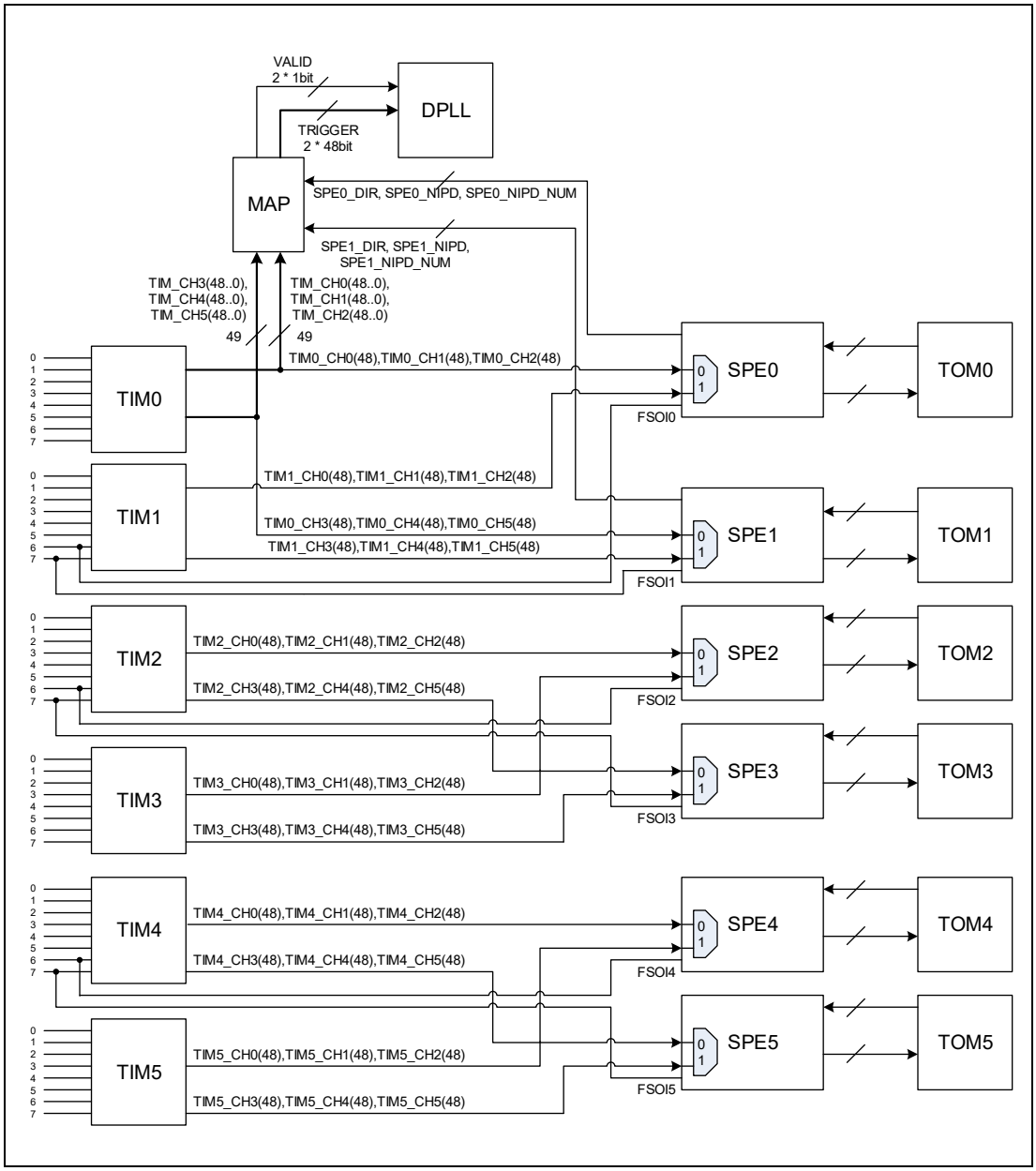


Figure 38.118 SPE sub-module integration concept into GTM-IP

The SPE sub-module can determine a rotation direction out of the combined TIM[i]_CHx(48), TIM[i]_CHy(48) and TIM[i]_CHz(48) signals. On this input signals a pattern match algorithm is applied to generate the SPEx_DIR signal on behalf of the temporal relation between these input

patterns. A possible sample pattern of the three input signals is shown in **Figure 38.119**. In general, the input pattern is programmable within the SPE sub-module.

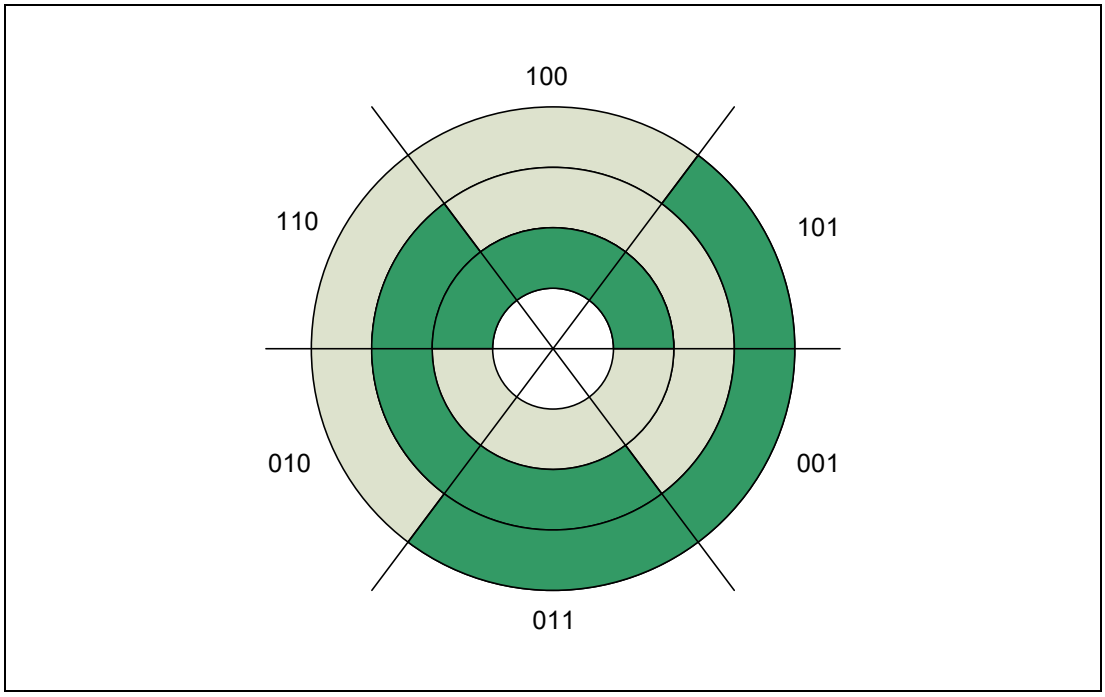


Figure 38.119 SPE Sample input pattern for TIM[i]_CH[x,y,z](48)

In **Figure 38.119** the input signals define the pattern from the input sensors which have a 50% high and 50% low phase. The pattern according to **Figure 38.119** is as follows:

100 110 010 011 001 101 100

where the first bit (smallest circle) represents TIM[i]_CH[x](48), the second bit represents TIM[i]_CH[y](48), and the third bit (greatest circle) represents TIM[i]_CH[z](48).

NOTE

The SPE module expects that with every new pattern only one of the three input signals changes its value.

38.23.2 SPE Sub-module description

The SPE sub-module can handle sensor pattern inputs. Every time if one of the input signals TIM[i]_CH[x](48), TIM[i]_CH[y](48) or TIM[i]_CH[z](48) changes its value, a sample of all three input signals is made. Derived from the sample of the three inputs the encoded rotation direction and the validity of the input pattern sequence is determined and signaled. When a valid input pattern is detected, the SPE sub-module can control the outputs of a dedicated connected TOM sub-module. This connection is shown in **Figure 38.120**.

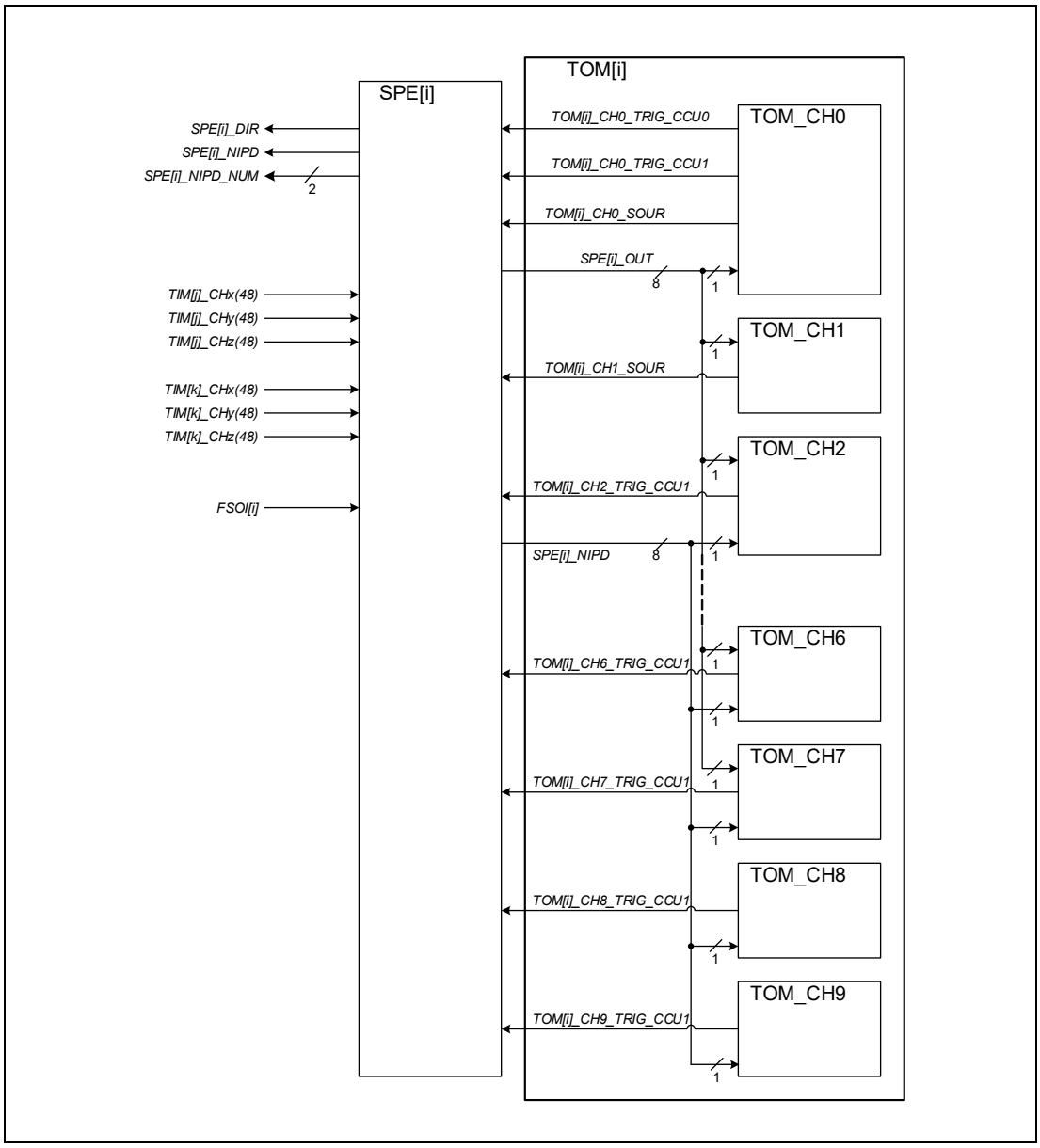


Figure 38.120 SPE to TOM Connections

The TOM[i]_CH0_TRIG_CCU[x] and TOM[i]_CH[x]_SOUR signal lines are used to evaluate the current state of the TOM outputs, whereas the SPE[i]_OUT output vector is used to control the TOM output depending on the new input pattern. The SPE[i]_OUT output vector is defined inside the SPE sub-module in a pattern definition table SPE[i]_OUT_PAT[x]. The internal SPE sub-module architecture is shown in **Figure 38.121**

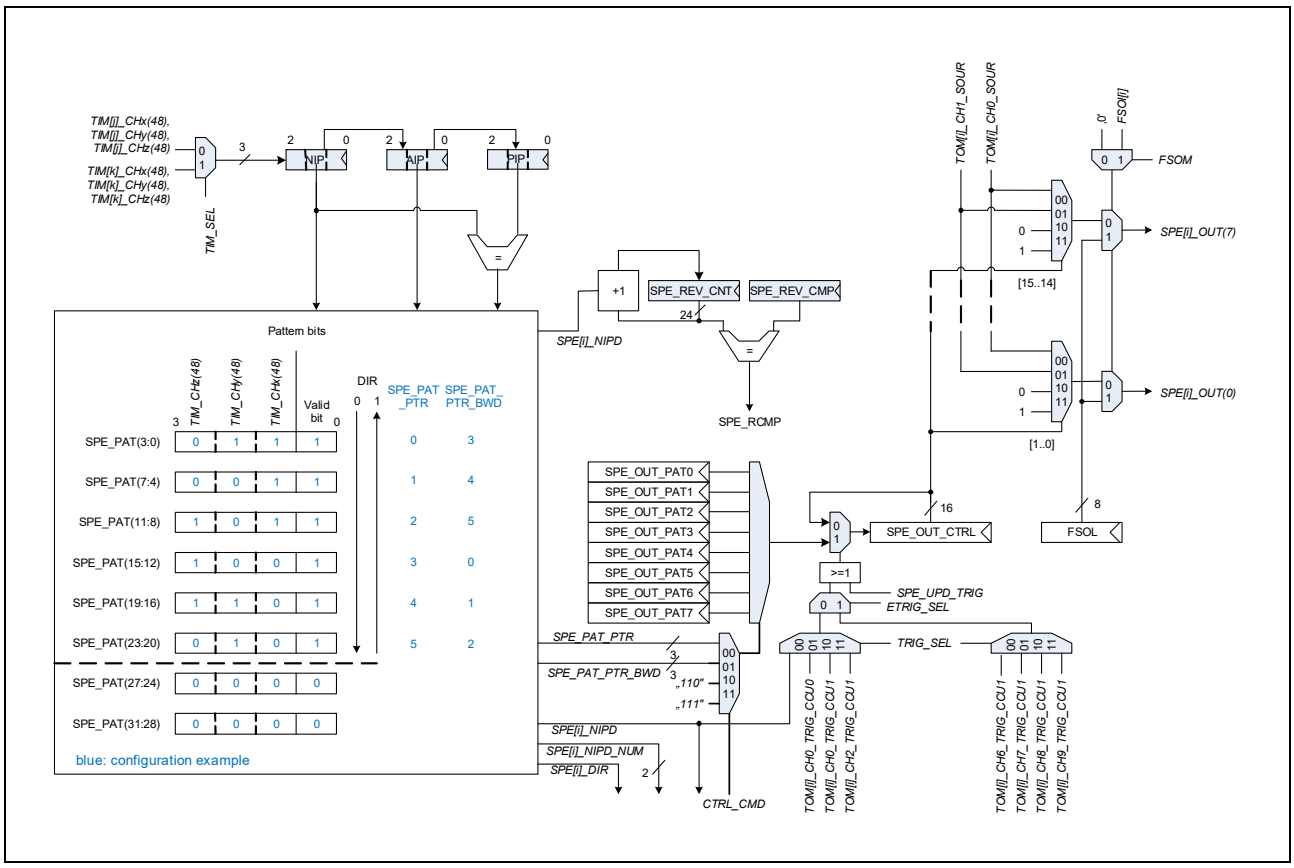


Figure 38.121 SPE Sub-module architecture

The SPE[i]_PAT register holds the valid input pattern for the three input patterns TIM[i]_CH[x](48), TIM[i]_CH[y](48) and TIM[i]_CH[z](48).

The input pattern is programmable. The valid bit shows if the programmed pattern is a valid one. **Figure 38.121** shows the programming of the SPE[i]_PAT register for the input pattern defined in **Figure 38.119**.

The rotation direction is determined by the order of the valid input pattern. This rotation direction defines if the SPE_PAT_PTR is incremented (DIR = 0) or decremented (DIR = 1). Whenever a valid input pattern is detected, the NIPD signal is raised, the SPE_PAT_PTR is incremented/decremented and a new output control signal SPE[i]_OUT(x) is send to the corresponding TOM sub-module.

To command directly the forward or backward rotation the SPE provides with SPE[i]_APT_PTR and SPE[i]_PAT_PTR_BWD two pointers to array SPE[i]_OUT_PAT[z]. Both can point to different values of SPE[i]_OUT_PAT[z] at the same point in time. SPE[i]_APT_PTR is intended to point to the pattern for forward commanding and SPE[i]_PAT_PTR_BWD is intended to point to the pattern for backward commanding. On startup both pointers have to be configured to an initial value that corresponds to different direction depending start pattern of SPE[i]_OUT_PAT[z]. With each valid new input pattern indicated by SPE_NIPD both pointers will be incremented or decremented according to the detected direction. Switching from command forward to command backward can then be done by changing the selected pointer to SPE[i]_OUT_PAT[z] array, i.e. changing SPE_CTRL_CMD in register SPE[i]_CMD from selecting SPE[i]_PAT_PTR to selecting SPE[i]_PAT_PTR_BWD or vice versa. The intended behavior is depicted in the following figure:

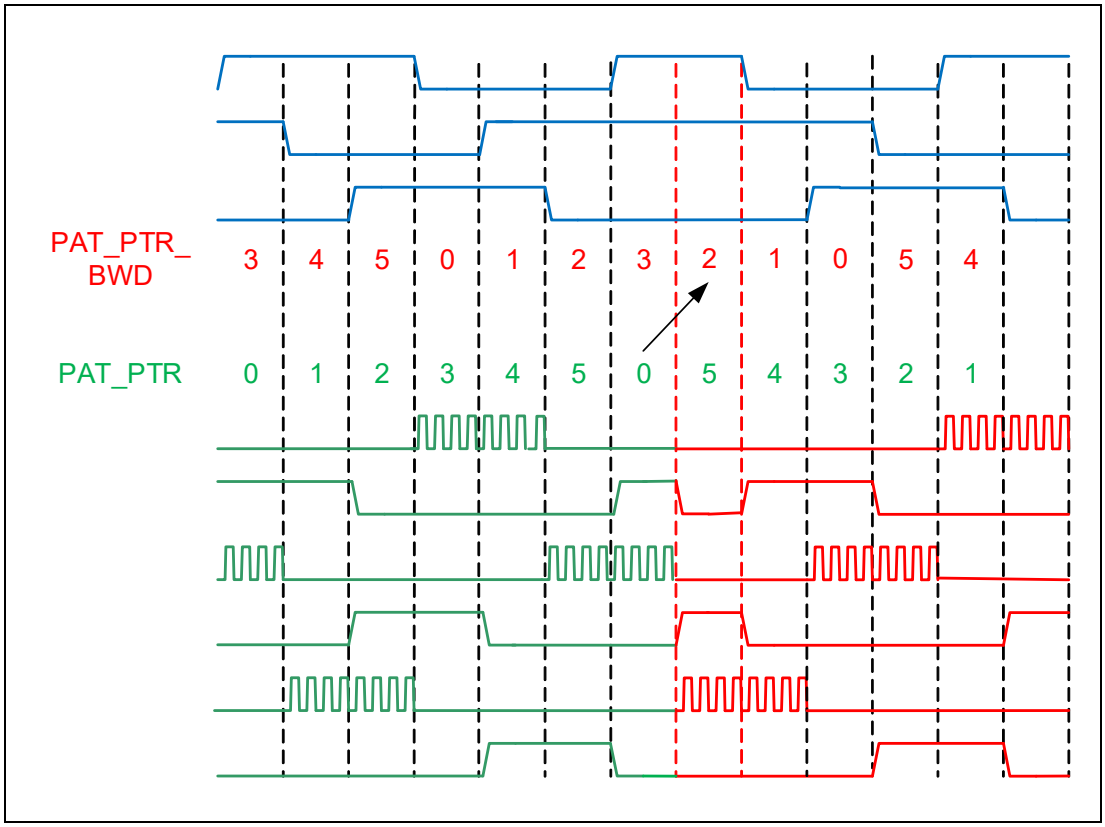


Figure 38.122 SPE forward-backward commanding

With command SPE_CTRL_CMD = 10_B or 11_B a dedicated configurable output pattern configured to the pattern SPE_OUT_PAT6 or SPE_OUT_PAT7 can be commanded to the outputs. An example is the introduction of a SW dead time if switching from pointer SPE[i]_PAT_PTR to SPE[i]_PAT_PTR_BWD or vice versa. E.g. if in SPE[i]_OUT_PAT6 the value 10_B for each output (i.e. set SPE_OUT(n) to 0) is programmed, this can be used as an intermediate step to introduce this 'all off' when switching between SPE[i]_PAT_PTR to SPE[i]_PAT_PTR_BWD or vice versa.

Selectable by TRIG_SEL and ETRIG_SEL the CCU1 trigger of either the TOM channel 2,6,7,8 and 9 can be used together with the SPE module to trigger a delayed update of the SPE_OUT_CTRL register after new input pattern detected by SPE (signaled by SPE[i]_NIPD). To do this, the TOM channel z=2, 6, 7, 8 or 9 has to be configured to work in one-shot mode (set bit OSM in register TOM[i]_CH[z]_CTRL). The SPE trigger of this channel has to be enabled, too (set description of bit SPEM and bit SPE_TRIG in register TOM[i]_CH[z]_CTRL). The SPE module has to be configured to update SPE_OUT_CTRL on TOM[i]_CH[z]_TRIG_CCU1 (set in SPE[i]_CTRL_STAT bits TRIG_SEL to 11_B). Then, on new input detected by SPE, the signal SPE[i]_NIPD triggers the start of the TOM channel z to generates one PWM period by resetting CN0 to 0.

On second PWM edge triggered by CCU1 of TOM channel z, the signal TOM[i]_CH[z]_TRIG_CCU1 triggers the update of SPE_OUT_CTRL.

The update of SPE[i]_OUT_CTRL with the content of one of the SPE_OUT_PAT[z] register can be triggered at any time by writing a 1 to bit SPE_UPD_TRIG in register SPE[i]_CMD.

The regular trigger for update of SPE[i]_OUT_CTRL (commutation trigger) is selected by TRIG_SEL and ETRIG_SEL.

According to **Figure 38.121**, the two input patterns 000_B and 111_B are not allowed combinations and will end in a SPE[i]_PERR interrupt.

These two patterns can be used to determine a sensor input error. A SPE[i]_PERR interrupt will also be raised, if the input patterns occur in a wrong order, e.g. if the pattern 010_B does not follow the pattern 110_B or 011_B.

The register SPE[i]_IN_PAT bit field inside the SPE[i]_CTRL_STAT register is implemented, where the input pattern history is stored by the SPE sub-module. The CPU can determine a broken sensor when the SPE[i]_PERR interrupt occurs by analyzing the bit pattern readable via bit field NIP inside the SPE[i]_CTRL_STAT register. The input pattern in the SPE[i]_CTRL_STAT register is updated whenever a valid edge is detected on one of the input lines TIM[i]_CH[x](48), TIM[i]_CH[y](48) or TIM[i]_CH[z](48). The pattern bit fields are then shifted. The input pattern history generation inside the SPE[i]_CTRL_STAT register is shown in **Figure 38.123**.

Additionally to the sensor pattern evaluation the SPE module also provides the feature of fast shutoff for all TOM channels controlled by the SPE module. The feature is enabled by setting bit FSOM in register SPE[i]_CTRL_STAT. The fast shutoff level itself is defined in the bit field FSOL of register SPE[i]_CTRL_STAT. The TIM input used to trigger the fast shutoff is either TIM channel 6 or TIM channel 7 depending on the TIM instance connected to the SPE module. For details of connections, refer to **Figure 38.118**.

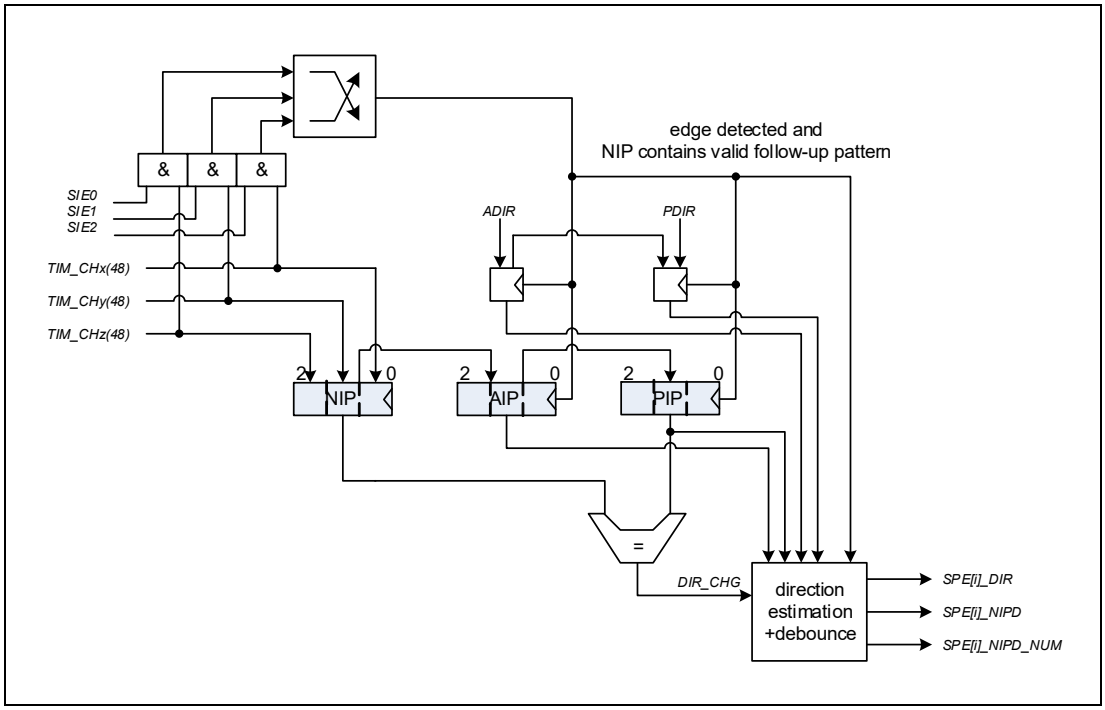


Figure 38.123 SPE[i]_IN_PAT register representation

The CPU can disable one of the three input signals, e.g. when a broken input sensor was detected, by disabling the input with the three input enable bits SIE inside the SPE[i]_CTRL_STAT register.

Whenever at least one of the input signal TIM[i]_CH[x](48), TIM[i]_CH[y](48) or TIM[i]_CH[z](48) changes the SPE sub-module stores the new bit pattern in an internal register NIP (New Input Pattern). If the current input pattern in NIP is the same as in the Previous Input Pattern (PIP) the direction of the engine changed, the SPE[i]_DCHG interrupt is raised, the direction change is stored internally and the pattern in the PIP bit field is filled with the AIP bit field and the AIP bit field is filled with the NIP bit field. The SPE[i]_DIR bit inside the SPE[i]_CTRL_STAT register is toggled and the SPE[i]_DIR signal is changed.

If the SPE encounters that with the next input pattern detected new input pattern NIP the direction change again, the input signal is categorized as bouncing and the bouncing input signal interrupt SPE[i]_BIS is raised.

Immediately after update of register NIP, when the new detected input pattern does not match the PIP pattern (i.e. no direction change was detected), the SPE shifts the value of register AIP to register PIP and the value of register NIP to register AIP. The SPE[i]_NIPD interrupt is raised.

The number of the channel that has been changed and thus leads to the new input pattern is encoded in the signal SPE[i]_NIPD_NUM.

If a sensor error was detected, the CPU has to define upon the pattern in the SPE[i]_CTRL_STAT register, which input line comes from the broken sensor. The faulty signal line has to be masked by the CPU and the SPE sub-module determines the rotation direction on behalf of the two remaining TIM[i]_CH[x] input lines.

The pattern history can be determined by the CPU by reading the two bit fields AIP and PIP of the SPE[i]_CTRL_STAT register. The AIP register field holds the actual detected input pattern at TIM[i]_CH[x](48), TIM[i]_CH[y](48) and TIM[i]_CH[z](48) and the PIP holds the previous detected pattern.

After reset the register NIP, AIP and PIP as well as the register SPE[i]_PAT_PTR and SPE[i]_OUT_CTRL will not contain valid startup values which would allow correct behavior after enabling SPE and detecting the first input patterns. Thus, it is necessary to initialize these register to correct values. To do this, before enabling the SPE, the bit field NIP of register SPE[i]_CTRL_STAT can be read and depending on this value the initialization values for the register AIP, PIP, SPT_PAT_PTR and SPE[i]_OUT_CTRL can be determined.

38.23.2.1 SPE Revolution detection

The SPE sub-module is able to detect and count the number of valid input patterns detected at the specified input ports. This is done with a 24 bit revolution counter SPE_REV_CNT. The counter is incremented by a value of one (1) when a new valid input pattern indicating forward direction is detected. The counter is decremented by a value of one (1) when a new valid input pattern indicating backward direction is detected.

In addition there exists a 24 bit SPE_REV_CMP register. The user can initialize this register with a compare value, where an interrupt SPE[i]_RCMP is raised, when the revolution counter equals the compare value either in forward or backward direction.

Both register may be written by software at any time.

38.23.2.2 SPE Interrupt signals

Signal	Description
SPE[i]_NIPD	SPE New valid input pattern detected.
SPE[i]_DCHG	SPE Rotation direction change detected on behalf of input pattern.
SPE[i]_PERR	SPE Invalid input pattern detected.
SPE[i]_BIS	SPE Bouncing input signal detected at input.
SPE[i]_RCMP	SPE Revolution counter compare value reached.

38.23.3 SPC Configuration Registers Overview

SPC contains following configuration registers:

Table 38.445 Register List

Register name	Description	Details in Section
SPE[i]_CTRL_STAT	SPEi Control status register	**38.23.4.1**
SPE[i]_PAT	SPEi Input pattern definition register.	**38.23.4.2**
SPE[i]_OUT_PAT[z] (z:0...7)	SPEi Output definition register.	**38.23.4.3**
SPE[i]_OUT_CTRL	SPEi output control register	**38.23.4.4**
SPE[i]_REV_CNT	SPEi input revolution counter	**38.23.4.5**
SPE[i]_REV_CMP	SPEi Revolution counter compare value	**38.23.4.6**
SPE[i]_IRQ_NOTIFY	SPEi Interrupt notification register.	**38.23.4.7**
SPE[i]_IRQ_EN	SPEi Interrupt enable register.	**38.23.4.8**
SPE[i]_EIRQ_EN	SPEi Error interrupt enable register.	**38.23.4.9**
SPE[i]_IRQ_FORCINT	SPEi Interrupt generation by software.	**38.23.4.10**
SPE[i]_IRQ_MODE	SPEi Interrupt mode configuration register	**38.23.4.11**
SPE[i]_CTRL_STAT2	SPEi Control status register 2	**38.23.4.12**
SPE[i]_CMD	SPEi Command register	**38.23.4.13**

38.23.4 SPC Configuration Registers Description

38.23.4.1 SPE[i]_CTRL_STAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00800_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FSOL								ETRIG_SEL	NIP			PDIR	PIP		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADIR	AIP			—	SPE_PAT_PTR			FSOM	TIM_SEL	TRIG_SEL		SIE2	SIE1	SIE0	EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.446 SPE[i]_CTRL_STAT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	FSOL	Fast Shutoff Level for TOM[i] channel 0 to 7.
23	ETRIG_SEL	Extended trigger selection ETRIG_SEL = 0 TRIG_SEL = 00_B: SPE[i]_NIPD selected TRIG_SEL = 01_B: TOM_CH0_TRIG_CCU0 selected TRIG_SEL = 10_B: TOM_CH0_TRIG_CCU1 selected TRIG_SEL = 11_B: TOM_CH2_TRIG_CCU1 selected ETRIG_SEL = 1 TRIG_SEL = 00_B: TOM_CH6_TRIG_CCU1 selected TRIG_SEL = 01_B: TOM_CH7_TRIG_CCU1 selected TRIG_SEL = 10_B: TOM_CH8_TRIG_CCU1 selected TRIG_SEL = 11_B: TOM_CH9_TRIG_CCU1 selected
22 to 20	NIP	New input pattern that was detected. **NOTE** This bit field mirrors the new input pattern. SPE internal functionality is triggered on each change of this bit field.
19	PDIR	Previous rotation direction. 0: Rotation direction is 0 according to SPE[i]_PAT register. 1: Rotation direction is 1 according to SPE[i]_PAT register.
18 to 16	PIP	Previous input pattern that was detected by a regular input pattern change.
15	ADIR	Actual rotation direction. 0: Rotation direction is 0 according to SPE[i]_PAT register. 1: Rotation direction is 1 according to SPE[i]_PAT register.
14 to 12	AIP	Actual input pattern that was detected by a regular input pattern change.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	SPE_PAT_PTR	Pattern selector for TOM output signals. Actual index into the SPE[i]_OUT_PAT[x] register table. Each register SPE[i]_OUT_PAT[x] is fixed assigned to one bit field IPx_PAT of register SPE[i]_PAT. Thus, the pointer SPE[i]_PAT_PTR represents an index to the selected SPE[i]_OUT_PAT[x] register as well as the actual detected input pattern IPx_PAT. 000_B = SPE[i]_OUT_PAT0 selected.

Table 38.446 SPE[i]_CTRL_STAT Register Contents (2/2)

Bit Position	Bit Name	Function
7	FSOM	Fast Shutoff Mode 0: Fast Shutoff mode disabled. 1: Fast Shutoff mode enabled.
6	TIM_SEL	Select TIM input signal If SPE0 0: TIM0_CH0 to TIM0_CH2 1: TIM1_CH0 to TIM1_CH2 If SPE1 0: TIM0_CH3 to TIM0_CH5 1: TIM1_CH3 to TIM1_CH5 If SPE2 0: TIM2_CH0 to TIM2_CH2 1: TIM3_CH0 to TIM3_CH2 If SPE3 0: TIM2_CH3 to TIM2_CH5 1: TIM3_CH3 to TIM3_CH5 If SPE4 0: TIM4_CH0 to TIM4_CH2 1: TIM5_CH0 to TIM5_CH2 If SPE5 0: TIM4_CH3 to TIM4_CH5 1: TIM5_CH3 to TIM5_CH5
5, 4	TRIG_SEL	Select trigger input signal. ETRIG_SEL = 0 TRIG_SEL = 00_B: SPE[i]_NIPD selected TRIG_SEL = 01_B: TOM_CH0_TRIG_CCU0 selected TRIG_SEL = 10_B: TOM_CH0_TRIG_CCU1 selected TRIG_SEL = 11_B: TOM_CH2_TRIG_CCU1 selected ETRIG_SEL = 1 TRIG_SEL = 00_B: TOM_CH6_TRIG_CCU1 selected TRIG_SEL = 01_B: TOM_CH7_TRIG_CCU1 selected TRIG_SEL = 10_B: TOM_CH8_TRIG_CCU1 selected TRIG_SEL = 11_B: TOM_CH9_TRIG_CCU1 selected **NOTE** In case of ETRIG_SEL=1, according to selected TOM_CH[x]_TRIG_CCU1 signal the configuration bits SPE_TRIG and OSM of register TOM_CH[x]_CTRL have to be set in same TOM channel x to enable the trigger signal generation in one-shot mode.
3 to 1	SIE[2:0]	SPE Input enable for TIM_CHx(48). 0: SPE Input is disabled. 1: SPE Input is enabled. **NOTE** When the input is disabled, a 0 signal is sampled for this input. However, the bit field NIP of this register shows the true value of the input signal.
0	EN	SPE sub-module enable. 0: SPE disabled. 1: SPE enabled.

38.23.4.2 SPE[i]_PAT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00804_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IP7_PAT			IP7_VAL	IP6_PAT			IP6_VAL	IP5_PAT			IP5_VAL	IP4_PAT			IP4_VAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IP3_PAT			IP3_VAL	IP2_PAT			IP2_VAL	IP1_PAT			IP1_VAL	IP0_PAT			IP0_VAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.447 SPE[i]_PAT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	IP7_PAT	Input pattern 7. See bits 3:1.
28	IP7_VAL	Input pattern 7 is a valid pattern. See bit 0.
27 to 25	IP6_PAT	Input pattern 6. See bits 3:1.
24	IP6_VAL	Input pattern 6 is a valid pattern. See bit 0
23 to 21	IP5_PAT	Input pattern 5. See bits 3:1
20	IP5_VAL	Input pattern 5 is a valid pattern. See bit 0
19 to 17	IP4_PAT	Input pattern 4. See bits 3:1
16	IP4_VAL	Input pattern 4 is a valid pattern. See bit 0
15 to 13	IP3_PAT	Input pattern 3. See bits 3:1
12	IP3_VAL	Input pattern 3 is a valid pattern. See bit 0
11 to 9	IP2_PAT	Input pattern 2. See bits 3:1
8	IP2_VAL	Input pattern 2 is a valid pattern. See bit 0
7 to 5	IP1_PAT	Input pattern 1. See bits 3:1
4	IP1_VAL	Input pattern 1 is a valid pattern. See bit 0
3 to 1	IP0_PAT	Input pattern 0. Bit field defines the first input pattern of the SPE input signals. Bit 1 defines the TIM[i]_CHx(48) input signal. Bit 2 defines the TIM[i]_CHy(48) input signal. Bit 3 defines the TIM[i]_CHz(48) input signal.

Table 38.447 SPE[i]_PAT Register Contents (2/2)

Bit Position	Bit Name	Function
0	IP0_VAL	Input pattern 0 is a valid pattern. 0: Pattern is invalid. 1: Pattern is valid.

NOTE

Only the first block of valid input patterns defines the commutation. All input pattern following the first marked invalid input pattern are ignored.

38.23.4.3 SPE[i]_OUT_PAT[z]

Access: This register can be read or written in 32-bit units.

Address: SPE[i]_OUT_PAT0: <GTM_base> 80_H × i + 00808_H
SPE[i]_OUT_PAT1: <GTM_base> 80_H × i + $0080C_H$
SPE[i]_OUT_PAT2: <GTM_base> 80_H × i + 00810_H
SPE[i]_OUT_PAT3: <GTM_base> 80_H × i + 00814_H
SPE[i]_OUT_PAT4: <GTM_base> 80_H × i + 00818_H
SPE[i]_OUT_PAT5: <GTM_base> 80_H × i + $0081C_H$
SPE[i]_OUT_PAT6: <GTM_base> 80_H × i + 00820_H
SPE[i]_OUT_PAT7: <GTM_base> 80_H × i + 00824_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPE_OUT_PAT15	SPE_OUT_PAT14	SPE_OUT_PAT13	SPE_OUT_PAT12	SPE_OUT_PAT11	SPE_OUT_PAT10	SPE_OUT_PAT9	SPE_OUT_PAT8	SPE_OUT_PAT7	SPE_OUT_PAT6	SPE_OUT_PAT5	SPE_OUT_PAT4	SPE_OUT_PAT3	SPE_OUT_PAT2	SPE_OUT_PAT1	SPE_OUT_PAT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.448 SPE[i]_OUT_PAT[z] Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15, 14	SPE_OUT_PAT[15:14]	SPE output control value for TOM_CH7 00_B: Set SPE_OUT7 to TOM_CH0_SOUR 01_B: Set SPE_OUT7 to TOM_CH1_SOUR 10_B: Set SPE_OUT7 to 0 11_B: Set SPE_OUT7 to 1
13, 12	SPE_OUT_PAT[13:12]	SPE output control value for TOM_CH6 00_B: Set SPE_OUT6 to TOM_CH0_SOUR 01_B: Set SPE_OUT6 to TOM_CH1_SOUR 10_B: Set SPE_OUT6 to 0 11_B: Set SPE_OUT6 to 1
11, 10	SPE_OUT_PAT[11:10]	SPE output control value for TOM_CH5 00_B: Set SPE_OUT5 to TOM_CH0_SOUR 01_B: Set SPE_OUT5 to TOM_CH1_SOUR 10_B: Set SPE_OUT5 to 0 11_B: Set SPE_OUT5 to 1
9, 8	SPE_OUT_PAT[9:8]	SPE output control value for TOM_CH4 00_B: Set SPE_OUT4 to TOM_CH0_SOUR 01_B: Set SPE_OUT4 to TOM_CH1_SOUR 10_B: Set SPE_OUT4 to 0 11_B: Set SPE_OUT4 to 1
7, 6	SPE_OUT_PAT[7:6]	SPE output control value for TOM_CH3 00_B: Set SPE_OUT3 to TOM_CH0_SOUR 01_B: Set SPE_OUT3 to TOM_CH1_SOUR 10_B: Set SPE_OUT3 to 0 11_B: Set SPE_OUT3 to 1
5, 4	SPE_OUT_PAT[5:4]	SPE output control value for TOM_CH2 00_B: Set SPE_OUT2 to TOM_CH0_SOUR 01_B: Set SPE_OUT2 to TOM_CH1_SOUR 10_B: Set SPE_OUT2 to 0 11_B: Set SPE_OUT2 to 1

Table 38.448 SPE[i]_OUT_PAT[z] Register Contents (2/2)

Bit Position	Bit Name	Function
3, 2	SPE_OUT_PAT[3:2]	SPE output control value for TOM_CH1 00_B: Set SPE_OUT1 to TOM_CH0_SOUR 01_B: Set SPE_OUT1 to TOM_CH1_SOUR 10_B: Set SPE_OUT1 to 0 11_B: Set SPE_OUT1 to 1
1,0	SPE_OUT_PAT[1:0]	SPE output control value for TOM_CH0 00_B: Set SPE_OUT0 to TOM_CH0_SOUR 01_B: Set SPE_OUT0 to TOM_CH1_SOUR 10_B: Set SPE_OUT0 to 0 11_B: Set SPE_OUT0 to 1

Note: Register SPE_OUT_PAT[x] defines the output selection for TOM[i]_CH0 to TOM[i]_CH7 depending on actual input pattern IP[z]_PAT with z:0...7.

38.23.4.4 SPE[i]_OUT_CTRL

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00828_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPE_OUT_CTRL15	SPE_OUT_CTRL14	SPE_OUT_CTRL13	SPE_OUT_CTRL12	SPE_OUT_CTRL11	SPE_OUT_CTRL10	SPE_OUT_CTRL9	SPE_OUT_CTRL8	SPE_OUT_CTRL7	SPE_OUT_CTRL6	SPE_OUT_CTRL5	SPE_OUT_CTRL4	SPE_OUT_CTRL3	SPE_OUT_CTRL2	SPE_OUT_CTRL1	SPE_OUT_CTRL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.449 SPE[i]_OUT_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15, 14	SPE_OUT_CTRL[15:14]	SPE output control value for TOM_CH7 00_B: Set SPE_OUT7 to TOM_CH0_SOUR 01_B: Set SPE_OUT7 to TOM_CH1_SOUR 10_B: Set SPE_OUT7 to 0 11_B: Set SPE_OUT7 to 1 **NOTE** Current output control selection for SPE7_OUT.
13, 12	SPE_OUT_CTRL[13:12]	SPE output control value for TOM_CH6 00_B: Set SPE_OUT6 to TOM_CH0_SOUR 01_B: Set SPE_OUT6 to TOM_CH1_SOUR 10_B: Set SPE_OUT6 to 0 11_B: Set SPE_OUT6 to 1 **NOTE** Current output control selection for SPE6_OUT.
11, 10	SPE_OUT_CTRL[11:10]	SPE output control value for TOM_CH5 00_B: Set SPE_OUT5 to TOM_CH0_SOUR 01_B: Set SPE_OUT5 to TOM_CH1_SOUR 10_B: Set SPE_OUT5 to 0 11_B: Set SPE_OUT5 to 1 **NOTE** Current output control selection for SPE5_OUT.

Table 38.449 SPE[i]_OUT_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
9, 8	SPE_OUT_CTRL[9:8]	SPE output control value for TOM_CH4 00_B: Set SPE_OUT4 to TOM_CH0_SOUR 01_B: Set SPE_OUT4 to TOM_CH1_SOUR 10_B: Set SPE_OUT4 to 0 11_B: Set SPE_OUT4 to 1 **NOTE** Current output control selection for SPE4_OUT.
7, 6	SPE_OUT_CTRL[7:6]	SPE output control value for TOM_CH3 00_B: Set SPE_OUT3 to TOM_CH0_SOUR 01_B: Set SPE_OUT3 to TOM_CH1_SOUR 10_B: Set SPE_OUT3 to 0 11_B: Set SPE_OUT3 to 1 **NOTE** Current output control selection for SPE3_OUT.
5, 4	SPE_OUT_CTRL[5:4]	SPE output control value for TOM_CH2 00_B: Set SPE_OUT2 to TOM_CH0_SOUR 01_B: Set SPE_OUT2 to TOM_CH1_SOUR 10_B: Set SPE_OUT2 to 0 11_B: Set SPE_OUT2 to 1 **NOTE** Current output control selection for SPE2_OUT.
3, 2	SPE_OUT_CTRL[3:2]	SPE output control value for TOM_CH1 00_B: Set SPE_OUT1 to TOM_CH0_SOUR 01_B: Set SPE_OUT1 to TOM_CH1_SOUR 10_B: Set SPE_OUT1 to 0 11_B: Set SPE_OUT1 to 1 **NOTE** Current output control selection for SPE1_OUT.
1,0	SPE_OUT_CTRL[1:0]	SPE output control value for TOM_CH0 00_B: Set SPE_OUT0 to TOM_CH0_SOUR 01_B: Set SPE_OUT0 to TOM_CH1_SOUR 10_B: Set SPE_OUT0 to 0 11_B: Set SPE_OUT0 to 1 **NOTE** Current output control selection for SPE0_OUT.

38.23.4.5 SPE[i]_REV_CNT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00840_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	REV_CNT							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REV_CNT															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.450 SPE[i]_REV_CNT Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	REV_CNT	Input signal revolution counter The counter is running if SPE module is enabled (bit SPE_EN). REV_CNT is incrementing if SPE_PAT_PTR is incrementing REV_CNT is decrementing if SPE_PAT_PTR is decrementing.

38.23.4.6 SPE[i]_REV_CMP

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00844_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	REV_CMP							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REV_CMP															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.451 SPE[i]_REV_CMP Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	REV_CMP	Input signal revolution counter compare value The interrupt SPE[i]_RCMP is raised when the SPE[i]_REV_CNT value equals the SPE[i]_REV_CMP register. **NOTES** 1. SPE[i]_RCMP is only raised if an incrementation or decrementation of SPE[i]_REV_CNT is applied, due to an input signal change. 2. Any update of SPE[i]_REV_CNT or SPE[i]_REV_CMP via AEI does not raise a SPE[i]_RCMP interrupt.

38.23.4.7 SPE[i]_IRQ_NOTIFY

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 0082C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPE_RCMP	SPE_BIS	SPE_PERR	SPE_DCHG	SPE_NIPD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.452 SPE[i]_IRQ_NOTIFY Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SPE_RCMP	SPE revolution counter match event. 0: No interrupt occurred. 1: New input pattern detected interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.
3	SPE_BIS	Bouncing input signal detected. 0: No interrupt occurred. 1: New input pattern detected interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.
2	SPE_PERR	Wrong or invalid pattern detected at input. 0: No interrupt occurred. 1: New input pattern detected interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.
1	SPE_DCHG	SPE_DIR bit changed on behalf of new input pattern. 0: No interrupt occurred. 1: New input pattern detected interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.

Table 38.452 SPE[i]_IRQ_NOTIFY Register Contents (2/2)

Bit Position	Bit Name	Function
0	SPE_NIPD	New input pattern interrupt occurred. 0: No interrupt occurred. 1: New input pattern detected interrupt occurred. **NOTE** This bit will be cleared on a CPU write access of value 1. A read access leaves the bit unchanged.

38.23.4.8 SPE[i]_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00830_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPE_RCMP_IRQ_EN	SPE_BIS_IRQ_EN	SPE_PERR_IRQ_EN	SPE_DCHG_IRQ_EN	SPE_NIPD_IRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.453 SPE[i]_IRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SPE_RCMP_IRQ_EN	SPE_RCMP_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.
3	SPE_BIS_IRQ_EN	SPE_BIS_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.
2	SPE_PERR_IRQ_EN	SPE_PERR_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.
1	SPE_DCHG_IRQ_EN	SPE_DCHG_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.
0	SPE_NIPD_IRQ_EN	SPE_NIPD_IRQ interrupt enable. 0: Disable interrupt, interrupt is not visible outside GTM-IP. 1: Enable interrupt, interrupt is visible outside GTM-IP.

38.23.4.9 SPE[i]_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00834_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TRG_SPE_RCMP	TRG_SPE_BIS	TRG_SPE_PERR	TRG_SPE_DCHG	TRG_SPE_NIPD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.454 SPE[i]_IRQ_FORCINT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TRG_SPE_RCMP	Force interrupt of SPE_RCMP. 0: Corresponding bit in status register will not be forced. 1: Assert corresponding SPE_RCMP bit in SPE_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after interrupt is released 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
3	TRG_SPE_BIS	Force interrupt of SPE_BIS. 0: Corresponding bit in status register will not be forced. 1: Assert corresponding TRG_SPE_BIS bit in SPE_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after interrupt is released 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
2	TRG_SPE_PERR	Force interrupt of SPE_PERR. 0: Corresponding bit in status register will not be forced. 1: Assert corresponding SPE_PERR bit in SPE_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after interrupt is released 2. This bit is write protected by bit RF_PROT of register GTM_CTRL
1	TRG_SPE_DCHG	Force interrupt of SPE_DCHG. 0: Corresponding bit in status register will not be forced. 1: Assert corresponding SPE_DCHG bit in SPE_IRQ_NOTIFY register. **NOTES** 1. This bit is cleared automatically after interrupt is released 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

Table 38.454 SPE[i]_IRQ_FORCINT Register Contents (2/2)

Bit Position	Bit Name	Function
0	TRG_SPE_NIPD	Force interrupt of SPE_NIPD. 0: Corresponding bit in status register will not be forced. 1: Assert corresponding SPE_NIPD bit in SPE_IRQ_NOTIFY. register. **NOTES** 1. This bit is cleared automatically after interrupt is released 2. This bit is write protected by bit RF_PROT of register GTM_CTRL

38.23.4.10 SPE[i]_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00834_H$

Value after reset: IRQ_MODE_RST_VAL

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.455 SPE[i]_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TRG_SPE_NIPD	IRQ mode selection. 00_B: Level mode. 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.23.4.11 SPE[i]_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 0083C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPE_RCMP_EIRQ_EN	SPE_BIS_EIRQ_EN	SPE_PERR_EIRQ_EN	SPE_DCHG_EIRQ_EN	SPE_NIPD_EIRQ_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 38.456 SPE[i]_EIRQ_EN Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SPE_RCMP_EIRQ_EN	SPE_RCMP_EIRQ error interrupt enable. 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.
3	SPE_BIS_EIRQ_EN	SPE_BIS_EIRQ error interrupt enable. 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.
2	SPE_PERR_EIRQ_EN	SPE_PERR_EIRQ error interrupt enable. 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.
1	SPE_DCHG_EIRQ_EN	SPE_DCHG_EIRQ error interrupt enable. 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.
0	SPE_NIPD_EIRQ_EN	SPE_NIPD_EIRQ interrupt enable. 0: Disable error interrupt, error interrupt is not visible outside GTM-IP. 1: Enable error interrupt, error interrupt is visible outside GTM-IP.

38.23.4.12 SPE[i]_CTRL_STAT2

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 00848_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SPE_PAT_PTR_BWD			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 38.457 SPE[i]_CTRL_STAT2 Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	SPE_PAT_PTR_BWD	Pattern selector for TOM output signals in case of SPE_CTRL_CMD = 01_B (e.g. backward direction). Index into the SPE[i]_OUT_PAT[z] register table in case of SPE_CTRL_CMD = 01_B which may be used for backward direction. Each register SPE[i]_OUT_PAT[x] is fixed assigned to one bit field IPx_PAT of register SPE[i]_PAT. Thus, the pointer SPE[i]_PAT_PTR_BWD represents an index to the selected SPE[i]_OUT_PAT[x] register as well as the actual detected input pattern IPx_PAT. The index pointer SPE_PAT_PTR_BWD is used if SPE_CTRL_CMD = 01_B. The index pointer SPE_PAT_PTR is used if SPE_CTRL_CMD = 00_B (by default). 000_B = SPE[i]_OUT_PAT0 selected.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

38.23.4.13 SPE[i]_CMD

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> $80_H \times i + 0084C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPE_UPD_TRIG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPE_CTRL_CMD	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.458 SPE[i]_CMD Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	SPE_UPD_TRIG	SPE updater trigger 0: No operation 1: Trigger update of SPE_OUT_CTRL with register selected by CTR_CMD multiplexer. **NOTE** This bit is automatically reset to 0.
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SPE_CTRL_CMD	SPE control command 00_B: Use SPE_PAT_PTR as an index pointer to select SPE[i]_OUT_PAT[z] 01_B: Use SPE_PAT_PTR_BWD as an index pointer to select SPE[i]_OUT_PAT[z] 10_B: Select SPE[i]_OUT_PAT6 11_B: Select SPE[i]_OUT_PAT7 **NOTE** On switch between 00_B and 01_B the direction flag will be set according to used pointer.

38.24 Interrupt Concentrator Module (ICM)

38.24.1 Overview

The Interrupt Concentrator Module (ICM) is used to bundle the GTM-IP interrupt lines of the individual sub-modules in a reasonable manner into interrupt groups. By this bundling a smaller amount of interrupt lines is visible at the outside of the GTM-IP.

The individual interrupts of the GTM-IP sub-modules and channels have to be enabled or disabled inside the sub-modules and channels.

The feed through architecture of bundled interrupt lines is used for the sub-modules AEI, ARU, BRC, CMP, SPE, PSM, TIM, DPLL, TOM, ATOM and MCS.

To determine the detailed interrupt source the microcontroller has to read the sub-module/channel interrupt notification register NOTIFY and serve the channel individual interrupt.

NOTE

The interrupts are only visible inside the ICM and in consequence outside of the GTM-IP, when the interrupt is enabled inside the sub-modules themselves.

38.24.2 Bundling

The GTM-IP sub-module individual interrupt sources are connected to the ICM. There, the individual interrupt lines are either feed through and signaled to the outside world or bundled a second time into groups and are then signaled to the outside world.

The ICM interrupt bundling is described in the following sections.

38.24.2.1 GTM Infrastructure Interrupt Bundling

The first interrupt group contains interrupts of the infrastructure and safety components of the GTM. This interrupt group includes therefore interrupt lines coming from the AEI, ARU, BRC, PSM, SPE and CMP sub-modules. In this interrupt group each individual channel of the sub-modules has its own interrupt line to the outside world.

Thus, the active interrupt line can be used by the CPU to determine the GTM-IP sub-module channel that raised the interrupt. The interrupts are also represented in the ICM_IRQG_0 register. This register is typically not read by the CPU, but it is readable.

In addition the interrupt line status for 8 channels of each FIFO are shown in the ICM_IRQG_PSM_0_CI register. Typically, the interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_PSM_0_CI register are typically not read out by the CPU, but they are readable.

In addition the interrupt line status for each SPE are shown in the ICM_IRQG_SPE_CI register. Typically, the interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_SPE_CI register are typically not read out by the CPU, but they are readable.

38.24.2.2 DPLL Interrupt Bundling

The DPLL Interrupt group handles the interrupts coming from the DPLL sub-module of the GTM-IP. Each of the individual DPLL interrupt lines has its own dedicated interrupt line to the outside world. The interrupts are additionally identified in the ICM_IRQG_1 interrupt group register. This register is typically not read out by the CPU, but it is readable.

38.24.2.3 TIM Interrupt Bundling

Inside this group sub-modules which handle GTM-IP input signals are treated. This is the case for the TIM[i] sub-modules. Each TIM sub-module channel is able to generate 6 individual interrupts if enabled inside the TIM channel. These six interrupts are bundled into one interrupt per TIM channel connected to the ICM.

The ICM does no further bundling. Thus, for the GTM-IP 32 interrupt lines TIM[i]_IRQ[y] are provided for the external microcontroller. The channel responsible for the interrupt can be determined by the raised interrupt line.

In addition, the registers ICM_IRQG_2 and ICM_IRQG_3 are mirrors for the TIM sub-module channel interrupts and typically not read out by the CPU, but they are readable.

38.24.2.4 MCS Interrupt Bundling

For complex signal output generation, the MCS sub-modules are used inside the GTM-IP. Each of these MCS sub-modules could have 8 channels with one interrupt line. This interrupt line is connected to the ICM sub-module and is feed through directly to the outside world.

In addition the interrupt line status for the first 8 channels of each MCS is shown in the ICM_IRQG_4 and ICM_IRQG_5 register. The interrupt line status for all used channels of each MCS are shown in the ICM_IRQG_MCS[i]_CI register. Typically, the interrupt source is determined by the corresponding interrupt line and the ICM_IRQ4(/_5) and ICM_IRQG_MCS[i]_CI register are typically not read out by the CPU, but they are readable.

38.24.2.5 TOM and ATOM Interrupt Bundling

For the TOM and ATOM sub-modules, the interrupts are bundled within the ICM sub-module a second time to reduce external interrupt lines. The interrupts are ORed in a manner that one GTM-IP external interrupt line represents two adjacent TOM or ATOM channel interrupts.

For TOM[i] and ATOM[i] the bundling is shown in **Table 38.459**.

Table 38.459 TOM[i] and ATOM[i] the bundling within ICM

TOM[i]-input IRQs	TOM-output IRQs (ORed)	ATOM[i]-input IRQs	ATOM-output IRQs (ORed)
[i]=0..number of TOM's-1		[i]=0..number of ATOM's-1	
TOM[i]_CH0_IRQ	GTM_TOM[i]_IRQ[0]	ATOM[i]_CH0_IRQ	GTM_ATOM[i]_IRQ[0]
TOM[i]_CH1_IRQ		ATOM[i]_CH1_IRQ	
TOM[i]_CH2_IRQ	GTM_TOM[i]_IRQ[1]	ATOM[i]_CH2_IRQ	GTM_ATOM[i]_IRQ[1]
TOM[i]_CH3_IRQ		ATOM[i]_CH3_IRQ	
TOM[i]_CH4_IRQ	GTM_TOM[i]_IRQ[2]	ATOM[i]_CH4_IRQ	GTM_ATOM[i]_IRQ[2]
TOM[i]_CH5_IRQ		ATOM[i]_CH5_IRQ	
TOM[i]_CH6_IRQ	GTM_TOM[i]_IRQ[3]	ATOM[i]_CH6_IRQ	GTM_ATOM[i]_IRQ[3]
TOM[i]_CH7_IRQ		ATOM[i]_CH7_IRQ	
TOM[i]_CH8_IRQ	GTM_TOM[i]_IRQ[4]		
TOM[i]_CH9_IRQ			
TOM[i]_CH10_IRQ	GTM_TOM[i]_IRQ[5]		
TOM[i]_CH11_IRQ			
TOM[i]_CH12_IRQ	GTM_TOM[i]_IRQ[6]		
TOM[i]_CH13_IRQ			
TOM[i]_CH14_IRQ	GTM_TOM[i]_IRQ[7]		
TOM[i]_CH15_IRQ			

The interrupts coming from the TOM[i] sub-modules are registered in the ICM_IRQG_6/ ICM_IRQG_7/ICM_IRQG_8 register. Always two TOM's are bundled in one ICM register, TOM0 and TOM1 are bundled in ICM_IRQG_6. To identify the TOM sub-module channel where the interrupt occurred, the CPU has to read out the ICM_IRQG_6(/_7/_8) register first before it goes to the TOM sub-module channel itself.

The ICM_IRQG_6(/_7/_8) register bits are cleared automatically, when their corresponding interrupt in the sub-module channels is cleared.

The interrupts coming from the ATOM[i] sub-modules are registered in the ICM_IRQG_9/ ICM_IRQG_10/ICM_IRQG_11 register. Always four ATOM's are bundled in one ICM register. ATOM0, ATOM1, ATOM2 and ATOM3 are bundled in ICM_IRQG_9. To identify the ATOM sub-module channel where the interrupt occurred, the CPU has to read out the ICM_IRQG_9(/_10/_11) register first before it goes to the ATOM sub-module channel itself.

The ICM_IRQG_9(/_10/_11) register bits are cleared automatically, when their corresponding interrupt in the sub-module channels is cleared.

In addition the interrupt line status of two 16 channels TOM are shown in each ICM_IRQG_TOM_[k]_CI (k:0 to 2) register, TOM0 and TOM1 are bundled in ICM_IRQG_TOM_0_CI. Typically, the interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_TOM_[k]_CI registers are typically not read out by the CPU, but they are readable.

In addition the interrupt line status of four 8 channels ATOM are shown in each ICM_IRQG_ATOM_[k]_CI (k: 0 to 2) register, ATOM0, ATOM1, ATOM2 and ATOM3 are bundled in ICM_IRQG_ATOM_0_CI. Typically, the interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_ATOM_[k]_CI registers are typically not read out by the CPU, but they are readable.

38.24.2.6 Module Error Interrupt Bundling

The Module Error Interrupt group handles the error interrupts coming from the BRC, FIFO, TIM, MCS, SPE, CMP, DPLL sub-module of the GTM-IP. The Module Error interrupts are additionally identified in the ICM_IRQG_MEI error interrupt group register. This register is typically not read out by the CPU, but it is readable.

In addition, the error interrupt line status for each SPE is shown in the ICM_IRQG_SPE_CEI register. Typically, the error interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_SPE_CEI register are typically not read out by the CPU, but they are readable.

38.24.2.7 FIFO Channel Error Interrupt Bundling

The FIFO Channel Error Interrupt group handles the error interrupts coming from the FIFO channel of the GTM-IP. The FIFO Channel Error interrupts are additionally identified in the ICM_IRQG_CEI0 error interrupt group register. This register is typically not read out by the CPU, but it is readable.

The ICM_IRQG_CEI0 register bits are cleared automatically, when their corresponding error interrupt in the sub-module channel is cleared.

In addition the error interrupt line status for 8 channels of each FIFO are shown in the ICM_IRQG_PSM_0_CEI register. Typically, the error interrupt source is determined by the corresponding interrupt line and the ICM_IRQG_PSM_0_CEI register are typically not read out by the CPU, but they are readable.

38.24.2.8 TIM Channel Error Interrupt Bundling

The TIM Channel Error Interrupt group handles the error interrupts coming from the TIM channel of the GTM-IP. The TIM Channel Error interrupts are additionally identified for the sub-modules TIM0, TIM1, TIM2 and TIM3 in the ICM_IRQG_CEI1 error interrupt group register and for the sub-modules TIM4, TIM5 and TIM6 in the ICM_IRQG_CEI2 error interrupt group register. These register are typically not read out by the CPU, but they are readable.

The ICM_IRQG_CEI1 and ICM_IRQG_CEI2 register bits are cleared automatically, when their corresponding error interrupt in the sub-module channel is cleared.

38.24.2.9 MCS Channel Error Interrupt Bundling

The MCS Channel Error Interrupt group handles the error interrupts coming from the MCS channel of the GTM-IP. All used 8 MCS Channel Error interrupts are additionally identified for each sub-modules MCS[i] in the ICM_IRQG_MCS[i]_CEI error interrupt group register. The first 8 MCS Channel Error interrupts are additionally identified for the sub-modules MCS0, MCS1, MCS2 and MCS3 in the ICM_IRQG_CEI3 error interrupt group register and for the sub-modules MCS4, MCS5, MCS6 and MCS7 in the ICM_IRQG_CEI4 error interrupt group register. These register are typically not read out by the CPU, but they are readable.

The ICM_IRQG_MCS[i]_CEI, ICM_IRQG_CEI3 and ICM_IRQG_CEI4 register bits are cleared automatically, when their corresponding error interrupt in the sub-module channel is cleared.

38.24.2.10 Error Interrupt Cluster Bundling

The Error Interrupt lines of up to 4 clusters are bundled in each ICM_IRQG_CLS_[i]_MEI. Actually each cluster collects one EIRQ of one TIM, MCS, SPE and FIFO. These register are typically not read out by the CPU, but they are readable.

38.24.3 ICM Interrupt Signals

Above table shows the GTM-IP interrupt lines that are visible at the outside of the IP.

Table 38.460 ICM interrupt Signals

Signal	Description
GTM_AEI_IRQ	AEI Shared interrupt
GTM_ARU_IRQ[2:0]	[0]: ARU_NEW_DATA0 Interrupt [1]: ARU_NEW_DATA1 Interrupt [2]: ARU_ACC_ACK Interrupt
GTM_BRC_IRQ	BRC Shared interrupt
GTM_CMP_IRQ	CMP Shared interrupt
GTM_SPE[i]_IRQ	SPE Shared interrupt (i: 0..number of SPE's-1)
GTM_PSM[i]_IRQ[x]	PSM Shared interrupts (x: 0…7) (i: 0..number of PSM's-1)
GTM_DPLL_IRQ[0]	*DPLL_DCGI:* DPLL direction change interrupt
GTM_DPLL_IRQ[1]	*DPLL_EDI;* DPLL enable or disable interrupt
GTM_DPLL_IRQ[2]	*DPLL_TINI:* DPLL *TRIG.* min. hold time (THMI) viol. detected
GTM_DPLL_IRQ[3]	*DPLL_TAXI:* DPLL *TRIG.* max. hold time (THMA) viol. detected
GTM_DPLL_IRQ[4]	*DPLL_SISI:* DPLL *STATE* inactive slope detected
GTM_DPLL_IRQ[5]	*DPLL_TISI:* DPLL *TRIGGER* inactive slope detected
GTM_DPLL_IRQ[6]	*DPLL_MSI:* DPLL Missing *STATE* interrupt
GTM_DPLL_IRQ[7]	*DPLL_MTI:* DPLL Missing *TRIGGER* interrupt
GTM_DPLL_IRQ[8]	*DPLL_SASI:* DPLL *STATE* active slope detected
GTM_DPLL_IRQ[9]	*DPLL_TASI:* DPLL *TRIG.* active slope det. while NTI_CNT is 0
GTM_DPLL_IRQ[10]	*DPLL_PWI:* DPLL Plausibility window (PVT) viol. int. of *TRIG.*
GTM_DPLL_IRQ[11]	*DPLL_W2I:* DPLL Write access to RAM region 2 interrupt
GTM_DPLL_IRQ[12]	*DPLL_W1I:* DPLL Write access to RAM region 1b or 1c int.
GTM_DPLL_IRQ[13]	*DPLL_GL1I:* DPLL Get of lock interrupt for *SUB_INC1*
GTM_DPLL_IRQ[14]	*DPLL_LL1I:* DPLL Lost of lock interrupt for *SUB_INC1*
GTM_DPLL_IRQ[15]	*DPLL_EI:* DPLL Error interrupt
GTM_DPLL_IRQ[16]	*DPLL_GL2I:* DPLL Get of lock interrupt for *SUB_INC2*
GTM_DPLL_IRQ[17]	*DPLL_LL2I:* DPLL Lost of lock interrupt for *SUB_INC2*
GTM_DPLL_IRQ[18]	*DPLL_TE0I:* DPLL *TRIGGER* event interrupt 0
GTM_DPLL_IRQ[19]	*DPLL_TE1I:* DPLL *TRIGGER* event interrupt 1
GTM_DPLL_IRQ[20]	*DPLL_TE2I:* DPLL *TRIGGER* event interrupt 2
GTM_DPLL_IRQ[21]	*DPLL_TE3I:* DPLL *TRIGGER* event interrupt 3
GTM_DPLL_IRQ[22]	*DPLL_TE4I;* DPLL *TRIGGER* event interrupt 4
GTM_DPLL_IRQ[23]	*DPLL_CDTI;* DPLL calculated duration interrupt for *TRIGGER*
GTM_DPLL_IRQ[24]	*DPLL_CDSI;* DPLL calculated duration interrupt for *STATE*
GTM_DPLL_IRQ[25]	*DPLL_TORI; TRIGGER* out of range interrupt
GTM_DPLL_IRQ[26]	*DPLL_SORI; STATE* out of range interrupt
GTM_TIM[i]_IRQ[x]	TIM Shared interrupts (i: 0..number of TIM's-1) (x: 0..7)
GTM_MCS[i]_IRQ[x]	MCS Interrupt for channel x (x: 0…8) (i: 0..number of MCS's-1)
GTM_TOM[i]_IRQ[x]	TOM Shared interrupts for x:0..7 = {ch0||ch1,...,ch14||ch15} (i: 0..number of TOM's-1)
GTM_ATOM[i]_IRQ[x]	ATOM Shared interrupts for x:0..3 = {ch0||ch1,...,ch6||ch7} (i: 0..number of ATOM's-1)
GTM_ERR_IRQ	GTM Error Interrupt

38.24.4 ICM Configuration Registers Overview

ICM contains following configuration registers:

Table 38.461 Register list (1/2)

Symbol	Register Name	Details in Section
ICM_IRQG_0	ICM Interrupt group register covering infrastructural and safety components (ARU, BRC, AEI, PSM0, PSM1, MAP, CMP,SPE)	**38.24.5.1**
ICM_IRQG_1	ICM Interrupt group register covering DPLL	**38.24.5.2**
ICM_IRQG_2	ICM Interrupt group register covering TIM0, TIM1, TIM2, TIM3	**38.24.5.3**
ICM_IRQG_3	ICM Interrupt group register covering TIM4, TIM5, TIM6, TIM7	**38.24.5.4**
ICM_IRQG_4	ICM Interrupt group register covering MCS0 to MCS3 sub-modules	**38.24.5.5**
ICM_IRQG_5	ICM Interrupt group register covering MCS4 to MCS6 sub-modules	**38.24.5.6**
ICM_IRQG_6	ICM Interrupt group register covering GTM-IP output sub-modules TOM0 to TOM1	**38.24.5.7**
ICM_IRQG_7	ICM Interrupt group register covering GTM-IP output sub-modules TOM2 to TOM3	**38.24.5.8**
ICM_IRQG_8	ICM Interrupt group register covering GTM-IP output sub-modules TOM4 to TOM5	**38.24.5.9**
ICM_IRQG_9	ICM Interrupt group register covering GTM-IP output sub-modules ATOM0 to ATOM3	**38.24.5.10**
ICM_IRQG_10	ICM Interrupt group register covering GTM-IP output sub-modules ATOM4 to ATOM7	**38.24.5.11**
ICM_IRQG_11	ICM Interrupt group register covering GTM-IP output sub-modules ATOM8 to ATOM11	**38.24.5.12**
ICM_IRQG_MEI	ICM Interrupt group register for module error interrupt information	**38.24.5.13**
ICM_IRQG_CEI0	ICM Interrupt group register 0 for channel error interrupt information	**38.24.5.14**
ICM_IRQG_CEI1	ICM Interrupt group register 1 for channel error interrupt information	**38.24.5.15**
ICM_IRQG_CEI2	ICM Interrupt group register 2 for channel error interrupt information	**38.24.5.16**
ICM_IRQG_CEI3	ICM Interrupt group register 3 for channel error interrupt information	**38.24.5.17**
ICM_IRQG_CEI4	ICM Interrupt group register 4 for channel error interrupt information	**38.24.5.18**
ICM_IRQG_MCS[i]_CI	ICM Interrupt group MCS i for Channel Interrupt information	**38.24.5.19**
ICM_IRQG_MCS[i]_CEI	ICM Interrupt group MCS i for Channel Error Interrupt information	**38.24.5.20**
ICM_IRQG_SPE_CI	ICM Interrupt group SPE for module Interrupt information	**38.24.5.21**
ICM_IRQG_SPE_CEI	ICM Interrupt group SPE for module Error Interrupt information	**38.24.5.22**
ICM_IRQG_PSM_0_CI	ICM Interrupt group PSM 0 for Channel Interrupt information of FIFO0, FIFO1, FIFO2	**38.24.5.23**
ICM_IRQG_PSM_1_CI	ICM Interrupt group PSM 0 for Channel Error Interrupt information of FIFO0, FIFO1, FIFO2	**38.24.5.24**

Table 38.461 Register list (2/2)

Symbol	Register Name	Details in Section
ICM_IRQG_TOM_[g]_CI	ICM Interrupt group TOM g for Channel Interrupt information of TOMm (m=2*g+(0 to 1))	**38.24.5.25**
ICM_IRQG_ATOM_[g]_CI	ICM Interrupt group ATOM g for Channel Interrupt information of ATOMm (m=4*g+(0 to 3))	**38.24.5.26**
ICM_IRQG_CLS_[g]_MEI	ICM Interrupt group for module Error Interrupt information for each TIMm, MCSm, SPEm, FIFOm (m=4*g+(0 to 3))	**38.24.5.27**

38.24.5 ICM Configuration Registers Description

38.24.5.1 ICM_IRQG_0

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00600_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSM1_CH7_IRQ	PSM1_CH6_IRQ	PSM1_CH5_IRQ	PSM1_CH4_IRQ	PSM1_CH3_IRQ	PSM1_CH2_IRQ	PSM1_CH1_IRQ	PSM1_CH0_IRQ	PSM0_CH7_IRQ	PSM0_CH6_IRQ	PSM0_CH5_IRQ	PSM0_CH4_IRQ	PSM0_CH3_IRQ	PSM0_CH2_IRQ	PSM0_CH1_IRQ	PSM0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SPE5_IRQ	SPE4_IRQ	SPE3_IRQ	SPE2_IRQ	SPE1_IRQ	SPE0_IRQ	CMP_IRQ	AEI_IRQ	BRC_IRQ	ARU_ACC_ACK_IRQ	ARU_NEW_DATA1_IRQ	ARU_NEW_DATA0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.462 ICM_IRQG_0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	PSM1_CH[k]_IRQ	PSM1 shared sub-module channel k interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** Set this bit represents an OR function of the four interrupt sources FIFO_EMPTY, FIFO_FULL, FIFO_LOWER_WM or FIFO_UPPER_WM of FIFO instance 0 channel 0.
23 to 16	PSM0_CH[k]_IRQ	PSM0 shared sub-module channel k interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** Set this bit represents an OR function of the four interrupt sources FIFO_EMPTY, FIFO_FULL, FIFO_LOWER_WM or FIFO_UPPER_WM of FIFO instance 0 channel 0.
15 to 12	Reserved	When read, the value after reset is returned.
11 to 6	SPE[k]_IRQ	SPE[k] shared sub-module interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** Set this bit represents an OR function of the five interrupt sources SPE_NIPD, SPE_DCHG, SPE_PERR, SPE_BIS or SPE_RCMP of SPE instance 0.
5	CMP_IRQ	CMP shared sub-module interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module.

Table 38.462 ICM_IRQG_0 Register Contents (2/2)

Bit Position	Bit Name	Function
4	AEI_IRQ	AEI_IRQ interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** Set this bit represents an OR function of the interrupt sources AEI_TO_XPT, AEI_USP_ADDR, AEI_IM_ADDR, AEI_USP_BE, AEIM_USP_ADDR, AEIM_IM_ADDR, AEIM_USP_BE, CLK_EN_ERR or CLK_PER_ERR.
3	BRC_IRQ	BRC shared sub-module interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module.
2	ARU_ACC_ACK_IRQ	ARU_ACC_ACK interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module.
1	ARU_NEW_DATA1_IRQ	ARU_NEW_DATA1 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module.
0	ARU_NEW_DATA0_IRQ	ARU_NEW_DATA0 interrupt 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module.

38.24.5.2 ICM_IRQG_1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00604_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DPLL_SORI_IRQ	DPLL_TORI_IRQ	DPLL_CDSI_IRQ	DPLL_CDTI_IRQ	DPLL_TE4I_IRQ	DPLL_TE3I_IRQ	DPLL_TE2I_IRQ	DPLL_TE1I_IRQ	DPLL_TE0I_IRQ	DPLL_LL2I_IRQ	DPLL_GL2I_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DPLL_EI_IRQ	DPLL_LL1I_IRQ	DPLL_GL1I_IRQ	DPLL_W1I_IRQ	DPLL_W2I_IRQ	DPLL_PWI_IRQ	DPLL_TASI_IRQ	DPLL_SASI_IRQ	DPLL_MTI_IRQ	DPLL_MSI_IRQ	DPLL_TISI_IRQ	DPLL_SISI_IRQ	DPLL_TAXI_IRQ	DPLL_TINI_IRQ	DPLL_EDI_IRQ	DPLL_DCGI_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.463 ICM_IRQG_1 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26	DPLL_SORI_IRQ	DPLL calculated duration interrupt for state. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
25	DPLL_TORI_IRQ	DPLL calculated duration interrupt for state. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
24	DPLL_CDSI_IRQ	DPLL calculated duration interrupt for state. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
23	DPLL_CDTI_IRQ	DPLL calculated duration interrupt for trigger. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
22	DPLL_TE4I_IRQ	TRIGGER event interrupt 4. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
21	DPLL_TE3I_IRQ	TRIGGER event interrupt 3. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
20	DPLL_TE2I_IRQ	TRIGGER event interrupt 2. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
19	DPLL_TE1I_IRQ	TRIGGER event interrupt 1. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
18	DPLL_TE0I_IRQ	TRIGGER event interrupt 0. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
17	DPLL_LL2I_IRQ	Lost of lock interrupt for SUB_INC2. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
16	DPLL_GL2I_IRQ	Get of lock interrupt for SUB_INC2. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module

Table 38.463 ICM_IRQG_1 Register Contents (2/3)

Bit Position	Bit Name	Function
15	DPLL_EI_IRQ	Error interrupt 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
14	DPLL_LL1I_IRQ	Lost of lock interrupt for SUB_INC1. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
13	DPLL_GL1I_IRQ	Get of lock interrupt for SUB_INC1. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
12	DPLL_W1I_IRQ	Write access to RAM region 1b or 1c interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
11	DPLL_W2I_IRQ	Write access to RAM region 2 interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
10	DPLL_PWI_IRQ	Plausibility window (PVT) violation interrupt of TRIGGER. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
9	DPLL_TASI_IRQ	TRIGGER active slope detected while NTI_CNT is zero. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
8	DPLL_SASI_IRQ	STATE active slope detected. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
7	DPLL_MTI_IRQ	Missing TRIGGER interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
6	DPLL_MSI_IRQ	Missing STATE interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
5	DPLL_TISI_IRQ	TRIGGER inactive slope detected interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
4	DPLL_SISI_IRQ	STATE inactive slope detected interrupt 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
3	DPLL_TAXI_IRQ	TRIGGER maximum hold time (THMA) violation detected interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
2	DPLL_TINI_IRQ	TRIGGER minimum hold time (THMI) violation detected interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module
1	DPLL_EDI_IRQ	DPLL enable/disable interrupt. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module **NOTE** Set this bit represents an OR function of the two interrupt sources DPLL_PDI or DPLL_PEI.

Table 38.463 ICM_IRQG_1 Register Contents (3/3)

Bit Position	Bit Name	Function
0	DPLL_DCGI_IRQ	TRIGGER direction change detected. 0: No interrupt occurred 1: Interrupt was raised by the corresponding sub-module **NOTE** This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module.

38.24.5.3 ICM_IRQG_2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00608_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIM3_CH7_IRQ	TIM3_CH6_IRQ	TIM3_CH5_IRQ	TIM3_CH4_IRQ	TIM3_CH3_IRQ	TIM3_CH2_IRQ	TIM3_CH1_IRQ	TIM3_CH0_IRQ	TIM2_CH7_IRQ	TIM2_CH6_IRQ	TIM2_CH5_IRQ	TIM2_CH4_IRQ	TIM2_CH3_IRQ	TIM2_CH2_IRQ	TIM2_CH1_IRQ	TIM2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIM1_CH7_IRQ	TIM1_CH6_IRQ	TIM1_CH5_IRQ	TIM1_CH4_IRQ	TIM1_CH3_IRQ	TIM1_CH2_IRQ	TIM1_CH1_IRQ	TIM1_CH0_IRQ	TIM0_CH7_IRQ	TIM0_CH6_IRQ	TIM0_CH5_IRQ	TIM0_CH4_IRQ	TIM0_CH3_IRQ	TIM0_CH2_IRQ	TIM0_CH1_IRQ	TIM0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.464 ICM_IRQG_2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TIM3_CH7_IRQ	TIM3 shared interrupt channel 7. See bit 0.
30	TIM3_CH6_IRQ	TIM3 shared interrupt channel 6. See bit 0.
29	TIM3_CH5_IRQ	TIM3 shared interrupt channel 5. See bit 0.
28	TIM3_CH4_IRQ	TIM3 shared interrupt channel 4. See bit 0.
27	TIM3_CH3_IRQ	TIM3 shared interrupt channel 3. See bit 0.
26	TIM3_CH2_IRQ	TIM3 shared interrupt channel 2. See bit 0.
25	TIM3_CH1_IRQ	TIM3 shared interrupt channel 1. See bit 0.
24	TIM3_CH0_IRQ	TIM3 shared interrupt channel 0. See bit 0.
23	TIM2_CH7_IRQ	TIM3 shared interrupt channel 7. See bit 0.
22	TIM2_CH6_IRQ	TIM3 shared interrupt channel 6. See bit 0.
21	TIM2_CH5_IRQ	TIM3 shared interrupt channel 5. See bit 0.
20	TIM2_CH4_IRQ	TIM3 shared interrupt channel 4. See bit 0.
19	TIM2_CH3_IRQ	TIM3 shared interrupt channel 3. See bit 0.
18	TIM2_CH2_IRQ	TIM3 shared interrupt channel 2. See bit 0.
17	TIM2_CH1_IRQ	TIM3 shared interrupt channel 1. See bit 0.
16	TIM2_CH0_IRQ	TIM3 shared interrupt channel 0. See bit 0.
15	TIM1_CH7_IRQ	TIM1 shared interrupt channel 7. See bit 0.
14	TIM1_CH6_IRQ	TIM1 shared interrupt channel 6. See bit 0.
13	TIM1_CH5_IRQ	TIM1 shared interrupt channel 5. See bit 0.
12	TIM1_CH4_IRQ	TIM1 shared interrupt channel 4. See bit 0.
11	TIM1_CH3_IRQ	TIM1 shared interrupt channel 3. See bit 0.
10	TIM1_CH2_IRQ	TIM1 shared interrupt channel 2. See bit 0.
9	TIM1_CH1_IRQ	TIM1 shared interrupt channel 1. See bit 0.
8	TIM1_CH0_IRQ	TIM1 shared interrupt channel 0. See bit 0.
7	TIM0_CH7_IRQ	TIM0 shared interrupt channel 7. See bit 0.
6	TIM0_CH6_IRQ	TIM0 shared interrupt channel 6. See bit 0.
5	TIM0_CH5_IRQ	TIM0 shared interrupt channel 5. See bit 0.
4	TIM0_CH4_IRQ	TIM0 shared interrupt channel 4. See bit 0.

Table 38.464 ICM_IRQG_2 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TIM0_CH3_IRQ	TIM0 shared interrupt channel 3. See bit 0.
2	TIM0_CH2_IRQ	TIM0 shared interrupt channel 2. See bit 0.
1	TIM0_CH1_IRQ	TIM0 shared interrupt channel 1. See bit 0.
0	TIM0_CH0_IRQ	TIM0 shared interrupt channel 0. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TODETx_IRQ of TIM instance 0 channel x.

38.24.5.4 ICM_IRQG_3

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0060C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIM7_CH7_IRQ	TIM7_CH6_IRQ	TIM7_CH5_IRQ	TIM7_CH4_IRQ	TIM7_CH3_IRQ	TIM7_CH2_IRQ	TIM7_CH1_IRQ	TIM7_CH0_IRQ	TIM6_CH7_IRQ	TIM6_CH6_IRQ	TIM6_CH5_IRQ	TIM6_CH4_IRQ	TIM6_CH3_IRQ	TIM6_CH2_IRQ	TIM6_CH1_IRQ	TIM6_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIM5_CH7_IRQ	TIM5_CH6_IRQ	TIM5_CH5_IRQ	TIM5_CH4_IRQ	TIM5_CH3_IRQ	TIM5_CH2_IRQ	TIM5_CH1_IRQ	TIM5_CH0_IRQ	TIM4_CH7_IRQ	TIM4_CH6_IRQ	TIM4_CH5_IRQ	TIM4_CH4_IRQ	TIM4_CH3_IRQ	TIM4_CH2_IRQ	TIM4_CH1_IRQ	TIM4_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.465 ICM_IRQG_3 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TIM7_CH7_IRQ	TIM7 shared interrupt channel 7. See bit 0.
30	TIM7_CH6_IRQ	TIM7 shared interrupt channel 6. See bit 0.
29	TIM7_CH5_IRQ	TIM7 shared interrupt channel 5. See bit 0.
28	TIM7_CH4_IRQ	TIM7 shared interrupt channel 4. See bit 0.
27	TIM7_CH3_IRQ	TIM7 shared interrupt channel 3. See bit 0.
26	TIM7_CH2_IRQ	TIM7 shared interrupt channel 2. See bit 0.
25	TIM7_CH1_IRQ	TIM7 shared interrupt channel 1. See bit 0.
24	TIM7_CH0_IRQ	TIM7 shared interrupt channel 0. See bit 0.
23	TIM6_CH7_IRQ	TIM6 shared interrupt channel 7. See bit 0.
22	TIM6_CH6_IRQ	TIM6 shared interrupt channel 6. See bit 0.
21	TIM6_CH5_IRQ	TIM6 shared interrupt channel 5. See bit 0.
20	TIM6_CH4_IRQ	TIM6 shared interrupt channel 4. See bit 0.
19	TIM6_CH3_IRQ	TIM6 shared interrupt channel 3. See bit 0.
18	TIM6_CH2_IRQ	TIM6 shared interrupt channel 2. See bit 0.
17	TIM6_CH1_IRQ	TIM6 shared interrupt channel 1. See bit 0.
16	TIM6_CH0_IRQ	TIM6 shared interrupt channel 0. See bit 0.
15	TIM5_CH7_IRQ	TIM5 shared interrupt channel 7. See bit 0.
14	TIM5_CH6_IRQ	TIM5 shared interrupt channel 6. See bit 0.
13	TIM5_CH5_IRQ	TIM5 shared interrupt channel 5. See bit 0.
12	TIM5_CH4_IRQ	TIM5 shared interrupt channel 4. See bit 0.
11	TIM5_CH3_IRQ	TIM5 shared interrupt channel 3. See bit 0.
10	TIM5_CH2_IRQ	TIM5 shared interrupt channel 2. See bit 0.
9	TIM5_CH1_IRQ	TIM5 shared interrupt channel 1. See bit 0.
8	TIM5_CH0_IRQ	TIM5 shared interrupt channel 0. See bit 0.
7	TIM4_CH7_IRQ	TIM4 shared interrupt channel 7. See bit 0.
6	TIM4_CH6_IRQ	TIM4 shared interrupt channel 6. See bit 0.
5	TIM4_CH5_IRQ	TIM4 shared interrupt channel 5. See bit 0.
4	TIM4_CH4_IRQ	TIM4 shared interrupt channel 4. See bit 0.

Table 38.465 ICM_IRQG_3 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TIM4_CH3_IRQ	TIM4 shared interrupt channel 3. See bit 0.
2	TIM4_CH2_IRQ	TIM4 shared interrupt channel 2. See bit 0.
1	TIM4_CH1_IRQ	TIM4 shared interrupt channel 1. See bit 0.
0	TIM4_CH0_IRQ	TIM4 shared interrupt channel 0. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the six interrupt sources NEWVALx_IRQ, ECNTOFLx_IRQ, CNTOFLx_IRQ, GPRXOFLx_IRQ, GLITCHDETx_IRQ or TODETx_IRQ of TIM instance 4 channel x.

38.24.5.5 ICM_IRQG_4

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00610_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCS3_CH7_IRQ	MCS3_CH6_IRQ	MCS3_CH5_IRQ	MCS3_CH4_IRQ	MCS3_CH3_IRQ	MCS3_CH2_IRQ	MCS3_CH1_IRQ	MCS3_CH0_IRQ	MCS2_CH7_IRQ	MCS2_CH6_IRQ	MCS2_CH5_IRQ	MCS2_CH4_IRQ	MCS2_CH3_IRQ	MCS2_CH2_IRQ	MCS2_CH1_IRQ	MCS2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCS1_CH7_IRQ	MCS1_CH6_IRQ	MCS1_CH5_IRQ	MCS1_CH4_IRQ	MCS1_CH3_IRQ	MCS1_CH2_IRQ	MCS1_CH1_IRQ	MCS1_CH0_IRQ	MCS0_CH7_IRQ	MCS0_CH6_IRQ	MCS0_CH5_IRQ	MCS0_CH4_IRQ	MCS0_CH3_IRQ	MCS0_CH2_IRQ	MCS0_CH1_IRQ	MCS0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.466 ICM_IRQG_4 Register Contents (1/2)

Bit Position	Bit Name	Function
31	MCS3_CH7_IRQ	MCS3 channel 7 interrupt. See bit 0.
30	MCS3_CH6_IRQ	MCS3 channel 6 interrupt. See bit 0.
29	MCS3_CH5_IRQ	MCS3 channel 5 interrupt. See bit 0.
28	MCS3_CH4_IRQ	MCS3 channel 4 interrupt. See bit 0.
27	MCS3_CH3_IRQ	MCS3 channel 3 interrupt. See bit 0.
26	MCS3_CH2_IRQ	MCS3 channel 2 interrupt. See bit 0.
25	MCS3_CH1_IRQ	MCS3 channel 1 interrupt. See bit 0.
24	MCS3_CH0_IRQ	MCS3 channel 0 interrupt. See bit 0.
23	MCS2_CH7_IRQ	MCS2 channel 7 interrupt. See bit 0.
22	MCS2_CH6_IRQ	MCS2 channel 6 interrupt. See bit 0.
21	MCS2_CH5_IRQ	MCS2 channel 5 interrupt. See bit 0.
20	MCS2_CH4_IRQ	MCS2 channel 4 interrupt. See bit 0.
19	MCS2_CH3_IRQ	MCS2 channel 3 interrupt. See bit 0.
18	MCS2_CH2_IRQ	MCS2 channel 2 interrupt. See bit 0.
17	MCS2_CH1_IRQ	MCS2 channel 1 interrupt. See bit 0.
16	MCS2_CH0_IRQ	MCS2 channel 0 interrupt. See bit 0.
15	MCS1_CH7_IRQ	MCS1 channel 7 interrupt. See bit 0.
14	MCS1_CH6_IRQ	MCS1 channel 6 interrupt. See bit 0.
13	MCS1_CH5_IRQ	MCS1 channel 5 interrupt. See bit 0.
12	MCS1_CH4_IRQ	MCS1 channel 4 interrupt. See bit 0.
11	MCS1_CH3_IRQ	MCS1 channel 3 interrupt. See bit 0.
10	MCS1_CH2_IRQ	MCS1 channel 2 interrupt. See bit 0.
9	MCS1_CH1_IRQ	MCS1 channel 1 interrupt. See bit 0.
8	MCS1_CH0_IRQ	MCS1 channel 0 interrupt. See bit 0.
7	MCS0_CH7_IRQ	MCS0 channel 7 interrupt. See bit 0.
6	MCS0_CH6_IRQ	MCS0 channel 6 interrupt. See bit 0.
5	MCS0_CH5_IRQ	MCS0 channel 5 interrupt. See bit 0.
4	MCS0_CH4_IRQ	MCS0 channel 4 interrupt. See bit 0.

Table 38.466 ICM_IRQG_4 Register Contents (2/2)

Bit Position	Bit Name	Function
3	MCS0_CH3_IRQ	MCS0 channel 3 interrupt. See bit 0.
2	MCS0_CH2_IRQ	MCS0 channel 2 interrupt. See bit 0.
1	MCS0_CH1_IRQ	MCS0 channel 1 interrupt. See bit 0.
0	MCS0_CH0_IRQ	MCS0 channel 0 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the three interrupt sources MCS_IRQ, STK_ERR_IRQ or ERR_IRQ of MCS instance 0 channel k.

38.24.5.6 ICM_IRQG_5

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00614_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCS7_CH7_IRQ	MCS7_CH6_IRQ	MCS7_CH5_IRQ	MCS7_CH4_IRQ	MCS7_CH3_IRQ	MCS7_CH2_IRQ	MCS7_CH1_IRQ	MCS7_CH0_IRQ	MCS6_CH7_IRQ	MCS6_CH6_IRQ	MCS6_CH5_IRQ	MCS6_CH4_IRQ	MCS6_CH3_IRQ	MCS6_CH2_IRQ	MCS6_CH1_IRQ	MCS6_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCS5_CH7_IRQ	MCS5_CH6_IRQ	MCS5_CH5_IRQ	MCS5_CH4_IRQ	MCS5_CH3_IRQ	MCS5_CH2_IRQ	MCS5_CH1_IRQ	MCS5_CH0_IRQ	MCS4_CH7_IRQ	MCS4_CH6_IRQ	MCS4_CH5_IRQ	MCS4_CH4_IRQ	MCS4_CH3_IRQ	MCS4_CH2_IRQ	MCS4_CH1_IRQ	MCS4_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.467 ICM_IRQG_5 Register Contents (1/2)

Bit Position	Bit Name	Function
31	MCS7_CH7_IRQ	MCS7 channel 7 interrupt. See bit 0.
30	MCS7_CH6_IRQ	MCS7 channel 6 interrupt. See bit 0.
29	MCS7_CH5_IRQ	MCS7 channel 5 interrupt. See bit 0.
28	MCS7_CH4_IRQ	MCS7 channel 4 interrupt. See bit 0.
27	MCS7_CH3_IRQ	MCS7 channel 3 interrupt. See bit 0.
26	MCS7_CH2_IRQ	MCS7 channel 2 interrupt. See bit 0.
25	MCS7_CH1_IRQ	MCS7 channel 1 interrupt. See bit 0.
24	MCS7_CH0_IRQ	MCS7 channel 0 interrupt. See bit 0.
23	MCS6_CH7_IRQ	MCS6 channel 7 interrupt. See bit 0.
22	MCS6_CH6_IRQ	MCS6 channel 6 interrupt. See bit 0.
21	MCS6_CH5_IRQ	MCS6 channel 5 interrupt. See bit 0.
20	MCS6_CH4_IRQ	MCS6 channel 4 interrupt. See bit 0.
19	MCS6_CH3_IRQ	MCS6 channel 3 interrupt. See bit 0.
18	MCS6_CH2_IRQ	MCS6 channel 2 interrupt. See bit 0.
17	MCS6_CH1_IRQ	MCS6 channel 1 interrupt. See bit 0.
16	MCS6_CH0_IRQ	MCS6 channel 0 interrupt. See bit 0.
15	MCS5_CH7_IRQ	MCS5 channel 7 interrupt. See bit 0.
14	MCS5_CH6_IRQ	MCS5 channel 6 interrupt. See bit 0.
13	MCS5_CH5_IRQ	MCS5 channel 5 interrupt. See bit 0.
12	MCS5_CH4_IRQ	MCS5 channel 4 interrupt. See bit 0.
11	MCS5_CH3_IRQ	MCS5 channel 3 interrupt. See bit 0.
10	MCS5_CH2_IRQ	MCS5 channel 2 interrupt. See bit 0.
9	MCS5_CH1_IRQ	MCS5 channel 1 interrupt. See bit 0.
8	MCS5_CH0_IRQ	MCS5 channel 0 interrupt. See bit 0.
7	MCS4_CH7_IRQ	MCS4 channel 7 interrupt. See bit 0.
6	MCS4_CH6_IRQ	MCS4 channel 6 interrupt. See bit 0.
5	MCS4_CH5_IRQ	MCS4 channel 5 interrupt. See bit 0.
4	MCS4_CH4_IRQ	MCS4 channel 4 interrupt. See bit 0.

Table 38.467 ICM_IRQG_5 Register Contents (2/2)

Bit Position	Bit Name	Function
3	MCS4_CH3_IRQ	MCS4 channel 3 interrupt. See bit 0.
2	MCS4_CH2_IRQ	MCS4 channel 2 interrupt. See bit 0.
1	MCS4_CH1_IRQ	MCS4 channel 1 interrupt. See bit 0.
0	MCS4_CH0_IRQ	MCS4 channel 0 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the three interrupt sources MCS_IRQ, STK_ERR_IRQ or ERR_IRQ of MCS instance 4 channel 0.

38.24.5.7 ICM_IRQG_6

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00618_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOM1_CH15_IRQ	TOM1_CH14_IRQ	TOM1_CH13_IRQ	TOM1_CH12_IRQ	TOM1_CH11_IRQ	TOM1_CH10_IRQ	TOM1_CH9_IRQ	TOM1_CH8_IRQ	TOM1_CH7_IRQ	TOM1_CH6_IRQ	TOM1_CH5_IRQ	TOM1_CH4_IRQ	TOM1_CH3_IRQ	TOM1_CH2_IRQ	TOM1_CH1_IRQ	TOM1_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM0_CH15_IRQ	TOM0_CH14_IRQ	TOM0_CH13_IRQ	TOM0_CH12_IRQ	TOM0_CH11_IRQ	TOM0_CH10_IRQ	TOM0_CH9_IRQ	TOM0_CH8_IRQ	TOM0_CH7_IRQ	TOM0_CH6_IRQ	TOM0_CH5_IRQ	TOM0_CH4_IRQ	TOM0_CH3_IRQ	TOM0_CH2_IRQ	TOM0_CH1_IRQ	TOM0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.468 ICM_IRQG_6 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TOM1_CH15_IRQ	TOM1 channel 15 shared interrupt. See bit 0.
30	TOM1_CH14_IRQ	TOM1 channel 14 shared interrupt. See bit 0.
29	TOM1_CH13_IRQ	TOM1 channel 13 shared interrupt. See bit 0.
28	TOM1_CH12_IRQ	TOM1 channel 12 shared interrupt. See bit 0.
27	TOM1_CH11_IRQ	TOM1 channel 11 shared interrupt. See bit 0.
26	TOM1_CH10_IRQ	TOM1 channel 10 shared interrupt. See bit 0.
25	TOM1_CH9_IRQ	TOM1 channel 9 shared interrupt. See bit 0.
24	TOM1_CH8_IRQ	TOM1 channel 8 shared interrupt. See bit 0.
23	TOM1_CH7_IRQ	TOM1 channel 7 shared interrupt. See bit 0.
22	TOM1_CH6_IRQ	TOM1 channel 6 shared interrupt. See bit 0.
21	TOM1_CH5_IRQ	TOM1 channel 5 shared interrupt. See bit 0.
20	TOM1_CH4_IRQ	TOM1 channel 4 shared interrupt. See bit 0.
19	TOM1_CH3_IRQ	TOM1 channel 3 shared interrupt. See bit 0.
18	TOM1_CH2_IRQ	TOM1 channel 2 shared interrupt. See bit 0.
17	TOM1_CH1_IRQ	TOM1 channel 1 shared interrupt. See bit 0.
16	TOM1_CH0_IRQ	TOM1 channel 0 shared interrupt. See bit 0.
15	TOM0_CH15_IRQ	TOM0 channel 15 shared interrupt. See bit 0.
14	TOM0_CH14_IRQ	TOM0 channel 14 shared interrupt. See bit 0.
13	TOM0_CH13_IRQ	TOM0 channel 13 shared interrupt. See bit 0.
12	TOM0_CH12_IRQ	TOM0 channel 12 shared interrupt. See bit 0.
11	TOM0_CH11_IRQ	TOM0 channel 11 shared interrupt. See bit 0.
10	TOM0_CH10_IRQ	TOM0 channel 10 shared interrupt. See bit 0.
9	TOM0_CH9_IRQ	TOM0 channel 9 shared interrupt. See bit 0.
8	TOM0_CH8_IRQ	TOM0 channel 8 shared interrupt. See bit 0.
7	TOM0_CH7_IRQ	TOM0 channel 7 shared interrupt. See bit 0.
6	TOM0_CH6_IRQ	TOM0 channel 6 shared interrupt. See bit 0.
5	TOM0_CH5_IRQ	TOM0 channel 5 shared interrupt. See bit 0.
4	TOM0_CH4_IRQ	TOM0 channel 4 shared interrupt. See bit 0.

Table 38.468 ICM_IRQG_6 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TOM0_CH3_IRQ	TOM0 channel 3 shared interrupt. See bit 0.
2	TOM0_CH2_IRQ	TOM0 channel 2 shared interrupt. See bit 0.
1	TOM0_CH1_IRQ	TOM0 channel 1 shared interrupt. See bit 0.
0	TOM0_CH0_IRQ	TOM0 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources TOM_CCU0TCx_IRQ or TOM_CCU1TCx_IRQ of TOM instance 0 channel x.

38.24.5.8 ICM_IRQG_7

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0061C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOM3_CH15_IRQ	TOM3_CH14_IRQ	TOM3_CH13_IRQ	TOM3_CH12_IRQ	TOM3_CH11_IRQ	TOM3_CH10_IRQ	TOM3_CH9_IRQ	TOM3_CH8_IRQ	TOM3_CH7_IRQ	TOM3_CH6_IRQ	TOM3_CH5_IRQ	TOM3_CH4_IRQ	TOM3_CH3_IRQ	TOM3_CH2_IRQ	TOM3_CH1_IRQ	TOM3_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM2_CH15_IRQ	TOM2_CH14_IRQ	TOM2_CH13_IRQ	TOM2_CH12_IRQ	TOM2_CH11_IRQ	TOM2_CH10_IRQ	TOM2_CH9_IRQ	TOM2_CH8_IRQ	TOM2_CH7_IRQ	TOM2_CH6_IRQ	TOM2_CH5_IRQ	TOM2_CH4_IRQ	TOM2_CH3_IRQ	TOM2_CH2_IRQ	TOM2_CH1_IRQ	TOM2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.469 ICM_IRQG_7 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TOM3_CH15_IRQ	TOM3 channel 15 shared interrupt. See bit 0.
30	TOM3_CH14_IRQ	TOM3 channel 14 shared interrupt. See bit 0.
29	TOM3_CH13_IRQ	TOM3 channel 13 shared interrupt. See bit 0.
28	TOM3_CH12_IRQ	TOM3 channel 12 shared interrupt. See bit 0.
27	TOM3_CH11_IRQ	TOM3 channel 11 shared interrupt. See bit 0.
26	TOM3_CH10_IRQ	TOM3 channel 10 shared interrupt. See bit 0.
25	TOM3_CH9_IRQ	TOM3 channel 9 shared interrupt. See bit 0.
24	TOM3_CH8_IRQ	TOM3 channel 8 shared interrupt. See bit 0.
23	TOM3_CH7_IRQ	TOM3 channel 7 shared interrupt. See bit 0.
22	TOM3_CH6_IRQ	TOM3 channel 6 shared interrupt. See bit 0.
21	TOM3_CH5_IRQ	TOM3 channel 5 shared interrupt. See bit 0.
20	TOM3_CH4_IRQ	TOM3 channel 4 shared interrupt. See bit 0.
19	TOM3_CH3_IRQ	TOM3 channel 3 shared interrupt. See bit 0.
18	TOM3_CH2_IRQ	TOM3 channel 2 shared interrupt. See bit 0.
17	TOM3_CH1_IRQ	TOM3 channel 1 shared interrupt. See bit 0.
16	TOM3_CH0_IRQ	TOM3 channel 0 shared interrupt. See bit 0.
15	TOM2_CH15_IRQ	TOM2 channel 15 shared interrupt. See bit 0.
14	TOM2_CH14_IRQ	TOM2 channel 14 shared interrupt. See bit 0.
13	TOM2_CH13_IRQ	TOM2 channel 13 shared interrupt. See bit 0.
12	TOM2_CH12_IRQ	TOM2 channel 12 shared interrupt. See bit 0.
11	TOM2_CH11_IRQ	TOM2 channel 11 shared interrupt. See bit 0.
10	TOM2_CH10_IRQ	TOM2 channel 10 shared interrupt. See bit 0.
9	TOM2_CH9_IRQ	TOM2 channel 9 shared interrupt. See bit 0.
8	TOM2_CH8_IRQ	TOM2 channel 8 shared interrupt. See bit 0.
7	TOM2_CH7_IRQ	TOM2 channel 7 shared interrupt. See bit 0.
6	TOM2_CH6_IRQ	TOM2 channel 6 shared interrupt. See bit 0.
5	TOM2_CH5_IRQ	TOM2 channel 5 shared interrupt. See bit 0.
4	TOM2_CH4_IRQ	TOM2 channel 4 shared interrupt. See bit 0.

Table 38.469 ICM_IRQG_7 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TOM2_CH3_IRQ	TOM2 channel 3 shared interrupt. See bit 0.
2	TOM2_CH2_IRQ	TOM2 channel 2 shared interrupt. See bit 0.
1	TOM2_CH1_IRQ	TOM2 channel 1 shared interrupt. See bit 0.
0	TOM2_CH0_IRQ	TOM2 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources TOM_CCU0TCx_IRQ or TOM_CCU1TCx_IRQ of TOM instance 2 channel x.

38.24.5.9 ICM_IRQG_8

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00620_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOM5_CH15_IRQ	TOM5_CH14_IRQ	TOM5_CH13_IRQ	TOM5_CH12_IRQ	TOM5_CH11_IRQ	TOM5_CH10_IRQ	TOM5_CH9_IRQ	TOM5_CH8_IRQ	TOM5_CH7_IRQ	TOM5_CH6_IRQ	TOM5_CH5_IRQ	TOM5_CH4_IRQ	TOM5_CH3_IRQ	TOM5_CH2_IRQ	TOM5_CH1_IRQ	TOM5_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM4_CH15_IRQ	TOM4_CH14_IRQ	TOM4_CH13_IRQ	TOM4_CH12_IRQ	TOM4_CH11_IRQ	TOM4_CH10_IRQ	TOM4_CH9_IRQ	TOM4_CH8_IRQ	TOM4_CH7_IRQ	TOM4_CH6_IRQ	TOM4_CH5_IRQ	TOM4_CH4_IRQ	TOM4_CH3_IRQ	TOM4_CH2_IRQ	TOM4_CH1_IRQ	TOM4_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.470 ICM_IRQG_8 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TOM5_CH15_IRQ	TOM5 channel 15 shared interrupt. See bit 0.
30	TOM5_CH14_IRQ	TOM5 channel 14 shared interrupt. See bit 0.
29	TOM5_CH13_IRQ	TOM5 channel 13 shared interrupt. See bit 0.
28	TOM5_CH12_IRQ	TOM5 channel 12 shared interrupt. See bit 0.
27	TOM5_CH11_IRQ	TOM5 channel 11 shared interrupt. See bit 0.
26	TOM5_CH10_IRQ	TOM5 channel 10 shared interrupt. See bit 0.
25	TOM5_CH9_IRQ	TOM5 channel 9 shared interrupt. See bit 0.
24	TOM5_CH8_IRQ	TOM5 channel 8 shared interrupt. See bit 0.
23	TOM5_CH7_IRQ	TOM5 channel 7 shared interrupt. See bit 0.
22	TOM5_CH6_IRQ	TOM5 channel 6 shared interrupt. See bit 0.
21	TOM5_CH5_IRQ	TOM5 channel 5 shared interrupt. See bit 0.
20	TOM5_CH4_IRQ	TOM5 channel 4 shared interrupt. See bit 0.
19	TOM5_CH3_IRQ	TOM5 channel 3 shared interrupt. See bit 0.
18	TOM5_CH2_IRQ	TOM5 channel 2 shared interrupt. See bit 0.
17	TOM5_CH1_IRQ	TOM5 channel 1 shared interrupt. See bit 0.
16	TOM5_CH0_IRQ	TOM5 channel 0 shared interrupt. See bit 0.
15	TOM4_CH15_IRQ	TOM4 channel 15 shared interrupt. See bit 0.
14	TOM4_CH14_IRQ	TOM4 channel 14 shared interrupt. See bit 0.
13	TOM4_CH13_IRQ	TOM4 channel 13 shared interrupt. See bit 0.
12	TOM4_CH12_IRQ	TOM4 channel 12 shared interrupt. See bit 0.
11	TOM4_CH11_IRQ	TOM4 channel 11 shared interrupt. See bit 0.
10	TOM4_CH10_IRQ	TOM4 channel 10 shared interrupt. See bit 0.
9	TOM4_CH9_IRQ	TOM4 channel 9 shared interrupt. See bit 0.
8	TOM4_CH8_IRQ	TOM4 channel 8 shared interrupt. See bit 0.
7	TOM4_CH7_IRQ	TOM4 channel 7 shared interrupt. See bit 0.
6	TOM4_CH6_IRQ	TOM4 channel 6 shared interrupt. See bit 0.
5	TOM4_CH5_IRQ	TOM4 channel 5 shared interrupt. See bit 0.
4	TOM4_CH4_IRQ	TOM4 channel 4 shared interrupt. See bit 0.

Table 38.470 ICM_IRQG_8 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TOM4_CH3_IRQ	TOM4 channel 3 shared interrupt. See bit 0.
2	TOM4_CH2_IRQ	TOM4 channel 2 shared interrupt. See bit 0.
1	TOM4_CH1_IRQ	TOM4 channel 1 shared interrupt. See bit 0.
0	TOM4_CH0_IRQ	TOM4 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources TOM_CCU0TCx_IRQ or TOM_CCU1TCx_IRQ of TOM instance 4 channel x.

38.24.5.10 ICM_IRQG_9

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00624_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ATOM3_CH7_IRQ	ATOM3_CH6_IRQ	ATOM3_CH5_IRQ	ATOM3_CH4_IRQ	ATOM3_CH3_IRQ	ATOM3_CH2_IRQ	ATOM3_CH1_IRQ	ATOM3_CH0_IRQ	ATOM2_CH7_IRQ	ATOM2_CH6_IRQ	ATOM2_CH5_IRQ	ATOM2_CH4_IRQ	ATOM2_CH3_IRQ	ATOM2_CH2_IRQ	ATOM2_CH1_IRQ	ATOM2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATOM1_CH7_IRQ	ATOM1_CH6_IRQ	ATOM1_CH5_IRQ	ATOM1_CH4_IRQ	ATOM1_CH3_IRQ	ATOM1_CH2_IRQ	ATOM1_CH1_IRQ	ATOM1_CH0_IRQ	ATOM0_CH7_IRQ	ATOM0_CH6_IRQ	ATOM0_CH5_IRQ	ATOM0_CH4_IRQ	ATOM0_CH3_IRQ	ATOM0_CH2_IRQ	ATOM0_CH1_IRQ	ATOM0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.471 ICM_IRQG_9 Register Contents (1/2)

Bit Position	Bit Name	Function
31	ATOM3_CH7_IRQ	ATOM3 channel 7 shared interrupt. See bit 0.
30	ATOM3_CH6_IRQ	ATOM3 channel 6 shared interrupt. See bit 0.
29	ATOM3_CH5_IRQ	ATOM3 channel 5 shared interrupt. See bit 0.
28	ATOM3_CH4_IRQ	ATOM3 channel 4 shared interrupt. See bit 0.
27	ATOM3_CH3_IRQ	ATOM3 channel 3 shared interrupt. See bit 0.
26	ATOM3_CH2_IRQ	ATOM3 channel 2 shared interrupt. See bit 0.
25	ATOM3_CH1_IRQ	ATOM3 channel 1 shared interrupt. See bit 0.
24	ATOM3_CH0_IRQ	ATOM3 channel 0 shared interrupt. See bit 0.
23	ATOM2_CH7_IRQ	ATOM2 channel 7 shared interrupt. See bit 0.
22	ATOM2_CH6_IRQ	ATOM2 channel 6 shared interrupt. See bit 0.
21	ATOM2_CH5_IRQ	ATOM2 channel 5 shared interrupt. See bit 0.
20	ATOM2_CH4_IRQ	ATOM2 channel 4 shared interrupt. See bit 0.
19	ATOM2_CH3_IRQ	ATOM2 channel 3 shared interrupt. See bit 0.
18	ATOM2_CH2_IRQ	ATOM2 channel 2 shared interrupt. See bit 0.
17	ATOM2_CH1_IRQ	ATOM2 channel 1 shared interrupt. See bit 0.
16	ATOM2_CH0_IRQ	ATOM2 channel 0 shared interrupt. See bit 0.
15	ATOM1_CH7_IRQ	ATOM1 channel 7 shared interrupt. See bit 0.
14	ATOM1_CH6_IRQ	ATOM1 channel 6 shared interrupt. See bit 0.
13	ATOM1_CH5_IRQ	ATOM1 channel 5 shared interrupt. See bit 0.
12	ATOM1_CH4_IRQ	ATOM1 channel 4 shared interrupt. See bit 0.
11	ATOM1_CH3_IRQ	ATOM1 channel 3 shared interrupt. See bit 0.
10	ATOM1_CH2_IRQ	ATOM1 channel 2 shared interrupt. See bit 0.
9	ATOM1_CH1_IRQ	ATOM1 channel 1 shared interrupt. See bit 0.
8	ATOM1_CH0_IRQ	ATOM1 channel 0 shared interrupt. See bit 0.
7	ATOM0_CH7_IRQ	ATOM0 channel 7 shared interrupt. See bit 0.
6	ATOM0_CH6_IRQ	ATOM0 channel 6 shared interrupt. See bit 0.
5	ATOM0_CH5_IRQ	ATOM0 channel 5 shared interrupt. See bit 0.
4	ATOM0_CH4_IRQ	ATOM0 channel 4 shared interrupt. See bit 0.

Table 38.471 ICM_IRQG_9 Register Contents (2/2)

Bit Position	Bit Name	Function
3	ATOM0_CH3_IRQ	ATOM0 channel 3 shared interrupt. See bit 0.
2	ATOM0_CH2_IRQ	ATOM0 channel 2 shared interrupt. See bit 0.
1	ATOM0_CH1_IRQ	ATOM0 channel 1 shared interrupt. See bit 0.
0	ATOM0_CH0_IRQ	ATOM0 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources CCU0TCx_IRQ or CCU1TCx_IRQ of ATOM instance 0 channel x.

38.24.5.11 ICM_IRQG_10

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00628_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ATOM7_CH7_IRQ	ATOM7_CH6_IRQ	ATOM7_CH5_IRQ	ATOM7_CH4_IRQ	ATOM7_CH3_IRQ	ATOM7_CH2_IRQ	ATOM7_CH1_IRQ	ATOM7_CH0_IRQ	ATOM6_CH7_IRQ	ATOM6_CH6_IRQ	ATOM6_CH5_IRQ	ATOM6_CH4_IRQ	ATOM6_CH3_IRQ	ATOM6_CH2_IRQ	ATOM6_CH1_IRQ	ATOM6_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATOM5_CH7_IRQ	ATOM5_CH6_IRQ	ATOM5_CH5_IRQ	ATOM5_CH4_IRQ	ATOM5_CH3_IRQ	ATOM5_CH2_IRQ	ATOM5_CH1_IRQ	ATOM5_CH0_IRQ	ATOM4_CH7_IRQ	ATOM4_CH6_IRQ	ATOM4_CH5_IRQ	ATOM4_CH4_IRQ	ATOM4_CH3_IRQ	ATOM4_CH2_IRQ	ATOM4_CH1_IRQ	ATOM4_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.472 ICM_IRQG_10 Register Contents (1/2)

Bit Position	Bit Name	Function
31	ATOM7_CH7_IRQ	ATOM7 channel 7 shared interrupt. See bit 0.
30	ATOM7_CH6_IRQ	ATOM7 channel 6 shared interrupt. See bit 0.
29	ATOM7_CH5_IRQ	ATOM7 channel 5 shared interrupt. See bit 0.
28	ATOM7_CH4_IRQ	ATOM7 channel 4 shared interrupt. See bit 0.
27	ATOM7_CH3_IRQ	ATOM7 channel 3 shared interrupt. See bit 0.
26	ATOM7_CH2_IRQ	ATOM7 channel 2 shared interrupt. See bit 0.
25	ATOM7_CH1_IRQ	ATOM7 channel 1 shared interrupt. See bit 0.
24	ATOM7_CH0_IRQ	ATOM7 channel 0 shared interrupt. See bit 0.
23	ATOM6_CH7_IRQ	ATOM6 channel 7 shared interrupt. See bit 0.
22	ATOM6_CH6_IRQ	ATOM6 channel 6 shared interrupt. See bit 0.
21	ATOM6_CH5_IRQ	ATOM6 channel 5 shared interrupt. See bit 0.
20	ATOM6_CH4_IRQ	ATOM6 channel 4 shared interrupt. See bit 0.
19	ATOM6_CH3_IRQ	ATOM6 channel 3 shared interrupt. See bit 0.
18	ATOM6_CH2_IRQ	ATOM6 channel 2 shared interrupt. See bit 0.
17	ATOM6_CH1_IRQ	ATOM6 channel 1 shared interrupt. See bit 0.
16	ATOM6_CH0_IRQ	ATOM6 channel 0 shared interrupt. See bit 0.
15	ATOM5_CH7_IRQ	ATOM5 channel 7 shared interrupt. See bit 0.
14	ATOM5_CH6_IRQ	ATOM5 channel 6 shared interrupt. See bit 0.
13	ATOM5_CH5_IRQ	ATOM5 channel 5 shared interrupt. See bit 0.
12	ATOM5_CH4_IRQ	ATOM5 channel 4 shared interrupt. See bit 0.
11	ATOM5_CH3_IRQ	ATOM5 channel 3 shared interrupt. See bit 0.
10	ATOM5_CH2_IRQ	ATOM5 channel 2 shared interrupt. See bit 0.
9	ATOM5_CH1_IRQ	ATOM5 channel 1 shared interrupt. See bit 0.
8	ATOM5_CH0_IRQ	ATOM5 channel 0 shared interrupt. See bit 0.
7	ATOM4_CH7_IRQ	ATOM4 channel 7 shared interrupt. See bit 0.
6	ATOM4_CH6_IRQ	ATOM4 channel 6 shared interrupt. See bit 0.
5	ATOM4_CH5_IRQ	ATOM4 channel 5 shared interrupt. See bit 0.
4	ATOM4_CH4_IRQ	ATOM4 channel 4 shared interrupt. See bit 0.

Table 38.472 ICM_IRQG_10 Register Contents (2/2)

Bit Position	Bit Name	Function
3	ATOM4_CH3_IRQ	ATOM4 channel 3 shared interrupt. See bit 0.
2	ATOM4_CH2_IRQ	ATOM4 channel 2 shared interrupt. See bit 0.
1	ATOM4_CH1_IRQ	ATOM4 channel 1 shared interrupt. See bit 0.
0	ATOM4_CH0_IRQ	ATOM4 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources CCU0TCx_IRQ or CCU1TCx_IRQ of ATOM instance 4 channel x.

38.24.5.12 ICM_IRQG_11

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0062C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ATOM11_CH7_IRQ	ATOM11_CH6_IRQ	ATOM11_CH5_IRQ	ATOM11_CH4_IRQ	ATOM11_CH3_IRQ	ATOM11_CH2_IRQ	ATOM11_CH1_IRQ	ATOM11_CH0_IRQ	ATOM10_CH7_IRQ	ATOM10_CH6_IRQ	ATOM10_CH5_IRQ	ATOM10_CH4_IRQ	ATOM10_CH3_IRQ	ATOM10_CH2_IRQ	ATOM10_CH1_IRQ	ATOM10_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATOM9_CH7_IRQ	ATOM9_CH6_IRQ	ATOM9_CH5_IRQ	ATOM9_CH4_IRQ	ATOM9_CH3_IRQ	ATOM9_CH2_IRQ	ATOM9_CH1_IRQ	ATOM9_CH0_IRQ	ATOM8_CH7_IRQ	ATOM8_CH6_IRQ	ATOM8_CH5_IRQ	ATOM8_CH4_IRQ	ATOM8_CH3_IRQ	ATOM8_CH2_IRQ	ATOM8_CH1_IRQ	ATOM8_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.473 ICM_IRQG_11 Register Contents (1/2)

Bit Position	Bit Name	Function
31	ATOM11_CH7_IRQ	ATOM11 channel 7 shared interrupt. See bit 0.
30	ATOM11_CH6_IRQ	ATOM11 channel 6 shared interrupt. See bit 0.
29	ATOM11_CH5_IRQ	ATOM11 channel 5 shared interrupt. See bit 0.
28	ATOM11_CH4_IRQ	ATOM11 channel 4 shared interrupt. See bit 0.
27	ATOM11_CH3_IRQ	ATOM11 channel 3 shared interrupt. See bit 0.
26	ATOM11_CH2_IRQ	ATOM11 channel 2 shared interrupt. See bit 0.
25	ATOM11_CH1_IRQ	ATOM11 channel 1 shared interrupt. See bit 0.
24	ATOM11_CH0_IRQ	ATOM11 channel 0 shared interrupt. See bit 0.
23	ATOM10_CH7_IRQ	ATOM10 channel 7 shared interrupt. See bit 0.
22	ATOM10_CH6_IRQ	ATOM10 channel 6 shared interrupt. See bit 0.
21	ATOM10_CH5_IRQ	ATOM10 channel 5 shared interrupt. See bit 0.
20	ATOM10_CH4_IRQ	ATOM10 channel 4 shared interrupt. See bit 0.
19	ATOM10_CH3_IRQ	ATOM10 channel 3 shared interrupt. See bit 0.
18	ATOM10_CH2_IRQ	ATOM10 channel 2 shared interrupt. See bit 0.
17	ATOM10_CH1_IRQ	ATOM10 channel 1 shared interrupt. See bit 0.
16	ATOM10_CH0_IRQ	ATOM10 channel 0 shared interrupt. See bit 0.
15	ATOM9_CH7_IRQ	ATOM9 channel 7 shared interrupt. See bit 0.
14	ATOM9_CH6_IRQ	ATOM9 channel 6 shared interrupt. See bit 0.
13	ATOM9_CH5_IRQ	ATOM9 channel 5 shared interrupt. See bit 0.
12	ATOM9_CH4_IRQ	ATOM9 channel 4 shared interrupt. See bit 0.
11	ATOM9_CH3_IRQ	ATOM9 channel 3 shared interrupt. See bit 0.
10	ATOM9_CH2_IRQ	ATOM9 channel 2 shared interrupt. See bit 0.
9	ATOM9_CH1_IRQ	ATOM9 channel 1 shared interrupt. See bit 0.
8	ATOM9_CH0_IRQ	ATOM9 channel 0 shared interrupt. See bit 0.
7	ATOM8_CH7_IRQ	ATOM8 channel 7 shared interrupt. See bit 0.
6	ATOM8_CH6_IRQ	ATOM8 channel 6 shared interrupt. See bit 0.
5	ATOM8_CH5_IRQ	ATOM8 channel 5 shared interrupt. See bit 0.
4	ATOM8_CH4_IRQ	ATOM8 channel 4 shared interrupt. See bit 0.

Table 38.473 ICM_IRQG_11 Register Contents (2/2)

Bit Position	Bit Name	Function
3	ATOM8_CH3_IRQ	ATOM8 channel 3 shared interrupt. See bit 0.
2	ATOM8_CH2_IRQ	ATOM8 channel 2 shared interrupt. See bit 0.
1	ATOM8_CH1_IRQ	ATOM8 channel 1 shared interrupt. See bit 0.
0	ATOM8_CH0_IRQ	ATOM8 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources CCU0TCx_IRQ or CCU1TCx_IRQ of ATOM instance 8 channel x.

38.24.5.13 ICM_IRQG_MEI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00630_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DPLL_EIRQ	CMP_EIRQ	SPE3_EIRQ	SPE2_EIRQ	SPE1_EIRQ	SPE0_EIRQ	MCS7_EIRQ	MCS6_EIRQ	MCS5_EIRQ	MCS4_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCS3_EIRQ	MCS2_EIRQ	MCS1_EIRQ	MCS0_EIRQ	TIM7_EIRQ	TIM6_EIRQ	TIM5_EIRQ	TIM4_EIRQ	TIM3_EIRQ	TIM2_EIRQ	TIM1_EIRQ	TIM0_EIRQ	FIFO1_EIRQ	FIFO0_EIRQ	BRC_EIRQ	GTM_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.474 ICM_IRQG_MEI Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	DPLL_EIRQ	DPLL error interrupt. See bit 0.
24	CMP_EIRQ	CMP error interrupt. See bit 0.
23	SPE3_EIRQ	SPE3 error interrupt. See bit 0.
22	SPE2_EIRQ	SPE2 error interrupt. See bit 0.
21	SPE1_EIRQ	SPE1 error interrupt. See bit 0.
20	SPE0_EIRQ	SPE0 error interrupt. See bit 0.
19	MCS7_EIRQ	MCS7 error interrupt. See bit 0.
18	MCS6_EIRQ	MCS6 error interrupt. See bit 0.
17	MCS5_EIRQ	MCS5 error interrupt. See bit 0.
16	MCS4_EIRQ	MCS4 error interrupt. See bit 0.
15	MCS3_EIRQ	MCS3 error interrupt. See bit 0.
14	MCS2_EIRQ	MCS2 error interrupt. See bit 0.
13	MCS1_EIRQ	MCS1 error interrupt. See bit 0.
12	MCS0_EIRQ	MCS0 error interrupt. See bit 0.
11	TIM7_EIRQ	TIM7 error interrupt. See bit 0.
10	TIM6_EIRQ	TIM6 error interrupt. See bit 0.
9	TIM5_EIRQ	TIM5 error interrupt. See bit 0.
8	TIM4_EIRQ	TIM4 error interrupt. See bit 0.
7	TIM3_EIRQ	TIM3 error interrupt. See bit 0.
6	TIM2_EIRQ	TIM2 error interrupt. See bit 0.
5	TIM1_EIRQ	TIM1 error interrupt. See bit 0.
4	TIM0_EIRQ	TIM0 error interrupt. See bit 0.
3	FIFO1_EIRQ	FIFO1 error interrupt. See bit 0.
2	FIFO0_EIRQ	FIFO0 error interrupt. See bit 0.
1	BRC_EIRQ	BRC error interrupt. See bit 0.

Table 38.474 ICM_IRQG_MEI Register Contents (2/2)

Bit Position	Bit Name	Function
0	GTM_EIRQ	AEI Error interrupt request 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the interrupt sources AEI_TO_XPT_EIRQ, AEI_USP_ADDR_EIRQ, AEI_IM_ADDR_EIRQ, AEI_USP_BE_EIRQ, AEIM_USP_ADDR_EIRQ, AEIM_IM_ADDR_EIRQ, AEIM_USP_BE_EIRQ, CLK_EN_ERR or CLK_PER_ERR.

38.24.5.14 ICM_IRQG_CEI0

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00634_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	FIFO2_CH7_EIRQ	FIFO2_CH6_EIRQ	FIFO2_CH5_EIRQ	FIFO2_CH4_EIRQ	FIFO2_CH3_EIRQ	FIFO2_CH2_EIRQ	FIFO2_CH1_EIRQ	FIFO2_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFO1_CH7_EIRQ	FIFO1_CH6_EIRQ	FIFO1_CH5_EIRQ	FIFO1_CH4_EIRQ	FIFO1_CH3_EIRQ	FIFO1_CH2_EIRQ	FIFO1_CH1_EIRQ	FIFO1_CH0_EIRQ	FIFO0_CH7_EIRQ	FIFO0_CH6_EIRQ	FIFO0_CH5_EIRQ	FIFO0_CH4_EIRQ	FIFO0_CH3_EIRQ	FIFO0_CH2_EIRQ	FIFO0_CH1_EIRQ	FIFO0_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.475 ICM_IRQG_CEI0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	FIFO2_CH7_EIRQ	FIFO2 channel 7 error interrupt. See bit 0.
22	FIFO2_CH6_EIRQ	FIFO2 channel 6 error interrupt. See bit 0.
21	FIFO2_CH5_EIRQ	FIFO2 channel 5 error interrupt. See bit 0.
20	FIFO2_CH4_EIRQ	FIFO2 channel 4 error interrupt. See bit 0.
19	FIFO2_CH3_EIRQ	FIFO2 channel 3 error interrupt. See bit 0.
18	FIFO2_CH2_EIRQ	FIFO2 channel 2 error interrupt. See bit 0.
17	FIFO2_CH1_EIRQ	FIFO2 channel 1 error interrupt. See bit 0.
16	FIFO2_CH1_EIRQ	FIFO2 channel 0 error interrupt. See bit 0.
15	FIFO1_CH7_EIRQ	FIFO1 channel 7 error interrupt. See bit 0.
14	FIFO1_CH6_EIRQ	FIFO1 channel 6 error interrupt. See bit 0.
13	FIFO1_CH5_EIRQ	FIFO1 channel 5 error interrupt. See bit 0.
12	FIFO1_CH4_EIRQ	FIFO1 channel 4 error interrupt. See bit 0.
11	FIFO1_CH3_EIRQ	FIFO1 channel 3 error interrupt. See bit 0.
10	FIFO1_CH2_EIRQ	FIFO1 channel 2 error interrupt. See bit 0.
9	FIFO1_CH1_EIRQ	FIFO1 channel 1 error interrupt. See bit 0.
8	FIFO1_CH0_EIRQ	FIFO1 channel 0 error interrupt. See bit 0.
7	FIFO0_CH7_EIRQ	FIFO0 channel 7 error interrupt. See bit 0.
6	FIFO0_CH6_EIRQ	FIFO0 channel 6 error interrupt. See bit 0.
5	FIFO0_CH5_EIRQ	FIFO0 channel 5 error interrupt. See bit 0.
4	FIFO0_CH4_EIRQ	FIFO0 channel 4 error interrupt. See bit 0.
3	FIFO0_CH3_EIRQ	FIFO0 channel 3 error interrupt. See bit 0.
2	FIFO0_CH2_EIRQ	FIFO0 channel 2 error interrupt. See bit 0.
1	FIFO0_CH1_EIRQ	FIFO0 channel 1 error interrupt. See bit 0.

Table 38.475 ICM_IRQG_CEI0 Register Contents (2/2)

Bit Position	Bit Name	Function
0	FIFO0_CH0_EIRQ	FIFO0 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.15 ICM_IRQG_CEI1

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00638_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIM3_CH7_EIRQ	TIM3_CH6_EIRQ	TIM3_CH5_EIRQ	TIM3_CH4_EIRQ	TIM3_CH3_EIRQ	TIM3_CH2_EIRQ	TIM3_CH1_EIRQ	TIM3_CH0_EIRQ	TIM2_CH7_EIRQ	TIM2_CH6_EIRQ	TIM2_CH5_EIRQ	TIM2_CH4_EIRQ	TIM2_CH3_EIRQ	TIM2_CH2_EIRQ	TIM2_CH1_EIRQ	TIM2_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIM1_CH7_EIRQ	TIM1_CH6_EIRQ	TIM1_CH5_EIRQ	TIM1_CH4_EIRQ	TIM1_CH3_EIRQ	TIM1_CH2_EIRQ	TIM1_CH1_EIRQ	TIM1_CH0_EIRQ	TIM0_CH7_EIRQ	TIM0_CH6_EIRQ	TIM0_CH5_EIRQ	TIM0_CH4_EIRQ	TIM0_CH3_EIRQ	TIM0_CH2_EIRQ	TIM0_CH1_EIRQ	TIM0_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.476 ICM_IRQG_CEI1 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TIM3_CH7_EIRQ	TIM3 channel 7 error interrupt. See bit 0.
30	TIM3_CH6_EIRQ	TIM3 channel 6 error interrupt. See bit 0.
29	TIM3_CH5_EIRQ	TIM3 channel 5 error interrupt. See bit 0.
28	TIM3_CH4_EIRQ	TIM3 channel 4 error interrupt. See bit 0.
27	TIM3_CH3_EIRQ	TIM3 channel 3 error interrupt. See bit 0.
26	TIM3_CH2_EIRQ	TIM3 channel 2 error interrupt. See bit 0.
25	TIM3_CH1_EIRQ	TIM3 channel 1 error interrupt. See bit 0.
24	TIM3_CH0_EIRQ	TIM3 channel 0 error interrupt. See bit 0.
23	TIM2_CH7_EIRQ	TIM2 channel 7 error interrupt. See bit 0.
22	TIM2_CH6_EIRQ	TIM2 channel 6 error interrupt. See bit 0.
21	TIM2_CH5_EIRQ	TIM2 channel 5 error interrupt. See bit 0.
20	TIM2_CH4_EIRQ	TIM2 channel 4 error interrupt. See bit 0.
19	TIM2_CH3_EIRQ	TIM2 channel 3 error interrupt. See bit 0.
18	TIM2_CH2_EIRQ	TIM2 channel 2 error interrupt. See bit 0.
17	TIM2_CH1_EIRQ	TIM2 channel 1 error interrupt. See bit 0.
16	TIM2_CH0_EIRQ	TIM2 channel 0 error interrupt. See bit 0.
15	TIM1_CH7_EIRQ	TIM1 channel 7 error interrupt. See bit 0.
14	TIM1_CH6_EIRQ	TIM1 channel 6 error interrupt. See bit 0.
13	TIM1_CH5_EIRQ	TIM1 channel 5 error interrupt. See bit 0.
12	TIM1_CH4_EIRQ	TIM1 channel 4 error interrupt. See bit 0.
11	TIM1_CH3_EIRQ	TIM1 channel 3 error interrupt. See bit 0.
10	TIM1_CH2_EIRQ	TIM1 channel 2 error interrupt. See bit 0.
9	TIM1_CH1_EIRQ	TIM1 channel 1 error interrupt. See bit 0.
8	TIM1_CH0_EIRQ	TIM1 channel 0 error interrupt. See bit 0.
7	TIM0_CH7_EIRQ	TIM0 channel 7 error interrupt. See bit 0.
6	TIM0_CH6_EIRQ	TIM0 channel 6 error interrupt. See bit 0.
5	TIM0_CH5_EIRQ	TIM0 channel 5 error interrupt. See bit 0.
4	TIM0_CH4_EIRQ	TIM0 channel 4 error interrupt. See bit 0.

Table 38.476 ICM_IRQG_CEI1 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TIM0_CH3_EIRQ	TIM0 channel 3 error interrupt. See bit 0.
2	TIM0_CH2_EIRQ	TIM0 channel 2 error interrupt. See bit 0.
1	TIM0_CH1_EIRQ	TIM0 channel 1 error interrupt. See bit 0.
0	TIM0_CH0_EIRQ	TIM0 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.16 ICM_IRQG_CEI2

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $0063C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TIM7_CH7_EIRQ	TIM7_CH6_EIRQ	TIM7_CH5_EIRQ	TIM7_CH4_EIRQ	TIM7_CH3_EIRQ	TIM7_CH2_EIRQ	TIM7_CH1_EIRQ	TIM7_CH0_EIRQ	TIM6_CH7_EIRQ	TIM6_CH6_EIRQ	TIM6_CH5_EIRQ	TIM6_CH4_EIRQ	TIM6_CH3_EIRQ	TIM6_CH2_EIRQ	TIM6_CH1_EIRQ	TIM6_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TIM5_CH7_EIRQ	TIM5_CH6_EIRQ	TIM5_CH5_EIRQ	TIM5_CH4_EIRQ	TIM5_CH3_EIRQ	TIM5_CH2_EIRQ	TIM5_CH1_EIRQ	TIM5_CH0_EIRQ	TIM4_CH7_EIRQ	TIM4_CH6_EIRQ	TIM4_CH5_EIRQ	TIM4_CH4_EIRQ	TIM4_CH3_EIRQ	TIM4_CH2_EIRQ	TIM4_CH1_EIRQ	TIM4_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.477 ICM_IRQG_CEI2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	TIM7_CH7_EIRQ	TIM7 channel 7 error interrupt. See bit 0.
30	TIM7_CH6_EIRQ	TIM7 channel 6 error interrupt. See bit 0.
29	TIM7_CH5_EIRQ	TIM7 channel 5 error interrupt. See bit 0.
28	TIM7_CH4_EIRQ	TIM7 channel 4 error interrupt. See bit 0.
27	TIM7_CH3_EIRQ	TIM7 channel 3 error interrupt. See bit 0.
26	TIM7_CH2_EIRQ	TIM7 channel 2 error interrupt. See bit 0.
25	TIM7_CH1_EIRQ	TIM7 channel 1 error interrupt. See bit 0.
24	TIM7_CH0_EIRQ	TIM7 channel 0 error interrupt. See bit 0.
23	TIM6_CH7_EIRQ	TIM6 channel 7 error interrupt. See bit 0.
22	TIM6_CH6_EIRQ	TIM6 channel 6 error interrupt. See bit 0.
21	TIM6_CH5_EIRQ	TIM6 channel 5 error interrupt. See bit 0.
20	TIM6_CH4_EIRQ	TIM6 channel 4 error interrupt. See bit 0.
19	TIM6_CH3_EIRQ	TIM6 channel 3 error interrupt. See bit 0.
18	TIM6_CH2_EIRQ	TIM6 channel 2 error interrupt. See bit 0.
17	TIM6_CH1_EIRQ	TIM6 channel 1 error interrupt. See bit 0.
16	TIM6_CH0_EIRQ	TIM6 channel 0 error interrupt. See bit 0.
15	TIM5_CH7_EIRQ	TIM5 channel 7 error interrupt. See bit 0.
14	TIM5_CH6_EIRQ	TIM5 channel 6 error interrupt. See bit 0.
13	TIM5_CH5_EIRQ	TIM5 channel 5 error interrupt. See bit 0.
12	TIM5_CH4_EIRQ	TIM5 channel 4 error interrupt. See bit 0.
11	TIM5_CH3_EIRQ	TIM5 channel 3 error interrupt. See bit 0.
10	TIM5_CH2_EIRQ	TIM5 channel 2 error interrupt. See bit 0.
9	TIM5_CH1_EIRQ	TIM5 channel 1 error interrupt. See bit 0.
8	TIM5_CH0_EIRQ	TIM5 channel 0 error interrupt. See bit 0.
7	TIM4_CH7_EIRQ	TIM4 channel 7 error interrupt. See bit 0.
6	TIM4_CH6_EIRQ	TIM4 channel 6 error interrupt. See bit 0.
5	TIM4_CH5_EIRQ	TIM4 channel 5 error interrupt. See bit 0.
4	TIM4_CH4_EIRQ	TIM4 channel 4 error interrupt. See bit 0.

Table 38.477 ICM_IRQG_CEI2 Register Contents (2/2)

Bit Position	Bit Name	Function
3	TIM4_CH3_EIRQ	TIM4 channel 3 error interrupt. See bit 0.
2	TIM4_CH2_EIRQ	TIM4 channel 2 error interrupt. See bit 0.
1	TIM4_CH1_EIRQ	TIM4 channel 1 error interrupt. See bit 0.
0	TIM4_CH0_EIRQ	TIM4 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.17 ICM_IRQG_CEI3

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00640_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCS3_CH7_EIRQ	MCS3_CH6_EIRQ	MCS3_CH5_EIRQ	MCS3_CH4_EIRQ	MCS3_CH3_EIRQ	MCS3_CH2_EIRQ	MCS3_CH1_EIRQ	MCS3_CH0_EIRQ	MCS2_CH7_EIRQ	MCS2_CH6_EIRQ	MCS2_CH5_EIRQ	MCS2_CH4_EIRQ	MCS2_CH3_EIRQ	MCS2_CH2_EIRQ	MCS2_CH1_EIRQ	MCS2_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCS1_CH7_EIRQ	MCS1_CH6_EIRQ	MCS1_CH5_EIRQ	MCS1_CH4_EIRQ	MCS1_CH3_EIRQ	MCS1_CH2_EIRQ	MCS1_CH1_EIRQ	MCS1_CH0_EIRQ	MCS0_CH7_EIRQ	MCS0_CH6_EIRQ	MCS0_CH5_EIRQ	MCS0_CH4_EIRQ	MCS0_CH3_EIRQ	MCS0_CH2_EIRQ	MCS0_CH1_EIRQ	MCS0_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.478 ICM_IRQG_CEI3 Register Contents (1/2)

Bit Position	Bit Name	Function
31	MCS3_CH7_EIRQ	MCS3 channel 7 error interrupt. See bit 0.
30	MCS3_CH6_EIRQ	MCS3 channel 6 error interrupt. See bit 0.
29	MCS3_CH5_EIRQ	MCS3 channel 5 error interrupt. See bit 0.
28	MCS3_CH4_EIRQ	MCS3 channel 4 error interrupt. See bit 0.
27	MCS3_CH3_EIRQ	MCS3 channel 3 error interrupt. See bit 0.
26	MCS3_CH2_EIRQ	MCS3 channel 2 error interrupt. See bit 0.
25	MCS3_CH1_EIRQ	MCS3 channel 1 error interrupt. See bit 0.
24	MCS3_CH0_EIRQ	MCS3 channel 0 error interrupt. See bit 0.
23	MCS2_CH7_EIRQ	MCS2 channel 7 error interrupt. See bit 0.
22	MCS2_CH6_EIRQ	MCS2 channel 6 error interrupt. See bit 0.
21	MCS2_CH5_EIRQ	MCS2 channel 5 error interrupt. See bit 0.
20	MCS2_CH4_EIRQ	MCS2 channel 4 error interrupt. See bit 0.
19	MCS2_CH3_EIRQ	MCS2 channel 3 error interrupt. See bit 0.
18	MCS2_CH2_EIRQ	MCS2 channel 2 error interrupt. See bit 0.
17	MCS2_CH1_EIRQ	MCS2 channel 1 error interrupt. See bit 0.
16	MCS2_CH0_EIRQ	MCS2 channel 0 error interrupt. See bit 0.
15	MCS1_CH7_EIRQ	MCS1 channel 7 error interrupt. See bit 0.
14	MCS1_CH6_EIRQ	MCS1 channel 6 error interrupt. See bit 0.
13	MCS1_CH5_EIRQ	MCS1 channel 5 error interrupt. See bit 0.
12	MCS1_CH4_EIRQ	MCS1 channel 4 error interrupt. See bit 0.
11	MCS1_CH3_EIRQ	MCS1 channel 3 error interrupt. See bit 0.
10	MCS1_CH2_EIRQ	MCS1 channel 2 error interrupt. See bit 0.
9	MCS1_CH1_EIRQ	MCS1 channel 1 error interrupt. See bit 0.
8	MCS1_CH0_EIRQ	MCS1 channel 0 error interrupt. See bit 0.
7	MCS0_CH7_EIRQ	MCS0 channel 7 error interrupt. See bit 0.
6	MCS0_CH6_EIRQ	MCS0 channel 6 error interrupt. See bit 0.
5	MCS0_CH5_EIRQ	MCS0 channel 5 error interrupt. See bit 0.
4	MCS0_CH4_EIRQ	MCS0 channel 4 error interrupt. See bit 0.

Table 38.478 ICM_IRQG_CEI3 Register Contents (2/2)

Bit Position	Bit Name	Function
3	MCS0_CH3_EIRQ	MCS0 channel 3 error interrupt. See bit 0.
2	MCS0_CH2_EIRQ	MCS0 channel 2 error interrupt. See bit 0.
1	MCS0_CH1_EIRQ	MCS0 channel 1 error interrupt. See bit 0.
0	MCS0_CH0_EIRQ	MCS0 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.18 ICM_IRQG_CEI4

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00644_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCS7_CH7_EIRQ	MCS7_CH6_EIRQ	MCS7_CH5_EIRQ	MCS7_CH4_EIRQ	MCS7_CH3_EIRQ	MCS7_CH2_EIRQ	MCS7_CH1_EIRQ	MCS7_CH0_EIRQ	MCS6_CH7_EIRQ	MCS6_CH6_EIRQ	MCS6_CH5_EIRQ	MCS6_CH4_EIRQ	MCS6_CH3_EIRQ	MCS6_CH2_EIRQ	MCS6_CH1_EIRQ	MCS6_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCS5_CH7_EIRQ	MCS5_CH6_EIRQ	MCS5_CH5_EIRQ	MCS5_CH4_EIRQ	MCS5_CH3_EIRQ	MCS5_CH2_EIRQ	MCS5_CH1_EIRQ	MCS5_CH0_EIRQ	MCS4_CH7_EIRQ	MCS4_CH6_EIRQ	MCS4_CH5_EIRQ	MCS4_CH4_EIRQ	MCS4_CH3_EIRQ	MCS4_CH2_EIRQ	MCS4_CH1_EIRQ	MCS4_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.479 ICM_IRQG_CEI4 Register Contents (1/2)

Bit Position	Bit Name	Function
31	MCS7_CH7_EIRQ	MCS7 channel 7 error interrupt. See bit 0.
30	MCS7_CH6_EIRQ	MCS7 channel 6 error interrupt. See bit 0.
29	MCS7_CH5_EIRQ	MCS7 channel 5 error interrupt. See bit 0.
28	MCS7_CH4_EIRQ	MCS7 channel 4 error interrupt. See bit 0.
27	MCS7_CH3_EIRQ	MCS7 channel 3 error interrupt. See bit 0.
26	MCS7_CH2_EIRQ	MCS7 channel 2 error interrupt. See bit 0.
25	MCS7_CH1_EIRQ	MCS7 channel 1 error interrupt. See bit 0.
24	MCS7_CH0_EIRQ	MCS7 channel 0 error interrupt. See bit 0.
23	MCS6_CH7_EIRQ	MCS6 channel 7 error interrupt. See bit 0.
22	MCS6_CH6_EIRQ	MCS6 channel 6 error interrupt. See bit 0.
21	MCS6_CH5_EIRQ	MCS6 channel 5 error interrupt. See bit 0.
20	MCS6_CH4_EIRQ	MCS6 channel 4 error interrupt. See bit 0.
19	MCS6_CH3_EIRQ	MCS6 channel 3 error interrupt. See bit 0.
18	MCS6_CH2_EIRQ	MCS6 channel 2 error interrupt. See bit 0.
17	MCS6_CH1_EIRQ	MCS6 channel 1 error interrupt. See bit 0.
16	MCS6_CH0_EIRQ	MCS6 channel 0 error interrupt. See bit 0.
15	MCS5_CH7_EIRQ	MCS5 channel 7 error interrupt. See bit 0.
14	MCS5_CH6_EIRQ	MCS5 channel 6 error interrupt. See bit 0.
13	MCS5_CH5_EIRQ	MCS5 channel 5 error interrupt. See bit 0.
12	MCS5_CH4_EIRQ	MCS5 channel 4 error interrupt. See bit 0.
11	MCS5_CH3_EIRQ	MCS5 channel 3 error interrupt. See bit 0.
10	MCS5_CH2_EIRQ	MCS5 channel 2 error interrupt. See bit 0.
9	MCS5_CH1_EIRQ	MCS5 channel 1 error interrupt. See bit 0.
8	MCS5_CH0_EIRQ	MCS5 channel 0 error interrupt. See bit 0.
7	MCS4_CH7_EIRQ	MCS4 channel 7 error interrupt. See bit 0.
6	MCS4_CH6_EIRQ	MCS4 channel 6 error interrupt. See bit 0.
5	MCS4_CH5_EIRQ	MCS4 channel 5 error interrupt. See bit 0.
4	MCS4_CH4_EIRQ	MCS4 channel 4 error interrupt. See bit 0.

Table 38.479 ICM_IRQG_CEI4 Register Contents (2/2)

Bit Position	Bit Name	Function
3	MCS4_CH3_EIRQ	MCS4 channel 3 error interrupt. See bit 0.
2	MCS4_CH2_EIRQ	MCS4 channel 2 error interrupt. See bit 0.
1	MCS4_CH1_EIRQ	MCS4 channel 1 error interrupt. See bit 0.
0	MCS4_CH0_EIRQ	MCS4 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.19 ICM_IRQG_MCS[i]_CI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + [i] ×4 + 00720_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MCS_CH7_IRQ	MCS_CH6_IRQ	MCS_CH5_IRQ	MCS_CH4_IRQ	MCS_CH3_IRQ	MCS_CH2_IRQ	MCS_CH1_IRQ	MCS_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.480 ICM_IRQG_MCS[i]_CI Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	MCS_CH7_IRQ	MCS[i] channel 7 interrupt. See bit 0. (i = 0 to 3)
6	MCS_CH6_IRQ	MCS[i] channel 6 interrupt. See bit 0. (i = 0 to 3)
5	MCS_CH5_IRQ	MCS[i] channel 5 interrupt. See bit 0. (i = 0 to 3)
4	MCS_CH4_IRQ	MCS[i] channel 4 interrupt. See bit 0. (i = 0 to 3)
3	MCS_CH3_IRQ	MCS[i] channel 3 interrupt. See bit 0. (i = 0 to 3)
2	MCS_CH2_IRQ	MCS[i] channel 2 interrupt. See bit 0. (i = 0 to 3)
1	MCS_CH1_IRQ	MCS[i] channel 1 interrupt. See bit 0. (i = 0 to 3)
0	MCS_CH0_IRQ	MCS[i] channel 0 interrupt. (i = 0 to 3) 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module.

38.24.5.20 ICM_IRQG_MCS[i]_CEI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + [i] ×4 + 00664_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MCS_CH7_EIRQ	MCS_CH6_EIRQ	MCS_CH5_EIRQ	MCS_CH4_EIRQ	MCS_CH3_EIRQ	MCS_CH2_EIRQ	MCS_CH1_EIRQ	MCS_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.481 ICM_IRQG_MCS[i]_CI Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	MCS_CH7_EIRQ	MCS channel 7 error interrupt. See bit 0.
6	MCS_CH6_EIRQ	MCS channel 6 error interrupt. See bit 0.
5	MCS_CH5_EIRQ	MCS channel 5 error interrupt. See bit 0.
4	MCS_CH4_EIRQ	MCS channel 4 error interrupt. See bit 0.
3	MCS_CH3_EIRQ	MCS channel 3 error interrupt. See bit 0.
2	MCS_CH2_EIRQ	MCS channel 2 error interrupt. See bit 0.
1	MCS_CH1_EIRQ	MCS channel 1 error interrupt. See bit 0.
0	MCS_CH0_EIRQ	MCS channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.21 ICM_IRQG_SPE_CI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00770_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SPE_CH5_IRQ	SPE_CH4_IRQ	SPE_CH3_IRQ	SPE_CH2_IRQ	SPE_CH1_IRQ	SPE_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.482 ICM_IRQG_SPE_CI Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5	SPE_CH5_IRQ	SPE channel 5 interrupt. See bit 0.
4	SPE_CH4_IRQ	SPE channel 4 interrupt. See bit 0.
3	SPE_CH3_IRQ	SPE channel 3 interrupt. See bit 0.
2	SPE_CH2_IRQ	SPE channel 2 interrupt. See bit 0.
1	SPE_CH1_IRQ	SPE channel 1 interrupt. See bit 0.
0	SPE_CH0_IRQ	SPE channel 0 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module.

38.24.5.22 ICM_IRQG_SPE_CEI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $006B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SPE_CH5_EIRQ	SPE_CH4_EIRQ	SPE_CH3_EIRQ	SPE_CH2_EIRQ	SPE_CH1_EIRQ	SPE_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.483 ICM_IRQG_SPE_CEI Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5	SPE_CH5_EIRQ	SPE channel 5 error interrupt. See bit 0.
4	SPE_CH4_EIRQ	SPE channel 4 error interrupt. See bit 0.
3	SPE_CH3_EIRQ	SPE channel 3 error interrupt. See bit 0.
2	SPE_CH2_EIRQ	SPE channel 2 error interrupt. See bit 0.
1	SPE_CH1_EIRQ	SPE channel 1 error interrupt. See bit 0.
0	SPE_CH0_EIRQ	SPE channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.23 ICM_IRQG_PSM_0_CI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00760_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSM_M2_CH7_IRQ	PSM_M2_CH6_IRQ	PSM_M2_CH5_IRQ	PSM_M2_CH4_IRQ	PSM_M2_CH3_IRQ	PSM_M2_CH2_IRQ	PSM_M2_CH1_IRQ	PSM_M2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSM_M1_CH7_IRQ	PSM_M1_CH6_IRQ	PSM_M1_CH5_IRQ	PSM_M1_CH4_IRQ	PSM_M1_CH3_IRQ	PSM_M1_CH2_IRQ	PSM_M1_CH1_IRQ	PSM_M1_CH0_IRQ	PSM_M0_CH7_IRQ	PSM_M0_CH6_IRQ	PSM_M0_CH5_IRQ	PSM_M0_CH4_IRQ	PSM_M0_CH3_IRQ	PSM_M0_CH2_IRQ	PSM_M0_CH1_IRQ	PSM_M0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.484 ICM_IRQG_PSM_0_CI Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	PSM_M2_CH7_IRQ	PSM2 channel 7 shared interrupt. See bit 0.
22	PSM_M2_CH6_IRQ	PSM2 channel 6 shared interrupt. See bit 0.
21	PSM_M2_CH5_IRQ	PSM2 channel 5 shared interrupt. See bit 0.
20	PSM_M2_CH4_IRQ	PSM2 channel 4 shared interrupt. See bit 0.
19	PSM_M2_CH3_IRQ	PSM2 channel 3 shared interrupt. See bit 0.
18	PSM_M2_CH2_IRQ	PSM2 channel 2 shared interrupt. See bit 0.
17	PSM_M2_CH1_IRQ	PSM2 channel 1 shared interrupt. See bit 0.
16	PSM_M2_CH0_IRQ	PSM2 channel 0 shared interrupt. See bit 0.
15	PSM_M1_CH7_IRQ	PSM1 channel 7 shared interrupt. See bit 0.
14	PSM_M1_CH6_IRQ	PSM1 channel 6 shared interrupt. See bit 0.
13	PSM_M1_CH5_IRQ	PSM1 channel 5 shared interrupt. See bit 0.
12	PSM_M1_CH4_IRQ	PSM1 channel 4 shared interrupt. See bit 0.
11	PSM_M1_CH3_IRQ	PSM1 channel 3 shared interrupt. See bit 0.
10	PSM_M1_CH2_IRQ	PSM1 channel 2 shared interrupt. See bit 0.
9	PSM_M1_CH1_IRQ	PSM1 channel 1 shared interrupt. See bit 0.
8	PSM_M1_CH0_IRQ	PSM1 channel 0 shared interrupt. See bit 0.
7	PSM_M0_CH7_IRQ	PSM0 channel 7 shared interrupt. See bit 0.
6	PSM_M0_CH6_IRQ	PSM0 channel 6 shared interrupt. See bit 0.
5	PSM_M0_CH5_IRQ	PSM0 channel 5 shared interrupt. See bit 0.
4	PSM_M0_CH4_IRQ	PSM0 channel 4 shared interrupt. See bit 0.
3	PSM_M0_CH3_IRQ	PSM0 channel 3 shared interrupt. See bit 0.
2	PSM_M0_CH2_IRQ	PSM0 channel 2 shared interrupt. See bit 0.
1	PSM_M0_CH1_IRQ	PSM0 channel 1 shared interrupt. See bit 0.

Table 38.484 ICM_IRQG_PSM_0_CI Register Contents (2/2)

Bit Position	Bit Name	Function
0	PSM_M0_CH0_IRQ	PSM0 channel 0 shared interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the four interrupt sources FIFO_EMPTY, FIFO_FULL, FIFO_LOWER_WM or FIFO_UPPER_WM of FIFO instance 0 channel 0.

38.24.5.24 ICM_IRQG_PSM_0_CEI

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + $006A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PSM_M2_CH7_EIRQ	PSM_M2_CH6_EIRQ	PSM_M2_CH5_EIRQ	PSM_M2_CH4_EIRQ	PSM_M2_CH3_EIRQ	PSM_M2_CH2_EIRQ	PSM_M2_CH1_EIRQ	PSM_M2_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSM_M1_CH7_EIRQ	PSM_M1_CH6_EIRQ	PSM_M1_CH5_EIRQ	PSM_M1_CH4_EIRQ	PSM_M1_CH3_EIRQ	PSM_M1_CH2_EIRQ	PSM_M1_CH1_EIRQ	PSM_M1_CH0_EIRQ	PSM_M0_CH7_EIRQ	PSM_M0_CH6_EIRQ	PSM_M0_CH5_EIRQ	PSM_M0_CH4_EIRQ	PSM_M0_CH3_EIRQ	PSM_M0_CH2_EIRQ	PSM_M0_CH1_EIRQ	PSM_M0_CH0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.485 ICM_IRQG_PSM_0_CEI Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	PSM_M2_CH7_EIRQ	PSM2 channel 7 error interrupt. See bit 0.
22	PSM_M2_CH6_EIRQ	PSM2 channel 6 error interrupt. See bit 0.
21	PSM_M2_CH5_EIRQ	PSM2 channel 5 error interrupt. See bit 0.
20	PSM_M2_CH4_EIRQ	PSM2 channel 4 error interrupt. See bit 0.
19	PSM_M2_CH3_EIRQ	PSM2 channel 3 error interrupt. See bit 0.
18	PSM_M2_CH2_EIRQ	PSM2 channel 2 error interrupt. See bit 0.
17	PSM_M2_CH1_EIRQ	PSM2 channel 1 error interrupt. See bit 0.
16	PSM_M2_CH0_EIRQ	PSM2 channel 0 error interrupt. See bit 0.
15	PSM_M1_CH7_EIRQ	PSM1 channel 7 error interrupt. See bit 0.
14	PSM_M1_CH6_EIRQ	PSM1 channel 6 error interrupt. See bit 0.
13	PSM_M1_CH5_EIRQ	PSM1 channel 5 error interrupt. See bit 0.
12	PSM_M1_CH4_EIRQ	PSM1 channel 4 error interrupt. See bit 0.
11	PSM_M1_CH3_EIRQ	PSM1 channel 3 error interrupt. See bit 0.
10	PSM_M1_CH2_EIRQ	PSM1 channel 2 error interrupt. See bit 0.
9	PSM_M1_CH1_EIRQ	PSM1 channel 1 error interrupt. See bit 0.
8	PSM_M1_CH0_EIRQ	PSM1 channel 0 error interrupt. See bit 0.
7	PSM_M0_CH7_EIRQ	PSM0 channel 7 error interrupt. See bit 0.
6	PSM_M0_CH6_EIRQ	PSM0 channel 6 error interrupt. See bit 0.
5	PSM_M0_CH5_EIRQ	PSM0 channel 5 error interrupt. See bit 0.
4	PSM_M0_CH4_EIRQ	PSM0 channel 4 error interrupt. See bit 0.
3	PSM_M0_CH3_EIRQ	PSM0 channel 3 error interrupt. See bit 0.
2	PSM_M0_CH2_EIRQ	PSM0 channel 2 error interrupt. See bit 0.
1	PSM_M0_CH1_EIRQ	PSM0 channel 1 error interrupt. See bit 0.

Table 38.485 ICM_IRQG_PSM_0_CEI Register Contents (2/2)

Bit Position	Bit Name	Function
0	PSM_M0_CH0_EIRQ	PSM0 channel 0 error interrupt. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.24.5.25 ICM_IRQG_TOM_[g]_CI

Access: This register is a read-only register that can be read in 32-bit units.

Address: ICM_IRQG_TOM_0_CI: <GTM_base> + $007A0_H$
ICM_IRQG_TOM_1_CI: <GTM_base> + $007A4_H$
ICM_IRQG_TOM_2_CI: <GTM_base> + $007A8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TOM_M1_CH15_IRQ	TOM_M1_CH14_IRQ	TOM_M1_CH13_IRQ	TOM_M1_CH12_IRQ	TOM_M1_CH11_IRQ	TOM_M1_CH10_IRQ	TOM_M1_CH9_IRQ	TOM_M1_CH8_IRQ	TOM_M1_CH7_IRQ	TOM_M1_CH6_IRQ	TOM_M1_CH5_IRQ	TOM_M1_CH4_IRQ	TOM_M1_CH3_IRQ	TOM_M1_CH2_IRQ	TOM_M1_CH1_IRQ	TOM_M1_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TOM_M0_CH15_IRQ	TOM_M0_CH14_IRQ	TOM_M0_CH13_IRQ	TOM_M0_CH12_IRQ	TOM_M0_CH11_IRQ	TOM_M0_CH10_IRQ	TOM_M0_CH9_IRQ	TOM_M0_CH8_IRQ	TOM_M0_CH7_IRQ	TOM_M0_CH6_IRQ	TOM_M0_CH5_IRQ	TOM_M0_CH4_IRQ	TOM_M0_CH3_IRQ	TOM_M0_CH2_IRQ	TOM_M0_CH1_IRQ	TOM_M0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.486 ICM_IRQG_TOM_[g]_CI Register Contents (1/2)

Bit Position	Bit Name	Function
31	TOM_M1_CH15_IRQ	TOM1 channel 15 interrupt. See bit 0.
30	TOM_M1_CH14_IRQ	TOM1 channel 14 interrupt. See bit 0.
29	TOM_M1_CH13_IRQ	TOM1 channel 13 interrupt. See bit 0.
28	TOM_M1_CH12_IRQ	TOM1 channel 12 interrupt. See bit 0.
27	TOM_M1_CH11_IRQ	TOM1 channel 11 interrupt. See bit 0.
26	TOM_M1_CH10_IRQ	TOM1 channel 10 interrupt. See bit 0.
25	TOM_M1_CH9_IRQ	TOM1 channel 9 interrupt. See bit 0.
24	TOM_M1_CH8_IRQ	TOM1 channel 8 interrupt. See bit 0.
23	TOM_M1_CH7_IRQ	TOM1 channel 7 interrupt. See bit 0.
22	TOM_M1_CH6_IRQ	TOM1 channel 6 interrupt. See bit 0.
21	TOM_M1_CH5_IRQ	TOM1 channel 5 interrupt. See bit 0.
20	TOM_M1_CH4_IRQ	TOM1 channel 4 interrupt. See bit 0.
19	TOM_M1_CH3_IRQ	TOM1 channel 3 interrupt. See bit 0.
18	TOM_M1_CH2_IRQ	TOM1 channel 2 interrupt. See bit 0.
17	TOM_M1_CH1_IRQ	TOM1 channel 1 interrupt. See bit 0.
16	TOM_M1_CH0_IRQ	TOM1 channel 0 interrupt. See bit 0.
15	TOM_M0_CH15_IRQ	TOM0 channel 15 interrupt. See bit 0.
14	TOM_M0_CH14_IRQ	TOM0 channel 14 interrupt. See bit 0.
13	TOM_M0_CH13_IRQ	TOM0 channel 13 interrupt. See bit 0.
12	TOM_M0_CH12_IRQ	TOM0 channel 12 interrupt. See bit 0.
11	TOM_M0_CH11_IRQ	TOM0 channel 11 interrupt. See bit 0.
10	TOM_M0_CH10_IRQ	TOM0 channel 10 interrupt. See bit 0.
9	TOM_M0_CH9_IRQ	TOM0 channel 9 interrupt. See bit 0.
8	TOM_M0_CH8_IRQ	TOM0 channel 8 interrupt. See bit 0.
7	TOM_M0_CH7_IRQ	TOM0 channel 7 interrupt. See bit 0.
6	TOM_M0_CH6_IRQ	TOM0 channel 6 interrupt. See bit 0.
5	TOM_M0_CH5_IRQ	TOM0 channel 5 interrupt. See bit 0.

Table 38.486 ICM_IRQG_TOM_[g]_CI Register Contents (2/2)

Bit Position	Bit Name	Function
4	TOM_M0_CH4_IRQ	TOM0 channel 4 interrupt. See bit 0.
3	TOM_M0_CH3_IRQ	TOM0 channel 3 interrupt. See bit 0.
2	TOM_M0_CH2_IRQ	TOM0 channel 2 interrupt. See bit 0.
1	TOM_M0_CH1_IRQ	TOM0 channel 1 interrupt. See bit 0.
0	TOM_M0_CH0_IRQ	TOM0 channel 0 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources TOM_CCU0TCx_IRQ or TOM_CCU1TCx_IRQ of TOM instance 0 channel 0.

38.24.5.26 ICM_IRQG_ATOM_[g]_CI

Access: This register is a read-only register that can be read in 32-bit units.

Address: ICM_IRQG_ATOM_0_CI: <GTM_base> + 00790_H
ICM_IRQG_ATOM_1_CI: <GTM_base> + 00794_H
ICM_IRQG_ATOM_2_CI: <GTM_base> + 00798_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ATOM_M3_CH7_IRQ	ATOM_M3_CH6_IRQ	ATOM_M3_CH5_IRQ	ATOM_M3_CH4_IRQ	ATOM_M3_CH3_IRQ	ATOM_M3_CH2_IRQ	ATOM_M3_CH1_IRQ	ATOM_M3_CH0_IRQ	ATOM_M2_CH7_IRQ	ATOM_M2_CH6_IRQ	ATOM_M2_CH5_IRQ	ATOM_M2_CH4_IRQ	ATOM_M2_CH3_IRQ	ATOM_M2_CH2_IRQ	ATOM_M2_CH1_IRQ	ATOM_M2_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ATOM_M1_CH7_IRQ	ATOM_M1_CH6_IRQ	ATOM_M1_CH5_IRQ	ATOM_M1_CH4_IRQ	ATOM_M1_CH3_IRQ	ATOM_M1_CH2_IRQ	ATOM_M1_CH1_IRQ	ATOM_M1_CH0_IRQ	ATOM_M0_CH7_IRQ	ATOM_M0_CH6_IRQ	ATOM_M0_CH5_IRQ	ATOM_M0_CH4_IRQ	ATOM_M0_CH3_IRQ	ATOM_M0_CH2_IRQ	ATOM_M0_CH1_IRQ	ATOM_M0_CH0_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.487 ICM_IRQG_ATOM_[g]_CI Register Contents (1/2)

Bit Position	Bit Name	Function
31	ATOM_M3_CH7_IRQ	ATOM3 channel 7 interrupt. See bit 0.
30	ATOM_M3_CH6_IRQ	ATOM3 channel 6 interrupt. See bit 0.
29	ATOM_M3_CH5_IRQ	ATOM3 channel 5 interrupt. See bit 0.
28	ATOM_M3_CH4_IRQ	ATOM3 channel 4 interrupt. See bit 0.
27	ATOM_M3_CH3_IRQ	ATOM3 channel 3 interrupt. See bit 0.
26	ATOM_M3_CH2_IRQ	ATOM3 channel 2 interrupt. See bit 0.
25	ATOM_M3_CH1_IRQ	ATOM3 channel 1 interrupt. See bit 0.
24	ATOM_M3_CH0_IRQ	ATOM3 channel 0 interrupt. See bit 0.
23	ATOM_M2_CH7_IRQ	ATOM2 channel 7 interrupt. See bit 0.
22	ATOM_M2_CH6_IRQ	ATOM2 channel 6 interrupt. See bit 0.
21	ATOM_M2_CH5_IRQ	ATOM2 channel 5 interrupt. See bit 0.
20	ATOM_M2_CH4_IRQ	ATOM2 channel 4 interrupt. See bit 0.
19	ATOM_M2_CH3_IRQ	ATOM2 channel 3 interrupt. See bit 0.
18	ATOM_M2_CH2_IRQ	ATOM2 channel 2 interrupt. See bit 0.
17	ATOM_M2_CH1_IRQ	ATOM2 channel 1 interrupt. See bit 0.
16	ATOM_M2_CH0_IRQ	ATOM2 channel 0 interrupt. See bit 0.
15	ATOM_M1_CH7_IRQ	ATOM1 channel 7 interrupt. See bit 0.
14	ATOM_M1_CH6_IRQ	ATOM1 channel 6 interrupt. See bit 0.
13	ATOM_M1_CH5_IRQ	ATOM1 channel 5 interrupt. See bit 0.
12	ATOM_M1_CH4_IRQ	ATOM1 channel 4 interrupt. See bit 0.
11	ATOM_M1_CH3_IRQ	ATOM1 channel 3 interrupt. See bit 0.
10	ATOM_M1_CH2_IRQ	ATOM1 channel 2 interrupt. See bit 0.
9	ATOM_M1_CH1_IRQ	ATOM1 channel 1 interrupt. See bit 0.
8	ATOM_M1_CH0_IRQ	ATOM1 channel 0 interrupt. See bit 0.
7	ATOM_M0_CH7_IRQ	ATOM0 channel 7 interrupt. See bit 0.
6	ATOM_M0_CH6_IRQ	ATOM0 channel 6 interrupt. See bit 0.
5	ATOM_M0_CH5_IRQ	ATOM0 channel 5 interrupt. See bit 0.

Table 38.487 ICM_IRQG_ATOM_[g]_CI Register Contents (2/2)

Bit Position	Bit Name	Function
4	ATOM_M0_CH4_IRQ	ATOM0 channel 4 interrupt. See bit 0.
3	ATOM_M0_CH3_IRQ	ATOM0 channel 3 interrupt. See bit 0.
2	ATOM_M0_CH2_IRQ	ATOM0 channel 2 interrupt. See bit 0.
1	ATOM_M0_CH1_IRQ	ATOM0 channel 1 interrupt. See bit 0.
0	ATOM_M0_CH0_IRQ	ATOM0 channel 0 interrupt. 0: No interrupt occurred. 1: Interrupt was raised by the corresponding sub-module. **NOTES** 1. This bit is only set, when the interrupt is enabled in the interrupt enable register of the corresponding sub-module. 2. Set this bit represents an OR function of the two interrupt sources CCU0TCx_IRQ or CCU1TCx_IRQ of ATOM instance 0 channel 0.

38.24.5.27 ICM_IRQG_CLS_[g]_MEI

Access: This register is a read-only register that can be read in 32-bit units.

Address: ICM_IRQG_CLS_0_CI: <GTM_base> + 00710_H
ICM_IRQG_CLS_1_CI: <GTM_base> + 00714_H
ICM_IRQG_CLS_2_CI: <GTM_base> + 00718_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	FIFO_M3_EIRQ	SPE_M3_EIRQ	MCS_M3_EIRQ	TIM_M3_EIRQ	—	—	—	—	FIFO_M2_EIRQ	SPE_M2_EIRQ	MCS_M2_EIRQ	TIM_M2_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FIFO_M1_EIRQ	SPE_M1_EIRQ	MCS_M1_EIRQ	TIM_M1_EIRQ	—	—	—	—	FIFO_M0_EIRQ	SPE_M0_EIRQ	MCS_M0_EIRQ	TIM_M0_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.488 ICM_IRQG_CLS_[g]_MEI Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned.
27	FIFO_M3_EIRQ	Error interrupt FIFO3_EIRQ. See bit 0.
26	SPE_M3_EIRQ	Error interrupt SPE3_EIRQ. See bit 0.
25	MCS_M3_EIRQ	Error interrupt MCS3_EIRQ. See bit 0.
24	TIM_M3_EIRQ	Error interrupt TIM3_EIRQ. See bit 0.
23 to 20	Reserved	When read, the value after reset is returned.
19	FIFO_M2_EIRQ	Error interrupt FIFO2_EIRQ. See bit 0.
18	SPE_M2_EIRQ	Error interrupt SPE2_EIRQ. See bit 0.
17	MCS_M2_EIRQ	Error interrupt MCS2_EIRQ. See bit 0.
16	TIM_M2_EIRQ	Error interrupt TIM2_EIRQ. See bit 0.
15 to 12	Reserved	When read, the value after reset is returned.
11	FIFO_M1_EIRQ	Error interrupt FIFO1_EIRQ. See bit 0.
10	SPE_M1_EIRQ	Error interrupt SPE1_EIRQ. See bit 0.
9	MCS_M1_EIRQ	Error interrupt MCS1_EIRQ. See bit 0.
8	TIM_M1_EIRQ	Error interrupt TIM1_EIRQ. See bit 0.
7 to 4	Reserved	When read, the value after reset is returned.
3	FIFO_M0_EIRQ	Error interrupt FIFO0_EIRQ. See bit 0.
2	SPE_M0_EIRQ	Error interrupt SPE0_EIRQ. See bit 0.
1	MCS_M0_EIRQ	Error interrupt MCS0_EIRQ. See bit 0.
0	TIM_M0_EIRQ	Error interrupt TIM0_EIRQ. 0: No error interrupt occurred. 1: Error interrupt was raised by the corresponding sub-module. **NOTE** This bit is only set, when the error interrupt is enabled in the error interrupt enable register of the corresponding sub-module.

38.25 Output Compare Unit (CMP)

38.25.1 Overview

The Output Compare Unit (CMP) is designed for the use in safety relevant applications. The main idea is to have the possibility to duplicate outputs in order to be compared in this unit. Because of the simple EXOR function used it is necessary to ensure the total cycle accurate output behavior of the output modules to be compared. This is given when two neighbored DTM channel (CDTM[n]_DTM[2*i] and CDTM[n]_-DTM[2*i+1]) generate identical signals with phase shift zero at their outputs. This can be reached if they start their output generation at the same time. This start of synchronization is possible by means of the trigger mechanisms TRIG_x provided by the TOM or ATOM as shown in **Section 38.16, Timer Output Module (TOM)** or **Section 38.17, ARU-connected Timer Output Module (ATOM)**. It is not necessary to compare each output channel with each other.

The CMP enables the comparison of 2×24 channels of the CDTM0, CDTM1 and CDTM2 and is restricted to neighbored channels. The first 24 CMP channels are the first 24 DTM channels placed behind TOM0 and TOM1 and the second 24 CMP channels are the first 24 DTM channels placed behind the ATOM0, ATOM1 and ATOM2.

NOTE

When the channels were generated with a higher frequency than the frequency of cluster 1 it is not certain to catch the interrupt in the notify register. Avoid a comparison if frequency is unequal cluster 1 AND (cluster 0 OR cluster 2).

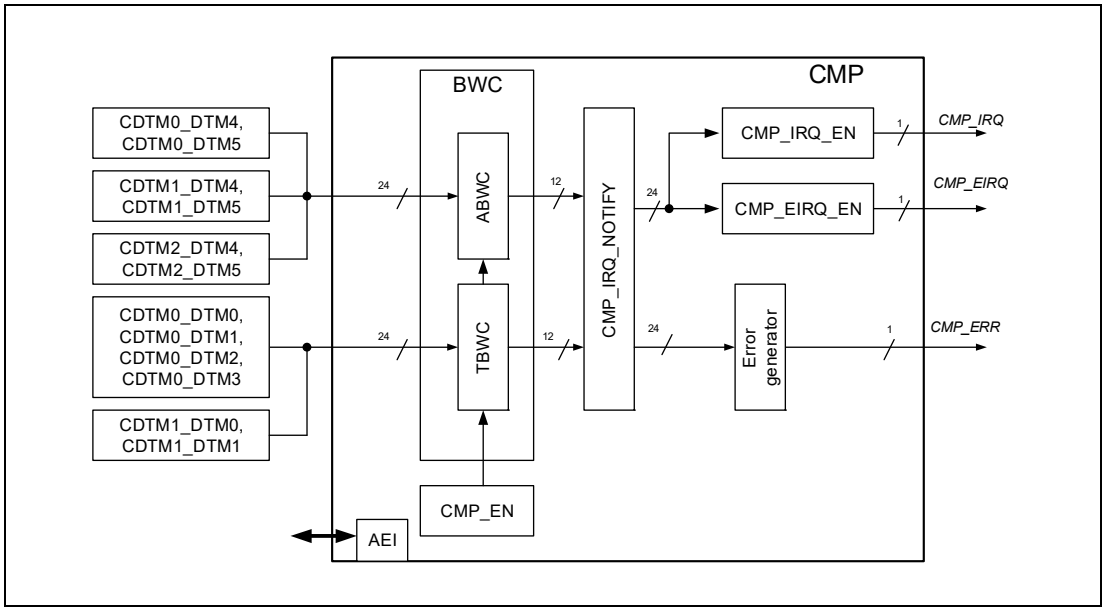


Figure 38.124 Architecture of the Compare Unit

38.25.2 Bitwise Compare Unit (BWC)

The Bitwise Compare Unit ABWC compares in pairs the combinations shown in following table.

Table 38.489 ABWC Compare Unit(1)

ABWC Comparator	Comparator Input 1	Comparator Input 2
ABWC0	CDTM0_DTM4_CH0_OUT	CDTM0_DTM4_CH1_OUT
ABWC1	CDTM0_DTM4_CH2_OUT	CDTM0_DTM4_CH3_OUT
ABWC2	CDTM0_DTM5_CH0_OUT	CDTM0_DTM5_CH1_OUT
ABWC3	CDTM0_DTM5_CH2_OUT	CDTM0_DTM5_CH3_OUT
ABWC4	CDTM1_DTM4_CH0_OUT	CDTM1_DTM4_CH1_OUT
ABWC5	CDTM1_DTM4_CH2_OUT	CDTM1_DTM4_CH3_OUT
ABWC6	CDTM1_DTM5_CH0_OUT	CDTM1_DTM5_CH1_OUT
ABWC7	CDTM1_DTM5_CH2_OUT	CDTM1_DTM5_CH3_OUT
ABWC8	CDTM2_DTM4_CH0_OUT	CDTM2_DTM4_CH1_OUT
ABWC9	CDTM2_DTM4_CH2_OUT	CDTM2_DTM4_CH3_OUT
ABWC10	CDTM2_DTM5_CH0_OUT	CDTM2_DTM5_CH1_OUT
ABWC11	CDTM2_DTM5_CH2_OUT	CDTM2_DTM5_CH3_OUT

The Bitwise Compare Unit TBWC compares in pairs the combinations shown in following table.

Table 38.490 TBWC compare unit

TBWC Comparator	Comparator Input 1	Comparator Input 2
TBWC0	CDTM0_DTM0_CH0_OUT	CDTM0_DTM0_CH1_OUT
TBWC1	CDTM0_DTM0_CH2_OUT	CDTM0_DTM0_CH3_OUT
TBWC2	CDTM0_DTM1_CH0_OUT	CDTM0_DTM1_CH1_OUT
TBWC3	CDTM0_DTM1_CH2_OUT	CDTM0_DTM1_CH3_OUT
TBWC4	CDTM0_DTM2_CH0_OUT	CDTM0_DTM2_CH1_OUT
TBWC5	CDTM0_DTM2_CH2_OUT	CDTM0_DTM2_CH3_OUT
TBWC6	CDTM0_DTM3_CH0_OUT	CDTM0_DTM3_CH1_OUT
TBWC7	CDTM0_DTM3_CH2_OUT	CDTM0_DTM3_CH3_OUT
TBWC8	CDTM1_DTM0_CH0_OUT	CDTM1_DTM0_CH1_OUT
TBWC9	CDTM1_DTM0_CH2_OUT	CDTM1_DTM0_CH3_OUT
TBWC10	CDTM1_DTM1_CH0_OUT	CDTM1_DTM1_CH1_OUT
TBWC11	CDTM1_DTM1_CH2_OUT	CDTM1_DTM1_CH3_OUT

38.25.3 Configuration of the Compare Unit

Because of the restrictions described in the section above the Compare Unit consists of 24 anti valence (EXOR) elements, a select register CMP_EN which selects the corresponding comparisons and a status register CMP_IRQ_NOTIFY which shows and stores each mismatching result, when selected.

For each with CMP_IRQ_EN enabled mismatching error an interrupt signal on CMP_IRQ is generated.

For each with CMP_EIRQ_EN enabled mismatching error an interrupt signal on CMP_EIRQ is generated.

38.25.4 Error Generator

The error generator generates an error signal to be transmitted directly to the MON unit and independently from the CMP_IRQ and CMP_EIRQ. The error is set when in the CMP_IRQ_NOTIFY register at least one bit is set. The CMP_IRQ_NOTIFY bits are not mask able for this purpose.

Additionally CMP_ERR is a primary output port for interrupt actions by CPU itself.

38.25.5 CMP Interrupt Signal

The CMP sub-module has two interrupt signals, one normal interrupt and one error interrupt. The source of both interrupt can be determined by reading the CMP_IRQ_NOTIFY register under consideration of CMP_IRQ_EN register and CMP_EIRQ_EN register. Each source can be forced separately for debug purposes using the interrupt force CMP_IRQ_FORCINT register. CMP_IRQ_MODE configures interrupt output characteristic. All interrupt modes are described in detail in **Section 38.5.5, GTM-IP Interrupt Concept**.

Table 38.491 CMP Interrupt Signal

Signal	Description
CMP_EIRQ	Mismatching interrupt of outputs to be compared, when enabled
CMP_IRQ	Mismatching interrupt of outputs to be compared, when enabled

38.25.6 CMP Configuration Registers Overview

CMP contains following configuration registers:

Table 38.492 Register list

Symbol	Register Name	Details in Section
CMP_EN	CMP comparator enable register	**38.25.7.1**
CMP_IRQ_NOTIFY	CMP event notification register	**38.25.7.2**
CMP_IRQ_EN	CMP interrupt enable register	**38.25.7.3**
CMP_IRQ_FORCINT	CMP interrupt force register	**38.25.7.4**
CMP_IRQ_MODE	CMP interrupt mode configuration register	**38.25.7.5**
CMP_EIRQ_EN	CMP error interrupt enable register	**38.25.7.6**

38.25.7 CMP Configuration Registers Description

38.25.7.1 CMP_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00200_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBWC11_EN	TBWC10_EN	TBWC9_EN	TBWC8_EN	TBWC7_EN	TBWC6_EN	TBWC5_EN	TBWC4_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBWC3_EN	TBWC2_EN	TBWC1_EN	TBWC0_EN	ABWC11_EN	ABWC10_EN	ABWC9_EN	ABWC8_EN	ABWC7_EN	ABWC6_EN	ABWC5_EN	ABWC4_EN	ABWC3_EN	ABWC2_EN	ABWC1_EN	ABWC0_EN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.493 CMP_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	TBWC11_EN	Enable comparator 11 in TBWC (see **Section 38.25.2**). See bit 12.
22	TBWC10_EN	Enable comparator 10 in TBWC (see **Section 38.25.2**). See bit 12.
21	TBWC9_EN	Enable comparator 9 in TBWC (see **Section 38.25.2**). See bit 12.
20	TBWC8_EN	Enable comparator 8 in TBWC (see **Section 38.25.2**). See bit 12.
19	TBWC7_EN	Enable comparator 7 in TBWC (see **Section 38.25.2**). See bit 12.
18	TBWC6_EN	Enable comparator 6 in TBWC (see **Section 38.25.2**). See bit 12.
17	TBWC5_EN	Enable comparator 5 in TBWC (see **Section 38.25.2**). See bit 12.
16	TBWC4_EN	Enable comparator 4 in TBWC (see **Section 38.25.2**). See bit 12.
15	TBWC3_EN	Enable comparator 3 in TBWC (see **Section 38.25.2**). See bit 12.
14	TBWC2_EN	Enable comparator 2 in TBWC (see **Section 38.25.2**). See bit 12.
13	TBWC1_EN	Enable comparator 1 in TBWC (see **Section 38.25.2**). See bit 12.
12	TBWC0_EN	Enable comparator 0 in TBWC (see **Section 38.25.2**). 0: TBWC comparator 0 is disabled. 1: TBWC comparator 0 is enabled.
11	ABWC11_EN	Enable comparator 11 in ABWC (see **Section 38.25.2**). See bit 0.
10	ABWC10_EN	Enable comparator 10 in ABWC (see **Section 38.25.2**). See bit 0.
9	ABWC9_EN	Enable comparator 9 in ABWC (see **Section 38.25.2**). See bit 0.
8	ABWC8_EN	Enable comparator 8 in ABWC (see **Section 38.25.2**). See bit 0.
7	ABWC7_EN	Enable comparator 7 in ABWC (see **Section 38.25.2**). See bit 0.
6	ABWC6_EN	Enable comparator 6 in ABWC (see **Section 38.25.2**). See bit 0.
5	ABWC5_EN	Enable comparator 5 in ABWC (see **Section 38.25.2**). See bit 0.
4	ABWC4_EN	Enable comparator 4 in ABWC (see **Section 38.25.2**). See bit 0.
3	ABWC3_EN	Enable comparator 3 in ABWC (see **Section 38.25.2**). See bit 0.
2	ABWC2_EN	Enable comparator 2 in ABWC (see **Section 38.25.2**). See bit 0.
1	ABWC1_EN	Enable comparator 1 in ABWC (see **Section 38.25.2**). See bit 0.

Table 38.493 CMP_EN Register Contents (2/2)

Bit Position	Bit Name	Function
0	ABWC0_EN	Enable comparator 0 in ABWC (see **Section 38.25.2**). 0: ABWC Comparator 0 is disabled 1: ABWC Comparator 0 is enabled

38.25.7.2 CMP_IRQ_NOTIFY

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GTM_base> + 00204_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBWC11	TBWC10	TBWC9	TBWC8	TBWC7	TBWC6	TBWC5	TBWC4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBWC3	TBWC2	TBWC1	TBWC0	ABWC11	ABWC10	ABWC9	ABWC8	ABWC7	ABWC6	ABWC5	ABWC4	ABWC3	ABWC2	ABWC1	ABWC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 38.494 CMP_IRQ_NOTIFY Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	TBWC11	TOM sub modules outputs bitwise comparator 11 error indication. See bit 12.
22	TBWC10	TOM sub modules outputs bitwise comparator 10 error indication. See bit 12.
21	TBWC9	TOM sub modules outputs bitwise comparator 9 error indication. See bit 12.
20	TBWC8	TOM sub modules outputs bitwise comparator 8 error indication. See bit 12.
19	TBWC7	TOM sub modules outputs bitwise comparator 7 error indication. See bit 12.
18	TBWC6	TOM sub modules outputs bitwise comparator 6 error indication. See bit 12.
17	TBWC5	TOM sub modules outputs bitwise comparator 5 error indication. See bit 12.
16	TBWC4	TOM sub modules outputs bitwise comparator 4 error indication. See bit 12.
15	TBWC3	TOM sub modules outputs bitwise comparator 3 error indication. See bit 12.
14	TBWC2	TOM sub modules outputs bitwise comparator 2 error indication. See bit 12.
13	TBWC1	TOM sub modules outputs bitwise comparator 1 error indication. See bit 12.
12	TBWC0	TOM sub modules outputs bitwise comparator 0 error indication. 0: No error recognized on DTMT sub modules bits 0 and 1 (see **Section 38.25.2**). 1: An error was recognized on corresponding DTMT sub modules bits. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
11	ABWC11	Error indication for ABWC11. See bit 0.
10	ABWC10	Error indication for ABWC10. See bit 0.
9	ABWC9	Error indication for ABWC9. See bit 0.
8	ABWC8	Error indication for ABWC8. See bit 0.
7	ABWC7	Error indication for ABWC7. See bit 0.
6	ABWC6	Error indication for ABWC6. See bit 0.
5	ABWC5	Error indication for ABWC5. See bit 0.
4	ABWC4	Error indication for ABWC4. See bit 0.
3	ABWC3	Error indication for ABWC3. See bit 0.

Table 38.494 CMP_IRQ_NOTIFY Register Contents (2/2)

Bit Position	Bit Name	Function
2	ABWC2	Error indication for ABWC2. See bit 0.
1	ABWC1	Error indication for ABWC1. See bit 0.
0	ABWC0	Error indication for ABWC0. 0: No error recognized on DTMA sub modules bits 0 and 1 (see **Section 38.25.2**). 1: An error was recognized on corresponding DTMA sub modules bits. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

38.25.7.3 CMP_IRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00208_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBWC11_EN_IRQ	TBWC10_EN_IRQ	TBWC9_EN_IRQ	TBWC8_EN_IRQ	TBWC7_EN_IRQ	TBWC6_EN_IRQ	TBWC5_EN_IRQ	TBWC4_EN_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBWC3_EN_IRQ	TBWC2_EN_IRQ	TBWC1_EN_IRQ	TBWC0_EN_IRQ	ABWC11_EN_IRQ	ABWC10_EN_IRQ	ABWC9_EN_IRQ	ABWC8_EN_IRQ	ABWC7_EN_IRQ	ABWC6_EN_IRQ	ABWC5_EN_IRQ	ABWC4_EN_IRQ	ABWC3_EN_IRQ	ABWC2_EN_IRQ	ABWC1_EN_IRQ	ABWC0_EN_IRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.495 CMP_IRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	TBWC11_EN_IRQ	Enable TBWC11 interrupt source for CMP_IRQ line. See bit 12.
22	TBWC10_EN_IRQ	Enable TBWC10 interrupt source for CMP_IRQ line. See bit 12.
21	TBWC9_EN_IRQ	Enable TBWC9 interrupt source for CMP_IRQ line. See bit 12.
20	TBWC8_EN_IRQ	Enable TBWC8 interrupt source for CMP_IRQ line. See bit 12.
19	TBWC7_EN_IRQ	Enable TBWC7 interrupt source for CMP_IRQ line. See bit 12.
18	TBWC6_EN_IRQ	Enable TBWC6 interrupt source for CMP_IRQ line. See bit 12.
17	TBWC5_EN_IRQ	Enable TBWC5 interrupt source for CMP_IRQ line. See bit 12.
16	TBWC4_EN_IRQ	Enable TBWC4 interrupt source for CMP_IRQ line. See bit 12.
15	TBWC3_EN_IRQ	Enable TBWC3 interrupt source for CMP_IRQ line. See bit 12.
14	TBWC2_EN_IRQ	Enable TBWC2 interrupt source for CMP_IRQ line. See bit 12.
13	TBWC1_EN_IRQ	Enable TBWC1 interrupt source for CMP_IRQ line. See bit 12.
12	TBWC0_EN_IRQ	Enable TBWC0 interrupt source for CMP_IRQ line. 0: Interrupt source TBWC0 is disabled. 1: Interrupt source TBWC0 is enabled.
11	ABWC11_EN_IRQ	Enable ABWC11 interrupt source for CMP_IRQ line. See bit 0.
10	ABWC10_EN_IRQ	Enable ABWC10 interrupt source for CMP_IRQ line. See bit 0.
9	ABWC9_EN_IRQ	Enable ABWC9 interrupt source for CMP_IRQ line. See bit 0.

Table 38.495 CMP_IRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
8	ABWC8_EN_IRQ	Enable ABWC8 interrupt source for CMP_IRQ line. See bit 0.
7	ABWC7_EN_IRQ	Enable ABWC7 interrupt source for CMP_IRQ line. See bit 0.
6	ABWC6_EN_IRQ	Enable ABWC6 interrupt source for CMP_IRQ line. See bit 0.
5	ABWC5_EN_IRQ	Enable ABWC5 interrupt source for CMP_IRQ line. See bit 0.
4	ABWC4_EN_IRQ	Enable ABWC4 interrupt source for CMP_IRQ line. See bit 0.
3	ABWC3_EN_IRQ	Enable ABWC3 interrupt source for CMP_IRQ line. See bit 0.
2	ABWC2_EN_IRQ	Enable ABWC2 interrupt source for CMP_IRQ line. See bit 0.
1	ABWC1_EN_IRQ	Enable ABWC1 interrupt source for CMP_IRQ line. See bit 0.
0	ABWC0_EN_IRQ	Enable ABWC0 interrupt source for CMP_IRQ line. 0: Interrupt source ABWC0 is disabled. 1: Interrupt source ABWC0 is enabled.

38.25.7.4 CMP_IRQ_FORCINT

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + $0020C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TRG_TBWC11	TRG_TBWC10	TRG_TBWC9	TRG_TBWC8	TRG_TBWC7	TRG_TBWC6	TRG_TBWC5	TRG_TBWC4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG_TBWC3	TRG_TBWC2	TRG_TBWC1	TRG_TBWC0	TRG_ABWC11	TRG_ABWC10	TRG_ABWC9	TRG_ABWC8	TRG_ABWC7	TRG_ABWC6	TRG_ABWC5	TRG_ABWC4	TRG_ABWC3	TRG_ABWC2	TRG_ABWC1	TRG_ABWC0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.496 CMP_IRQ_FORCINT Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	TRG_TBWC11	Trigger TBWC11 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
22	TRG_TBWC10	Trigger TBWC10 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
21	TRG_TBWC9	Trigger TBWC9 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
20	TRG_TBWC8	Trigger TBWC8 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
19	TRG_TBWC7	Trigger TBWC7 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
18	TRG_TBWC6	Trigger TBWC6 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
17	TRG_TBWC5	Trigger TBWC5 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
16	TRG_TBWC4	Trigger TBWC4 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
15	TRG_TBWC3	Trigger TBWC3 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
14	TRG_TBWC2	Trigger TBWC2 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
13	TRG_TBWC1	Trigger TBWC1 bit in GTM0CMPIRQNOTIFY register by software. See bit 12.
12	TRG_TBWC0	Trigger TBWC0 bit in CMP_IRQ_NOTIFY register by software 0 = No event triggering 1 = Assert corresponding field in CMP_IRQ_NOTIFY register **NOTE** This bit is cleared automatically after write.
11	TRG_ABWC11	Trigger ABWC11 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
10	TRG_ABWC10	Trigger ABWC10 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
9	TRG_ABWC9	Trigger ABWC9 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
8	TRG_ABWC8	Trigger ABWC8 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
7	TRG_ABWC7	Trigger ABWC7 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
6	TRG_ABWC6	Trigger ABWC6 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
5	TRG_ABWC5	Trigger ABWC5 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
4	TRG_ABWC4	Trigger ABWC4 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
3	TRG_ABWC3	Trigger ABWC3 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
2	TRG_ABWC2	Trigger ABWC2 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.

Table 38.496 CMP_IRQ_FORCINT Register Contents (2/2)

Bit Position	Bit Name	Function
1	TRG_ABWC1	Trigger ABWC1 bit in GTM0CMPIRQNOTIFY register by software. See bit 0.
0	TRG_ABWC0	Trigger ABWC0 bit in GTM0CMPIRQNOTIFY register by software. 0: No event triggering. 1: Assert corresponding field in GTM0CMPIRQNOTIFY register. **NOTES** 1. This bit is cleared automatically after write. 2. This bit is write protected by bit RF_PROT of register GTM0GTMCTRL.

38.25.7.5 CMP_IRQ_MODE

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00210_H

Value after reset: $0000\ 000X_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ_MODE	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 38.497 CMP_IRQ_MODE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IRQ_MODE	IRQ mode selection 00_B: Level mode 01_B: Pulse mode 10_B: Pulse-Notify mode 11_B: Single-Pulse mode **NOTE** The interrupt modes are described in **Section 38.5.5, GTM-IP Interrupt Concept**.

38.25.7.6 CMP_EIRQ_EN

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00214_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	TBWC11_EN_EIRQ	TBWC10_EN_EIRQ	TBWC9_EN_EIRQ	TBWC8_EN_EIRQ	TBWC7_EN_EIRQ	TBWC6_EN_EIRQ	TBWC5_EN_EIRQ	TBWC4_EN_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBWC3_EN_EIRQ	TBWC2_EN_EIRQ	TBWC1_EN_EIRQ	TBWC0_EN_EIRQ	ABWC11_EN_EIRQ	ABWC10_EN_EIRQ	ABWC9_EN_EIRQ	ABWC8_EN_EIRQ	ABWC7_EN_EIRQ	ABWC6_EN_EIRQ	ABWC5_EN_EIRQ	ABWC4_EN_EIRQ	ABWC3_EN_EIRQ	ABWC2_EN_EIRQ	ABWC1_EN_EIRQ	ABWC0_EN_EIRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.498 CMP_EIRQ_EN Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23	TBWC11_EN_EIRQ	Enable TBWC11 interrupt source for CMP_EIRQ line. See bit 12.
22	TBWC10_EN_EIRQ	Enable TBWC10 interrupt source for CMP_EIRQ line. See bit 12.
21	TBWC9_EN_EIRQ	Enable TBWC9 interrupt source for CMP_EIRQ line. See bit 12.
20	TBWC8_EN_EIRQ	Enable TBWC8 interrupt source for CMP_EIRQ line. See bit 12.
19	TBWC7_EN_EIRQ	Enable TBWC7 interrupt source for CMP_EIRQ line. See bit 12.
18	TBWC6_EN_EIRQ	Enable TBWC6 interrupt source for CMP_EIRQ line. See bit 12.
17	TBWC5_EN_EIRQ	Enable TBWC5 interrupt source for CMP_EIRQ line. See bit 12.
16	TBWC4_EN_EIRQ	Enable TBWC4 interrupt source for CMP_EIRQ line. See bit 12.
15	TBWC3_EN_EIRQ	Enable TBWC3 interrupt source for CMP_EIRQ line. See bit 12.
14	TBWC2_EN_EIRQ	Enable TBWC2 interrupt source for CMP_EIRQ line. See bit 12.
13	TBWC1_EN_EIRQ	Enable TBWC1 interrupt source for CMP_EIRQ line. See bit 12.
12	TBWC0_EN_EIRQ	Enable TBWC0 interrupt source for CMP_EIRQ line. 0: Interrupt source TBWC0 is disabled. 1: Interrupt source TBWC0 is enabled.
11	ABWC11_EN_EIRQ	Enable ABWC11 interrupt source for CMP_EIRQ line. See bit 0.
10	ABWC10_EN_EIRQ	Enable ABWC10 interrupt source for CMP_EIRQ line. See bit 0.
9	ABWC9_EN_EIRQ	Enable ABWC9 interrupt source for CMP_EIRQ line. See bit 0.

Table 38.498 CMP_EIRQ_EN Register Contents (2/2)

Bit Position	Bit Name	Function
8	ABWC8_EN_EIRQ	Enable ABWC8 interrupt source for CMP_EIRQ line. See bit 0.
7	ABWC7_EN_EIRQ	Enable ABWC7 interrupt source for CMP_EIRQ line. See bit 0.
6	ABWC6_EN_EIRQ	Enable ABWC6 interrupt source for CMP_EIRQ line. See bit 0.
5	ABWC5_EN_EIRQ	Enable ABWC5 interrupt source for CMP_EIRQ line. See bit 0.
4	ABWC4_EN_EIRQ	Enable ABWC4 interrupt source for CMP_EIRQ line. See bit 0.
3	ABWC3_EN_EIRQ	Enable ABWC3 interrupt source for CMP_EIRQ line. See bit 0.
2	ABWC2_EN_EIRQ	Enable ABWC2 interrupt source for CMP_EIRQ line. See bit 0.
1	ABWC1_EN_EIRQ	Enable ABWC1 interrupt source for CMP_EIRQ line. See bit 0.
0	ABWC0_EN_EIRQ	Enable ABWC0 interrupt source for CMP_EIRQ line 0: Interrupt source ABWC0 is disabled. 1: Interrupt source ABWC0 is enabled.

38.26 Monitor Unit (MON)

38.26.1 Overview

The Monitor Unit (MON) is designed for the use in safety relevant applications. The main idea is to have a possibility to supervise common used circuitry and resources. In this way the activity of the clocks is supervised. In addition the characteristics of output signals can be checked in a MCS channel by a re-read-in via TIM and routing to the MCS. When the comparison fails an error signal is generated in MCS and sent to the monitor unit. One error signal per MCS summarizes the errors of all channels. By generating of an activity signal per channel for each such performed comparison, the activity of TIM, ARU and the used clocks is checked implicitly. In addition the ARU cycle time could be also compared in a MCS channel to given values.

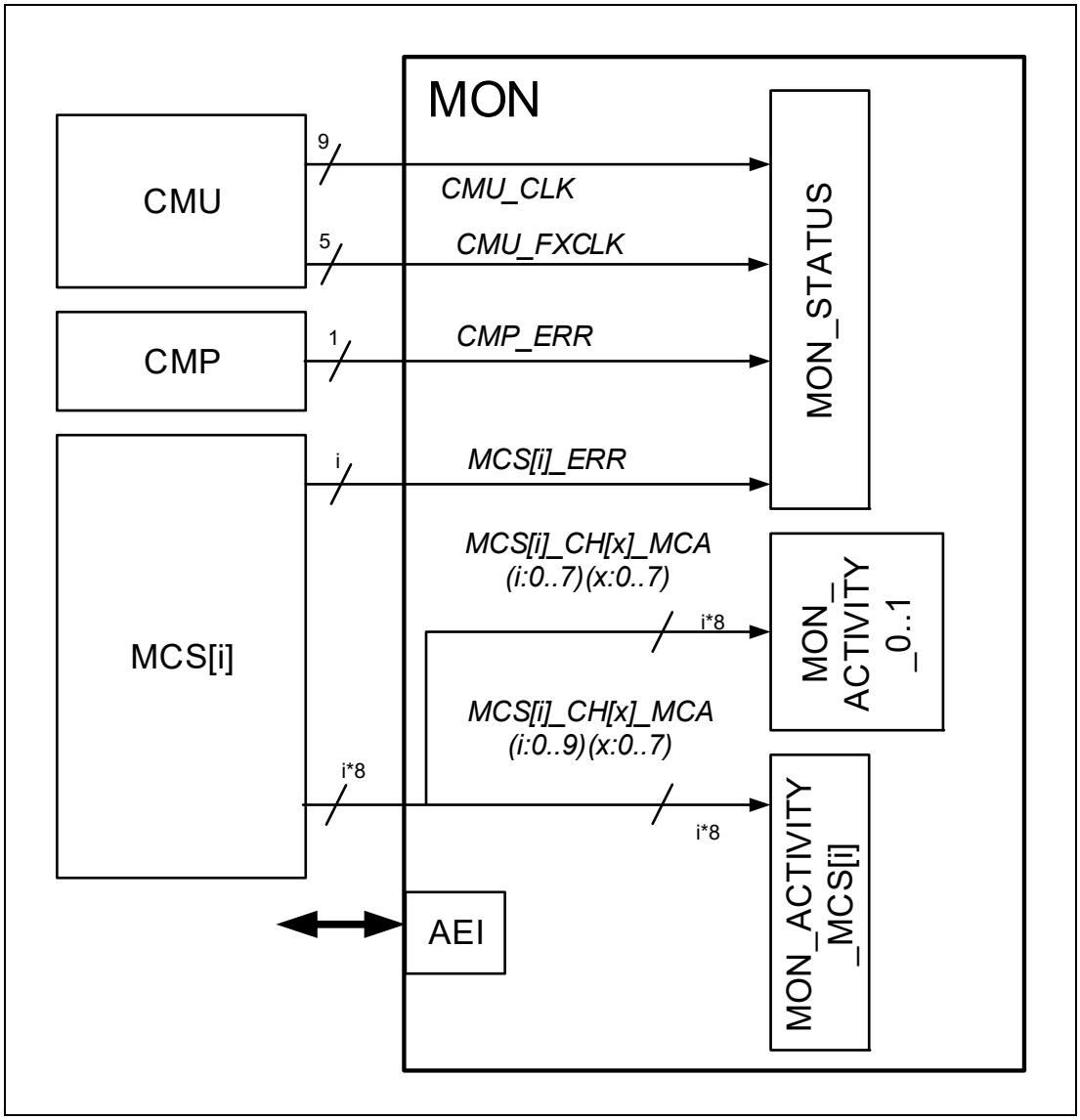


Figure 38.125 MON Block Diagram

38.26.1.1 Realization without Activity Checker of the clock signals

An activity checker of the clock signals used is not needed because these signals are only enables to be used in combination with the system clock. Therefore the clock enables are to be checked to have a high value.

38.26.2 Clock Monitoring

The monitor unit has a connection to each of the 9 clocks CMU_CLK[x] (x = 0 to 8), provided by the CMU. Some of these clocks can be used for special tasks (see **Section 38.12, Clock Management Unit (CMU)**.

In addition the 5 clock inputs of the TOMs CMU_FXCLK[y] (y=0 to 4) are also connected to the MON unit.

The supervising of the clocks is done by scanning for activity of each clock.

A high value is defined as the state to be monitored.

When a high value of the clock enable is detected, the corresponding bit in the status register MON_STATUS is set.

The status register bits are reset by writing a one.

When the register is polled by the CPU and the time between two read accesses is higher than the period of the slowest clock, all bits of the corresponding clocks must have been set.

When polling in shorter time distances, not for all clocks an activity can be shown, although they are still working.

Because of the realization without a select register for the clock signals only the bits of the status register are to be considered for which the clock signal is enabled in the CMU.

38.26.3 CMP error Monitoring

The signal CMP_ERR is to be received directly from module CMP and is set if an error occurred.

38.26.4 Checking the Characteristics of Signals by MCS

By use of the MCS some given properties of signals can be checked. Such signals can be generated output signals of TOM or ATOM channels including DTM function, which are reread in into a TIM and the time stamp information is routed via ARU to the MCS module.

The corresponding MCS signal performs the check according to given properties. In this way signal high or low time as well as signal periods can be checked, also taking into account tolerances. When the check fails a MCS internal error signal is generated and ORed with the error signals of the other channels of the MCS module to a summarized error signal MCS[i]_ERR.

For each MCS a summarized error signal is transmitted to MON and monitored in the MON_STATUS register.

In order to check the execution of the comparison for each MCS channel an activity signal is generated. In the MCS[i]_CH[x]_MCA (i = 0 to 9)(x=0 to 7) vector 8 bits for each MCS[i] (i = 0 to 9) instance are combined. The activity signals are stored in the MON_ACTIVITY_MCS[i] register.

In addition the first 8 bits of MCS0 to 3 are stored in MON_ACTIVITY_0 and the first 8 bits of MCS4 to 7 are stored in MON_ACTIVITY_1. The bits are set by a one signal and reset by writing a one to it (preferably after polling the status of the register).

Because the activity signal shows the execution of a comparison, the involved units for providing the signals and execution of comparison (like TIM, ARU and MCS itself) are checked implicitly to work accordingly. Also the involved clocks and time bases are checked in this way.

38.26.5 Checking ARU Cycle Time

The cycle time of the ARU can be checked, when this is essential for safety purposes. This check can be performed by an MCS channel.

NOTE

The MCS program for measuring the ARU round trip time must add a tolerance value.

The resulting error is reported to the MON unit using the summarized error signal MCS[i]_ERR for each MCS module in addition to an interrupt, generated in MCS. The same signals and status bits are used as in the case of checking the signal characteristics.

The corresponding MCS is programmed to get a fixed data value at address $1FF_H$. The data value is always zero and is not blocked. When getting the access the time stamp value TBU_TS0 is stored in a register. The next time getting the access the new TBU_TS0 value is stored and the difference between both values is compared with a given value. When the comparison fails, an error flag is set in the MCS internal status register, an interrupt is generated and the error signal MCS[i]_ERR is provided.

When the check is performed, an activity signal MCS[i]_CH[x]_MCA (i = 0 to 9)(x = 0 to 7) is provided for each channel x for each MCS[i] (i = 0 to 9) instance together with a summarized interrupt MCS[i]_ERR for each MCS.

The activity signal sets a bit in the MON_ACTIVITY register.

The bits in the MON_ACTIVITY registers are reset by writing a one.

When the check fails, an interrupt is generated and the error signal MCS[i]_ERR is provided for the MON unit.

Figure 38.125 shows the block diagram of the Monitor Unit.

38.26.6 MON Interrupt Signals

The MON sub-module has no interrupt signals.

38.26.7 MON Registers Overview

Following configuration registers are considered in MON sub module

Table 38.499 Register list

Symbol	Register Name	Details in Section
MON_STATUS	Monitor Status register	**38.26.8.1**
MON_ACTIVITY_0	Monitor activity register 0	**38.26.8.2**
MON_ACTIVITY_1	Monitor activity register 1	**38.26.8.3**
MON_ACTIVITY_MCS[z]	Monitor activity register for MCS z	**38.26.8.4**

38.26.8 MON Configuration Registers Description

38.26.8.1 MON_STATUS

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00180_H

Value after reset: $0000\ 4000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MCS9_ERR	MCS8_ERR	MCS7_ERR	MCS6_ERR	MCS5_ERR	MCS4_ERR	MCS3_ERR	MCS2_ERR	MCS1_ERR	MCS0_ERR	—	—	—	CMP_ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ATC_CMU8	—	ATC_CMUFX4	ATC_CMUFX3	ATC_CMUFX2	ATC_CMUFX1	ATC_CMUFX0	ACT_CMU7	ACT_CMU6	ACT_CMU5	ACT_CMU4	ACT_CMU3	ACT_CMU2	ACT_CMU1	ACT_CMU0
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.500 MON_STATUS Register Contents (1/4)

Bit Position	Bit Name	Function
31 to 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MCS9_ERR	Error detected at MCS9 **NOTE** This bit will be readable only.
28	MCS8_ERR	Error detected at MCS8 **NOTE** This bit will be readable only.
27	MCS7_ERR	Error detected at MCS7 **NOTE** This bit will be readable only.
26	MCS6_ERR	Error detected at MCS6 **NOTE** This bit will be readable only.
25	MCS5_ERR	Error detected at MCS5 **NOTE** This bit will be readable only.

Table 38.500 MON_STATUS Register Contents (2/4)

Bit Position	Bit Name	Function
24	MCS4_ERR	Error detected at MCS4 **NOTE** This bit will be readable only.
23	MCS3_ERR	Error detected at MCS3 **NOTE** This bit will be readable only.
22	MCS2_ERR	Error detected at MCS2 **NOTE** This bit will be readable only.
21	MCS1_ERR	Error detected at MCS1 **NOTE** This bit will be readable only.
20	MCS0_ERR	Error detected at MCS0 **NOTE** This bit will be readable only.
19 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	CMP_ERR	Error detected at CMP **NOTE** This bit will be readable only.
15, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	ATC_CMU8	CMU_CLK8 activity **NOTES** 1. This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged. 2. Bit is set, when a rising edge is detected at the considered clock.
12	ATC_CMUFX4	CMU_CLKFX4 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.500 MON_STATUS Register Contents (3/4)

Bit Position	Bit Name	Function
11	ATC_CMUFX3	CMU_CLKFX3 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
10	ATC_CMUFX2	CMU_CLKFX2 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
9	ATC_CMUFX1	CMU_CLKFX1 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
8	ATC_CMUFX0	CMU_CLKFX0 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
7	ACT_CMU7	CMU_CLK7 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
6	ACT_CMU6	CMU_CLK6 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
5	ACT_CMU5	CMU_CLK5 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
4	ACT_CMU4	CMU_CLK4 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.500 MON_STATUS Register Contents (4/4)

Bit Position	Bit Name	Function
3	ACT_CMU3	CMU_CLK3 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
2	ACT_CMU2	CMU_CLK2 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
1	ACT_CMU1	CMU_CLK1 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	ACT_CMU0	CMU_CLK0 activity **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

NOTES

1. Bits 0 to 12 are set, when a rising edge is detected at the considered clock.
2. Bits16 and 20 to 29 are set, when the corresponding unit reports an error.
3. The MCS can be programmed to generate an error, when the comparison of signal values (duty time, cycle time) fails or also when the cycle time of the ARU (checking of the TBU_TS0 between two periodic accesses) is out of the expected range.

38.26.8.2 MON_ACTIVITY_0

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00184_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCA_3_7	MCA_3_6	MCA_3_5	MCA_3_4	MCA_3_3	MCA_3_2	MCA_3_1	MCA_3_0	MCA_2_7	MCA_2_6	MCA_2_5	MCA_2_4	MCA_2_3	MCA_2_2	MCA_2_1	MCA_2_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCA_1_7	MCA_1_6	MCA_1_5	MCA_1_4	MCA_1_3	MCA_1_2	MCA_1_1	MCA_1_0	MCA_0_7	MCA_0_6	MCA_0_5	MCA_0_4	MCA_0_3	MCA_0_2	MCA_0_1	MCA_0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.501 MON_ACTIVITY_0 Register Contents (1/5)

Bit Position	Bit Name	Function
31	MCA_3_7	Activity of check performed in module MCS 3 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
30	MCA_3_6	Activity of check performed in module MCS 3 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
29	MCA_3_5	Activity of check performed in module MCS 3 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
28	MCA_3_4	Activity of check performed in module MCS 3 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
27	MCA_3_3	Activity of check performed in module MCS 3 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.501 MON_ACTIVITY_0 Register Contents (2/5)

Bit Position	Bit Name	Function
26	MCA_3_2	Activity of check performed in module MCS 3 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
25	MCA_3_1	Activity of check performed in module MCS 3 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24	MCA_3_0	Activity of check performed in module MCS 3 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
23	MCA_2_7	Activity of check performed in module MCS 2 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
22	MCA_2_6	Activity of check performed in module MCS 2 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
21	MCA_2_5	Activity of check performed in module MCS 2 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
20	MCA_2_4	Activity of check performed in module MCS 2 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
19	MCA_2_3	Activity of check performed in module MCS 2 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.501 MON_ACTIVITY_0 Register Contents (3/5)

Bit Position	Bit Name	Function
18	MCA_2_2	Activity of check performed in module MCS 2 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
17	MCA_2_1	Activity of check performed in module MCS 2 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
16	MCA_2_0	Activity of check performed in module MCS 2 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
15	MCA_1_7	Activity of check performed in module MCS 1 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
14	MCA_1_6	Activity of check performed in module MCS 1 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
13	MCA_1_5	Activity of check performed in module MCS 1 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
12	MCA_1_4	Activity of check performed in module MCS 1 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
11	MCA_1_3	Activity of check performed in module MCS 1 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.501 MON_ACTIVITY_0 Register Contents (4/5)

Bit Position	Bit Name	Function
10	MCA_1_2	Activity of check performed in module MCS 1 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
9	MCA_1_1	Activity of check performed in module MCS 1 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
8	MCA_1_0	Activity of check performed in module MCS 1 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
7	MCA_0_7	Activity of check performed in module MCS 0 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
6	MCA_0_6	Activity of check performed in module MCS 0 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
5	MCA_0_5	Activity of check performed in module MCS 0 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
4	MCA_0_4	Activity of check performed in module MCS 0 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
3	MCA_0_3	Activity of check performed in module MCS 0 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.501 MON_ACTIVITY_0 Register Contents (5/5)

Bit Position	Bit Name	Function
2	MCA_0_2	Activity of check performed in module MCS 0 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
1	MCA_0_1	Activity of check performed in module MCS 0 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	MCA_0_0	Activity of check performed in module MCS 0 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Note: When not all MCS modules are implemented or the channels are not used for check purposes with supervising, the corresponding activity bits remain zero.

38.26.8.3 MON_ACTIVITY_1

Access: This register can be read or written in 32-bit units.

Address: <GTM_base> + 00188_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MCA_7_7	MCA_7_6	MCA_7_5	MCA_7_4	MCA_7_3	MCA_7_2	MCA_7_1	MCA_7_0	MCA_6_7	MCA_6_6	MCA_6_5	MCA_6_4	MCA_6_3	MCA_6_2	MCA_6_1	MCA_6_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCA_5_7	MCA_5_6	MCA_5_5	MCA_5_4	MCA_5_3	MCA_5_2	MCA_5_1	MCA_5_0	MCA_4_7	MCA_4_6	MCA_4_5	MCA_4_4	MCA_4_3	MCA_4_2	MCA_4_1	MCA_4_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.502 MON_ACTIVITY_1 Register Contents (1/5)

Bit Position	Bit Name	Function
31	MCA_7_7	Activity of check performed in module MCS 7 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
30	MCA_7_6	Activity of check performed in module MCS 7 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
29	MCA_7_5	Activity of check performed in module MCS 7 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
28	MCA_7_4	Activity of check performed in module MCS 7 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
27	MCA_7_3	Activity of check performed in module MCS 7 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.502 MON_ACTIVITY_1 Register Contents (2/5)

Bit Position	Bit Name	Function
26	MCA_7_2	Activity of check performed in module MCS 7 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
25	MCA_7_1	Activity of check performed in module MCS 7 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
24	MCA_7_0	Activity of check performed in module MCS 7 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
23	MCA_6_7	Activity of check performed in module MCS 6 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
22	MCA_6_6	Activity of check performed in module MCS 6 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
21	MCA_6_5	Activity of check performed in module MCS 6 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
20	MCA_6_4	Activity of check performed in module MCS 6 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
19	MCA_6_3	Activity of check performed in module MCS 6 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.502 MON_ACTIVITY_1 Register Contents (3/5)

Bit Position	Bit Name	Function
18	MCA_6_2	Activity of check performed in module MCS 6 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
17	MCA_6_1	Activity of check performed in module MCS 6 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
16	MCA_6_0	Activity of check performed in module MCS 6 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
15	MCA_5_7	Activity of check performed in module MCS 5 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
14	MCA_5_6	Activity of check performed in module MCS 5 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
13	MCA_5_5	Activity of check performed in module MCS 5 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
12	MCA_5_4	Activity of check performed in module MCS 5 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
11	MCA_5_3	Activity of check performed in module MCS 5 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.502 MON_ACTIVITY_1 Register Contents (4/5)

Bit Position	Bit Name	Function
10	MCA_5_2	Activity of check performed in module MCS 5 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
9	MCA_5_1	Activity of check performed in module MCS 5 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
8	MCA_5_0	Activity of check performed in module MCS 5 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
7	MCA_4_7	Activity of check performed in module MCS 4 at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
6	MCA_4_6	Activity of check performed in module MCS 4 at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
5	MCA_4_5	Activity of check performed in module MCS 4 at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
4	MCA_4_4	Activity of check performed in module MCS 4 at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
3	MCA_4_3	Activity of check performed in module MCS 4 at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.502 MON_ACTIVITY_1 Register Contents (5/5)

Bit Position	Bit Name	Function
2	MCA_4_2	Activity of check performed in module MCS 4 at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
1	MCA_4_1	Activity of check performed in module MCS 4 at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	MCA_4_0	Activity of check performed in module MCS 4 at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Note: When not all MCS modules are implemented or the channels are not used for check purposes with supervising, the corresponding activity bits remain zero.

38.26.8.4 MON_ACTIVITY_MCS[z]

Access: This register can be read or written in 32-bit units.

Address: MON_ACTIVITY_MCS0: <GTM_base> + $0018C_H$
MON_ACTIVITY_MCS1: <GTM_base> + 00190_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—								MCA_7	MCA_6	MCA_5	MCA_4	MCA_3	MCA_2	MCA_1	MCA_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 38.503 MON_ACTIVITY_MCS[z] Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MCA_7	Activity of check performed in module MCS z at channel 7. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
6	MCA_6	Activity of check performed in module MCS z at channel 6. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
5	MCA_5	Activity of check performed in module MCS z at channel 5. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
4	MCA_4	Activity of check performed in module MCS z at channel 4. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
3	MCA_3	Activity of check performed in module MCS z at channel 3. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Table 38.503 MON_ACTIVITY_MCS[z] Register Contents (2/2)

Bit Position	Bit Name	Function
2	MCA_2	Activity of check performed in module MCS z at channel 2. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
1	MCA_1	Activity of check performed in module MCS z at channel 1. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.
0	MCA_0	Activity of check performed in module MCS z at channel 0. **NOTE** This bit will be cleared on a CPU write access of value '1'. A read access leaves the bit unchanged.

Note 1. Unused MCA bits are reserved.

Note 2. Read as zero should be written as zero.

38.27 Appendix A

Table 38.504 Register Bit Attributes

Mode	Description
R	Read access
W	Write access
Cr	Clear on read access
Sr	Set on read access
Cw	Clear by write 1 (clears only those bits with value 1)
Sw	Set by write 1 (sets only those bits with value 1)
Aw	Auto clear after write (e.g. trigger something)
Pw	Protected write (separate write enable bit, e.g. init)

Below the bit name in a register table, the attributes “Access Mode” and “Reset Value” of each bit are described with the syntax above.

NOTE

When using Cw or Sw for a bit field e.g. representing a number, a clear/set has to be applied to all bits of the data field, to avoid construction of unintended values different to $00..00_B$ and $11..11_B$.

Table 38.505 Register Reset Value

Mode	Description
0	logic value is 0 after reset
1	logic value is 1 after reset
U	logic value is undefined after reset (0/1, e.g. monitor of external pin)

38.27.1 ARU Write Address Overview

The ARU write address map is specified in **Section 38.28, GTM Device 358**.

38.27.2 GTM Configuration Register Address Map

The addresses of the implemented sub-modules are specified in **Section 38.28, GTM Device 358**.

The start and end address of the configured rams are specified in **Section 38.28, GTM Device 358**.

The full address map of all implemented registers, the start and end addresses of configured rams are recorded in **Section 38.28, GTM Device 358**.

38.27.3 GTM Application Constraints

The ARU write address map is specified in **Section 38.28, GTM Device 358**.

38.27.4 GTM Internal Functional Dependencies

Table 38.506 GTM Internal Functional Dependencies (part 1) (Signal paths between GTM-IP modules)

	to:							
from:	**map**	**mcs**	**mon**	**psm**	**spe**	**tbu**	**tim**	**tom**
arch								
aru		53 bit data		53 bit data			53 bit data	
atom								
brc								
ccm							CMUCLK[x]	CMUFXCLK[x] TIM[i]_EXT_CAPTURE
cmp			CMP_ERR					
cmu			CMU_CLK[x], CMU_FXCLK[x]			CMU_CLK[x]		
dpll						SUB_INC1c, SUB_INC2c,...		
dtm							DTM[i]_OUT DTM[i]_OUT_N	
icm								
map								
mcs			MCS[i]_ERR, MCS[i]_CH[x]_MCA					
mon								
psm								
spe	SPE[i]_OUT, SPE[i]_NIPD, ...							SPE[i]_OUT, SPE[i]NIPD
tbu		TBU_TS[x]					TBU_TS[x]	TBU_TS[x]
tim	TIM0_CH0(48:0)	TIM[i]_EXT_CAPTURE			TIM[i]_CH[x](48)			
tom					TOM[i]_CH_TRIG_CCU0, TOM[i]_CH_TRIG_CCU1, TOM[i]_CH[x]_SOUR,..			

**Table 38.507 GTM Internal Functional Dependencies (part 2)
(Signal paths between GTM-IP modules)**

	to:									
from:	**arch**	**aru**	**atom**	**brc**	**ccm**	**cmp**	**cmu**	**dpll**	**dtm**	**icm**
arch										IRQ + EIRQ signals
aru			53 bit data	53 bit data				53 bit data		IRQ + EIRQ signals
atom		53 bit data							ATOM[i]_OUT ATOM[i]_OUT_T	IRQ + EIRQ signals
brc		53 bit data								IRQ+ EIRQ signals
ccm			CMU_CLK[x] TIM[i]_EXT_CAPTURE				CCMO_CMU_CLK6	CMU_CLKO		
cmp										IRQ + EIRQ signals
cmu		CARD reset	CMU_FXCLK[x] CMU_CLK[x]		CMU_FXCLK[x] CMU_CLK[x]			CMU_CLK0		
dpll		53 bit data					SUB_INC1, SUB_INC2,...			IRQ + EIRQ signals
dtm						DTM[i]_OUT				
icm										
map								TRIGGER, STATE, T_DIR, S_DIR, ..		
mcs		53 bit data								IRQ + EIRQ signals
mon										
psm		53 bit data								IRQ + EIRQ signals
spe										IRQ + EIRQ signals
tbu			TBU_TS[x]					TBU_TS[x], LOW_RES,TS_CLK		
tim		53 bit data			TIM[i]_CH[x](48)				TIM_FOUT	IRQ + EIRQ signals
tom									TOM[i]_OUT TOM[i]_OUT_T	IRQ + EIRQ signals

38.27.5 Compatibility Notes

38.27.5.1 DPLL

The following features of DPLL have changed since GTM v3.1.0 release:

1. In case of TORI/SORI the DPLL internal pointer handling is continued and the inc_cnt is not frozen.
2. The acceptance of input signals by configuration of DPLL_CTRL_1.TSL/SSL is extended such that after enabling the by setting DPLL by DPLL_CTRL_1.DEN=1 while DPLL_STATUS.FTD/FSD still '0' the first input signal is treated as level sensitive. In this case the signal level only is used to decide if an input signal is treated as active input signal or not.

38.27.5.2 MCS

Since GTM v2.x following features of MCS have changed:

1. MCS[i]_CTRL was replaced by MCS[i]_CTRL_STAT: MCS[i]_CTRL is obsolete. MCS[i]_CTRL and MCS[i]_CTRL_STAT have different addresses.
2. MCS[i]_RST was replaced by MCS[i]_RESET: MCS[i]_RST is obsolete. MCS[i]_RST and MCS[i]_RESET have different addresses.
3. The new register MCS[i]_CAT contains bits of obsolete register MCS[i]_CTRL
4. The new register MCS[i]_CWT contains bits of obsolete register MCS[i]_CTRL
5. The instruction execution time in accelerated mode compared to a GTM v1.x may be faster due to instruction pre-fetching.
6. On mode switch to SCD_MODE = 0b01 - Round Robin Scheduling - the value of SCD_CH has to be set to appropriate number of tasks. The reset value 0000_b for SCD_CH means that only task 0 is scheduled.

Since GTM v3.1.x following features of MCS have changed:

7. The instruction execution time in accelerated mode compared to a GTM v2.x and v1.x may be different due to increased pipeline depth.

38.28 GTM Device 358

38.28.1 GTM Device 358 Configuration

The **Table 38.508** and **Table 38.509** list the device configuration of device GTM-IP_358.

Table 38.508 GTM-IP_358 Configuration part 1

	Instances	Numbering	Channel
CCM	4	0,1,2,3 fast cluster	
TIM	4	0,1,2,3	All 8
TOM	0	—	—
ATOM	4	0,1,2,3	All 8
ADC	4	0,1,2,3	All 32
MCS	4	0,1,2,3	All 8
MCFG	1	—	—
MAP	0	—	—
SPE	0	—	—
DPLL	0	—	—
TBU	1	—	3
CMU	1	—	—
ARU	1	—	—
PSM	0	—	—
BRC	0	—	—
CMP	1	—	—
MON	1	—	—
ICM	1	—	—

Table 38.509 GTM-IP_358 configuration part 2

	Instances	CDTM Instance Numbering	DTM instances	DTM Instance Numbering
CDTM	4	0,1,2,3	2 DTM each	DTM 4,5

Table 38.510 shows defined constants referenced in GTM-IP_358 3.5.0.2 specification.

Table 38.510 GTM-IP_358 configuration constants

Module	Constant	Value
CCM	NARP	10
CCM	AAW	15
MCFG	MAW	11
MCFG	ERM	0
DPLL	NOAC	32

38.28.2 Reset values of GTM Register

The GTM specification is device independently and some register reset values are undefined and marked with ‘x’.

Reset values of the GTM register are either device depended hardcoded or defined by hardware configuration of silicon vendor.

38.28.2.1 Register reset values device depended hardcoded

Following reset values are device depended hardcoded.

NOTE

The value of GTM_REV changes with every GTM-IP_358 release.

Table 38.511 Device depended hardcoded reset values

Register Name	Reset value
GTM_REV	3583 $50A2_H$
ARU_CADDR_END	0000 0050_H

38.28.2.2 Register reset values defined by hardware configuration of silicon vendor

Following reset values depend on the hardware configuration chosen by silicon vendor.

Table 38.512 hardware configured reset values by silicon vendor

Register Name	Reset value
GTM_BRIDGE_MODE	0400 0000_H
GTM_BRIDGE_PTR1	004X $XXXX_H$
CCM[i]_HW_CONF (i: 0..3)	204F 0039_H
CCM[i]_ATOM_OUT (i = 0,1,2)	FFFF $FFFF_H$
CCM[i]_ATOM_OUT (i = 3)	0000 $FFFF_H$
ATOM[i]_CH[x]_CTRL (x: 0...7)(i: 0…3)	0000 0000_H
ATOM[i]_CH[x]_STAT (x: 0...7)(i: 0…3)	0000 0001_H
MCS[i]_CTRL_STAT (i: 0…3)	0001 0000_H
GTM_IRQ_MODE	0000 0002_H
CMP_IRQ_MODE	0000 0002_H
ARU_IRQ_MODE	0000 0002_H
TIM[i]_CH[x]_IRQ_MODE (x: 0...7)(i: 0…3)	0000 0002_H
ATOM[i]_CH[x]_IRQ_MODE (x: 0...7)(i: 0…3)	0000 0002_H
MCS[i]_CH[x]_IRQ_MODE (x: 0...7)(i: 0…3)	0000 0002_H

38.28.3 AEI write status 10_B to GTM Register under special conditions

Write access to addresses in **Table 38.513** returns status 10_B under special conditions.

Table 38.513 Register which could return status 10_B on AEI write (1/2)

Register name	RegBitModeWrite	unProtected by
ARU ACCESS	Pw	RREQ and WREQ bits are zero
ARU_IRQ_FORCINT	Aw	RF_PROT = 0
ARU_CADDR_END	W	RF_PROT = 0
ATOM[i]_CH[x]_CTRL,11	W	Channel and its output are disabled
ATOM[i]_CH[x]_CTRL,20	W	OSM = 0
ATOM[i]_CH[x]_CTRL,21	Pw	OSM = 1 and bit RST_CCU0 = 0
ATOM[i]_CH[x]_CM0	W	SOMC:serve last compare strategy CCU1 match occurred
ATOM[i]_CH[x]_CM1	Pw	SOMC:serve last compare strategy CCU1 match occurred
ATOM[i]_CH[x]_RDADDR	Pw	Channel is disabled
ATOM[i]_CH[x]_IRQ_FORCINT	Aw	RF_PROT = 0
BRC_IRQ_FORCINT	Aw	RF_PROT = 0
BRC_SRC_[z]_ADDR (z=0..11),8:0	Pw	Channel is disabled
BRC_SRC_[z]_ADDR (z=0..11),12	W	Channel is disabled
CCM[i]_ARP[z]_CTRL (z: 0..NARP-1)	Pw	CLS_PROT = 0
CCM[i]_ARP[z]_PROT (z: 0..NARP-1)	Pw	CLS_PROT = 0
CCM[i]_CFG	Pw	CLS_PROT = 0
CCM[i]_CMU_CLK_CFG	Pw	CLS_PROT = 0
CCM[i]_CMU_FXCLK_CFG	Pw	CLS_PROT = 0
CMP_IRQ_FORCINT	Aw	RF_PROT = 0
CMU_CLK_CTRL,7:0	Pw	EN_ECLK1 and separate EN_CLK are disabled
CMU_CLK_CTRL,8	Pw	EN_ECLK0 is disabled
CMU_CLK_[x]_CTRL (x=0..7)	Pw	EN_CLK[x] is disabled
CMU_CLK_[x]_CTRL (x=5,6),25:24	Pw	Value = 00_B or 11_B
CMU_ECLK_NUM	Pw	EN_ECLK[z] is disabled
CMU_ECLK_DEN	Pw	EN_ECLK[z] is disabled
CMU_GCLK_NUM	Pw	All EN_CLK[x] and the EN_FXCLK are disabled
CMU_GCLK_DEN	Pw	All EN_CLK[x] and the EN_FXCLK are disabled
CMU_GLB_CTRL	Pw	RF_PROT = 0
GTM_CLS_CLK_CFG	Pw	RF_PROT = 0
GTM_IRQ_FORCINT	Aw	RF_PROT = 0
GTM_RST	Aw	RF_PROT = 0
MCFG_CTRL	Pw	RF_PROT = 0
MCS[i]_CH[x]_CTRL,0	Pw	After RAM reset
MCS[i]_CH[x]_PC	Pw	channel is disabled
MCS[i]_CH[x]_IRQ_FORCINT	Aw	RF_PROT = 0
MCS[i]_CTRL_STAT,16	Pw	RF_PROT = 0 and all channels disabled
MCS[i]_REG_PROT	Pw	RF_PROT = 0
SPE[i]_IRQ_FORCINT	Aw	RF_PROT = 0
TBU_CH[y]_CTRL (y:0..2)	Pw	Channel is disabled

Table 38.513 Register which could return status 10_B on AEI write (2/2)

Register name	RegBitModeWrite	unProtected by
TBU_CH[y]_BASE (y:0..2)	Pw	Channel is disabled
TIM[i]_CH[x]_GPR1,23:0	Pw	Channel mode TGPS or TSSM
TIM[i]_CH[x]_CNTS,23:0	Pw	Channel mode TIPM, TBCM, TGPS or TSSM
TIM[i]_CH[x]_IRQ_FORCINT	Aw	RF_PROT = 0
TIM[i]_CH[x]_TDUC (x:0..m-1)	Pw	TOCTRL = 00
TIM[i]_CH[x]_ECTRL (x:0..m-1),3-0	W	Value < 1111_B
MCS RAM	-	After initialization
Several register	-	Do not write 1_B on not implemented bits

38.28.4 GTM Sub-module Base Addresses

The following **Table 38.514** lists all sub-modules of the GTM-IP_358 with its corresponding base addresses.

Table 38.514 Sub-module Base Addresses

Sub-module	Base address
GTM Global	0000 0000$_H$
BRIDGE	0000 0030$_H$
TBU	0000 0100$_H$
MON	0000 0180$_H$
CMP	0000 0200$_H$
ARU	0000 0280$_H$
CMU	0000 0300$_H$
ICM	0000 0600$_H$
MCFG	0000 0F40$_H$
TIM0	0000 1000$_H$
TIM1	0000 1800$_H$
TIM2	0000 2000$_H$
TIM3	0000 2800$_H$
MCS0 RAM	0003 8000$_H$
MCS1 RAM	0004 0000$_H$
MCS2 RAM	0004 8000$_H$
MCS3 RAM	0005 0000$_H$
CCM0	000E 2000$_H$
CCM1	000E 2200$_H$
CCM2	000E 2400$_H$
CCM3	000E 2600$_H$
CDTM0_DTM4	000E 4100$_H$
CDTM0_DTM5	000E 4140$_H$
CDTM1_DTM4	000E 4500$_H$
CDTM1_DTM5	000E 4540$_H$
CDTM2_DTM4	000E 4900$_H$
CDTM2_DTM5	000E 4940$_H$
CDTM3_DTM4	000E 4D00$_H$
CDTM3_DTM5	000E 4D40$_H$
ATOM0	000E 8000$_H$
ATOM1	000E 8800$_H$
ATOM2	000E 9000$_H$
ATOM3	000E 9800$_H$
MCS0	000F 0000$_H$
MCS1	000F 1000$_H$
MCS2	000F 2000$_H$
MCS3	000F 3000$_H$

38.28.5 GTM Sub-module Base Addresses Mapping Legacy Base Addresses

The following **Table 38.515** lists all sub-modules of the GTM-IP_358 which have additional legacy base addresses defined by older GTM Generation.

Table 38.515 Sub-module Base Addresses Mapping Legacy Base Addresses

Sub-module	Base address	Legacy Sub-module	LegacyBase address
CDTM0_DTM4	000E 4100$_H$	DTM24	0001 3600$_H$
CDTM0_DTM5	000E 4140$_H$	DTM25	0001 3640$_H$
CDTM1_DTM4	000E 4500$_H$	DTM26	0001 3680$_H$
CDTM1_DTM5	000E 4540$_H$	DTM27	0001 36C0$_H$
CDTM2_DTM4	000E 4900$_H$	DTM28	0001 3700$_H$
CDTM2_DTM5	000E 4940$_H$	DTM29	0001 3740$_H$
CDTM3_DTM4	000E 4D00$_H$	DTM30	0001 3780$_H$
CDTM3_DTM5	000E 4D40$_H$	DTM31	0001 37C0$_H$
ATOM0	000E 8000$_H$	LEG_ATOM0	0000 D000$_H$
ATOM1	000E 8800$_H$	LEG_ATOM1	0000 D800$_H$
ATOM2	000E 9000$_H$	LEG_ATOM2	0000 E000$_H$
ATOM3	000E 9800$_H$	LEG_ATOM3	0000 E800$_H$
MCS0	000F 0000$_H$	LEG_MCS0	0003 0000$_H$
MCS1	000F 1000$_H$	LEG_MCS1	0003 1000$_H$
MCS2	000F 2000$_H$	LEG_MCS2	0003 2000$_H$
MCS3	000F 3000$_H$	LEG_MCS3	0003 3000$_H$

38.28.6 GTM Register and Memory Addresses

The following **Table 38.516** lists all register and memory addresses used by GTM-IP_358.

Table 38.516 Register and memory addresses used by GTM-IP_358 (1/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
GTM_REV	0000 0000$_H$	3583 50A2$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
GTM_RST	0000 0004$_H$	0000 0000$_H$	F7FF FFFE$_H$	0800 0001$_H$	0
GTM_CTRL	0000 0008$_H$	0000 0001$_H$	FFFF 0E0C$_H$	0000 F1F3$_H$	0
GTM_AEI_ADDR_XPT	0000 000C$_H$	0000 0000$_H$	FFE0 0000$_H$	001F FFFF$_H$	0
GTM_IRQ_NOTIFY	0000 0010$_H$	0000 0000$_H$	00FF FE00$_H$	FF00 01FF$_H$	0
GTM_IRQ_EN	0000 0014$_H$	0000 0000$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
GTM_IRQ_FORCINT	0000 0018$_H$	0000 0000$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
GTM_IRQ_MODE	0000 001C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
GTM_EIRQ_EN	0000 0020$_H$	0000 0180$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
GTM_AEI_STA_XPT	0000 002C$_H$	0000 0000$_H$	FFE0 0000$_H$	001F FFFF$_H$	0
GTM_BRIDGE_MODE	0000 0030$_H$	0400 0000$_H$	00FE ECF8$_H$	FF01 1307$_H$	---
GTM_BRIDGE_PTR1	0000 0034$_H$	0040 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
GTM_BRIDGE_PTR2	0000 0038$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	---
GTM_MCS_AEM_DIS	0000 003C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
GTM_CLS_CLK_CFG	0000 00B0$_H$	0000 00AA$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
SVM_RESERVED0	0000 00C0$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED1	0000 00C4$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED2	0000 00C8$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED3	0000 00CC$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED4	0000 00D0$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED5	0000 00D4$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED6	0000 00D8$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED7	0000 00DC$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED8	0000 00E0$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED9	0000 00E4$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED10	0000 00E8$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED11	0000 00EC$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED12	0000 00F0$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED13	0000 00F4$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED14	0000 00F8$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
SVM_RESERVED15	0000 00FC$_H$	0000 0000$_H$	FFFF FFFF$_H$	0000 0000$_H$	---
TBU_CHEN	0000 0100$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
TBU_CH0_CTRL	0000 0104$_H$	0000 0000$_H$	FFFF FFF0$_H$	0000 000F$_H$	0
TBU_CH0_BASE	0000 0108$_H$	0000 0000$_H$	F800 0000$_H$	07FF FFFF$_H$	0
TBU_CH1_CTRL	0000 010C$_H$	0000 0000$_H$	FFFF FFF0$_H$	0000 000F$_H$	0
TBU_CH1_BASE	0000 0110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TBU_CH2_CTRL	0000 0114$_H$	0000 0000$_H$	FFFF FFF0$_H$	0000 000F$_H$	0
TBU_CH2_BASE	0000 0118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MON_STATUS	0000 0180$_H$	0000 4000$_H$	C00E A000$_H$	3FF1 5FFF$_H$	1
MON_ACTIVITY_0	0000 0184$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (2/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MON_ACTIVITY_MCS0	0000 018C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MON_ACTIVITY_MCS1	0000 0190$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MON_ACTIVITY_MCS2	0000 0194$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MON_ACTIVITY_MCS3	0000 0198$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CMP_EN	0000 0200$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
CMP_IRQ_NOTIFY	0000 0204$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
CMP_IRQ_EN	0000 0208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
CMP_IRQ_FORCINT	0000 020C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
CMP_IRQ_MODE	0000 0210$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
CMP_EIRQ_EN	0000 0214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ARU_ACCESS	0000 0280$_H$	0000 01FE$_H$	FFFF CE00$_H$	0000 31FF$_H$	0
ARU_DATA_H	0000 0284$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_DATA_L	0000 0288$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_DBG_ACCESS0	0000 028C$_H$	0000 01FE$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
ARU_DBG_DATA0_H	0000 0290$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_DBG_DATA0_L	0000 0294$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_DBG_ACCESS1	0000 0298$_H$	0000 01FE$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
ARU_DBG_DATA1_H	0000 029C$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_DBG_DATA1_L	0000 02A0$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_IRQ_NOTIFY	0000 02A4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
ARU_IRQ_EN	0000 02A8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
ARU_IRQ_FORCINT	0000 02AC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
ARU_IRQ_MODE	0000 02B0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ARU_CADDR_END	0000 02B4$_H$	0000 0050$_H$	FFFF FF80$_H$	0000 007F$_H$	0
ARU_CTRL	0000 02BC$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
ARU_0_DYN_CTRL	0000 02C0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ARU_1_DYN_CTRL	0000 02C4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ARU_0_DYN_ROUTE_LOW	0000 02C8$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ARU_1_DYN_ROUTE_LOW	0000 02CC$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ARU_0_DYN_ROUTE_HIGH	0000 02D0$_H$	0000 0000$_H$	F000 0000$_H$	0FFF FFFF$_H$	0
ARU_1_DYN_ROUTE_HIGH	0000 02D4$_H$	0000 0000$_H$	F000 0000$_H$	0FFF FFFF$_H$	0
ARU_0_DYN_ROUTE_SR_LOW	0000 02D8$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ARU_1_DYN_ROUTE_SR_LOW	0000 02DC$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ARU_0_DYN_ROUTE_SR_HIGH	0000 02E0$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_1_DYN_ROUTE_SR_HIGH	0000 02E4$_H$	0000 0000$_H$	E000 0000$_H$	1FFF FFFF$_H$	0
ARU_0_DYN_RDADDR	0000 02E8$_H$	0000 0000$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
ARU_1_DYN_RDADDR	0000 02EC$_H$	0000 0000$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
ARU_CADDR	0000 02FC$_H$	0000 0000$_H$	FF80 FF80$_H$	007F 007F$_H$	0
CMU_CLK_EN	0000 0300$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_GCLK_NUM	0000 0304$_H$	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (3/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
CMU_GCLK_DEN	0000 0308$_H$	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_0_CTRL	0000 030C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_1_CTRL	0000 0310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_2_CTRL	0000 0314	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_3_CTRL	0000 0318	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_4_CTRL	0000 031C	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_5_CTRL	0000 0320	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_CLK_6_CTRL	0000 0324	0000 0000$_H$	FC00 0000$_H$	03FF FFFF$_H$	0
CMU_CLK_7_CTRL	0000 0328	0000 0000$_H$	FC00 0000$_H$	03FF FFFF$_H$	0
CMU_ECLK_0_NUM	0000 032C	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_ECLK_0_DEN	0000 0330	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_ECLK_1_NUM	0000 0334	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_ECLK_1_DEN	0000 0338	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_ECLK_2_NUM	0000 033C	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_ECLK_2_DEN	0000 0340	0000 0001$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
CMU_GLB_CTRL	0000 0348	0000 0000$_H$	FFFF FFFE$_H$	0000 0001$_H$	0
CMU_CLK_CTRL	0000 034C	0000 0000$_H$	FFFF FE00$_H$	0000 01FF$_H$	0
ICM_IRQG_0	0000 0600	0000 0000$_H$	0000 F000$_H$	FFFF 0FFF$_H$	---
ICM_IRQG_2	0000 0608	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
ICM_IRQG_4	0000 0610	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
ICM_IRQG_9	0000 0624	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
ICM_IRQG_MEI	0000 0630	0000 0000$_H$	FC00 0000$_H$	03FF FFFF$_H$	---
ICM_IRQG_CEI1	0000 0638	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
ICM_IRQG_CEI3	0000 0640	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
ICM_IRQG_MCS0_CEI	0000 0664	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS1_CEI	0000 0668	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS2_CEI	0000 066C	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS3_CEI	0000 0670	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_CLS_0_MEI	0000 0710	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	---
ICM_IRQG_MCS0_CI	0000 0720	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS1_CI	0000 0724	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS2_CI	0000 0728	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_MCS3_CI	0000 072C	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	---
ICM_IRQG_ATOM_0_CI	0000 0790	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
MCFG_CTRL	0000 0F40	0000 0000$_H$	FFF0 0000$_H$	000F FFFF$_H$	0
TIM0_CH0_GPR0	0000 1000	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH0_GPR1	0000 1004	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH0_CNT	0000 1008	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH0_ECNT	0000 100C	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
TIM0_CH0_CNTS	0000 1010	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH0_TDUC	0000 1014	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH0_TDUV	0000 1018	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	0
TIM0_CH0_FLT_RE	0000 101C	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (4/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM0_CH0_FLT_FE	0000 1020	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH0_CTRL	0000 1024	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH0_ECTRL	0000 1028	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	0
TIM0_CH0_IRQ_NOTIFY	0000 102C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH0_IRQ_EN	0000 1030$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH0_IRQ_FORCINT	0000 1034$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH0_IRQ_MODE	0000 1038$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
TIM0_CH0_EIRQ_EN	0000 103C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_INP_VAL	0000 1074$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_IN_SRC	0000 1078$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_RST	0000 107C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
TIM0_CH1_GPR0	0000 1080$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH1_GPR1	0000 1084$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH1_CNT	0000 1088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH1_ECNT	0000 108C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
TIM0_CH1_CNTS	0000 1090$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH1_TDUC	0000 1094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH1_TDUV	0000 1098$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	0
TIM0_CH1_FLT_RE	0000 109C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH1_FLT_FE	0000 10A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH1_CTRL	0000 10A4$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH1_ECTRL	0000 10A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	0
TIM0_CH1_IRQ_NOTIFY	0000 10AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH1_IRQ_EN	0000 10B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH1_IRQ_FORCINT	0000 10B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH1_IRQ_MODE	0000 10B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
TIM0_CH1_EIRQ_EN	0000 10BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH2_GPR0	0000 1100$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH2_GPR1	0000 1104$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH2_CNT	0000 1108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH2_ECNT	0000 110C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
TIM0_CH2_CNTS	0000 1110$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH2_TDUC	0000 1114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH2_TDUV	0000 1118$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	0
TIM0_CH2_FLT_RE	0000 111C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH2_FLT_FE	0000 1120$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH2_CTRL	0000 1124$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH2_ECTRL	0000 1128$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	0
TIM0_CH2_IRQ_NOTIFY	0000 112C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH2_IRQ_EN	0000 1130$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH2_IRQ_FORCINT	0000 1134$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH2_IRQ_MODE	0000 1138$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
TIM0_CH2_EIRQ_EN	0000 113C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (5/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM0_CH3_GPR0	0000 1180$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH3_GPR1	0000 1184$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH3_CNT	0000 1188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH3_ECNT	0000 118C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	---
TIM0_CH3_CNTS	0000 1190$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH3_TDUC	0000 1194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH3_TDUV	0000 1198$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	---
TIM0_CH3_FLT_RE	0000 119C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH3_FLT_FE	0000 11A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH3_CTRL	0000 11A4$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH3_ECTRL	0000 11A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	---
TIM0_CH3_IRQ_NOTIFY	0000 11AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH3_IRQ_EN	0000 11B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH3_IRQ_FORCINT	0000 11B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH3_IRQ_MODE	0000 11B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	---
TIM0_CH3_EIRQ_EN	0000 11BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH4_GPR0	0000 1200$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH4_GPR1	0000 1204$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH4_CNT	0000 1208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH4_ECNT	0000 120C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	---
TIM0_CH4_CNTS	0000 1210$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH4_TDUC	0000 1214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH4_TDUV	0000 1218$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	---
TIM0_CH4_FLT_RE	0000 121C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH4_FLT_FE	0000 1220$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH4_CTRL	0000 1224$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH4_ECTRL	0000 1228$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	---
TIM0_CH4_IRQ_NOTIFY	0000 122C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH4_IRQ_EN	0000 1230$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH4_IRQ_FORCINT	0000 1234$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH4_IRQ_MODE	0000 1238$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	---
TIM0_CH4_EIRQ_EN	0000 123C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH5_GPR0	0000 1280$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH5_GPR1	0000 1284$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH5_CNT	0000 1288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH5_ECNT	0000 128C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	---
TIM0_CH5_CNTS	0000 1290$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH5_TDUC	0000 1294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH5_TDUV	0000 1298$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	---
TIM0_CH5_FLT_RE	0000 129C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH5_FLT_FE	0000 12A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH5_CTRL	0000 12A4$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH5_ECTRL	0000 12A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	---

Table 38.516 Register and memory addresses used by GTM-IP_358 (6/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM0_CH5_IRQ_NOTIFY	0000 12AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH5_IRQ_EN	0000 12B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH5_IRQ_FORCINT	0000 12B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH5_IRQ_MODE	0000 12B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	---
TIM0_CH5_EIRQ_EN	0000 12BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	---
TIM0_CH6_GPR0	0000 1300$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH6_GPR1	0000 1304$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	---
TIM0_CH6_CNT	0000 1308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	---
TIM0_CH6_ECNT	0000 130C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
TIM0_CH6_CNTS	0000 1310$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH6_TDUC	0000 1314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH6_TDUV	0000 1318$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	0
TIM0_CH6_FLT_RE	0000 131C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH6_FLT_FE	0000 1320$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH6_CTRL	0000 1324$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH6_ECTRL	0000 1328$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	0
TIM0_CH6_IRQ_NOTIFY	0000 132C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH6_IRQ_EN	0000 1330$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH6_IRQ_FORCINT	0000 1334$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH6_IRQ_MODE	0000 1338$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
TIM0_CH6_EIRQ_EN	0000 133C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH7_GPR0	0000 1380$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH7_GPR1	0000 1384$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH7_CNT	0000 1388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH7_ECNT	0000 138C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
TIM0_CH7_CNTS	0000 1390$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH7_TDUC	0000 1394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH7_TDUV	0000 1398$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	0
TIM0_CH7_FLT_RE	0000 139C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH7_FLT_FE	0000 13A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
TIM0_CH7_CTRL	0000 13A4$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
TIM0_CH7_ECTRL	0000 13A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	0
TIM0_CH7_IRQ_NOTIFY	0000 13AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH7_IRQ_EN	0000 13B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH7_IRQ_FORCINT	0000 13B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM0_CH7_IRQ_MODE	0000 13B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
TIM0_CH7_EIRQ_EN	0000 13BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	0
TIM1_CH0_GPR0	0000 1800$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH0_GPR1	0000 1804$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH0_CNT	0000 1808$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH0_ECNT	0000 180C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH0_CNTS	0000 1810$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH0_TDUC	0000 1814$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (7/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM1_CH0_TDUV	0000 1818$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH0_FLT_RE	0000 181C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH0_FLT_FE	0000 1820$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH0_CTRL	0000 1824$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH0_ECTRL	0000 1828$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH0_IRQ_NOTIFY	0000 182C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH0_IRQ_EN	0000 1830$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH0_IRQ_FORCINT	0000 1834$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH0_IRQ_MODE	0000 1838$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH0_EIRQ_EN	0000 183C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_INP_VAL	0000 1874$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_IN_SRC	0000 1878$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_RST	0000 187C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
TIM1_CH1_GPR0	0000 1880$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH1_GPR1	0000 1884$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH1_CNT	0000 1888$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH1_ECNT	0000 188C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH1_CNTS	0000 1890$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH1_TDUC	0000 1894$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH1_TDUV	0000 1898$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH1_FLT_RE	0000 189C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH1_FLT_FE	0000 18A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH1_CTRL	0000 18A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH1_ECTRL	0000 18A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH1_IRQ_NOTIFY	0000 18AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH1_IRQ_EN	0000 18B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH1_IRQ_FORCINT	0000 18B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH1_IRQ_MODE	0000 18B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH1_EIRQ_EN	0000 18BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH2_GPR0	0000 1900$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH2_GPR1	0000 1904$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH2_CNT	0000 1908$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH2_ECNT	0000 190C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH2_CNTS	0000 1910$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH2_TDUC	0000 1914$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH2_TDUV	0000 1918$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH2_FLT_RE	0000 191C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH2_FLT_FE	0000 1920$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH2_CTRL	0000 1924$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH2_ECTRL	0000 1928$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH2_IRQ_NOTIFY	0000 192C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH2_IRQ_EN	0000 1930$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH2_IRQ_FORCINT	0000 1934$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (8/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM1_CH2_IRQ_MODE	0000 1938$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH2_EIRQ_EN	0000 193C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH3_GPR0	0000 1980$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH3_GPR1	0000 1984$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH3_CNT	0000 1988$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH3_ECNT	0000 198C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH3_CNTS	0000 1990$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH3_TDUC	0000 1994$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH3_TDUV	0000 1998$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH3_FLT_RE	0000 199C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH3_FLT_FE	0000 19A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH3_CTRL	0000 19A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH3_ECTRL	0000 19A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH3_IRQ_NOTIFY	0000 19AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH3_IRQ_EN	0000 19B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH3_IRQ_FORCINT	0000 19B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH3_IRQ_MODE	0000 19B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH3_EIRQ_EN	0000 19BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH4_GPR0	0000 1A00$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH4_GPR1	0000 1A04$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH4_CNT	0000 1A08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH4_ECNT	0000 1A0C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH4_CNTS	0000 1A10$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH4_TDUC	0000 1A14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH4_TDUV	0000 1A18$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH4_FLT_RE	0000 1A1C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH4_FLT_FE	0000 1A20$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH4_CTRL	0000 1A24$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH4_ECTRL	0000 1A28$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH4_IRQ_NOTIFY	0000 1A2C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH4_IRQ_EN	0000 1A30$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH4_IRQ_FORCINT	0000 1A34$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH4_IRQ_MODE	0000 1A38$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH4_EIRQ_EN	0000 1A3C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH5_GPR0	0000 1A80$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH5_GPR1	0000 1A84$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH5_CNT	0000 1A88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH5_ECNT	0000 1A8C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH5_CNTS	0000 1A90$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH5_TDUC	0000 1A94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH5_TDUV	0000 1A98$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH5_FLT_RE	0000 1A9C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH5_FLT_FE	0000 1AA0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (9/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM1_CH5_CTRL	0000 1AA4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH5_ECTRL	0000 1AA8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH5_IRQ_NOTIFY	0000 1AAC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH5_IRQ_EN	0000 1AB0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH5_IRQ_FORCINT	0000 1AB4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH5_IRQ_MODE	0000 1AB8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH5_EIRQ_EN	0000 1ABC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH6_GPR0	0000 1B00$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH6_GPR1	0000 1B04$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH6_CNT	0000 1B08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH6_ECNT	0000 1B0C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH6_CNTS	0000 1B10$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH6_TDUC	0000 1B14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH6_TDUV	0000 1B18$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH6_FLT_RE	0000 1B1C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH6_FLT_FE	0000 1B20$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH6_CTRL	0000 1B24$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH6_ECTRL	0000 1B28$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH6_IRQ_NOTIFY	0000 1B2C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH6_IRQ_EN	0000 1B30$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH6_IRQ_FORCINT	0000 1B34$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH6_IRQ_MODE	0000 1B38$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH6_EIRQ_EN	0000 1B3C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH7_GPR0	0000 1B80$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH7_GPR1	0000 1B84$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH7_CNT	0000 1B88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH7_ECNT	0000 1B8C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
TIM1_CH7_CNTS	0000 1B90$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
TIM1_CH7_TDUC	0000 1B94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH7_TDUV	0000 1B98$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	1
TIM1_CH7_FLT_RE	0000 1B9C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH7_FLT_FE	0000 1BA0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
TIM1_CH7_CTRL	0000 1BA4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	1
TIM1_CH7_ECTRL	0000 1BA8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	1
TIM1_CH7_IRQ_NOTIFY	0000 1BAC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH7_IRQ_EN	0000 1BB0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH7_IRQ_FORCINT	0000 1BB4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM1_CH7_IRQ_MODE	0000 1BB8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
TIM1_CH7_EIRQ_EN	0000 1BBC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	1
TIM2_CH0_GPR0	0000 2000$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH0_GPR1	0000 2004$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH0_CNT	0000 2008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH0_ECNT	0000 200C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (10/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM2_CH0_CNTS	0000 2010$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH0_TDUC	0000 2014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH0_TDUV	0000 2018$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH0_FLT_RE	0000 201C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH0_FLT_FE	0000 2020$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH0_CTRL	0000 2024$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH0_ECTRL	0000 2028$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH0_IRQ_NOTIFY	0000 202C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH0_IRQ_EN	0000 2030$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH0_IRQ_FORCINT	0000 2034$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH0_IRQ_MODE	0000 2038$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH0_EIRQ_EN	0000 203C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_INP_VAL	0000 2074$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_IN_SRC	0000 2078$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_RST	0000 207C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
TIM2_CH1_GPR0	0000 2080$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH1_GPR1	0000 2084$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH1_CNT	0000 2088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH1_ECNT	0000 208C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH1_CNTS	0000 2090$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH1_TDUC	0000 2094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH1_TDUV	0000 2098$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH1_FLT_RE	0000 209C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH1_FLT_FE	0000 20A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH1_CTRL	0000 20A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH1_ECTRL	0000 20A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH1_IRQ_NOTIFY	0000 20AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH1_IRQ_EN	0000 20B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH1_IRQ_FORCINT	0000 20B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH1_IRQ_MODE	0000 20B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH1_EIRQ_EN	0000 20BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH2_GPR0	0000 2100$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH2_GPR1	0000 2104$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH2_CNT	0000 2108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH2_ECNT	0000 210C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH2_CNTS	0000 2110$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH2_TDUC	0000 2114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH2_TDUV	0000 2118$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH2_FLT_RE	0000 211C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH2_FLT_FE	0000 2120$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH2_CTRL	0000 2124$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH2_ECTRL	0000 2128$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH2_IRQ_NOTIFY	0000 212C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (11/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM2_CH2_IRQ_EN	0000 2130$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH2_IRQ_FORCINT	0000 2134$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH2_IRQ_MODE	0000 2138$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH2_EIRQ_EN	0000 213C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH3_GPR0	0000 2180$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH3_GPR1	0000 2184$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH3_CNT	0000 2188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH3_ECNT	0000 218C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH3_CNTS	0000 2190$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH3_TDUC	0000 2194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH3_TDUV	0000 2198$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH3_FLT_RE	0000 219C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH3_FLT_FE	0000 21A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH3_CTRL	0000 21A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH3_ECTRL	0000 21A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH3_IRQ_NOTIFY	0000 21AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH3_IRQ_EN	0000 21B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH3_IRQ_FORCINT	0000 21B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH3_IRQ_MODE	0000 21B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH3_EIRQ_EN	0000 21BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH4_GPR0	0000 2200$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH4_GPR1	0000 2204$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH4_CNT	0000 2208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH4_ECNT	0000 220C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH4_CNTS	0000 2210$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH4_TDUC	0000 2214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH4_TDUV	0000 2218$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH4_FLT_RE	0000 221C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH4_FLT_FE	0000 2220$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH4_CTRL	0000 2224$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH4_ECTRL	0000 2228$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH4_IRQ_NOTIFY	0000 222C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH4_IRQ_EN	0000 2230$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH4_IRQ_FORCINT	0000 2234$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH4_IRQ_MODE	0000 2238$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH4_EIRQ_EN	0000 223C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH5_GPR0	0000 2280$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH5_GPR1	0000 2284$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH5_CNT	0000 2288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH5_ECNT	0000 228C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH5_CNTS	0000 2290$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH5_TDUC	0000 2294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH5_TDUV	0000 2298$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (12/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM2_CH5_FLT_RE	0000 229C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH5_FLT_FE	0000 22A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH5_CTRL	0000 22A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH5_ECTRL	0000 22A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH5_IRQ_NOTIFY	0000 22AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH5_IRQ_EN	0000 22B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH5_IRQ_FORCINT	0000 22B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH5_IRQ_MODE	0000 22B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH5_EIRQ_EN	0000 22BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH6_GPR0	0000 2300$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH6_GPR1	0000 2304$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH6_CNT	0000 2308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH6_ECNT	0000 230C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH6_CNTS	0000 2310$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH6_TDUC	0000 2314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH6_TDUV	0000 2318$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH6_FLT_RE	0000 231C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH6_FLT_FE	0000 2320$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH6_CTRL	0000 2324$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH6_ECTRL	0000 2328$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH6_IRQ_NOTIFY	0000 232C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH6_IRQ_EN	0000 2330$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH6_IRQ_FORCINT	0000 2334$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH6_IRQ_MODE	0000 2338$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH6_EIRQ_EN	0000 233C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH7_GPR0	0000 2380$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH7_GPR1	0000 2384$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH7_CNT	0000 2388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH7_ECNT	0000 238C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
TIM2_CH7_CNTS	0000 2390$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
TIM2_CH7_TDUC	0000 2394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH7_TDUV	0000 2398$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	2
TIM2_CH7_FLT_RE	0000 239C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH7_FLT_FE	0000 23A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
TIM2_CH7_CTRL	0000 23A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	2
TIM2_CH7_ECTRL	0000 23A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	2
TIM2_CH7_IRQ_NOTIFY	0000 23AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH7_IRQ_EN	0000 23B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH7_IRQ_FORCINT	0000 23B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM2_CH7_IRQ_MODE	0000 23B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
TIM2_CH7_EIRQ_EN	0000 23BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	2
TIM3_CH0_GPR0	0000 2800$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH0_GPR1	0000 2804$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (13/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM3_CH0_CNT	0000 2808$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH0_ECNT	0000 280C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH0_CNTS	0000 2810$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH0_TDUC	0000 2814$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH0_TDUV	0000 2818$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH0_FLT_RE	0000 281C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH0_FLT_FE	0000 2820$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH0_CTRL	0000 2824$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH0_ECTRL	0000 2828$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH0_IRQ_NOTIFY	0000 282C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH0_IRQ_EN	0000 2830$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH0_IRQ_FORCINT	0000 2834$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH0_IRQ_MODE	0000 2838$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH0_EIRQ_EN	0000 283C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_INP_VAL	0000 2874$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_IN_SRC	0000 2878$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_RST	0000 287C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
TIM3_CH1_GPR0	0000 2880$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH1_GPR1	0000 2884$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH1_CNT	0000 2888$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH1_ECNT	0000 288C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH1_CNTS	0000 2890$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH1_TDUC	0000 2894$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH1_TDUV	0000 2898$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH1_FLT_RE	0000 289C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH1_FLT_FE	0000 28A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH1_CTRL	0000 28A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH1_ECTRL	0000 28A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH1_IRQ_NOTIFY	0000 28AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH1_IRQ_EN	0000 28B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH1_IRQ_FORCINT	0000 28B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH1_IRQ_MODE	0000 28B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH1_EIRQ_EN	0000 28BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH2_GPR0	0000 2900$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH2_GPR1	0000 2904$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH2_CNT	0000 2908$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH2_ECNT	0000 290C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH2_CNTS	0000 2910$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH2_TDUC	0000 2914$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH2_TDUV	0000 2918$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH2_FLT_RE	0000 291C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH2_FLT_FE	0000 2920$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH2_CTRL	0000 2924$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (14/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM3_CH2_ECTRL	0000 2928$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH2_IRQ_NOTIFY	0000 292C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH2_IRQ_EN	0000 2930$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH2_IRQ_FORCINT	0000 2934$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH2_IRQ_MODE	0000 2938$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH2_EIRQ_EN	0000 293C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH3_GPR0	0000 2980$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH3_GPR1	0000 2984$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH3_CNT	0000 2988$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH3_ECNT	0000 298C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH3_CNTS	0000 2990$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH3_TDUC	0000 2994$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH3_TDUV	0000 2998$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH3_FLT_RE	0000 299C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH3_FLT_FE	0000 29A0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH3_CTRL	0000 29A4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH3_ECTRL	0000 29A8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH3_IRQ_NOTIFY	0000 29AC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH3_IRQ_EN	0000 29B0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH3_IRQ_FORCINT	0000 29B4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH3_IRQ_MODE	0000 29B8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH3_EIRQ_EN	0000 29BC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH4_GPR0	0000 2A00$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH4_GPR1	0000 2A04$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH4_CNT	0000 2A08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH4_ECNT	0000 2A0C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH4_CNTS	0000 2A10$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH4_TDUC	0000 2A14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH4_TDUV	0000 2A18$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH4_FLT_RE	0000 2A1C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH4_FLT_FE	0000 2A20$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH4_CTRL	0000 2A24$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH4_ECTRL	0000 2A28$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH4_IRQ_NOTIFY	0000 2A2C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH4_IRQ_EN	0000 2A30$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH4_IRQ_FORCINT	0000 2A34$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH4_IRQ_MODE	0000 2A38$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH4_EIRQ_EN	0000 2A3C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH5_GPR0	0000 2A80$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH5_GPR1	0000 2A84$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH5_CNT	0000 2A88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH5_ECNT	0000 2A8C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH5_CNTS	0000 2A90$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (15/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
TIM3_CH5_TDUC	0000 2A94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH5_TDUV	0000 2A98$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH5_FLT_RE	0000 2A9C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH5_FLT_FE	0000 2AA0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH5_CTRL	0000 2AA4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH5_ECTRL	0000 2AA8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH5_IRQ_NOTIFY	0000 2AAC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH5_IRQ_EN	0000 2AB0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH5_IRQ_FORCINT	0000 2AB4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH5_IRQ_MODE	0000 2AB8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH5_EIRQ_EN	0000 2ABC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH6_GPR0	0000 2B00$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH6_GPR1	0000 2B04$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH6_CNT	0000 2B08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH6_ECNT	0000 2B0C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH6_CNTS	0000 2B10$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH6_TDUC	0000 2B14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH6_TDUV	0000 2B18$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH6_FLT_RE	0000 2B1C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH6_FLT_FE	0000 2B20$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH6_CTRL	0000 2B24$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH6_ECTRL	0000 2B28$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH6_IRQ_NOTIFY	0000 2B2C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH6_IRQ_EN	0000 2B30$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH6_IRQ_FORCINT	0000 2B34$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH6_IRQ_MODE	0000 2B38$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH6_EIRQ_EN	0000 2B3C$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH7_GPR0	0000 2B80$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH7_GPR1	0000 2B84$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH7_CNT	0000 2B88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH7_ECNT	0000 2B8C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
TIM3_CH7_CNTS	0000 2B90$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
TIM3_CH7_TDUC	0000 2B94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH7_TDUV	0000 2B98$_H$	0000 0000$_H$	8000 0000$_H$	7FFF FFFF$_H$	3
TIM3_CH7_FLT_RE	0000 2B9C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH7_FLT_FE	0000 2BA0$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
TIM3_CH7_CTRL	0000 2BA4$_H$	0000 0000$_H$	0000 0080$_H$	FFFF FF7F$_H$	3
TIM3_CH7_ECTRL	0000 2BA8$_H$	0000 0000$_H$	0C30 8810$_H$	F3CF 77EF$_H$	3
TIM3_CH7_IRQ_NOTIFY	0000 2BAC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH7_IRQ_EN	0000 2BB0$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH7_IRQ_FORCINT	0000 2BB4$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3
TIM3_CH7_IRQ_MODE	0000 2BB8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
TIM3_CH7_EIRQ_EN	0000 2BBC$_H$	0000 0000$_H$	FFFF FFC0$_H$	0000 003F$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (16/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS0_MEM	0003 8000$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
MCS1_MEM	0004 0000$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
MCS2_MEM	0004 8000$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
MCS3_MEM	0005 0000$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CCM0_ARP0_CTRL	000E 2000$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP0_PROT	000E 2004$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP1_CTRL	000E 2008$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP1_PROT	000E 200C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP2_CTRL	000E 2010$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP2_PROT	000E 2014$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP3_CTRL	000E 2018$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP3_PROT	000E 201C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP4_CTRL	000E 2020$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP4_PROT	000E 2024$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP5_CTRL	000E 2028$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP5_PROT	000E 202C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP6_CTRL	000E 2030$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP6_PROT	000E 2034$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP7_CTRL	000E 2038$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP7_PROT	000E 203C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP8_CTRL	000E 2040$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP8_PROT	000E 2044$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_ARP9_CTRL	000E 2048$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	0
CCM0_ARP9_PROT	000E 204C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_HW_CONF2	000E 21D4$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CCM0_AEIM_STA	000E 21D8$_H$	0000 0000$_H$	FCFF 0000$_H$	0300 FFFF$_H$	0
CCM0_HW_CONF	000E 21DC$_H$	284F 1339$_H$	C000 8000$_H$	3FFF 7FFF$_H$	0
CCM0_TIM_AUX_IN_SRC	000E 21E0$_H$	0000 0000$_H$	FF00 FF00$_H$	00FF 00FF$_H$	0
CCM0_EXT_CAP_EN	000E 21E4$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
CCM0_ATOM_OUT	000E 21EC$_H$	FFFF FFFF$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
CCM0_CMU_CLK_CFG	000E 21F0$_H$	0000 0000$_H$	CCCC CCCC$_H$	3333 3333$_H$	0
CCM0_CFG	000E 21F8$_H$	0002 000D$_H$	3FFC FF00$_H$	C003 00FF$_H$	0
CCM0_PROT	000E 21FC$_H$	0000 0001$_H$	FFFF FFFE$_H$	0000 0001$_H$	0
CCM1_ARP0_CTRL	000E 2200$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP0_PROT	000E 2204$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP1_CTRL	000E 2208$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP1_PROT	000E 220C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP2_CTRL	000E 2210$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP2_PROT	000E 2214$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP3_CTRL	000E 2218$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP3_PROT	000E 221C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP4_CTRL	000E 2220$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP4_PROT	000E 2224$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (17/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
CCM1_ARP5_CTRL	000E 2228$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP5_PROT	000E 222C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP6_CTRL	000E 2230$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP6_PROT	000E 2234$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP7_CTRL	000E 2238$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP7_PROT	000E 223C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP8_CTRL	000E 2240$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP8_PROT	000E 2244$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_ARP9_CTRL	000E 2248$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	1
CCM1_ARP9_PROT	000E 224C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_HW_CONF2	000E 23D4$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CCM1_AEIM_STA	000E 23D8$_H$	0000 0000$_H$	FCFF 0000$_H$	0300 FFFF$_H$	1
CCM1_HW_CONF	000E 23DC$_H$	284F 1339$_H$	C000 8000$_H$	3FFF 7FFF$_H$	1
CCM1_TIM_AUX_IN_SRC	000E 23E0$_H$	0000 0000$_H$	FF00 FF00$_H$	00FF 00FF$_H$	1
CCM1_EXT_CAP_EN	000E 23E4$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
CCM1_ATOM_OUT	000E 23EC$_H$	FFFF FFFF$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
CCM1_CMU_CLK_CFG	000E 23F0$_H$	0000 0000$_H$	CCCC CCCC$_H$	3333 3333$_H$	1
CCM1_CFG	000E 23F8$_H$	0002 008D$_H$	3FFC FF00$_H$	C003 00FF$_H$	1
CCM1_PROT	000E 23FC$_H$	0000 0001$_H$	FFFF FFFE$_H$	0000 0001$_H$	1
CCM2_ARP0_CTRL	000E 2400$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP0_PROT	000E 2404$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP1_CTRL	000E 2408$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP1_PROT	000E 240C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP2_CTRL	000E 2410$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP2_PROT	000E 2414$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP3_CTRL	000E 2418$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP3_PROT	000E 241C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP4_CTRL	000E 2420$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP4_PROT	000E 2424$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP5_CTRL	000E 2428$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP5_PROT	000E 242C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP6_CTRL	000E 2430$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP6_PROT	000E 2434$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP7_CTRL	000E 2438$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP7_PROT	000E 243C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP8_CTRL	000E 2440$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP8_PROT	000E 2444$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_ARP9_CTRL	000E 2448$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	2
CCM2_ARP9_PROT	000E 244C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_HW_CONF2	000E 25D4$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CCM2_AEIM_STA	000E 25D8$_H$	0000 0000$_H$	FCFF 0000$_H$	0300 FFFF$_H$	2
CCM2_HW_CONF	000E 25DC$_H$	284F 1339$_H$	C000 8000$_H$	3FFF 7FFF$_H$	2
CCM2_TIM_AUX_IN_SRC	000E 25E0$_H$	0000 0000$_H$	FF00 FF00$_H$	00FF 00FF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (18/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
CCM2_EXT_CAP_EN	000E 25E4$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
CCM2_ATOM_OUT	000E 25EC$_H$	FFFF FFFF$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
CCM2_CMU_CLK_CFG	000E 25F0$_H$	0000 0000$_H$	CCCC CCCC$_H$	3333 3333$_H$	2
CCM2_CFG	000E 25F8$_H$	0002 000D$_H$	3FFC FF00$_H$	C003 00FF$_H$	2
CCM2_PROT	000E 25FC$_H$	0000 0001$_H$	FFFF FFFE$_H$	0000 0001$_H$	2
CCM3_ARP0_CTRL	000E 2600$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP0_PROT	000E 2604$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP1_CTRL	000E 2608$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP1_PROT	000E 260C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP2_CTRL	000E 2610$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP2_PROT	000E 2614$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP3_CTRL	000E 2618$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP3_PROT	000E 261C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP4_CTRL	000E 2620$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP4_PROT	000E 2624$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP5_CTRL	000E 2628$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP5_PROT	000E 262C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP6_CTRL	000E 2630$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP6_PROT	000E 2634$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP7_CTRL	000E 2638$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP7_PROT	000E 263C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP8_CTRL	000E 2640$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP8_PROT	000E 2644$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_ARP9_CTRL	000E 2648$_H$	0003 0000$_H$	7EF0 0000$_H$	810F FFFF$_H$	3
CCM3_ARP9_PROT	000E 264C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_HW_CONF2	000E 27D4$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CCM3_AEIM_STA	000E 27D8$_H$	0000 0000$_H$	FCFF 0000$_H$	0300 FFFF$_H$	3
CCM3_HW_CONF	000E 27DC$_H$	284F 1339$_H$	C000 8000$_H$	3FFF 7FFF$_H$	3
CCM3_TIM_AUX_IN_SRC	000E 27E0$_H$	0000 0000$_H$	FF00 FF00$_H$	00FF 00FF$_H$	3
CCM3_EXT_CAP_EN	000E 27E4$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
CCM3_ATOM_OUT	000E 27EC$_H$	0000 FFFF$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CCM3_CMU_CLK_CFG	000E 27F0$_H$	0000 0000$_H$	CCCC CCCC$_H$	3333 3333$_H$	3
CCM3_CFG	000E 27F8$_H$	0002 000D$_H$	3FFC FF00$_H$	C003 00FF$_H$	3
CCM3_PROT	000E 27FC$_H$	0000 0001$_H$	FFFF FFFE$_H$	0000 0001$_H$	3
CDTM0_DTM4_CTRL	000E 4100$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	0
CDTM0_DTM4_CH_CTRL1	000E 4104$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	0
CDTM0_DTM4_CH_CTRL2	000E 4108$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
CDTM0_DTM4_CH_CTRL2_SR	000E 410C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
CDTM0_DTM4_PS_CTRL	000E 4110$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	0
CDTM0_DTM4_CH0_DTV	000E 4114$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM4_CH1_DTV	000E 4118$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM4_CH2_DTV	000E 411C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM4_CH3_DTV	000E 4120$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (19/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
CDTM0_DTM4_CH_SR	000E 4124$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CDTM0_DTM4_CH_CTRL3	000E 4128$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	0
CDTM0_DTM5_CTRL	000E 4140$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	0
CDTM0_DTM5_CH_CTRL1	000E 4144$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	0
CDTM0_DTM5_CH_CTRL2	000E 4148$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
CDTM0_DTM5_CH_CTRL2_SR	000E 414C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
CDTM0_DTM5_PS_CTRL	000E 4150$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	0
CDTM0_DTM5_CH0_DTV	000E 4154$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM5_CH1_DTV	000E 4158$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM5_CH2_DTV	000E 415C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM5_CH3_DTV	000E 4160$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	0
CDTM0_DTM5_CH_SR	000E 4164$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
CDTM0_DTM5_CH_CTRL3	000E 4168$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	0
CDTM1_DTM4_CTRL	000E 4500$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	1
CDTM1_DTM4_CH_CTRL1	000E 4504$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	1
CDTM1_DTM4_CH_CTRL2	000E 4508$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
CDTM1_DTM4_CH_CTRL2_SR	000E 450C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
CDTM1_DTM4_PS_CTRL	000E 4510$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	1
CDTM1_DTM4_CH0_DTV	000E 4514$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM4_CH1_DTV	000E 4518$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM4_CH2_DTV	000E 451C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM4_CH3_DTV	000E 4520$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM4_CH_SR	000E 4524$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CDTM1_DTM4_CH_CTRL3	000E 4528$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	1
CDTM1_DTM5_CTRL	000E 4540$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	1
CDTM1_DTM5_CH_CTRL1	000E 4544$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	1
CDTM1_DTM5_CH_CTRL2	000E 4548$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
CDTM1_DTM5_CH_CTRL2_SR	000E 454C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
CDTM1_DTM5_PS_CTRL	000E 4550$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	1
CDTM1_DTM5_CH0_DTV	000E 4554$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM5_CH1_DTV	000E 4558$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM5_CH2_DTV	000E 455C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM5_CH3_DTV	000E 4560$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	1
CDTM1_DTM5_CH_SR	000E 4564$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
CDTM1_DTM5_CH_CTRL3	000E 4568$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	1
CDTM2_DTM4_CTRL	000E 4900$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	2
CDTM2_DTM4_CH_CTRL1	000E 4904$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	2
CDTM2_DTM4_CH_CTRL2	000E 4908$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
CDTM2_DTM4_CH_CTRL2_SR	000E 490C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
CDTM2_DTM4_PS_CTRL	000E 4910$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	2
CDTM2_DTM4_CH0_DTV	000E 4914$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM4_CH1_DTV	000E 4918$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM4_CH2_DTV	000E 491C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (20/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
CDTM2_DTM4_CH3_DTV	000E 4920$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM4_CH_SR	000E 4924$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CDTM2_DTM4_CH_CTRL3	000E 4928$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	2
CDTM2_DTM5_CTRL	000E 4940$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	2
CDTM2_DTM5_CH_CTRL1	000E 4944$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	2
CDTM2_DTM5_CH_CTRL2	000E 4948$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
CDTM2_DTM5_CH_CTRL2_SR	000E 494C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
CDTM2_DTM5_PS_CTRL	000E 4950$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	2
CDTM2_DTM5_CH0_DTV	000E 4954$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM5_CH1_DTV	000E 4958$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM5_CH2_DTV	000E 495C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM5_CH3_DTV	000E 4960$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	2
CDTM2_DTM5_CH_SR	000E 4964$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
CDTM2_DTM5_CH_CTRL3	000E 4968$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	2
CDTM3_DTM4_CTRL	000E 4D00$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	3
CDTM3_DTM4_CH_CTRL1	000E 4D04$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	3
CDTM3_DTM4_CH_CTRL2	000E 4D08$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CDTM3_DTM4_CH_CTRL2_SR	000E 4D0C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CDTM3_DTM4_PS_CTRL	000E 4D10$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	3
CDTM3_DTM4_CH0_DTV	000E 4D14$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM4_CH1_DTV	000E 4D18$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM4_CH2_DTV	000E 4D1C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM4_CH3_DTV	000E 4D20$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM4_CH_SR	000E 4D24$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CDTM3_DTM4_CH_CTRL3	000E 4D28$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	3
CDTM3_DTM5_CTRL	000E 4D40$_H$	0000 0000$_H$	FFFE FE80$_H$	0001 017F$_H$	3
CDTM3_DTM5_CH_CTRL1	000E 4D44$_H$	0000 0000$_H$	C080 C084$_H$	3F7F 3F7B$_H$	3
CDTM3_DTM5_CH_CTRL2	000E 4D48$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CDTM3_DTM5_CH_CTRL2_SR	000E 4D4C$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
CDTM3_DTM5_PS_CTRL	000E 4D50$_H$	0000 0000$_H$	FFC8 FC00$_H$	0037 03FF$_H$	3
CDTM3_DTM5_CH0_DTV	000E 4D54$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM5_CH1_DTV	000E 4D58$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM5_CH2_DTV	000E 4D5C$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM5_CH3_DTV	000E 4D60$_H$	0000 0000$_H$	FC00 FC00$_H$	03FF 03FF$_H$	3
CDTM3_DTM5_CH_SR	000E 4D64$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
CDTM3_DTM5_CH_CTRL3	000E 4D68$_H$	0000 0000$_H$	F0F0 F0F0$_H$	0F0F 0F0F$_H$	3
ATOM0_CH0_RDADDR	000E 8000$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH0_CTRL	000E 8004$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH0_SR0	000E 8008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH0_SR1	000E 800C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH0_CM0	000E 8010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH0_CM1	000E 8014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH0_CN0	000E 8018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (21/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM0_CH0_STAT	000E 801C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH0_IRQ_NOTIFY	000E 8020$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH0_IRQ_EN	000E 8024$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH0_IRQ_FORCINT	000E 8028$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH0_IRQ_MODE	000E 802C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_AGC_GLB_CTRL	000E 8040$_H$	0000 0000$_H$	0000 00FE$_H$	FFFF FF01$_H$	0
ATOM0_AGC_ENDIS_CTRL	000E 8044$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
ATOM0_AGC_ENDIS_STAT	000E 8048$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
ATOM0_AGC_ACT_TB	000E 804C$_H$	0000 0000$_H$	F800 0000$_H$	07FF FFFF$_H$	0
ATOM0_AGC_OUTEN_CTRL	000E 8050$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
ATOM0_AGC_OUTEN_STAT	000E 8054$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
ATOM0_AGC_FUPD_CTRL	000E 8058$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	0
ATOM0_AGC_INT_TRIG	000E 805C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
ATOM0_CH1_RDADDR	000E 8080$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH1_CTRL	000E 8084$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH1_SR0	000E 8088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH1_SR1	000E 808C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH1_CM0	000E 8090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH1_CM1	000E 8094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH1_CN0	000E 8098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH1_STAT	000E 809C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH1_IRQ_NOTIFY	000E 80A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH1_IRQ_EN	000E 80A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH1_IRQ_FORCINT	000E 80A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH1_IRQ_MODE	000E 80AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH2_RDADDR	000E 8100$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH2_CTRL	000E 8104$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH2_SR0	000E 8108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH2_SR1	000E 810C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH2_CM0	000E 8110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH2_CM1	000E 8114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH2_CN0	000E 8118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH2_STAT	000E 811C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH2_IRQ_NOTIFY	000E 8120$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH2_IRQ_EN	000E 8124$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH2_IRQ_FORCINT	000E 8128$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH2_IRQ_MODE	000E 812C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH3_RDADDR	000E 8180$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH3_CTRL	000E 8184$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH3_SR0	000E 8188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH3_SR1	000E 818C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH3_CM0	000E 8190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH3_CM1	000E 8194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (22/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM0_CH3_CN0	000E 8198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH3_STAT	000E 819C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH3_IRQ_NOTIFY	000E 81A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH3_IRQ_EN	000E 81A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH3_IRQ_FORCINT	000E 81A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH3_IRQ_MODE	000E 81AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH4_RDADDR	000E 8200$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH4_CTRL	000E 8204$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH4_SR0	000E 8208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH4_SR1	000E 820C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH4_CM0	000E 8210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH4_CM1	000E 8214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH4_CN0	000E 8218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH4_STAT	000E 821C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH4_IRQ_NOTIFY	000E 8220$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH4_IRQ_EN	000E 8224$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH4_IRQ_FORCINT	000E 8228$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH4_IRQ_MODE	000E 822C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH5_RDADDR	000E 8280$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH5_CTRL	000E 8284$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH5_SR0	000E 8288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH5_SR1	000E 828C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH5_CM0	000E 8290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH5_CM1	000E 8294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH5_CN0	000E 8298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH5_STAT	000E 829C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH5_IRQ_NOTIFY	000E 82A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH5_IRQ_EN	000E 82A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH5_IRQ_FORCINT	000E 82A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH5_IRQ_MODE	000E 82AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH6_RDADDR	000E 8300$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0
ATOM0_CH6_CTRL	000E 8304$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH6_SR0	000E 8308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH6_SR1	000E 830C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH6_CM0	000E 8310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH6_CM1	000E 8314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH6_CN0	000E 8318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH6_STAT	000E 831C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH6_IRQ_NOTIFY	000E 8320$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH6_IRQ_EN	000E 8324$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH6_IRQ_FORCINT	000E 8328$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH6_IRQ_MODE	000E 832C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH7_RDADDR	000E 8380$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (23/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM0_CH7_CTRL	000E 8384$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	0
ATOM0_CH7_SR0	000E 8388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH7_SR1	000E 838C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH7_CM0	000E 8390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH7_CM1	000E 8394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH7_CN0	000E 8398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
ATOM0_CH7_STAT	000E 839C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	0
ATOM0_CH7_IRQ_NOTIFY	000E 83A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH7_IRQ_EN	000E 83A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH7_IRQ_FORCINT	000E 83A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM0_CH7_IRQ_MODE	000E 83AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
ATOM1_CH0_RDADDR	000E 8800$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH0_CTRL	000E 8804$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH0_SR0	000E 8808$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH0_SR1	000E 880C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH0_CM0	000E 8810$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH0_CM1	000E 8814$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH0_CN0	000E 8818$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH0_STAT	000E 881C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH0_IRQ_NOTIFY	000E 8820$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH0_IRQ_EN	000E 8824$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH0_IRQ_FORCINT	000E 8828$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH0_IRQ_MODE	000E 882C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_AGC_GLB_CTRL	000E 8840$_H$	0000 0000$_H$	0000 00FE$_H$	FFFF FF01$_H$	1
ATOM1_AGC_ENDIS_CTRL	000E 8844$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
ATOM1_AGC_ENDIS_STAT	000E 8848$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
ATOM1_AGC_ACT_TB	000E 884C$_H$	0000 0000$_H$	F800 0000$_H$	07FF FFFF$_H$	1
ATOM1_AGC_OUTEN_CTRL	000E 8850$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
ATOM1_AGC_OUTEN_STAT	000E 8854$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
ATOM1_AGC_FUPD_CTRL	000E 8858$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	1
ATOM1_AGC_INT_TRIG	000E 885C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
ATOM1_CH1_RDADDR	000E 8880$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH1_CTRL	000E 8884$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH1_SR0	000E 8888$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH1_SR1	000E 888C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH1_CM0	000E 8890$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH1_CM1	000E 8894$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH1_CN0	000E 8898$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH1_STAT	000E 889C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH1_IRQ_NOTIFY	000E 88A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH1_IRQ_EN	000E 88A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH1_IRQ_FORCINT	000E 88A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH1_IRQ_MODE	000E 88AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (24/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM1_CH2_RDADDR	000E 8900$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH2_CTRL	000E 8904$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH2_SR0	000E 8908$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH2_SR1	000E 890C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH2_CM0	000E 8910$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH2_CM1	000E 8914$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH2_CN0	000E 8918$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH2_STAT	000E 891C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH2_IRQ_NOTIFY	000E 8920$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH2_IRQ_EN	000E 8924$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH2_IRQ_FORCINT	000E 8928$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH2_IRQ_MODE	000E 892C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH3_RDADDR	000E 8980$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH3_CTRL	000E 8984$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH3_SR0	000E 8988$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH3_SR1	000E 898C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH3_CM0	000E 8990$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH3_CM1	000E 8994$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH3_CN0	000E 8998$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH3_STAT	000E 899C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH3_IRQ_NOTIFY	000E 89A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH3_IRQ_EN	000E 89A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH3_IRQ_FORCINT	000E 89A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH3_IRQ_MODE	000E 89AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH4_RDADDR	000E 8A00$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH4_CTRL	000E 8A04$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH4_SR0	000E 8A08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH4_SR1	000E 8A0C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH4_CM0	000E 8A10$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH4_CM1	000E 8A14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH4_CN0	000E 8A18$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH4_STAT	000E 8A1C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH4_IRQ_NOTIFY	000E 8A20$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH4_IRQ_EN	000E 8A24$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH4_IRQ_FORCINT	000E 8A28$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH4_IRQ_MODE	000E 8A2C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH5_RDADDR	000E 8A80$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH5_CTRL	000E 8A84$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH5_SR0	000E 8A88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH5_SR1	000E 8A8C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH5_CM0	000E 8A90$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH5_CM1	000E 8A94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH5_CN0	000E 8A98$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (25/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM1_CH5_STAT	000E 8A9C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH5_IRQ_NOTIFY	000E 8AA0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH5_IRQ_EN	000E 8AA4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH5_IRQ_FORCINT	000E 8AA8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH5_IRQ_MODE	000E 8AAC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH6_RDADDR	000E 8B00$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH6_CTRL	000E 8B04$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH6_SR0	000E 8B08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH6_SR1	000E 8B0C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH6_CM0	000E 8B10$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH6_CM1	000E 8B14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH6_CN0	000E 8B18$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH6_STAT	000E 8B1C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH6_IRQ_NOTIFY	000E 8B20$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH6_IRQ_EN	000E 8B24$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH6_IRQ_FORCINT	000E 8B28$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH6_IRQ_MODE	000E 8B2C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH7_RDADDR	000E 8B80$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	1
ATOM1_CH7_CTRL	000E 8B84$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	1
ATOM1_CH7_SR0	000E 8B88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH7_SR1	000E 8B8C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH7_CM0	000E 8B90$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH7_CM1	000E 8B94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH7_CN0	000E 8B98$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
ATOM1_CH7_STAT	000E 8B9C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	1
ATOM1_CH7_IRQ_NOTIFY	000E 8BA0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH7_IRQ_EN	000E 8BA4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH7_IRQ_FORCINT	000E 8BA8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM1_CH7_IRQ_MODE	000E 8BAC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
ATOM2_CH0_RDADDR	000E 9000$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH0_CTRL	000E 9004$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH0_SR0	000E 9008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH0_SR1	000E 900C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH0_CM0	000E 9010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH0_CM1	000E 9014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH0_CN0	000E 9018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH0_STAT	000E 901C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH0_IRQ_NOTIFY	000E 9020$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH0_IRQ_EN	000E 9024$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH0_IRQ_FORCINT	000E 9028$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH0_IRQ_MODE	000E 902C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_AGC_GLB_CTRL	000E 9040$_H$	0000 0000$_H$	0000 00FE$_H$	FFFF FF01$_H$	2
ATOM2_AGC_ENDIS_CTRL	000E 9044$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (26/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM2_AGC_ENDIS_STAT	000E 9048$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
ATOM2_AGC_ACT_TB	000E 904C$_H$	0000 0000$_H$	F800 0000$_H$	07FF FFFF$_H$	2
ATOM2_AGC_OUTEN_CTRL	000E 9050$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
ATOM2_AGC_OUTEN_STAT	000E 9054$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
ATOM2_AGC_FUPD_CTRL	000E 9058$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	2
ATOM2_AGC_INT_TRIG	000E 905C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
ATOM2_CH1_RDADDR	000E 9080$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH1_CTRL	000E 9084$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH1_SR0	000E 9088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH1_SR1	000E 908C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH1_CM0	000E 9090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH1_CM1	000E 9094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH1_CN0	000E 9098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH1_STAT	000E 909C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH1_IRQ_NOTIFY	000E 90A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH1_IRQ_EN	000E 90A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH1_IRQ_FORCINT	000E 90A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH1_IRQ_MODE	000E 90AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH2_RDADDR	000E 9100$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH2_CTRL	000E 9104$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH2_SR0	000E 9108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH2_SR1	000E 910C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH2_CM0	000E 9110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH2_CM1	000E 9114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH2_CN0	000E 9118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH2_STAT	000E 911C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH2_IRQ_NOTIFY	000E 9120$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH2_IRQ_EN	000E 9124$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH2_IRQ_FORCINT	000E 9128$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH2_IRQ_MODE	000E 912C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH3_RDADDR	000E 9180$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH3_CTRL	000E 9184$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH3_SR0	000E 9188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH3_SR1	000E 918C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH3_CM0	000E 9190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH3_CM1	000E 9194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH3_CN0	000E 9198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH3_STAT	000E 919C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH3_IRQ_NOTIFY	000E 91A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH3_IRQ_EN	000E 91A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH3_IRQ_FORCINT	000E 91A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH3_IRQ_MODE	000E 91AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH4_RDADDR	000E 9200$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (27/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM2_CH4_CTRL	000E 9204$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH4_SR0	000E 9208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH4_SR1	000E 920C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH4_CM0	000E 9210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH4_CM1	000E 9214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH4_CN0	000E 9218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH4_STAT	000E 921C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH4_IRQ_NOTIFY	000E 9220$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH4_IRQ_EN	000E 9224$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH4_IRQ_FORCINT	000E 9228$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH4_IRQ_MODE	000E 922C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH5_RDADDR	000E 9280$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH5_CTRL	000E 9284$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH5_SR0	000E 9288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH5_SR1	000E 928C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH5_CM0	000E 9290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH5_CM1	000E 9294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH5_CN0	000E 9298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH5_STAT	000E 929C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH5_IRQ_NOTIFY	000E 92A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH5_IRQ_EN	000E 92A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH5_IRQ_FORCINT	000E 92A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH5_IRQ_MODE	000E 92AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH6_RDADDR	000E 9300$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH6_CTRL	000E 9304$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH6_SR0	000E 9308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH6_SR1	000E 930C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH6_CM0	000E 9310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH6_CM1	000E 9314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH6_CN0	000E 9318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH6_STAT	000E 931C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2
ATOM2_CH6_IRQ_NOTIFY	000E 9320$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH6_IRQ_EN	000E 9324$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH6_IRQ_FORCINT	000E 9328$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH6_IRQ_MODE	000E 932C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH7_RDADDR	000E 9380$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	2
ATOM2_CH7_CTRL	000E 9384$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	2
ATOM2_CH7_SR0	000E 9388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH7_SR1	000E 938C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH7_CM0	000E 9390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH7_CM1	000E 9394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH7_CN0	000E 9398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
ATOM2_CH7_STAT	000E 939C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (28/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM2_CH7_IRQ_NOTIFY	000E 93A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH7_IRQ_EN	000E 93A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH7_IRQ_FORCINT	000E 93A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM2_CH7_IRQ_MODE	000E 93AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
ATOM3_CH0_RDADDR	000E 9800$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH0_CTRL	000E 9804$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH0_SR0	000E 9808$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH0_SR1	000E 980C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH0_CM0	000E 9810$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH0_CM1	000E 9814$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH0_CN0	000E 9818$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH0_STAT	000E 981C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH0_IRQ_NOTIFY	000E 9820$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH0_IRQ_EN	000E 9824$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH0_IRQ_FORCINT	000E 9828$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH0_IRQ_MODE	000E 982C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_AGC_GLB_CTRL	000E 9840$_H$	0000 0000$_H$	0000 00FE$_H$	FFFF FF01$_H$	3
ATOM3_AGC_ENDIS_CTRL	000E 9844$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
ATOM3_AGC_ENDIS_STAT	000E 9848$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
ATOM3_AGC_ACT_TB	000E 984C$_H$	0000 0000$_H$	F800 0000$_H$	07FF FFFF$_H$	3
ATOM3_AGC_OUTEN_CTRL	000E 9850$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
ATOM3_AGC_OUTEN_STAT	000E 9854$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
ATOM3_AGC_FUPD_CTRL	000E 9858$_H$	0000 0000$_H$	0000 0000$_H$	FFFF FFFF$_H$	3
ATOM3_AGC_INT_TRIG	000E 985C$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
ATOM3_CH1_RDADDR	000E 9880$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH1_CTRL	000E 9884$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH1_SR0	000E 9888$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH1_SR1	000E 988C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH1_CM0	000E 9890$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH1_CM1	000E 9894$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH1_CN0	000E 9898$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH1_STAT	000E 989C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH1_IRQ_NOTIFY	000E 98A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH1_IRQ_EN	000E 98A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH1_IRQ_FORCINT	000E 98A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH1_IRQ_MODE	000E 98AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH2_RDADDR	000E 9900$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH2_CTRL	000E 9904$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH2_SR0	000E 9908$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH2_SR1	000E 990C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH2_CM0	000E 9910$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH2_CM1	000E 9914$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH2_CN0	000E 9918$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (29/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM3_CH2_STAT	000E 991C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH2_IRQ_NOTIFY	000E 9920$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH2_IRQ_EN	000E 9924$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH2_IRQ_FORCINT	000E 9928$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH2_IRQ_MODE	000E 992C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH3_RDADDR	000E 9980$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH3_CTRL	000E 9984$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH3_SR0	000E 9988$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH3_SR1	000E 998C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH3_CM0	000E 9990$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH3_CM1	000E 9994$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH3_CN0	000E 9998$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH3_STAT	000E 999C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH3_IRQ_NOTIFY	000E 99A0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH3_IRQ_EN	000E 99A4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH3_IRQ_FORCINT	000E 99A8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH3_IRQ_MODE	000E 99AC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH4_RDADDR	000E 9A00$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH4_CTRL	000E 9A04$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH4_SR0	000E 9A08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH4_SR1	000E 9A0C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH4_CM0	000E 9A10$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH4_CM1	000E 9A14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH4_CN0	000E 9A18$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH4_STAT	000E 9A1C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH4_IRQ_NOTIFY	000E 9A20$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH4_IRQ_EN	000E 9A24$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH4_IRQ_FORCINT	000E 9A28$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH4_IRQ_MODE	000E 9A2C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH5_RDADDR	000E 9A80$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH5_CTRL	000E 9A84$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH5_SR0	000E 9A88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH5_SR1	000E 9A8C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH5_CM0	000E 9A90$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH5_CM1	000E 9A94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH5_CN0	000E 9A98$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH5_STAT	000E 9A9C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH5_IRQ_NOTIFY	000E 9AA0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH5_IRQ_EN	000E 9AA4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH5_IRQ_FORCINT	000E 9AA8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH5_IRQ_MODE	000E 9AAC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH6_RDADDR	000E 9B00$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH6_CTRL	000E 9B04$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (30/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
ATOM3_CH6_SR0	000E 9B08$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH6_SR1	000E 9B0C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH6_CM0	000E 9B10$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH6_CM1	000E 9B14$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH6_CN0	000E 9B18$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH6_STAT	000E 9B1C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH6_IRQ_NOTIFY	000E 9B20$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH6_IRQ_EN	000E 9B24$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH6_IRQ_FORCINT	000E 9B28$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH6_IRQ_MODE	000E 9B2C$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH7_RDADDR	000E 9B80$_H$	01FE 01FE$_H$	FE00 FE00$_H$	01FF 01FF$_H$	3
ATOM3_CH7_CTRL	000E 9B84$_H$	0000 0000$_H$	1000 0000$_H$	EFFF FFFF$_H$	3
ATOM3_CH7_SR0	000E 9B88$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH7_SR1	000E 9B8C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH7_CM0	000E 9B90$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH7_CM1	000E 9B94$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH7_CN0	000E 9B98$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
ATOM3_CH7_STAT	000E 9B9C$_H$	0000 0001$_H$	E000 FFFE$_H$	1FFF 0001$_H$	3
ATOM3_CH7_IRQ_NOTIFY	000E 9BA0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH7_IRQ_EN	000E 9BA4$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH7_IRQ_FORCINT	000E 9BA8$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
ATOM3_CH7_IRQ_MODE	000E 9BAC$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS0_CH0_R0	000F 0000$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R1	000F 0004$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R2	000F 0008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R3	000F 000C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R4	000F 0010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R5	000F 0014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R6	000F 0018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_R7	000F 001C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_CTRL	000F 0020$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH0_ACB	000F 0024$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CTRG	000F 0028$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_STRG	000F 002C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH0_MHB	000F 003C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH0_PC	000F 0040$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH0_IRQ_NOTIFY	000F 0044$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH0_IRQ_EN	000F 0048$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH0_IRQ_FORCINT	000F 004C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH0_IRQ_MODE	000F 0050$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH0_EIRQ_EN	000F 0054$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_REG_PROT	000F 0060$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CTRL_STAT	000F 0064$_H$	0001 0000$_H$	F88C F0FC$_H$	0773 0F03$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (31/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS0_RESET	000F 0068$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CAT	000F 006C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CWT	000F 0070$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_ERR	000F 007C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH1_R0	000F 0080$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R1	000F 0084$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R2	000F 0088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R3	000F 008C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R4	000F 0090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R5	000F 0094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R6	000F 0098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_R7	000F 009C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH1_CTRL	000F 00A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH1_ACB	000F 00A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH1_MHB	000F 00BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH1_PC	000F 00C0$_H$	0000 0004$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH1_IRQ_NOTIFY	000F 00C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH1_IRQ_EN	000F 00C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH1_IRQ_FORCINT	000F 00CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH1_IRQ_MODE	000F 00D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH1_EIRQ_EN	000F 00D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH2_R0	000F 0100$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R1	000F 0104$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R2	000F 0108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R3	000F 010C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R4	000F 0110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R5	000F 0114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R6	000F 0118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_R7	000F 011C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH2_CTRL	000F 0120$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH2_ACB	000F 0124$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH2_MHB	000F 013C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH2_PC	000F 0140$_H$	0000 0008$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH2_IRQ_NOTIFY	000F 0144$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH2_IRQ_EN	000F 0148$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH2_IRQ_FORCINT	000F 014C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH2_IRQ_MODE	000F 0150$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH2_EIRQ_EN	000F 0154$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH3_R0	000F 0180$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R1	000F 0184$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R2	000F 0188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R3	000F 018C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R4	000F 0190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (32/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS0_CH3_R5	000F 0194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R6	000F 0198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_R7	000F 019C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH3_CTRL	000F 01A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH3_ACB	000F 01A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH3_MHB	000F 01BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH3_PC	000F 01C0$_H$	0000 000C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH3_IRQ_NOTIFY	000F 01C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH3_IRQ_EN	000F 01C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH3_IRQ_FORCINT	000F 01CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH3_IRQ_MODE	000F 01D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH3_EIRQ_EN	000F 01D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH4_R0	000F 0200$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R1	000F 0204$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R2	000F 0208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R3	000F 020C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R4	000F 0210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R5	000F 0214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R6	000F 0218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_R7	000F 021C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH4_CTRL	000F 0220$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH4_ACB	000F 0224$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH4_MHB	000F 023C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH4_PC	000F 0240$_H$	0000 0010$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH4_IRQ_NOTIFY	000F 0244$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH4_IRQ_EN	000F 0248$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH4_IRQ_FORCINT	000F 024C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH4_IRQ_MODE	000F 0250$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH4_EIRQ_EN	000F 0254$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH5_R0	000F 0280$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R1	000F 0284$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R2	000F 0288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R3	000F 028C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R4	000F 0290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R5	000F 0294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R6	000F 0298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_R7	000F 029C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH5_CTRL	000F 02A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH5_ACB	000F 02A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH5_MHB	000F 02BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH5_PC	000F 02C0$_H$	0000 0014$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH5_IRQ_NOTIFY	000F 02C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH5_IRQ_EN	000F 02C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0

Table 38.516 Register and memory addresses used by GTM-IP_358 (33/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS0_CH5_IRQ_FORCINT	000F 02CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH5_IRQ_MODE	000F 02D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH5_EIRQ_EN	000F 02D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH6_R0	000F 0300$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R1	000F 0304$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R2	000F 0308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R3	000F 030C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R4	000F 0310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R5	000F 0314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R6	000F 0318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_R7	000F 031C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH6_CTRL	000F 0320$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH6_ACB	000F 0324$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH6_MHB	000F 033C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH6_PC	000F 0340$_H$	0000 0018$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH6_IRQ_NOTIFY	000F 0344$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH6_IRQ_EN	000F 0348$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH6_IRQ_FORCINT	000F 034C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH6_IRQ_MODE	000F 0350$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH6_EIRQ_EN	000F 0354$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH7_R0	000F 0380$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R1	000F 0384$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R2	000F 0388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R3	000F 038C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R4	000F 0390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R5	000F 0394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R6	000F 0398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_R7	000F 039C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	0
MCS0_CH7_CTRL	000F 03A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	0
MCS0_CH7_ACB	000F 03A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	0
MCS0_CH7_MHB	000F 03BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	0
MCS0_CH7_PC	000F 03C0$_H$	0000 001C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	0
MCS0_CH7_IRQ_NOTIFY	000F 03C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH7_IRQ_EN	000F 03C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH7_IRQ_FORCINT	000F 03CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS0_CH7_IRQ_MODE	000F 03D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	0
MCS0_CH7_EIRQ_EN	000F 03D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	0
MCS1_CH0_R0	000F 1000$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R1	000F 1004$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R2	000F 1008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R3	000F 100C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R4	000F 1010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R5	000F 1014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (34/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS1_CH0_R6	000F 1018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_R7	000F 101C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_CTRL	000F 1020$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH0_ACB	000F 1024$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CTRG	000F 1028$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_STRG	000F 102C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH0_MHB	000F 103C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH0_PC	000F 1040$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH0_IRQ_NOTIFY	000F 1044$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH0_IRQ_EN	000F 1048$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH0_IRQ_FORCINT	000F 104C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH0_IRQ_MODE	000F 1050$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH0_EIRQ_EN	000F 1054$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_REG_PROT	000F 1060$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CTRL_STAT	000F 1064$_H$	0001 0000$_H$	F88C F0FC$_H$	0773 0F03$_H$	1
MCS1_RESET	000F 1068$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CAT	000F 106C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CWT	000F 1070$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_ERR	000F 107C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH1_R0	000F 1080$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R1	000F 1084$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R2	000F 1088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R3	000F 108C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R4	000F 1090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R5	000F 1094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R6	000F 1098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_R7	000F 109C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH1_CTRL	000F 10A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH1_ACB	000F 10A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH1_MHB	000F 10BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH1_PC	000F 10C0$_H$	0000 0004$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH1_IRQ_NOTIFY	000F 10C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH1_IRQ_EN	000F 10C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH1_IRQ_FORCINT	000F 10CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH1_IRQ_MODE	000F 10D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH1_EIRQ_EN	000F 10D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH2_R0	000F 1100$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R1	000F 1104$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R2	000F 1108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R3	000F 110C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R4	000F 1110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R5	000F 1114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_R6	000F 1118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (35/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS1_CH2_R7	000F 111C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH2_CTRL	000F 1120$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH2_ACB	000F 1124$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH2_MHB	000F 113C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH2_PC	000F 1140$_H$	0000 0008$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH2_IRQ_NOTIFY	000F 1144$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH2_IRQ_EN	000F 1148$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH2_IRQ_FORCINT	000F 114C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH2_IRQ_MODE	000F 1150$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH2_EIRQ_EN	000F 1154$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH3_R0	000F 1180$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R1	000F 1184$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R2	000F 1188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R3	000F 118C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R4	000F 1190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R5	000F 1194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R6	000F 1198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_R7	000F 119C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH3_CTRL	000F 11A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH3_ACB	000F 11A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH3_MHB	000F 11BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH3_PC	000F 11C0$_H$	0000 000C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH3_IRQ_NOTIFY	000F 11C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH3_IRQ_EN	000F 11C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH3_IRQ_FORCINT	000F 11CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH3_IRQ_MODE	000F 11D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH3_EIRQ_EN	000F 11D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH4_R0	000F 1200$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R1	000F 1204$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R2	000F 1208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R3	000F 120C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R4	000F 1210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R5	000F 1214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R6	000F 1218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_R7	000F 121C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH4_CTRL	000F 1220$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH4_ACB	000F 1224$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH4_MHB	000F 123C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH4_PC	000F 1240$_H$	0000 0010$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH4_IRQ_NOTIFY	000F 1244$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH4_IRQ_EN	000F 1248$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH4_IRQ_FORCINT	000F 124C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH4_IRQ_MODE	000F 1250$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (36/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS1_CH4_EIRQ_EN	000F 1254$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH5_R0	000F 1280$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R1	000F 1284$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R2	000F 1288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R3	000F 128C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R4	000F 1290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R5	000F 1294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R6	000F 1298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_R7	000F 129C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH5_CTRL	000F 12A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH5_ACB	000F 12A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH5_MHB	000F 12BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH5_PC	000F 12C0$_H$	0000 0014$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH5_IRQ_NOTIFY	000F 12C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH5_IRQ_EN	000F 12C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH5_IRQ_FORCINT	000F 12CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH5_IRQ_MODE	000F 12D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH5_EIRQ_EN	000F 12D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH6_R0	000F 1300$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R1	000F 1304$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R2	000F 1308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R3	000F 130C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R4	000F 1310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R5	000F 1314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R6	000F 1318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_R7	000F 131C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH6_CTRL	000F 1320$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH6_ACB	000F 1324$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH6_MHB	000F 133C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH6_PC	000F 1340$_H$	0000 0018$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH6_IRQ_NOTIFY	000F 1344$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH6_IRQ_EN	000F 1348$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH6_IRQ_FORCINT	000F 134C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH6_IRQ_MODE	000F 1350$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH6_EIRQ_EN	000F 1354$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH7_R0	000F 1380$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R1	000F 1384$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R2	000F 1388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R3	000F 138C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R4	000F 1390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R5	000F 1394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R6	000F 1398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1
MCS1_CH7_R7	000F 139C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	1

Table 38.516 Register and memory addresses used by GTM-IP_358 (37/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS1_CH7_CTRL	000F 13A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	1
MCS1_CH7_ACB	000F 13A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	1
MCS1_CH7_MHB	000F 13BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	1
MCS1_CH7_PC	000F 13C0$_H$	0000 001C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	1
MCS1_CH7_IRQ_NOTIFY	000F 13C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH7_IRQ_EN	000F 13C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH7_IRQ_FORCINT	000F 13CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS1_CH7_IRQ_MODE	000F 13D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	1
MCS1_CH7_EIRQ_EN	000F 13D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	1
MCS2_CH0_R0	000F 2000$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R1	000F 2004$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R2	000F 2008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R3	000F 200C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R4	000F 2010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R5	000F 2014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R6	000F 2018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_R7	000F 201C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_CTRL	000F 2020$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH0_ACB	000F 2024$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CTRG	000F 2028$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_STRG	000F 202C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH0_MHB	000F 203C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH0_PC	000F 2040$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH0_IRQ_NOTIFY	000F 2044$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH0_IRQ_EN	000F 2048$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH0_IRQ_FORCINT	000F 204C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH0_IRQ_MODE	000F 2050$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH0_EIRQ_EN	000F 2054$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_REG_PROT	000F 2060$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CTRL_STAT	000F 2064$_H$	0001 0000$_H$	F88C F0FC$_H$	0773 0F03$_H$	2
MCS2_RESET	000F 2068$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CAT	000F 206C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CWT	000F 2070$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_ERR	000F 207C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH1_R0	000F 2080$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R1	000F 2084$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R2	000F 2088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R3	000F 208C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R4	000F 2090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R5	000F 2094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R6	000F 2098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_R7	000F 209C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH1_CTRL	000F 20A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (38/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS2_CH1_ACB	000F 20A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH1_MHB	000F 20BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH1_PC	000F 20C0$_H$	0000 0004$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH1_IRQ_NOTIFY	000F 20C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH1_IRQ_EN	000F 20C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH1_IRQ_FORCINT	000F 20CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH1_IRQ_MODE	000F 20D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH1_EIRQ_EN	000F 20D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH2_R0	000F 2100$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R1	000F 2104$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R2	000F 2108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R3	000F 210C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R4	000F 2110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R5	000F 2114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R6	000F 2118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_R7	000F 211C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH2_CTRL	000F 2120$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH2_ACB	000F 2124$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH2_MHB	000F 213C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH2_PC	000F 2140$_H$	0000 0008$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH2_IRQ_NOTIFY	000F 2144$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH2_IRQ_EN	000F 2148$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH2_IRQ_FORCINT	000F 214C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH2_IRQ_MODE	000F 2150$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH2_EIRQ_EN	000F 2154$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH3_R0	000F 2180$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R1	000F 2184$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R2	000F 2188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R3	000F 218C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R4	000F 2190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R5	000F 2194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R6	000F 2198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_R7	000F 219C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH3_CTRL	000F 21A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH3_ACB	000F 21A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH3_MHB	000F 21BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH3_PC	000F 21C0$_H$	0000 000C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH3_IRQ_NOTIFY	000F 21C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH3_IRQ_EN	000F 21C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH3_IRQ_FORCINT	000F 21CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH3_IRQ_MODE	000F 21D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH3_EIRQ_EN	000F 21D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH4_R0	000F 2200$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (39/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS2_CH4_R1	000F 2204$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R2	000F 2208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R3	000F 220C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R4	000F 2210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R5	000F 2214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R6	000F 2218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_R7	000F 221C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH4_CTRL	000F 2220$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH4_ACB	000F 2224$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH4_MHB	000F 223C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH4_PC	000F 2240$_H$	0000 0010$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH4_IRQ_NOTIFY	000F 2244$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH4_IRQ_EN	000F 2248$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH4_IRQ_FORCINT	000F 224C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH4_IRQ_MODE	000F 2250$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH4_EIRQ_EN	000F 2254$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH5_R0	000F 2280$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R1	000F 2284$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R2	000F 2288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R3	000F 228C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R4	000F 2290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R5	000F 2294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R6	000F 2298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_R7	000F 229C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH5_CTRL	000F 22A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH5_ACB	000F 22A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH5_MHB	000F 22BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH5_PC	000F 22C0$_H$	0000 0014$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH5_IRQ_NOTIFY	000F 22C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH5_IRQ_EN	000F 22C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH5_IRQ_FORCINT	000F 22CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH5_IRQ_MODE	000F 22D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH5_EIRQ_EN	000F 22D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH6_R0	000F 2300$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R1	000F 2304$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R2	000F 2308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R3	000F 230C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R4	000F 2310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R5	000F 2314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R6	000F 2318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_R7	000F 231C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH6_CTRL	000F 2320$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH6_ACB	000F 2324$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2

Table 38.516 Register and memory addresses used by GTM-IP_358 (40/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS2_CH6_MHB	000F 233C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH6_PC	000F 2340$_H$	0000 0018$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH6_IRQ_NOTIFY	000F 2344$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH6_IRQ_EN	000F 2348$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH6_IRQ_FORCINT	000F 234C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH6_IRQ_MODE	000F 2350$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH6_EIRQ_EN	000F 2354$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH7_R0	000F 2380$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R1	000F 2384$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R2	000F 2388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R3	000F 238C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R4	000F 2390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R5	000F 2394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R6	000F 2398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_R7	000F 239C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	2
MCS2_CH7_CTRL	000F 23A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	2
MCS2_CH7_ACB	000F 23A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	2
MCS2_CH7_MHB	000F 23BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	2
MCS2_CH7_PC	000F 23C0$_H$	0000 001C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	2
MCS2_CH7_IRQ_NOTIFY	000F 23C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH7_IRQ_EN	000F 23C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH7_IRQ_FORCINT	000F 23CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS2_CH7_IRQ_MODE	000F 23D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	2
MCS2_CH7_EIRQ_EN	000F 23D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	2
MCS3_CH0_R0	000F 3000$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R1	000F 3004$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R2	000F 3008$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R3	000F 300C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R4	000F 3010$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R5	000F 3014$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R6	000F 3018$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_R7	000F 301C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_CTRL	000F 3020$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH0_ACB	000F 3024$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CTRG	000F 3028$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_STRG	000F 302C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH0_MHB	000F 303C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH0_PC	000F 3040$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH0_IRQ_NOTIFY	000F 3044$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH0_IRQ_EN	000F 3048$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH0_IRQ_FORCINT	000F 304C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH0_IRQ_MODE	000F 3050$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH0_EIRQ_EN	000F 3054$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (41/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS3_REG_PROT	000F 3060$_H$	0000 0000$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CTRL_STAT	000F 3064$_H$	0001 0000$_H$	F88C F0FC$_H$	0773 0F03$_H$	3
MCS3_RESET	000F 3068$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CAT	000F 306C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CWT	000F 3070$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_ERR	000F 307C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH1_R0	000F 3080$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R1	000F 3084$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R2	000F 3088$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R3	000F 308C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R4	000F 3090$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R5	000F 3094$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R6	000F 3098$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_R7	000F 309C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH1_CTRL	000F 30A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH1_ACB	000F 30A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH1_MHB	000F 30BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH1_PC	000F 30C0$_H$	0000 0004$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH1_IRQ_NOTIFY	000F 30C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH1_IRQ_EN	000F 30C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH1_IRQ_FORCINT	000F 30CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH1_IRQ_MODE	000F 30D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH1_EIRQ_EN	000F 30D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH2_R0	000F 3100$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R1	000F 3104$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R2	000F 3108$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R3	000F 310C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R4	000F 3110$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R5	000F 3114$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R6	000F 3118$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_R7	000F 311C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH2_CTRL	000F 3120$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH2_ACB	000F 3124$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH2_MHB	000F 313C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH2_PC	000F 3140$_H$	0000 0008$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH2_IRQ_NOTIFY	000F 3144$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH2_IRQ_EN	000F 3148$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH2_IRQ_FORCINT	000F 314C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH2_IRQ_MODE	000F 3150$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH2_EIRQ_EN	000F 3154$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH3_R0	000F 3180$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R1	000F 3184$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R2	000F 3188$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (42/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS3_CH3_R3	000F 318C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R4	000F 3190$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R5	000F 3194$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R6	000F 3198$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_R7	000F 319C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH3_CTRL	000F 31A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH3_ACB	000F 31A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH3_MHB	000F 31BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH3_PC	000F 31C0$_H$	0000 000C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH3_IRQ_NOTIFY	000F 31C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH3_IRQ_EN	000F 31C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH3_IRQ_FORCINT	000F 31CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH3_IRQ_MODE	000F 31D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH3_EIRQ_EN	000F 31D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH4_R0	000F 3200$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R1	000F 3204$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R2	000F 3208$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R3	000F 320C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R4	000F 3210$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R5	000F 3214$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R6	000F 3218$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_R7	000F 321C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH4_CTRL	000F 3220$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH4_ACB	000F 3224$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH4_MHB	000F 323C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH4_PC	000F 3240$_H$	0000 0010$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH4_IRQ_NOTIFY	000F 3244$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH4_IRQ_EN	000F 3248$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH4_IRQ_FORCINT	000F 324C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH4_IRQ_MODE	000F 3250$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH4_EIRQ_EN	000F 3254$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH5_R0	000F 3280$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R1	000F 3284$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R2	000F 3288$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R3	000F 328C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R4	000F 3290$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R5	000F 3294$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R6	000F 3298$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_R7	000F 329C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH5_CTRL	000F 32A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH5_ACB	000F 32A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH5_MHB	000F 32BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH5_PC	000F 32C0$_H$	0000 0014$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3

Table 38.516 Register and memory addresses used by GTM-IP_358 (43/43)

Register Label	Register Address	Register Reset Value	Register Reserved Bits	Register Volatile Bits	Cluster #
MCS3_CH5_IRQ_NOTIFY	000F 32C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH5_IRQ_EN	000F 32C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH5_IRQ_FORCINT	000F 32CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH5_IRQ_MODE	000F 32D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH5_EIRQ_EN	000F 32D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH6_R0	000F 3300$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R1	000F 3304$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R2	000F 3308$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R3	000F 330C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R4	000F 3310$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R5	000F 3314$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R6	000F 3318$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_R7	000F 331C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH6_CTRL	000F 3320$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH6_ACB	000F 3324$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH6_MHB	000F 333C$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH6_PC	000F 3340$_H$	0000 0018$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH6_IRQ_NOTIFY	000F 3344$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH6_IRQ_EN	000F 3348$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH6_IRQ_FORCINT	000F 334C$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH6_IRQ_MODE	000F 3350$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH6_EIRQ_EN	000F 3354$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH7_R0	000F 3380$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R1	000F 3384$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R2	000F 3388$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R3	000F 338C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R4	000F 3390$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R5	000F 3394$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R6	000F 3398$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_R7	000F 339C$_H$	0000 0000$_H$	FF00 0000$_H$	00FF FFFF$_H$	3
MCS3_CH7_CTRL	000F 33A0$_H$	0000 0000$_H$	FFF8 F808$_H$	0007 07F7$_H$	3
MCS3_CH7_ACB	000F 33A4$_H$	0000 0000$_H$	FFFF FFE0$_H$	0000 001F$_H$	3
MCS3_CH7_MHB	000F 33BC$_H$	0000 0000$_H$	FFFF FF00$_H$	0000 00FF$_H$	3
MCS3_CH7_PC	000F 33C0$_H$	0000 001C$_H$	FFFF 0000$_H$	0000 FFFF$_H$	3
MCS3_CH7_IRQ_NOTIFY	000F 33C4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH7_IRQ_EN	000F 33C8$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH7_IRQ_FORCINT	000F 33CC$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3
MCS3_CH7_IRQ_MODE	000F 33D0$_H$	0000 0000$_H$	FFFF FFFC$_H$	0000 0003$_H$	3
MCS3_CH7_EIRQ_EN	000F 33D4$_H$	0000 0000$_H$	FFFF FFF8$_H$	0000 0007$_H$	3

38.28.7 GTM Register Addresses Mapping Legacy Addresses

The following **Table 38.517** lists all register addresses used by GTM-IP_358 which have additional legacy addresses defined by older GTM Generation.

Table 38.517 GTM Register Addresses Mapping Legacy Addresses (1/3)

Register Label	Register Address	Legacy Register Label	Legacy Address
CCM0_HW_CONF	000E 21DC$_H$	GTM_HW_CONF	0000 0024$_H$
CCM1_HW_CONF	000E 23DC$_H$	GTM_HW_CONF	0000 0024$_H$
CCM2_HW_CONF	000E 25DC$_H$	GTM_HW_CONF	0000 0024$_H$
CCM3_HW_CONF	000E 27DC$_H$	GTM_HW_CONF	0000 0024$_H$
CCM0_TIM_AUX_IN_SRC	000E 21E0$_H$	GTM_TIM0_AUX_IN_SRC	0000 0040$_H$
CCM1_TIM_AUX_IN_SRC	000E 23E0$_H$	GTM_TIM1_AUX_IN_SRC	0000 0044$_H$
CCM2_TIM_AUX_IN_SRC	000E 25E0$_H$	GTM_TIM2_AUX_IN_SRC	0000 0048$_H$
CCM3_TIM_AUX_IN_SRC	000E 27E0$_H$	GTM_TIM3_AUX_IN_SRC	0000 004C$_H$
CCM0_EXT_CAP_EN	000E 21E4$_H$	GTM_EXT_CAP_EN_0	0000 005C$_H$
CCM1_EXT_CAP_EN	000E 23E4$_H$	GTM_EXT_CAP_EN_1	0000 0060$_H$
CCM2_EXT_CAP_EN	000E 25E4$_H$	GTM_EXT_CAP_EN_2	0000 0064$_H$
CCM3_EXT_CAP_EN	000E 27E4$_H$	GTM_EXT_CAP_EN_3	0000 0068$_H$
CCM0_ATOM_OUT	000E 21EC$_H$	GTM_ATOM0_OUT	0000 0098$_H$
CCM1_ATOM_OUT	000E 23EC$_H$	—	—
CCM2_ATOM_OUT	000E 25EC$_H$	GTM_ATOM2_OUT	0000 009C$_H$
CCM3_ATOM_OUT	000E 27EC$_H$	—	—
ICM_IRQG_MCS0_CI	0000 0720$_H$	LEG_ICM_IRQG_MCS0_CI	0000 0648$_H$
ICM_IRQG_MCS1_CI	0000 0724$_H$	LEG_ICM_IRQG_MCS1_CI	0000 064C$_H$
ICM_IRQG_MCS2_CI	0000 0728$_H$	LEG_ICM_IRQG_MCS2_CI	0000 0650$_H$
ICM_IRQG_MCS3_CI	0000 072C$_H$	LEG_ICM_IRQG_MCS3_CI	0000 0654$_H$
CDTM0_DTM4_CTRL	000E 4100$_H$	DTM24_CTRL	0001 3600$_H$
CDTM0_DTM4_CH_CTRL1	000E 4104$_H$	DTM24_CH_CTRL1	0001 3604$_H$
CDTM0_DTM4_CH_CTRL2	000E 4108$_H$	DTM24_CH_CTRL2	0001 3608$_H$
CDTM0_DTM4_CH_CTRL2_SR	000E 410C$_H$	DTM24_CH_CTRL2_SR	0001 360C$_H$
CDTM0_DTM4_PS_CTRL	000E 4110$_H$	DTM24_PS_CTRL	0001 3610$_H$
CDTM0_DTM4_CH0_DTV	000E 4114$_H$	DTM24_CH0_DTV	0001 3614$_H$
CDTM0_DTM4_CH1_DTV	000E 4118$_H$	DTM24_CH1_DTV	0001 3618$_H$
CDTM0_DTM4_CH2_DTV	000E 411C$_H$	DTM24_CH2_DTV	0001 361C$_H$
CDTM0_DTM4_CH3_DTV	000E 4120$_H$	DTM24_CH3_DTV	0001 3620$_H$
CDTM0_DTM4_CH_SR	000E 4124$_H$	DTM24_CH_SR	0001 3624$_H$
CDTM0_DTM4_CH_CTRL3	000E 4128$_H$	DTM24_CH_CTRL3	0001 3628$_H$
CDTM0_DTM5_CTRL	000E 4140$_H$	DTM25_CTRL	0001 3640$_H$
CDTM0_DTM5_CH_CTRL1	000E 4144$_H$	DTM25_CH_CTRL1	0001 3644$_H$
CDTM0_DTM5_CH_CTRL2	000E 4148$_H$	DTM25_CH_CTRL2	0001 3648$_H$
CDTM0_DTM5_CH_CTRL2_SR	000E 414C$_H$	DTM25_CH_CTRL2_SR	0001 364C$_H$
CDTM0_DTM5_PS_CTRL	000E 4150$_H$	DTM25_PS_CTRL	0001 3650$_H$
CDTM0_DTM5_CH0_DTV	000E 4154$_H$	DTM25_CH0_DTV	0001 3654$_H$
CDTM0_DTM5_CH1_DTV	000E 4158$_H$	DTM25_CH1_DTV	0001 3658$_H$
CDTM0_DTM5_CH2_DTV	000E 415C$_H$	DTM25_CH2_DTV	0001 365C$_H$
CDTM0_DTM5_CH3_DTV	000E 4160$_H$	DTM25_CH3_DTV	0001 3660$_H$

Table 38.517 GTM Register Addresses Mapping Legacy Addresses (2/3)

Register Label	Register Address	Legacy Register Label	Legacy Address
CDTM0_DTM5_CH_SR	000E 4164$_H$	DTM25_CH_SR	0001 3664$_H$
CDTM0_DTM5_CH_CTRL3	000E 4168$_H$	DTM25_CH_CTRL3	0001 3668$_H$
CDTM1_DTM4_CTRL	000E 4500$_H$	DTM26_CTRL	0001 3680$_H$
CDTM1_DTM4_CH_CTRL1	000E 4504$_H$	DTM26_CH_CTRL1	0001 3684$_H$
CDTM1_DTM4_CH_CTRL2	000E 4508$_H$	DTM26_CH_CTRL2	0001 3688$_H$
CDTM1_DTM4_CH_CTRL2_SR	000E 450C$_H$	DTM26_CH_CTRL2_SR	0001 368C$_H$
CDTM1_DTM4_PS_CTRL	000E 4510$_H$	DTM26_PS_CTRL	0001 3690$_H$
CDTM1_DTM4_CH0_DTV	000E 4514$_H$	DTM26_CH0_DTV	0001 3694$_H$
CDTM1_DTM4_CH1_DTV	000E 4518$_H$	DTM26_CH1_DTV	0001 3698$_H$
CDTM1_DTM4_CH2_DTV	000E 451C$_H$	DTM26_CH2_DTV	0001 369C$_H$
CDTM1_DTM4_CH3_DTV	000E 4520$_H$	DTM26_CH3_DTV	0001 36A0$_H$
CDTM1_DTM4_CH_SR	000E 4524$_H$	DTM26_CH_SR	0001 36A4$_H$
CDTM1_DTM4_CH_CTRL3	000E 4528$_H$	DTM26_CH_CTRL3	0001 36A8$_H$
CDTM1_DTM5_CTRL	000E 4540$_H$	DTM27_CTRL	0001 36C0$_H$
CDTM1_DTM5_CH_CTRL1	000E 4544$_H$	DTM27_CH_CTRL1	0001 36C4$_H$
CDTM1_DTM5_CH_CTRL2	000E 4548$_H$	DTM27_CH_CTRL2	0001 36C8$_H$
CDTM1_DTM5_CH_CTRL2_SR	000E 454C$_H$	DTM27_CH_CTRL2_SR	0001 36CC$_H$
CDTM1_DTM5_PS_CTRL	000E 4550$_H$	DTM27_PS_CTRL	0001 36D0$_H$
CDTM1_DTM5_CH0_DTV	000E 4554$_H$	DTM27_CH0_DTV	0001 36D4$_H$
CDTM1_DTM5_CH1_DTV	000E 4558$_H$	DTM27_CH1_DTV	0001 36D8$_H$
CDTM1_DTM5_CH2_DTV	000E 455C$_H$	DTM27_CH2_DTV	0001 36DC$_H$
CDTM1_DTM5_CH3_DTV	000E 4560$_H$	DTM27_CH3_DTV	0001 36E0$_H$
CDTM1_DTM5_CH_SR	000E 4564$_H$	DTM27_CH_SR	0001 36E4$_H$
CDTM1_DTM5_CH_CTRL3	000E 4568$_H$	DTM27_CH_CTRL3	0001 36E8$_H$
CDTM2_DTM4_CTRL	000E 4900$_H$	DTM28_CTRL	0001 3700$_H$
CDTM2_DTM4_CH_CTRL1	000E 4904$_H$	DTM28_CH_CTRL1	0001 3704$_H$
CDTM2_DTM4_CH_CTRL2	000E 4908$_H$	DTM28_CH_CTRL2	0001 3708$_H$
CDTM2_DTM4_CH_CTRL2_SR	000E 490C$_H$	DTM28_CH_CTRL2_SR	0001 370C$_H$
CDTM2_DTM4_PS_CTRL	000E 4910$_H$	DTM28_PS_CTRL	0001 3710$_H$
CDTM2_DTM4_CH0_DTV	000E 4914$_H$	DTM28_CH0_DTV	0001 3714$_H$
CDTM2_DTM4_CH1_DTV	000E 4918$_H$	DTM28_CH1_DTV	0001 3718$_H$
CDTM2_DTM4_CH2_DTV	000E 491C$_H$	DTM28_CH2_DTV	0001 371C$_H$
CDTM2_DTM4_CH3_DTV	000E 4920$_H$	DTM28_CH3_DTV	0001 3720$_H$
CDTM2_DTM4_CH_SR	000E 4924$_H$	DTM28_CH_SR	0001 3724$_H$
CDTM2_DTM4_CH_CTRL3	000E 4928$_H$	DTM28_CH_CTRL3	0001 3728$_H$
CDTM2_DTM5_CTRL	000E 4940$_H$	DTM29_CTRL	0001 3740$_H$
CDTM2_DTM5_CH_CTRL1	000E 4944$_H$	DTM29_CH_CTRL1	0001 3744$_H$
CDTM2_DTM5_CH_CTRL2	000E 4948$_H$	DTM29_CH_CTRL2	0001 3748$_H$
CDTM2_DTM5_CH_CTRL2_SR	000E 494C$_H$	DTM29_CH_CTRL2_SR	0001 374C$_H$
CDTM2_DTM5_PS_CTRL	000E 4950$_H$	DTM29_PS_CTRL	0001 3750$_H$
CDTM2_DTM5_CH0_DTV	000E 4954$_H$	DTM29_CH0_DTV	0001 3754$_H$
CDTM2_DTM5_CH1_DTV	000E 4958$_H$	DTM29_CH1_DTV	0001 3758$_H$
CDTM2_DTM5_CH2_DTV	000E 495C$_H$	DTM29_CH2_DTV	0001 375C$_H$
CDTM2_DTM5_CH3_DTV	000E 4960$_H$	DTM29_CH3_DTV	0001 3760$_H$

Table 38.517 GTM Register Addresses Mapping Legacy Addresses (3/3)

Register Label	Register Address	Legacy Register Label	Legacy Address
CDTM2_DTM5_CH_SR	000E 4964$_H$	DTM29_CH_SR	0001 3764$_H$
CDTM2_DTM5_CH_CTRL3	000E 4968$_H$	DTM29_CH_CTRL3	0001 3768$_H$
CDTM3_DTM4_CTRL	000E 4D00$_H$	DTM30_CTRL	0001 3780$_H$
CDTM3_DTM4_CH_CTRL1	000E 4D04$_H$	DTM30_CH_CTRL1	0001 3784$_H$
CDTM3_DTM4_CH_CTRL2	000E 4D08$_H$	DTM30_CH_CTRL2	0001 3788$_H$
CDTM3_DTM4_CH_CTRL2_SR	000E 4D0C$_H$	DTM30_CH_CTRL2_SR	0001 378C$_H$
CDTM3_DTM4_PS_CTRL	000E 4D10$_H$	DTM30_PS_CTRL	0001 3790$_H$
CDTM3_DTM4_CH0_DTV	000E 4D14$_H$	DTM30_CH0_DTV	0001 3794$_H$
CDTM3_DTM4_CH1_DTV	000E 4D18$_H$	DTM30_CH1_DTV	0001 3798$_H$
CDTM3_DTM4_CH2_DTV	000E 4D1C$_H$	DTM30_CH2_DTV	0001 379C$_H$
CDTM3_DTM4_CH3_DTV	000E 4D20$_H$	DTM30_CH3_DTV	0001 37A0$_H$
CDTM3_DTM4_CH_SR	000E 4D24$_H$	DTM30_CH_SR	0001 37A4$_H$
CDTM3_DTM4_CH_CTRL3	000E 4D28$_H$	DTM30_CH_CTRL3	0001 37A8$_H$
CDTM3_DTM5_CTRL	000E 4D40$_H$	DTM31_CTRL	0001 37C0$_H$
CDTM3_DTM5_CH_CTRL1	000E 4D44$_H$	DTM31_CH_CTRL1	0001 37C4$_H$
CDTM3_DTM5_CH_CTRL2	000E 4D48$_H$	DTM31_CH_CTRL2	0001 37C8$_H$
CDTM3_DTM5_CH_CTRL2_SR	000E 4D4C$_H$	DTM31_CH_CTRL2_SR	0001 37CC$_H$
CDTM3_DTM5_PS_CTRL	000E 4D50$_H$	DTM31_PS_CTRL	0001 37D0$_H$
CDTM3_DTM5_CH0_DTV	000E 4D54$_H$	DTM31_CH0_DTV	0001 37D4$_H$
CDTM3_DTM5_CH1_DTV	000E 4D58$_H$	DTM31_CH1_DTV	0001 37D8$_H$
CDTM3_DTM5_CH2_DTV	000E 4D5C$_H$	DTM31_CH2_DTV	0001 37DC$_H$
CDTM3_DTM5_CH3_DTV	000E 4D60$_H$	DTM31_CH3_DTV	0001 37E0$_H$
CDTM3_DTM5_CH_SR	000E 4D64$_H$	DTM31_CH_SR	0001 37E4$_H$
CDTM3_DTM5_CH_CTRL3	000E 4D68$_H$	DTM31_CH_CTRL3	0001 37E8$_H$

38.28.8 MCS master interface address map

Each cluster uses the same MCS AEI bus master address map. If a cluster does not contain modules mentioned in the address map, the AEI status signal report a value “10” (illegal module access). The register label names mentioned in the table below equal to the label names in the specification, with exception to the cluster/module index i, which is always omitted in the address map of the MCS bus master interface (e.g. register TIM[i]_CH0_GPR0 is accessible as register TIM_CH0_GPR0 by the MCS of the i-th cluster).

NOTE

The registers CCM[i]_HW_CONF, CCM[i]_AUX_IN_SRC, CCM[i]_EXT_CAP_EN, CCM[i]_TOM_OUT, and CCM[i]_ATOM_OUT are global status and configuration registers that are mirrored from the group of TOP-Level registers to i-cluster. An access to the old GTM addresses are marked with LEG_ (legacy).

Table 38.518 MCS master interface address map (1/16)

Register Label	Register Address
TIM_CH0_GPR0	0000 0000$_H$
TIM_CH0_GPR1	0000 0004$_H$
TIM_CH0_CNT	0000 0008$_H$
TIM_CH0_ECNT	0000 000C$_H$
TIM_CH0_CNTS	0000 0010$_H$
TIM_CH0_TDUC	0000 0014$_H$
TIM_CH0_TDUV	0000 0018$_H$
TIM_CH0_FLT_RE	0000 001C$_H$
TIM_CH0_FLT_FE	0000 0020$_H$
TIM_CH0_CTRL	0000 0024$_H$
TIM_CH0_ECTRL	0000 0028$_H$
TIM_CH0_IRQ_NOTIFY	0000 002C$_H$
TIM_CH0_IRQ_EN	0000 0030$_H$
TIM_CH0_IRQ_FORCINT	0000 0034$_H$
TIM_CH0_IRQ_MODE	0000 0038$_H$
TIM_CH0_EIRQ_EN	0000 003C$_H$
TIM_INP_VAL	0000 0074$_H$
TIM_IN_SRC	0000 0078$_H$
TIM_RST	0000 007C$_H$
TIM_CH1_GPR0	0000 0080$_H$
TIM_CH1_GPR1	0000 0084$_H$
TIM_CH1_CNT	0000 0088$_H$
TIM_CH1_ECNT	0000 008C$_H$
TIM_CH1_CNTS	0000 0090$_H$
TIM_CH1_TDUC	0000 0094$_H$
TIM_CH1_TDUV	0000 0098$_H$
TIM_CH1_FLT_RE	0000 009C$_H$
TIM_CH1_FLT_FE	0000 00A0$_H$
TIM_CH1_CTRL	0000 00A4$_H$

Table 38.518 MCS master interface address map (2/16)

Register Label	Register Address
TIM_CH1_ECTRL	0000 00A8$_H$
TIM_CH1_IRQ_NOTIFY	0000 00AC$_H$
TIM_CH1_IRQ_EN	0000 00B0$_H$
TIM_CH1_IRQ_FORCINT	0000 00B4$_H$
TIM_CH1_IRQ_MODE	0000 00B8$_H$
TIM_CH1_EIRQ_EN	0000 00BC$_H$
TIM_CH2_GPR0	0000 0100$_H$
TIM_CH2_GPR1	0000 0104$_H$
TIM_CH2_CNT	0000 0108$_H$
TIM_CH2_ECNT	0000 010C$_H$
TIM_CH2_CNTS	0000 0110$_H$
TIM_CH2_TDUC	0000 0114$_H$
TIM_CH2_TDUV	0000 0118$_H$
TIM_CH2_FLT_RE	0000 011C$_H$
TIM_CH2_FLT_FE	0000 0120$_H$
TIM_CH2_CTRL	0000 0124$_H$
TIM_CH2_ECTRL	0000 0128$_H$
TIM_CH2_IRQ_NOTIFY	0000 012C$_H$
TIM_CH2_IRQ_EN	0000 0130$_H$
TIM_CH2_IRQ_FORCINT	0000 0134$_H$
TIM_CH2_IRQ_MODE	0000 0138$_H$
TIM_CH2_EIRQ_EN	0000 013C$_H$
TIM_CH3_GPR0	0000 0180$_H$
TIM_CH3_GPR1	0000 0184$_H$
TIM_CH3_CNT	0000 0188$_H$
TIM_CH3_ECNT	0000 018C$_H$
TIM_CH3_CNTS	0000 0190$_H$
TIM_CH3_TDUC	0000 0194$_H$
TIM_CH3_TDUV	0000 0198$_H$
TIM_CH3_FLT_RE	0000 019C$_H$
TIM_CH3_FLT_FE	0000 01A0$_H$
TIM_CH3_CTRL	0000 01A4$_H$
TIM_CH3_ECTRL	0000 01A8$_H$
TIM_CH3_IRQ_NOTIFY	0000 01AC$_H$
TIM_CH3_IRQ_EN	0000 01B0$_H$
TIM_CH3_IRQ_FORCINT	0000 01B4$_H$
TIM_CH3_IRQ_MODE	0000 01B8$_H$
TIM_CH3_EIRQ_EN	0000 01BC$_H$
TIM_CH4_GPR0	0000 0200$_H$
TIM_CH4_GPR1	0000 0204$_H$
TIM_CH4_CNT	0000 0208$_H$
TIM_CH4_ECNT	0000 020C$_H$
TIM_CH4_CNTS	0000 0210$_H$
TIM_CH4_TDUC	0000 0214$_H$

Table 38.518 MCS master interface address map (3/16)

Register Label	Register Address
TIM_CH4_TDUV	0000 0218$_H$
TIM_CH4_FLT_RE	0000 021C$_H$
TIM_CH4_FLT_FE	0000 0220$_H$
TIM_CH4_CTRL	0000 0224$_H$
TIM_CH4_ECTRL	0000 0228$_H$
TIM_CH4_IRQ_NOTIFY	0000 022C$_H$
TIM_CH4_IRQ_EN	0000 0230$_H$
TIM_CH4_IRQ_FORCINT	0000 0234$_H$
TIM_CH4_IRQ_MODE	0000 0238$_H$
TIM_CH4_EIRQ_EN	0000 023C$_H$
TIM_CH5_GPR0	0000 0280$_H$
TIM_CH5_GPR1	0000 0284$_H$
TIM_CH5_CNT	0000 0288$_H$
TIM_CH5_ECNT	0000 028C$_H$
TIM_CH5_CNTS	0000 0290$_H$
TIM_CH5_TDUC	0000 0294$_H$
TIM_CH5_TDUV	0000 0298$_H$
TIM_CH5_FLT_RE	0000 029C$_H$
TIM_CH5_FLT_FE	0000 02A0$_H$
TIM_CH5_CTRL	0000 02A4$_H$
TIM_CH5_ECTRL	0000 02A8$_H$
TIM_CH5_IRQ_NOTIFY	0000 02AC$_H$
TIM_CH5_IRQ_EN	0000 02B0$_H$
TIM_CH5_IRQ_FORCINT	0000 02B4$_H$
TIM_CH5_IRQ_MODE	0000 02B8$_H$
TIM_CH5_EIRQ_EN	0000 02BC$_H$
TIM_CH6_GPR0	0000 0300$_H$
TIM_CH6_GPR1	0000 0304$_H$
TIM_CH6_CNT	0000 0308$_H$
TIM_CH6_ECNT	0000 030C$_H$
TIM_CH6_CNTS	0000 0310$_H$
TIM_CH6_TDUC	0000 0314$_H$
TIM_CH6_TDUV	0000 0318$_H$
TIM_CH6_FLT_RE	0000 031C$_H$
TIM_CH6_FLT_FE	0000 0320$_H$
TIM_CH6_CTRL	0000 0324$_H$
TIM_CH6_ECTRL	0000 0328$_H$
TIM_CH6_IRQ_NOTIFY	0000 032C$_H$
TIM_CH6_IRQ_EN	0000 0330$_H$
TIM_CH6_IRQ_FORCINT	0000 0334$_H$
TIM_CH6_IRQ_MODE	0000 0338$_H$
TIM_CH6_EIRQ_EN	0000 033C$_H$
TIM_CH7_GPR0	0000 0380$_H$
TIM_CH7_GPR1	0000 0384$_H$

Table 38.518 MCS master interface address map (4/16)

Register Label	Register Address
TIM_CH7_CNT	0000 0388$_H$
TIM_CH7_ECNT	0000 038C$_H$
TIM_CH7_CNTS	0000 0390$_H$
TIM_CH7_TDUC	0000 0394$_H$
TIM_CH7_TDUV	0000 0398$_H$
TIM_CH7_FLT_RE	0000 039C$_H$
TIM_CH7_FLT_FE	0000 03A0$_H$
TIM_CH7_CTRL	0000 03A4$_H$
TIM_CH7_ECTRL	0000 03A8$_H$
TIM_CH7_IRQ_NOTIFY	0000 03AC$_H$
TIM_CH7_IRQ_EN	0000 03B0$_H$
TIM_CH7_IRQ_FORCINT	0000 03B4$_H$
TIM_CH7_IRQ_MODE	0000 03B8$_H$
TIM_CH7_EIRQ_EN	0000 03BC$_H$
ATOM_CH0_RDADDR	0000 1800$_H$
ATOM_CH0_CTRL	0000 1804$_H$
ATOM_CH0_SR0	0000 1808$_H$
ATOM_CH0_SR1	0000 180C$_H$
ATOM_CH0_CM0	0000 1810$_H$
ATOM_CH0_CM1	0000 1814$_H$
ATOM_CH0_CN0	0000 1818$_H$
ATOM_CH0_STAT	0000 181C$_H$
ATOM_CH0_IRQ_NOTIFY	0000 1820$_H$
ATOM_CH0_IRQ_EN	0000 1824$_H$
ATOM_CH0_IRQ_FORCINT	0000 1828$_H$
ATOM_CH0_IRQ_MODE	0000 182C$_H$
ATOM_AGC_GLB_CTRL	0000 1840$_H$
ATOM_AGC_ENDIS_CTRL	0000 1844$_H$
ATOM_AGC_ENDIS_STAT	0000 1848$_H$
ATOM_AGC_ACT_TB	0000 184C$_H$
ATOM_AGC_OUTEN_CTRL	0000 1850$_H$
ATOM_AGC_OUTEN_STAT	0000 1854$_H$
ATOM_AGC_FUPD_CTRL	0000 1858$_H$
ATOM_AGC_INT_TRIG	0000 185C$_H$
ATOM_CH1_RDADDR	0000 1880$_H$
ATOM_CH1_CTRL	0000 1884$_H$
ATOM_CH1_SR0	0000 1888$_H$
ATOM_CH1_SR1	0000 188C$_H$
ATOM_CH1_CM0	0000 1890$_H$
ATOM_CH1_CM1	0000 1894$_H$
ATOM_CH1_CN0	0000 1898$_H$
ATOM_CH1_STAT	0000 189C$_H$
ATOM_CH1_IRQ_NOTIFY	0000 18A0$_H$
ATOM_CH1_IRQ_EN	0000 18A4$_H$

Table 38.518 MCS master interface address map (5/16)

Register Label	Register Address
ATOM_CH1_IRQ_FORCINT	0000 18A8$_H$
ATOM_CH1_IRQ_MODE	0000 18AC$_H$
ATOM_CH2_RDADDR	0000 1900$_H$
ATOM_CH2_CTRL	0000 1904$_H$
ATOM_CH2_SR0	0000 1908$_H$
ATOM_CH2_SR1	0000 190C$_H$
ATOM_CH2_CM0	0000 1910$_H$
ATOM_CH2_CM1	0000 1914$_H$
ATOM_CH2_CN0	0000 1918$_H$
ATOM_CH2_STAT	0000 191C$_H$
ATOM_CH2_IRQ_NOTIFY	0000 1920$_H$
ATOM_CH2_IRQ_EN	0000 1924$_H$
ATOM_CH2_IRQ_FORCINT	0000 1928$_H$
ATOM_CH2_IRQ_MODE	0000 192C$_H$
ATOM_CH3_RDADDR	0000 1980$_H$
ATOM_CH3_CTRL	0000 1984$_H$
ATOM_CH3_SR0	0000 1988$_H$
ATOM_CH3_SR1	0000 198C$_H$
ATOM_CH3_CM0	0000 1990$_H$
ATOM_CH3_CM1	0000 1994$_H$
ATOM_CH3_CN0	0000 1998$_H$
ATOM_CH3_STAT	0000 199C$_H$
ATOM_CH3_IRQ_NOTIFY	0000 19A0$_H$
ATOM_CH3_IRQ_EN	0000 19A4$_H$
ATOM_CH3_IRQ_FORCINT	0000 19A8$_H$
ATOM_CH3_IRQ_MODE	0000 19AC$_H$
ATOM_CH4_RDADDR	0000 1A00$_H$
ATOM_CH4_CTRL	0000 1A04$_H$
ATOM_CH4_SR0	0000 1A08$_H$
ATOM_CH4_SR1	0000 1A0C$_H$
ATOM_CH4_CM0	0000 1A10$_H$
ATOM_CH4_CM1	0000 1A14$_H$
ATOM_CH4_CN0	0000 1A18$_H$
ATOM_CH4_STAT	0000 1A1C$_H$
ATOM_CH4_IRQ_NOTIFY	0000 1A20$_H$
ATOM_CH4_IRQ_EN	0000 1A24$_H$
ATOM_CH4_IRQ_FORCINT	0000 1A28$_H$
ATOM_CH4_IRQ_MODE	0000 1A2C$_H$
ATOM_CH5_RDADDR	0000 1A80$_H$
ATOM_CH5_CTRL	0000 1A84$_H$
ATOM_CH5_SR0	0000 1A88$_H$
ATOM_CH5_SR1	0000 1A8C$_H$
ATOM_CH5_CM0	0000 1A90$_H$
ATOM_CH5_CM1	0000 1A94$_H$

Table 38.518 MCS master interface address map (6/16)

Register Label	Register Address
ATOM_CH5_CN0	0000 1A98$_H$
ATOM_CH5_STAT	0000 1A9C$_H$
ATOM_CH5_IRQ_NOTIFY	0000 1AA0$_H$
ATOM_CH5_IRQ_EN	0000 1AA4$_H$
ATOM_CH5_IRQ_FORCINT	0000 1AA8$_H$
ATOM_CH5_IRQ_MODE	0000 1AAC$_H$
ATOM_CH6_RDADDR	0000 1B00$_H$
ATOM_CH6_CTRL	0000 1B04$_H$
ATOM_CH6_SR0	0000 1B08$_H$
ATOM_CH6_SR1	0000 1B0C$_H$
ATOM_CH6_CM0	0000 1B10$_H$
ATOM_CH6_CM1	0000 1B14$_H$
ATOM_CH6_CN0	0000 1B18$_H$
ATOM_CH6_STAT	0000 1B1C$_H$
ATOM_CH6_IRQ_NOTIFY	0000 1B20$_H$
ATOM_CH6_IRQ_EN	0000 1B24$_H$
ATOM_CH6_IRQ_FORCINT	0000 1B28$_H$
ATOM_CH6_IRQ_MODE	0000 1B2C$_H$
ATOM_CH7_RDADDR	0000 1B80$_H$
ATOM_CH7_CTRL	0000 1B84$_H$
ATOM_CH7_SR0	0000 1B88$_H$
ATOM_CH7_SR1	0000 1B8C$_H$
ATOM_CH7_CM0	0000 1B90$_H$
ATOM_CH7_CM1	0000 1B94$_H$
ATOM_CH7_CN0	0000 1B98$_H$
ATOM_CH7_STAT	0000 1B9C$_H$
ATOM_CH7_IRQ_NOTIFY	0000 1BA0$_H$
ATOM_CH7_IRQ_EN	0000 1BA4$_H$
ATOM_CH7_IRQ_FORCINT	0000 1BA8$_H$
ATOM_CH7_IRQ_MODE	0000 1BAC$_H$
CDTM_DTM4_CTRL	0000 3100$_H$
CDTM_DTM4_CH_CTRL1	0000 3104$_H$
CDTM_DTM4_CH_CTRL2	0000 3108$_H$
CDTM_DTM4_CH_CTRL2_SR	0000 310C$_H$
CDTM_DTM4_PS_CTRL	0000 3110$_H$
CDTM_DTM4_CH0_DTV	0000 3114$_H$
CDTM_DTM4_CH1_DTV	0000 3118$_H$
CDTM_DTM4_CH2_DTV	0000 311C$_H$
CDTM_DTM4_CH3_DTV	0000 3120$_H$
CDTM_DTM4_CH_SR	0000 3124$_H$
CDTM_DTM4_CH_CTRL3	0000 3128$_H$
CDTM_DTM5_CTRL	0000 3140$_H$
CDTM_DTM5_CH_CTRL1	0000 3144$_H$
CDTM_DTM5_CH_CTRL2	0000 3148$_H$

Table 38.518 MCS master interface address map (7/16)

Register Label	Register Address
CDTM_DTM5_CH_CTRL2_SR	0000 314C$_H$
CDTM_DTM5_PS_CTRL	0000 3150$_H$
CDTM_DTM5_CH0_DTV	0000 3154$_H$
CDTM_DTM5_CH1_DTV	0000 3158$_H$
CDTM_DTM5_CH2_DTV	0000 315C$_H$
CDTM_DTM5_CH3_DTV	0000 3160$_H$
CDTM_DTM5_CH_SR	0000 3164$_H$
CDTM_DTM5_CH_CTRL3	0000 3168$_H$
MCS_CH0_R0	0000 3800$_H$
MCS_CH0_R1	0000 3804$_H$
MCS_CH0_R2	0000 3808$_H$
MCS_CH0_R3	0000 380C$_H$
MCS_CH0_R4	0000 3810$_H$
MCS_CH0_R5	0000 3814$_H$
MCS_CH0_R6	0000 3818$_H$
MCS_CH0_R7	0000 381C$_H$
MCS_CH0_CTRL	0000 3820$_H$
MCS_CH0_ACB	0000 3824$_H$
MCS_CTRG	0000 3828$_H$
MCS_STRG	0000 382C$_H$
MCS_CH0_MHB	0000 383C$_H$
MCS_CH0_PC	0000 3840$_H$
MCS_CH0_IRQ_NOTIFY	0000 3844$_H$
MCS_CH0_IRQ_EN	0000 3848$_H$
MCS_CH0_IRQ_FORCINT	0000 384C$_H$
MCS_CH0_IRQ_MODE	0000 3850$_H$
MCS_CH0_EIRQ_EN	0000 3854$_H$
MCS_REG_PROT	0000 3860$_H$
MCS_CTRL_STAT	0000 3864$_H$
MCS_RESET	0000 3868$_H$
MCS_CAT	0000 386C$_H$
MCS_CWT	0000 3870$_H$
MCS_ERR	0000 387C$_H$
MCS_CH1_R0	0000 3880$_H$
MCS_CH1_R1	0000 3884$_H$
MCS_CH1_R2	0000 3888$_H$
MCS_CH1_R3	0000 388C$_H$
MCS_CH1_R4	0000 3890$_H$
MCS_CH1_R5	0000 3894$_H$
MCS_CH1_R6	0000 3898$_H$
MCS_CH1_R7	0000 389C$_H$
MCS_CH1_CTRL	0000 38A0$_H$
MCS_CH1_ACB	0000 38A4$_H$
MCS_CH1_MHB	0000 38BC$_H$

Table 38.518 MCS master interface address map (8/16)

Register Label	Register Address
MCS_CH1_PC	0000 38C0$_H$
MCS_CH1_IRQ_NOTIFY	0000 38C4$_H$
MCS_CH1_IRQ_EN	0000 38C8$_H$
MCS_CH1_IRQ_FORCINT	0000 38CC$_H$
MCS_CH1_IRQ_MODE	0000 38D0$_H$
MCS_CH1_EIRQ_EN	0000 38D4$_H$
MCS_CH2_R0	0000 3900$_H$
MCS_CH2_R1	0000 3904$_H$
MCS_CH2_R2	0000 3908$_H$
MCS_CH2_R3	0000 390C$_H$
MCS_CH2_R4	0000 3910$_H$
MCS_CH2_R5	0000 3914$_H$
MCS_CH2_R6	0000 3918$_H$
MCS_CH2_R7	0000 391C$_H$
MCS_CH2_CTRL	0000 3920$_H$
MCS_CH2_ACB	0000 3924$_H$
MCS_CH2_MHB	0000 393C$_H$
MCS_CH2_PC	0000 3940$_H$
MCS_CH2_IRQ_NOTIFY	0000 3944$_H$
MCS_CH2_IRQ_EN	0000 3948$_H$
MCS_CH2_IRQ_FORCINT	0000 394C$_H$
MCS_CH2_IRQ_MODE	0000 3950$_H$
MCS_CH2_EIRQ_EN	0000 3954$_H$
MCS_CH3_R0	0000 3980$_H$
MCS_CH3_R1	0000 3984$_H$
MCS_CH3_R2	0000 3988$_H$
MCS_CH3_R3	0000 398C$_H$
MCS_CH3_R4	0000 3990$_H$
MCS_CH3_R5	0000 3994$_H$
MCS_CH3_R6	0000 3998$_H$
MCS_CH3_R7	0000 399C$_H$
MCS_CH3_CTRL	0000 39A0$_H$
MCS_CH3_ACB	0000 39A4$_H$
MCS_CH3_MHB	0000 39BC$_H$
MCS_CH3_PC	0000 39C0$_H$
MCS_CH3_IRQ_NOTIFY	0000 39C4$_H$
MCS_CH3_IRQ_EN	0000 39C8$_H$
MCS_CH3_IRQ_FORCINT	0000 39CC$_H$
MCS_CH3_IRQ_MODE	0000 39D0$_H$
MCS_CH3_EIRQ_EN	0000 39D4$_H$
MCS_CH4_R0	0000 3A00$_H$
MCS_CH4_R1	0000 3A04$_H$
MCS_CH4_R2	0000 3A08$_H$
MCS_CH4_R3	0000 3A0C$_H$

Table 38.518 MCS master interface address map (9/16)

Register Label	Register Address
MCS_CH4_R4	0000 3A10$_H$
MCS_CH4_R5	0000 3A14$_H$
MCS_CH4_R6	0000 3A18$_H$
MCS_CH4_R7	0000 3A1C$_H$
MCS_CH4_CTRL	0000 3A20$_H$
MCS_CH4_ACB	0000 3A24$_H$
MCS_CH4_MHB	0000 3A3C$_H$
MCS_CH4_PC	0000 3A40$_H$
MCS_CH4_IRQ_NOTIFY	0000 3A44$_H$
MCS_CH4_IRQ_EN	0000 3A48$_H$
MCS_CH4_IRQ_FORCINT	0000 3A4C$_H$
MCS_CH4_IRQ_MODE	0000 3A50$_H$
MCS_CH4_EIRQ_EN	0000 3A54$_H$
MCS_CH5_R0	0000 3A80$_H$
MCS_CH5_R1	0000 3A84$_H$
MCS_CH5_R2	0000 3A88$_H$
MCS_CH5_R3	0000 3A8C$_H$
MCS_CH5_R4	0000 3A90$_H$
MCS_CH5_R5	0000 3A94$_H$
MCS_CH5_R6	0000 3A98$_H$
MCS_CH5_R7	0000 3A9C$_H$
MCS_CH5_CTRL	0000 3AA0$_H$
MCS_CH5_ACB	0000 3AA4$_H$
MCS_CH5_MHB	0000 3ABC$_H$
MCS_CH5_PC	0000 3AC0$_H$
MCS_CH5_IRQ_NOTIFY	0000 3AC4$_H$
MCS_CH5_IRQ_EN	0000 3AC8$_H$
MCS_CH5_IRQ_FORCINT	0000 3ACC$_H$
MCS_CH5_IRQ_MODE	0000 3AD0$_H$
MCS_CH5_EIRQ_EN	0000 3AD4$_H$
MCS_CH6_R0	0000 3B00$_H$
MCS_CH6_R1	0000 3B04$_H$
MCS_CH6_R2	0000 3B08$_H$
MCS_CH6_R3	0000 3B0C$_H$
MCS_CH6_R4	0000 3B10$_H$
MCS_CH6_R5	0000 3B14$_H$
MCS_CH6_R6	0000 3B18$_H$
MCS_CH6_R7	0000 3B1C$_H$
MCS_CH6_CTRL	0000 3B20$_H$
MCS_CH6_ACB	0000 3B24$_H$
MCS_CH6_MHB	0000 3B3C$_H$
MCS_CH6_PC	0000 3B40$_H$
MCS_CH6_IRQ_NOTIFY	0000 3B44$_H$
MCS_CH6_IRQ_EN	0000 3B48$_H$

Table 38.518 MCS master interface address map (10/16)

Register Label	Register Address
MCS_CH6_IRQ_FORCINT	0000 3B4C$_H$
MCS_CH6_IRQ_MODE	0000 3B50$_H$
MCS_CH6_EIRQ_EN	0000 3B54$_H$
MCS_CH7_R0	0000 3B80$_H$
MCS_CH7_R1	0000 3B84$_H$
MCS_CH7_R2	0000 3B88$_H$
MCS_CH7_R3	0000 3B8C$_H$
MCS_CH7_R4	0000 3B90$_H$
MCS_CH7_R5	0000 3B94$_H$
MCS_CH7_R6	0000 3B98$_H$
MCS_CH7_R7	0000 3B9C$_H$
MCS_CH7_CTRL	0000 3BA0$_H$
MCS_CH7_ACB	0000 3BA4$_H$
MCS_CH7_MHB	0000 3BBC$_H$
MCS_CH7_PC	0000 3BC0$_H$
MCS_CH7_IRQ_NOTIFY	0000 3BC4$_H$
MCS_CH7_IRQ_EN	0000 3BC8$_H$
MCS_CH7_IRQ_FORCINT	0000 3BCC$_H$
MCS_CH7_IRQ_MODE	0000 3BD0$_H$
MCS_CH7_EIRQ_EN	0000 3BD4$_H$
CCM_ARP0_CTRL	0000 5200$_H$
CCM_ARP0_PROT	0000 5204$_H$
CCM_ARP1_CTRL	0000 5208$_H$
CCM_ARP1_PROT	0000 520C$_H$
CCM_ARP2_CTRL	0000 5210$_H$
CCM_ARP2_PROT	0000 5214$_H$
CCM_ARP3_CTRL	0000 5218$_H$
CCM_ARP3_PROT	0000 521C$_H$
CCM_ARP4_CTRL	0000 5220$_H$
CCM_ARP4_PROT	0000 5224$_H$
CCM_ARP5_CTRL	0000 5228$_H$
CCM_ARP5_PROT	0000 522C$_H$
CCM_ARP6_CTRL	0000 5230$_H$
CCM_ARP6_PROT	0000 5234$_H$
CCM_ARP7_CTRL	0000 5238$_H$
CCM_ARP7_PROT	0000 523C$_H$
CCM_ARP8_CTRL	0000 5240$_H$
CCM_ARP8_PROT	0000 5244$_H$
CCM_ARP9_CTRL	0000 5248$_H$
CCM_ARP9_PROT	0000 524C$_H$
CCM_AEIM_STA	0000 53D8$_H$
CCM_HW_CONF	0000 53DC$_H$
CCM_TIM_AUX_IN_SRC	0000 53E0$_H$
CCM_EXT_CAP_EN	0000 53E4$_H$

Table 38.518 MCS master interface address map (11/16)

Register Label	Register Address
CCM_ATOM_OUT	0000 53EC$_H$
CCM_CMU_CLK_CFG	0000 53F0$_H$
CCM_CFG	0000 53F8$_H$
CCM_PROT	0000 53FC$_H$
ADC_CH0_DATA	0000 5400$_H$
ADC_CH0_STA	0000 5404$_H$
ADC_CH1_DATA	0000 5408$_H$
ADC_CH1_STA	0000 540C$_H$
ADC_CH2_DATA	0000 5410$_H$
ADC_CH2_STA	0000 5414$_H$
ADC_CH3_DATA	0000 5418$_H$
ADC_CH3_STA	0000 541C$_H$
ADC_CH4_DATA	0000 5420$_H$
ADC_CH4_STA	0000 5424$_H$
ADC_CH5_DATA	0000 5428$_H$
ADC_CH5_STA	0000 542C$_H$
ADC_CH6_DATA	0000 5430$_H$
ADC_CH6_STA	0000 5434$_H$
ADC_CH7_DATA	0000 5438$_H$
ADC_CH7_STA	0000 543C$_H$
ADC_CH8_DATA	0000 5440$_H$
ADC_CH8_STA	0000 5444$_H$
ADC_CH9_DATA	0000 5448$_H$
ADC_CH9_STA	0000 544C$_H$
ADC_CH10_DATA	0000 5450$_H$
ADC_CH10_STA	0000 5454$_H$
ADC_CH11_DATA	0000 5458$_H$
ADC_CH11_STA	0000 545C$_H$
ADC_CH12_DATA	0000 5460$_H$
ADC_CH12_STA	0000 5464$_H$
ADC_CH13_DATA	0000 5468$_H$
ADC_CH13_STA	0000 546C$_H$
ADC_CH14_DATA	0000 5470$_H$
ADC_CH14_STA	0000 5474$_H$
ADC_CH15_DATA	0000 5478$_H$
ADC_CH15_STA	0000 547C$_H$
ADC_CH16_DATA	0000 5480$_H$
ADC_CH16_STA	0000 5484$_H$
ADC_CH17_DATA	0000 5488$_H$
ADC_CH17_STA	0000 548C$_H$
ADC_CH18_DATA	0000 5490$_H$
ADC_CH18_STA	0000 5494$_H$
ADC_CH19_DATA	0000 5498$_H$
ADC_CH19_STA	0000 549C$_H$

Table 38.518 MCS master interface address map (12/16)

Register Label	Register Address
ADC_CH20_DATA	0000 54A0$_H$
ADC_CH20_STA	0000 54A4$_H$
ADC_CH21_DATA	0000 54A8$_H$
ADC_CH21_STA	0000 54AC$_H$
ADC_CH22_DATA	0000 54B0$_H$
ADC_CH22_STA	0000 54B4$_H$
ADC_CH23_DATA	0000 54B8$_H$
ADC_CH23_STA	0000 54BC$_H$
ADC_CH24_DATA	0000 54C0$_H$
ADC_CH24_STA	0000 54C4$_H$
ADC_CH25_DATA	0000 54C8$_H$
ADC_CH25_STA	0000 54CC$_H$
ADC_CH26_DATA	0000 54D0$_H$
ADC_CH26_STA	0000 54D4$_H$
ADC_CH27_DATA	0000 54D8$_H$
ADC_CH27_STA	0000 54DC$_H$
ADC_CH28_DATA	0000 54E0$_H$
ADC_CH28_STA	0000 54E4$_H$
ADC_CH29_DATA	0000 54E8$_H$
ADC_CH29_STA	0000 54EC$_H$
ADC_CH30_DATA	0000 54F0$_H$
ADC_CH30_STA	0000 54F4$_H$
ADC_CH31_DATA	0000 54F8$_H$
ADC_CH31_STA	0000 54FC$_H$
ADC_CH32_DATA	0000 5500$_H$
ADC_CH32_STA	0000 5504$_H$
ADC_CH33_DATA	0000 5508$_H$
ADC_CH33_STA	0000 550C$_H$
ADC_CH34_DATA	0000 5510$_H$
ADC_CH34_STA	0000 5514$_H$
ADC_CH35_DATA	0000 5518$_H$
ADC_CH35_STA	0000 551C$_H$
ADC_CH36_DATA	0000 5520$_H$
ADC_CH36_STA	0000 5524$_H$
ADC_CH37_DATA	0000 5528$_H$
ADC_CH37_STA	0000 552C$_H$
ADC_CH38_DATA	0000 5530$_H$
ADC_CH38_STA	0000 5534$_H$
ADC_CH39_DATA	0000 5538$_H$
ADC_CH39_STA	0000 553C$_H$
ADC_CH40_DATA	0000 5540$_H$
ADC_CH40_STA	0000 5544$_H$
ADC_CH41_DATA	0000 5548$_H$
ADC_CH41_STA	0000 554C$_H$

Table 38.518 MCS master interface address map (13/16)

Register Label	Register Address
ADC_CH42_DATA	0000 5550$_H$
ADC_CH42_STA	0000 5554$_H$
ADC_CH43_DATA	0000 5558$_H$
ADC_CH43_STA	0000 555C$_H$
ADC_CH44_DATA	0000 5560$_H$
ADC_CH44_STA	0000 5564$_H$
ADC_CH45_DATA	0000 5568$_H$
ADC_CH45_STA	0000 556C$_H$
ADC_CH46_DATA	0000 5570$_H$
ADC_CH46_STA	0000 5574$_H$
ADC_CH47_DATA	0000 5578$_H$
ADC_CH47_STA	0000 557C$_H$
ADC_CH48_DATA	0000 5580$_H$
ADC_CH48_STA	0000 5584$_H$
ADC_CH49_DATA	0000 5588$_H$
ADC_CH49_STA	0000 558C$_H$
ADC_CH50_DATA	0000 5590$_H$
ADC_CH50_STA	0000 5594$_H$
ADC_CH51_DATA	0000 5598$_H$
ADC_CH51_STA	0000 559C$_H$
ADC_CH52_DATA	0000 55A0$_H$
ADC_CH52_STA	0000 55A4$_H$
ADC_CH53_DATA	0000 55A8$_H$
ADC_CH53_STA	0000 55AC$_H$
ADC_CH54_DATA	0000 55B0$_H$
ADC_CH54_STA	0000 55B4$_H$
ADC_CH55_DATA	0000 55B8$_H$
ADC_CH55_STA	0000 55BC$_H$
ADC_CH56_DATA	0000 55C0$_H$
ADC_CH56_STA	0000 55C4$_H$
ADC_CH57_DATA	0000 55C8$_H$
ADC_CH57_STA	0000 55CC$_H$
ADC_CH58_DATA	0000 55D0$_H$
ADC_CH58_STA	0000 55D4$_H$
ADC_CH59_DATA	0000 55D8$_H$
ADC_CH59_STA	0000 55DC$_H$
ADC_CH60_DATA	0000 55E0$_H$
ADC_CH60_STA	0000 55E4$_H$
ADC_CH61_DATA	0000 55E8$_H$
ADC_CH61_STA	0000 55EC$_H$
ADC_CH62_DATA	0000 55F0$_H$
ADC_CH62_STA	0000 55F4$_H$
ADC_CH63_DATA	0000 55F8$_H$
ADC_CH63_STA	0000 55FC$_H$

Table 38.518 MCS master interface address map (14/16)

Register Label	Register Address
CMU_CLK_EN	0000 7000$_H$
CMU_GCLK_NUM	0000 7004$_H$
CMU_GCLK_DEN	0000 7008$_H$
CMU_CLK_0_CTRL	0000 700C$_H$
CMU_CLK_1_CTRL	0000 7010$_H$
CMU_CLK_2_CTRL	0000 7014$_H$
CMU_CLK_3_CTRL	0000 7018$_H$
CMU_CLK_4_CTRL	0000 701C$_H$
CMU_CLK_5_CTRL	0000 7020$_H$
CMU_CLK_6_CTRL	0000 7024$_H$
CMU_CLK_7_CTRL	0000 7028$_H$
CMU_ECLK_0_NUM	0000 702C$_H$
CMU_ECLK_0_DEN	0000 7030$_H$
CMU_ECLK_1_NUM	0000 7034$_H$
CMU_ECLK_1_DEN	0000 7038$_H$
CMU_ECLK_2_NUM	0000 703C$_H$
CMU_ECLK_2_DEN	0000 7040$_H$
CMU_FXCLK_CTRL	0000 7044$_H$
CMU_GLB_CTRL	0000 7048$_H$
CMU_CLK_CTRL	0000 704C$_H$
TBU_CHEN	0000 7080$_H$
TBU_CH0_CTRL	0000 7084$_H$
TBU_CH0_BASE	0000 7088$_H$
TBU_CH1_CTRL	0000 708C$_H$
TBU_CH1_BASE	0000 7090$_H$
TBU_CH2_CTRL	0000 7094$_H$
TBU_CH2_BASE	0000 7098$_H$
ARU_ACCESS	0000 7180$_H$
ARU_DATA_H	0000 7184$_H$
ARU_DATA_L	0000 7188$_H$
ARU_DBG_ACCESS0	0000 718C$_H$
ARU_DBG_DATA0_H	0000 7190$_H$
ARU_DBG_DATA0_L	0000 7194$_H$
ARU_DBG_ACCESS1	0000 7198$_H$
ARU_DBG_DATA1_H	0000 719C$_H$
ARU_DBG_DATA1_L	0000 71A0$_H$
ARU_IRQ_NOTIFY	0000 71A4$_H$
ARU_IRQ_EN	0000 71A8$_H$
ARU_IRQ_FORCINT	0000 71AC$_H$
ARU_IRQ_MODE	0000 71B0$_H$
ARU_CADDR_END	0000 71B4$_H$
ARU_reserved	0000 71B8$_H$
ARU_CTRL	0000 71BC$_H$
ARU_0_DYN_CTRL	0000 71C0$_H$

Table 38.518 MCS master interface address map (15/16)

Register Label	Register Address
ARU_1_DYN_CTRL	0000 71C4$_H$
ARU_0_DYN_ROUTE_LOW	0000 71C8$_H$
ARU_1_DYN_ROUTE_LOW	0000 71CC$_H$
ARU_0_DYN_ROUTE_HIGH	0000 71D0$_H$
ARU_1_DYN_ROUTE_HIGH	0000 71D4$_H$
ARU_0_DYN_ROUTE_SR_LOW	0000 71D8$_H$
ARU_1_DYN_ROUTE_SR_LOW	0000 71DC$_H$
ARU_0_DYN_ROUTE_SR_HIGH	0000 71E0$_H$
ARU_1_DYN_ROUTE_SR_HIGH	0000 71E4$_H$
ARU_0_DYN_RDADDR	0000 71E8$_H$
ARU_1_DYN_RDADDR	0000 71EC$_H$
ARU_CADDR	0000 71FC$_H$
MCFG_CTRL	0000 7200$_H$
MON_STATUS	0000 7280$_H$
MON_ACTIVITY_0	0000 7284$_H$
MON_ACTIVITY_1	0000 7288$_H$
MON_ACTIVITY_MCS0	0000 728C$_H$
MON_ACTIVITY_MCS1	0000 7290$_H$
MON_ACTIVITY_MCS2	0000 7294$_H$
MON_ACTIVITY_MCS3	0000 7298$_H$
CMP_EN	0000 72C0$_H$
CMP_IRQ_NOTIFY	0000 72C4$_H$
CMP_IRQ_EN	0000 72C8$_H$
CMP_IRQ_FORCINT	0000 72CC$_H$
CMP_IRQ_MODE	0000 72D0$_H$
CMP_EIRQ_EN	0000 72D4$_H$
GTM_REV	0000 7300$_H$
GTM_RST	0000 7304$_H$
GTM_CTRL	0000 7308$_H$
GTM_AEI_ADDR_XPT	0000 730C$_H$
GTM_IRQ_NOTIFY	0000 7310$_H$
GTM_IRQ_EN	0000 7314$_H$
GTM_IRQ_FORCINT	0000 7318$_H$
GTM_IRQ_MODE	0000 731C$_H$
GTM_EIRQ_EN	0000 7320$_H$
LEG_GTM_HW_CONF	0000 7324$_H$
GTM_AEI_STA_XPT	0000 732C$_H$
LEG_GTM_TIM0_AUX_IN_SRC	0000 7340$_H$
LEG_GTM_TIM1_AUX_IN_SRC	0000 7344$_H$
LEG_GTM_TIM2_AUX_IN_SRC	0000 7348$_H$
LEG_GTM_TIM3_AUX_IN_SRC	0000 734C$_H$
LEG_GTM_EXT_CAP_EN_0	0000 735C$_H$
LEG_GTM_EXT_CAP_EN_1	0000 7360$_H$
LEG_GTM_EXT_CAP_EN_2	0000 7364$_H$

Table 38.518 MCS master interface address map (16/16)

Register Label	Register Address
LEG_GTM_EXT_CAP_EN_3	0000 7368_H
LEG_GTM_ATOM0_OUT	0000 7398_H
LEG_GTM_ATOM2_OUT	0000 $739C_H$
GTM_CLS_CLK_CFG	0000 $73B0_H$

38.28.9 Memory Address Ranges

Table 38.519 lists for each memory mapped MCS RAM region implemented in the GTM-IP the default address range configuration with ERM=0.

Table 38.519 Memory Address Ranges

Sub-module	Memory Address Ranges
MCS0_MEMORY	0003 8000_H to 0003 $AFFC_H$
MCS1_MEMORY	0004 0000_H to 0004 $2FFC_H$
MCS2_MEMORY	0004 8000_H to 0004 $AFFC_H$
MCS3_MEMORY	0005 0000_H to 0005 $2FFC_H$

38.28.10 ARU

38.28.10.1 ARU Write Address Overview

Table 38.520 ARU Write Addresses

Name	Address	MCS Write Index	Name	Address
ARU_ACCESS	000_H		ATOM [0 : 3]	
TIM [0 : 3]			ATOM0_WRADDR[0:7]	$11F_H$ to 126_H
TIM0_WRADDR[0:7]	001_H to 008_H		ATOM1_WRADDR[0:7]	127_H to $12E_H$
TIM1_WRADDR[0:7]	009_H to 010_H		ATOM2_WRADDR[0:7]	$12F_H$ to 136_H
TIM2_WRADDR[0:7]	011_H to 018_H		ATOM3_WRADDR[0:7]	137_H to $13E_H$
TIM3_WRADDR[0:7]	019_H to 020_H		unused	$13F_H$ to $1FD_H$
unused	021_H to 076_H		misc	
MCS [0 : 3]			ARU_EMPTY_ADDR	$1FE_H$
MCS0_WRADDR[0:23]	077_H to $08E_H$	0…23	ARU_FULL_ADDR	$1FF_H$
MCS1_WRADDR[0:23]	$08F_H$ to $0A6_H$	0…23		
MCS2_WRADDR[0:23]	$0A7_H$ to $0BE_H$	0…23		
MCS3_WRADDR[0:23]	$0BF_H$ to $0D6_H$	0…23		
unused	$0D7_H$ to $11E_H$			

38.28.10.2 ARU port partitioning

All GTM sub-modules which are reading from ARU can be connected to one of two ARU read ports. Therefore, it can be read from two different ARU addresses in parallel.

Table 38.521 ARU partitioning

	2 ARU ports	
	ARU-0	ARU-1
ATOM-0	x	
ATOM-1		x
ATOM-2	X	
ATOM-3		X
MCS-0	X	
MCS-1		X
MCS-2	X	
MCS-3		X

38.28.10.3 ARU read ID table

Each ARU connected data destination is defined by a combination of ARU port(ARU- 0 or ARU-1) and an ARU read ID. The two ARU counter are addressing two ARU read IDs in parallel. Depending on ARU mode both counter may have different values at different point in time (i.e. in dynamic routing mode). The maximum ARU round-trip time is determined by value of last ARU read ID. The following table describes the detailed addressing of GTM sub-modules by ARU read IDs.

Table 38.522 ARU read IDs (1/2)

ARU read ID (dec)	ARU-0	ARU-1	ARU read ID (dec)	ARU-0	ARU-1
0	reserved	reserved	37	MCS-2 channel 1	ATOM-1 channel 1
1	ARU-0	ARU-1	38	ATOM-0 channel 6	Unused
2	Unused	Unused	39	MCS-2 channel 2	ATOM-1 channel 2
3	Unused	Unused	40	ATOM-0 channel 7	Unused
4	Unused	Unused	41	MCS-2 channel 3	ATOM-1 channel 3
5	Unused	Unused	42	**ATOM-2** channel 0	Unused
6	Unused	Unused	43	MCS-2 channel 4	ATOM-1 channel 4
7	Unused	Unused	44	ATOM-2 channel 1	Unused
8	Unused	Unused	45	MCS-2 channel 5	ATOM-1 channel 5
9	Unused	Unused	46	ATOM-2 channel 2	Unused
10	Unused	Unused	47	MCS-2 channel 6	ATOM-1 channel 6
11	Unused	Unused	48	ATOM-2 channel 3	Unused
12	Unused	Unused	49	MCS-2 channel 7	ATOM-1 channel 7
13	Unused	Unused	50	ATOM-2 channel 4	Unused
14	Unused	Unused	51	Unused	**MCS-3** channel 0
15	Unused	Unused	52	ATOM-2 channel 5	Unused
16	Unused	Unused	53	Unused	MCS-3 channel 1
17	Unused	Unused	54	ATOM-2 channel 6	Unused
18	Unused	Unused	55	Unused	MCS-3 channel 2
19	**MCS-0** channel 0	**MCS-1** channel 0	56	ATOM-2 channel 7	Unused
20	Unused	Unused	57	Unused	MCS-3 channel 3
21	MCS-0 channel 1	MCS-1 channel 1	58	Unused	Unused
22	Unused	Unused	59	Unused	MCS-3 channel 4
23	MCS-0 channel 2	MCS-1 channel 2	60	Unused	Unused
24	Unused	Unused	61	Unused	MCS-3 channel 5
25	MCS-0 channel 3	MCS-1 channel 3	62	Unused	Unused
26	**ATOM-0** channel 0	Unused	63	Unused	MCS-3 channel 6
27	MCS-0 channel 4	MCS-1 channel 4	64	Unused	Unused
28	ATOM-0 channel 1	Unused	65	Unused	MCS-3 channel 7
29	MCS-0 channel 5	MCS-1 channel 5	66	Unused	**ATOM-3** channel 0
30	ATOM-0 channel 2	Unused	67	Unused	Unused
31	MCS-0 channel 6	MCS-1 channel 6	68	Unused	ATOM-3 channel 1
32	ATOM-0 channel 3	Unused	69	Unused	Unused
33	MCS-0 channel 7	MCS-1 channel 7	70	Unused	ATOM-3 channel 2
34	ATOM-0 channel 4	Unused	71	Unused	Unused
35	MCS-2 channel 0	**ATOM-1** channel 0	72	Unused	ATOM-3 channel 3
36	ATOM-0 channel 5	Unused	73	Unused	Unused

Table 38.522 ARU read IDs (2/2)

ARU read ID (dec)	ARU-0	ARU-1		ARU read ID (dec)	ARU-0	ARU-1
74	Unused	ATOM-3 channel 4		101	Unused	Unused
75	Unused	Unused		102	Unused	Unused
76	Unused	ATOM-3 channel 5		103	Unused	Unused
77	Unused	Unused		104	Unused	Unused
78	Unused	ATOM-3 channel 6		105	Unused	Unused
79	Unused	Unused		106	Unused	Unused
80	Unused	ATOM-3 channel 7		107	Unused	Unused
81	Unused	Unused		108	Unused	Unused
82	Unused	Unused		109	Unused	Unused
83	Unused	Unused		110	Unused	Unused
84	Unused	Unused		111	Unused	Unused
85	Unused	Unused		112	Unused	Unused
86	Unused	Unused		113	Unused	Unused
87	Unused	Unused		114	Unused	Unused
88	Unused	Unused		115	Unused	Unused
89	Unused	Unused		116	Unused	Unused
90	Unused	Unused		117	Unused	Unused
91	Unused	Unused		118	Unused	Unused
92	Unused	Unused		119	Unused	Unused
93	Unused	Unused		120	Unused	Unused
94	Unused	Unused		121	Unused	Unused
95	Unused	Unused		122	Unused	Unused
96	Unused	Unused		123	Unused	Unused
97	Unused	Unused		124	Unused	Unused
98	Unused	Unused		125	Unused	Unused
99	Unused	Unused		126	Unused	Unused
100	Unused	Unused		127	Unused	Unused

38.28.11 ATOM

Bit-reversed mode (PCM) in ATOM SOMP is available in the following channels:

Table 38.523 Available PCM in ATOM SOMP mode

Module	Ch-0	Ch-1	Ch-2	Ch-3	Ch-4	Ch-5	Ch-6	Ch-7
ATOM-0	no	yes	no	yes	no	yes	no	yes
ATOM-1	no	yes	no	yes	no	yes	no	yes
ATOM-2	no	yes	no	yes	no	yes	no	yes
ATOM-3	no	yes	no	yes	no	yes	no	yes

38.28.12 GTM Application constraints

If setting up an application on GTM-IP one has to take following constraints into account. Otherwise, the GTM-IP may not be able to fulfill the specified behavior.

Table 38.524 GTM-IP_358 application constraints

#	Module	Description	Required value	Effect, when not considered
1	MCS	Worst Case Execution Time of an N-cycle instruction.	<= 9*N system clock periods	MCS program execution may be too long for the application's requirements.
2	MCS	If MCS scheduling is configured in round robin mode, the MCS features deterministic program execution (the program of MCS channel x does not modify execution time of MCS channel y).		No channel independent execution time.
3	ATOM	If channel is triggered by preceding channel via TRIG_<x-1> signal, the selected CMU_CLK of both channels has to be the same	Identical CMU_CLK	The trigger of preceding channel may be lost
4	TIM	If a TIM channel x uses ARU transfer with enabled TDU, the minimal time T_{IN}between two subsequent measurement cycles has to be greater than the maximum time T_{SAMPLE} between two subsequent ARU read request events on TIM channel x.	$T_{IN} > T_{SAMPLE}$ (FIFO: T_{SAMPLE} = worst case ARU round trip time, MCS: T_{SAMPLE}= max. time between two (N)ARD(I) instructions for TIM channel x)	The ARU destination of TIM channel x cannot distinguish between a measurement cycle overflow and a timeout with subsequent valid measurement cycle.
5	ARU	The ARU round trip time in pure round robin mode is device specific.	(ARU_CADDR_END +1) times system clock periods with ARU_CADDR_END = 80 (default value);	If dynamic mode is activated with CLK_WAIT> 0 the ARU round trip time becomes longer and the value is undefined.

38.28.13 GTM DTM Register Mapping Table

Table 38.525 GTM DTM Register Mapping Table

GTM v2.0 ..V3.0	GTM v3.5		
DTM[i]	CDTM[n]	CDTM[n]_ DTM[j]	TOM / ATOM
24	0	4	ATOM0
25	0	5	ATOM0
26	1	4	ATOM1
27	1	5	ATOM1
28	2	4	ATOM2
29	2	5	ATOM2
30	3	4	ATOM3
31	3	5	ATOM3

38.28.14 GTM DTMA port mapping Table

Table 38.526 GTM DTMA port mapping Table

gtm_dtma_aux_in[i]	CDTM[n]	CDTM[n]_ DTM[j]
0	0	4
1	0	5
2	1	4
3	1	5
4	2	4
5	2	5
6	3	4
7	3	5

NOTE

In RH850/U2A-EVA Group, gtm_dtma_aux_in[i] pins are fixed to "0".

38.29 Interconnection of GTM-IP

38.29.1 Outputs to ADCJ

GTM outputs can trigger start of ADCJ conversion. The following GTM outputs can be used as ADCJ trigger sources.

- ATOM_OUT
- MCS_IRQ
- TIM_IRQ

Signals specified above are multiplexed together and routed to the ADCJ. Multiplexing scheme is described in **Section 41, Peripheral Interconnect (PIC)**. The GTM can trigger AD groups 0 to 4.

For more information about ADCJ functionality, refer to **Section 43, Analog to Digital Converter (ADCJ)**.

38.29.2 Outputs to ENCA

GTM outputs can trigger the ENCA timer counter capture. The following GTM outputs can be used as ENCA trigger sources.

- ATOM_OUT
- TIM_IRQ

Signals specified above are multiplexed together and routed to the ENCA. The GTM outputs can trigger capture trigger inputs 0 and 1. Multiplexing scheme is described in **Section 41, Peripheral Interconnect (PIC)**.

For more information about ENCA functionality, refer to **Section 40, Encoder Timer A (ENCA)**.

38.29.3 Outputs to PSI5-S

GTM outputs can routed to PSI5-S signals enable, clear, clock and sync pulse signal. The following GTM outputs can be routed to PSI5-S.

- ATOM_OUT

Signals specified above are multiplexed together and routed to the PSI5-S. Multiplexing scheme is described in **Section 41, Peripheral Interconnect (PIC)**. For more information about PSI5-S functionality, refer to **Section 28, Peripheral Sensor Interface 5 S (PSI5S)**.

38.29.4 Outputs to MSPIn

GTM output can be used as hardware triggers for MSPIn's operation. The following GTM outputs can be used.

- MCS_IRQ

Signals specified above are selected by functionality of MSPIn. For more information, refer to **Section 19.5.2.12, MSPInCFGm4 — MSPIn channel Configuration Register m4 (m = 0 to 7)**.

NOTE

In U2A-EVA, it does not support the MCS0 and MCS2 interrupt as MSPI HW trigger.

38.29.5 Peripheral Inputs

Outputs of other peripherals can be connected to GTM inputs. Those outputs are routed to TIM input channels. See **Section 41, Peripheral Interconnect (PIC)** for routing schemes.

38.29.5.1 Inputs from PORT

For GTM inputs of TIM channel, we can select from both alternative pin function of GTM and RLIN3nRX.

If port pin alternative RLIN3nRX function is selected a signal from a micro pin is routed to a corresponding TIM channel. See **Section 41, Peripheral Interconnect (PIC)** for possible routing options to TIM channels.

If port pin alternative GTM function is selected, a signal from a micro pin is routed to a corresponding TIM channel. See **Section 2, Pin Functions** for possible routing options to TIM channels. Whether a signal from a micro pin is used by TIM channel as an input also depends on TIM sub-module configuration inside GTM.

38.29.5.2 Inputs from ADCJ

The following interrupt factors of ADCJ can be routed to TIM input channels.

– End of conversion of ADCJ scan group

38.29.5.3 Input from ENCA

All ENCA interrupt factors can be routed to GTM TIM input channels.

– Encoder input (phase A, B, Z)
– Compare match or capture
– Clear interrupt by ENCA input
– Count clock
– Down-count enable signal

38.29.6 ADC Interface

Results of A/D conversion are stored in ADC_CH[y]_DATA registers (y= 0...31). Either registers 0...15 or registers 16...31 can be used based on GTM_ADCI_CHSELj.CHS. In addition, a register designation is also based on GTMTAG (ADCJ). For more information about GTMTAG configuration, refer to **Section 43, Analog to Digital Converter (ADCJ)**.

It is not possible to send one A/D data to multiple ADC_CH[y]_DATA registers based on GTMTAG. Sending multiple A/D results to a same ADC_CH[y]_DATA register based on GTMTAG is possible but has to be carefully designed to avoid data overwrites.

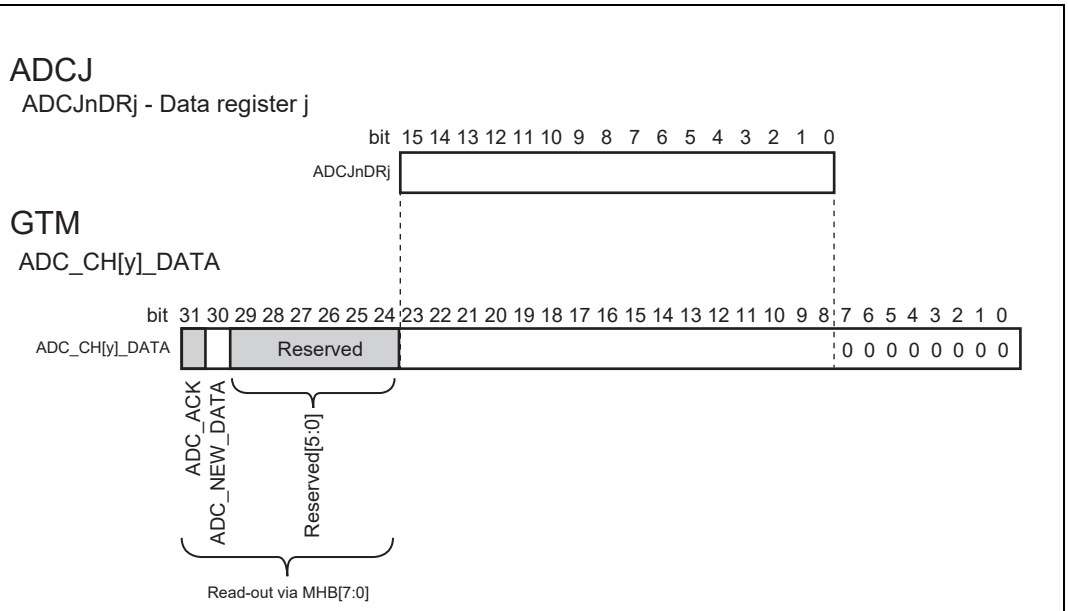
The data format in ADC_CH[y]_DATA register matches configuration of corresponding A/D channel.

ADC_CH[y]_DATA registers selects the data to be stored from the ADC conversion result.

Table 38.527 Allocation of an ADC_CH[y]_DATA Register and GTMTAG Setting

	ADCJ	
GTM Registers	CHS	GTMTAG[3:0]
ADC_CH0_DATA	0_H	0_H
ADC_CH1_DATA	0_H	1_H
ADC_CH2_DATA	0_H	2_H
ADC_CH3_DATA	0_H	3_H
ADC_CH4_DATA	0_H	4_H
ADC_CH5_DATA	0_H	5_H
ADC_CH6_DATA	0_H	6_H
ADC_CH7_DATA	0_H	7_H
ADC_CH8_DATA	0_H	8_H
ADC_CH9_DATA	0_H	9_H
ADC_CH10_DATA	0_H	A_H
ADC_CH11_DATA	0_H	B_H
ADC_CH12_DATA	0_H	C_H
ADC_CH13_DATA	0_H	D_H
ADC_CH14_DATA	0_H	E_H
ADC_CH15_DATA	0_H	F_H
ADC_CH16_DATA	1_H	0_H
ADC_CH17_DATA	1_H	1_H
ADC_CH18_DATA	1_H	2_H
ADC_CH19_DATA	1_H	3_H
ADC_CH20_DATA	1_H	4_H
ADC_CH21_DATA	1_H	5_H
ADC_CH22_DATA	1_H	6_H
ADC_CH23_DATA	1_H	7_H
ADC_CH24_DATA	1_H	8_H
ADC_CH25_DATA	1_H	9_H
ADC_CH26_DATA	1_H	A_H
ADC_CH27_DATA	1_H	B_H
ADC_CH28_DATA	1_H	C_H
ADC_CH29_DATA	1_H	D_H
ADC_CH30_DATA	1_H	E_H
ADC_CH31_DATA	1_H	F_H

The figure below shows the bit correspondence between the AD conversion result and the ADCI_CH[y]_DATA register of GTM.



Note: For the transfer formats from ADCJnDRj, refer to **Section 43, Analog to Digital Converter (ADCJ)**.

Figure 38.126 Comparison of AD conversion result with ADC_CH[y]_DATA register of GTM

38.29.7 Safety Mechanism

MCS RAM is protected by ECC function. ECC error of MCS RAM is also reported by MCS channel. For more details about ECC function, refer to **Section 44, Functional Safety**.

38.30 GTM Debug Function

GTM debug feature is as follows.

- Run, break
- Single step execution of MCS
- On-chip breakpoint by:
  - MCS instruction or data access
  - ARU data access
  - TBU counter
- Trace the following with timestamp: (U2A-EVA only)
  - MCS instruction or data access
  - ARU data access
  - TBU counter
  - TIM/ATOM state change

Section 39 Real-Time Clock (RTCA)

This section contains a generic description of Real-Time Clock (RTCA).

First part of this section describes all RH850/U2A-EVA specific properties, such as number of units, register base addresses, etc. Remainder of section describes functions and registers of RTCA.

39.1 Features RTCA for RH850/U2A-EVA

39.1.1 Number of Units and Channels

This microcontroller has following number of RTCA units.

Each RTCA unit has one channel RTCA. “Number of channels” is used with same meaning as “number of units” in this section.

Table 39.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	RTCAn									

Table 39.2 Index

Index	Description
n	Throughout this section, individual RTCA units are identified by index “n” (n = 0); for example, RTCAnCTL0 is RTCAn control register 0.

39.1.2 Register Base Addresses

RTCAn base address is listed in following table.

RTCAn register addresses are given as offsets from base address.

Table 39.3 Register Base Address

Base Address Name	Base Address	Bus Group
<RTCA0_base>	FF99 7000_H	P-Bus Group 2L

39.1.3 Clock Supply

RTCAn clock supply is shown in following table.

Table 39.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
RTCA0	RTCATCKI	CLKA_RTCA	Macro clock
	PCLK	CLK_LSB	Module clock

39.1.4 Interrupt Requests and Error Notifications

RTCAn interrupt requests are listed in following table.

Table 39.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Wake up
RTCA0				
INTRTCA01S	1-second interval interrupt	654	—	√
INTRTCA0AL	Alarm interrupt	655	—	√
INTRTCA0R	Fixed interval interrupt	656	—	√

This module has no error notifications.

39.1.5 Reset Sources

RTCAn reset sources are listed in following table. RTCAn is initialized by these reset sources.

Table 39.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
RTCAn	All registers	√	√	√	—	—	—	—

39.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

Table 39.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
RTCA0		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT*1

Note 1. RTCA0OUT is connected to TAUJ3. For details, see **Section 34, Timer Array Unit J (TAUJ)**.

39.2 Overview

39.2.1 Functional Overview

Real-Time Clock (RTCA) has following features:

- Count clock selection from 240 kHz to 2.5 MHz
- Counters for years, months, day of month, day of week, hours, minutes, seconds, and a sub-counter. Calendar covers 99 years. Leap years are handled by hardware automatically.
- One Hz pulse output function
- Fixed interval interrupt function
- Alarm interrupt function

39.2.2 Block Diagram

Block diagram shows main components of RTCA.

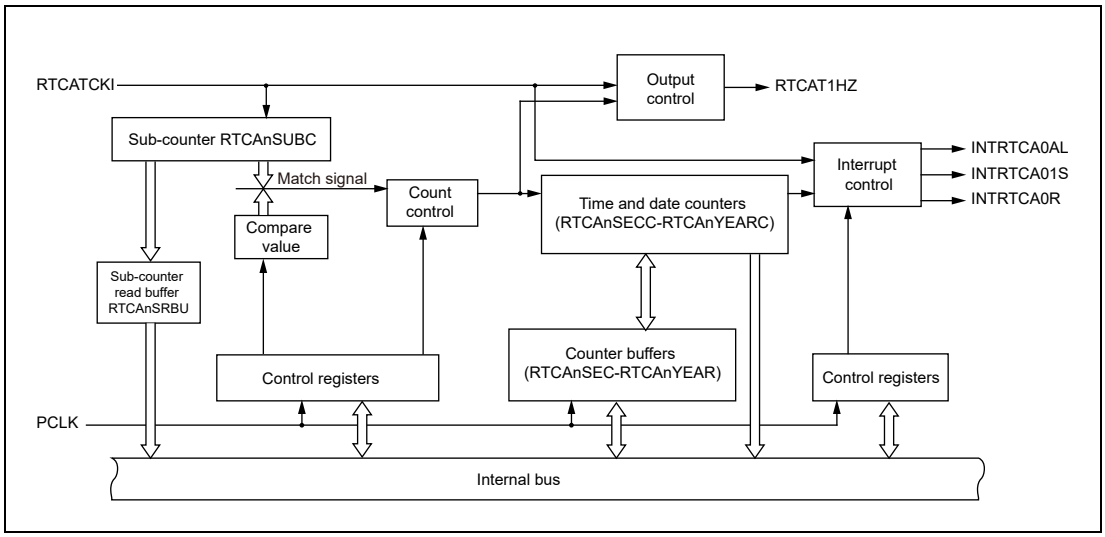


Figure 39.1 Block Diagram of RTCA

39.2.3 Description of Blocks

Real-Time Clock RTCA provides information about present time and date and can generate wake-up signals (interrupts, alarms). This information is derived from count clock RTCATCKI.

Sub-counter

RTCATCKI is input to sub-counter RTCAnSUBC. Sub-counter counts up from 0 until it reaches compare value. Compare value is always defined as frequency of RTCATCKI – 1 (in Hz). Thus, sub-counter overflows after one second. It is then reset to 0 and triggers seconds counter RTCAnSECC (and, if specified, interrupt INTRTCA01S).

Sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5 seconds, or 1 second, and a 1-Hz output pulse.

Time and date counters

Counters for minutes, hours, day of week, day of month, months, and years also count up. They have their own overflow limits. If all lower counters overflow, upper counter counts up.

Overflow limit of counter for day of month (RTCAnDAYC) depends on present month (28, 30, or 31 days) and (in February) on year counter RTCAnYEARC (years 0, 4, 8, 12, etc. are considered leap years).

Hours counter RTCAnHOURC can be switched between 12- and 24-hour formats.

Counters for seconds, minutes, hours, day of month, and months can generate a fixed interval interrupt upon overflow (INTRTCA0R).

Counters for minutes, hours, and day of week can also generate an alarm interrupt (INTRTCA0AL), e.g. every Tuesday and Thursday at 10:32.

Counter buffers

All counters can be read directly at any time. Clock signal used to access read/write registers and count clock are usually asynchronous. An overflow of sub-counter during read operation can make all read values obsolete. Therefore, reading counters must be performed using a special procedure. For details, see **Section 39.5.1.3, Reading Clock Counters**.

For reasons of synchronization, counters cannot be written directly.

For reading and writing, all counters are accompanied by buffer registers. Buffer registers provide a synchronized way for reading counters and for setting time and date. When they are used, operation of sub-counter must first be suspended and then re-activated (see also **Section 39.5.1.3, Reading Clock Counters** and **Section 39.5.1.2, Updating Clock Counters**.

RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to check and set time (hours, minutes and seconds) or date (day of week, day of month, month, and year) with one read/write operation.

39.3 Registers

39.3.1 List of Registers

RTCA registers are listed in following table.

<RTCAn_base> is defined in **Section 39.1.2, Register Base Addresses**.

Table 39.8 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Others
Control registers						
RTCAn	Control register 0	RTCAnCTL0	<RTCAn_base> + 00_H	8	PBG20#10	—
RTCAn	Control register 1	RTCAnCTL1	<RTCAn_base> + 04_H	8	PBG20#10	—
RTCAn	Control register 2	RTCAnCTL2	<RTCAn_base> + 08_H	8	PBG20#10	—
Sub-counter registers						
RTCAn	Sub-count register	RTCAnSUBC	<RTCAn_base> + $0C_H$	32	PBG20#10	—
RTCAn	Sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10_H	32	PBG20#10	—
RTCAn	Sub-counter compare register	RTCAnSCMP	<RTCAn_base> + $3C_H$	32	PBG20#10	—
Clock counter and buffer registers						
RTCAn	Seconds count register	RTCAnSECC	<RTCAn_base> + $4C_H$	8	PBG20#10	—
RTCAn	Seconds count buffer register	RTCAnSEC	<RTCAn_base> + 14_H	8	PBG20#10	—
RTCAn	Minute count register	RTCAnMINC	<RTCAn_base> + 50_H	8	PBG20#10	—
RTCAn	Minute count buffer register	RTCAnMIN	<RTCAn_base> + 18_H	8	PBG20#10	—
RTCAn	Hour count register	RTCAnHOURC	<RTCAn_base> + 54_H	8	PBG20#10	—
RTCAn	Hour count buffer register	RTCAnHOUR	<RTCAn_base> + $1C_H$	8	PBG20#10	—
RTCAn	Day of week count register	RTCAnWEEKC	<RTCAn_base> + 58_H	8	PBG20#10	—
RTCAn	Day of week count buffer register	RTCAnWEEK	<RTCAn_base> + 20_H	8	PBG20#10	—
RTCAn	Day count register	RTCAnDAYC	<RTCAn_base> + $5C_H$	8	PBG20#10	—
RTCAn	Day count buffer register	RTCAnDAY	<RTCAn_base> + 24_H	8	PBG20#10	—
RTCAn	Month count register	RTCAnMONC	<RTCAn_base> + 60_H	8	PBG20#10	—
RTCAn	Month count buffer register	RTCAnMONTH	<RTCAn_base> + 28_H	8	PBG20#10	—
RTCAn	Year count register	RTCAnYEARC	<RTCAn_base> + 64_H	8	PBG20#10	—
RTCAn	Year count buffer register	RTCAnYEAR	<RTCAn_base> + $2C_H$	8	PBG20#10	—
Special counter and buffer registers						
RTCAn	Time count register	RTCAnTIMEC	<RTCAn_base> + 68_H	32	PBG20#10	—
RTCAn	Time count buffer register	RTCAnTIME	<RTCAn_base> + 30_H	32	PBG20#10	—
RTCAn	Calendar count register	RTCAnCALC	<RTCAn_base> + $6C_H$	32	PBG20#10	—
RTCAn	Calendar count buffer register	RTCAnCAL	<RTCAn_base> + 34_H	32	PBG20#10	—
Alarm time setting registers						
RTCAn	Alarm minute setting register	RTCAnALM	<RTCAn_base> + 40_H	8	PBG20#10	—
RTCAn	Alarm hour setting register	RTCAnALH	<RTCAn_base> + 44_H	8	PBG20#10	—
RTCAn	Alarm day of week setting register	RTCAnALW	<RTCAn_base> + 48_H	8	PBG20#10	—

39.3.2 Details of RTCA Control Registers

39.3.2.1 RTCAnCTL0 — RTCA Control Register 0

This register controls count operation of sub-counter RTCAnSUBC, format (12-hour/24-hour) of hours counter RTCAnHOURC and alarm hour setting register RTCAnALH, and operation mode.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnCE	RTCAnCEST	RTCAnAMPM	RTCAnSLSB	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R	R

Table 39.9 RTCAnCTL0 Register Contents

Bit Position	Bit Name	Function
7	RTCAnCE	Starts/stops sub-counter RTCAnSUBC operation. 0: Stops sub-counter operation. All output pins and all status flags in control register RTCAnCTL2 are cleared. 1: Starts sub-counter operation. sub-counter counts up.
6	RTCAnCEST	Indicates operation enabled/stopped status of sub-counter: 0: Operation stopped status 1: Operation enabled status For details on how to use this status flag, see **Section 39.5.1.1, Initial Setting of RTCA**.
5	RTCAnAMPM	Selects format of hours counter RTCAnHOURC and alarm hour setting register RTCAnALH: 0: 12-hour format (1 to 12, am/pm) 1: 24-hour format (0 to 23, military time) For details on format, see **Table 39.20, 12- and 24-Hour Format**.
4	RTCAnSLSB	Selects operation mode: 0: Setting prohibited 1: Frequency selection mode*1 For details on operation modes, see **Section 39.4, Operation**. Operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1). For details on initialization of RTCAn, see **Section 39.5.1.1, Initial Setting of RTCA**.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write value after reset.

Note 1. RTCAnSLSB must be set “1” before RTCAnSCMP register setting.

39.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls interrupt request generation and 1-Hz pulse output.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnEN1HZ	RTCAnENALM	RTCAnEN1S	RTCAnCT2	RTCAnCT1	RTCAnCT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.10 RTCAnCTL1 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write value after reset.
5	RTCAnEN1HZ	Enables/stops 1-Hz pulse output (RTCAT1HZ): 0: RTCAT1HZ disabled (RTCAT1HZ is fixed to 0) 1: RTCAT1HZ enabled
4	RTCAnENALM	Enables/disables alarm interrupt request generation (INTRTCA0AL): 0: INTRTCA0AL disabled 1: INTRTCA0AL enabled
3	RTCAnEN1S	Enables/disables 1-second interrupt request generation (INTRTCA01S): 0: INTRTCA01S disabled 1: INTRTCA01S enabled
2 to 0	RTCAnCT[2:0]	Specifies fixed interval interrupt request (INTRTCA0R) setting:

RTCAnCT[2:0]	RTCATINTR Interrupt Request Generation	
	Interval	Timing
000	No interrupt request generation	
001	Every 0.25 seconds	Every 0.25, 0.5, 0.75 and 1 second
010	Every 0.5 seconds	Every 0.5 and 1 second
011	Every second	Every 1 second
100	Every minute	Every 1 minute 00 seconds
101	Every hour	Every 1 hour 00 minutes 00 seconds
110	Every day	Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)
111	Every month	Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)

If settings of RTCAnCT[2:0] are changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to INTRTCA0R. Implement appropriate interrupt mask processing procedures.

39.3.2.3 RTCAnCTL2 — RTCA Control Register 2

This register contains status information and controls data transfer from sub-counter RTCAnSUBC to dedicated sub-counter read buffer RTCAnSRBU and operation setting of clock counters (RTCAnSECC to RTCAnYEARC).

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnWSST	RTCAnRSST	RTCAnRSUB	RTCAnWST	RTCAnWAIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W

Table 39.11 RTCAnCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write value after reset.
4	RTCAnWSST	Indicates whether RTCAnSCMP write operation has been completed: 0: RTCAnSCMP write completed 1: RTCAnSCMP write in progress The write operation ends with next sub-counter overflow. This bit is set (to 1) when RTCAnSCMP is written to while clock counter operation is enabled (RTCAnCE = 1). It is then cleared (to 0) upon completion of RTCAnSCMP write (with next RTCAnSUBC overflow). See **Section 39.5.1.5, Writing to RTCAnSCMP**, for details.
3	RTCAnRSST	Indicates whether value of sub-counter (RTCAnSUBC) has been transferred to sub-count register read buffer (RTCAnSRBU): 0: Transfer in progress, or waiting for a transfer trigger 1: Transfer completed This bit is cleared (transfer is triggered) by RTCAnRSUB=1.This bit is automatically set when transfer is completed. See **Section 39.5.1.4, Reading RTCAnSRBU**, for details.
2	RTCAnRSUB	Triggers transfer of value of sub-counter (RTCAnSUBC) to dedicated read buffer (RTCAnSRBU) or clears transfer state of sub-counter: 0: Transfer status (RTCAnRSST) is cleared. 1: Transfer is triggered. This bit is used to read value of RTCAnSRBU when sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). value of RTCAnSUBC is synchronized with RTCATCKI and loaded to RTCAnSRBU. For details, see **Section 39.5.1.4, Reading RTCAnSRBU**.
1	RTCAnWST	Indicates status of all clock counters (RTCAnSECC to RTCAnYEARC): 0: All clock counters are running. 1: All clock counters are stopped sub-counter is still running. Clock counters must be stopped before reading or writing clock counter values during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). To stop clock counters, set RTCAnWAIT = 1.

Table 39.11 RTCAnCTL2 Register Contents (2/2)

Bit Position	Bit Name	Function
0	RTCAnWAIT	Restarts/stops all clock counters (RTCAnSECC to RTCAnYEARC): 0: Restarts all clock counters either immediately or immediately after clock counter write operation finishes. 1: Stops all clock counters temporarily. Sub-counter is still running. Clock counters must be stopped before reading or writing counter buffers during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). **CAUTION** Only one overflow can be held internally. When two overflows occur, seconds counter is incremented only by one when it is restarted. Thus, procedure must be completed within one second.

39.3.3 Details of RTCA Sub-Counter Registers

39.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts 1-second reference time. It operates using count clock RTCATCKI.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

This register is initialized:

– When write operation is performed to seconds count buffer register (RTCAnSEC) or to time count buffer register (RTCAnTIME) and value is reflected to seconds count register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSUBC[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSUBC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.12 RTCAnSUBC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSUBC [21:0]	Sub-counter value Sub-counter only operates while RTCAnCTL0.RTCAnCEST = 1.

NOTES

1. This sub-counter operates with RTCATCKI while read operation is clocked by PCLK. Reading this sub-counter during operation (RTCAnCTL0.RTCAnCEST = 1) is asynchronous to RTCATCKI and can lead to wrong results.
   Use sub-count register read buffer (RTCAnSRBU) to read sub-counter value during operation. For details, see **Section 39.5.1.4, Reading RTCAnSRBU**.
2. Count-operation of this sub-counter depends on selected operation mode. See **Section 39.4, Operation**, for details.

39.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is read buffer for sub-counter RTCAnSUBC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSRBU[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSRBU[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.13 RTCAnSRBU Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSRBU [21:0]	Sub-counter value at time of last RTCAnSUBC read. When RTCAnCTL2.RTCAnRSUB is set to 1, value of RTCAnSUBC is loaded to read buffer in synchronization with RTCATCKI.

NOTE

Perform RTCAnSRBU read according to flow described in **Section 39.5.1.4, Reading RTCAnSRBU**.

39.3.3.3 RTCAnSCMP — RTCA Sub-Counter Compare Register

This register sets compare value of sub-counter RTCAnSUBC in frequency selection mode (RTCAnCTL0.RTCAnSLSB = 1).

When sub-counter values matches value of this register, an overflow signal is output to seconds counter RTCAnSECC and sub-counter is cleared.

Set value for this register according to frequency of input clock RTCATCKI.

Access: This register can be read or written in 32-bit units.
Note following when writing this register during sub-counter operation:
- Previous RTCAnSCMP write must be completed (RTCAnCTL2.RTCAnWSST = 0).
- write operation ends with next sub-counter overflow.

Address: <RTCAn_base> + $3C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSCMP[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.14 RTCAnSCMP Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write value after reset.
21 to 0	RTCAnSCMP [21:0]	Sub-counter compare value in frequency selection mode.

Example

Following example illustrates setting of RTCAnSCMP:

- RTCATCKI = 2.5 MHz = 2,500,000 Hz
- RTCAnSCMP = 2,500,000 – 1 = 2,499,999 (decimal code) = $26259F_H$
- Seconds counter RTCAnSECC is triggered when sub-counter value changes from $26259F_H$ to 0_H.

NOTES

1. Operation of RTCA cannot be guaranteed if a value of 3198 (decimal code) or lower is set in this register.
2. Perform RTCAnSCMP write as described in **Section 39.5.1.1, Initial Setting of RTCA** and **Section 39.5.1.5, Writing to RTCAnSCMP**.

39.3.4 Details of RTCA Clock Counter and Buffer Registers

39.3.4.1 RTCAnSECC — RTCA Seconds Count Register

This register is seconds counter. It counts seconds from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of sub-counter RTCAnSUBC.

  If sub-counter overflows while seconds counter is stopped (RTCAnCTL2.RTCAnWST = 1), seconds counter behaves as follows:

  – If one sub-counter overflow occurs while seconds counter is stopped, overflow is held internally.
  seconds counter is incremented by one when it is restarted.

  – If two or more overflows occur while seconds counter is stopped, overflow count cannot be held internally.
  seconds counter is incremented by one when it is restarted.

  – If seconds counter was updated while seconds counter is stopped, sub-counter overflow(s) are ignored.

- It outputs an overflow signal when value changes from 59 to 00. Overflow signal triggers minutes counter (RTCAnMINC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $4C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 39.15 RTCAnSECC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnSECC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to seconds count buffer register RTCAnSEC or to clock time setting register RTCAnTIME. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters**

39.3.4.2 RTCAnSEC — RTCA Seconds Count Buffer Register

This register is a buffer register to read/write seconds counter RTCAnSECC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.16 RTCAnSEC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnSEC read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.3 RTCAnMINC — RTCA Minutes Count Register

This register is minutes counter. It counts minutes from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of seconds counter RTCAnSECC.
- It outputs an overflow signal when value changes from 59 to 00. Overflow signal triggers hours counter (RTCAnHOURC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 39.17 RTCAnMINC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnMINC [6:0]	Minutes in BCD

NOTES

1. Perform RTCAnMINC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to minutes count buffer register RTCAnMIN or to time count buffer register RTCAnTIME. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.4.4 RTCAnMIN — RTCA Minutes Count Buffer Register

This register is a buffer register to read/write minutes counter RTCAnMINC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.18 RTCAnMIN Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write value after reset.
6 to 0	RTCAnMIN [6:0]	Minutes in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnMIN read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.5 RTCAnHOURC — RTCA Hours Count Register

This register is hours counter. It counts hours in BCD. Count range depends on selected hour format. See **Table 39.20, 12- and 24-Hour Format**.

This register counts as follows.

- It is triggered by every overflow of minutes counter RTCAnMINC.
- It outputs an overflow signal when value changes from 23 to 00 (in 24-hour format) or from 31 to 12 (in 12-hour format). Overflow signal triggers two counters:
  - Day of week counter (RTCAnWEEKC)
  - Day of month counter (RTCAnDAYC)

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 54_H

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R

Table 39.19 RTCAnHOURC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnHOURC [5:0]	Hours in BCD. See **Table 39.20, 12- and 24-Hour Format**, for details.

NOTES

1. Perform RTCAnHOURC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to hours count buffer register RTCAnHOUR or to time count buffer register RTCAnTIME. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

12- or 24-hour format

Count values of RTCAnHOURC depend on selected hour format.

If 12-hour format is selected (RTCAnCTL0.RTCAnAMPM = 0), bit 5 in RTCAnHOURC register is am/pm indicator:

- RTCAnHOURC[5] = 0: am
- RTCAnHOURC[5] = 1: pm

Following table shows count range of RTCAnHOURC in both 12- and 24-hour format.

Table 39.20 12- and 24-Hour Format

12-Hour Format (RTCAnAMPM = 0)			24-Hour Format (RTCAnAMPM = 1)	
Time	RTCAnHOURC		Time	RTCAnHOURC
0 am	12_H		0	00_H
1 am	01_H		1	01_H
2 am	02_H		2	02_H
3 am	03_H		3	03_H
4 am	04_H		4	04_H
5 am	05_H		5	05_H
6 am	06_H		6	06_H
7 am	07_H		7	07_H
8 am	08_H		8	08_H
9 am	09_H		9	09_H
10 am	10_H		10	10_H
11 am	11_H		11	11_H
0 pm	32_H	↓	12	12_H
1 pm	21_H	pm indicator in 12-hour format: RTCAnHOUR.RTCAnHOUR[5] = 1	13	13_H
2 pm	22_H		14	14_H
3 pm	23_H		15	15_H
4 pm	24_H		16	16_H
5 pm	25_H		17	17_H
6 pm	26_H		18	18_H
7 pm	27_H		19	19_H
8 pm	28_H		20	20_H
9 pm	29_H		21	21_H
10 pm	30_H		22	22_H
11 pm	31_H		23	23_H

39.3.4.6 RTCAnHOUR — RTCA Hours Count Buffer Register

This register is a buffer register to read/write hours counter RTCAnHOURC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $1C_H$

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.21 RTCAnHOUR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write value after reset.
5 to 0	RTCAnHOUR [5:0]	Hours in BCD See **Table 39.20, 12- and 24-Hour Format**, for details.

NOTES

1. When writing this register, only following decimal values in BCD are allowed:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0):
     01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1):
     00 to 23
2. Perform RTCAnHOUR read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.7 RTCAnWEEKC — RTCA Day of Week Count Register

This register is day of week counter. It counts from 0 to 6.

This register counts as follows.

- It is triggered by every overflow of hours counter RTCAnHOURC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 39.22 RTCAnWEEKC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of week

NOTES

1. Perform RTCAnWEEKC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to day of week count buffer register RTCAnWEEK or to calendar count buffer register RTCAnCAL. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.4.8 RTCAnWEEK — RTCA Day of Week Count Buffer Register

This register is a buffer register to read/write day of week counter RTCAnWEEKC.

There is no particular correspondence between value of RTCAnWEEK and day of week. Set correspondence according to application to be used.

Example: 0 = Sunday, 1 = Monday, …, 6 = Saturday

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 39.23 RTCAnWEEK Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of week

NOTES

1. When writing this register, only decimal values between 0 and 6 in BCD are allowed.
2. Perform RTCAnWEEK read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.9 RTCAnDAYC — RTCA Day of Month Count Register

This register is day of month counter. It counts from 01 to a maximum of 31 in BCD, depending on value of month counter (RTCAnMONC) and year counter (RTCAnYEARC):

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, non-leap year)

Years 0, 4, 8, 12, etc., are considered leap years.

This register counts as follows.

- It is triggered by every overflow of hours counter RTCAnHOURC.
- It outputs an overflow signal when value changes from 28, 29, 30, or 31 to 01, depending on current month and year. Overflow signal triggers month counter (RTCAnMONC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $5C_H$

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 39.24 RTCAnDAYC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnDAYC [5:0]	Day of month in BCD

NOTES

1. Perform RTCAnDAYC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to day of month count buffer register RTCAnDAY or to calendar count buffer register RTCAnCAL. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.4.10 RTCAnDAY — RTCA Day of Month Count Buffer Register

This register is a buffer register to read/write day of month counter RTCAnDAYC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 24_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.25 RTCAnDAY Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write value after reset.
5 to 0	RTCAnDAY [5:0]	Day of month in BCD

NOTES

1. When writing this register, only decimal values between 01 and 31 in BCD are allowed:
   - 01 to 31 (January, March, May, July, August, October, December)
   - 01 to 30 (April, June, September, November)
   - 01 to 29 (February, leap year)
   - 01 to 28 (February, non-leap year)
2. Perform RTCAnDAY read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.11 RTCAnMONC — RTCA Month Count Register

This register is month counter. It counts month of year, starting from 01 to 12 in BCD.

This register counts as follows.

- It is triggered by every overflow of counter for day of month RTCAnDAYC.
- It outputs an overflow signal when value changes from 12 to 01. Overflow signal triggers year counter (RTCAnYEARC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 60_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 39.26 RTCAnMONC Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	RTCAnMONC [4:0]	Month of year in BCD

NOTES

1. Perform RTCAnMONC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to month count buffer register RTCAnMONTH or to calendar count buffer register RTCAnCAL. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.4.12 RTCAnMONTH — RTCA Month Count Buffer Register

This register is a buffer register to read/write month counter RTCAnMONC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 28_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 39.27 RTCAnMONTH Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write value after reset.
4 to 0	RTCAnMONTH [4:0]	Month of year in BCD

NOTES

1. When writing this register, only decimal values between 01 and 12 in BCD are allowed.
2. Perform RTCAnMONTH read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.4.13 RTCAnYEARC — RTCA Year Count Register

This register is year counter. It counts years from 00 to a maximum of 99 in BCD.

Years 00, 04, 08, …, 92, and 96 (every four years) are considered leap years.

This register counts as follows.

- It is triggered by every overflow of month counter RTCAnMONC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEARC[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 39.28 RTCAnYEARC Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEARC [7:0]	Year in BCD

NOTES

1. Perform RTCAnYEARC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to year count buffer register RTCAnYEAR or to calendar count buffer register RTCAnCAL. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.4.14 RTCAnYEAR — RTCA Year Count Buffer Register

This register is a buffer register to read/write year counter RTCAnYEARC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEAR[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.29 RTCAnYEAR Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEAR [7:0]	Year in BCD

NOTES

1. When writing this register, only decimal values between 00 and 99 in BCD are allowed.
2. Perform RTCAnYEAR read/write as described in
   - **Section 39.5.1.1, Initial Setting of RTCA,**
   - **Section 39.5.1.2, Updating Clock Counters**, and
   - **Section 39.5.1.3, Reading Clock Counters.**

39.3.5 Details of RTCA Special Counter and Buffer Registers

39.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 68_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]							—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.30 RTCAnTIMEC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	RTCAnHOURC [5:0]	Hours in BCD. See **Table 39.20, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned.
14 to 8	RTCAnMINC [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnTIMEC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to time count buffer register RTCAnTIME. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.5.2 RTCAnTIME — RTCA Time Count Buffer Register

This register enables RTCAnHOUR, RTCAnMIN, and RTCAnSEC buffer registers to be read/written simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <RTCAn_base> + 30_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]							—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.31 RTCAnTIME Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write value after reset.
21 to 16	RTCAnHOUR [5:0]	Hours in BCD See **Table 39.20, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned. When writing, write value after reset.
14 to 8	RTCAnMIN [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned. When writing, write value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTE

Perform RTCAnTIME read/write as described in

- **Section 39.5.1.1, Initial Setting of RTCA,**
- **Section 39.5.1.2, Updating Clock Counters**, and
- **Section 39.5.1.3, Reading Clock Counters.**

39.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + $6C_H$

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEARC[7:0]								—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]						—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 39.32 RTCAnCALC Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEARC [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned.
20 to 16	RTCAnMONC [4:0]	Month of year in BCD
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RTCAnDAYC [5:0]	Day of month in BCD
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of week in BCD

NOTES

1. Perform RTCAnCALC read according to flow described in **Section 39.5.1.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to clock time setting register RTCAnCAL. See
   - **Section 39.5.1.1, Initial Setting of RTCA**, and
   - **Section 39.5.1.2, Updating Clock Counters.**

39.3.5.4 RTCAnCAL — RTCA Calendar Count Buffer Register

This register enables RTCAnYEAR, RTCAnMONTH, RTCAnDAY, and RTCAnWEEK buffer registers to be read/written simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <RTCAn_base> + 34_H

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEAR[7:0]								—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]						—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 39.33 RTCAnCAL Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEAR [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned. When writing, write value after reset.
20 to 16	RTCAnMONTH [4:0]	Month of year in BCD
15, 14	Reserved	When read, the value after reset is returned. When writing, write value after reset.
13 to 8	RTCAnDAY [5:0]	Day of month in BCD
7 to 3	Reserved	When read, the value after reset is returned. When writing, write value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of week in BCD

NOTE

Perform RTCAnCAL read/write as described in

- **Section 39.5.1.1, Initial Setting of RTCA.**
- **Section 39.5.1.2, Updating Clock Counters**, and
- **Section 39.5.1.3, Reading Clock Counters.**

39.3.6 Details of RTCA Alarm Setting Registers

39.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies minute of alarm interrupt.

For details and example settings, see **Section 39.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 40_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALM[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.34 RTCAnALM Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write value after reset.
6 to 0	RTCAnALM [6:0]	Minute of alarm interrupt in BCD

NOTES

1. If decimal values outside range of 00 to 59 in BCD are set, no alarm interrupt request will be generated.
2. When setting of RTCAnALM is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

39.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies hour of alarm interrupt.

For details and example settings, see **Section 39.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 44_H

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnALH[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.35 RTCAnALH Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write value after reset.
5 to 0	RTCAnALH [5:0]	Hour of alarm interrupt in BCD

NOTES

1. If decimal values outside following range are set, no alarm interrupt request will be generated:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0): 01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1): 00 to 23
2. When setting of RTCAnALH is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

39.3.6.3 RTCAnALW — RTCA Alarm Day of Week Setting Register

This register specifies day(s) of week of alarm interrupt.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 48_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALW[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 39.36 RTCAnALW Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write value after reset.
6 to 0	RTCAnALW [6:0]	Specifies day of week m (m = 0 to 6) as a day, when an alarm interrupt request is generated: 0: No alarm interrupt request is generated on day m. 1: Alarm interrupt request is generated on day m at time set using RTCAnALM and RTCAnALH. Bits of this register correspond to count value of day of week counter (RTCAnWEEKC).

NOTE

When setting of RTCAnALW is changed during sub-counter operation (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to INTRTCA0AL. Implement appropriate interrupt mask processing procedures.

Example

If Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

- To set alarm for Sunday, set RTCAnALW = $0000\ 0001_B$.
- To set alarm for Monday and Wednesday, set RTCAnALW = $0000\ 1010_B$.
- To set alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = $0101\ 0100_B$.

For more examples, see **Section 39.4.3, Alarm Interrupt Function**.

39.4 Operation

RTCA provides only one operation mode:

- Frequency selection mode

Operation mode that can be used depends on available input clock RTCATCKI. Operation mode specifies sub-counter compare value that is used to trigger seconds counter and thus all subsequent counters.

Following table provides an overview of property of the operation mode.

Table 39.37 RTCA Operation Mode Overview

	Frequency Selection Mode
Allowed input clock RTCATCKI	Any frequency from 240 kHz to 2.5 MHz
Sub-counter RTCAnSUBC operation	• Counter overflow at value of RTCAnSCMP • RTCAnSCMP must be set to RTCATCKI-1 (in Hz)

Operation mode is selected by control bit RTCAnCTL0.RTCAnSLSB. For details on how to set operation mode during RTCA initialization, see **Section 39.5.1.1, Initial Setting of RTCA**.

CAUTIONS

1. **Input clock RTCATCKI must not be outside allowed frequency range.**
2. **Operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1).**

39.4.1 Clock Counter Format

Clock counters (RTCAnSECC to RTCAnYEARC) operate on binary coded decimals (BCD): Each digit is represented by its own binary sequence.

Depending on valid data range, number of bits for a digit differs. For example, tens digit of month of year counter has only one bit (for 0 and 1) whereas tens digit of minutes counter has 3 bits (for 0 to 5).

Following table lists decimals 0 to 59 in binary and BCD.

Table 39.38 Example of BCD Code – Seconds or Minutes Counter (0 to 59)

Decimal	Binary	BCD
0	000000	000 0000
1	000001	000 0001
2	000010	000 0010
3	000011	000 0011
4	000100	000 0100
5	000101	000 0101
6	000110	000 0110
7	000111	000 0111
8	001000	000 1000
9	001001	000 1001
10	001010	001 0000
11	001011	001 0001
12	001100	001 0010
:	:	:
58	111010	101 1000
59	111011	101 1001

39.4.2 Fixed Interval Interrupt Function

Interrupt INTRTCA0R can be specified to occur after every 0.25 seconds, 0.5 seconds, 1 (full) second, 1 (full) minute, 1 (full) hour, 1 (full) day, or 1 (full) month.

Fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

39.4.3 Alarm Interrupt Function

Interrupt INTRTCA0AL can be specified to occur at a certain time on one or several days of week. This interrupt can be used as a wake-up signal.

Alarm interrupt function is enabled and disabled by bit RTCAnCTL1.RTCAnENALM.

Alarm setting is specified by following control registers:

- RTCAnALW selects weekday(s).

Allocation of bits to weekdays is defined by day of week count buffer register RTCAnWEEK.

- RTCAnALH and RTCAnALM specify hour and minute in BCD.

Examples

Following tables show some exemplary settings of alarm control registers for both 12-hour and 24-hour format.

In this example, Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

Table 39.39 Alarm Setting in 12-Hour Format (RTCAnCTL0.RTCAnAMPM = 0)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00 am	01_H	07_H	00_H
Sunday, Monday 12:15 pm	03_H	32_H	15_H
Monday, Wednesday, Friday 5:30 pm	$2A_H$	25_H	30_H
Daily, 10:45 pm	$7F_H$	30_H	45_H

Table 39.40 Alarm Setting in 24-Hour Format (RTCAnCTL0.RTCAnAMPM = 1)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00	01_H	07_H	00_H
Sunday, Monday 12:15	03_H	12_H	15_H
Monday, Wednesday, Friday 17:30	$2A_H$	17_H	30_H
Daily, 22:45	$7F_H$	22_H	45_H

39.5 Procedures

39.5.1 Procedures for Setup, Writing and Reading

Following subsections provide flow charts that illustrate procedures for RTCA setup and for reading and writing RTCA clock counters.

39.5.1.1 Initial Setting of RTCA

RTCA must be stopped before setting initial setting value of each counter.

(1) RTCA Stop Procedure

Stop RTCA according to following flow.

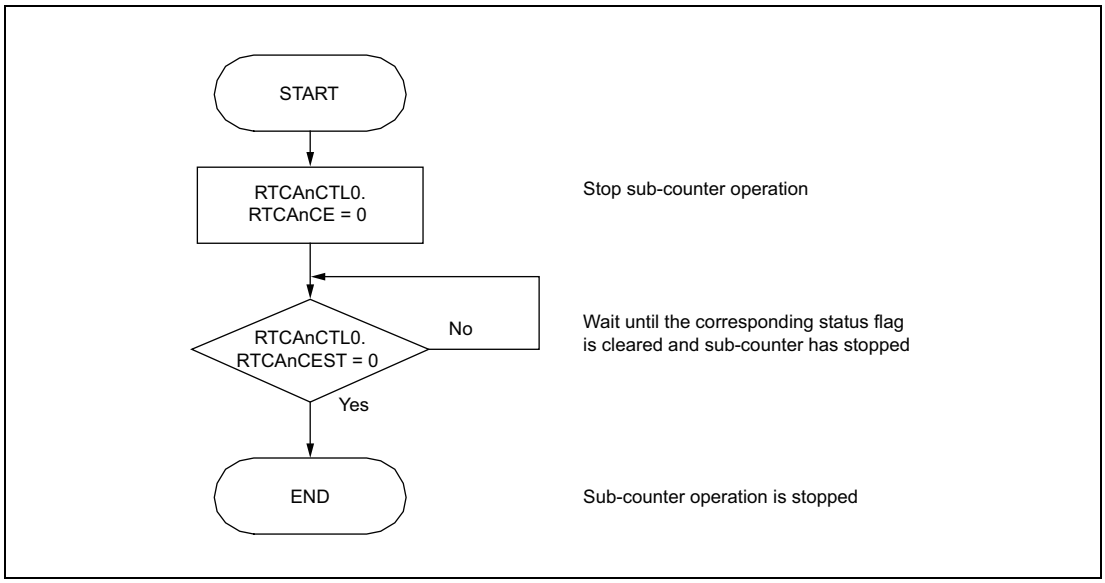


Figure 39.2 RTCA Stop Procedure

(2) RTCA Initialization Procedure

Perform initial setting of RTCA according to following flow:

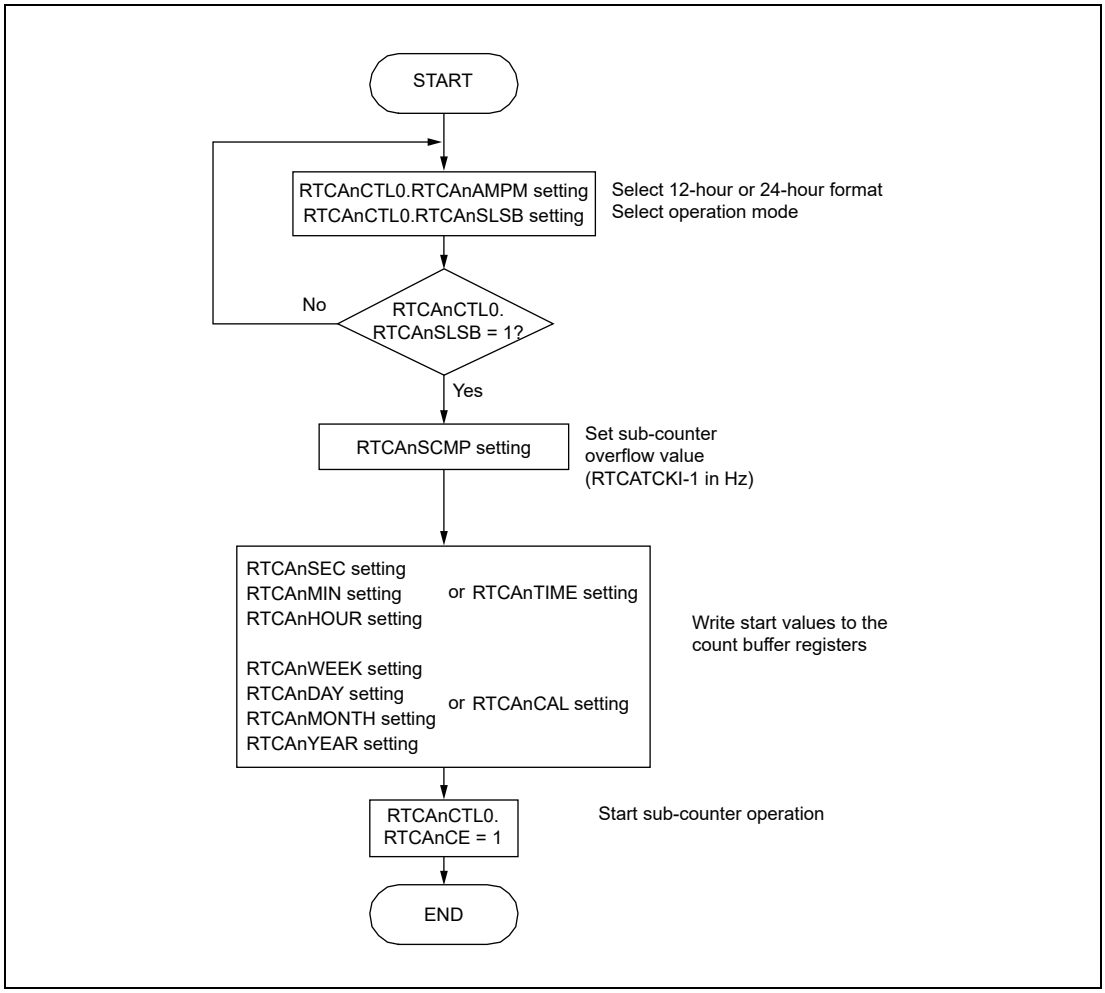


Figure 39.3 RTCA Initial Setup Procedure

CAUTION

Internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required before clock counter starting behind END of above flow.

Therefore, PCLK must be continuously supplied until completion of initial setting.

Check that RTCAnCTL0.RTCAnCEST = 1, when supply of PCLK is stopped after setting initial setting value of RTCA.

39.5.1.2 Updating Clock Counters

Clock counters RTCAnSECC to RTCAnYEARC can be stopped and updated while sub-counter is running.

To update clock counter when sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow flowchart shown below.

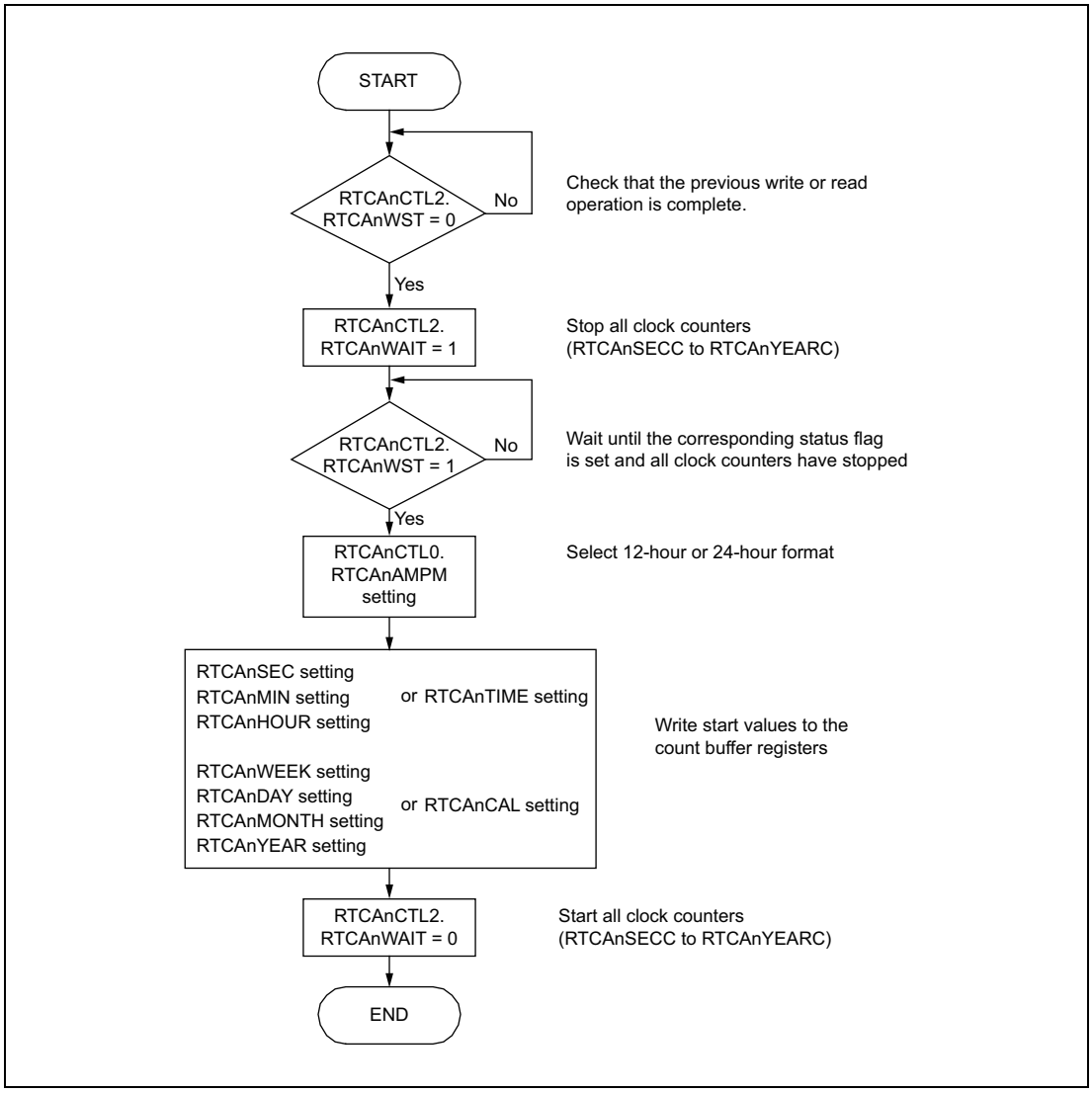


Figure 39.4 Updating Clock Counter Values

CAUTIONS

1. **Internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before clock counter updating behind END of above flow.
   Therefore, PCLK must be continuously supplied until completion of clock counter updating.
   Check that RTCAnCTL2.RTCAnWST = 0 before stopping supply of PCLK after completion of clock counter updating.
2. **Update procedure must be completed within one second. Otherwise Real-Time Clock will not count correctly any more:**
3. **Only one sub-counter overflow can be held internally and increment seconds counter after restarting clock counters if value is held.**
4. **If sub-counter overflows more than once during clock counter stop, overflow count cannot be held internally. Thus seconds counter is incremented by one instead of by two when it is restarted.**

39.5.1.3 Reading Clock Counters

There are two methods to read clock counters while sub-counter operation is enabled:

- Reading count buffer registers
- Reading counter registers

Advantages and disadvantages of two methods are summarized in following table.

Table 39.41 Comparison of Two Read Methods

	Advantage	Disadvantage
Reading count buffer registers	It is unnecessary to read clock counters several times because clock counters are read synchronously.	A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.
Reading count registers	Program wait state does not occur.	If sub-counter increments, clock counters must be read several times because they are read asynchronously to RTCATCKI.

(1) Procedure for Reading Count Buffer Registers

Following operations are necessary:

1. Stop all clock counters (RTCAnCTL2.RTCAnWAIT = 1). Value of clock counters is transferred to corresponding count buffer registers.
2. Read count buffer registers.

A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.

Maximum delay is three PCLK periods plus two RTCATCKI periods.

To read count buffer register when sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1), follow flowchart shown below.

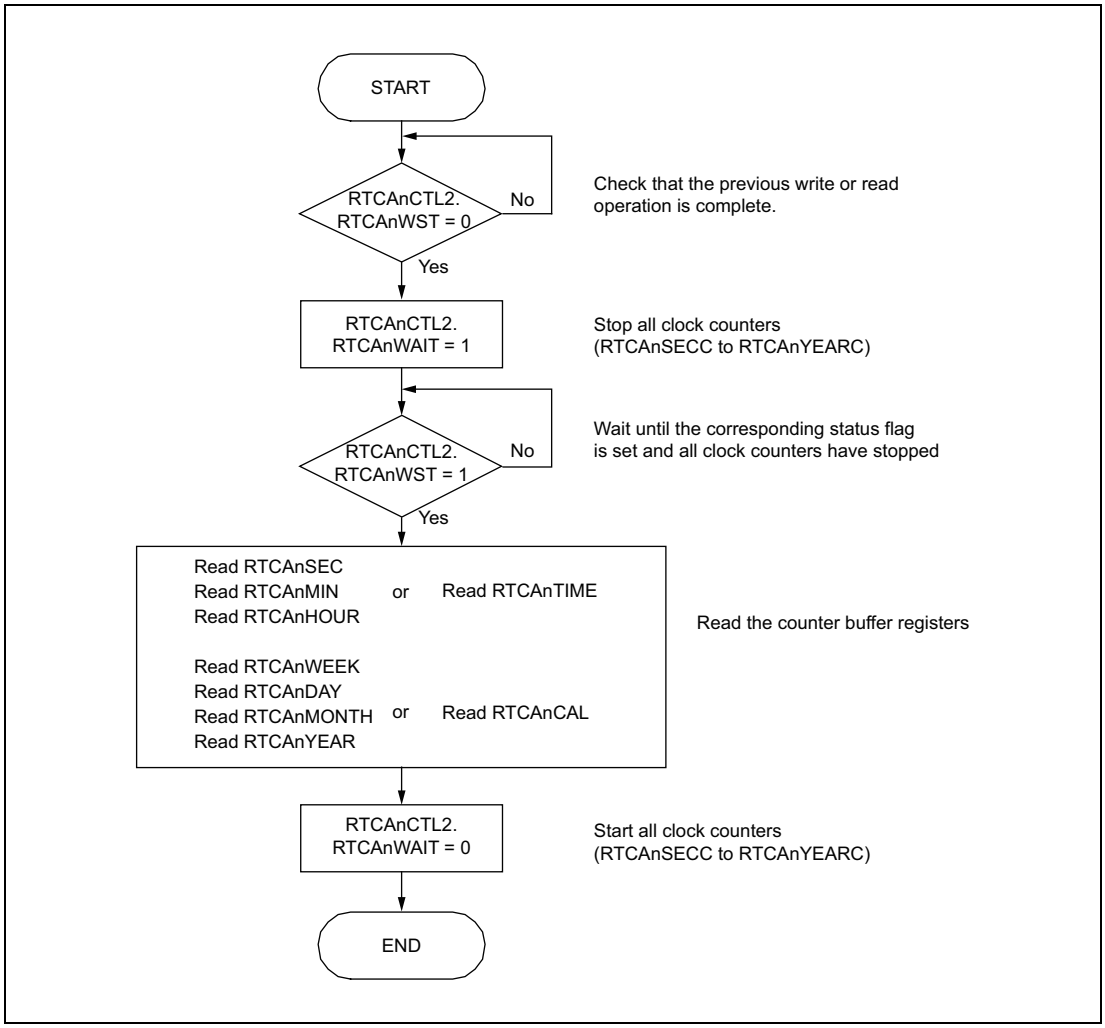


Figure 39.5 Reading Clock Count Buffer Registers

CAUTIONS

1. **Internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before resuming counter behind END of above flow.
   Therefore, PCLK must be continuously supplied until counter resuming.
   Check that RTCAnCTL0.RTCAnCEST = 1 first to stop supply of PCLK after count buffer register reading.
2. **Reading procedure must be completed within one second. Otherwise Real-Time Clock will not count correctly any more.**
3. **Only one sub-counter overflow can be held internally. If there is a value held internally when clock counter restarts, seconds counter will be incremented by 1.**
4. **If sub-counter overflows more than once during clock counter stop, overflow count cannot be held internally. Thus seconds counter is incremented by one instead of by two when it is restarted.**

(2) Procedure for Reading Counter Registers Directly

To ensure that sub-counter did not overflow while reading counters, seconds counter RTCAnSECC must be read twice in beginning and at end of procedure. first read value is compared with second read value.

- First read value = second read value:
  No overflow of sub-counter occurred during counter read operation.
- First read value ≠ second read value:
  Overflow of sub-counter occurred during counter read operation. counters must be read again to get current counter values.

To read counter register directly when sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow flowchart shown below.

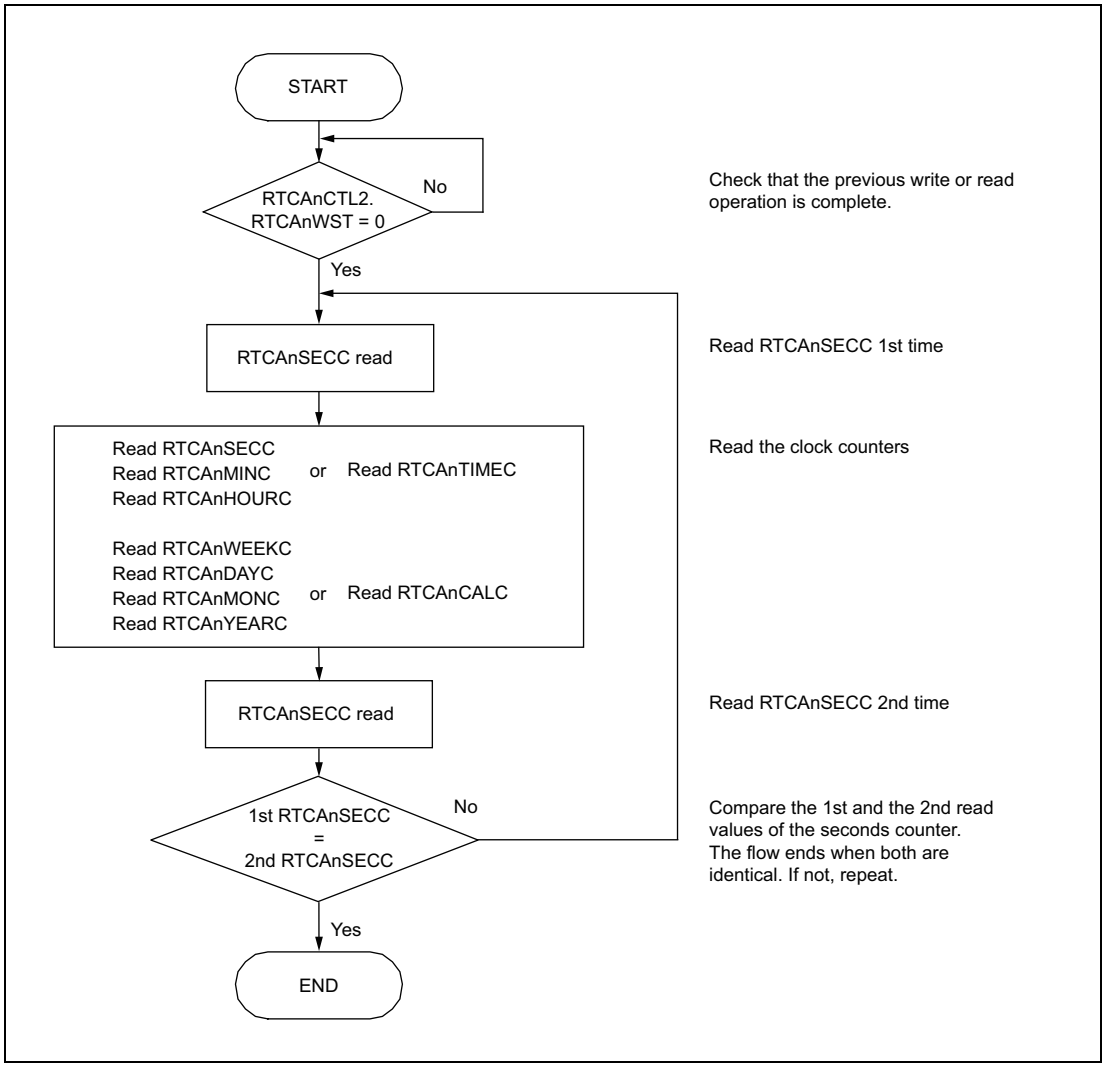


Figure 39.6 Reading Clock Counter Registers

NOTE

Procedure must be completed within one second.

39.5.1.4 Reading RTCAnSRBU

RTCAnSRBU is read buffer register for sub-counter.

When sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), read RTCAnSRBU according to following flow.

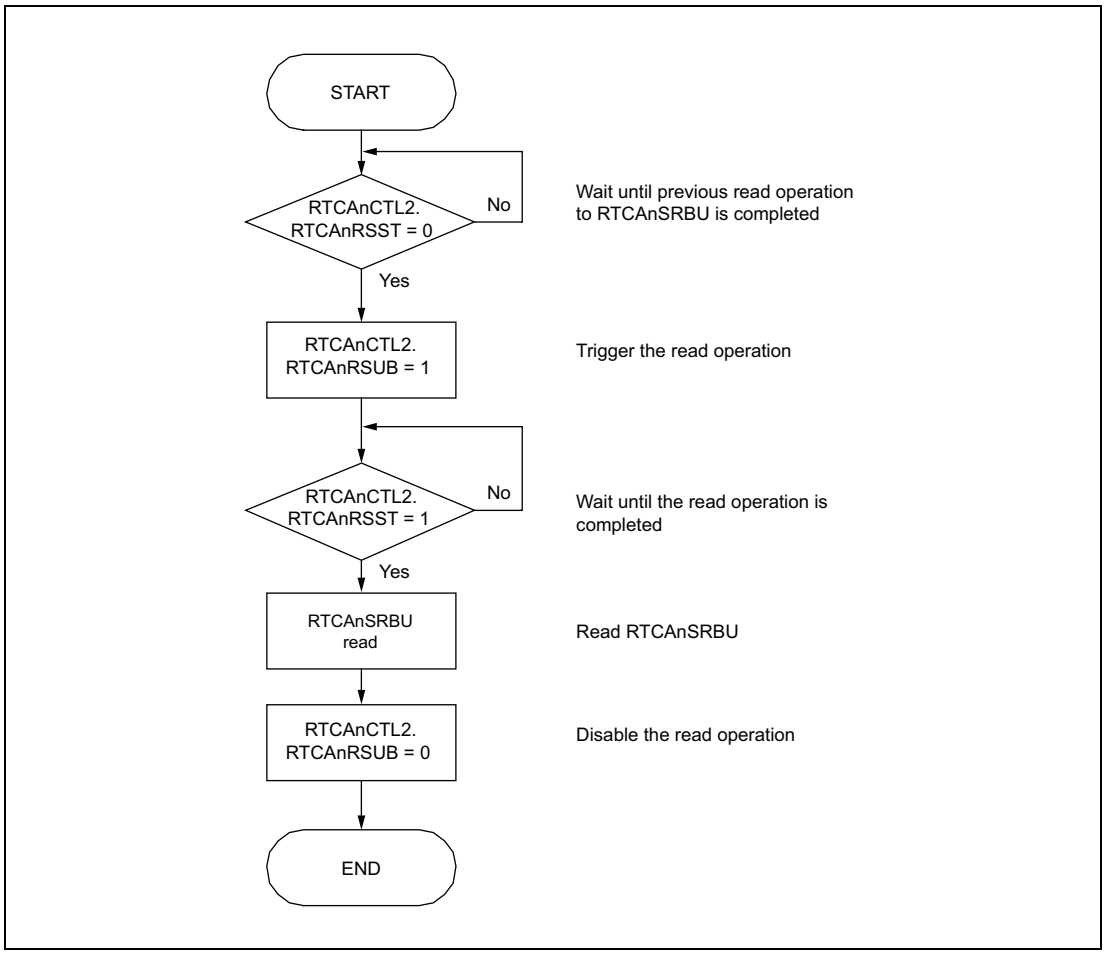


Figure 39.7 Reading RTCAnSRBU Register

39.5.1.5 Writing to RTCAnSCMP

RTCAnSCMP is sub-counter compare register.

When sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSCMP according to flow described below.

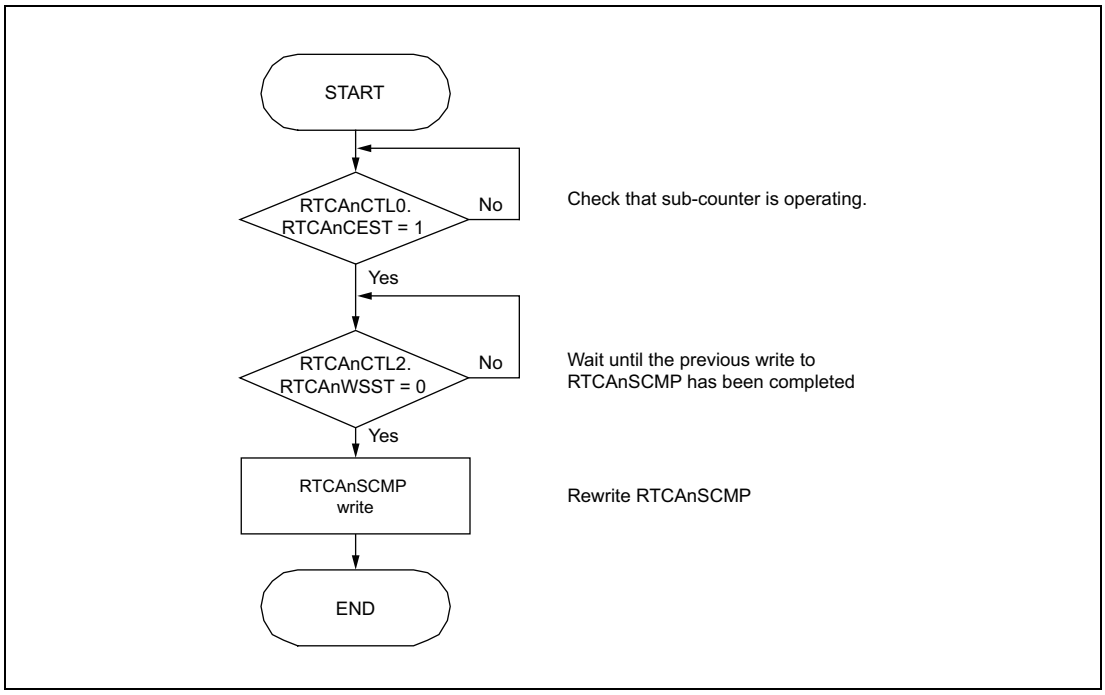


Figure 39.8 Writing RTCAnSCMP Register

NOTE

While sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), status flag RTCAnCTL2.RTCAnWSST is set when RTCAnSCMP is written to. It is cleared when write operation to RTCAnSCMP is completed. This is synchronous with next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWSST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWSST = 1 at beginning of this flow).

39.5.2 Timing Diagrams

39.5.2.1 Timing of Counter Start

Following diagram illustrates counter start after setting time in buffer registers.

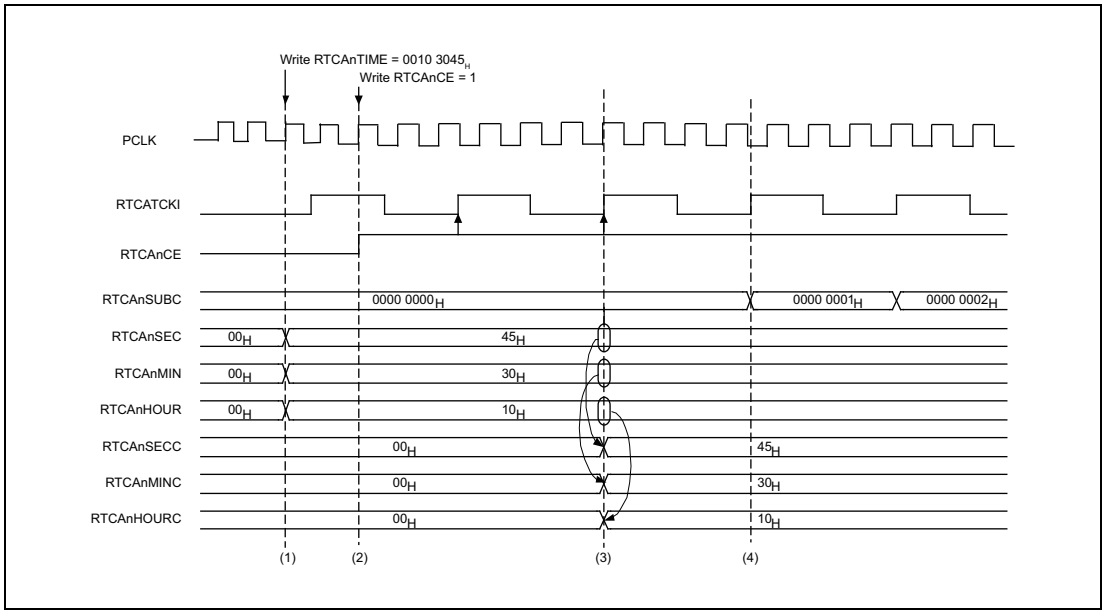


Figure 39.9 Counter Start Timing

Timing diagram above shows following:

(1) Initial setting value of time count buffer is set to 10:30:45 by setting RTCAnTIME = 0010 3045_H. Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(2) Sub-counter operation is started by setting RTCAnCTL0.RTCAnCE = 1.

(3) When second rising edge of RTCATCKI occurs, buffer register values are loaded to corresponding count registers.

(4) When next rising edge of RTCATCKI occurs, count up of sub-counter starts.

39.5.2.2 Timing of Clock Counter Update while Counter Is Enabled

Following diagram illustrates counter restart after setting time in buffer registers.

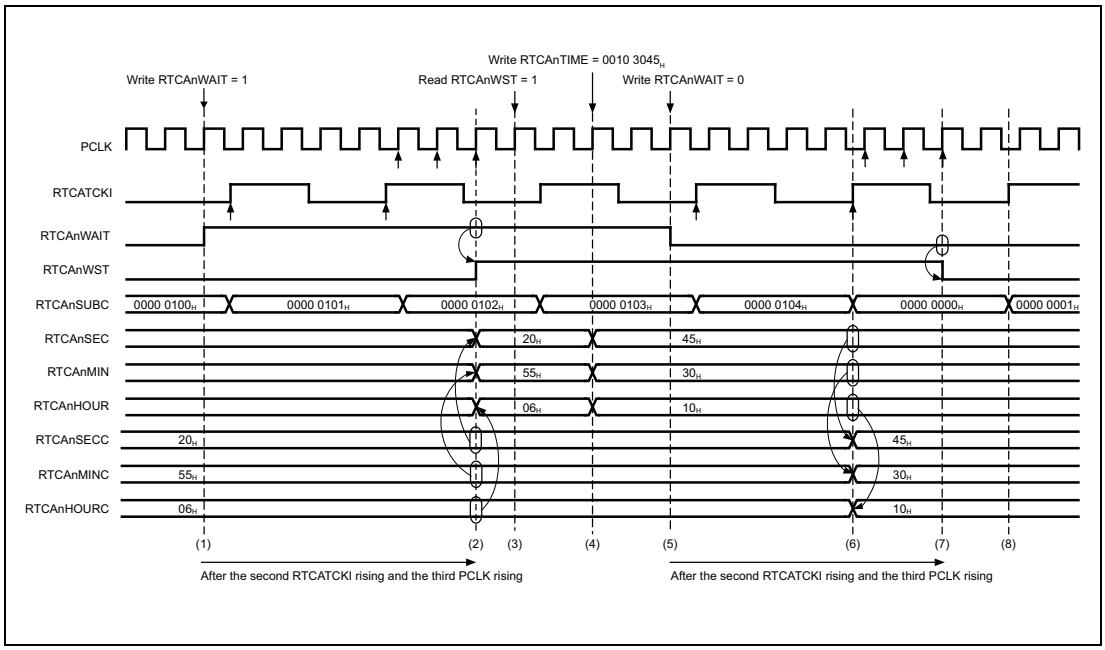


Figure 39.10 Clock Counter Update Timing

Timing diagram above shows following:

(1) Trigger clock counters stop (RTCAnCTL2.RTCAnWAIT = 1).

(2) RTCAnCTL2.RTCAnWST is set to 1 after second rising edge of RTCATCKI and third rising edge of PCLK, and counter clock stops. Sub-counter continues counting.

(3) RTCAnCTL2.RTCAnWST = 1 can be readable.

(4) Initial setting value of time count buffer is set to 10:30:45 by setting RTCAnTIME to 0010 3045_H.
Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(5) Trigger clock counters restart (RTCAnCTL2.RTCAnWAIT = 0).

(6) When second rising edge of RTCATCKI occurs, values of buffer registers are loaded to corresponding count registers. Write operation to RTCAnSECC is performed and RTCAnSUBC is cleared.

(7) When third rising edge of PCLK occurs, RTCAnCTL2.RTCAnWST is set to 0.

(8) Clock counter operation is resumed.

39.5.2.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled

Following diagram illustrates timing when reading sub-counter read buffer RTCAnSRBU.

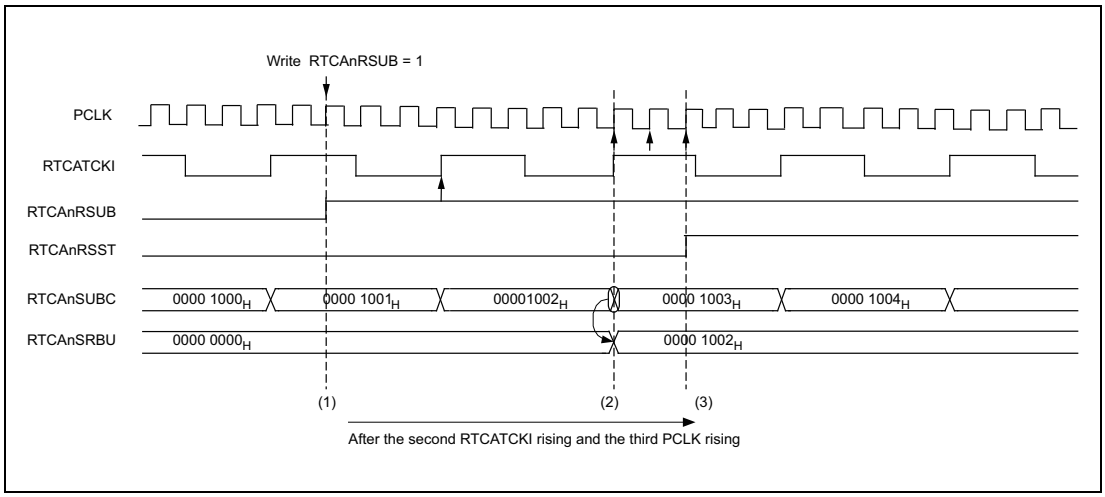


Figure 39.11 Timing when Reading Sub-Counter Read Buffer Register Value

Timing diagram above shows following:

(1) Setting RTCAnRSUB = 1 triggers loading of sub-counter value to RTCAnSRBU.

(2) When second rising edge of RTCATCKI occurs, value of RTCAnSUBC is loaded to RTCAnSRBU.

(3) When third rising edge of PCLK occurs, RTCAnCTL2.RTCAnRSST is set to 1 and RTCAnSRBU can be read.

Section 40 Encoder Timer A (ENCA)

This section contains a generic description of the Encoder Timer A (ENCA).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ENCA.

40.1 Features ENCA for RH850/U2A-EVA

40.1.1 Number of Units and Channels

This microcontroller has the following number of ENCA units.

Each ENCA unit has one channel ENCA. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 40.1 Number of Units

Product	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	2 (n = 0, 1)	1 (n = 1)	—	1 (n = 1)
Name	ENCAn									

Table 40.2 Unit Configurations and Channels

Unit Name (Channel Name) ENCAn	Channels per Unit	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
ENCA0	1	√	√	√	√	√	√	√	—	—	—
ENCA1	1	√	√	√	√	√	√	√	√	—	√

Table 40.3 Index

Index	Description
n	Throughout this section, the individual ENCA units are identified by the index “n” (n = 0, 1); for example, ENCAnCTL is the ENCAn control register.

40.1.2 Register Base Addresses

ENCAn base addresses are listed in the following table.

ENCAn register addresses are given as offsets from the base addresses.

Table 40.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ENCA0_base>	FFBF 3000_H	P-Bus Group 5
<ENCA1_base>	FFBF 3100_H	P-Bus Group 5

40.1.3 Clock Supply

Clock supply by and to ENCAn is listed in the following table.

Table 40.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ENCAn	PCLK	CLK_HSB

40.1.4 Interrupt Requests and Error Notifications

ENCAn interrupt requests are listed in the following table.

Table 40.6 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number	DTS Trigger Number	Others
ENCA0					
INTENCA0IOV	ENCA0 overflow interrupt	534	Group0-181	Group1-54	—
INTENCA0I0	ENCA0 capture/compare match interrupt 0	536	Group0-183	Group1-56	PIC2
INTENCA0I1	ENCA0 capture/compare match interrupt 1	537	Group0-184	Group1-57	PIC1, PIC2
INTENCA0IUD	ENCA0 underflow interrupt	535	Group0-182	Group1-55	—
INTENCA0IEC	ENCA0 encoder clear interrupt	538	Group0-185	Group1-58	PIC1, PIC2
ENCA1					
INTENCA1IOV	ENCA1 overflow interrupt	539	Group0-186	Group1-59	—
INTENCA1I0	ENCA1 capture/compare match interrupt 0	541	Group0-188	Group1-61	PIC2
INTENCA1I1	ENCA1 capture/compare match interrupt 1	542	Group0-189	Group1-62	PIC1, PIC2
INTENCA1IUD	ENCA1 underflow interrupt	540	Group0-187	Group1-60	—
INTENCA1IEC	ENCA1 encoder clear interrupt	543	Group0-190	Group1-63	PIC1, PIC2

This module has no error notifications.

40.1.5 Reset Sources

ENCAn reset sources are listed in the following table. ENCAn is initialized by these reset sources.

Table 40.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ENCAn	All registers	√	√	√	√	√	√	—

40.1.6 External Input/Output Signals

External input/output signals of ENCAn are listed below.

Table 40.8 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
ENCA0			
ENCA0TTIN0	I	ENCA0 capture trigger input 0*1	ENCA0TIN0*3
ENCA0TTIN1	I	ENCA0 capture trigger input 1*1	ENCA0TIN1*2
ENCA0E0	I	ENCA0 encoder input 0*1	ENCA0E0*2
ENCA0E1	I	ENCA0 encoder input 1*1	ENCA0E1*2
ENCA0EC	I	ENCA0 encoder clear input*1	ENCA0EC*2
ENCA1			
ENCA1TTIN0	I	ENCA1 capture trigger input 0*1	ENCA1TIN0*3
ENCA1TTIN1	I	ENCA1 capture trigger input 1*1	ENCA1TIN1*2
ENCA1E0	I	ENCA1 encoder input 0*1	ENCA1E0*2
ENCA1E1	I	ENCA1 encoder input 1*1	ENCA1E1*2
ENCA1EC	I	ENCA1 encoder clear input*1	ENCA1EC*2

Note 1. Setting of the noise filter for the port is required when the input pin is used. For details, see **Section 2.7, Noise Filter & Edge/Level Detector**.

Note 2. The input signal can be selected by the PIC function. For details, see **Section 41.2.2.23, PIC1REG30 — Timer Input/Output Control Register 30**.

Note 3. The input signal can be selected by the PIC function. For details, see **Section 41.3.2.11, PIC2ENCAnTSEL — Encoder Timer n Trigger Selection Control Register (n = 0, 1)**.

40.1.7 Internal Input/Output Signals

The internal input/output signals of ENCAn are listed below.

Table 40.9 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
ENCAnTSST	Simultaneous start trigger	PIC1
ENCAnTTIN1	ENCAn capture trigger input 1	PIC1
ENCAnTTIN0	ENCAn capture trigger input 0	PIC2
ENCAnE0	ENCAn encoder input 0	PIC1
ENCAnE1	ENCAn encoder input 1	PIC1
ENCAnEC	ENCAn encoder clear input	PIC1
ENCAnTUDC	ENCAn down-count enable signal output	PIC2
ENCAnTCKEN	ENCAn count clock output	PIC2

40.2 Overview

40.2.1 Functional Overview

- Generation of the counter control signal from the encoder input signal, and performs count operation in synchronization with PCLK.
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow and output of error flags and error interrupts
- Five interrupts: 2 capture/compare interrupts, 1 counter clear interrupt, 1 overflow interrupt, and 1 underflow interrupt

40.2.2 Block Diagram

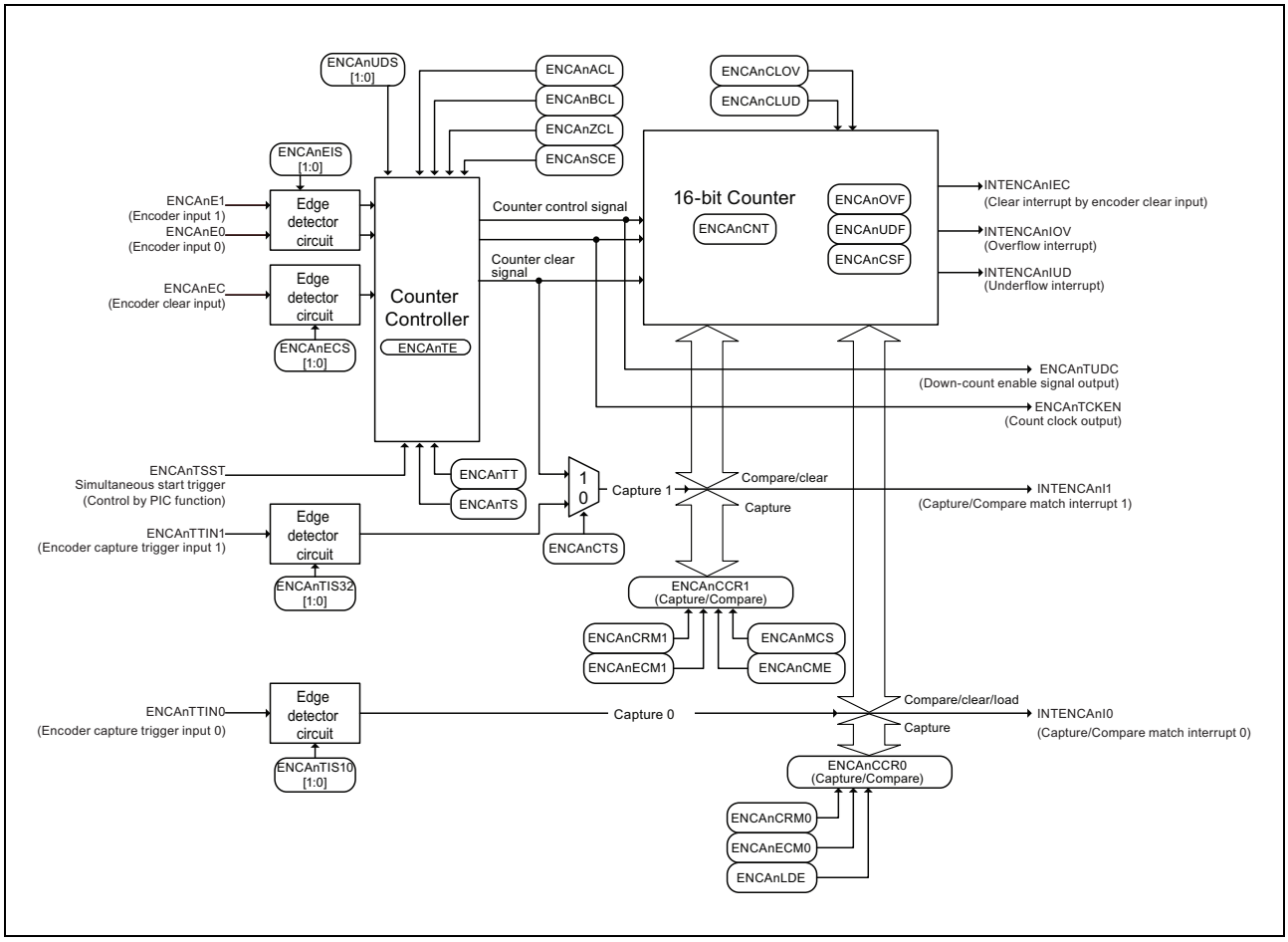


Figure 40.1 Encoder Timer Block Diagram

40.3 Registers

40.3.1 List of Registers

ENCA registers are listed in the following table.

For details about <ENCA_base>, see **Section 40.1.2, Register Base Addresses**.

Table 40.10 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ENCAn	ENCAn capture/compare register 0	ENCAnCCR0	<ENCAn_base>	16	*1	—
ENCAn	ENCAn capture/compare register 1	ENCAnCCR1	<ENCAn_base> + 04_H	16	*1	—
ENCAn	ENCAn counter register	ENCAnCNT	<ENCAn_base> + 08_H	16	*1	—
ENCAn	ENCAn status flag register	ENCAnFLG	<ENCAn_base> + $0C_H$	8	*1	—
ENCAn	ENCAn status flag clear register	ENCAnFGC	<ENCAn_base> + 10_H	8	*1	—
ENCAn	ENCAn timer enable status register	ENCAnTE	<ENCAn_base> + 14_H	8	*1	—
ENCAn	ENCAn timer start trigger register	ENCAnTS	<ENCAn_base> + 18_H	8	*1	—
ENCAn	ENCAn timer stop trigger register	ENCAnTT	<ENCAn_base> + $1C_H$	8	*1	—
ENCAn	ENCAn I/O control register 0	ENCAnIOC0	<ENCAn_base> + 20_H	8	*1	—
ENCAn	ENCAn control register	ENCAnCTL	<ENCAn_base> + 40_H	16	*1	—
ENCAn	ENCAn I/O control register 1	ENCAnIOC1	<ENCAn_base> + 44_H	8	*1	—

Note 1. n = 0: PBG51#8
n = 1: PBG51#9

40.3.2 ENCAnCTL — ENCA Control Register

This register is used to configure various operation settings of the Encoder Timer.

Access: This register can be read or written in 16-bit units.
Writing to this register during operation is prohibited.

Address: <ENCAn_base> + 40_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAn CME	ENCAn MCS	—	—	—	—	ENCAn CRM1	ENCAn CRM0	ENCAn CTS	—	—	ENCAn LDE	ENCAn ECM1	ENCAn ECM0	ENCAnUDS [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 40.11 ENCAnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	ENCAnCME	Encoder Clear Mask Enable This bit is used to enable/disable masking of compare match interrupt detection when the compare function is used. 0: Disables the compare match interrupt (INTENCAnI1) mask function for the ENCAnCCR1 register 1: Enables the compare match interrupt (INTENCAnI1) mask function for the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 0. When this bit is set to 1, setting ENCAnECM1 to 1 is prohibited.
14	ENCAnMCS	Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare match interrupt detection INTENCAnI1 when the compare function is used. This bit is valid only when ENCAnCRM1 = 0. 0: Masking of compare match interrupt detection is cancelled when the ENCAnCCR1 register is written. 1: Masking of compare match interrupt detection is cancelled when one of the following three operations is performed. – Timer counter clear operation accompanying encoder clear input – Timer counter clear operation upon compare match between ENCAnCNT and ENCAnCCR0 when ENCAnECM0 = 1 – Loading from ENCAnCCR0 to timer counter upon underflow detection when ENCAnLDE = 1
13 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	ENCAnCRM1	ENCAnCCR1 Register Mode 0: ENCAnCCR1 used as compare register. 1: ENCAnCCR1 used as capture register.
8	ENCAnCRM0	ENCAnCCR0 register mode bit 0: ENCAnCCR0 used as compare register. 1: ENCAnCCR0 used as capture register.
7	ENCAnCTS	ENCAnCCR1 Capture Trigger Select This is a trigger selection bit for the capture operation to the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 1. 0: Uses ENCAnTTIN1 of capture trigger 1 signal as the trigger for capturing to the ENCAnCCR1 register. 1: Uses the encoder clear signal selected by ENCAnSCE as the capture trigger to the ENCAnCCR1 register.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 40.11 ENCAnCTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	ENCAnLDE	ENCAn Counter Load Enable This bit is used to enable/disable setting value loading to the counter upon underflow occurrence. This bit is valid only when ENCAnCRM0 = 0. When ENCAnCRM0 = 1, loading of the ENCAnCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit. 0: Disable loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow. 1: Enable loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow.
3	ENCAnECM1	Encoder Clear Mode 1 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR1 setting value. This bit is valid only when ENCAnCRM1 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value if the next count is a down-count.
2	ENCAnECM0	Encoder Clear Mode 0 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR0 setting value. This bit is valid only when ENCAnCRM0 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value if the next count is a up-count.
1, 0	ENCAnUDS[1:0]	Up/down Count Selection 1 and 0 These are the counter up/down control bits that use ENCAnE0 and ENCAnE1. 00: Upon detection of valid edge of ENCAnE0, - down-count when ENCAnE1 = H, - up-count when ENCAnE1 = L 01: Upon detection of valid edge of ENCAnE0, up-count, Upon detection of valid edge of ENCAnE1, down-count 10: At rising edge of ENCAnE0, down-count At falling edge of ENCAnE0, up-count However, count operation is performed only when ENCAnE1 = L. 11: Detection of both edges of ENCAnE0 and ENCAnE1. Judgment of count operation combining both detected edge and level.

40.3.3 ENCAnIOC0 — ENCA I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCAnTTIN0, ENCAnTTIN1).

Access: This register can be read or written in 8-bit units.

Address: <ENCAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	ENCAnTIS32[1:0]		ENCAnTIS10[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 40.12 ENCAnIOC0 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	ENCAnTIS32 [1:0]	Input Edge Selection for Capture Trigger 1 These bits are valid only when ENCAnCTL.ENCAnCRM1 = 1 and ENCAnCTL.ENCAnCTS = 0. All other settings of ENCAnCRM1 and ENCAnCTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAnTIS10 [1:0]	Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCAnCTL.ENCAnCRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

40.3.4 ENCAnIOC1 — ENCA I/O Control Register 1

This register is used to perform the clear condition setting and edge selection upon encoder input.

Access: This register can be read or written in 8-bit units.
Writing to this register during operation is prohibited.

Address: <ENCAn_base> + 44_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS[1:0]		ENCAnEIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.13 ENCAnIOC1 Register Contents

Bit Position	Bit Name	Function
7	ENCAnSCE	Encoder Special-clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCAnUDS[1:0] to 10_B or 11_B. The operation is not guaranteed if this bit is set to 1 with ENCAnUDS[1:0] set to 00_B or 01_B. 0: Clears the counter upon detection of ENCAnEC valid edge (set with ENCAnECS[1:0]). 1: Clears the counter upon detection of input level condition of ENCAnEC, ENCAnE1, and ENCAnE0 (set with ENCAnZCL bit, ENCAnBCL bit, and ENCAnACL bit).
6	ENCAnZCL	Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCAnEC) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCAnBCL	Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCAnE1) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
4	ENCAnACL	Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 input (ENCAnE0) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCAnECS[1:0]	Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCAnSCE = 0; they are invalid when ENCAnSCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAnEIS[1:0]	Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCAnUDS[1:0] is 00_B or 01_B, and are invalid when ENCAnUDS[1:0] is 10_B or 11_B. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

40.3.5 ENCAnFLG — ENCA Status Flag Register

This register is holds the status flags of the timer counter of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENCAnCSF	ENCAnUDF	ENCAnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40.14 ENCAnFLG Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	ENCAnCSF	Counter Status Flag This bit reflects the current timer counter operation. This bit is cleared at the start of count operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCAnUDF	Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: – 1 is written to ENCAnFGC.ENCAnCLUD – ENCAnTS is set while ENCAnTE = 0 – The input signal of ENCAnTSST is set to “High level” 1: This flag is set to 1 upon occurrence of an underflow during the encoder timer count operation.
0	ENCAnOVF	Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: – 1 is written to ENCAnFGC.ENCAnCLOV – ENCAnTS is set while ENCAnTE = 0 – The input signal of ENCAnTSST is set to “High level” 1: This flag is set to 1 upon occurrence of an overflow during the encoder timer count operation.

40.3.6 ENCAnFGC — ENCA Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAnFLG.

Access: This register is a write-only register that can be written in 8-bit units.
This register always returns 0 when read.

Address: <ENCAn_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ENCAnCLUD	ENCAnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 40.15 ENCAnFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	ENCAnCLUD	Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCAnUDF of the ENCAnFLG register (clears underflow detection).
0	ENCAnCLOV	Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCAnOVF of the ENCAnFLG register (clears overflow detection).

40.3.7 ENCAnCCR0 — ENCA Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

Access: This register can be read or written in 16-bit units.
When used as a capture register, this register is only readable. Writing to this register is ignored.
When used as a compare register, this register is readable/writable.

Address: <ENCAn_base> + 00_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.16 ENCAnCCR0 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR0 [15:0]	Capture/Compare Register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAnCTL.ENCAnLDE setting. See the description of the ENCAnLDE bit in ENCA control register ENCAnCTL for details. • If ENCAnCTL.ENCAnCRM0 = 0: ENCAnCCR0 is compare register. Set the value to be compared with the timer counter value. • If ENCAnCTL.ENCAnCRM0 = 1: ENCAnCCR0 is capture register The captured timer counter value is stored.

40.3.8 ENCAnCCR1 – ENCA Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

Access: This register can be read or written in 16-bit units.
When used as a capture register, this register is only readable. Writing to this register is ignored.
When used as a compare register, this register is readable/writable.

Address: <ENCAn_base> + 04_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.17 ENCAnCCR1 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR1 [15:0]	Capture/Compare Register 1 During capture operation, the trigger for capturing to this register differs according to the ENCAnCTL.ENCAnCTS setting. See the description of the ENCAnCTS bit in ENCA control register ENCAnCTL for details. • If ENCAnCTL.ENCAnCRM1 = 0: ENCAnCCR1 is compare register Set the value to be compared with the timer counter value. • If ENCAnCTL.ENCAnCRM1 = 1: ENCAnCCR1 is capture register The captured timer counter value is stored.

40.3.9 ENCAnCNT — ENCA Counter Register

This register is the 16-bit timer counter register.

Access: This register can be read or written in 16-bit units.
This register can be written only when the operation is stopped.

Address: <ENCAn_base> + 08_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.18 ENCAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCNT [15:0]	Counter Register • ENCAnTE.ENCAnTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. • ENCAnTE.ENCAnTE status: 0 → 1 (operation start)): Count operation start Up/down count operation is started with the set arbitrary value. • ENCAnTE.ENCAnTE status: 1 (operating): Counting Up/down count operation is performed. • ENCAnTE.ENCAnTE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped.

40.3.10 ENCAnTE — ENCA Timer Enable Status Register

This register indicates the operating status of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 40.19 ENCAnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ENCAnTE	Timer Status Enable This is a status bit that indicates the operation enabled/stopped status of ENCAn. This bit is cleared to 0 when 1 is written to ENCAnTT.ENCAnTT. This bit is set to 1 when 1 is written to ENCAnTS.ENCAnTS, or when the input signal of ENCAnTSST is set to High level. 0: Operation stopped status 1: Operation enabled status

40.3.11 ENCAnTS — ENCA Timer Start Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation enabled state.

Access: This register is a write-only register that can be written in 8-bit units.
This register always returns 00_H when read. This register can be written only when ENCAnTE.ENCAnTE is 0.

Address: <ENCAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40.20 ENCAnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTS	Timer Start Trigger This is the trigger bit that sets the ENCAn to the operation enabled state. 0: Writing is ignored. 1: The ENCAn is set to the operation enabled state by setting ENCAnTE.ENCAnTE = 1.

40.3.12 ENCAnTT — ENCA Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation stopped state.

Access: This register is a write-only register that can be written in 8-bit units.
This register always returns 00_H when read.

Address: <ENCAn_base> + $1C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ENCAnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 40.21 ENCAnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTT	Timer Stop Trigger This is the trigger bit that sets the ENCAn to the operation stopped state. 0: Writing is ignored. 1: Clears ENCAnTE.ENCAnTE to 0, to set the ENCAn to the count operation stopped state.

40.4 Operation

The ENCAn operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCAnCCR0 and ENCAnCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

40.4.1 Timer Counter Operation

The timer counter operations of the ENCAn are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

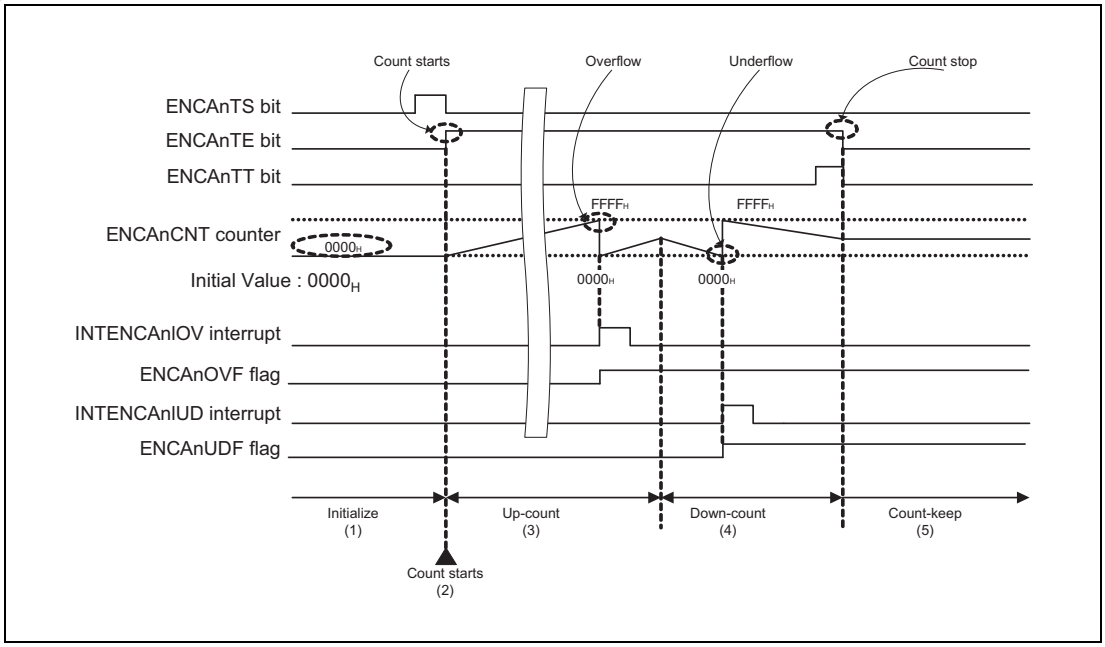


Figure 40.2 Timer Counter Initial Setting Value Setting/Start/Stop

(1) Timer counter initial setting value setting

The initial setting value of the ENCAn counter register (ENCAnCNT) can be set in the counter operation stopped status (ENCAnTE = 0).

(2) Timer counter startup

By writing 1 to the timer start trigger bit (ENCAnTS), the timer status enable bit (ENCAnTE) is set to 1, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

(3) Overflow operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. If the counter value changes from $FFFF_H$ to 0000_H, an overflow interrupt (INTENCAnlOV) is generated, and the overflow flag (ENCAnOVF) is set to 1. The overflow flag (ENCAnOVF) is cleared to 0 when 1 is set to the overflow flag clear (ENCAnCLOV). For details about the operation, see **Section 40.4.8.1, Overflow Occurrence and Overflow Flag Clear Operation**.

(4) Underflow operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. If the counter value changes from 0000_H to $FFFF_H$, an underflow interrupt (INTENCAnlUD) is generated, and the underflow flag (ENCAnUDF) is set to 1. The underflow flag (ENCAnUDF) is cleared to 0 when 1 is set to the underflow flag clear (ENCAnCLUD). For details about the operation, see **Section 40.4.8.2, Underflow Occurrence and Underflow Flag Clear Operation**.

(5) Timer counter stop

By writing 1 to the timer stop trigger bit (ENCAnTT), the timer status enable bit (ENCAnTE) is cleared to 0, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stop.

40.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCAnE0, ENCAnE1) according to the settings of ENCAnUDS[1:0].

40.4.2.1 When ENCAnUDS[1:0] Bits in ENCAnCTL = 00_B

Table 40.22 When ENCAnUDS[1:0] = 00_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 input	ENCAnE1 input	Counting operation
0	0	Rising edge	High level	Down
		Falling edge		
		Both edges		
		Rising edge	Low level	Up
		Falling edge		
		Both edges		

The valid edge for ENCAnE0 is specified by setting ENCAnEIS[1:0].

"Counting operation" indicates that counting up or down proceeds when the valid edges of the ENCAnE0 signal matches the valid level ENCAnE1 signal.

The following timing chart shows counting operation when ENCAnUDS[1:0] = 00_B.

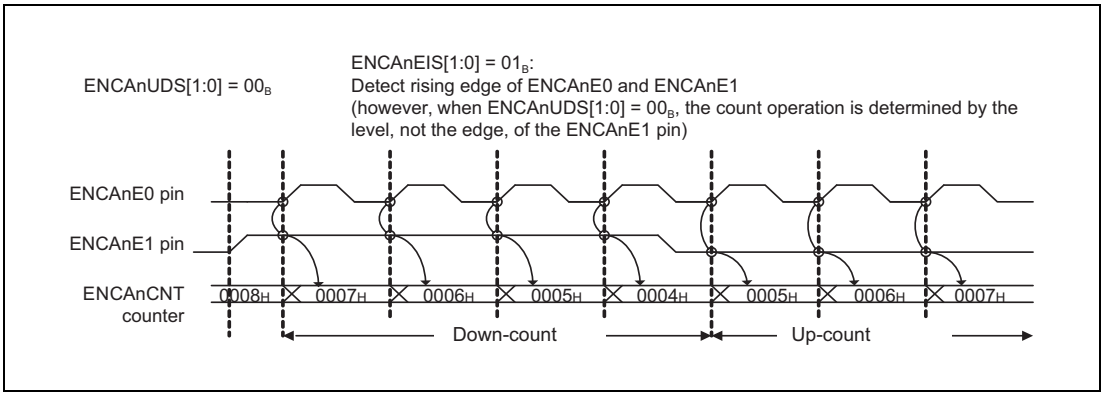


Figure 40.3 Counting Operation when ENCAnUDS[1:0] in ENCAnCTL = 00_B

40.4.2.2 When ENCAnUDS[1:0] Bits in ENCAnCTL = 01_B

Table 40.23 When ENCAnUDS[1:0] = 01_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 input	ENCAnE1 input	Counting operation
0	1	Low level	Rising edge	Down
			Falling edge	
			Both edges	
		High level	Rising edge	
			Falling edge	
			Both edges	
		Rising edge	Low level	Up
		Falling edge		
		Both edges		
		Rising edge	High level	
		Falling edge		
		Both edges		
		Simultaneous input		Hold

The valid edges for ENCAnE0 and ENCAnE1 are specified by setting ENCAnEIS[1:0].

“Counting operation” indicates that counting up or down proceeds when the valid edges and levels of the signals on ENCAnE0 and ENCAnE1 match.

The following timing chart shows counting operation when ENCAnUDS[1:0] = 01_B.

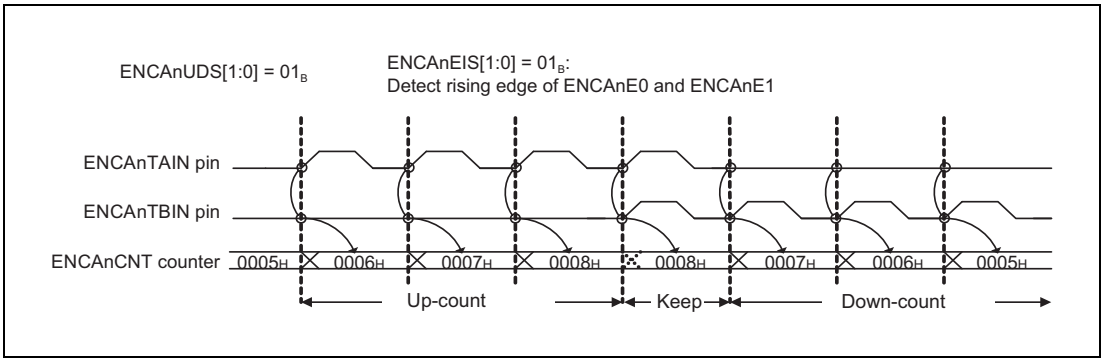


Figure 40.4 Counting Operation when ENCAnUDS[1:0] in ENCAnCTL = 01_B

40.4.2.3 When ENCAnUDS[1:0] Bits in ENCAnCTL = 10_B

Table 40.24 When ENCAnUDS[1:0] = 10_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 input	ENCAnE1 input	Counting operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

Valid edge specification for ENCAnE0 and ENCAnE1 (setting to ENCAnEIS[1:0]) is invalid.

The following timing chart shows counting operation when ENCAnUDS[1:0] = 10_B.

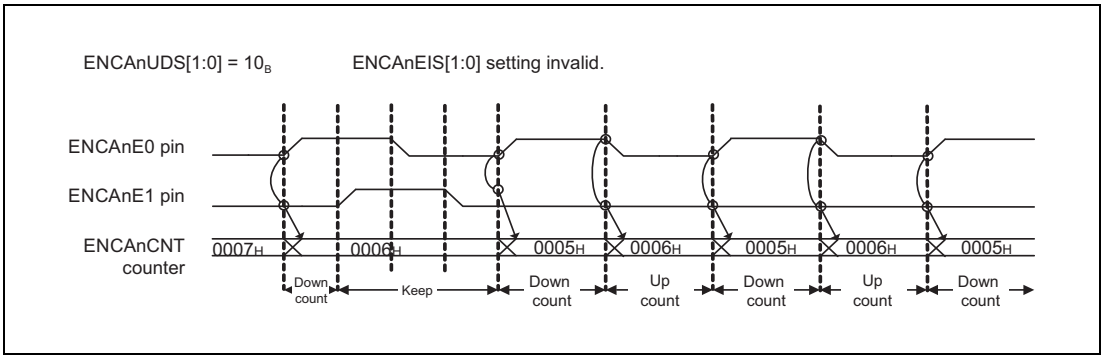


Figure 40.5 Counting Operation when ENCAnUDS[1:0] in ENCAnCTL = 10_B

40.4.2.4 When ENCAnUDS[1:0] Bits in ENCAnCTL = 11_B

Table 40.25 When ENCAnUDS[1:0] = 11_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 input	ENCAnE1 input	Counting operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Hold

Valid edge specification for ENCAnE0 and ENCAnE1 (settings of ENCAnEIS[1:0]) is invalid.

When the valid edges of the signals on ENCAnE0 and ENCAnE1 coincide, the count is retained.

The following timing chart shows counting operation when ENCAnUDS[1:0] = 11_B

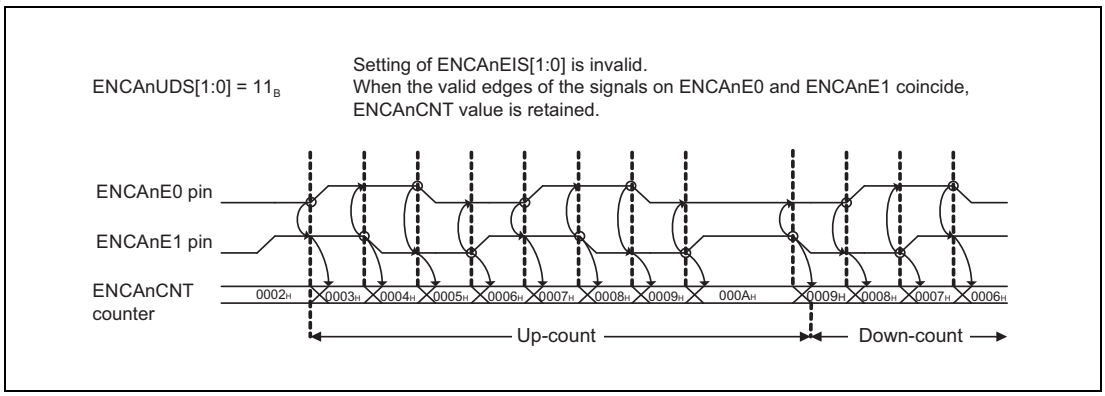


Figure 40.6 Counting Operation when ENCAnUDS[1:0] = 11_B

40.4.3 Timer Counter Clear Control by Encoder Input

The following conditions lead to clearing of the timer counter to 0000_H.

- Detection of a valid edge of the encoder clearing input signal (signal on the ENCAnEC pin)
- Level detection of the encoder input and encoder clearing input signals (signals on the ENCAnE0, ENCAnE1, and ENCAnEC pins)

Two types of clearing methods can be selected by controlling the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS[1:0] bits of the ENCAnIOC1 register.

Table 40.26 Timer Counter Clear Control by Encoder Input

Clearing method	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS[1:0]
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

40.4.3.1 Clearing Method when ENCAnSCE = 0

- Upon detection of the valid edge of ENCAnEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCAnEC is specified by the setting of the ENCAnECS[1:0] bits.
- The settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits are invalid.
- An encoder clear interrupt request signal (INTENCAnlEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCAnSCE = 0, see the timing chart in **Section 40.4.8.19, Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0**.

40.4.3.2 Clearing Method when ENCAnSCE = 1

- When the clear levels of the ENCAnEC, ENCAnE1, and ENCAnE0 inputs are detected, the timer counter is cleared to 0000_H in synchronization with the operating clock.
- Specify the clear levels of the ENCAnEC, ENCAnE1, and ENCAnE0 inputs by the settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits.
- The settings of the ENCAnECS[1:0] bits are invalid.
- An encoder clear interrupt request signal (INTENCAnlEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCAnZCL, ENCAnBCL, and ENCAnACL settings are listed in the table below.

Table 40.27 Clearing Conditions of Timer Counter

Counter Clear Condition Setting			Encoder Input Level		
ENCAnZCL	**ENCAnBCL**	**ENCAnACL**	**ENCAnEC**	**ENCAnE1**	**ENCAnE0**
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

40.4.4 Functions of ENCAnCCR0

40.4.4.1 Compare Function

- When ENCAnCRM0 = 0, the ENCAnCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR0 setting value, a compare 0 match interrupt (INTENCAnl0) is output.
- When ENCAnECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 40.28 Compare Function of ENCAnCCR0

ENCAnCCR0 function	Compare match clear control	Next count operation	Timer counter clearing upon compare match with ENCAnCCR0
ENCAnCRM0	ENCAnECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000_H.
		Down-count	Does not clear (continues count operation).

When ENCAnLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (INTENCAnlUD) is output.

NOTE

For the timing chart when ENCAnLDE = 1, see the timing charts in **Section 40.4.8.8, Using the ENCAnLDE Function Immediately after Startup** and **Section 40.4.8.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

40.4.4.2 Capture Function

- When ENCAnCRM0 = 1, the ENCAnCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCAnTTIN0), the value of the timer counter is stored into ENCAnCCR0.
- A capture 0 interrupt (INTENCAnl0) is output during capture operation.

NOTE

For details about capture operation to ENCAnCCR0, see the timing charts in **Section 40.4.8.14, Capture Operation between Count Clocks (ENCAnCCR0)** and **Section 40.4.8.17, Encoder operation when compare match clear control is disabled**.

40.4.5 Functions of ENCAnCCR1

40.4.5.1 Compare Function

- When ENCAnCRM1 = 0, the ENCAnCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR1 setting value, a compare 1 match interrupt (INTENCAnl1) is output.
- When ENCAnECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 40.29 Compare Function of ENCAnCCR1

ENCAnCCR1 function	Compare match clear control	Next count operation	Timer counter clearing upon compare match with ENCAnCCR1.
ENCAnCRM1	ENCAnECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000_H.

Compare match interrupt detection mask function

- When ENCAnCME = 1, the compare 1 match interrupt detection mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAnCME = 1 and ENCAnMCS = 0, the compare 1 match interrupt detection mask function is disabled by a write operation to the ENCAnCCR1 register.
- When ENCAnCME = 1 and ENCAnMCS = 1, the compare 1 match interrupt detection mask function is disabled by a timer counter clear operation accompanying phase Z or by a timer counter clear operation upon match between the ENCAnCCR0 register value and the timer counter value.
- When ENCAnCME = 1, ENCAnMCS = 1 and ENCAnLDE = 1, the compare 1 match interrupt detection mask function is disabled by a loading operation of the ENCAnCCR0 register to the timer counter upon underflow detection.
- Setting ENCAnECM1 to 1 is prohibited when enabling the compare 1 match interrupt detection mask function.

Table 40.30 Compare Match Interrupt Detection Mask Function of ENCAnCCR1

ENCAnCCR1 function	Compare 1 match interrupt mask	Interrupt mask cancel trigger	Compare 1 match interrupt output upon compare match with ENCAnCCR1
ENCAnCRM1	ENCAnCME	ENCAnMCS	
0 (Compare)	0 (Mask function disabled)	— (Setting invalid)	Outputs compare 1 match interrupt upon each compare match.
	1 (Mask function enabled)	0 (Write operation to ENCAnCCR1)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)
		1 (Timer counter clear operation) (Loading of ENCAnCCR0 to timer counter upon underflow occurrence when ENCAnLDE = 1)	

40.4.5.2 Capture Function

When ENCAnCRM1 = 1, the ENCAnCCR1 register functions as a dedicated capture register.

NOTE

For details about capture operation to ENCAnCCR1, see the timing chart in **Section 40.4.8.13, Capture Operation between Count Clocks (ENCAnCCR1)**.

The operations for each of the ENCAnCTS settings are shown in the table below.

Table 40.31 Operations for each of the ENCAnCTS settings

ENCAnCCR1 function	Capture trigger selection			
ENCAnCRM1	ENCAnCTS	Capture trigger signal	Timer counter clearing	Interrupt occurrence
1 (Capture)	0	Capture trigger 1 input (ENCAnTTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (INTENCAnI1)
	1	Encoder clear input (specified by ENCAnSCE)	Clears timer counter.	(1) Capture 1 interrupt (INTENCAnI1) (2) Encoder clear interrupt (INTENCAnIEC)

NOTE

For details about the timing chart when ENCAnCTS = 0 or ENCAnCTS = 1, see the following:

Section 40.4.8.3, Count Clear and Capture Operation by Encoder Clear Input (ENCAnEC pin), **Section 40.4.8.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 40.4.8.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 40.4.8.11, Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input** and **Section 40.4.8.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

40.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the value of the timer counter and the ENCAnCCR0/ENCAnCCR1 setting value, according to the settings of the ENCAnECM1 and ENCAnECM0 bits in the ENCAnCTL register, is detailed in the following table.

Table 40.32 Timer Counter Clearing Operation upon Compare Register Match

ENCAnECM1 and ENCAnECM0	Next count operation	Timer counter clearing upon compare match with ENCAnCCR1	Timer counter clearing upon compare match with ENCAnCCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).

40.4.6 Startup/Stop of Timer Counter

40.4.6.1 Startup of Timer

The timer operation can be started by setting the ENCAnTS bit to 1. Simultaneous start with other timer can be possible by setting the PIC. For details, see **Section 41.2.3.1, Simultaneous Start Trigger Function**

40.4.6.2 Stop of Timer

When the ENCAnTT bit is set to 1, the ENCAnTE bit becomes 0 and the timer stops.

40.4.7 ENCA Setting Sequences

40.4.7.1 Encoder timer setting procedure

The encoder timer setting procedure is described below.

Table 40.33 Encoder Timer Setting Procedure

	Action	Setting status
Initial Setting	Reset deassertion	Power-on status, operation stopped status (Writing to each register is enabled)
ENCAn initial Settings	Perform the following initial settings. • Setting for counter • Setting for counter clear • Setting for ENCAnCCR0 register • Setting for ENCAnCCR1 register	This is the count operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
	Perform the counter initial value settings. • Set any 16-bit value to ENCAnCNT register. (When, after setting this register, the ENCAnTS bit is set to 1, the counter operation starts from the set count value.)	The set value is set as the initial value of the counter register value.
Operation Start	Perform the counter operation start setting. • Set the ENCAnTS bit to 1.	This is the counter operation start status. The value of the ENCAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. • ENCAnCCR0 register setting • ENCAnCCR1 register setting • ENCAnIOC0 register setting	The count operation set with the initial setting is performed, and up/down counting is performed according to the ENCAnE0 and ENCAnE1 pins.
Operation Stop	Perform the counter operation stop setting during operation. • Set the ENCAnTT bit to 1.	This is the counter operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
ENCA stop	Reset	The setting registers are initialized.

(1) Initial Setting Procedure for Counter

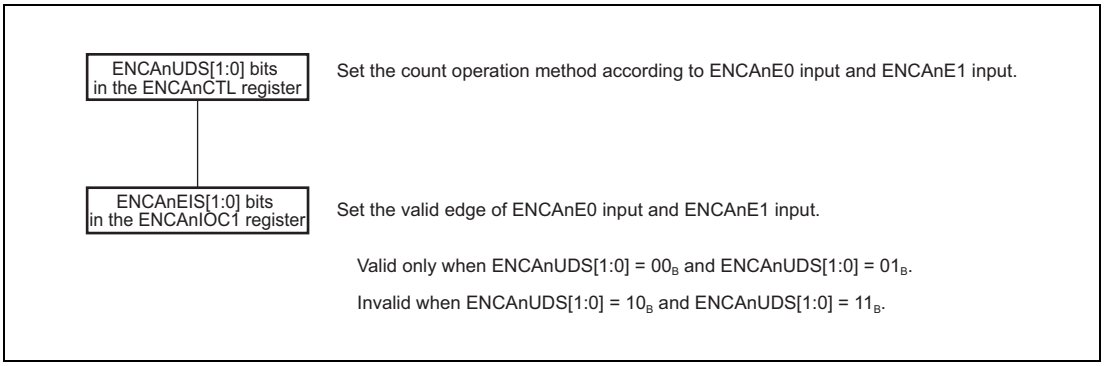


Figure 40.7 Initial Setting Procedure for Counter

(2) Initial Setting Procedure for Counter Clear

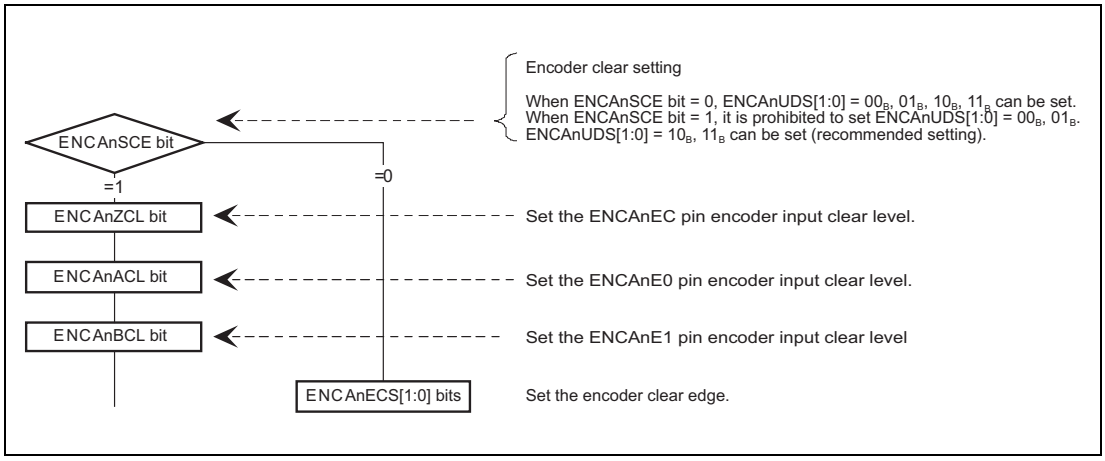


Figure 40.8 Initial Setting Procedure for Counter Clear

(3) Setting Procedure for ENCAnCCR0 Register

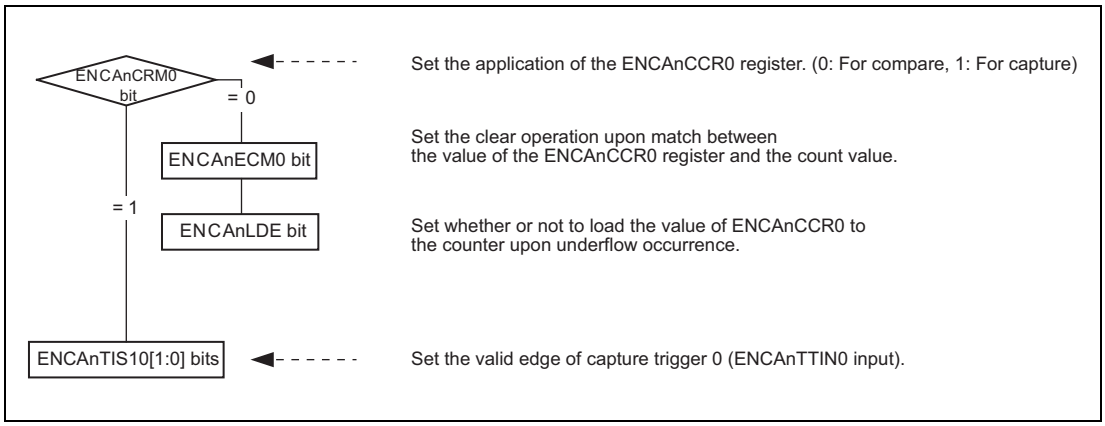


Figure 40.9 Setting Procedure for ENCAnCCR0 register

(4) Setting Procedure for ENCAnCCR1 Register

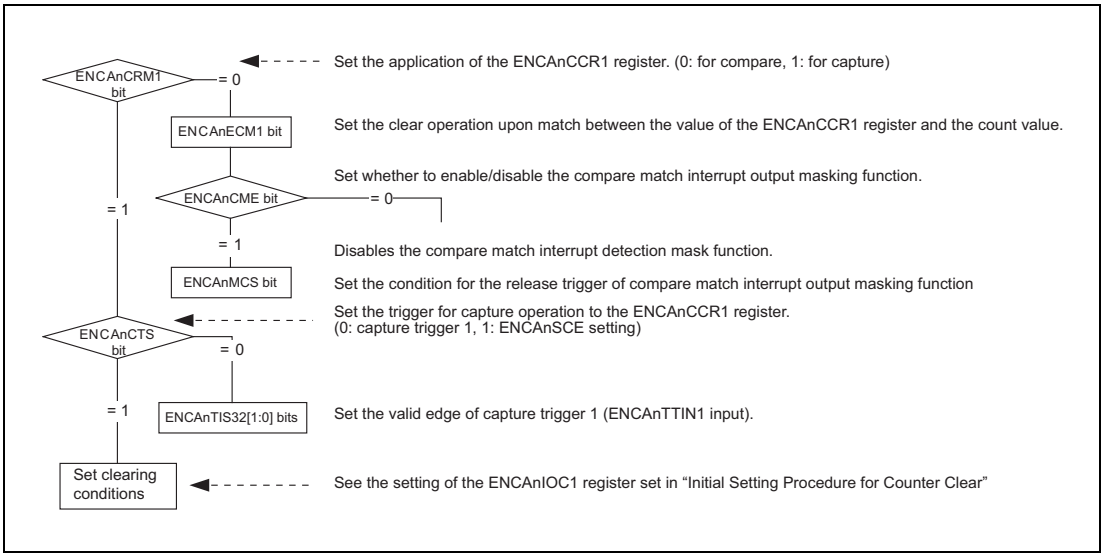


Figure 40.10 Setting Procedure for ENCAnCCR1 register

40.4.8 Timing Chart

40.4.8.1 Overflow Occurrence and Overflow Flag Clear Operation

When an up-counting is performed while the counter value is $FFFF_H$, an overflow occurs. If an overflow occurs, an overflow interrupt (INTENCAnlOV) is output and the overflow flag (ENCAnOVF) is set to 1. The overflow flag clear bit (ENCAnCLOV) is set to 1, the overflow flag (ENCAnOVF) is cleared to 0.

The overflow occurrence and overflow flag clear operation are described as follows.

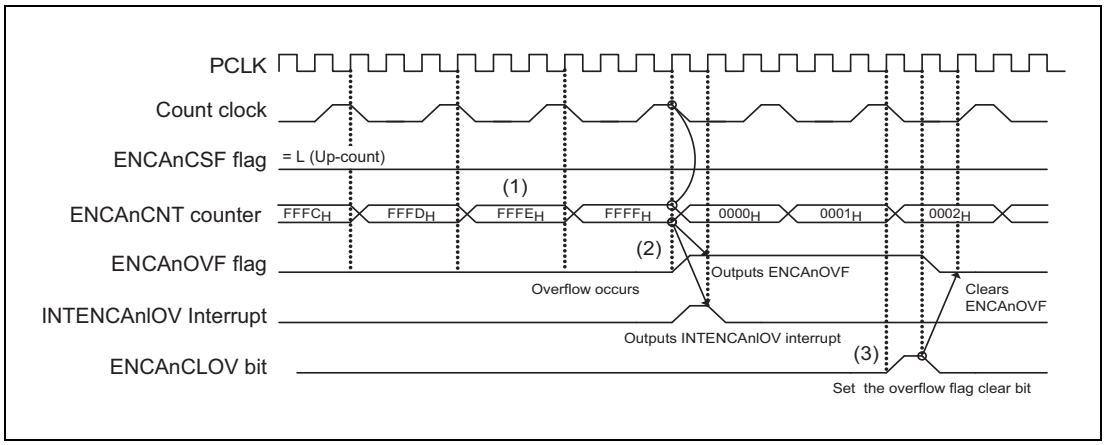


Figure 40.11 Overflow Occurrence and Overflow Flag Clear Setting

(1) The count value is up-counted from $FFFE_H$ to $FFFF_H$.

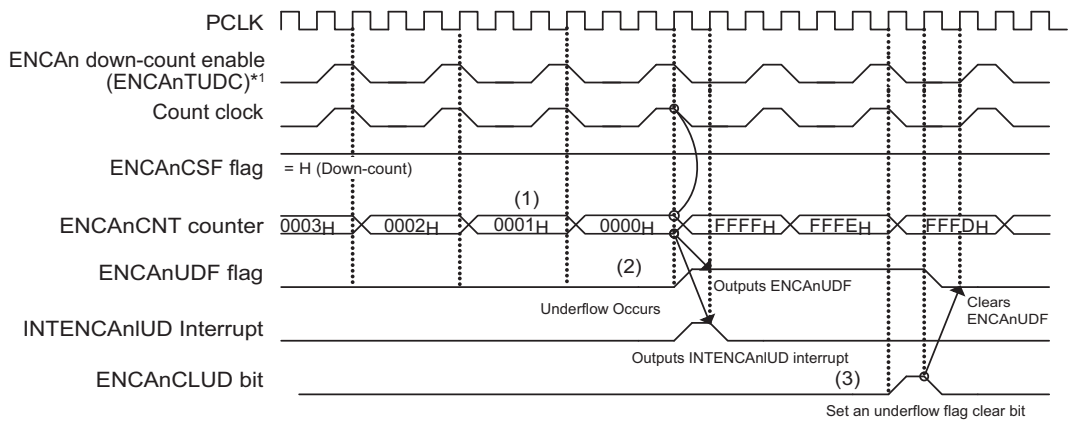
(2) When the count value changes from $FFFF_H$ to 0000_H, an overflow occurs. At the same time, an overflow interrupt is output, and the overflow flag is set to 1.

(3) An overflow flag is cleared to 0 by setting 1 to the ENCAnCLOV bit in the ENCAnFGC register according to the clearing procedure. In addition, an overflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 while ENCAnTE.ENCAnTE is 0, or by setting an input signal of the ENCAnTSST (simultaneous start trigger input) to high.

40.4.8.2 Underflow Occurrence and Underflow Flag Clear Operation

When down-counting is performed while the counter value is 0000_H, an underflow occurs. If an underflow occurs, an underflow interrupt (INTENCAnlUD) is output and the underflow flag (ENCAnUDF) is set to 1. The underflow flag clear bit (ENCAnCLUD) is set to 1, the underflow flag (ENCAnUDF) is cleared to 0.

The underflow occurrence and underflow flag clear operation are described as follows.



Note 1. During high-level output of the count clock signal (ENCATCKEN), Down-count enable signal (ENCAnTUDC) output high level to indicate the down status.

Figure 40.12 Underflow Occurrence and Underflow Flag Clear Setting

(1) The count value is down-counted from 0001_H to 0000_H.

(2) When the count value changes from 0000_H to $FFFF_H$, an underflow occurs. At the same time, an underflow interrupt is output, and the underflow flag is set to 1.

(3) An underflow flag is cleared to 0 by setting 1 to the ENCAnCLUD bit in the ENCAnFGC register according to the clearing procedure. In addition, an underflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS to 1 while ENCAnTE.ENCAnTE is 0, or by setting an input signal of the ENCAnTSST (simultaneous start trigger input) to high.

40.4.8.3 Count Clear and Capture Operation by Encoder Clear Input (ENCAnEC pin)

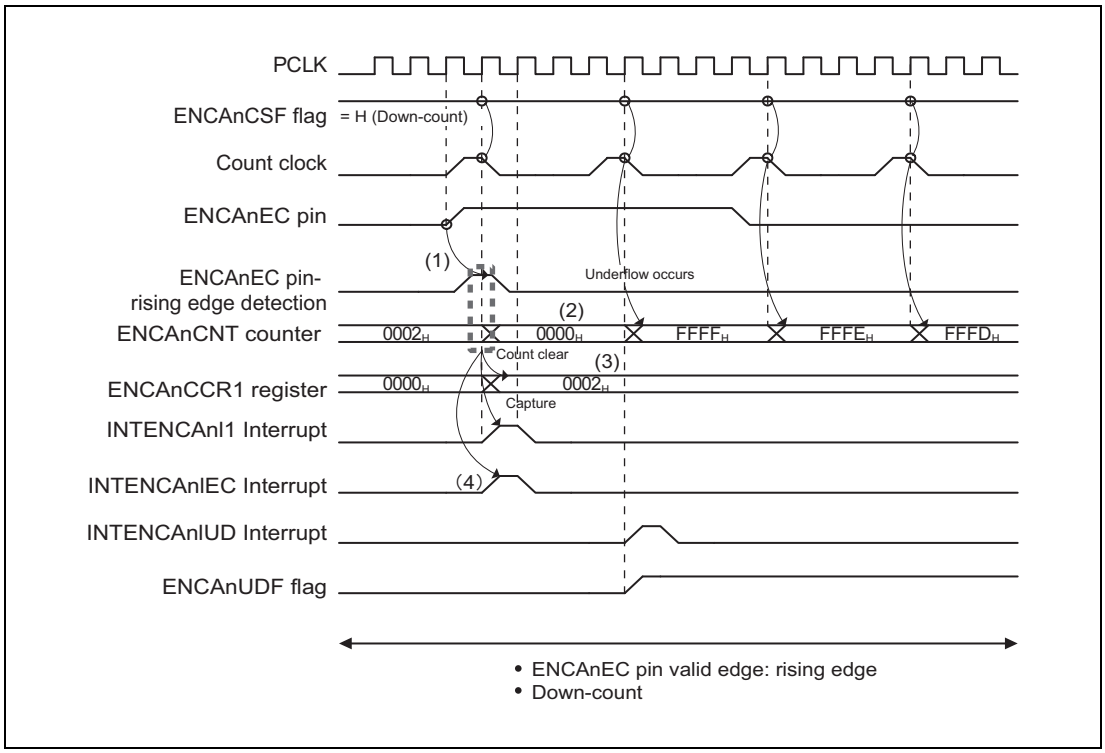


Figure 40.13 Timing Diagram of Count Clear and Capture Operation by Encoder Clear Input (ENCAnEC pin)

[Setting Conditions]

- ENCAnCTL.ENCAnCRM1 = 1
  (ENCAnCCR1 register is selected as capture)
- ENCAnCTL.ENCAnCTS = 1
  (ENCAnEC pin input is selected as a capture trigger input)
- ENCAnIOC1.ENCAnECS[1:0] bits = 01_B
  (Selected as a rising edge detection of the ENCAnEC pin input)

(1) Capture operation is performed by the rising edge of the ENCAnEC pin trigger.

(2) Clear operation is performed by an input to the ENCAnEC pin and the count value is reset to 0000_H.

(3) The counter value (0002_H) is captured to the ENCAnCCR1 at the rising edge of the ENCAnEC pin.

(4) At the same time, by the ENCAnEC pin input, a clear interrupt (INTENCAnlEC) and capture interrupt (INTENCAnl1) are output.

40.4.8.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

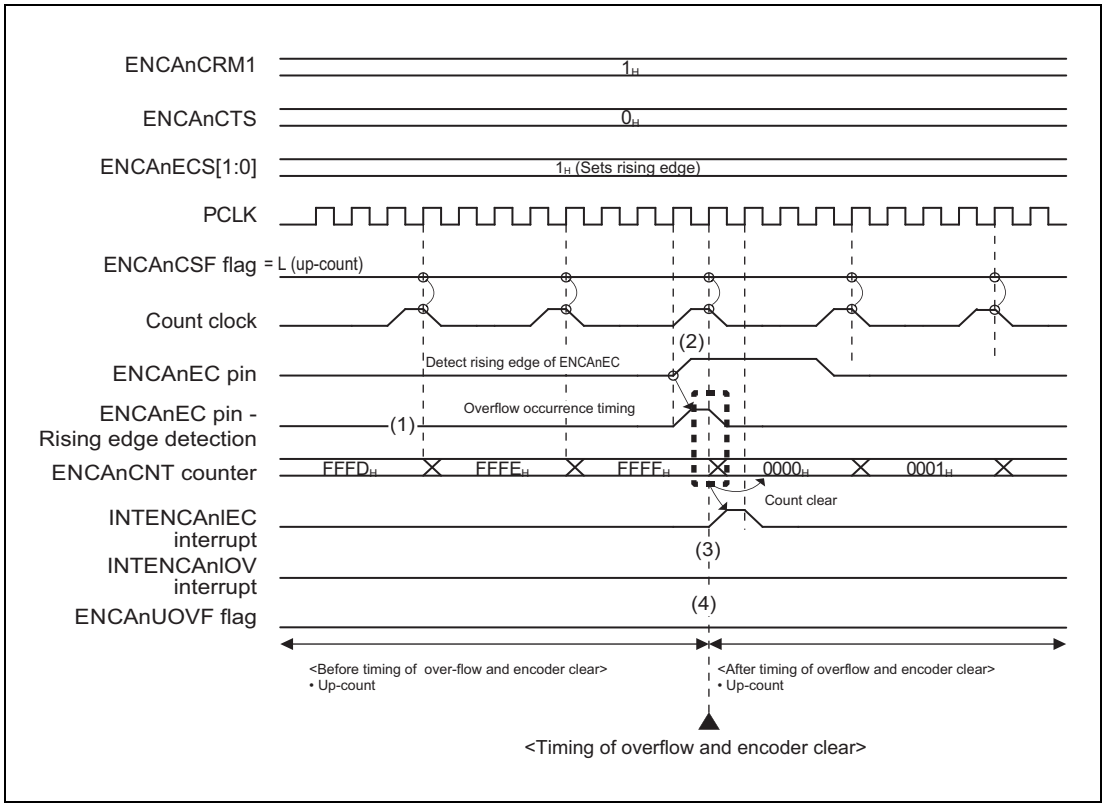


Figure 40.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

(1) An up-count from $FFFD_H$ is continuously performed.

(2) When an overflow occurs if the count value is $FFFF_H$, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The counter value is cleared to 0000_H.

(3) When the counter value is cleared by the encoder clear input, a clear interrupt (INTENCAnIEC) by encoder clear input is output simultaneously. Because a clear operation by the phase Z input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the phase Z input).

(4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

40.4.8.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

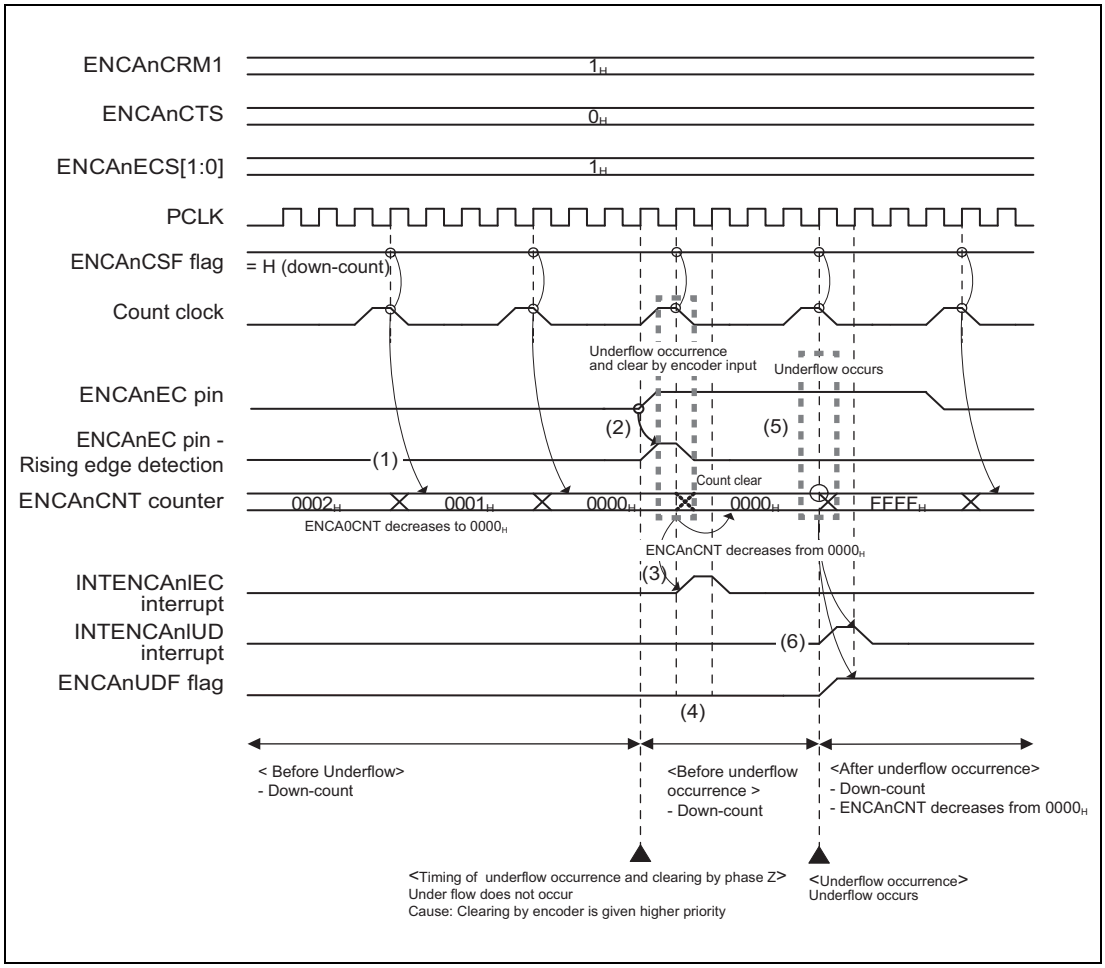


Figure 40.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

(1) A down-count from 0002_H is continuously performed.

(2) When an underflow occurs if the count value is 0000_H, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.

(3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (INTENCAnIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the encoder clear input).

(4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.

(5) When a further down-count is performed after the counter value changes to 0000_H by clear operation by the encoder clear input, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs.

(6) When an underflow occurs, an underflow interrupt ((INTENCAnIUD) is output, and the underflow flag (ENCAnUDF) is set.

40.4.8.6 Overflow Operation Immediately after Startup

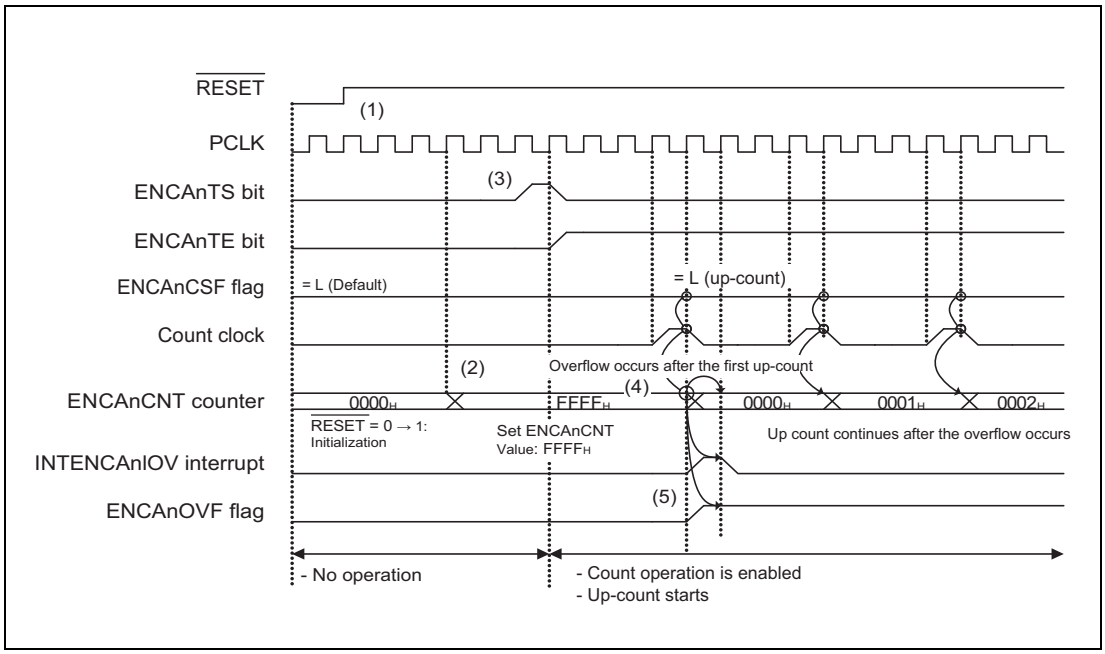


Figure 40.16 Overflow Operation Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.

(2) The timer counter is set to $FFFF_H$ as the initial value.

(3) ENCAnTS is set to 1, and operation starts. ENCAnTE changes to 1, which indicates that operation is enabled.

(4) When an up-count is performed from $FFFF_H$ which is the initially set count value, the counter value changes from $FFFF_H$ to 0000_H, and an overflow occurs immediately after operation starts.

(5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (INTENCAnlOV) is output, and the overflow flag (ENCAnOVF) is set.

40.4.8.7 Underflow Operation Immediately after Startup

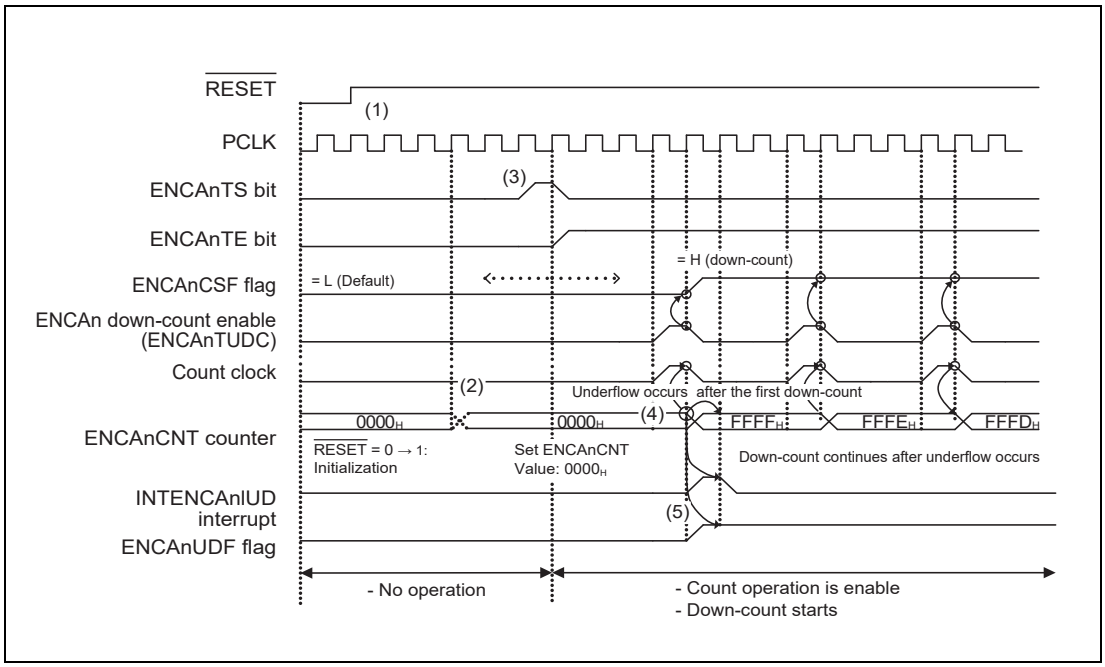


Figure 40.17 Underflow Operation Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.

(2) The timer counter is set to 0000_H as the initial value.

(3) ENCAnTS is set to 1, and operation starts. ENCAnTE changes to 1, which indicates that operation is enabled.

(4) When a down-count is performed from 0000_H which is the initially set count value, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs immediately after operation starts.

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (INTENCAnlUD) is output, and the underflow flag (ENCAnUDF) is set.

40.4.8.8 Using the ENCAnLDE Function Immediately after Startup

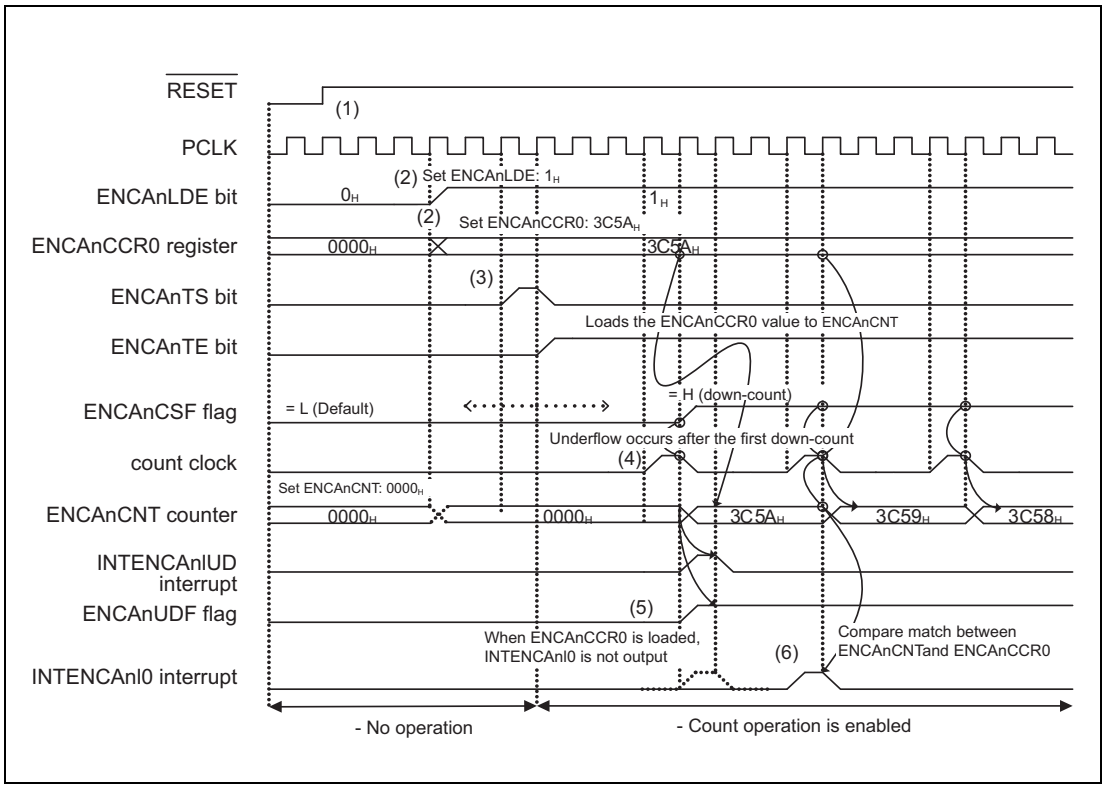


Figure 40.18 Using the ENCAnLDE Function Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from 0 to 1, the status changes from “reset” to “reset release”.

(2) The load enable bit (ENCAnLDE) is set to 1, capture/compare register 0 (ENCAnCCR0) is set to $3C5A_H$, and the timer counter is set to the initial value 0000_H.

(3) ENCAnTS is set to 1, and operation starts. ENCAnTE changes to 1, which indicates that operation is enabled.

(4) When a down-count is performed from 0000_H which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCAnLDE is set to 1, the ENCAnCCR0 value, $3C5A_H$, is loaded to the timer counter (INTENCAnl0 is not output during loading).

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (INTENCAnlUD) is output, and the underflow flag (ENCAnUDF) is set (after an underflow occurs, down-count operation from the loaded value ($3C5A_H$) continues).

(6) After the ENCAnCCR0 value is loaded to ENCAnCNT, a match with ENCAnCCR0 is detected, and INTENCAnl0 is output.

40.4.8.9 ENCAnLDE Function (Loading Count Value)

(1) <When ENCAnLDE = 0>

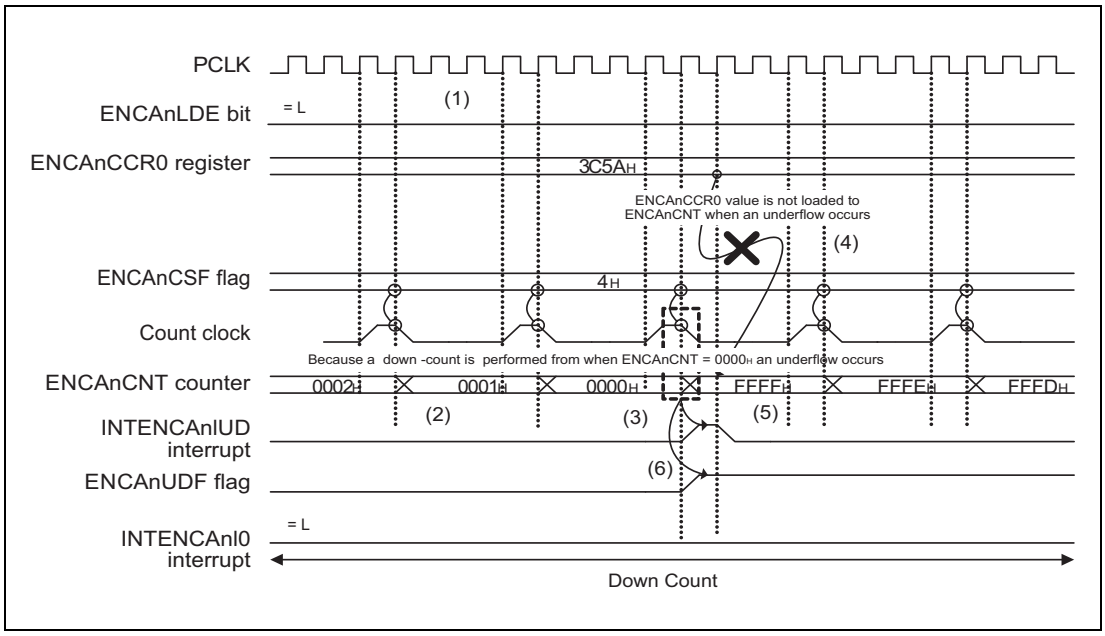


Figure 40.19 ENCAnLDE Function (when ENCAnLDE = 0)

(1) ENCAnLDE is set to 0 (even if an underflow occurs, the ENCAnCCR0 value is not loaded).

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) Because ENCAnLDE is set to 0, the setting value of the ENCAnCCR0 register is not loaded to the counter even if an underflow occurs.

(5) Operation changes to underflow operation (counter value: $0000_H \rightarrow FFFF_H$).

(6) An underflow interrupt (INTENCAnlUD) is output, and the underflow flag (ENCAnUDF) is set.

(2) <When ENCAnLDE = 1>

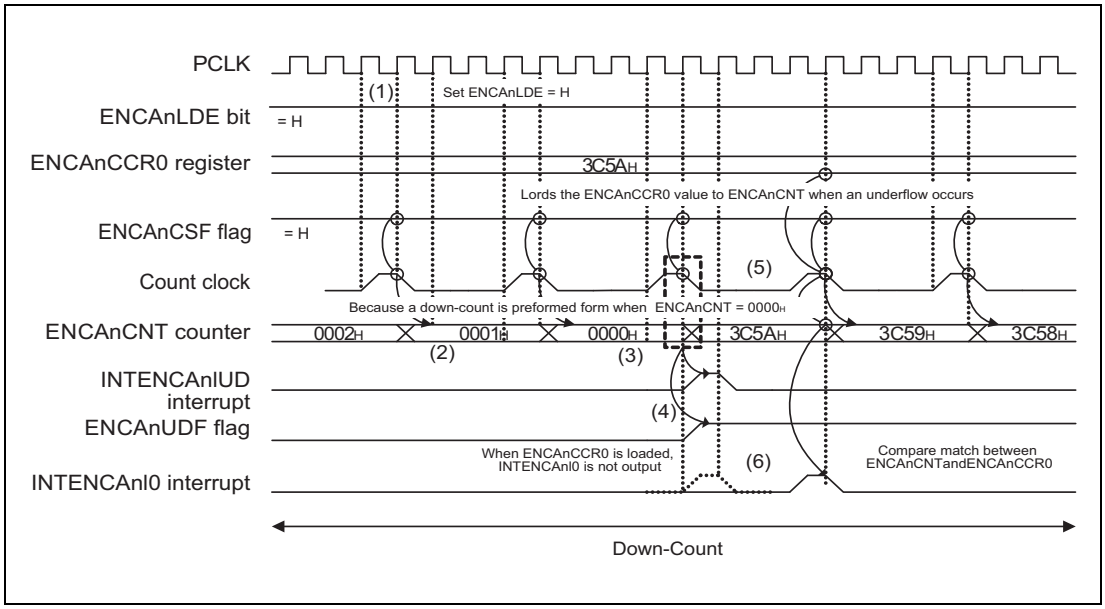


Figure 40.20 ENCAnLDE Function (when ENCAnLDE = 1)

(1) ENCAnLDE is set to 1 (if an underflow occurs, the ENCAnCCR0 value is loaded to the counter).

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) An underflow interrupt is output, and the underflow flag is set.

(5) Because ENCAnLDE is set to 1, the setting value of the ENCAnCCR0 register is loaded to the counter if an underflow occurs. ENCAnCNT is set to $3C5A_H$.

(6) After the ENCAnCCR0 value is set to ENCAnCNT, if the ENCAnCNT value matches with the ENCAnCCR0 value on a count clock, a compare match interrupt (INTENCAnI0) is output.

40.4.8.10 Conflict between ENCAnLDE Function (Loading Counter Value) and Rewrite of ENCAnCCR0 Register

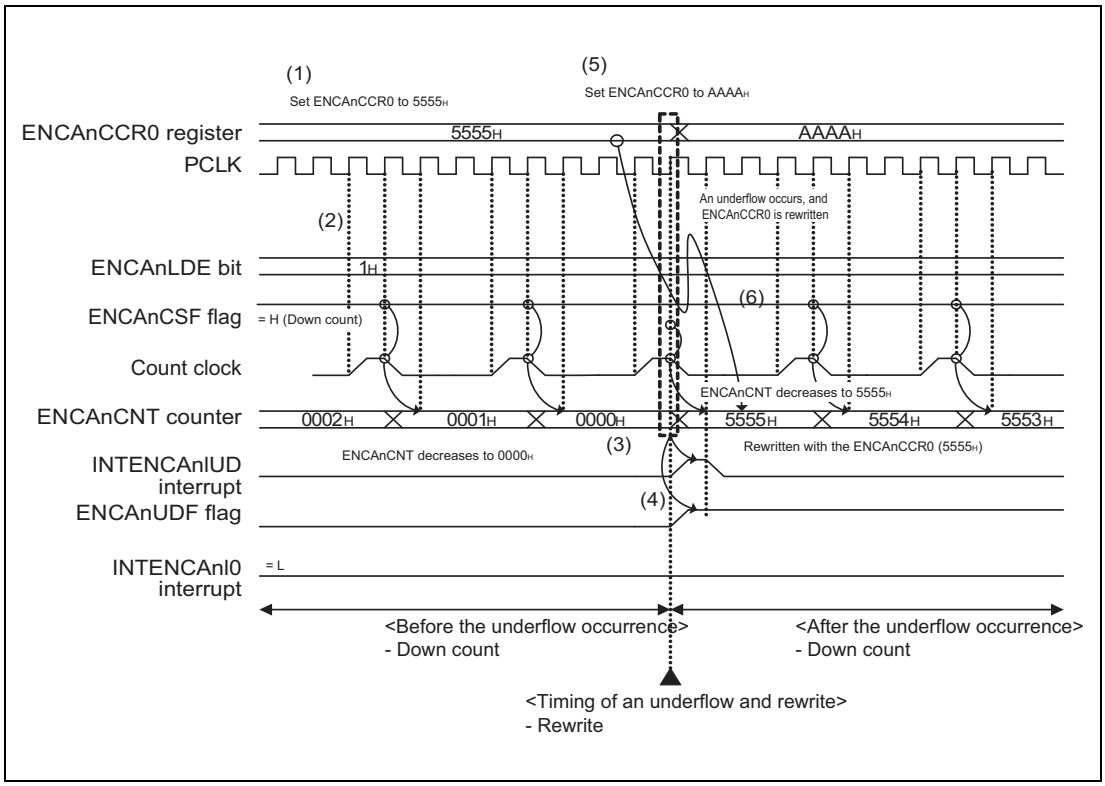


Figure 40.21 Conflict between ENCAnLDE Function and Rewrite of ENCAnCCR0 Register

(1) The ENCAnCCR0 register is currently set to 5555_H.

(2) ENCAnLDE is currently set to 1.

(3) A down-count is performed ($0002_H \rightarrow 0001_H \rightarrow 0000_H$), and an underflow occurs.

(4) An underflow interrupt is output, and the underflow flag is set.

(5) When an underflow occurs, the ENCAnCCR0 register value is changed from 5555_H to $AAAA_H$.

(6) When an underflow occurs, the ENCAnCCR0 value before the rewrite was performed (5555_H) is set in ENCAnCNT.

40.4.8.11 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

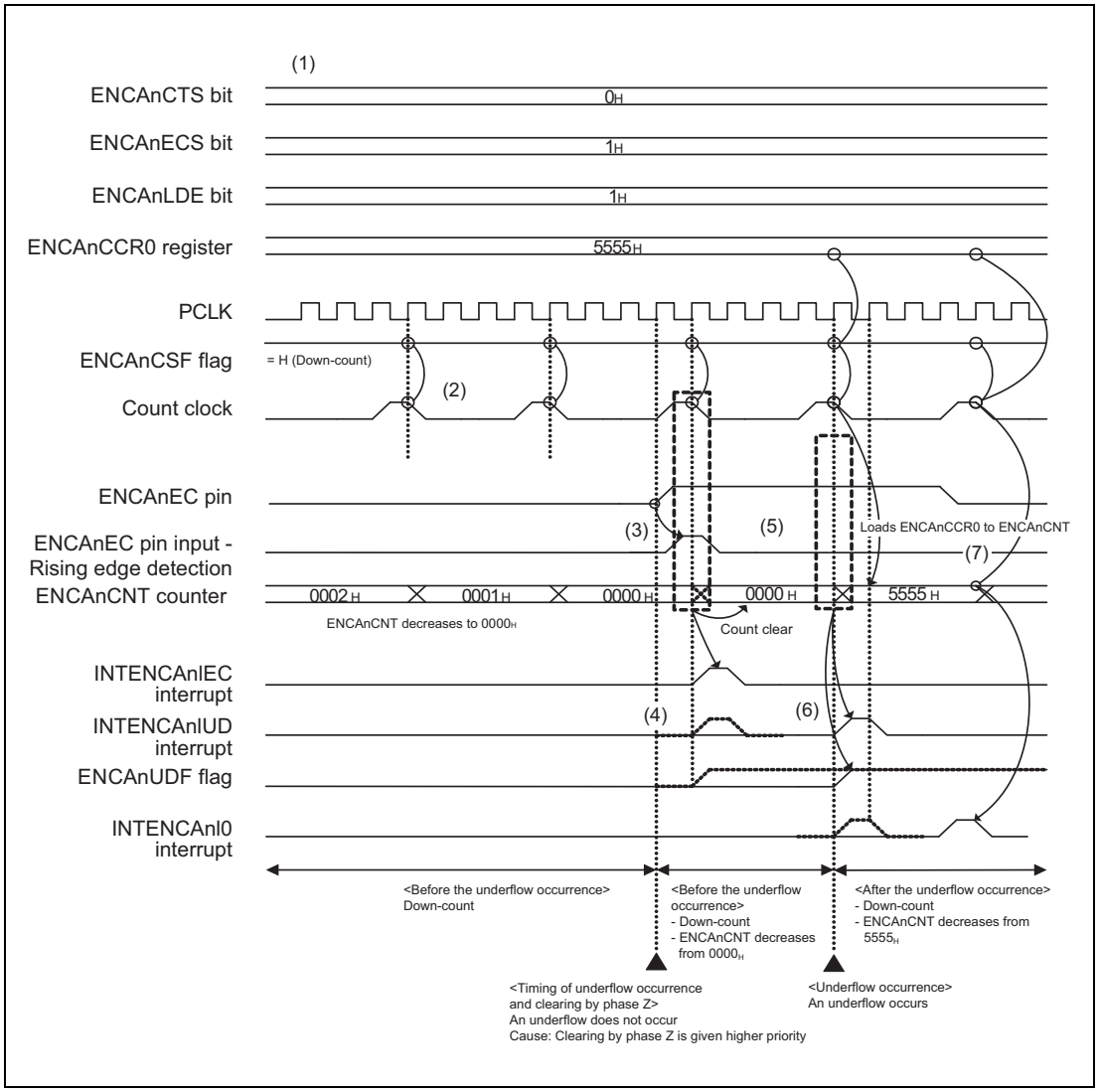


Figure 40.22 Conflict between ENCAnLDE Function and Clear Operation by Encoder Clear Input

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS1 and ENCAnECS0 = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (INTENCAnlEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (INTENCAnlUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, a down-count is performed and an underflow occurs.

(6) An underflow interrupt (INTENCAnlUD) is output, and the underflow flag (ENCAnUDF) is set.

(7) Because ENCAnLDE = 1, if an underflow occurs, the ENCAnCCR0 value is loaded to ENCAnCNT.

(8) After the ENCAnCCR0 value is set to ENCAnCNT, a compare match is detected according to the count clock. If the ENCAnCNT value matches with the ENCAnCCR0 value, a compare match interrupt (INTENCAnl0) is output.

40.4.8.12 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

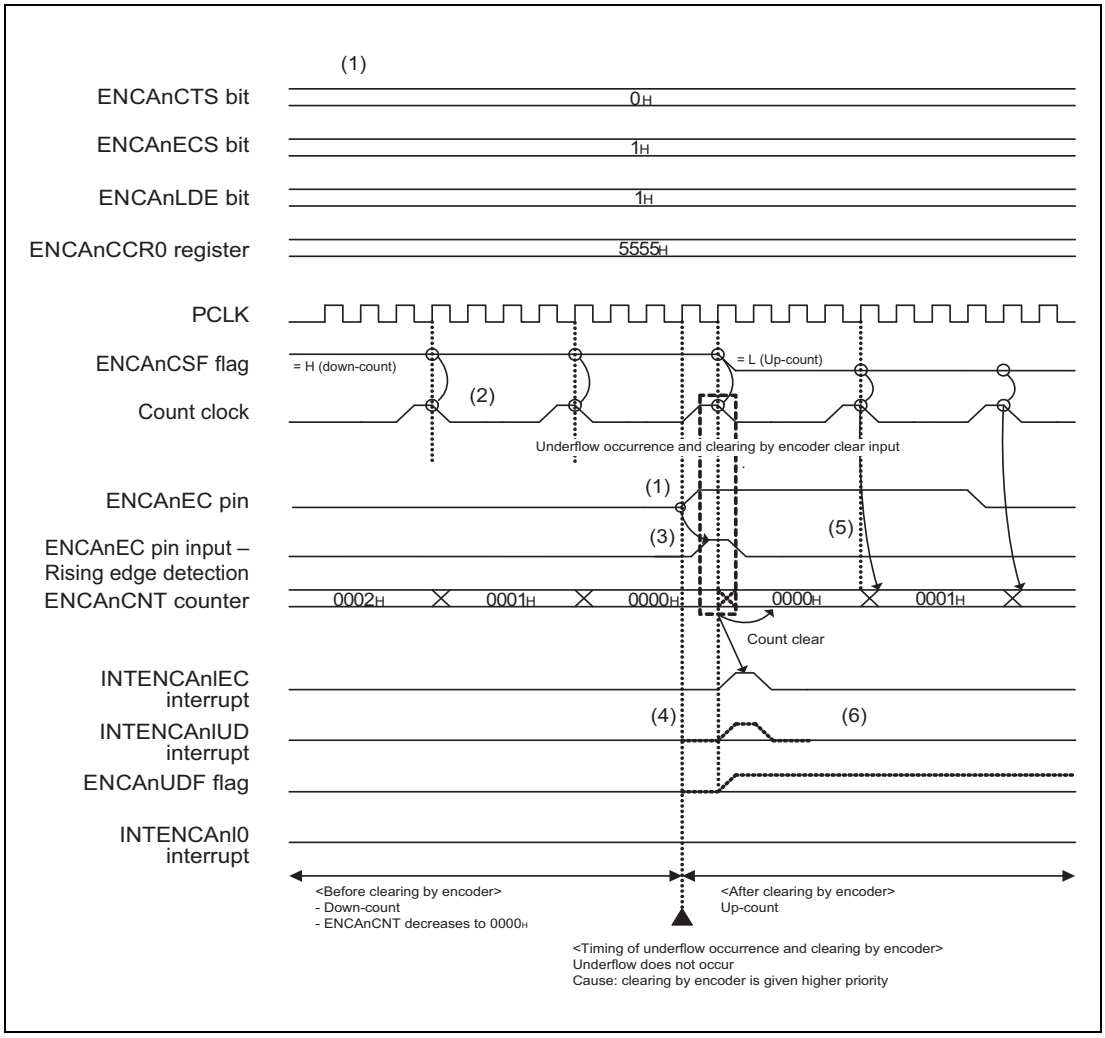


Figure 40.23 Up-count after Conflict between ENCAnLDE Function and Encoder Clear Input

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS1 and ENCAnECS0 = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (INTENCAnIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (INTENCAnIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, an up-count is performed.

(6) An underflow interrupt (INTENCAnIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

40.4.8.13 Capture Operation between Count Clocks (ENCAnCCR1)

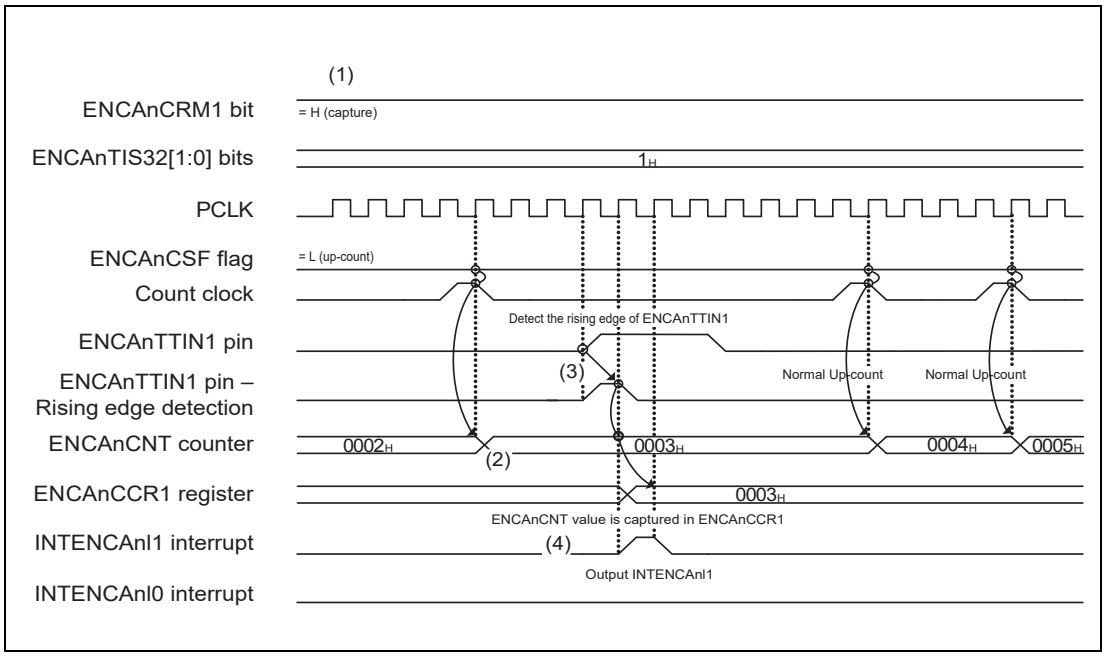


Figure 40.24 Capture Operation between Count Clocks (ENCAnCCR1)

(1) The values are set as follows: ENCAnCRM1 = 1 and ENCAnTIS32[1:0] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCAnTTIN1 input is detected, and the count value is captured in ENCAnCCR1.

(4) An interrupt request signal (INTENCAnI1) corresponding to the capture to the ENCAnCCR1 register is output.

40.4.8.14 Capture Operation between Count Clocks (ENCAnCCR0)

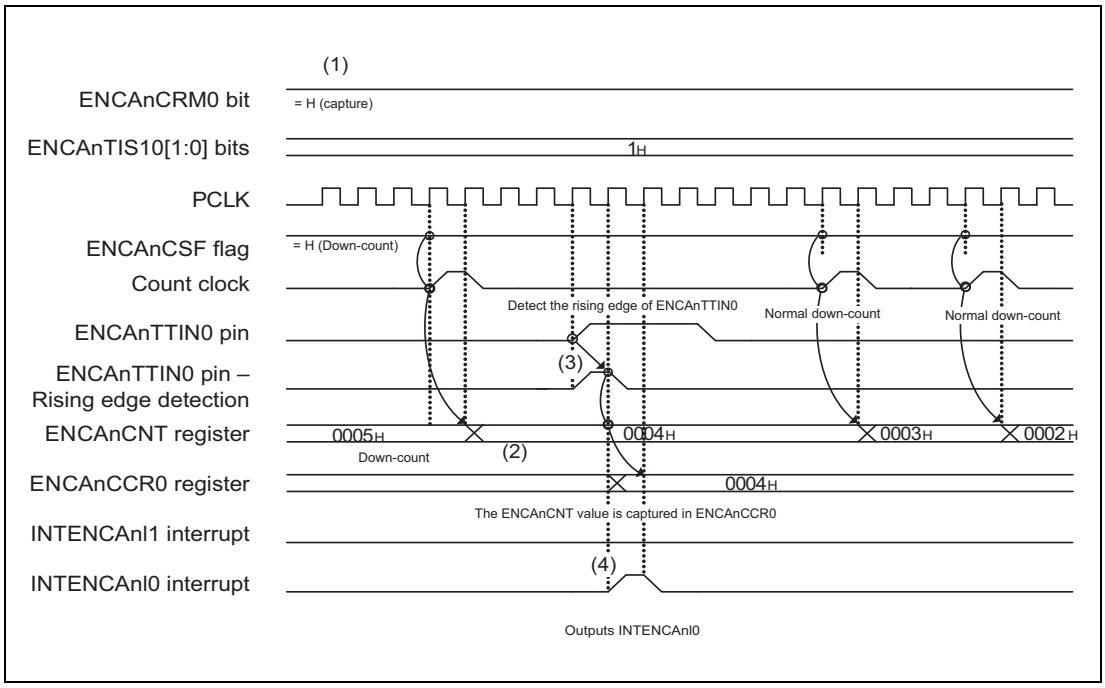


Figure 40.25 Capture Operation between Count Clocks (ENCAnCCR0)

(1) The values are set as follows: ENCAnCRM0 = 1, and ENCAnTIS10[1:0] = 01_B.

(2) A down-count is performed.

(3) The rising edge of the ENCAnTTIN0 input is detected, and the count value is captured in ENCAnCCR0.

(4) An interrupt request signal (INTENCAnI0) corresponding to the capture to the ENCAnCCR0 register is output.

40.4.8.15 Encoder operation when compare match clear control is enabled and ENCAnCTS = 0

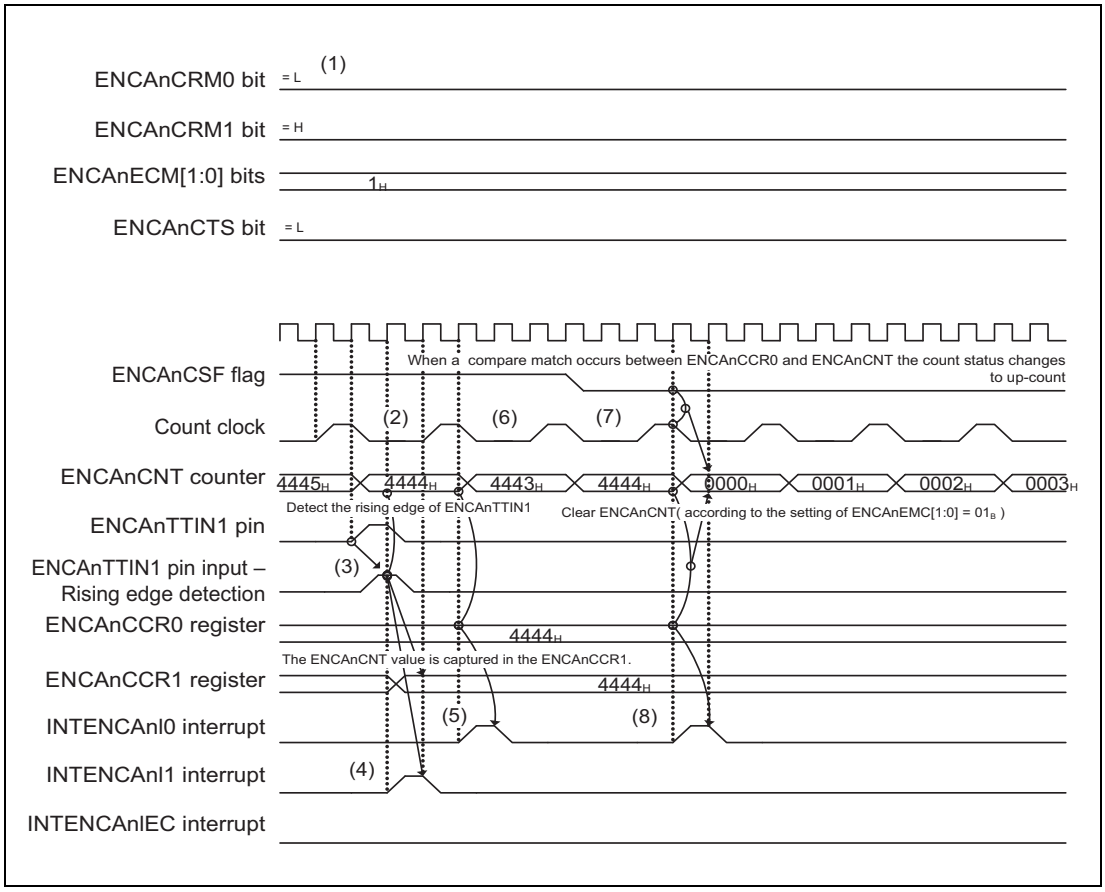


Figure 40.26 Encoder Operation when ENCAnECM[1:0] = 01_B and ENCAnCTS = 0

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) The rising edge of the ENCAnTTIN1 input is detected, and the ENCAnCNT value (4444_H) is captured in the ENCAnCCR1 register.

(4) An interrupt signal (INTENCAnl1) corresponding to the capture to the ENCAnCCR1 register is output.

(5) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare match interrupt (INTENCAnl0) with ENCAnCCR0 is output.

(6) The count operation changes to up-count.

(7) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(8) When ENCAnCNT changes to 4444_H, a compare match interrupt (INTENCAnl0) with ENCAnCCR0 is output.

40.4.8.16 Encoder operation when compare match clear control is enabled and ENCAnCTS = 1

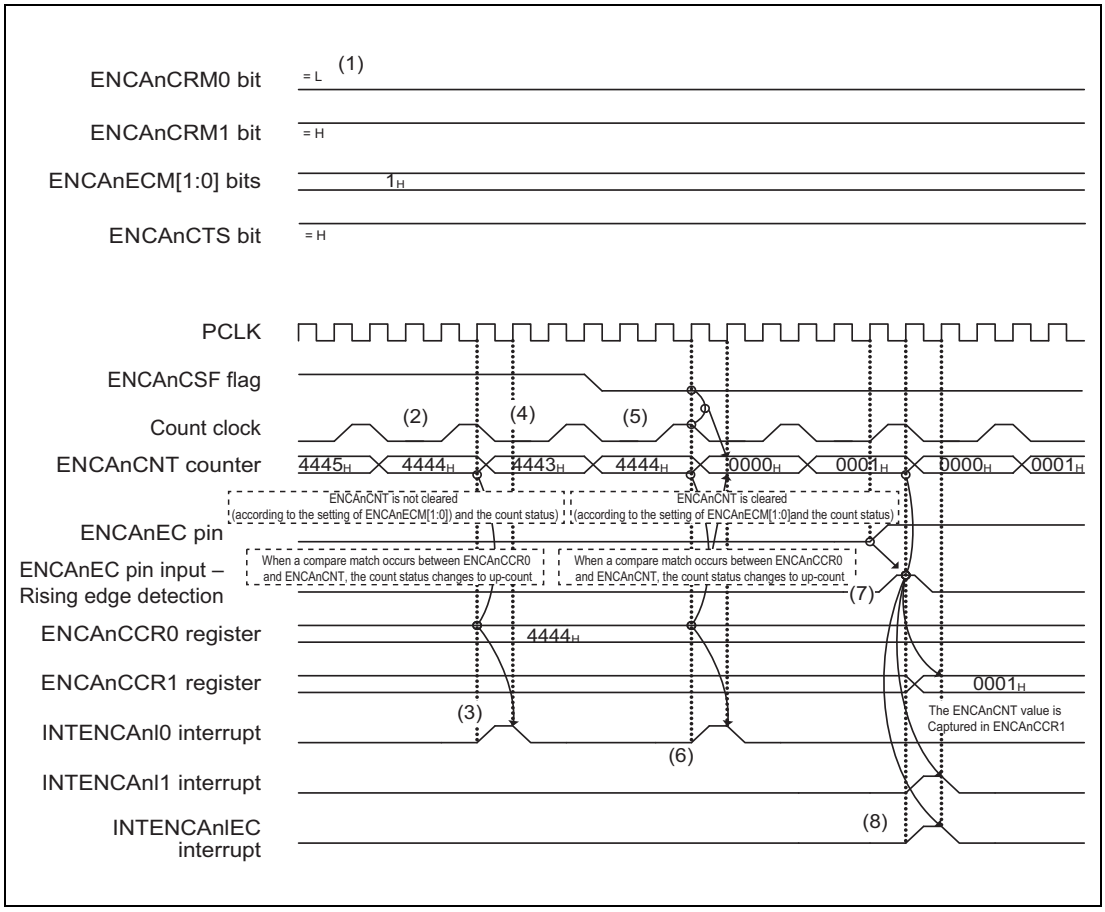


Figure 40.27 Encoder Operation when ENCAnECM[1:0] = 01_B and ENCAnCTS = 1

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 1.

(2) A down-count is performed.

(3) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), an interrupt request signal (INTENCAnl0) is output.

(4) The count operation changes to up-count.

(5) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(6) When ENCAnCNT changes to 4444_H, a compare match interrupt (INTENCAnl0) with ENCAnCCR0 is output.

(7) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCAnCNT value (0001_H) is captured in ENCAnCCR1 by detecting the rising edge of the ENCAnEC signal, and the counter is cleared to 0000_H.

(8) An interrupt (INTENCAnl1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (INTENCAnlEC) by ENCAnEC are output.

40.4.8.17 Encoder operation when compare match clear control is disabled

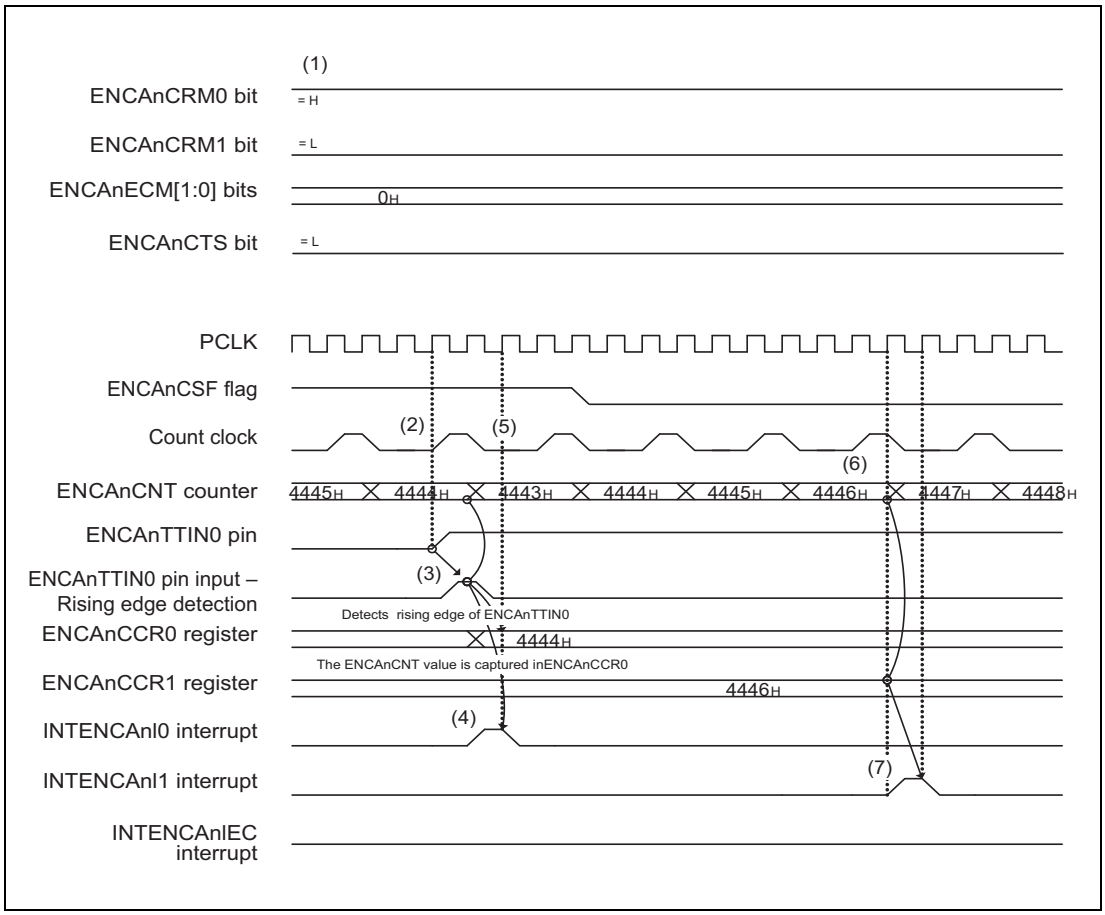


Figure 40.28 Encoder Operation when ENCAnECM[1:0] = 00_B

(1) The values are set as follows: ENCAnCCR1 = 4446_H, ENCAnCRM0 = 1, ENCAnCRM1 = 0, ENCAnECM[1:0] = 00_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) When the rising edge of ENCAnTTIN0 is detected, the ENCAnCNT value (4444_H) is captured in ENCAnCCR0.

(4) An interrupt signal (INTENCAnl0) corresponding to the capture to the ENCAnCCR0 register is output.

(5) The count operation changes to up-count.

(6) When ENCAnCNT changes to 4446_H, a compare match with ENCAnCCR1 is detected.

(7) A compare match interrupt (INTENCAnl1) with ENCAnCCR1 is output.

40.4.8.18 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, and ENCAnE1 when ENCAnSCE = 1

(1) Accompanying Capture Operation

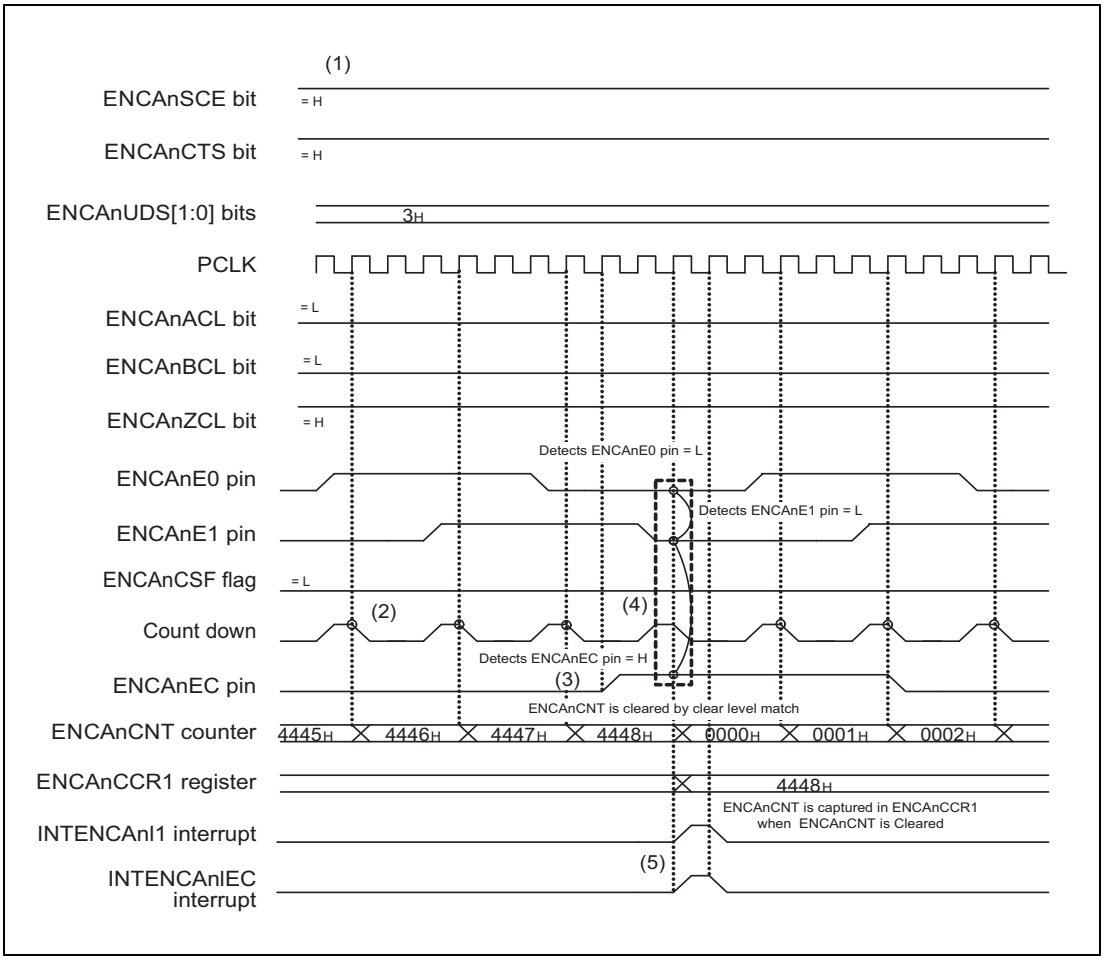


Figure 40.29 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, and ENCAnE1 when ENCAnSCE = 1

(1) The values are set as follows: ENCAnSCE = 1, ENCAnCTS = 1, ENCAnUDS[1:0] = 11_B, ENCAnACL = 0, ENCAnBCL = 0, and ENCAnZCL = 1.

(2) An up-count is performed.

(3) The count value is not cleared upon the rising edge of ENCAnEC.

(4) When ENCAnE0, ENCAnE1, and ENCAnEC reach the set clear level, the count value is cleared. The count value is captured in ENCAnCCR1 at the time of the clearing.

(5) At the time of the clearing, an interrupt (INTENCAnI1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (INTENCAnIEC) by ENCAnEC are output.

(2) When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count

(When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

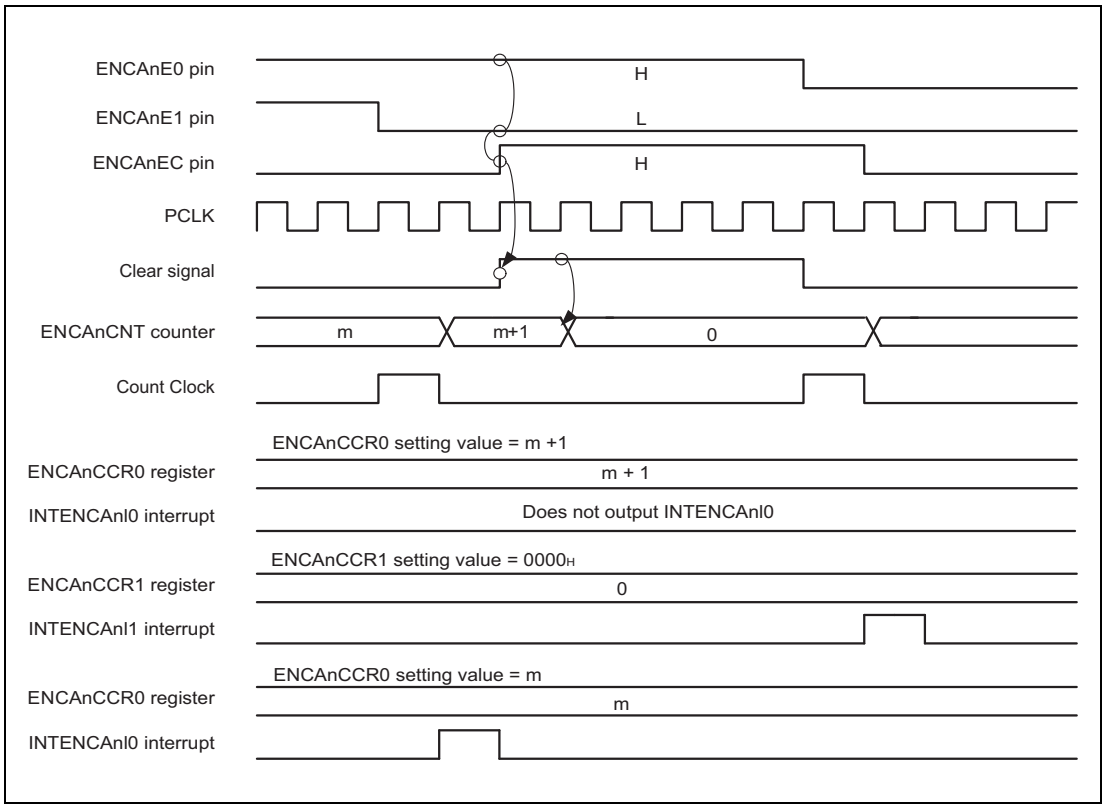


Figure 40.30 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count

(3) When the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count

(When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

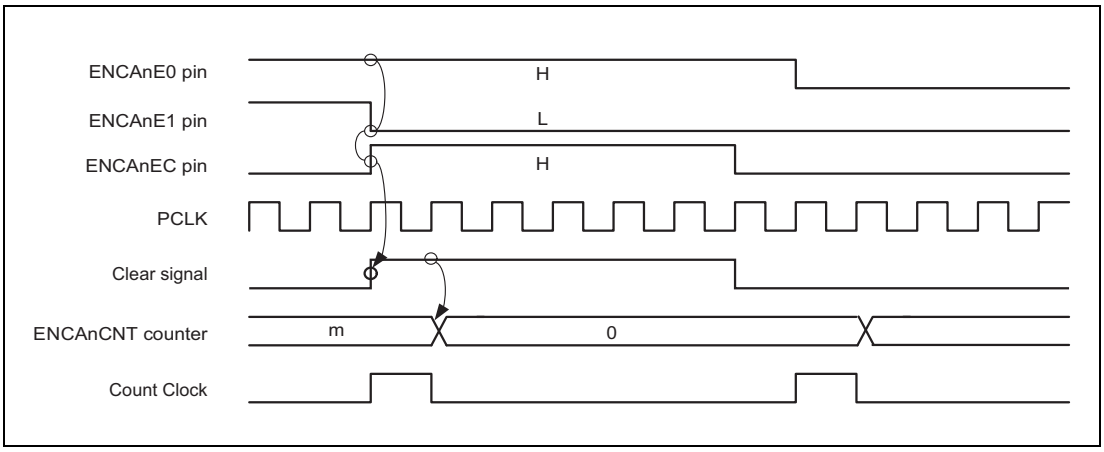


Figure 40.31 Clearing Timing for when the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input During Up-count

(4) When the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count

(When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

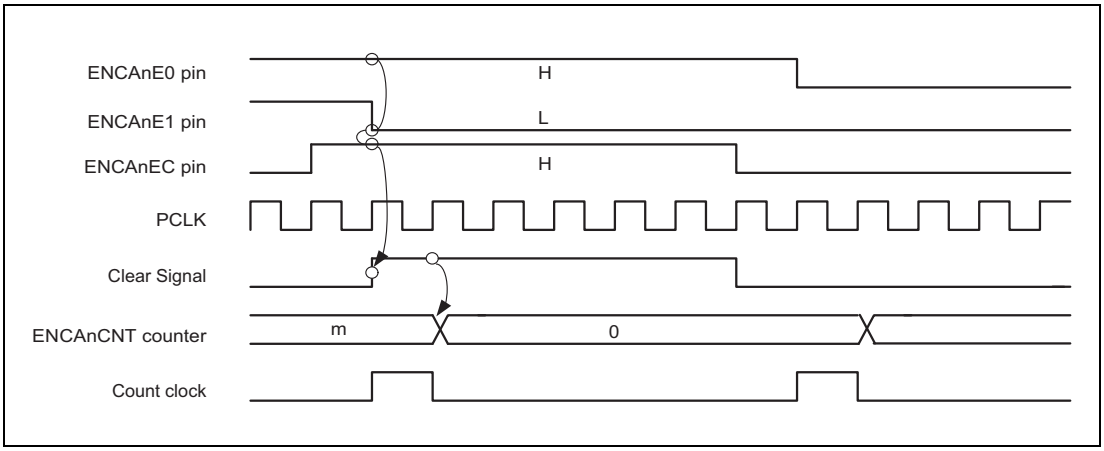


Figure 40.32 Clearing Timing for when the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count

(5) When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count

(When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, ENCAnUDS[1:0] = 11_B)

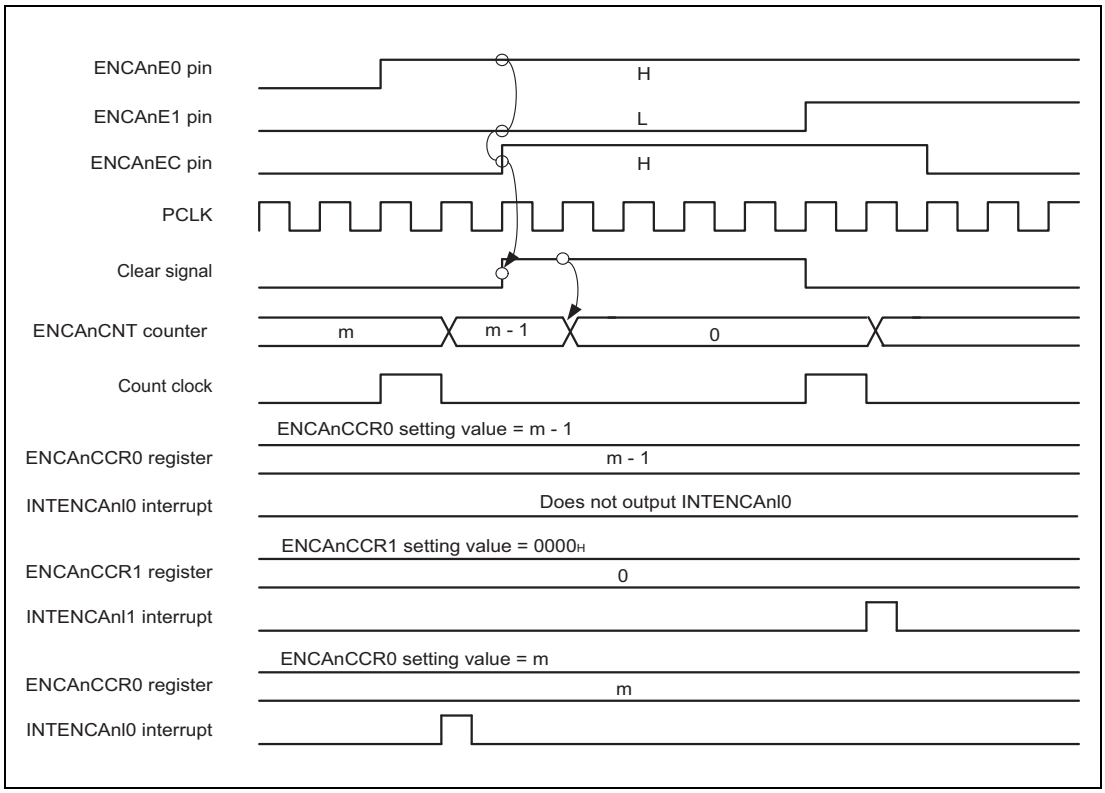


Figure 40.33 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count

40.4.8.19 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

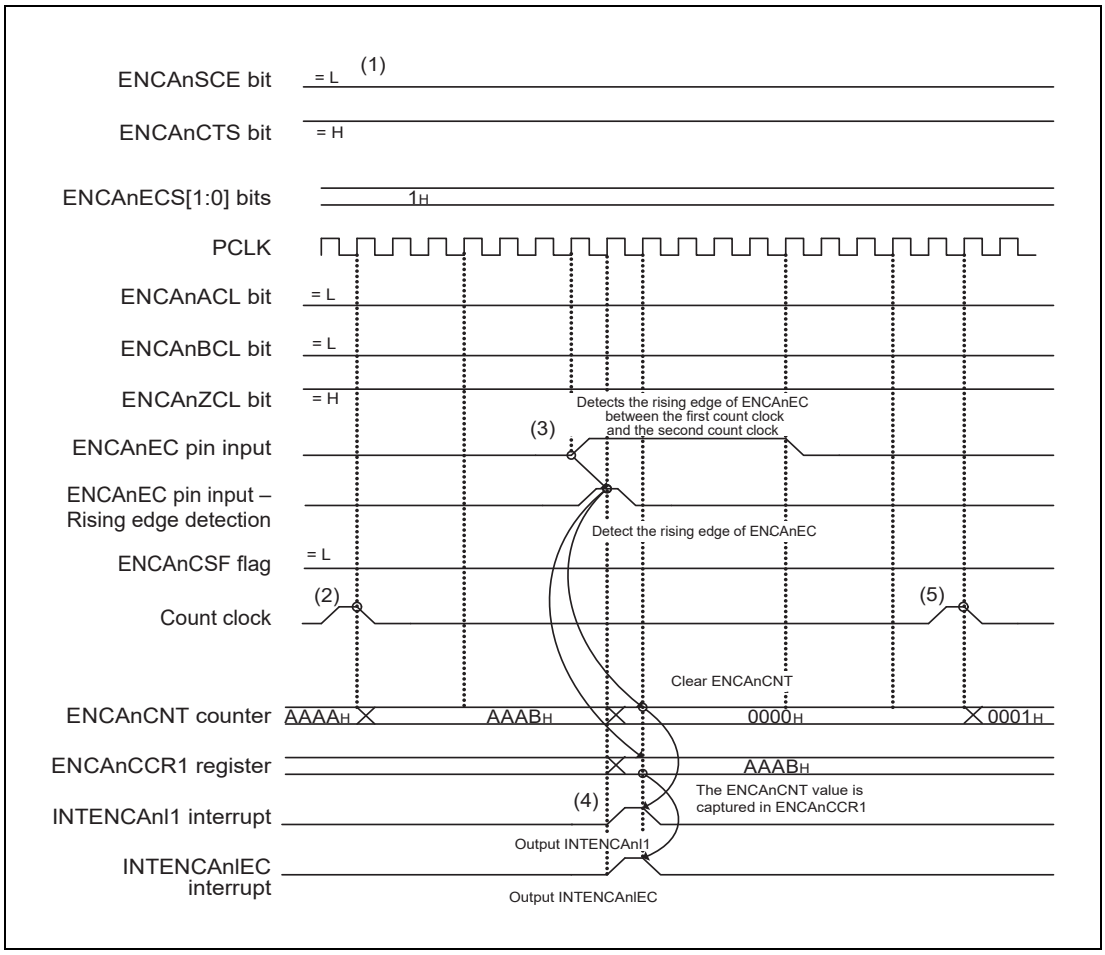


Figure 40.34 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

(1) The values are set as follows: ENCAnSCE = 0, ENCAnCTS = 1, and ENCAnECS[1:0] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCAnEC input is detected, and the ENCAnCNT value ($AAAB_H$) is captured in the ENCAnCCR1 register. Concurrently, clear operation by ENCAnEC is performed, and ENCAnCNT is reset to 0000_H.

(4) An interrupt (INTENCAnI1) corresponding to the capture to the ENCAnCCR1 register and an encoder clear interrupt (INTENCAnIEC) by ENCAnEC are output.

(5) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

Section 41 Peripheral Interconnect (PIC)

41.1 Features of RH850/U2A-EVA PIC

Peripheral interconnect (PIC) connects some peripherals with each other in order to achieve enhanced stand-alone functionality. This product includes two PIC units: PIC1 and PIC2.

41.1.1 Number of Units

This microcontroller has the following number of PIC units.

Table 41.1 Number of Units (PIC1)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1									
Name	PIC1									

Table 41.2 Number of Units (PIC2)

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	3									
Name	PIC20, PIC21, PIC22									

Table 41.3 Index

Index	Meaning
n	The individual unit of each timer and A/D converter is identified by the index “n”.
m	The individual channel of each timer is identified by the index “m”.
i	The individual unit of each GTM sub-modules (ATOM, TIM, MCS) is identified by the index “i”.
x	The individual channel of each GTM sub-modules (ATOM, TIM, MCS) is identified by the index “x”.
j	The scan group number of A/D converter is indicated by the index “j”.
k	The variable used for description is indicated by the index “k”.

41.1.2 Register Base Addresses

PIC base addresses are listed in the following table.

PIC register addresses are given as offsets from the base addresses.

Table 41.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PIC1_base>	FFBF AF00$_H$	P-Bus Group 5
<PIC20_base>	FFBF C000$_H$	P-Bus Group 5
<PIC21_base>	FFBF D000$_H$	P-Bus Group 5
<PIC22_base>	FFBF E000$_H$	P-Bus Group 5

41.1.3 Clock Supply

Clock supplies by and to PIC are listed in the following table.

Table 41.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
PIC1	PCLK	CLK_HSB	Peripheral high speed clock
PIC2	PCLK	CLK_HSB	Peripheral high speed clock

41.1.4 Reset Sources

PIC1, PIC2 reset sources are listed in the following table. PIC1, PIC2 are initialized by these reset sources.

Table 41.6 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PIC1	All registers	√	√	√	√	√	—	—
PIC2	All registers	√	√	√	√	√	—	—

41.1.5 Input/Output Signals

Table 41.7 PIC Input/Output Signals (1/4)

Signal Name	I/O	Function	Connected to
ENCA0TSST	O	ENCA0 simultaneous start trigger signal	ENCA0
ENCA0E0	O	ENCA0 encoder input signal (A phase)	
ENCA0E1	O	ENCA0 encoder input signal (B phase)	
ENCA0EC	O	ENCA0 encoder input signal (Z phase)	
ENCA0TTIN1	O	ENCA0 capture trigger	
INTENCA0IEC	I	Clear interrupt signal by the ENCA0 encoder input (Z phase)	
ENCAT0EQ0	I	ENCA0 match detection signal 0	
ENCAT0EQ1	I	ENCA0 match detection signal 1	
INTENCA0I0	I	ENCA0 capture/compare match interrupt 0	
INTENCA0I1	I	ENCA0 capture/compare match interrupt 1	
ENCA0TUDC	I	ENCA0 down-count enable signal output	
ENCA0TCKEN	I	ENCA0 count clock output	
ENCA1TSST	O	ENCA1 simultaneous start trigger signal	ENCA1
ENCA1E0	O	ENCA1 encoder input signal (A phase)	
ENCA1E1	O	ENCA1 encoder input signal (B phase)	
ENCA1EC	O	ENCA1 encoder input signal (Z phase)	
ENCA1TTIN1	O	ENCA1 capture trigger	
INTENCA1IEC	I	Clear interrupt signal by the ENCA1 encoder input (Z phase)	
ENCAT1EQ0	I	ENCA1 match detection signal 0	
ENCAT1EQ1	I	ENCA1 match detection signal 1	
INTENCA1I0	I	ENCA1 capture/compare match interrupt 0	
INTENCA1I1	I	ENCA1 capture/compare match interrupt 1	
ENCA1TUDC	I	ENCA1 down-count enable signal output	
ENCA1TCKEN	I	ENCA1 count clock output	
TAPA0THASIN	O	Hi-Z asynchronous signal	TAPA0
TAPA0TSIM0	O	TAUD0 master 0 INT input	
TAPA0TUDCM0	O	TAUD0 master 0 up/down input	
TAPA0TADOUT0	I	A/D conversion trigger output 0	
TAPA0TADOUT1	I	A/D conversion trigger output 1	
TAPA1THASIN	O	Hi-Z asynchronous signal	TAPA1
TAPA2THASIN	O	Hi-Z asynchronous signal	TAPA2
TAPA3THASIN	O	Hi-Z asynchronous signal	TAPA3
TAUD0TSSTm (m = 0 to 15)	O	TAUD0 simultaneous start trigger signal	TAUD0
TAUD0TTOUTm (m = 0 to 15)	I	TAUD0 channel output signal	
TAUD0TTINm (m = 0 to 15)	O	TAUD0 channel input signal	
INTTAUD0Im (m = 0 to 15)	I	TAUD0 interrupt signal	
TAUD1TSSTm (m = 0 to 15)	O	TAUD1 simultaneous start trigger signal	TAUD1
TAUD1TTOUTm (m = 0 to 15)	I	TAUD1 channel output signal	
TAUD1TTINm (m = 0 to 15)	O	TAUD1 channel input signal	
INTTAUD1Im (m = 0 to 15)	I	TAUD1 interrupt signal	

Table 41.7 PIC Input/Output Signals (2/4)

Signal Name	I/O	Function	Connected to
TAUJ0TSSTm (m = 0 to 3)	O	TAUJ0 simultaneous start trigger signal	TAUJ0
TAUJ0TTINm (m = 0 to 3)	O	TAUJ0 channel input signal	
INTTAUJ0I3	I	Interrupt output 3 of TAUJ0	
TAUJ1TSSTm (m = 0 to 3)	O	TAUJ1 simultaneous start trigger signal	TAUJ1
INTTAUJ1I3	I	Interrupt output 3 of TAUJ1	
TAUJ2TTINm (m = 0 to 3)	O	TAUJ2 channel input signal	TAUJ2
TPBA0TSST	O	TPBA0 simultaneous start trigger signal	TPBA0
TPBA1TSST	O	TPBA1 simultaneous start trigger signal	TPBA1
TSG30TSST	O	TSG30 simultaneous start trigger signal	TSG30
TSG30OPCI0	O	Timer output pattern control 0 signal	
TSG30OPCI1	O	Timer output pattern control 1 signal	
TSG30PTE	I	TSTAPT0-2 edge detection signal	
TSG30PEC	I	Two-phase encoder count signal	
TSG30PUD	I	Two-phase encoder up/down signal	
TSG30PTSI0	O	Hall sensor input signal 0	
TSG30PTSI1	O	Hall sensor input signal 1	
TSG30PTSI2	O	Hall sensor input signal 2	
TSG30ADTRG0	I	A/D trigger output signal 0	
TSG30ADTRG1	I	A/D trigger output signal 1	
TSG30O1	I	PWM output 1	
TSG30O2	I	PWM output 2	
TSG30O3	I	PWM output 3	
TSG30O4	I	PWM output 4	
TSG30O5	I	PWM output 5	
TSG30O6	I	PWM output 6	
INTTSG30I3	I	Timer interrupt 3	
INTTSG30I4	I	Timer interrupt 4	
INTTSG30I7	I	Timer interrupt 7	
INTTSG30I8	I	Timer interrupt 8	
INTTSG30I11	I	Timer interrupt 11	
INTTSG30I12	I	Timer interrupt 12	

Table 41.7 PIC Input/Output Signals (3/4)

Signal Name	I/O	Function	Connected to
TSG31TSST	O	TSG31 simultaneous start trigger signal	TSG31
TSG31OPCI0	O	Timer output pattern control 0 signal	
TSG31OPCI1	O	Timer output pattern control 1 signal	
TSG31PTE	I	TSTAPT0-2 edge detection signal	
TSG31PEC	I	Two-phase encoder count signal	
TSG31PUD	I	Two-phase encoder up/down signal	
TSG31PTSI0	O	Hall sensor input signal 0	
TSG31PTSI1	O	Hall sensor input signal 1	
TSG31PTSI2	O	Hall sensor input signal 2	
TSG31ADTRG0	I	A/D trigger output signal 0	
TSG31ADTRG1	I	A/D trigger output signal 1	
TSG31O1	I	PWM output 1	
TSG31O2	I	PWM output 2	
TSG31O3	I	PWM output 3	
TSG31O4	I	PWM output 4	
TSG31O5	I	PWM output 5	
TSG31O6	I	PWM output 6	
INTTSG31I3	I	Timer interrupt 3	
INTTSG31I4	I	Timer interrupt 4	
INTTSG31I7	I	Timer interrupt 7	
INTTSG31I8	I	Timer interrupt 8	
INTTSG31I11	I	Timer interrupt 11	
INTTSG31I12	I	Timer interrupt 12	
ADCJTTOUT0[j] (j = 0 to 4)	O	A/D converter 0 hardware trigger	ADCJ0
ADCJTTOUT1[j] (j = 0 to 4)	O	A/D converter 1 hardware trigger	ADCJ1
OSTM8TSST	O	Simultaneous start trigger signal of OSTM8	OSTM8
OSTM9TSST	O	Simultaneous start trigger signal of OSTM9	OSTM9
INTTSG30IER	I	Hi-Z control signal by TSG30 error interrupt	TSG30
INTTSG31IER	I	Hi-Z control signal by TSG31 error interrupt	TSG31
ERROROUTZ	I	ERROR output signal	ECM
INTADCJ0ERR	I	ADCJ0 error interrupt signal for Hi-Z control	ADCJ0
INTADCJ0Ij (j = 0 to 4)	I	Scan group j (SGj) end interrupt	
INTADCJ1ERR	I	ADCJ1 error interrupt signal for Hi-Z control	ADCJ1
INTADCJ1Ij (j = 0 to 4)	I	Scan group j (SGj) end interrupt	
WDTB0TNMI	I	WDTB0 error detection interrupt	WDTB0
WDTB1TNMI	I	WDTB1 error detection interrupt	WDTB1
WDTB2TNMI	I	WDTB2 error detection interrupt	WDTB2
WDTB3TNMI	I	WDTB3 error detection interrupt	WDTB3
INTTPTMU00	I	TPTM up-counter interrupt	TPTM PE0
INTTPTMU01	I	TPTM up-counter interrupt	
INTTPTMU10	I	TPTM up-counter interrupt	TPTM PE1
INTTPTMU11	I	TPTM up-counter interrupt	
INTTPTMU20	I	TPTM up-counter interrupt	TPTM PE2

Table 41.7 PIC Input/Output Signals (4/4)

Signal Name	I/O	Function	Connected to
INTTPTMU30	I	TPTM up-counter interrupt	TPTM PE3
INTGTMA0TIM[i]0 to INTGTMA0TIM[i]7 (i = 0 to 3)	I	TIM[i] (i = 0 to 3) shared interrupts	GTM
GTM_ATOM[i]_OUT0 to GTM_ATOM[i]_OUT7 (i = 0 to 3)	I	Timer output signals of ATOM[i] (i = 0 to 3)	
GTM_ATOM[i]_OUT0_N to GTM_ATOM[i]_OUT7_N (i = 0 to 3)	I	Inverted timer output signals of ATOM[i] (i = 0 to 3)	
INTGTMA0MCS[i]0 to INTGTMA0MCS[i]7 (i = 0 to 3)	I	MCS[i] (i = 0 to 3) interrupt for channel	
GTM_TIM[i]_IN0 to GTM_TIM[i]_IN7 (i = 0 to 3)	O	Timer input signals for TIM[i] (i = 0 to 3)	
PSIS0_TRG_SYNC_CH1	O	PSI5-S0 Synchronous trigger Ch1	PSI5-S0
PSIS0_TRG_SYNC_CH2	O	PSI5-S0 Synchronous trigger Ch2	
PSIS0_TRG_SYNC_CH3	O	PSI5-S0 Synchronous trigger Ch3	
PSIS0_TRG_SYNC_CH4	O	PSI5-S0 Synchronous trigger Ch4	
PSIS0_TRG_SYNC_CH5	O	PSI5-S0 Synchronous trigger Ch5	
PSIS0_TRG_SYNC_CH6	O	PSI5-S0 Synchronous trigger Ch6	
PSIS0_TRG_SYNC_CH7	O	PSI5-S0 Synchronous trigger Ch7	
PSIS0_CLK_TIMESTAMP_A	O	PSI5-S0 Timestamp counter A clock	
PSIS0_CLK_TIMESTAMP_B	O	PSI5-S0 Timestamp counter B clock	
PSIS0_CLR_TIMESTAMP_A	O	PSI5-S0 Timestamp counter A clear	
PSIS0_CLR_TIMESTAMP_B	O	PSI5-S0 Timestamp counter B clear	
PSIS0_STSP_TIMESTAMP_A	O	PSI5-S0 Timestamp counter A enable	
PSIS0_STSP_TIMESTAMP_B	O	PSI5-S0 Timestamp counter B enable	
PSIS1_TRG_SYNC_CH1	O	PSI5-S1 Synchronous trigger Ch1	PSI5-S1
PSIS1_TRG_SYNC_CH2	O	PSI5-S1 Synchronous trigger Ch2	
PSIS1_TRG_SYNC_CH3	O	PSI5-S1 Synchronous trigger Ch3	
PSIS1_TRG_SYNC_CH4	O	PSI5-S1 Synchronous trigger Ch4	
PSIS1_TRG_SYNC_CH5	O	PSI5-S1 Synchronous trigger Ch5	
PSIS1_TRG_SYNC_CH6	O	PSI5-S1 Synchronous trigger Ch6	
PSIS1_TRG_SYNC_CH7	O	PSI5-S1 Synchronous trigger Ch7	
PSIS1_CLK_TIMESTAMP_A	O	PSI5-S1 Timestamp counter A clock	
PSIS1_CLK_TIMESTAMP_B	O	PSI5-S1 Timestamp counter B clock	
PSIS1_CLR_TIMESTAMP_A	O	PSI5-S1 Timestamp counter A clear	
PSIS1_CLR_TIMESTAMP_B	O	PSI5-S1 Timestamp counter B clear	
PSIS1_STSP_TIMESTAMP_A	O	PSI5-S1 Timestamp counter A enable	
PSIS1_STSP_TIMESTAMP_B	O	PSI5-S1 Timestamp counter B enable	

41.1.6 External Input/Output Signals

External input/output signals of PIC are listed in the following table.

Table 41.8 PIC1 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
ENCA0I1	I	ENCA0 capture trigger input 1	ENCA0TIN1
ENCA1I1	I	ENCA1 capture trigger input 1	ENCA1TIN1
ENCA0E0	I	ENCA0 encoder input (count pulse 0)	ENCA0E0
ENCA0E1	I	ENCA0 encoder input (count pulse 1)	ENCA0E1
ENCA0EC	I	ENCA0 encoder input (clear pulse)	ENCA0EC
TAUD0Im (m = 0 to 15)	I	TAUD0 channel input m (m = 0 to 15)	TAUD0Im (m = 0 to 15)
TAUD1Im (m = 0 to 15)	I	TAUD1 channel input m (m = 0 to 15)	TAUD1Im (m = 0 to 15)
TAUD2Im (m = 0 to 15)	I	TAUD2 channel input m (m = 0 to 15)	TAUD2Im (m = 0 to 15)
TAPA0ESO	I	Hi-Z control for TAUD0/TSG30 output	TAPA0ESO
TAPA1ESO	I	Hi-Z control for TAUD1/TSG31 output	TAPA1ESO
TOP0U	O	Motor control output U phase	TAPA0UP
TOP0UB	O	Motor control output UB phase	TAPA0UN
TOP0V	O	Motor control output V phase	TAPA0VP
TOP0VB	O	Motor control output VB phase	TAPA0VN
TOP0W	O	Motor control output W phase	TAPA0WP
TOP0WB	O	Motor control output WB phase	TAPA0WN
TOP1U	O	Motor control output U phase	TAUD1O10
TOP1UB	O	Motor control output UB phase	TAUD1O11
TOP1V	O	Motor control output V phase	TAUD1O12
TOP1VB	O	Motor control output VB phase	TAUD1O13
TOP1W	O	Motor control output W phase	TAUD1O14
TOP1WB	O	Motor control output WB phase	TAUD1O15
TOP2U	O	Motor control output U phase	TSG30O1
TOP2UB	O	Motor control output UB phase	TSG30O2
TOP2V	O	Motor control output V phase	TSG30O3
TOP2VB	O	Motor control output VB phase	TSG30O4
TOP2W	O	Motor control output W phase	TSG30O5
TOP2WB	O	Motor control output WB phase	TSG30O6
TOP3U	O	Motor control output U phase	TSG31O1
TOP3UB	O	Motor control output UB phase	TSG31O2
TOP3V	O	Motor control output V phase	TSG31O3
TOP3VB	O	Motor control output VB phase	TSG31O4
TOP3W	O	Motor control output W phase	TSG31O5
TOP3WB	O	Motor control output WB phase	TSG31O6
TAUJ0Im	I	TAUJ input	TAUJ0Im (m = 0 to 3)
TAUJ1Im	I	TAUJ input	TAUJ1Im (m = 0 to 3)
TAUJ2Im	I	TAUJ input	TAUJ2Im (m = 0 to 3)

Table 41.9 PIC2 External Input/Output Signals

Unit Signal Name	I/O	Description	Alternative Port Pin Signal Name
ADCJ0TRGj (j = 0, 1, 2, 3, 4)	I	ADCJ0 trigger	ADCJ0TRGj (j = 0, 1, 2, 3, 4)
ADCJ1TRGj (j = 0, 1, 2, 3, 4)	I	ADCJ1 trigger	ADCJ1TRGj (j = 0, 1, 2, 3, 4)
ENCA0I0	I	ENCA0 capture trigger input 0	ENCA0TIN0
ENCA1I0	I	ENCA1 capture trigger input 0	ENCA1TIN0
ENCA1E0	I	ENCA1 encoder input (count pulse 0)	ENCA1E0
ENCA1E1	I	ENCA1 encoder input (count pulse 1)	ENCA1E1
ENCA1EC	I	ENCA1 encoder input (clear pulse)	ENCA1EC
$\overline{\text{ESO0}}$	I	Hi-Z control for GTM output	$\overline{\text{ESO0}}$
$\overline{\text{ESO1}}$	I	Hi-Z control for GTM output	$\overline{\text{ESO1}}$
$\overline{\text{ESO2}}$	I	Hi-Z control for GTM output	$\overline{\text{ESO2}}$
$\overline{\text{ESO3}}$	I	Hi-Z control for GTM output	$\overline{\text{ESO3}}$
GTM0I[x] (x = 0 to 7)	I	GTM timer input signal (TIM0)	GTM0I[x] (x = 0 to 7)
GTM1I[x] (x = 0 to 7)	I	GTM timer input signal (TIM1)	GTM1I[x] (x = 0 to 7)
GTM2I[x] (x = 0 to 7)	I	GTM timer input signal (TIM2)	GTM2I[x] (x = 0 to 7)
GTM3[x] (x = 0 to 7)	I	GTM timer input signal (TIM3)	GTM3I[x] (x = 0 to 7)
RLIN30RX	I	RLIN30 soft macro data input (RX)	RLIN30RX
RLIN31RX	I	RLIN31 soft macro data input (RX)	RLIN31RX
RLIN32RX	I	RLIN32 soft macro data input (RX)	RLIN32RX
RLIN33RX	I	RLIN33 soft macro data input (RX)	RLIN33RX

41.1.7 Block diagram

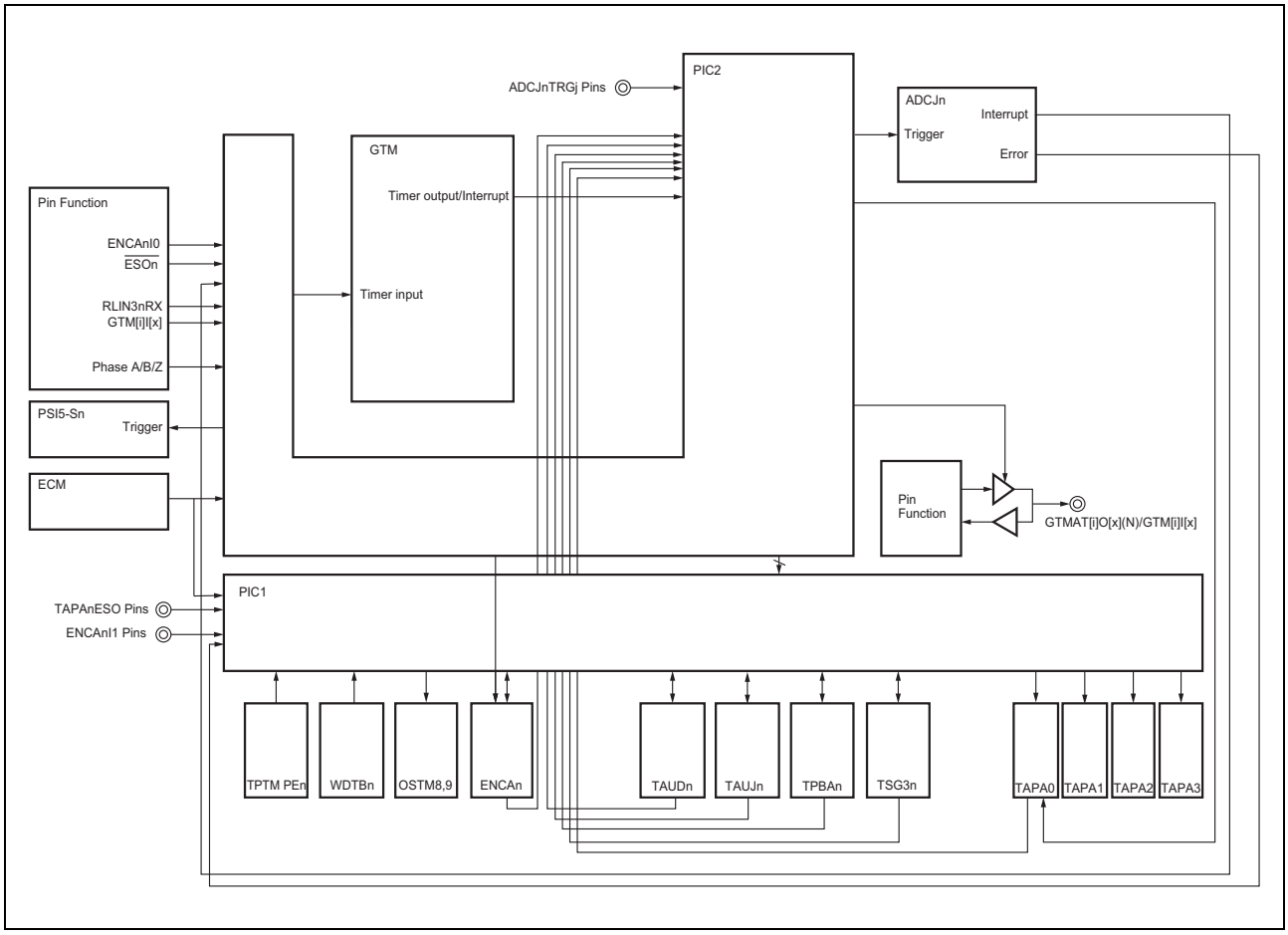


Figure 41.1 PIC block diagram

41.2 Peripheral Interconnect — 1 (PIC1)

41.2.1 Overview

41.2.1.1 Functional Overview

The peripheral interconnect-1 (PIC1) realizes various functions by synchronous operation using multiple timers and by connecting the timer internal I/O signals between the timers.

CAUTION

The signal names used in the following descriptions are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

INTm →INTTAUDnIm

TINm → TAUDnTTINm

TOUTm → TAUDnTTOUTm

CDRm → TAUDnCDRm

CNTm → TAUDnCNTm

The PIC1 has the following functions.

- Simultaneous start trigger function
- INT Signal Output Selection Function
- PWM output function with dead time
- High accuracy triangle wave PWM output function with dead time
- Delay pulse output function with dead time
- Trigger pulse width measurement function
- Encoder capture trigger select function
- Two-phase encoder control function (control method 1)
- Two-phase encoder control function (control method 2)
- Three-phase pulse input control function
- Three-phase encoder control function
- Timer input select function
- Hi-Z control function
- Timer output monitor function
- Timer input monitor function
- TSG3 Synchronous Start and Clear Function

41.2.2 Registers

41.2.2.1 List of Registers

The registers are listed in the following table.

The bits in 32-bit registers can be accessed only in 32-bit units. Access in 16-bits or 8-bits is operated as 32-bit access.

Table 41.10 List of Registers

Register Function	Symbol	Address	Access size	Access Protection	
				PBG	Other
Simultaneous start trigger control register	PIC1SST	<PIC1_base> + 04_H	8	PBG52#0	—
Simultaneous start control register 0	PIC1SSER0	<PIC1_base> + 10_H	16	PBG52#0	—
Simultaneous start control register 1	PIC1SSER1	<PIC1_base> + 14_H	16	PBG52#0	—
Simultaneous start control register 2	PIC1SSER2	<PIC1_base> + 18_H	16	PBG52#0	—
Simultaneous start control register 3	PIC1SSER3	<PIC1_base> + $1C_H$	16	PBG52#0	—
RS flip-flop circuit initialization register 00	PIC1INI00	<PIC1_base> + 20_H	8	PBG52#0	—
DT initialization register 01	PIC1INI01	<PIC1_base> + 24_H	8	PBG52#0	—
RS flip-flop circuit initialization register 10	PIC1INI10	<PIC1_base> + $2C_H$	8	PBG52#0	—
DT initialization register 11	PIC1INI11	<PIC1_base> + 30_H	8	PBG52#0	—
Hall sensor input select register	PIC1TSGHALLSEL	<PIC1_base> + 74_H	8	PBG52#0	—
TAUD0 input select register	PIC1TAUD0SEL	<PIC1_base> + 78_H	32	PBG52#0	—
TAUD1 input select register	PIC1TAUD1SEL	<PIC1_base> + $7C_H$	32	PBG52#0	—
Hi-Z control register 0	PIC1HIZCEN0	<PIC1_base> + 80_H	8	PBG52#0	—
Hi-Z control register 1	PIC1HIZCEN1	<PIC1_base> + 84_H	8	PBG52#0	—
Hi-Z control register 2	PIC1HIZCEN2	<PIC1_base> + 88_H	8	PBG52#0	—
Hi-Z control register 3	PIC1HIZCEN3	<PIC1_base> + $8C_H$	8	PBG52#0	—
ENCATIN1 input select register 400	PIC1ENCSEL400	<PIC1_base> + $B8_H$	8	PBG52#0	—
ENCATIN1 input select register 410	PIC1ENCSEL410	<PIC1_base> + BC_H	8	PBG52#0	—
Timer input/output control register 200	PIC1REG200	<PIC1_base> + $C0_H$	32	PBG52#0	—
Timer input/output control register 201	PIC1REG201	<PIC1_base> + $C4_H$	32	PBG52#0	—
Timer input/output control register 202	PIC1REG202	<PIC1_base> + $C8_H$	32	PBG52#0	—
Timer input/output control register 203	PIC1REG203	<PIC1_base> + CC_H	32	PBG52#0	—
Timer input/output control register 210	PIC1REG210	<PIC1_base> + $D4_H$	32	PBG52#0	—
Timer input/output control register 211	PIC1REG211	<PIC1_base> + $D8_H$	32	PBG52#0	—
Timer input/output control register 212	PIC1REG212	<PIC1_base> + DC_H	32	PBG52#0	—
Timer input/output control register 213	PIC1REG213	<PIC1_base> + $E0_H$	32	PBG52#0	—
Timer input/output control register 30	PIC1REG30	<PIC1_base> + $E8_H$	32	PBG52#0	—
Timer input/output control register 31	PIC1REG31	<PIC1_base> + EC_H	32	PBG52#0	—
Timer input/output control register 50	PIC1REG50	<PIC1_base> + $F8_H$	16	PBG52#0	—
Timer input/output control register 51	PIC1REG51	<PIC1_base> + FC_H	16	PBG52#0	—
Synchronous start and clear enable register	PIC1SELBSSER	FFBF $B200_H$	8	PBG52#0	—
Port output monitor select register	PIC1POMONSEL	FFBF $B400_H$	8	PBG52#0	—
Port input monitor select register	PIC1PIMONSEL	FFBF $B600_H$	8	PBG52#0	—
TAUD2TTINm Input Signal Selection Register	PIC1SELB_TAUD2I	FFBF 6800_H	16	PBG51#3	—
TAUJ2TTINm Input Signal Selection Register	PIC1SELB_TAUJ2I	FFE8 0800_H	8	PBG20#5	—
TAUJ3TTINm Input Signal Selection Register	PIC1SELB_TAUJ3I	FFE8 1800_H	8	PBG20#6	—

Combinations of registers used for each function are listed in the following table.

Table 41.11 Registers Used by Each Function

Section Number	Function Name	PIC1SST	PIC1SSER				PIC1INI				PIC1TSGHALLSEL	PIC1TAUD0SEL	PIC1TAUD1SEL	PIC1HIZCEN				PIC1ENCSEL4		PIC1REG												PIC1POMONSEL	PIC1PIMONSEL	PIC1SELBSSER	PIC1SELB_TAUD2I	PIC1SELB_TAUJ2I	PIC1SELB_TAUJ3I
			0	1	2	3	00	01	10	11	—	—	—	0	1	2	3	00	10	200	201	202	203	210	211	212	213	30	31	50	51						
41.2.3.1	Simultaneous Start Trigger Function	√	√	√	√	√	—	—	—	—	—	√	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
41.2.3.2	INT Signal Output Selection Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
41.2.3.3	PWM Output Function With Dead Time	—	—	—	—	—	√	—	√	—	—	√	√	—	—	—	—	—	—	—	√	√	√	—	√	√	√	—	—	—	—	—	—	—	—	—
41.2.3.4	High Accuracy Triangle Wave PWM Output Function with Dead Time	—	—	—	—	—	—	—	—	—	—	√	√	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	—	—	—	—	—	—	—
41.2.3.5	Delay Pulse Output Function with Dead Time	—	—	—	—	—	—	—	—	—	—	√	√	—	—	—	—	—	—	—	√	√	√	—	√	√	√	—	—	—	—	—	—	—	—	—
41.2.3.6	Trigger Pulse Width Measurement Function	—	—	—	—	—	—	√	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	—	—	—	—	—	—	—
41.2.3.7	Encoder Capture Trigger Select Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—	—	—
41.2.3.8	Two-Phase Encoder Control Function (Control Method 1)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	—	—	—	—
41.2.3.9	Two-Phase Encoder Control Function (Control Method 2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	—	—	—	—
41.2.3.10	Three-Phase Pulse Input Control Function	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—	—	—	√	—	√	—	√	—	√	—	—	—	√	√	—	—	—	—	—
41.2.3.11	Three-Phase Encoder Control Function	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	—	—	—	—
41.2.3.12	Timer Input Select Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	—	√	—	√	—	—	√	—	—	—	—	—	√	√	√
41.2.3.13	Hi-Z Control Function	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
41.2.3.14	Timer Output Monitor Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—
41.2.3.15	Timer Input Monitor Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—	—
41.2.3.16	TSG3 Synchronous Start and Clear Function	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	—	—

41.2.2.2 PIC1SST — Simultaneous Start Trigger Control Register

The PIC1SST register is an 8-bit register that selects the simultaneous start trigger.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PIC1SYNCTRG
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 41.12 PIC1SST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1SYNCTRG	Generates a start trigger for the timer for which simultaneous start is enabled. 0: Disabled 1: Simultaneous start trigger (pulse with a width of 1PCLK is output).

Note: PIC1SYNCTRG always reads 0 when read.

41.2.2.3 PIC1SSER0 — Simultaneous Start Control Register 0

The PIC1SSER0 register enables a start trigger for each channel of TAUD0.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + 10_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1SSER015	PIC1SSER014	PIC1SSER013	PIC1SSER012	PIC1SSER011	PIC1SSER010	PIC1SSER009	PIC1SSER008	PIC1SSER007	PIC1SSER006	PIC1SSER005	PIC1SSER004	PIC1SSER003	PIC1SSER002	PIC1SSER001	PIC1SSER000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.13 PIC1SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC1SSER0m	Enables or disables a simultaneous start trigger for CHm of TAUD0. 0: Disabled 1: Enabled

41.2.2.4 PIC1SSER1 — Simultaneous Start Control Register 1

The PIC1SSER1 register enables a start trigger for each channel of TAUD1.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + 14_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1SSER115	PIC1SSER114	PIC1SSER113	PIC1SSER112	PIC1SSER111	PIC1SSER110	PIC1SSER109	PIC1SSER108	PIC1SSER107	PIC1SSER106	PIC1SSER105	PIC1SSER104	PIC1SSER103	PIC1SSER102	PIC1SSER101	PIC1SSER100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.14 PIC1SSER1 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC1SSER1m	Enables or disables a simultaneous start trigger for CHm of TAUD1. 0: Disabled 1: Enabled

41.2.2.5 PIC1SSER2 — Simultaneous Start Control Register 2

The PIC1SSER2 register enables a start trigger of TAUJn, TSG3n, TPBAn, and ENCAn.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + 18_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PIC1SSER213	PIC1SSER212	PIC1SSER211	PIC1SSER210	PIC1SSER209	PIC1SSER208	PIC1SSER207	PIC1SSER206	PIC1SSER205	PIC1SSER204	PIC1SSER203	PIC1SSER202	PIC1SSER201	PIC1SSER200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.15 PIC1SSER2 Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	PIC1SSER213	Enables or disables a simultaneous start trigger for the ENCA1. 0: Disabled 1: Enabled
12	PIC1SSER212	Enables or disables a simultaneous start trigger for the ENCA0. 0: Disabled 1: Enabled
11	PIC1SSER211	Enables or disables a simultaneous start trigger for the TPBA1. 0: Disabled 1: Enabled
10	PIC1SSER210	Enables or disables a simultaneous start trigger for the TPBA0. 0: Disabled 1: Enabled
9	PIC1SSER209	Enables or disables a simultaneous start trigger for the TSG31. 0: Disabled 1: Enabled
8	PIC1SSER208	Enables or disables a simultaneous start trigger for the TSG30. 0: Disabled 1: Enabled
7	PIC1SSER207	Enables or disables a simultaneous start trigger for CH03 of TAUJ1. 0: Disabled 1: Enabled
6	PIC1SSER206	Enables or disables a simultaneous start trigger for CH02 of TAUJ1. 0: Disabled 1: Enabled
5	PIC1SSER205	Enables or disables a simultaneous start trigger for CH01 of TAUJ1. 0: Disabled 1: Enabled
4	PIC1SSER204	Enables or disables a simultaneous start trigger for CH00 of TAUJ1. 0: Disabled 1: Enabled
3	PIC1SSER203	Enables or disables a simultaneous start trigger for CH03 of TAUJ0. 0: Disabled 1: Enabled
2	PIC1SSER202	Enables or disables a simultaneous start trigger for CH02 of TAUJ0. 0: Disabled 1: Enabled
1	PIC1SSER201	Enables or disables a simultaneous start trigger for CH01 of TAUJ0. 0: Disabled 1: Enabled

Table 41.15 PIC1SSER2 Register Contents (2/2)

Bit Position	Bit Name	Function
0	PIC1SSER200	Enables or disables a simultaneous start trigger for CH00 of TAUJ0. 0: Disabled 1: Enabled

41.2.2.6 PIC1SSER3 — Simultaneous Start Control Register 3

The PIC1SSER3 register enables a start trigger for each channel of OSTM8 and OSTM9.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + $1C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC1SSER301	PIC1SSER300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 41.16 PIC1SSER3 Register Contents

Bit Position	Bit Name	Function
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PIC1SSER301	Enables or disables a simultaneous start trigger for OSTM9. 0: Disabled 1: Enabled
0	PIC1SSER300	Enables or disables a simultaneous start trigger for OSTM8. 0: Disabled 1: Enabled

41.2.2.7 PIC1INIn0 — Flip-Flop Circuit Initialization Register n0 (n = 0, 1)

The PIC1INIn0 register enables initialization of the RS flip-flop circuits 4 to 2 (RSn4 to 2).

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 20_H (n=0), <PIC1_base> + $2C_H$ (n=1)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	PIC1INIn04	PIC1INIn03	PIC1INIn02	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R	R

Table 41.17 PIC1INIn0 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 2	PIC1INIn0[4:2]	Enables or disables initialization of the RS flip-flop circuits 4 to 2 (RSn4 to RSn2) used for the PWM output function with dead time. These bits are always read as 0. 0: Disabled 1: Initialized
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

41.2.2.8 PIC1INIn1 — DT Initialization Register n1 (n = 0, 1)

The PIC1INIn1 register enables initialization of the latch & toggle (DT) circuit.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 24_H (n=0), <PIC1_base> + 30_H (n=1)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	PIC1INIn12	PIC1INIn11	PIC1INIn10
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 41.18 PIC1INIn1 Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PIC1INIn1[2:0]	Enables or disables initialization of the DT circuit to be used for the trigger pulse width measurement function. These bits are always read as 0. 0: Disabled 1: Initialized

41.2.2.9 PIC1TSGHALLSEL — Hall Sensor Input Select Register

The PIC1TSGHALLSEL register sets the pin conditions to input the external hall sensor signal to the Motor Control Timer (TSG3).

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 74_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSG1HALLSEL	TSG0HALLSEL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 41.19 PIC1TSGHALLSEL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TSG1HALLSEL	Sets pin condition to input the external hall sensor signal.*1 0: Separate input 1: Alternative input with ENCA
0	TSG0HALLSEL	Sets pin condition to input the external hall sensor signal.*1 0: Separate input 1: Alternative input with ENCA

Note 1. When TSG3nPTSI2 to TSG3nPTSI0 are used for input pattern to the TSG3, always set this bit to 1. In addition, set bit 0 in the PIC1REG50 register and bit 0 in the PIC1REG51 register as described below. When those bits are not used, do not change the setting from the value after reset.

TSG1HALLSEL	PIC1REG5100	Function
1	1	Select input pin ENCA1E0, ENCA1E1, and ENCA1EC.
Other than above		Setting prohibited.

TSG0HALLSEL	PIC1REG5000	Function
1	0	Select input pin ENCA0E0, ENCA0E1 and ENCA0EC
Other than above		Setting prohibited

41.2.2.10 PIC1TAUD0SEL — TAUD0 Input Select Register

The PIC1TAUD0SEL register is a 32-bit register that selects TAUDTIN input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + 78_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC1TAUD0IN143	PIC1TAUD0IN142	PIC1TAUD0IN141	PIC1TAUD0IN140	PIC1TAUD0IN123	PIC1TAUD0IN122	PIC1TAUD0IN121	PIC1TAUD0IN120	PIC1TAUD0IN103	PIC1TAUD0IN102	PIC1TAUD0IN101	PIC1TAUD0IN100	PIC1TAUD0IN83	PIC1TAUD0IN82	PIC1TAUD0IN81	PIC1TAUD0IN80
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1TAUD0IN63	PIC1TAUD0IN62	PIC1TAUD0IN61	PIC1TAUD0IN60	PIC1TAUD0IN43	PIC1TAUD0IN42	PIC1TAUD0IN41	PIC1TAUD0IN40	PIC1TAUD0IN23	PIC1TAUD0IN22	PIC1TAUD0IN21	PIC1TAUD0IN20	PIC1TAUD0IN03	PIC1TAUD0IN02	PIC1TAUD0IN01	PIC1TAUD0IN00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.20 PIC1TAUD0SEL Register Contents

Bit Position	Bit Name	Function
2m+3 2m+2	PIC1TAUD0INm [3:2]	Selects the signal to output to TAUD0TTIN [m+1] output pin. 00: TAUD0I [m+1] is selected. 01: TAUD0I [m] is selected. 10: TAUD1I [m+1] is selected. 11: TAUD1I [m] is selected.
2m+1 2m	PIC1TAUD0INm [1:0]	Selects the signal to output to TAUD0TTIN [m] output pin. 00: TAUD0I [m] is selected. 01: TAUD0I [m+1] is selected. 10: TAUD1I [m] is selected. 11: TAUD1I [m+1] is selected.

Note: m is an even channel number of TAUD0 (CHm_even)

41.2.2.11 PIC1TAUD1SEL — TAUD1 Input Select Register

The PIC1TAUD1SEL register is a 32-bit register that selects TAUDTIN input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $7C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC1TAUD1IN143	PIC1TAUD1IN142	PIC1TAUD1IN141	PIC1TAUD1IN140	PIC1TAUD1IN123	PIC1TAUD1IN122	PIC1TAUD1IN121	PIC1TAUD1IN120	PIC1TAUD1IN103	PIC1TAUD1IN102	PIC1TAUD1IN101	PIC1TAUD1IN100	PIC1TAUD1IN83	PIC1TAUD1IN82	PIC1TAUD1IN81	PIC1TAUD1IN80
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1TAUD1IN63	PIC1TAUD1IN62	PIC1TAUD1IN61	PIC1TAUD1IN60	PIC1TAUD1IN43	PIC1TAUD1IN42	PIC1TAUD1IN41	PIC1TAUD1IN40	PIC1TAUD1IN23	PIC1TAUD1IN22	PIC1TAUD1IN21	PIC1TAUD1IN20	PIC1TAUD1IN03	PIC1TAUD1IN02	PIC1TAUD1IN01	PIC1TAUD1IN00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.21 PIC1TAUD1SEL Register Contents

Bit Position	Bit Name	Function
2m+3 2m+2	PIC1TAUD1INm [3:2]	Selects the signal to output to TAUD1TTIN [m+1] output pin 00: TAUD1I [m+1] is selected. 01: TAUD1I [m] is selected. 10: TAUD0I [m+1] is selected. 11: TAUD0I [m] is selected.
2m+1 2m	PIC1TAUD1INm [1:0]	Selects the signal to output to TAUD1TTIN [m] output pin 00: TAUD1I [m] is selected. 01: TAUD1I [m+1] is selected. 10: TAUD0I [m] is selected. 11: TAUD0I [m+1] is selected.

Note: m is an even channel number of TAUD1 (CHm_even)

41.2.2.12 PIC1HIZCEN0 — Hi-Z Control Register 0

The PIC1HIZCEN0 register selects Hi-Z control input signals of ADCJ0, ADCJ1, ERROROUTZ, WDTB, and TAPA0ESO for TAUD0.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 80_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1HIZCEN 07	PIC1HIZCEN 06	PIC1HIZCEN 05	—	—	PIC1HIZCEN 02	—	PIC1HIZCEN 00
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R/W	R	R/W

Table 41.22 PIC1HIZCEN0 Register Contents

Bit Position	Bit Name	Function
7	PIC1HIZCEN 07	Enables or disables Hi-Z control by ADCJ1 error interrupt (INTADCJ1ERR). 0: Disabled 1: Enabled
6	PIC1HIZCEN 06	Enables or disables Hi-Z control by ADCJ0 error interrupt (INTADCJ0ERR). 0: Disabled 1: Enabled
5	PIC1HIZCEN 05	Enables or disables Hi-Z control by ERROROUTZ signal. 0: Disabled 1: Enabled
4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC1HIZCEN 02	Enables or disables Hi-Z control by WDTBn(n=0 to 3) (WDTBTNMI). 0: Disabled 1: Enabled
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1HIZCEN 00	Enables or disables Hi-Z control by TAPA0ESO pin input. 0: Disabled 1: Enabled

CAUTIONS

1. Set this register before starting U/V/W outputs or UB/VB/WB outputs of TAUD0.
2. Set TAPA0CTL0.TAPA0DCN = 0 and TAPA0CTL0.TAPA0DCP = 1 when performing Hi-Z control by ADCJ error signal, WDTB error interrupt signal and ERROROUTZ signal.

41.2.2.13 PIC1HIZCEN1 — Hi-Z Control Register 1

The PIC1HIZCEN1 register selects Hi-Z control input signals of ADCJ1, ADCJ0, ERROROUTZ, and TAPA1ESO for TAUD1.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 84_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1HIZCEN 17	PIC1HIZCEN 16	PIC1HIZCEN 15	—	—	—	—	PIC1HIZCEN 10
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 41.23 PIC1HIZCEN1 Register Contents

Bit Position	Bit Name	Function
7	PIC1HIZCEN 17	Enables or disables Hi-Z control by ADCJ1 error interrupt (INTADCJ1ERR). 0: Disabled 1: Enabled
6	PIC1HIZCEN 16	Enables or disables Hi-Z control by ADCJ0 error interrupt (INTADCJ0ERR). 0: Disabled 1: Enabled
5	PIC1HIZCEN 15	Enables or disables Hi-Z control by ERROROUTZ signal. 0: Disabled 1: Enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1HIZCEN 10	Enables or disables Hi-Z control by TAPA1ESO pin input. 0: Disabled 1: Enabled

CAUTIONS

1. Set this register before starting U/V/W outputs or UB/VB/WB outputs of TAUD1.
2. Set TAPA1CTL0.TAPA1DCN = 0 and TAPA1CTL0.TAPA1DCP = 1 when performing Hi-Z control by ADCJ error signal and ERROROUTZ signal.

41.2.2.14 PIC1HIZCEN2 — Hi-Z Control Register 2

The PIC1HIZCEN2 register selects Hi-Z control input signals of ADCJ1, ADCJ0, ERROROUTZ, TSG30, and TAPA0ESO for TSG30.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + 88_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1HIZCEN 27	PIC1HIZCEN 26	PIC1HIZCEN 25	—	PIC1HIZCEN 23	—	—	PIC1HIZCEN 20
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R	R	R/W

Table 41.24 PIC1HIZCEN2 Register Contents

Bit Position	Bit Name	Function
7	PIC1HIZCEN 27	Enables or disables Hi-Z control by ADCJ1 error interrupt (INTADCJ1ERR). 0: Disabled 1: Enabled
6	PIC1HIZCEN 26	Enables or disables Hi-Z control by ADCJ0 error interrupt (INTADCJ0ERR). 0: Disabled 1: Enabled
5	PIC1HIZCEN 25	Enables or disables Hi-Z control by ERROROUTZ signal. 0: Disabled 1: Enabled
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PIC1HIZCEN 23	Enables or disables Hi-Z control by INTTSG30IER interrupt signal. 0: Disabled 1: Enabled
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1HIZCEN 20	Enables or disables Hi-Z control by TAPA0ESO pin input. 0: Disabled 1: Enabled

CAUTIONS

1. Set this register before starting TSG30 output.
2. Set TAPA2CTL0.TAPA2DCN = 0 and TAPA2CTL0.TAPA2DCP = 1 when performing Hi-Z control by ADCJ error signal, ERROROUTZ signal, and TSG30 error signal.

41.2.2.15 PIC1HIZCEN3 — Hi-Z Control Register 3

The PIC1HIZCEN3 register selects Hi-Z control input signals of ADCJ1, ADCJ0, ERROROUTZ, TSG31, and TAPA1ESO for TSG31.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + $8C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1HIZCEN 37	PIC1HIZCEN 36	PIC1HIZCEN 35	PIC1HIZCEN 34	—	—	—	PIC1HIZCEN 30
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 41.25 PIC1HIZCEN3 Register Contents

Bit Position	Bit Name	Function
7	PIC1HIZCEN 37	Enables or disables Hi-Z control by ADCJ1 error interrupt (INTADCJ1ERR) 0: Disabled 1: Enabled
6	PIC1HIZCEN 36	Enables or disables Hi-Z control by ADCJ0 error interrupt (INTADCJ0ERR). 0: Disabled 1: Enabled
5	PIC1HIZCEN 35	Enables or disables Hi-Z control by ERROROUTZ signal. 0: Disabled 1: Enabled
4	PIC1HIZCEN 34	Enables or disables Hi-Z control by INTTSG31IER interrupt signal. 0: Disabled 1: Enabled
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1HIZCEN 30	Enables or disables Hi-Z control by TAPA1ESO pin input. 0: Disabled 1: Enabled

CAUTION

Set this register before starting TSG31 output.

Set TAPA3CTL0.TAPA3DCN = 0 and TAPA3CTL0.TAPA3DCP = 1 when performing Hi-Z control by ADCJ error signal, ERROROUTZ signal, and TSG31 error signal.

41.2.2.16 PIC1ENCSEL400 — ENCATIN1 Input Select Register 400

The PIC1ENCSEL400 register is used for encoder capture trigger select function.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + $B8_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1ENCSEL 4007	—	—	—	PIC1ENCSEL 4003	PIC1ENCSEL 4002	PIC1ENCSEL 4001	PIC1ENCSEL 4000
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 41.26 PIC1ENCSEL400 Register Contents

Bit Position	Bit Name	Function
7	PIC1ENCSEL 4007	Enables or disables output of INTTAUD0Im signal selected by PIC1ENCSEL400[3:0]. 0: Disabled 1: Enabled
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	PIC1ENCSEL 400[3:0]	Selects INTTAUD0Im to be used as a capture trigger signal for ENCA0 and ENCA1. 0000_B: INTTAUD0I0 is selected 0001_B: INTTAUD0I1 is selected 0010_B: INTTAUD0I2 is selected 0011_B: INTTAUD0I3 is selected 0100_B: INTTAUD0I4 is selected 0101_B: INTTAUD0I5 is selected 0110_B: INTTAUD0I6 is selected 0111_B: INTTAUD0I7 is selected 1000_B: INTTAUD0I8 is selected 1001_B: INTTAUD0I9 is selected 1010_B: INTTAUD0I10 is selected 1011_B: INTTAUD0I11 is selected 1100_B: INTTAUD0I12 is selected 1101_B: INTTAUD0I13 is selected 1110_B: INTTAUD0I14 is selected 1111_B: INTTAUD0I15 is selected

41.2.2.17 PIC1ENCSEL410 — ENCATIN1 Input Select Register 410

The PIC1ENCSEL410 register is used for encoder capture trigger select function.

Access: This register can be read or written in 8-bit units.

Address: <PIC1_base> + BC_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PIC1ENCSEL4107	—	—	—	PIC1ENCSEL4103	PIC1ENCSEL4102	PIC1ENCSEL4101	PIC1ENCSEL4100
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 41.27 PIC1ENCSEL410 Register Contents

Bit Position	Bit Name	Function
7	PIC1ENCSEL4107	Enables or disables output of INTTAUD1Im signal selected by PIC1ENCSEL410[3:0]. 0: Disabled 1: Enabled
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	PIC1ENCSEL410[3:0]	Selects INTTAUD1Im to be used as a capture trigger signal for ENCA0 and ENCA1. 0000_B: INTTAUD1I0 is selected 0001_B: INTTAUD1I1 is selected 0010_B: INTTAUD1I2 is selected 0011_B: INTTAUD1I3 is selected 0100_B: INTTAUD1I4 is selected 0101_B: INTTAUD1I5 is selected 0110_B: INTTAUD1I6 is selected 0111_B: INTTAUD1I7 is selected 1000_B: INTTAUD1I8 is selected 1001_B: INTTAUD1I9 is selected 1010_B: INTTAUD1I10 is selected 1011_B: INTTAUD1I11 is selected 1100_B: INTTAUD1I12 is selected 1101_B: INTTAUD1I13 is selected 1110_B: INTTAUD1I14 is selected 1111_B: INTTAUD1I15 is selected

41.2.2.18 PIC1REG200 — Timer Input/Output Control Register 200

The PIC1REG200 register selects TAUD0 input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $C0_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC1REG20025	PIC1REG20024	—	—	—	—	—	PIC1REG20018	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIC1REG20011	PIC1REG20010	PIC1REG20009	PIC1REG20008	—	—	—	—	PIC1REG20003	PIC1REG20002	PIC1REG20001	PIC1REG20000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 41.28 PIC1REG200 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	PIC1REG200 [25:24]	Select the TAUD0 channel used by TAPA0TSIM0 and TAPA0TUDCM0. 00: Not selected. 01: TAUD0 channel 0 selected. 10: TAUD0 channel 2 selected. 11: TAUD0 channel 8 selected.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PIC1REG20018	Select the signal for input as TAUD0TTIN10, TAUD0TTIN12, and TAUD0TTIN14 signals of TAUD0. 1: Select TOUT of TAUD0 CH02. Settings other than above are prohibited.*1
17 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11, 10	PIC1REG200 [11:10]	Select the signal for input as TAUD0TTIN6, TAUD0TTIN7 signals of TAUD0. 10: TSG30PTE signal of TSG30. Settings other than above are prohibited.*1
9, 8	PIC1REG200 [09:08]	Select the signal for input as TAUD0TTIN4, TAUD0TTIN5 signals of TAUD0. 10: TSG30PTE signal of TSG30. Settings other than above are prohibited.*1
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PIC1REG20003	Selects the signal for input as TAUD0TTIN7 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1REG20011, PIC1REG20010 bits (TSG30PTE signal).
2	PIC1REG20002	Select the signal for input as TAUD0TTIN6 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1REG20011 and PIC1REG20010 bits (TSG30PTE signal).
1	PIC1REG20001	Select the signal for input as TAUD0TTIN5 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1REG20009, PIC1REG20008 bits (TSG30PTE signal).

Table 41.28 PIC1REG200 Register Contents (2/2)

Bit Position	Bit Name	Function
0	PIC1REG20000	Select the signal for input as TAUD0TTIN4 signal of TAUD0. 0: TIN pin input 1: Input signal selected by PIC1REG20009 and PIC1REG20008 bits (TSG30PTE signal).

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used.

41.2.2.19 PIC1REG210 — Timer Input/Output Control Register 210

The PIC1REG210 register selects TAUD1 input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $D4_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC1REG21018	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PIC1REG21011	PIC1REG21010	PIC1REG21009	PIC1REG21008	—	—	—	—	PIC1REG21003	PIC1REG21002	PIC1REG21001	PIC1REG21000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 41.29 PIC1REG210 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PIC1REG21018	Selects the signal for input as TAUD1TTIN10, TAUD1TTIN12, and TAUD1TTIN14 signals of TAUD1. 1: Select TOUT of TAUD1 CH02. Settings other than above are prohibited.*1
17 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11, 10	PIC1REG210 [11:10]	Selects the signal to be input as TAUD1TTIN6 and TAUD1TTIN7 signals of TAUD1. 10: TSG31PTE signal of TSG31 Settings other than above are prohibited.*1
9, 8	PIC1REG210 [09:08]	Selects the signal to be input as TAUD1TTIN4 and TAUD1TTIN5 signals of TAUD1. 10: TSG31PTE signal of TSG31 Settings other than above are prohibited.*1
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	PIC1REG21003	Selects the signal to be input as TAUD1TTIN7 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1REG21011 and PIC1REG21010 bits (TSG31PTE signal).
2	PIC1REG21002	Selects the signal to be input as TAUD1TTIN6 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1REG21011 and PIC1REG21010 bits (TSG31PTE signal).
1	PIC1REG21001	Selects the signal to be input as TAUD1TTIN5 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1REG21009 and PIC1REG21008 bits (TSG31PTE signal).
0	PIC1REG21000	Selects the signal to be input as TAUD1TTIN4 signal of TAUD1. 0: TIN pin input 1: Input signal selected by PIC1REG21009 and PIC1REG21008 bits (TSG31PTE signal).

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used.

41.2.2.20 PIC1REG2n1 — Timer Input/Output Control Register 2n1 (n = 0, 1)

The PIC1REG2n1 register selects the logical operation for the combination circuit PFN0xx.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $C4_H$(n = 0), <PIC1_base> + $D8_H$(n = 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC1REG2n1 27	PIC1REG2n1 26	PIC1REG2n1 25	PIC1REG2n1 24	PIC1REG2n1 23	PIC1REG2n1 22	PIC1REG2n1 21	PIC1REG2n1 20	PIC1REG2n1 19	PIC1REG2n1 18	PIC1REG2n1 17	PIC1REG2n1 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 41.30 PIC1REG2n1 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27, 26	PIC1REG2n1 [27:26]	Selects PFN045 WO2 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
25, 24	PIC1REG2n1 [25:24]	Selects PFN045 WO1 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
23, 22	PIC1REG2n1 [23:22]	Selects PFCN023 VO2 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
21, 20	PIC1REG2n1 [21:20]	Selects PFN023 VO1 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
19, 18	PIC1REG2n1 [19:18]	Selects PFN001 UO2 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
17, 16	PIC1REG2n1 [17:16]	Selects PFN001 UO1 output.*1 10: Combination circuit output. 11: Inverted combination circuit output. Settings other than above are prohibited.*2
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. The register value for some functions needs to be set depending on the value of TAUD. For the setting values, see **Section 41.2.3, Function**.

Note 2. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used.

Block diagram of PFN001 is shown in the following figure.

PFN023 and PFN045 are operated under the same logic with different input signals and select registers.

For connection of PFN0xx with peripheral circuits, see **Figure 41.14**.

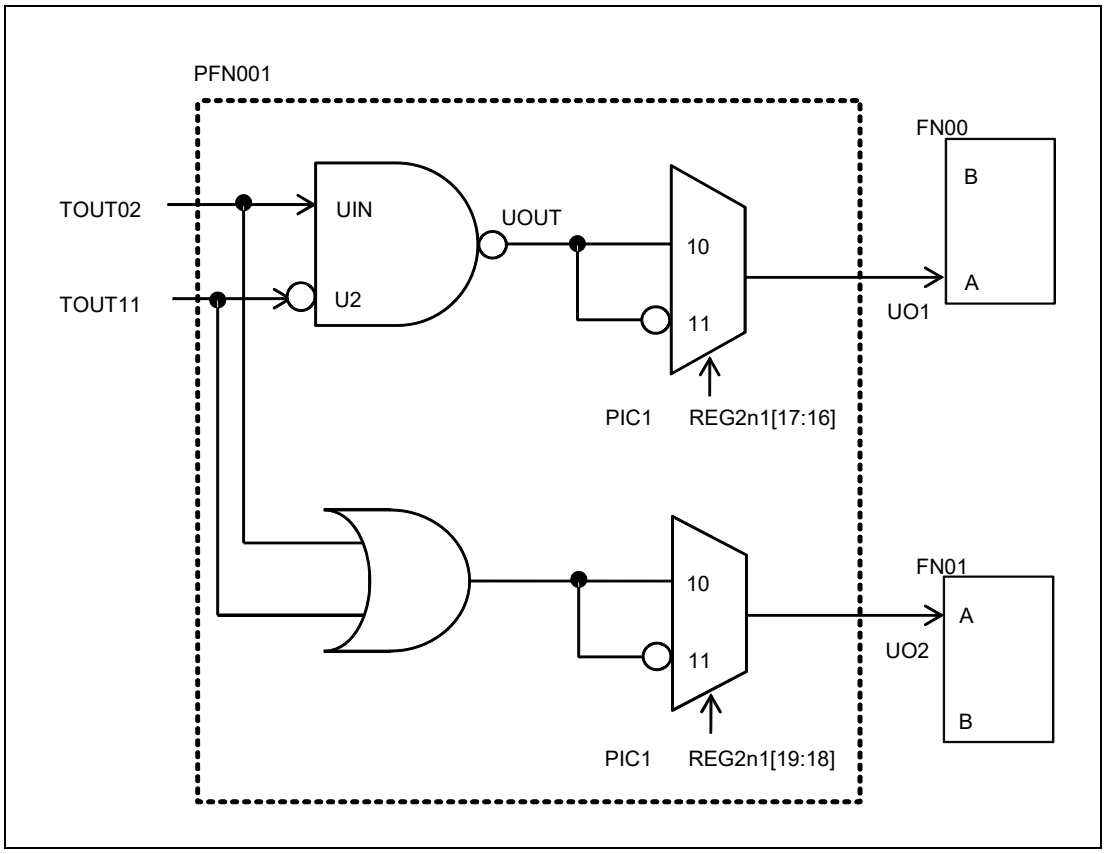


Figure 41.2 Block Diagram of PFN001

41.2.2.21 PIC1REG2n2 — Timer Input/Output Control Register 2n2 (n = 0, 1)

The PIC1REG2n2 register selects input signals of TAUDn CHm.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $C8_H$ (n = 0), <PIC1_base> + DC_H (n = 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC1REG2n227	PIC1REG2n226	PIC1REG2n225	PIC1REG2n224	PIC1REG2n223	PIC1REG2n222	PIC1REG2n221	PIC1REG2n220	PIC1REG2n219	PIC1REG2n218	PIC1REG2n217	PIC1REG2n216
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC1REG2n204	PIC1REG2n203	PIC1REG2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 41.31 PIC1REG2n2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27, 26	PIC1REG2n2 [27:26]	Selects the TIN input signal of TAUDnCH15. 00: TIN pin input. 10: Signal selected by PIC1REG2n204 bit (TOUT of TAUDnCH09). Settings other than above are prohibited.
25, 24	PIC1REG2n2 [25:24]	Selects the TIN input signal of TAUDnCH14. 00: TIN pin input 10: Signal selected by PIC1REG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited.
23, 22	PIC1REG2n2 [23:22]	Selects the TIN input signal of TAUDnCH13. 00: TIN pin input 10: Signal selected by PIC1REG2n203 bit (TOUT of TAUDnCH07). Settings other than above are prohibited.
21, 20	PIC1REG2n2 [21:20]	Selects the TIN input signal of TAUDnCH12. 00: TIN pin input 10: Signal selected by PIC1REG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited.
19, 18	PIC1REG2n2 [19:18]	Selects the TIN input signal of TAUDnCH11. 00: TIN pin input 10: Signal selected by PIC1REG2n202 bit (TOUT of TAUDnCH05). Settings other than above are prohibited.
17, 16	PIC1REG2n2 [17:16]	Selects the TIN input signal of TAUDnCH10. 00: TIN pin input 10: Signal selected by PIC1REG2n018 register bit (TOUT of TAUDnCH02). Settings other than above are prohibited
15 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	PIC1REG2n2 04	Selects the signal to be supplied to TIN of TAUDnCH15. 0: TOUT of TAUDnCH09. 1: Set/clear output by INTTAUDnI8 and INTTAUDnI9.
3	PIC1REG2n2 03	Selects the signal to be supplied to TIN of TAUDnCH13. 0: TOUT of TAUDnCH07. 1: Set/clear output by INTTAUDnI6 and INTTAUDnI7.

Table 41.31 PIC1REG2n2 Register Contents (2/2)

Bit Position	Bit Name	Function
2	PIC1REG2n202	Selects the signal to be supplied to TIN of TAUDnCH11. 0: TOUT of TAUDnCH05. 1: Set/clear output by INTTAUDnI4 and INTTAUDnI5.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

41.2.2.22 PIC1REG2n3 — Timer Input/Output Control Register 2n3 (n = 0, 1)

The PIC1REG2n3 register selects logical operation for the combination circuit FN0i.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + CC_H (n = 0), <PIC1_base> + $E0_H$ (n = 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC1REG2n322	PIC1REG2n321	PIC1REG2n320	—	PIC1REG2n318	PIC1REG2n317	PIC1REG2n316
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC1REG2n314	PIC1REG2n313	PIC1REG2n312	—	PIC1REG2n310	PIC1REG2n309	PIC1REG2n308	—	PIC1REG2n306	PIC1REG2n305	PIC1REG2n304	—	PIC1REG2n302	PIC1REG2n301	PIC1REG2n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 41.32 PIC1REG2n3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	PIC1REG2n3 [22:20]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.
19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	PIC1REG2n3 [18:16]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 12	PIC1REG2n3 [14:12]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 41.32 PIC1REG2n3 Register Contents (2/2)

Bit Position	Bit Name	Function
10 to 8	PIC1REG2n3 [10:08]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	PIC1REG2n3 [06:04]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PIC1REG2n3 [02:00]	Selects the logical operation to be performed on input signals A and B.*1 000: A 100: A and B 101: A or B Settings other than above are prohibited.

Note 1. The register value for some functions needs to be set according to the value of TAUD. For the setting values, see **Section 41.2.3, Function**.

Block diagram of FN00 is shown in the following figure.

FN01 to FN05 are operated under the same logic with different input signals and select registers.

For connection of FN0i with peripheral circuits, see **Figure 41.14**.

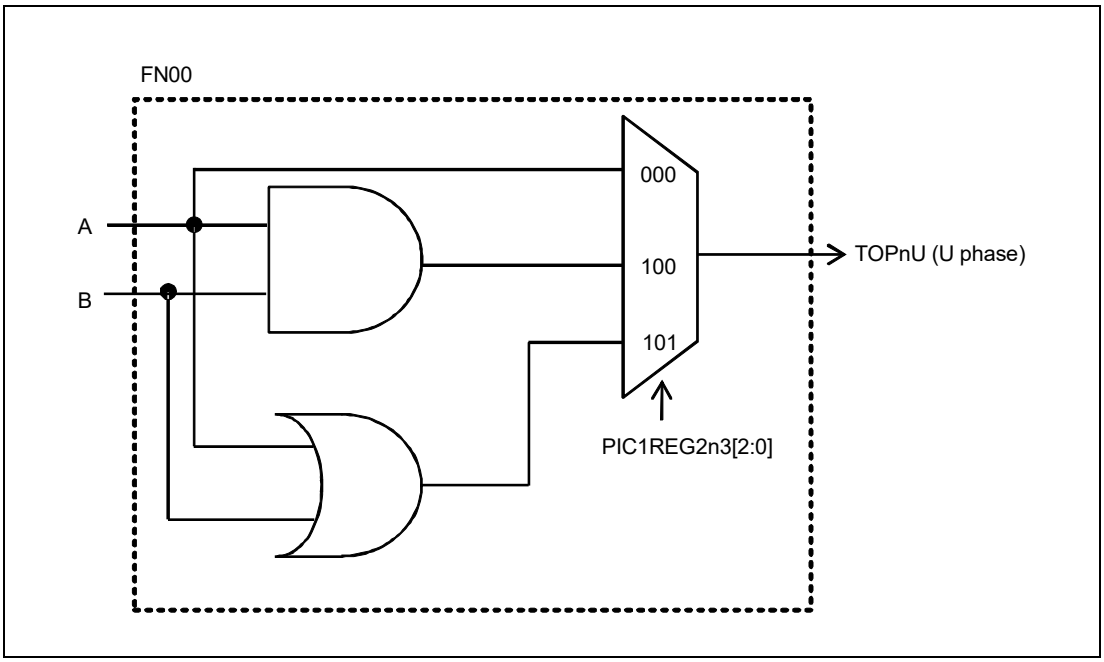


Figure 41.3 Block Diagram of FN00

41.2.2.23 PIC1REG30 — Timer Input/Output Control Register 30

The PIC1REG30 register selects ENCAn input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + $E8_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC1REG3022	PIC1REG3021	PIC1REG3020	PIC1REG3019	PIC1REG3018	PIC1REG3017	PIC1REG3016
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1REG3015	PIC1REG3014	PIC1REG3013	PIC1REG3012	PIC1REG3011	PIC1REG3010	PIC1REG3009	PIC1REG3008	PIC1REG3007	PIC1REG3006	PIC1REG3005	PIC1REG3004	PIC1REG3003	PIC1REG3002	PIC1REG3001	PIC1REG3000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.33 PIC1REG30 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22	PIC1REG3022	Selects the input pins (ENCA0E0, ENCA0E1, ENCA0EC internal inputs) of ENCA0 timer. 0: The signal selected by PIC1REG3000 (ENCA0E0), PIC1REG3001 (ENCA0E1), and PIC1REG3017-16 (ENCA0EC) 1: The signal selected by PIC1REG3020 and PIC1REG3019
21	PIC1REG3021	Selects the signal to be supplied to PIC1REG3012-15. 0: ENCA1I1 (signal 1 of the ENCA1 external pin) 1: The signal selected by the PIC1ENCSEL4107 bit of the PIC1ENCSEL410 register.
20,19	PIC1REG30 [20:19]	Selects the input pins (ENCA1E0, ENCA1E1, ENCA1EC) of ENCA1 timer. 00: The ENCA1E0, ENCA1E1, ENCA1EC input pins of ENCA1 timer. Settings other than above are prohibited.
18	PIC1REG3018	Selects the signal to be supplied to PIC1REG3002 to 05. 0: ENCA0I1 (signal 1 of ENCA0 external pin) 1: The signal selected by the PIC1ENCSEL4007 bit of the PIC1ENCSEL400 register.
17, 16	PIC1REG30 [17:16]	Selects the input pins (ENCA0E0, ENCA0E1, ENCA0EC) of ENCA0 timer. 00: The ENCA0E0, ENCA0E1, ENCA0EC input pins of ENCA0 timer. Settings other than above are prohibited.
15 to 12	PIC1REG30 [15:12]	Selects the signal for input as the ENCA1TTIN1 signal. 0000_B: The signal selected by PIC1REG3021. 0001_B: The signal selected by PIC1REG3018. 0010_B: ADCJTTOUT0[4] 0011_B: ADCJTTOUT0[3] 0100_B: ADCJTTOUT0[2] 0101_B: ADCJTTOUT0[1] 0110_B: ADCJTTOUT0[0] 0111_B: ADCJTTOUT1[4] 1000_B: ADCJTTOUT1[3] 1001_B: ADCJTTOUT1[2] 1010_B: ADCJTTOUT1[1] 1011_B: ADCJTTOUT1[0] Settings other than above are prohibited.

Table 41.33 PIC1REG30 Register Contents (2/2)

Bit Position	Bit Name	Function
11, 10	PIC1REG30 [11:10]	Selects the ENCA1EC pin input of timer ENCA1. 00: The signal selected by the PIC1REG3019 and PIC1REG3020 bits. 10: The signal selected by the PIC1REG3016 and PIC1REG3017 bits. 11: ENCAT0EQ1 (ENCA0 timer) or INTTPTMU01 signal Settings other than above are prohibited
9, 8	PIC1REG30 [09:08]	Selects the ENCA1E1 pin input of timer ENCA1. 00: Signal selected by the PIC1REG3019 and PIC1REG3020 bits. 01: The signal selected by the PIC1REG3016 and PIC1REG3017 bits. 10: TSG31PUD signal of TSG31. Settings other than above are prohibited.
7, 6	PIC1REG30 [07:06]	Selects the ENCA1E0 pin input of timer ENCA1. 00: The signal selected by the PIC1REG3019 and PIC1REG3020 bits. 01: The signal selected by the PIC1REG3016 and PIC1REG3017 bits. 10: TSG31PEC signal of TSG31. Settings other than above are prohibited
5 to 2	PIC1REG30 [05:02]	Selects the signal to be input as the ENCA0TTIN1 signal. 0000_B: The signal selected by the PIC1REG3018 bit. 0001_B: The signal selected by the PIC1REG3021 bit. 0010_B: ADCJTTOUT0[4] 0011_B: ADCJTTOUT0[3] 0100_B: ADCJTTOUT0[2] 0101_B: ADCJTTOUT0[1] 0110_B: ADCJTTOUT0[0] 0111_B: ADCJTTOUT1[4] 1000_B: ADCJTTOUT1[3] 1001_B: ADCJTTOUT1[2] 1010_B: ADCJTTOUT1[1] 1011_B: ADCJTTOUT1[0] Settings other than above are prohibited.
1	PIC1REG3001	Selects the signal to input as the ENCA0E1 internal signal. 0: Signal selected by PIC1REG3017 and PIC1REG3016. 1: TSG30PUD signal of TSG30.
0	PIC1REG3000	Selects the signal to input as the ENCA0E0 internal signal. 0: Signal selected by PIC1REG3017 and PIC1REG3016. 1: TSG30PEC signal of TSG30.

41.2.2.24 PIC1REG31 — Timer Input/Output Control Register 31

The PIC1REG31 register selects TAUD0, TAUD1 and TAUJ0 input signals.

Access: This register can be read or written in 32-bit units.

Address: <PIC1_base> + EC_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC1REG3122	PIC1REG3121	PIC1REG3120	PIC1REG3119	PIC1REG3118	PIC1REG3117	PIC1REG3116
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC1REG3115	—	PIC1REG3113	PIC1REG3112	PIC1REG3111	PIC1REG3110	PIC1REG3109	PIC1REG3108	PIC1REG3107	PIC1REG3106	—	PIC1REG3104	PIC1REG3103	—	PIC1REG3101	PIC1REG3100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R/W	R/W

Table 41.34 PIC1REG31 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22, 21	PIC1REG31 [22:21]	Selects the TIN input signal of TAUD1CH02. 00: TIN pin input 01: DT output signal of ENCAT1EQ0 or INTTPTMU11 Settings other than above are prohibited.
20	PIC1REG3120	Selects the TIN input signal of TAUD1CH01. 0: TIN pin input 1: The signal selected by PIC1REG3115 to PIC1REG3117
19, 18	PIC1REG31 [19:18]	Selects the TIN input signal of TAUD1CH00. 00: The signal selected by PIC1REG3115 to PIC1REG3117 10: DT output signal of ENCAT1EQ0 or INTTPTMU11 Settings other than above are prohibited.
17 to 15	PIC1REG31 [17:15]	Selects the TIN input signal of TAUD1CH00 and TAUD1CH01. 000: TIN pin input 001: DT output signal of ENCAT1EQ1 or INTTPTMU20 Settings other than above are prohibited.
14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	PIC1REG31 [13:12]	Selects the TIN input signal of TAUD0CH02. 00: TIN pin input 01: DT output signal of ENCAT0EQ0 or INTTPTMU00 Settings other than above are prohibited.
11	PIC1REG3111	Selects the TIN input signal of TAUD0CH01. 0: TIN pin input 1: The signal selected by PIC1REG3106 to PIC1REG3108
10, 9	PIC1REG31 [10:09]	Selects the TIN input signal of TAUD0CH00. 00: The signal selected by PIC1REG3106 to PIC1REG3108 10: DT output signal of ENCAT0EQ0 or INTTPTMU00 Settings other than above are prohibited.
8 to 6	PIC1REG31 [08:06]	Selects the TIN input signal of TAUD0CH00 and TAUD0CH01. 000: TIN pin input 001: DT output signal of ENCAT0EQ1 or INTTPTMU01 Settings other than above are prohibited.

Table 41.34 PIC1REG31 Register Contents (2/2)

Bit Position	Bit Name	Function
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	PIC1REG3104	Selects the TIN input signal of TAUJ0CH03. 0: TIN pin input 1: DT output signal of INTENCA1IEC or INTTPTMU30
3	PIC1REG3103	Selects the TIN input signal of TAUJ0CH02. 0: TIN pin input 1: DT output signal of INTENCA1IEC or INTTPTMU30
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PIC1REG3101	Selects the TIN input signal of TAUJ0CH01. 0: TIN pin input 1: DT output signal of INTENCA0IEC or INTTPTMU10
0	PIC1REG3100	Selects the TIN input signal of TAUJ0CH00. 0: TIN pin input 1: DT output signal of INTENCA0IEC or INTTPTMU10

41.2.2.25 PIC1REG50 — Timer Input/Output Control Register 50

The PIC1REG50 register selects TSG30 input signals.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + $F8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PIC1REG5010	—	PIC1REG5008	PIC1REG5007	PIC1REG5006	PIC1REG5005	—	—	—	—	PIC1REG5000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 41.35 PIC1REG50 Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	PIC1REG5010	Selects the signal to be input as the TSG30OPCI0 signal of the TSG30 timer. 0: INTENCA1I1 input of the ENCA1 timer 1: Setting prohibited.
9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PIC1REG5008	Selects the signal to be input as the TSG30OPCI0 signal of the TSG30 timer. 0: INTENCA0I1 input of the ENCA0 timer 1: Setting prohibited.
7	PIC1REG5007	Selects the signal to be input as the TSG30OPCI1 signal of the TSG30 timer. 0: INTTAUD0I7 input of the TAUD0 1: Setting prohibited.
6, 5	PIC1REG50 [06:05]	Selects the signal to be input as the TSG30OPCI0 signal of the TSG30 timer. 01: The signal selected by the PIC1REG5008 bit 10: The signal selected by the PIC1REG5010 bit 11: INTTAUD0I5 output signal of TAUD0. Settings other than above are prohibited.*1
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1REG5000	Switches the ENCA signal and external hall sensor signal. For the note on selection of the signals, see Note 1 of **Section 41.2.2.9, PIC1TSGHALLSEL — Hall Sensor Input Select Register**. 0: Select the pin input ENCA0E0, ENCA0E1, ENCA0EC. 1: Setting prohibited.

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used

41.2.2.26 PIC1REG51 — Timer Input/Output Control Register 51

The PIC1REG51 register selects TSG31 input signals.

Access: This register can be read or written in 16-bit units.

Address: <PIC1_base> + FC_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PIC1REG5110	—	PIC1REG5108	PIC1REG5107	PIC1REG5106	PIC1REG5105	—	—	—	—	PIC1REG5100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 41.36 PIC1REG51 Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	PIC1REG5110	Selects the signal to be input as the TSG31OPCI0 signal from the TSG31 timer. 0: INTENCA1I1 input signal of the ENCA1 timer 1: Setting prohibited.
9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PIC1REG5108	Selects the signal to be input as the TSG31OPCI0 signal from TSG31 timer. 0: INTENCA0I1 input signal of the ENCA0 timer 1: Setting prohibited.
7	PIC1REG5107	Selects the signal to be input as the TSG31OPCI1 signal from TSG31 timer. 0: The INTTAUD1I7 signal input of the TAUD1 timer. 1: Setting prohibited.
6, 5	PIC1REG51 [06:05]	Selects the signal to be input as the TSG31OPCI0 signal from TSG31 timer. 01: The signal selected by the PIC1REG5108 bit. 10: The signal selected by the PIC1REG5110 bit. 11: The INTTAUD1I5 output of the TAUD1. Settings other than above are prohibited.*1
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC1REG5100	Switches the ENCA signal and external hall sensor signal. For the note on selection of the signals, see Note 1 of **Section 41.2.2.9, PIC1TSGHALLSEL — Hall Sensor Input Select Register**. 1: Select the pin input ENCA1E0, ENCA1E1, ENCA1EC. Settings other than above are prohibited.*1

Note 1. Make sure to set an appropriate value when a selectable signal is used. Do not change the value from the value after reset when not used

41.2.2.27 PIC1POMONSEL — Port Output Monitor Select Register

The PIC1POMONSEL register selects the channels to monitor output of TAUD0, TAUD1, TSG30, TSG31, OSTM8, and OSTM9 timers using TAUD2.

Access: This register can be read or written in 8-bit units.

Address: FFBF B400$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	POMONSEL2	POMONSEL1	POMONSEL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 41.37 PIC1POMONSEL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	POMONSEL [2:0]	Selects the input pin to monitor. For the selectable input pins, see **Table 41.38**.

Table 41.38 Selection of Input Pins to be Monitored by the PIC1POMONSEL Register

PIC1POMONSEL setting	000	001	010	011	100
ch0	Port (TAUD2I0)	TAUD0 ch1	TAUD0 ch0	TAUD1 ch0	TSG30 ch1
ch1	Port (TAUD2I1)	TAUD0 ch3	TAUD0 ch1	TAUD1 ch1	TSG30 ch2
ch2	Port (TAUD2I2)	TAUD0 ch5	TAUD0 ch2	TAUD1 ch2	TSG30 ch3
ch3	Port (TAUD2I3)	TAUD0 ch7	TAUD0 ch3	TAUD1 ch3	TSG30 ch4
ch4	Port (TAUD2I4)	TAUD0 ch9	TAUD0 ch4	TAUD1 ch4	TSG30 ch5
ch5	Port (TAUD2I5)	TAUD0 ch11	TAUD0 ch5	TAUD1 ch5	TSG30 ch6
ch6	Port (TAUD2I6)	TAUD0 ch13	TAUD0 ch6	TAUD1 ch6	TSG31 ch1
ch7	Port (TAUD2I7)	TAUD0 ch15	TAUD0 ch7	TAUD1 ch7	TSG31 ch2
ch8	Port (TAUD2I8)	TAUD1 ch1	TAUD0 ch8	TAUD1 ch8	TSG31 ch3
ch9	Port (TAUD2I9)	TAUD1 ch3	TAUD0 ch9	TAUD1 ch9	TSG31 ch4
ch10	Port (TAUD2I10)	TAUD1 ch5	TAUD0 ch10	TAUD1 ch10	TSG31 ch5
ch11	Port (TAUD2I11)	TAUD1 ch7	TAUD0 ch11	TAUD1 ch11	TSG31 ch6
ch12	Port (TAUD2I12)	TAUD1 ch9	TAUD0 ch12	TAUD1 ch12	—
ch13	Port (TAUD2I13)	TAUD1 ch11	TAUD0 ch13	TAUD1 ch13	—
ch14	Port (TAUD2I14)	TAUD1 ch13	TAUD0 ch14	TAUD1 ch14	OSTM8
ch15	Port (TAUD2I15)	TAUD1 ch15	TAUD0 ch15	TAUD1 ch15	OSTM9

41.2.2.28 PIC1PIMONSEL — Port Input Monitor Select Register

The PIC1PIMONSEL register selects the channels to monitor input of TAUJ0 and TAUJ1 timers using TAUJ2.

Access: This register can be read or written in 8-bit units.

Address: FFBF $B600_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIMONSEL1	PIMONSEL0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 41.39 PIC1PIMONSEL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PIMONSEL[1:0]	Selects the input pin to monitor. For the selectable input pins, see **Table 41.40**.

Table 41.40 Selection of Input Pins to be Monitored by the PIC1PIMONSEL Register

PIMONSEL Setting	00	01	10
ch0	Port (TAUJ2I0)	TAUJ0 ch0	TAUJ1 ch0
ch1	Port (TAUJ2I1)	TAUJ0 ch1	TAUJ1 ch1
ch2	Port (TAUJ2I2)	TAUJ0 ch2	TAUJ1 ch2
ch3	Port (TAUJ2I3)	TAUJ0 ch3	TAUJ1 ch3

41.2.2.29 PIC1SELBSSER — Synchronous Start and Clear Enable Register

The PIC1SELBSSER register selects TAUD0 channels 14 and 15 for synchronous starting or clearing of TSG30 and TSG31.

Access: This register can be read or written in 8-bit units.

Address: FFBF $B200_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SELBSSER[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 41.41 PIC1SELBSSER Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	SELBSSER[2:0]	The SELBSSER[2:0] bits enables synchronous starting or clearing of TSG3 with TAUD0.

SELBSSER[2:0]	Setting for TSG3 Synchronous Starting or Clearing
000_B	TSG30: Disabled TSG31: Disabled
001_B	TSG30: INTTAUD0I14 enabled TSG31: Disabled
010_B	TSG30: Disabled TSG31: INTTAUD0I15 enabled
011_B	TSG30: INTTAUD0I14 enabled TSG31: INTTAUD0I15 enabled
100_B	TSG30: INTTAUD0I15 enabled TSG31: INTTAUD0I15 enabled
Other than above	Setting prohibited

CAUTIONS

1. Using the synchronous clearing and simultaneous starting trigger functions for TSG3 at the same time is prohibited. If you are using synchronous clearing of TSG3, only use the simultaneous start trigger function at the start of counting. After counting has started, set the bits of the simultaneous start control register listed below to 0.
   - PIC1SSER0.PIC1SSER014 (TAUD0 channel 14)
   - PIC1SSER0.PIC1SSER015 (TAUD0 channel 15)
   - PIC1SSER2.PIC1SSER208 (TSG30)
   - PIC1SSER2.PIC1SSER209 (TSG31)

   To start TSG30 and TSG31, and TAUD0 channels 14 and 15 simultaneously while synchronous clearing of TSG3 is in use, disable synchronous clearing of TSG3 before using the simultaneous start trigger function.

   Switching of the above setting while the trigger for synchronous clearing or simultaneous starting of TSG3 is active is prohibited.
2. Writing in the following ways to SSER during counter operation is prohibited.
   - Writing a value other than 000_B or the setting at the start of operation.
   - Writing at a timing that coincides with the occurrence of an interrupt for TAUD0 channel 14 or 15.

41.2.2.30 PIC1SELB_TAUD2I — TAUD2TTINm Input Signal Selection Register

The register selects the TAUD2TTINm input signals.

Access: This register can be read or written in 16-bit units.

Address: FFBF 6800_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SELB_TAUD2I15	SELB_TAUD2I14	SELB_TAUD2I13	SELB_TAUD2I12	SELB_TAUD2I11	SELB_TAUD2I10	SELB_TAUD2I9	SELB_TAUD2I8	SELB_TAUD2I7	SELB_TAUD2I6	SELB_TAUD2I5	SELB_TAUD2I4	SELB_TAUD2I3	SELB_TAUD2I2	SELB_TAUD2I1	SELB_TAUD2I0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.42 PIC1SELB_TAUD2I Register Contents

Bit Position	Bit Name	Function
15 to 0	SELB_TAUD2Im	Selection of TAUD2TTINm input signal

TAUD Input	Bit [m+1]	Bit [m]	Input signal
TAUD2TTIN[m]	x	0	Selection of port TAUD2I[m]
	x	1	Selection of port TAUD2I[m+1]
TAUD2TTIN[m+1]	0	x	Selection of port TAUD2I[m+1]
	1	x	Selection of port TAUD2I[m]

(m = 0, 2, 4, 6, 8, 10, 12, 14)

CAUTION

Do not change the input signal of each channel during the timer counting.

41.2.2.31 PIC1SELB_TAUJ2I — TAUJ2TTINm Input Signal Selection Register

This register selects the input signals to several TAUJ2 inputs.

Access: This register can be read or written in 8-bit units.

Address: FFE8 0800_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SELB_TAUJ2I1	SELB_TAUJ2I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 41.43 PIC1SELB_TAUJ2I Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SELB_TAUJ2I1	Select of TAUJ2TTIN3 input signal: 0: Port TAUJ2I3 1: TAUJ1TTOUT0
0	SELB_TAUJ2I0	Select of TAUJ2TTIN2 input signal: 0: Port TAUJ2I2 1: TAUJ1TTOUT0

41.2.2.32 PIC1SELB_TAUJ3I — TAUJ3TTINm Input Signal Selection Register

This register selects the TAUJ3 input signals.

Access: This register can be read or written in 8-bit units.

Address: FFE8 1800_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SELB_TAUJ3I3	SELB_TAUJ3I2	SELB_TAUJ3I1	SELB_TAUJ3I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 41.44 PIC1SELB_TAUJ3I Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SELB_TAUJ3I3	Select of TAUJ3TTIN3 input signal: 0: RTCA0OUT 1: TAUJ0TTOUT0
2	SELB_TAUJ3I2	Select of TAUJ3TTIN2 input signal: 0: RTCA0OUT 1: TAUJ0TTOUT0
1	SELB_TAUJ3I1	Select of TAUJ3TTIN3 input signal: 0: Port TAUJ3I3 1: Timer Input
0	SELB_TAUJ3I0	Select of TAUJ3TTIN2 input signal: 0: Port TAUJ3I2 1: Timer Input

41.2.3 Function

41.2.3.1 Simultaneous Start Trigger Function

(1) Overview

This function allows any combination of timers (TAUDn, TAUJn, TSG3n, TPBAn, OSTMn, and ENCAn) to be started simultaneously.

(2) Configuration

The timers which support simultaneous start trigger function are as follows.

- TAUDn
- TAUJn
- TSG3n
- TPBAn
- OSTMn
- ENCAn

NOTE

n = 8, 9 for OSTM; n = 0, 1 for other timers.

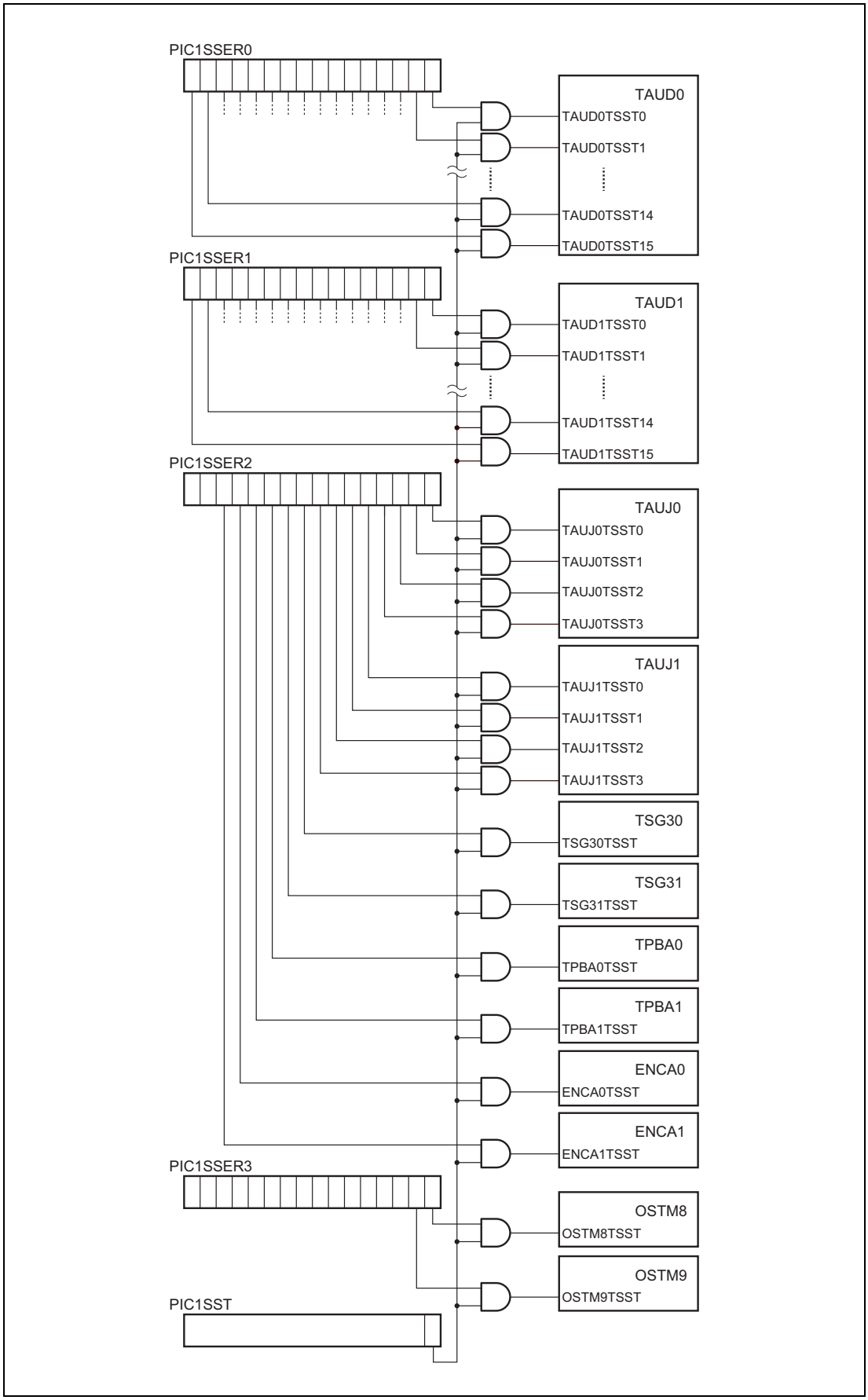


Figure 41.4 Block Diagram of Simultaneous Start Trigger Function

Set 1 to the PIC1SYNCTRG bit of the simultaneous start trigger control register (PIC1SST) after unmasking the target timers. The timer operations start by active signal input to the start trigger of each timer.

(3) Registers

The PIC1 registers set by this function are as follows. For setting value of the registers, see **Section 41.2.2.2** to **Section 41.2.2.6.**

- PIC1 registers to be set
  PIC1SST
  PIC1SSER0
  PIC1SSER1
  PIC1SSER2
  PIC1SSER3

(4) Function

This function allows any combination of timers (TAUDn, TAUJn, TSG3n, TPBAn, OSTMn, and ENCAn) to be started simultaneously.

(5) Flow Chart

The following figure shows the setting flow of this function.

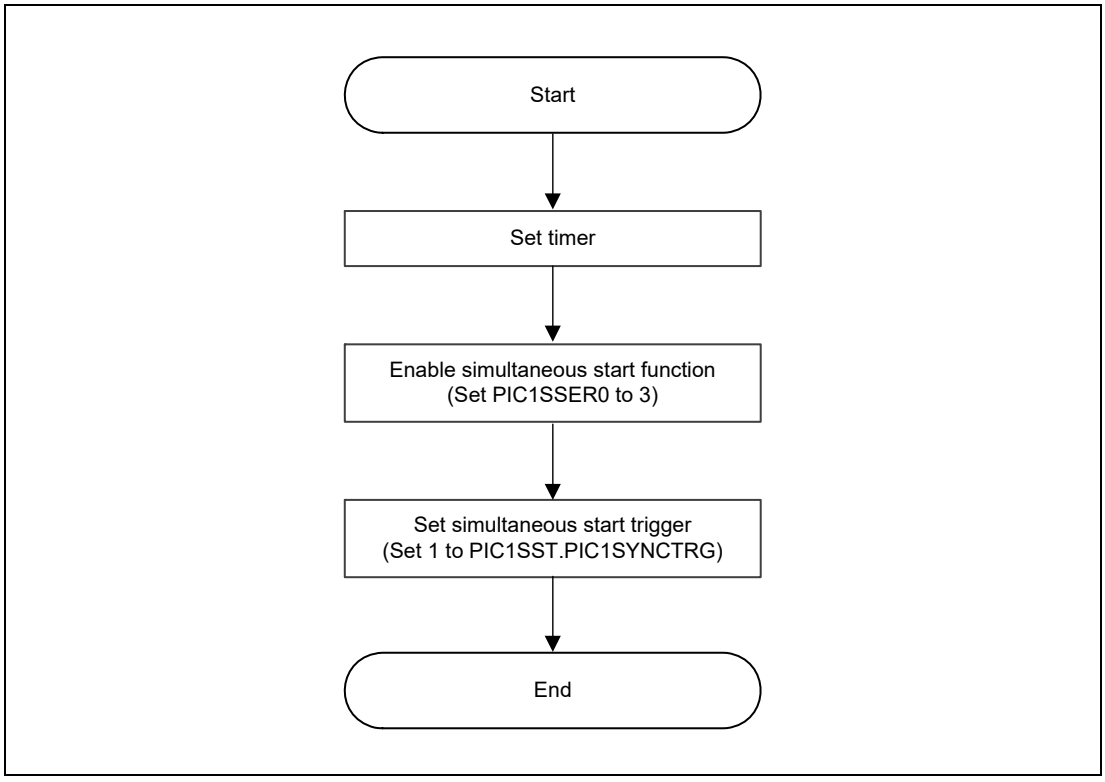


Figure 41.5 Setting Flow

Setting procedures are as follows.

- Set timer
  Set the timers to start simultaneously.
- Enable simultaneous start function
  Set 1 to the applicable bits of the PIC1SSER0, PIC1SSER1, PIC1SSER2, and PIC1SSER3 registers to enable simultaneous start of the corresponding timers.
- Set simultaneous start trigger
  Setting 1 to the PIC1SYNCTRG bit of the simultaneous start trigger control register (PIC1SST) simultaneously starts the timers.

41.2.3.2 INT Signal Output Selection Function

(1) Overview

Request signals for two types of interrupts, peak interrupts and valley interrupts, can be output by the INT signals output by TAUD0.

(2) Configuration

This function generates the peak interrupt INTTAPA0IPEK0 and valley interrupt INTTAPA0IVLY0 by using the TAPA0TSIM0 signal, which is connected to the INT signal on the TAUD's triangular carrier cycle generation channel (master) and TAPA0TUDCM0 signal, which is connected to the counter up/down signal.

For the connection destination of TAPA0TSIM0, see **Section 36.1.6, Internal Input/Output Signal**.

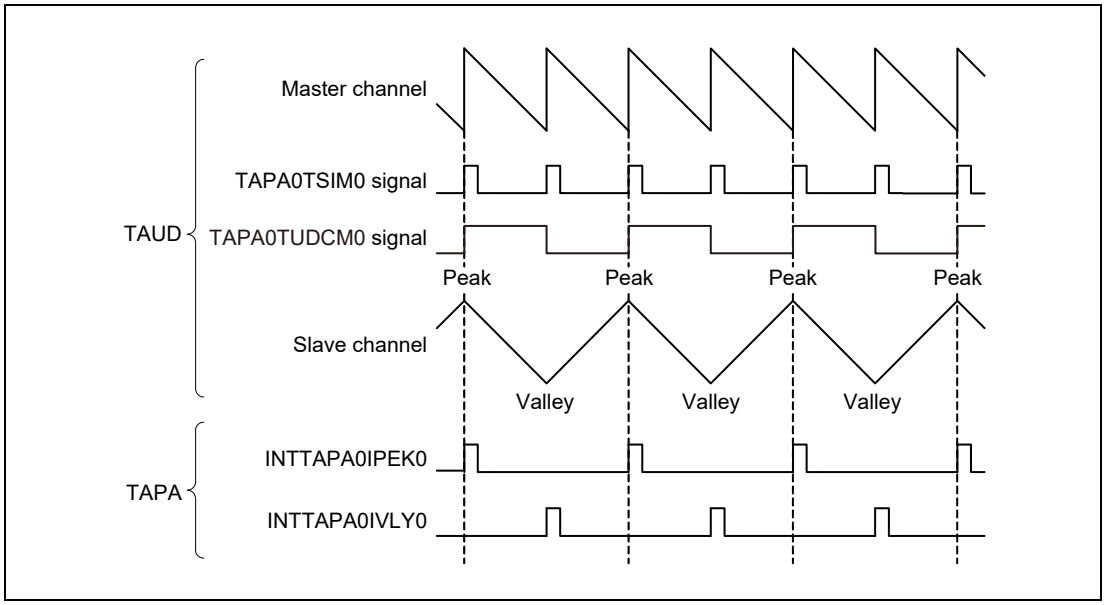


Figure 41.6 Basic Timing of Signals for the INT Signal Output Selection Function

Triangular carrier cycles are generated on the master channel.

The INT signal generated on the master channel in each half triangular carrier cycle is input to TAPA0 as the TAPA0TSIM0 signal. TAPA0 generates the INTTAPA0IPEK0 signal (peak interrupt) during high level of the TAPA0TUDCM0 signal and the INTTAPA0IVLY0 signal (valley interrupt) during low level of the TAPA0TUDCM0 signal by using the TAPA0TSIM0 and TAPA0TUDCM0 input signals.

CAUTION

The peak interrupt INTTAPA0IPEK0 and valley interrupt INTTAPA0IVLY0 are generated regardless of the function of the master channel of TAUD.

When not using these peak and valley interrupts, mask them by using the ICTAPA0IPEK0 and ICTAPA0IVLY0 registers, respectively.

(3) Block Diagram

TAUD0 and TAPA0 are connected in the registers shown below by the INT signal output selection function.

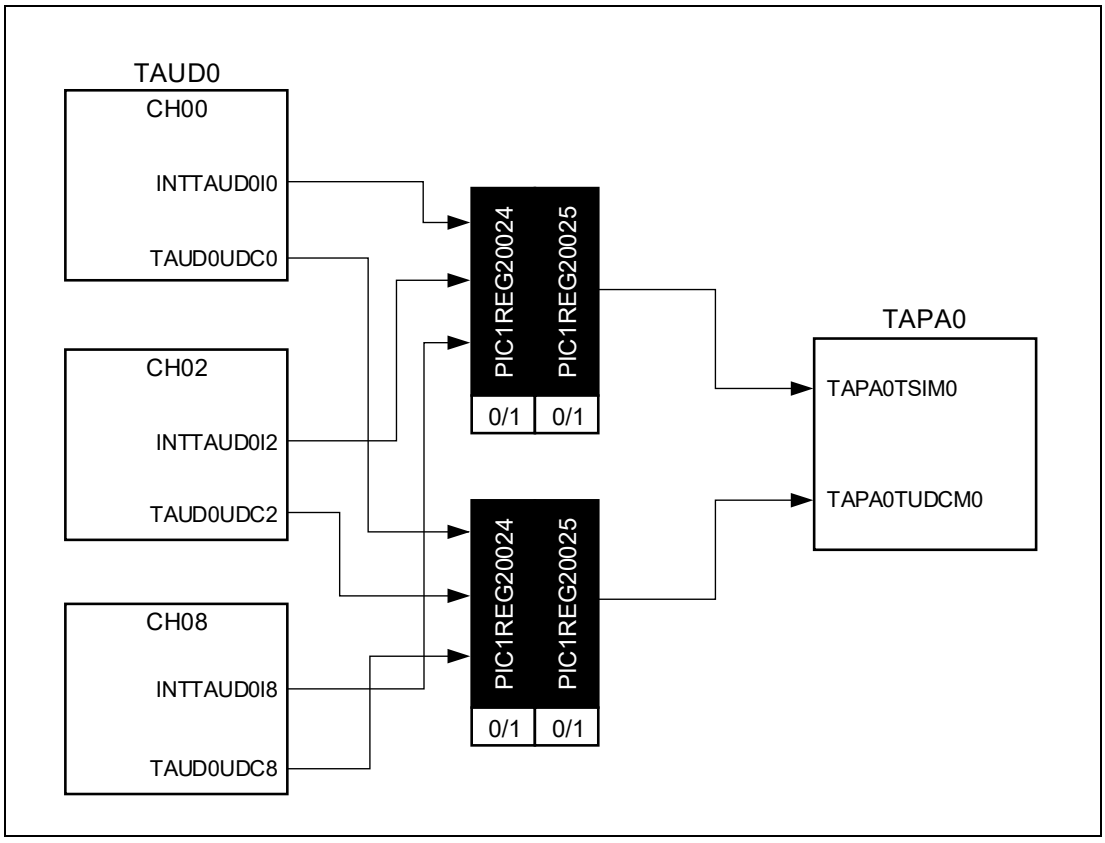


Figure 41.7 Connection of the INT Signals

41.2.3.3 PWM Output Function With Dead Time

(1) Overview

This function generates and outputs PWM waveforms with the dead time from one phase to three phases using TAUDn.

The PWM output function of TAUD sets only the clear timing in a period under the duty ratio specification. This function enables the set timing to be specified in addition to the clear timing to output more flexible PWM waveforms with the dead time.

The following table lists the number of TAUDn channels used.

PWM Output with Dead Time	Number of TAUDn Channels
One-phase PWM output (U phase/UB phase)	5 channels (master channel: 1, slave channel: 4)
Two-phase PWM output (U phase/UB phase, V phase/VB phase)	9 channels (master channel: 1, slave channel: 8)
Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase)	13 channels (master channel: 1, slave channel: 12)

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel.

TAUDn Channel	U phase / UB phase	V phase / VB phase	W phase / WB phase	Usage
CH0	—	—	—	Not used
CH1	—	—	—	Not used
CH2	√	√	√	Carrier period (common to each phase)
CH3	—	—	—	Not used
CH4	√	—	—	Duty (U phase setting)
CH5	√	—	—	Duty (U phase clearing)
CH6	—	√	—	Duty (V phase setting)
CH7	—	√	—	Duty (V phase clearing)
CH8	—	—	√	Duty (W phase setting)
CH9	—	—	√	Duty (W phase clearing)
CH10	√	—	—	U phase output (TOUT10)
CH11	√	—	—	UB phase output (TOUT11)
CH12	—	√	—	V phase output (TOUT12)
CH13	—	√	—	VB phase output (TOUT13)
CH14	—	—	√	W phase output (TOUT14)
CH15	—	—	√	WB phase output (TOUT15)

Note: √: Used; —: Not used

(2) Configuration

The PWM output function with the dead time is realized by using the PWM output function/one-phase output PWM function of TAUDn and PIC1 in combination. The following figure shows the block diagram of the PWM output function with the dead time.

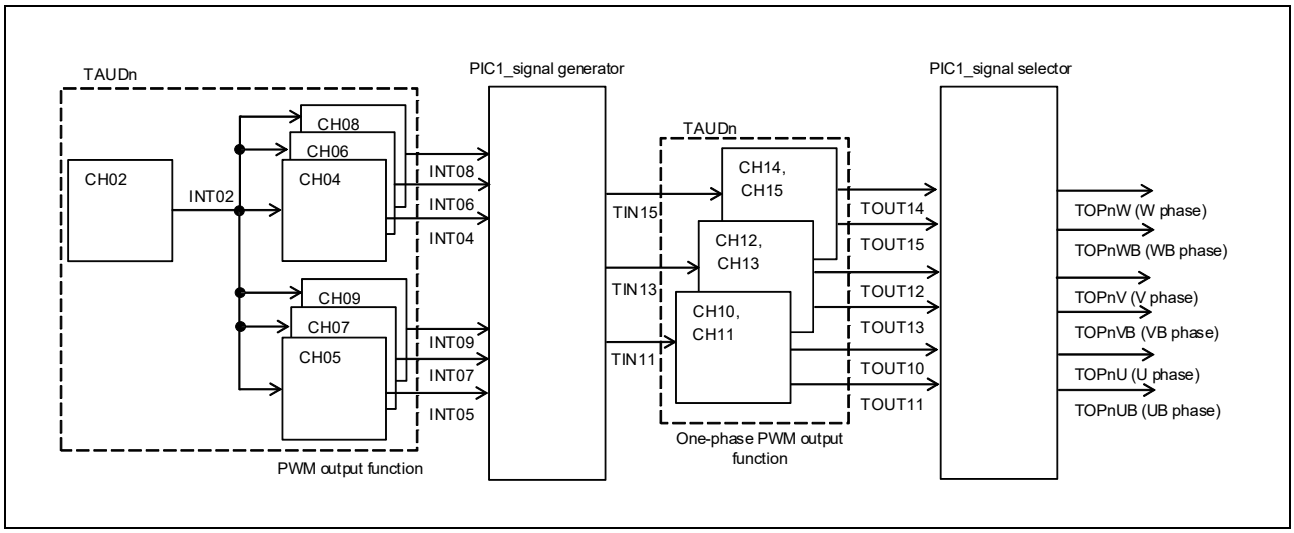


Figure 41.8 Block Diagram of PWM Output Function with Dead Time

The configuration of this function is described below using the PWM output of U phase/UB phase as an example.

- [TAUDn] PWM output function
  CH02, CH04, and CH05 are used in combination. Setting the period, U phase set value, and U phase clear value to CDR02, CDR04, and CDR05, respectively, generates the set and clear PWM signals (INT04 and INT05).
- [PIC1_signal generator] RS flip-flop circuit (RSn2)
  The INT04 and INT05 inputs are selected and TIN11 (PWM signal) is generated.
- [TAUDn] One-phase PWM output function
  CH10 and CH11 are used in combination. A dead time value is set to CDR11, dead time is inserted in the PWM signal to be input to TIN11, and TOUT10 (U phase PWM signal) and TOUT11 (UB phase PWM signal) are output.
- [PIC1_signal selector]
  TOUT10 and TOUT11 inputs are selected and output to TOPnU and TOPnUB pins, respectively.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

The block diagram of PIC1 is shown in the following figure.

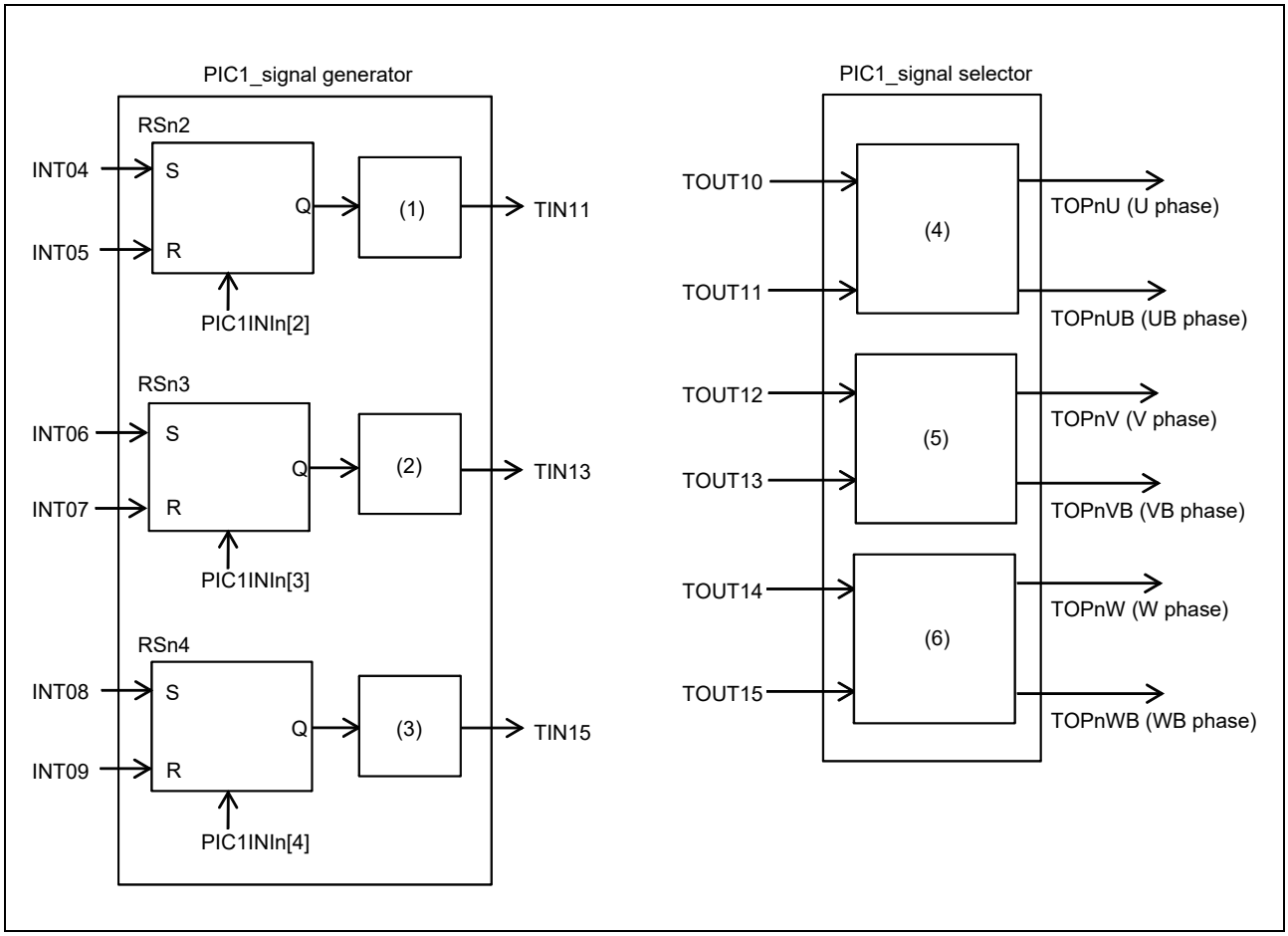


Figure 41.9 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

U phase/ UB phase

The values to output the signal Q from RSn2 as TIN11. (**Figure 41.9**, unit (1))

PIC1REG2n2[19:18] = 10_B

PIC1REG2n2[2] = 1_B

PIC1TAUDnSEL[23:22] = 00_B

The values to output TOUT10 and TOUT11 as TOPnU and TOPnUB, respectively (**Figure 41.9** , unit (4))

PIC1REG2n1[19:16] = 0000_B

PIC1REG2n3[2:0] = 000_B

PIC1REG2n3[6:4] = 000_B

V phase/ VB phase

The values to output the signal Q from RSn3 as TIN13. (**Figure 41.9**, unit (2))

PIC1REG2n2[23:22] = 10_B

PIC1REG2n2[3] = 1_B

PIC1TAUDnSEL[27:26] = 00_B

The values to output TOUT12 and TOUT13 as TOPnV and TOPnVB, respectively (**Figure 41.9**, unit (5))

PIC1REG2n1[23:20] = 0000_B

PIC1REG2n3[10:8] = 000_B

PIC1REG2n3[14:12] = 000_B

W phase/ WB phase

The values to output the signal Q from RSn4 as TIN15. (**Figure 41.9**, unit (3))

PIC1REG2n2[27:26] = 10_B

PIC1REG2n2[4] = 1_B

PIC1TAUDnSEL[31:30] = 00_B

The values to output TOUT14 and TOUT15 as TOPnW and TOPnWB, respectively (**Figure 41.9**, unit (6))

PIC1REG2n1[27:24] = 0000_B

PIC1REG2n3 [18:16] = 000_B

PIC1REG2n3[22:20] = 000_B

Enables initialization of RSn2 to RSn4

The values to enable initialization of RSn2 to RSn4

PIC1INIn0 [4] = 1_B (initialized)

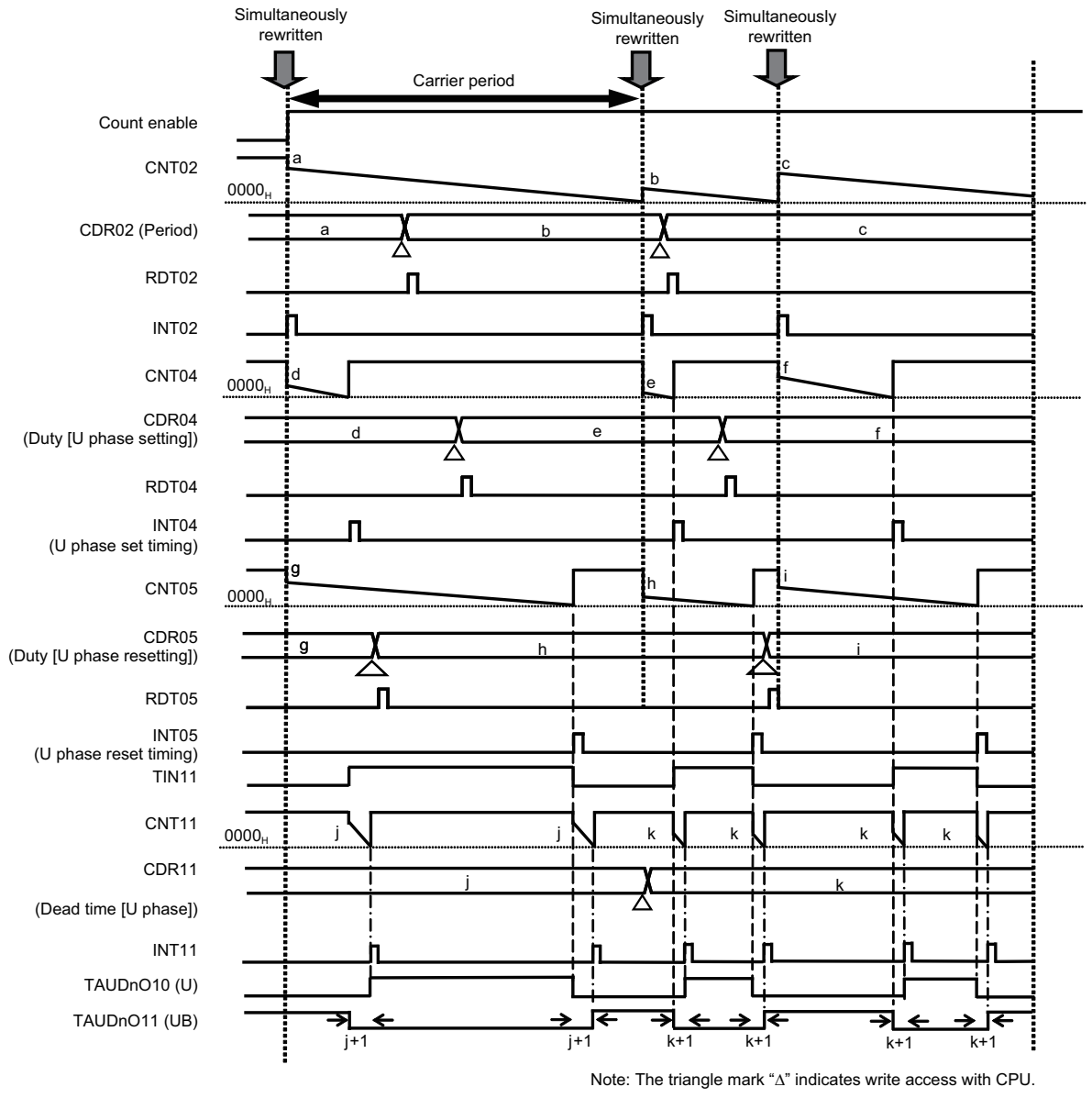
PIC1INIn0 [3] = 1_B (initialized)

PIC1INIn0 [2] = 1_B (initialized)

(4) Function

Details of this function are described using the one-phase PWM output (U phase/UB phase) with dead time as an example.

The following figure shows the timing diagram.



Note: For details of TAUD function including operation timing, see **Section 33, Timer Array Unit D (TAUD)**.

Figure 41.10 Timing Diagram of One-Phase PWM Output with Dead Time (U phase/ UB phase)

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.*1

(2) For CH04 and CH05, the values set to CDR04 and CDR05 are reloaded to CNT04 and CNT05, respectively, upon the occurrence of a CH02 underflow.

(3) INT04 is generated and TIN11 becomes high level upon the occurrence of a CH04 underflow, and INT05 is generated and TIN11 becomes low level upon the occurrence of a CH05 underflow. This process generates a PWM waveform.

(4) The set values are reloaded to CNT11 at both edges of TIN11.*2

(5) INT11 is generated and TAUDnO10 becomes high level upon the occurrence of a CH11 underflow, and INT05 is generated and TAUDnO10 becomes low level upon the occurrence of a CH05 underflow. This process generates a U-phase PWM waveform, which is output to TOPnU.

(6) TAUDnO11 becomes low level at the rising edge of TIN11, and INT11 is generated and TAUDnO11 becomes high level upon the occurrence of a CH11 underflow. This process generates a UB-phase PWM waveform, which is output to TOPnUB.

A similar process occurs for V phase/VB phase and W phase/WB phase.

Note 1. Select the count clock signal of the same clock for TAUDn.

Note 2. Set both edges (rising and falling) as the valid edge of TIN11 of TAUDn to be detected.

Details of this function when setting a clear timing longer than the carrier period are described below using V phase/VB phase as an example.

The following figure shows the timing diagram.

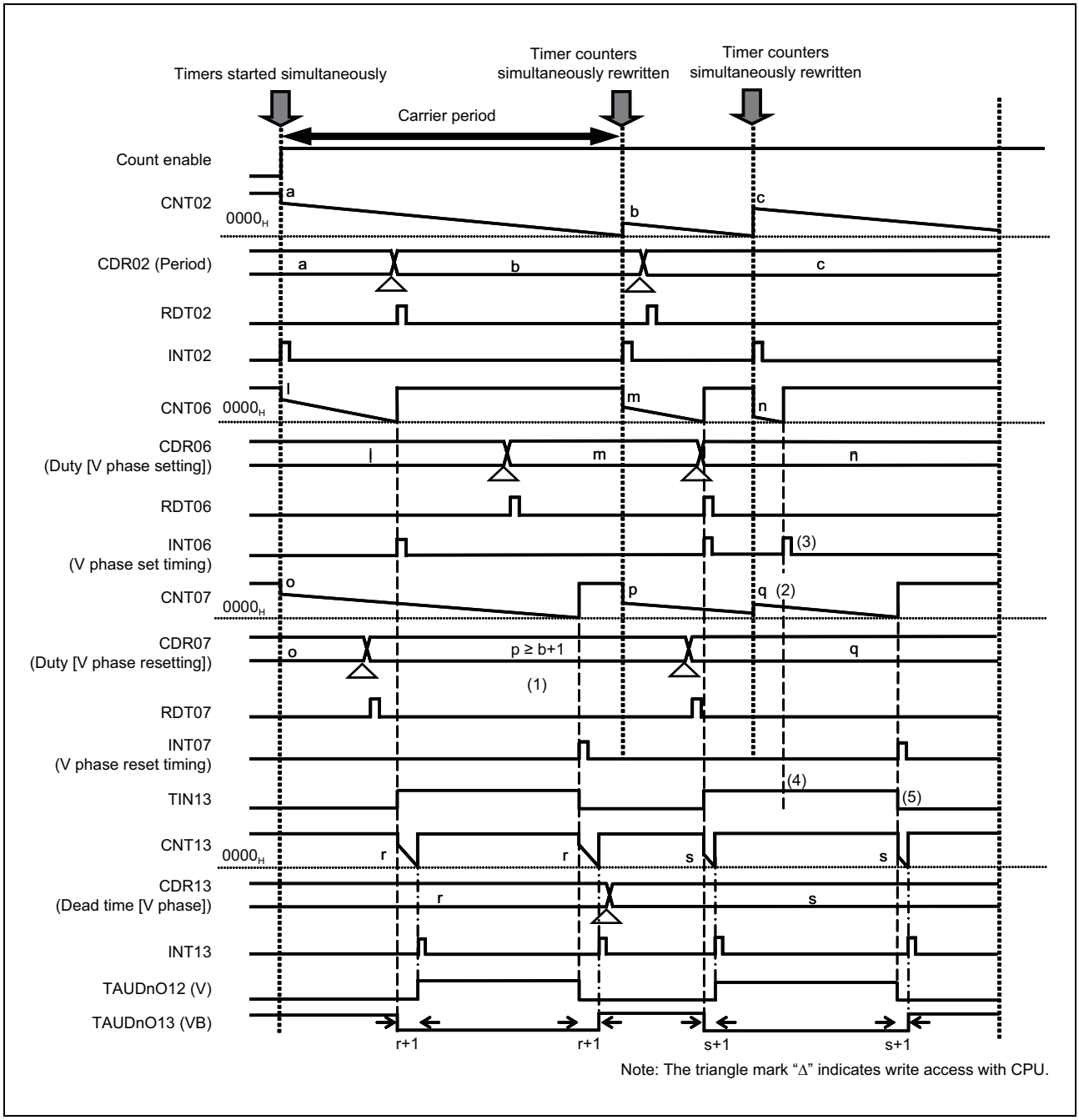


Figure 41.11 Clear Timing Value > Carrier Period Value (V phase/VB phase)

Setting a clear timing longer than the carrier period allows the waveform output to be extended over the carrier period.

An operation example of one-phase PWM output (V phase/VB phase) is shown below. The operation flow from timer operation start to one-phase PWM output by one-phase PWM output function, refer to the description for one-phase PWM output with dead time (U phase/UB phase).

When the value set in CH07 is larger than the value set in CH02 (**Figure 41.11** (1)), underflow of the carrier period timer is generated before generation of V phase clear timing signal (INT07), and the value is reloaded (**Figure 41.11** (2)).

As a result, the V phase clear timing signal (INT07) that should be generated is not generated, and instead the V phase set timing signal (INT06) is generated again (**Figure 41.11** (3)).

At this time, the PWM waveform is not affected because V phase set timing signal is ignored in the PIC circuit (**Figure 41.11** (4)). Therefore, the PWM waveform is output extended over the carrier period (**Figure 41.11** (5))

The following figure shows the timing diagram of three-phase PWM output with the dead time.

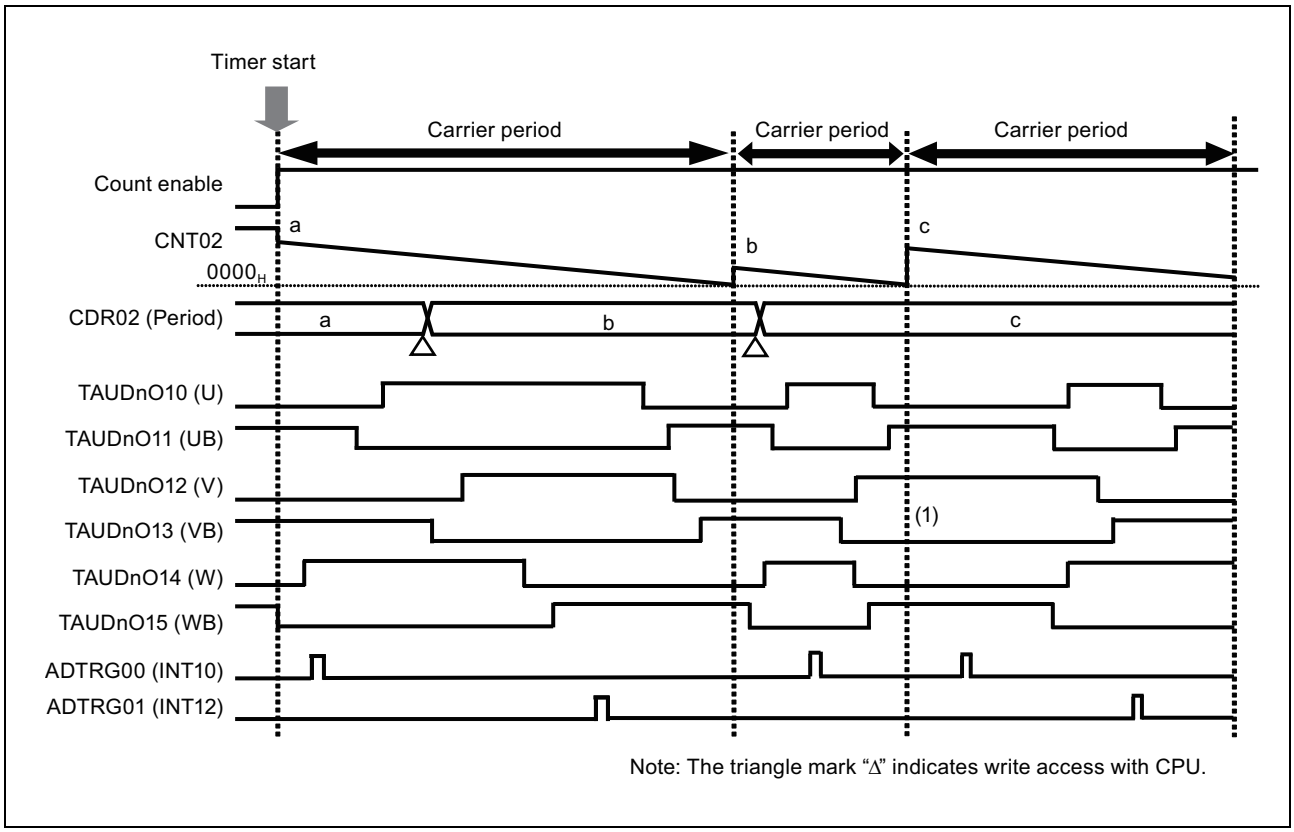
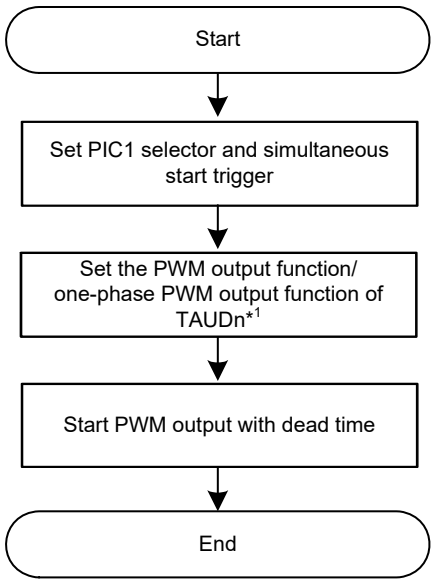


Figure 41.12 Timing Diagram of Three-Phase PWM Output with Dead Time

(5) Flow Chart

The following flow chart shows the PMW output function with the dead time.



Note 1. For the setting of PWM output function of TAUDn, see **Section 33.15.1, PWM Output Function**.

(6) Setting Examples for Operation Functions

This section provides example settings for each register.

TAUDn Settings (Active High Example)

Table 41.45 TAUDn: CH2-related (PWM Output Master Channel*1)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 41.46 TAUDn: CH4 to CH9-related (PWM Output Master Channel*1) (m = 4 to 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Any*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register used during PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Table 41.47 TAUDn: CH11, CH13, and CH15-related (One-phase PWM Output) (m = 11, 13, 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*1	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output). For details, see **Section 33, Timer Array Unit D (TAUD)**.

Table 41.48 Common TAUDn Channel Settings (1/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0	These are fixed to 0 because TOUT09 to TOUT04 are not used.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01, TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*1	Output a low-level signal to TOUT15 to TOUT10.
	9 to 4	TAUDnTO09 to TAUDnTO04	0	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01, TAUDnTO00	Don't care	
TAUDnTOM	15 to 4	TAUDnTOM15 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01, TAUDnTOM00	Don't care	
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	0	Synchronous operation mode 1
	3	TAUDnTOC03	Don't care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01, TAUDnTOC00	Don't care	
TAUDnTOL	15 to 4	TAUDnTOL15 to TAUDnTOL04	0*1	Positive logic output (active high)
	3	TAUDnTOL03	Don't care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01, TAUDnTOL00	Don't care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control.*2
	9 to 4	TAUDnTDE09 to TAUDnTDE04	0	Disable dead time control.
	3	TAUDnTDE03	Don't care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01, TAUDnTDE00	Don't care	

Table 41.48 Common TAUDn Channel Settings (2/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 4	TAUDnTDM09 to TAUDnTDM04	0	Invalid because dead time control is disabled.
	3	TAUDnTDM03	Don't care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01, TAUDnTDM00	Don't care	
TAUDnTDL	15	TAUDnTDL15	1*1	Dead time is in the negative segment of the W phase output
	14	TAUDnTDL14	0*1	Dead time is in the positive segment of the W phase output
	13	TAUDnTDL13	1*1	Dead time is in the negative segment of the V phase output
	12	TAUDnTDL12	0*1	Dead time is in the positive segment of the V phase output
	11	TAUDnTDL11	1*1	Dead time is in the negative segment of the U phase output
	10	TAUDnTDL10	0*1	Dead time is in the positive segment of the U phase output
	9 to 4	TAUDnTDL09 to TAUDnTDL04	0	Invalid because dead time control is disabled.
	3	TAUDnTDL03	Don't care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01, TAUDnTDL00	Don't care	
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Stop real-time output.
	3	TAUDnTRE03	Don't care	
	2	TAUDnTRE02	0	Stop real-time output.
	1, 0	TAUDnTRE01, TAUDnTRE00	Don't care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01, TAUDnTRO00	Don't care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01, TAUDnTRC00	Don't care	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01, TAUDnTME00	Don't care	

Table 41.48 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't care	
	9 to 4	TAUDnRDE09 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01, TAUDnRDE00	Don't care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 4	TAUDnRDS09 to TAUDnRDS04	0	Enable simultaneous rewriting by using a master channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01, TAUDnRDS00	Don't care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't care	
	9 to 4	TAUDnRDM09 to TAUDnRDM04	0	Load the signal when the master channel starts counting.
	3	TAUDnRDM03	Don't care	
	2	TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01, TAUDnRDM00	Don't care	

Table 41.48 Common TAUDn Channel Settings (4/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't care	
	9 to 4	TAUDnRDC09 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't care	
	2	TAUDnRDC02	1	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01, TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 33, Timer Array Unit D (TAUD)**.

PIC Settings

Table 41.49 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC1REG2n2	27	PIC1REG2n227	1	Select the input selected by the PIC1REG2n204 bit.
	23	PIC1REG2n223	1	Select the input selected by the PIC1REG2n203 bit.
	19	PIC1REG2n219	1	Select the input selected by the PIC1REG2n202 bit.
	4	PIC1REG2n204	1	Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
	3	PIC1REG2n203	1	Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
	2	PIC1REG2n202	1	Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.

41.2.3.4 High Accuracy Triangle Wave PWM Output Function with Dead Time

(1) Overview

This function generates triangle wave PWM output with dead time from one-phase to three-phases. Compared to the triangle wave PWM output function with dead time of TAUDn, this function enables a control of the variable dead time range, where duty cycle is close to 100% and 0%.

The TAUDn channels used in this function and the numbers of channels are listed in the following table.

High Accuracy Triangle Wave PWM Output with Dead Time	Number of TAUDn Channels
One-phase PWM output (U phase/UB phase)	5 channels (master channel: 1, slave channel: 4)
Two-phase PWM output (U phase/UB phase, V phase/VB phase)	9 channels (master channel: 1, slave channel: 8)
Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase)	13 channels (master channel: 1, slave channel: 12)

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel of CH3 to 9.

CHm is used as the master channel of CHm+1 (m = 10, 12, 14).

TAUDn Channel	U phase / UB phase	V phase / VB phase	W phase / WB phase	Usage
CH0	—	—	—	Not used
CH1	—	—	—	Not used
CH2	√	√	√	Carrier period (common to each phase)
CH3	—	—	—	Not used
CH4	√	—	—	Triangle PWM output with dead time (U phase/UB phase)
CH5	√	—	—	
CH6	—	√	—	Triangle PWM output with dead time (V phase/VB phase)
CH7	—	√	—	
CH8	—	—	√	Triangle PWM output with dead time (W phase/WB phase)
CH9	—	—	√	
CH10	√	—	—	Reduced dead time pulse (U phase/UB phase)
CH11	√	—	—	
CH12	—	√	—	Reduced dead time pulse (V phase/VB phase)
CH13	—	√	—	
CH14	—	—	√	Reduced dead time pulse (W phase/WB phase)
CH15	—	—	√	

Note: √: Used; —: Not used

(2) Configuration

The high accuracy triangle wave PWM output function with the dead time is realized by using the triangle wave PWM output function/one-shot pulse output function of TAUDn and PIC1 in combination. The following figure shows the block diagram of the high accuracy triangle wave PWM output function with the dead time.

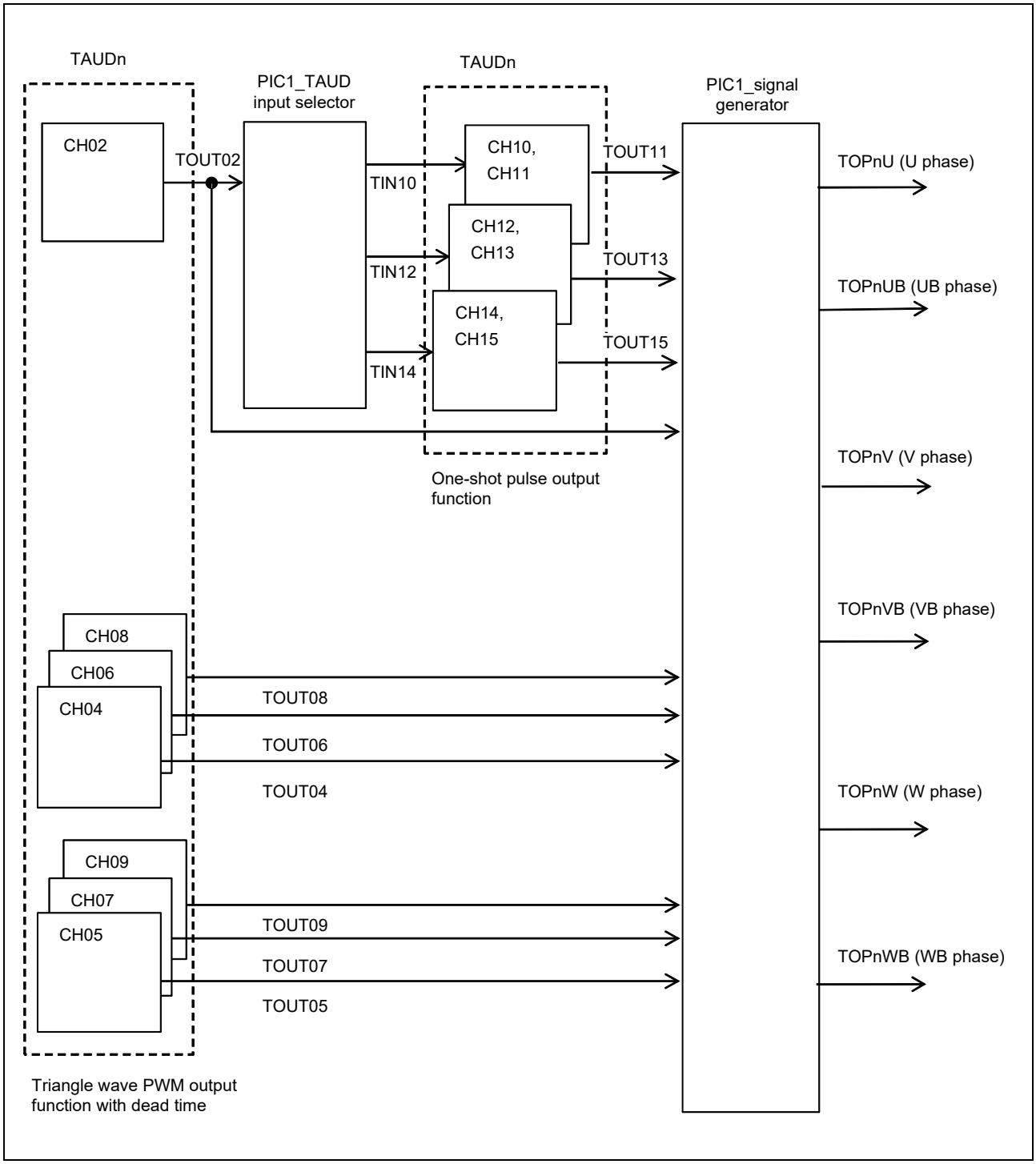


Figure 41.13 Block Diagram of High Accuracy Triangle Wave Three-Phase PWM Output With Dead Time

The configuration of this function is described below using the PWM output of U phase/UB phase as an example.

- [PIC1_TAUD input selector]
  TOUT02 is selected and output to TIN10.
- [TAUDn] One-shot pulse output function
  CH10 and CH11 are used in combination. The delay value and the pulse width are set to CDR10 and CDR11, respectively, and the one-shot pulse output signal (TOUT11) is generated.
- [TAUDn] Triangle wave PWM output function with dead time
  CH02, CH04, and CH05 are used in combination. The period, duty, and dead time are set to CDR02, CDR04, and CDR05, respectively, and a triangle wave PWM output signal with dead time (TOUT04 and TOUT05) is generated.
- [PIC1_signal generator]
  The reduced dead time pulses (UO1 and UO2) are generated at PFN001 from the one-shot pulse output signal.
  UO1 and UO2 are synthesized with TOUT04 and TOUT05 at FN00 and FN01, respectively, a dead time variable range pulse is added, and TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal) are generated.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

The following figure shows the block diagram of PIC1.

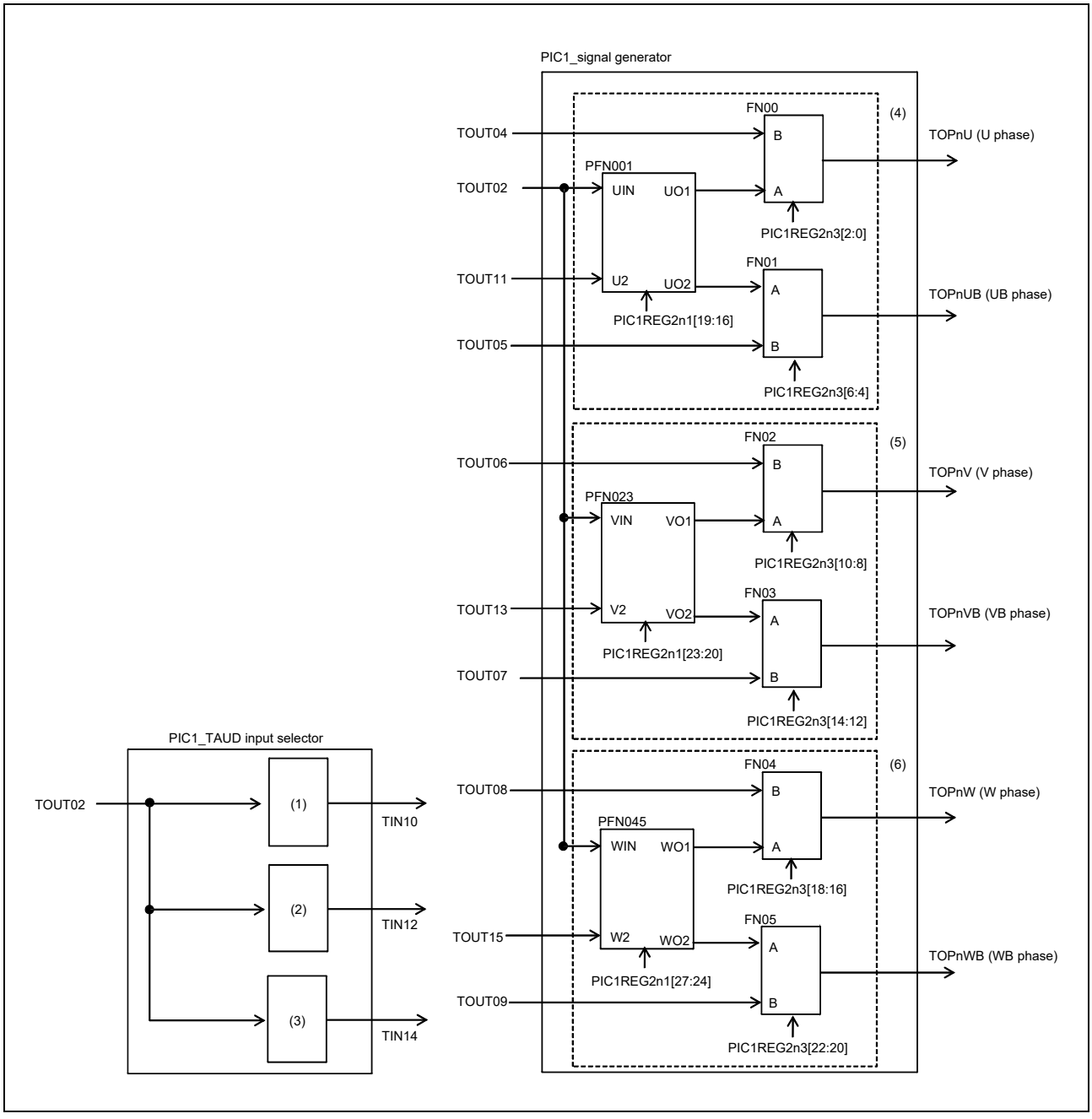


Figure 41.14 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

(1) PIC1_TAUD input selector (U phase/ UB phase)
The values to output TOUT02 as TIN10 (common to active high/low)
PIC1REG2n0[18] = 1_B
PIC1REG2n2[17:16] = 10_B
PIC1TAUDnSEL[21:20] = 00_B

(2) PIC1_TAUD input selector (V phase/ VB phase)
The values to output TOUT02 as TIN12 (common to active high/low)
PIC1REG2n0[18] = 1_B
PIC1REG2n2[21:20] = 10_B
PIC1TAUDnSEL[25:24] = 00_B

(3) PIC1_TAUD input selector (W phase/ WB phase)
The values to output TOUT02 as TIN14 (common to active high/low)
PIC1REG2n0[18] = 1_B
PIC1REG2n2[25:24] = 10_B
PIC1TAUDnSEL[29:28] = 00_B

(4) PIC1_ signal generator (U phase/ UB phase)
The values to output one-phase PWM (active high/low) from TAUDnO10 and TAUDnO11
PIC1REG2n1[19:16] = 1010_B (active high), 1111_B (active low)
PIC1REG2n3 [06:04] = 100_B (active high), 101_B (active low)
PIC1REG2n3[02:00] = 100_B (active high), 101_B (active low)

(5) PIC1_signal generator (V phase/ VB phase)
The values to output one-phase PWM (active high/low) from TAUDnO12 and TAUDnO13
PIC1REG2n1[23:20] = 1010_B (active high), 1111_B (active low)
PIC1REG2n3 [14:12] = 100_B (active high), 101_B (active low)
PIC1REG2n3[10:08] = 100_B (active high), 101_B (active low)

(6) PIC1_signal generator (W phase/ WB phase)
The values to output one-phase PWM (active high/low) from TAUDnO14 and TAUDnO15
PIC1REG2n1[27:24] = 1010_B (active high), 1111_B (active low)
PIC1REG2n3 [22:20] = 100_B (active high), 101_B (active low)
PIC1REG2n3[18:16] = 100_B (active high), 101_B (active low)

(4) Function

Details of this function are described using U phase/UB phase as an example. The function with V phase/VB phase and W phase/WB phase are operated under the same logic as that of U phase/UB phase with different input signals and register settings.

- U phase combination circuit (PFN001)
  This circuit generates reduced dead time pulses*1 (FN00A and FN01A) that are used to insert the pulse generated by the one-shot pulse output function into the triangle wave PWM generated by the triangle wave PWM output function with the dead time. For the block diagram, see **Figure 41.2, Block Diagram of PFN001**.

Note 1. Reduced dead time pulses are the pseudo pulses that are inserted into the PWM output provided by the triangle wave PWM output function with dead time of TAUDn and that are modeled on the dead time pulses that are generated in the range where duty cycle is close to 100% or 0% in PWM output in HT-PWM mode of TSG3n.

- Logical operation circuits (FN0i) (i = 0, 1)
  This circuit synthesizes the triangle wave PWM output (TOUT04 and TOUT05) from the triangle wave PWM output function with the dead time and the outputs from the combinational circuit PFN001 (UO0 and UO1) to generate PWM with variable dead time range pulse inserted. The synthesizing logic of this circuit is selected with PIC1REG2n3k (k = 00 to 02, 04 to 06). For the block diagram, see **Figure 41.3, Block Diagram of FN00**.

Details of this function are described using high accuracy triangle wave PWM output function (U phase/UB phase) as an example.

The following figure shows the timing diagram when U phase duty cycle = 0% and UB phase duty cycle = 100% at active high.

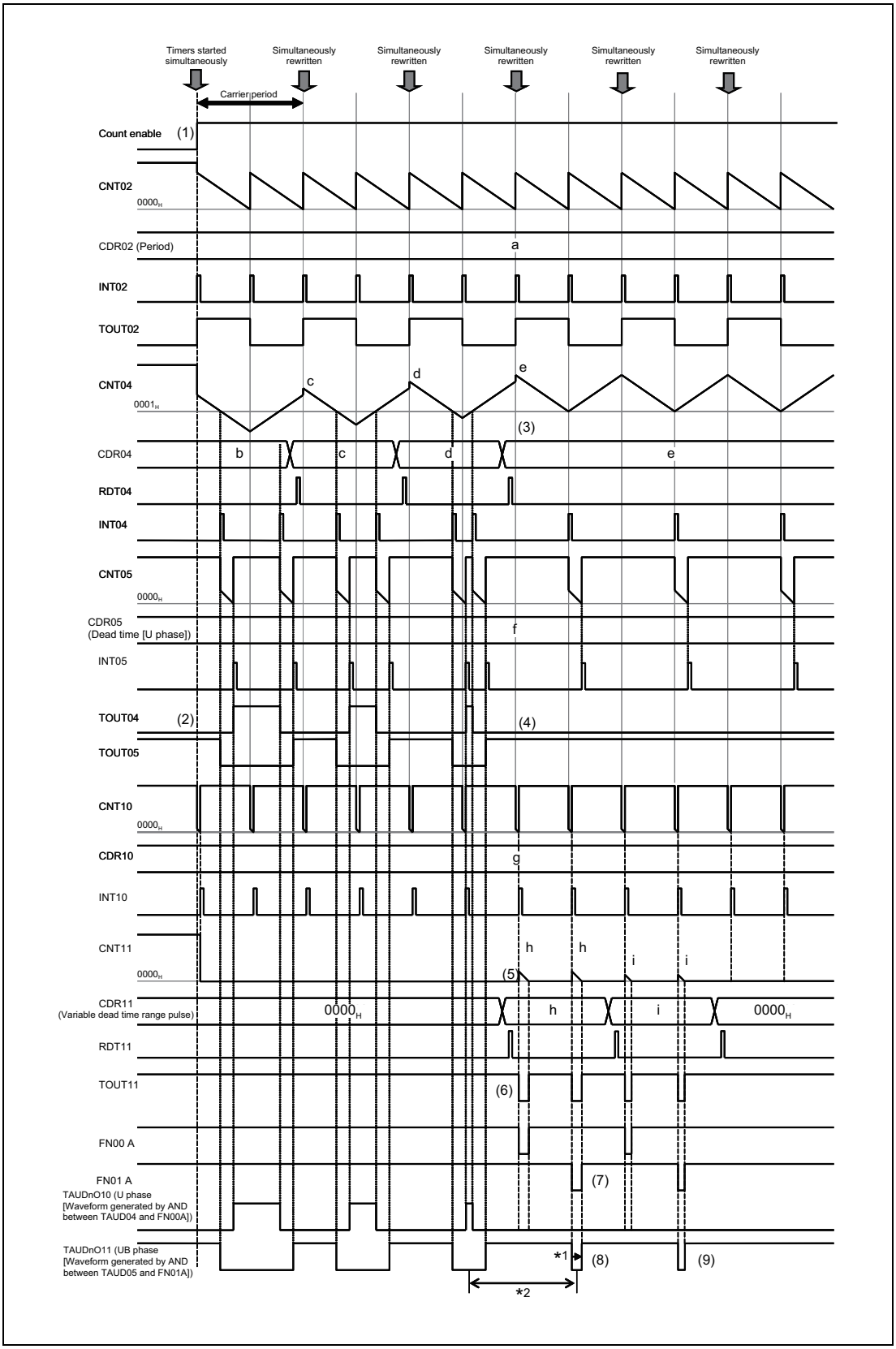


Figure 41.15 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 0%, UB phase duty cycle = 100%) Output with Dead Time (Active High)

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.

(2) TOUT04 and TOUT05 are generated by the triangle wave PWM output function with dead time.

(3) A 0% duty cycle value is set in CDR04 for U phase output.

(4) The setting described in step (3) sets the TOUT04 output to the inactive level and the TOUT05 output to the active level.

(5) To generate the variable dead time range pulse, a value for the reduced dead time pulse width is set in CDR11 when a 0% duty cycle value for U phase output is set in step (3).

(6) CH10 count starts with the effective edge of TOUT02 and INT10 is generated when the counter underflows. CH11 count starts by generation of INT10 and the reduced dead time pulse (TOUT11) of the width set in CDR11 is output.

(7) The reduced dead time pulse (UO1 and UO2) are generated from TOUT02 and TOUT11 in PFN001.

(8) UO1 and UO2 are synthesized with TOUT04 and TOUT05 in FN00 and FN01, and are output as TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal).

CAUTION

Since the reduced dead time pulses are saw tooth waves, they expand and contract on one side, unlike the triangle wave pulses, which expand and contract on both sides. Since one-side expansion and contraction applies to the reduced dead time pulses, a one-phase PWM output period in the variable dead time range is longer by half the width of the inserted reduced dead time pulse.

The following figure shows the timing diagram when U phase duty cycle = 100% and UB phase duty cycle = 0% at active high.

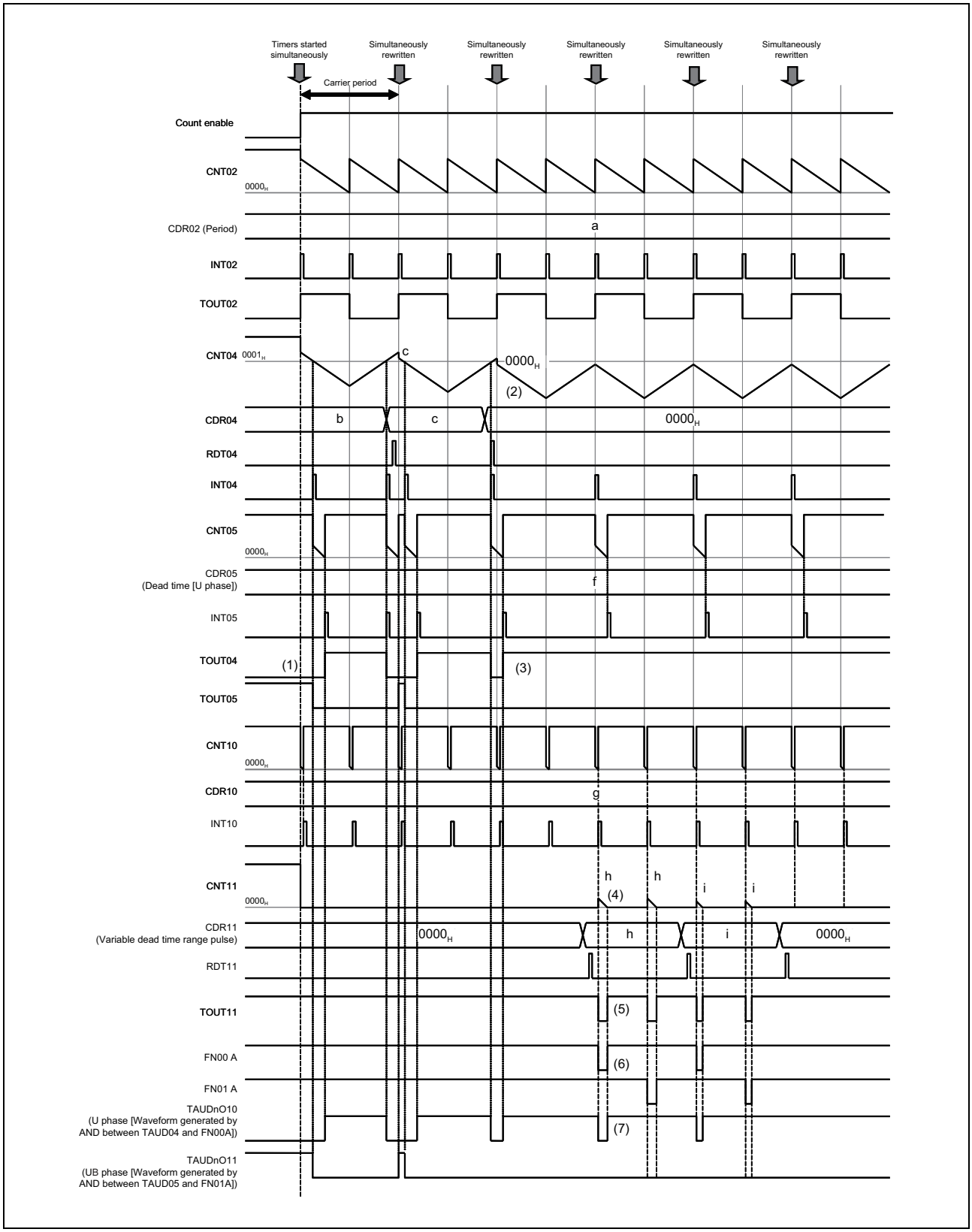


Figure 41.16 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active High)

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.

(2) TOUT04 and TOUT05 are generated by the triangle wave PWM output function with dead time.

(3) A 100% duty cycle value (CDR4 = 0000_H) is set in CDR04 for U phase output.

(4) TOUT04 outputs the active level and TOUT05 outputs the inactive level.

(5) A value for the reduced dead time pulse width is set in CDR11 one period after setting 100% duty cycle for U phase output.

(6) CH10 count starts with the effective edge of TOUT02 and INT10 is generated when the counter underflows. CH11 count starts by generation of INT10 and the reduced dead time pulse (TOUT11) of the width set in CDR11 is output.

(7) The reduced dead time pulse (UO1 and UO2) are generated from TOUT02 and TOUT11 in PFN001.

(8) UO1 and UO2 are synthesized with TOUT04 and TOU05 in FN00 and FN01, and are output as TOPnU (U phase PWM signal) and TOPnUB (UB phase PWM signal).

CAUTION

As shown in Figure 41.17, if a value is set for the variable dead time range pulse width in CDR11 at the same time as a 100% duty cycle value is set for U phase output in CDR04, the variable dead time range pulse affects the last PWM output from TOUT04 (indicated by Figure 41.17, (1)) by the amount of time indicated by Figure 41.17, (2). CDR11 must be set one period after setting CDR04.

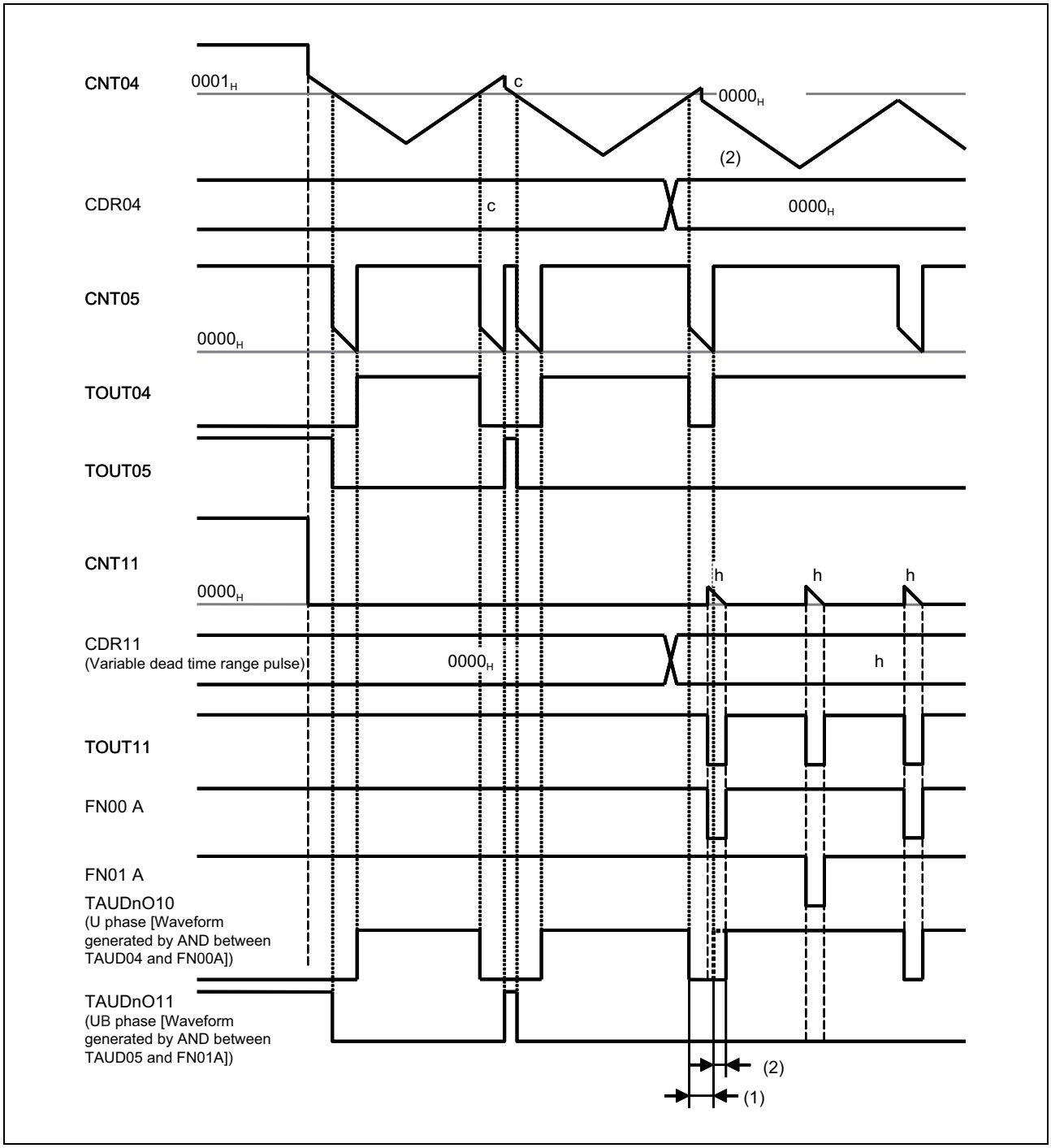


Figure 41.17 Timing Diagram of Variable Dead Time Range Pulse Affecting Triangle Wave PWM Output with Dead Time

The following figure shows the timing diagram when U phase duty cycle = 100% and UB phase duty cycle = 0% at active low.

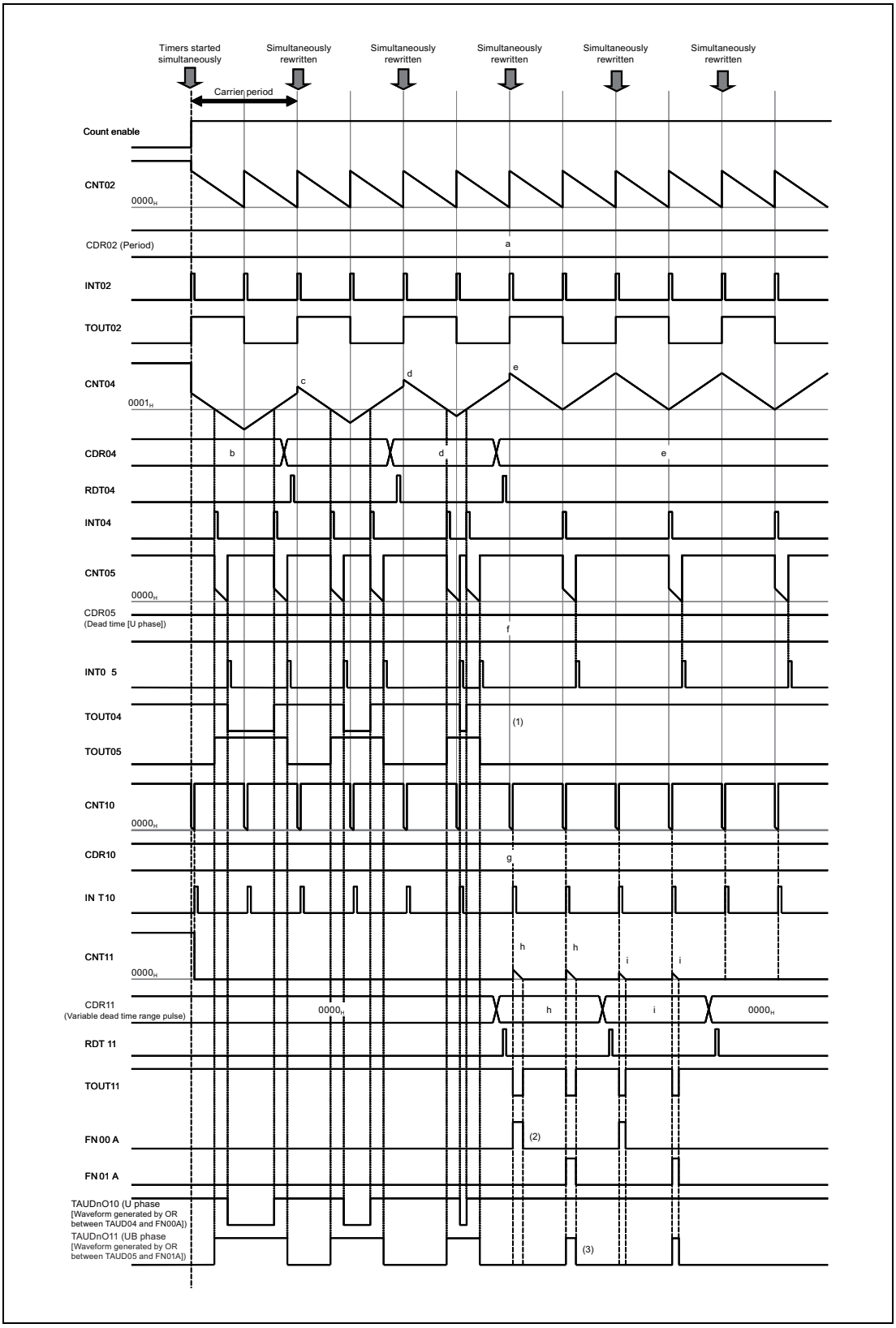


Figure 41.18 Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active Low)

The operation flow from timer operation start to triangle wave PWM output with dead time is same as **Figure 41.16, Timing Diagram of High Accuracy Triangle Wave PWM (U phase duty cycle = 100%, UB phase duty cycle = 0%) Output with Dead Time (Active High)**, with the difference of the PWM output signal from TOUT04 and TOUT05 are active low.

CAUTION

Set each CDR for the one-shot pulse output function so that the following condition is satisfied.

CDR05 ≥ (CDR10 + CDR11)

If the condition above is not satisfied, the output waveform may be influenced. To minimize the influence, satisfy the condition above, and also fix the value of CDR11 to 0000_H until the reduced dead time pulse is required.

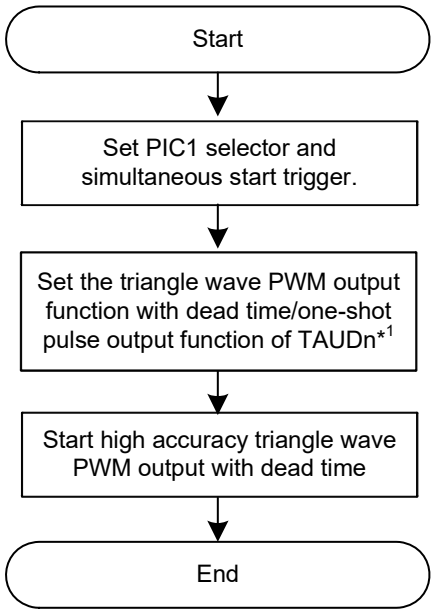
Set the both edges to be detected of TIN10 (TOUT02) as effective, and set TAUDnTOL11 to 1 (active low).

Select the count clock signal (CK0 to 3) of the same clock for TAUDn.

After high accuracy triangle wave PWM output with dead time is started, do not set the value of variable dead time pulse width at the same time as setting a 100% duty cycle value for U phase, V phase, and W phase.

(5) Flow Chart

The flow chart of this function is shown in the following figure.



Note 1. For the setting of triangle wave PWM output function with dead time of TAUDn, see **Section 33.15.8, Triangle PWM Output Function with Dead Time**, or the setting of one-shot pulse output function, see **Section 33.15.2, One-Shot Pulse Output Function**.

(6) Setting Examples for Operation Functions

This section provides example settings for each register.

TAUDn Settings (Active High Example)

Table 41.50 TAUDn: CH2-related (Master Channel Used to Output a Triangle PWM Signal with Dead Time*1)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	At the start of operation, output INTm and toggle TOUTm.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	Fixed

Note 1. The master channel and slave channel names are defined for TAUD triangle PWM output with dead time. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register of the master channel used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.
For this feature, set TAUDnMD0 to 1.

Table 41.51 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Triangle PWM Signal with Dead Time*1) (m = 4, 6, 8)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	111	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1001	
	0	TAUDnMD0	0	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 41.52 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Triangle PWM Signal with Dead Time*1) (m = 5, 7, 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	110	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 41.53 TAUDn: CH10, CH12, and CH14-related (Master Channel used to Output a One-shot Pulse*1) (m = 10, 12, 14)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	10	Detect both rising and falling edges as valid.

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 41.54 TAUDn: CH11, CH13, and CH15-related (Slave Channel used to Output a One-shot Pulse*1) (m = 11, 13, 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	0	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel. Specify the same clock setting as for the master channel (CH2) used to output a triangle PWM signal with dead time.

NOTE

For the TAUDnCMORm register used during one-shot pulse output, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.
For this feature clear TAUDnMD0 to 0.

Table 41.55 Common TAUDn Channel Settings (1/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15	TAUDnTOE15	0 1	Disable the timer. Enable the timer.
	14	TAUDnTOE14	0	
	13	TAUDnTOE13	0 1	Disable the timer. Enable the timer.
	12	TAUDnTOE12	0	
	11	TAUDnTOE11	0 1	Disable the timer. Enable the timer.
	10	TAUDnTOE10	0	
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0 1	Disable the timer. Enable the timer.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0 1	Disable the timer. Enable the timer.
	1, 0	TAUDnTOE01, TAUDnTOE00	Don't care	

Table 41.55 Common TAUDn Channel Settings (2/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTO	15	TAUDnTO15	1*[1]	Output a high-level signal to TOUT15.
	14	TAUDnTO14	Don't care	
	13	TAUDnTO13	1*[1]	Output a high-level signal to TOUT13.
	12	TAUDnTO12	Don't care	
	11	TAUDnTO11	1*[1]	Output a high-level signal to TOUT11.
	10	TAUDnTO10	Don't care	
	9 to 4	TAUDnTO09 to TAUDnTO04	0*[1]	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
	2	TAUDnTO02	0	Output a low-level signal to TOUT02.
	1, 0	TAUDnTO01, TAUDnTO00	Don't care	
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	0	Independent operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01, TAUDnTOM00	Don't care	
TAUDnTOC	15	TAUDnTOC15	1	Operation mode 2
	14	TAUDnTOC14	0	Operation mode 1
	13	TAUDnTOC13	1	Operation mode 2
	12	TAUDnTOC12	0	Operation mode 1
	11	TAUDnTOC11	1	Operation mode 2
	10	TAUDnTOC10	0	Operation mode 1
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1	Operation mode 2
	3	TAUDnTOC03	Don't care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01, TAUDnTOC00	Don't care	
TAUDnTOL	15	TAUDnTOL15	1*[1]	Inverted logic output (active low)
	14	TAUDnTOL14	Don't care	
	13	TAUDnTOL13	1*[1]	Inverted logic output (active low)
	12	TAUDnTOL12	Don't care	
	11	TAUDnTOL11	1*[1]	Inverted logic output (active low)
	10	TAUDnTOL10	Don't care	
	9 to 4	TAUDnTOL09 to TAUDnTOL04	0*[1]	Positive logic output (active high)
	3	TAUDnTOL03	Don't care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01, TAUDnTOL00	Don't care	

Table 41.55 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	0	Disable dead time control.
	9 to 4	TAUDnTDE09 to TAUDnTDE04	1	Enable dead time control.*2
	3	TAUDnTDE03	Don't care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01, TAUDnTDE00	Don't care	
TAUDnTDM	15 to 4	TAUDnTDM15 to TAUDnTDM04	0	
	3	TAUDnTDM03	Don't care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01, TAUDnTDM00	Don't care	
TAUDnTDL	15 to 10	TAUDnTDL15 to TAUDnTDL10	0	Invalid because dead time control is disabled.
	9	TAUDnTDL09	1*1	Dead time is in the negative segment of the W phase output
	8	TAUDnTDL08	0*1	Dead time is in the positive segment of the W phase output
	7	TAUDnTDL07	1*1	Dead time is in the negative segment of the V phase output
	6	TAUDnTDL06	0*1	Dead time is in the positive segment of the V phase output
	5	TAUDnTDL05	1*1	Dead time is in the negative segment of the U phase output
	4	TAUDnTDL04	0*1	Dead time is in the positive segment of the U phase output
	3	TAUDnTDL03	Don't care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01, TAUDnTDL00	Don't care	
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Disable real-time output.
	3	TAUDnTRE03	Don't care	
	2	TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01, TAUDnTRE00	Don't care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01, TAUDnTRO00	Don't care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01, TAUDnTRC00	Don't care	

Table 41.55 Common TAUDn Channel Settings (4/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01, TAUDnTME00	Don't care	
TAUDnRDE	15 to 4	TAUDnRDE15 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01, TAUDnRDE00	Don't care	
TAUDnRDS	15 to 4	TAUDnRDS15 to TAUDnRDS04	0	Do not enable simultaneous rewriting by using another upper channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Do not enable simultaneous rewriting by using another upper channel.
	1, 0	TAUDnRDS01, TAUDnRDS00	Don't care	
TAUDnRDM	15 to 10	TAUDnRDM15 to TAUDnRDM10	0	Perform simultaneous rewriting when the master channel starts counting.
	9 to 4	TAUDnRDM09 to TAUDnRDM04	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	3	TAUDnRDM03	Don't care	
	2	TAUDnRDM02	1	Perform simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	1, 0	TAUDnRDM01, TAUDnRDM00	Don't care	
TAUDnRDC	15 to 4	TAUDnRDC15 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't care	
	2	TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01, TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 33, Timer Array Unit D (TAUD)**.

PIC Settings (Active High Example)

Table 41.56 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC1REG2n0	18	PIC1REG2n018	1	Select the TOUT signal of CH2 of TAUDn.
PIC1REG2n1	27, 26	PIC1REG2n127, PIC1REG2n126	1 0	Negative W phase active high combination circuit output
	25, 24	PIC1REG2n125, PIC1REG2n124	1 0	Positive W phase active high combination circuit output
	23, 22	PIC1REG2n123, PIC1REG2n122	1 0	Negative V phase active high combination circuit output
	21, 20	PIC1REG2n121, PIC1REG2n120	1 0	Positive V phase active high combination circuit output
	19, 18	PIC1REG2n119, PIC1REG2n118	1 0	Negative U phase active high combination circuit output
	17, 16	PIC1REG2n117, PIC1REG2n116	1 0	Positive U phase active high combination circuit output
PIC1REG2n2	25	PIC1REG2n225	1	Select the input selected by the PIC1REG2n018 bit.
	21	PIC1REG2n221	1	Select the input selected by the PIC1REG2n018 bit.
	17	PIC1REG2n217	1	Select the input selected by the PIC1REG2n018 bit.
PIC1REG2n3	22, 21, 20	PIC1REG2n322, PIC1REG2n321, PIC1REG2n320	1 0 0	Negative W phase active high logical operation circuit output
	18, 17, 16	PIC1REG2n318, PIC1REG2n317, PIC1REG2n316	1 0 0	Positive W phase active high logical operation circuit output
	14, 13, 12	PIC1REG2n314, PIC1REG2n313, PIC1REG2n312	1 0 0	Negative V phase active high logical operation circuit output
	10, 9, 8	PIC1REG2n310, PIC1REG2n309, PIC1REG2n308	1 0 0	Positive V phase active high logical operation circuit output
	6, 5, 4	PIC1REG2n306, PIC1REG2n305, PIC1REG2n304	1 0 0	Negative U phase active high logical operation circuit output
	2, 1, 0	PIC1REG2n302, PIC1REG2n301, PIC1REG2n300	1 0 0	Positive U phase active high logical operation circuit output

41.2.3.5 Delay Pulse Output Function with Dead Time

(1) Overview

This function allows generation of PWM output with the dead time, that is, delayed as specified from the period timing using TAUDn.

With this function, PWM output can be reset in the next period unlike with the function described in **Section 41.2.3.3, PWM Output Function With Dead Time**.

The following table lists the number of channels used.

PWM Output with Dead Time	Number of TAUDn Channels
One-phase PWM output (U phase/UB phase)	5 channels (master channel: 1, slave channel: 4)
Two-phase PWM output (U phase/UB phase, V phase/VB phase)	9 channels (master channel: 1, slave channel: 8)
Three-phase PWM output (U phase/UB phase, V phase/VB phase, W phase/WB phase)	13 channels (master channel: 1, slave channel: 12)

Note: Above are examples of PWM output combinations.

Usages of each TAUDn channel are listed in the following table. CH2 is used as the master channel of CH3 to 9.

TAUDn Channel	U phase / UB phase	V phase / VB phase	W phase / WB phase	Usage
CH0	—	—	—	Not used
CH1	—	—	—	Not used
CH2	√	√	√	Carrier period (common to each phase)
CH3	√	√	√	Reserved
CH4	√	—	—	Delay pulse input (U phase/UB phase)
CH5	√	—	—	
CH6	—	√	—	Delay pulse input (V phase/VB phase)
CH7	—	√	—	
CH8	—	—	√	Delay pulse input (W phase/WB phase)
CH9	—	—	√	
CH10	√	—	—	U phase output (TOUT10)
CH11	√	—	—	UB phase output (TOUT11)
CH12	—	√	—	V phase output (TOUT12)
CH13	—	√	—	VB phase output (TOUT13)
CH14	—	—	√	W phase output (TOUT14)
CH15	—	—	√	WB phase output (TOUT15)

Note: √: Used; —: Not used

(2) Configuration

The delay pulse output function with the dead time is realized by using the delay pulse output function/ one-phase PWM output function of TAUDn and PIC1 in combination. The following figure shows the block diagram of the delay pulse output function with the dead time.

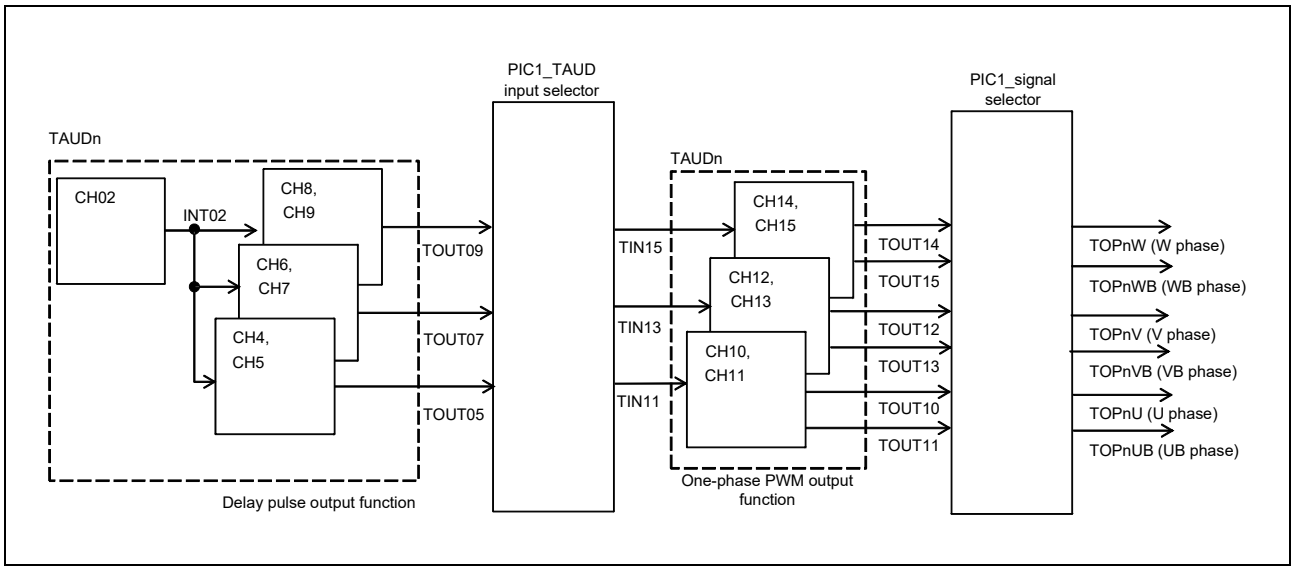


Figure 41.19 Block Diagram of Delay Pulse Output Function with Dead Time

The configuration of this function is described as follows using the PWM output of U phase/UB phase as an example.

- [TAUDn] Delay pulse output function
  CH02, CH04, and CH05 are used in combination. The period, delay value, and the pulse width are set to CDR02, CDR04, and CDR05, respectively, and the delay pulse output signal (TOUT05) is generated.
- [PIC1_TAUD input selector]
  TOUT05 is selected for output as TIN11.
- [TAUDn] One-phase PWM output function
  CH10 and CH11 are used in combination. The dead time value is set to CDR11, dead time is inserted in the PWM signal to be input to TIN11, and TOUT10 (U phase PWM signal) and TOUT11 (UB phase PWM signal) are output.
- [PIC1_signal selector]
  TOUT10 and TOUT11 inputs are selected and output to TOPnU and TOPnUB pins, respectively.

A similar process occurs for V phase/VB phase and W phase/WB phase.

(3) Registers

A block diagram of PIC1 is shown in the following figure.

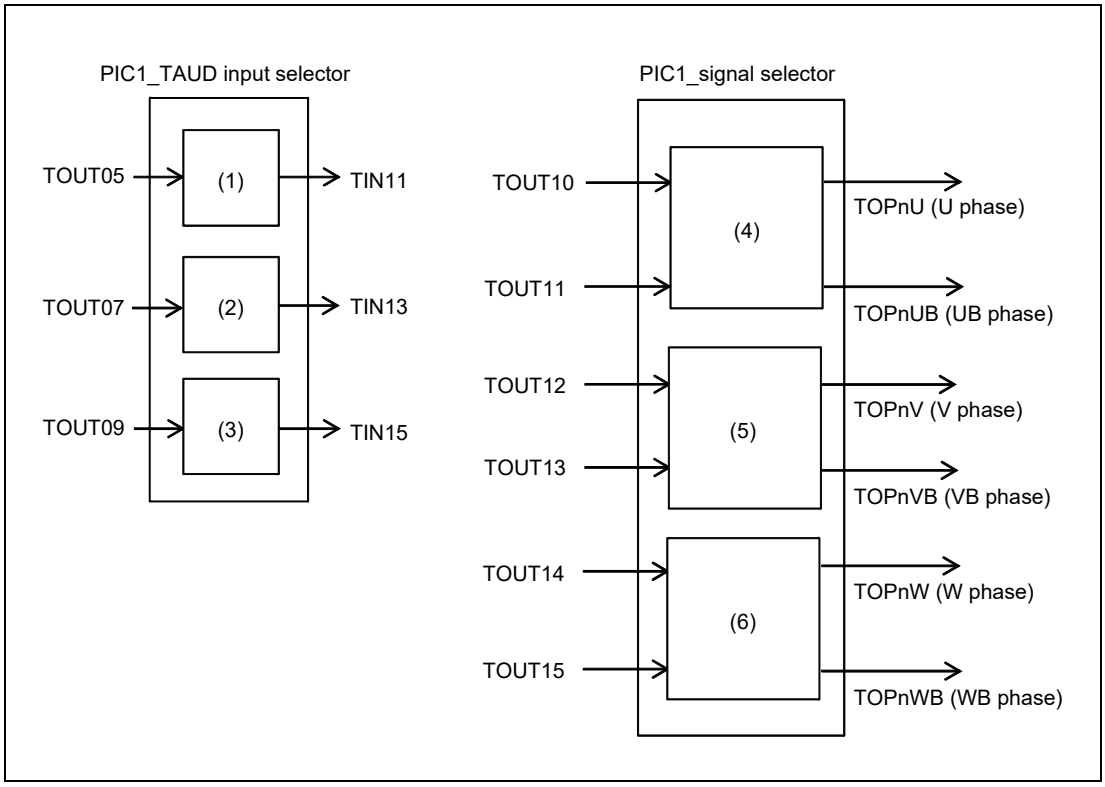


Figure 41.20 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

U phase/ UB phase

The values to output TOUT05 as TIN11 (**Figure 41.20**, unit (1))

PIC1REG2n2[19:18] = 10_B

PIC1REG2n2[2] = 0_B

PIC1TAUDnSEL[23:22] = 00_B

The values to output the TOUT10 and TOUT11 as TOPnU and TOPnUB, respectively (**Figure 41.20**, unit (4))

PIC1REG2n1[19:16] = 0000_B

PIC1REG2n3[2:0] = 000_B

PIC1REG2n3[6:4] = 000_B

V phase/ VB phase

The values to output TOUT07 as TIN13 (**Figure 41.20**, unit (2))

PIC1REG2n2[23:22] = 10_B

PIC1REG2n2[3] = 0_B

PIC1TAUDnSEL[27:26] = 00_B

The values to output TOUT12 and TOUT13 as TOPnV and TOPnVB, respectively (**Figure 41.20**, unit (5))

PIC1REG2n1[23:20] = 0000_B

PIC1REG2n3[10:8] = 000_B

PIC1REG2n3[14:12] = 000_B

W phase/ WB phase

The values to output TOUT09 as TIN15 (**Figure 41.20**, unit (3))

PIC1REG2n2[27:26] = 10_B

PIC1REG2n2[4] = 0_B

PIC1TAUDnSEL[31:30] = 00_B

The values to output TOUT14 and TOUT15 as TOPnW and TOPnWB, respectively (**Figure 41.20**, unit (6))

PIC1REG2n1[27:24] = 0000_B

PIC1REG2n3 [18:16] =000_B

PIC1REG2n3[22:20] =000_B

(4) Function

Details of this function are described using the delay pulse output with dead time (U phase/UB phase) as an example.

The following figure shows the timing diagram.

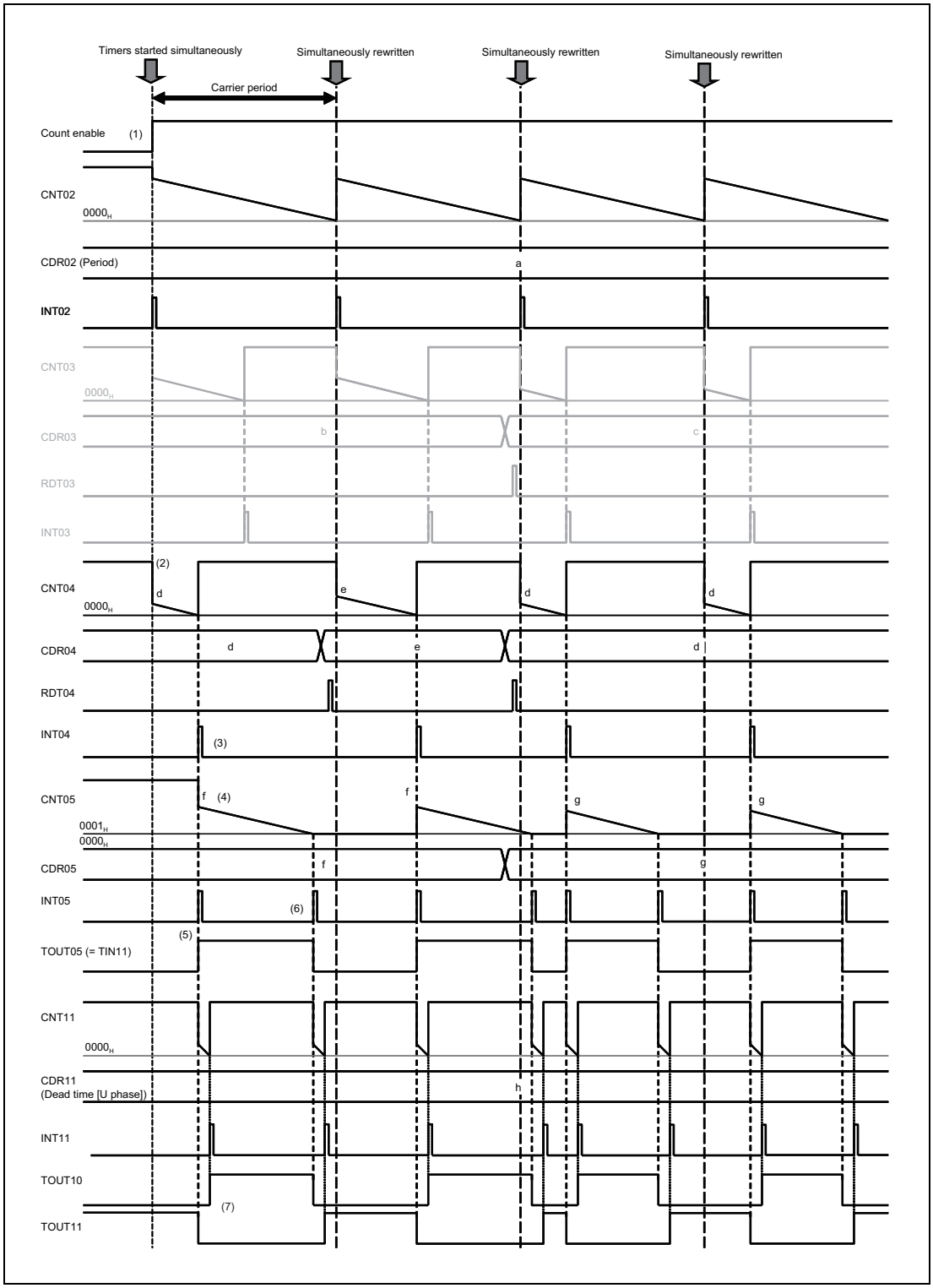


Figure 41.21 Timing Diagram of Delay Pulse Output with Dead Time (U phase/ UB phase)

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.

(2) For CH04, the set value is reloaded to CNT04 upon the occurrence of a CH02 underflow.

(3) INT04 is generated upon the occurrence of a CH04 underflow.

(4) The set value is reloaded to CDR05 upon the generation of INT04, causing CH05 to start operating.

(5) When CH05 starts operating, INT05 is generated and the TOUT05 output level changes to the active level.

(6) Upon CH05 underflow, INT05 is generated again and the TOUT05 output level changes to the inactive level. TOUT05 is supplied to TIN11.

(7) The U phase PWM signal (TOUT10) and UB phase PWM signal (TOUT11) with the dead time are generated and output at the detection of the TIN11 edge, and output to TOPnU and TOPnUB.

A similar process occurs for V phase/VB phase and W phase/WB phase.

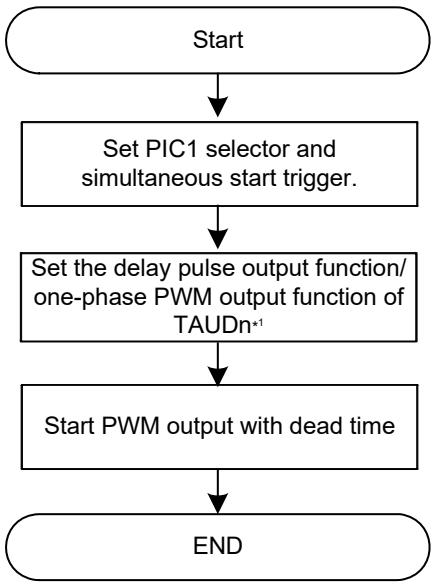
CAUTION

Do not set the delay value which extends over carrier period.

Select the count clock signal of the same clock for TAUDn.

(5) Flow Chart

The following flow chart shows the PMW output function with the dead time.



Note 1. For the setting of delay pulse output function of TAUDn, see **Section 33.15.4, Delay Pulse Output Function**.

(6) Setting Examples for Operation Functions

This section provides example settings for each register.

TAUDn Settings

Table 41.57 TAUDn: CH2-related (Master Channel used to Output a Delay Pulse*1)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	Output INTm at the start of operation.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 41.58 TAUDn: CH3-related (Slave Channel used to Output a Delay Pulse*1*2)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR3	15, 14	TAUDnCKS[1:0]	Don't care*3	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMUR3	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Note 3. For this feature, the channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Table 41.59 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Delay Pulse*1) (m = 4, 6, 8)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Table 41.60 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Delay Pulse*1) (m = 5, 7, 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	101	Start trigger: INTm detection on an upper channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 33, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 41, Peripheral Interconnect (PIC)**.

Table 41.61 TAUDn: CH11, CH13, and CH15-related (One-phase PWM Output) (m = 11, 13, 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*1	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	Start trigger: Detection of a TINm-input valid edge
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 33, Timer Array Unit D (TAUD)**. CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output).

Table 41.62 Common TAUDn Channel Settings (1/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9	TAUDnTOE09	0 1	Disable the timer. Enable the timer.
	8	TAUDnTOE08	0	This is fixed to 0 because TOUT08 is not used.
	7	TAUDnTOE07	0 1	Disable the timer. Enable the timer.
	6	TAUDnTOE06	0	This is fixed to 0 because TOUT06 is not used.
	5	TAUDnTOE05	0 1	Disable the timer. Enable the timer.
	4	TAUDnTOE04	0	This is fixed to 0 because TOUT04 is not used.
	3	TAUDnTOE03	0	This is fixed to 0 because TOUT03 is not used.
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01, TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*1	Output a low-level signal to TOUT15 to TOUT10.
	9 to 2	TAUDnTO09 to TAUDnTO02	0	Output a low-level signal to TOUT09 to TOUT02.
	1, 0	TAUDnTO01, TAUDnTO00	Don't care	

Table 41.62 Common TAUDn Channel Settings (2/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	1	Synchronous operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	0	Independent operation mode
	3	TAUDnTOM03	1	Synchronous operation mode
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01, TAUDnTOM00	Don't care	
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1, 0, 1 0, 1, 0	CH5, CH7, CH9: Operation mode 2 CH4, CH6, CH8: Operation mode 1
	3	TAUDnTOC03	0	Operation mode 1
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01, TAUDnTOC00	Don't care	
TAUDnTOL	15 to 10	TAUDnTOL15 to TAUDnTOL10	0*1	Positive logic output (active high)
	9 to 2	TAUDnTOL09 to TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01, TAUDnTOL00	Don't care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enable dead time control.*2
	9 to 2	TAUDnTDE09 to TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01, TAUDnTDE00	Don't care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 2	TAUDnTDM09 to TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01, TAUDnTDM00	Don't care	
TAUDnTDL	15	TAUDnTDL15	1*1	Add dead time to the negative W phase period.
	14	TAUDnTDL14	0*1	Add dead time to the positive W phase period.
	13	TAUDnTDL13	1*1	Add dead time to the negative V phase period.
	12	TAUDnTDL12	0*1	Add dead time to the positive V phase period.
	11	TAUDnTDL11	1*1	Add dead time to the negative U phase period.
	10	TAUDnTDL10	0*1	Add dead time to the positive U phase period.
	9 to 2	TAUDnTDL09 to TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01, TAUDnTDL00	Don't care	
TAUDnTRE	15 to 2	TAUDnTRE15 to TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01, TAUDnTRE00	Don't care	

Table 41.62 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRO	15 to 2	TAUDnTRO15 to TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01, TAUDnTRO00	Don't care	
TAUDnTRC	15 to 2	TAUDnTRC15 to TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01, TAUDnTRC00	Don't care	
TAUDnTME	15 to 2	TAUDnTME15 to TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01, TAUDnTME00	Don't care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't care	
	9 to 2	TAUDnRDE09 to TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01, TAUDnRDE00	Don't care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 2	TAUDnRDS09 to TAUDnRDS02	0	Enable simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01, TAUDnRDS00	Don't care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't care	
	9 to 2	TAUDnRDM09 to TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01, TAUDnRDM00	Don't care	

Table 41.62 Common TAUDn Channel Settings (4/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't care	
	9 to 2	TAUDnRDC09 to TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01, TAUDnRDC00	Don't care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 33, Timer Array Unit D (TAUD)**.

PIC Settings

Table 41.63 Peripheral Interconnect Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC1REG2n2	27	PIC1REG2n227	1	Select the input selected by the PIC1REG2n204 bit.
	23	PIC1REG2n223	1	Select the input selected by the PIC1REG2n203 bit.
	19	PIC1REG2n219	1	Select the input selected by the PIC1REG2n202 bit.
	4	PIC1REG2n204	0	Select TAUDnTTOUT9.
	3	PIC1REG2n203	0	Select TAUDnTTOUT7.
	2	PIC1REG2n202	0	Select TAUDnTTOUT5.

41.2.3.6 Trigger Pulse Width Measurement Function

(1) Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCAn and TPTM PEn to TAUJ0 and TAUDn.

The following table lists the ENCAn and TPTM PEn interrupt trigger signals to be measured by each timer and channel.

Timer	Channel	Interrupt Signal to be Measured
TAUJ0	CH0	INTENCA0IEC or INTTPTMU10
	CH1	INTENCA0IEC or INTTPTMU10
	CH2	INTENCA1IEC or INTTPTMU30
	CH3	INTENCA1IEC or INTTPTMU30
TAUD0	CH0	ENCAT0EQ0 or ENCAT0EQ1 or INTTPTMU00
	CH1	ENCAT0EQ1 or INTTPTMU01
	CH2	ENCAT0EQ0 or INTTPTMU00
TAUD1	CH0	ENCAT1EQ0 or ENCAT1EQ1 or INTTPTMU11
	CH1	ENCAT1EQ1 or INTTPTMU20
	CH2	ENCAT1EQ0 or INTTPTMU11

(2) Configuration

The trigger and pulse width measurement function is realized by using the TAUJ0 and TINm input pulse width measurement function of TAUDn and PIC1 in combination. The following figure shows the block diagram of the trigger and pulse width measurement function.

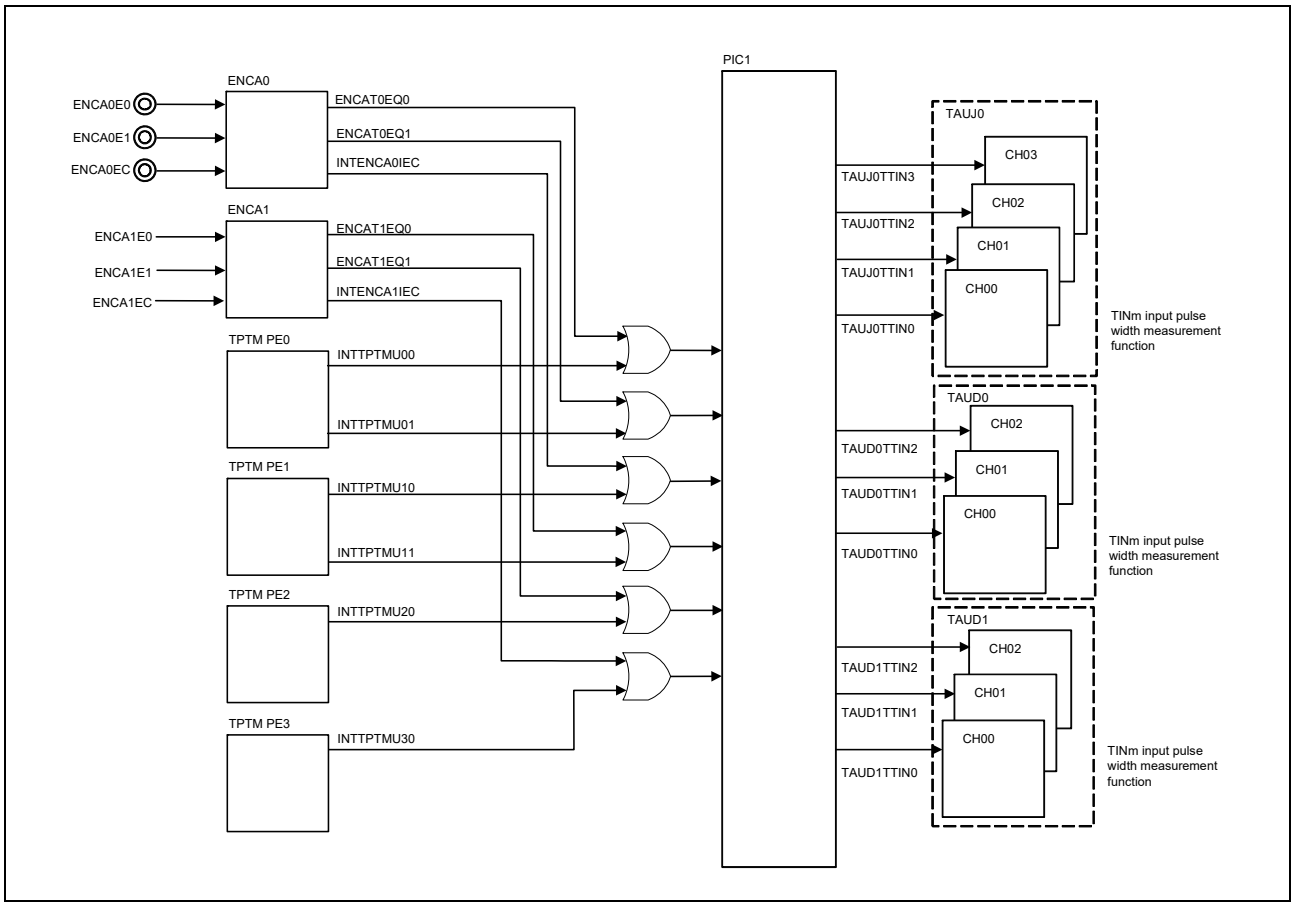


Figure 41.22 Block Diagram of Trigger and Pulse Width Measurement Function

The configuration of this function is described below using TAUJ0 CH0 as an example.

- [ENCA0]
  INTENCA0IEC interrupt signal is generated each time ENCA0 timer counter is cleared by input from ENCA0EC pin.
- [TPTM PE1]
  INTTPTMU10 interrupt signal is generated each time compare hit (CMP0) of each up counter from CPU1 occurs.
- [PIC1] Latch and toggle output (DT) circuit
  INTENCA0IEC or INTTPTMU10 interrupt trigger signal selected by the DT circuit is converted into a level-sensitive toggle signal and output to TAUJ0TTIN0.
- [TAUJ0] TINm input pulse interval measurement function
  TAUJ0 CH0 is used. TAUJ0CNT0 is captured each time input signal is toggled. The counter is cleared and restarted.

Similar configuration is applied for trigger and pulse width measurement of TAUD0 and TAUD1.

(3) Registers

Block diagram of PIC1 is shown in the following figure.

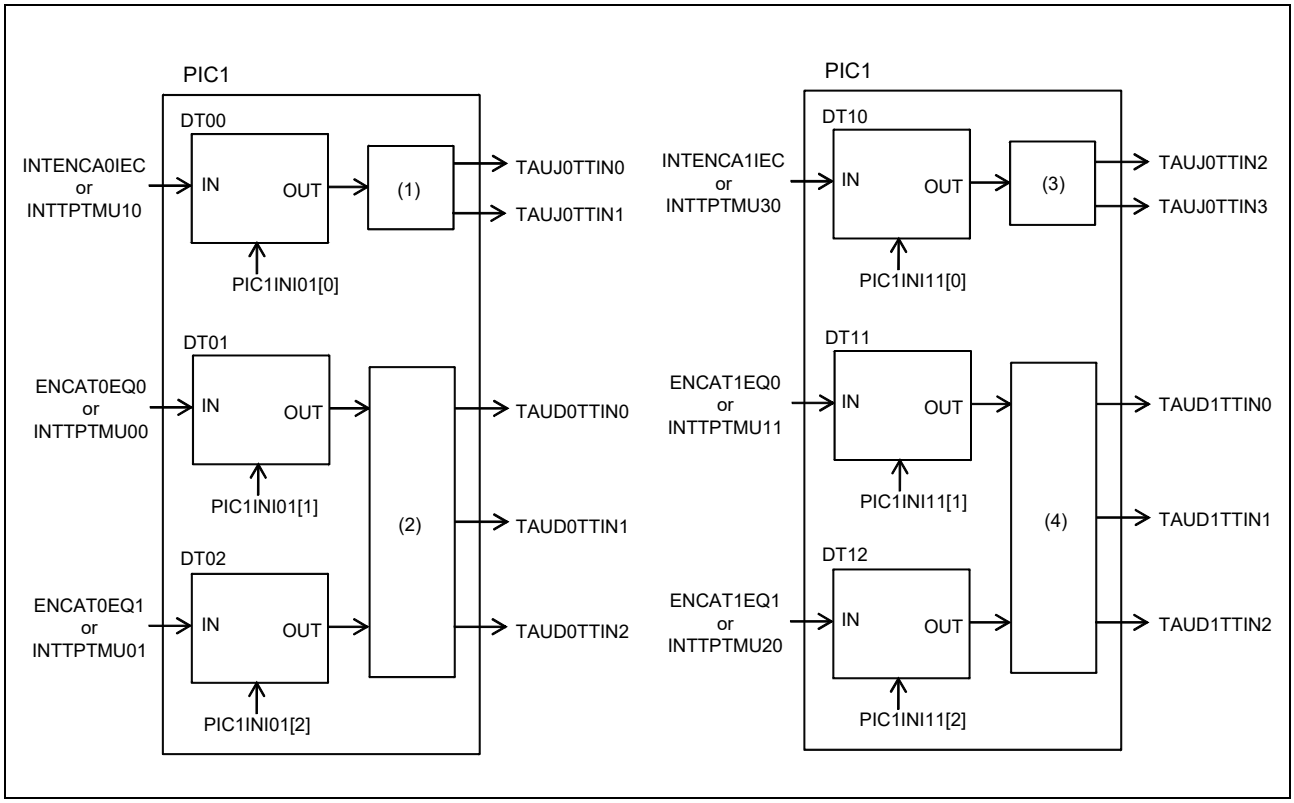


Figure 41.23 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

ENCA0 and TPTM PEn (n = 0,1)

- INTENCA0IEC/INTTPTMU10 trigger and pulse width measurement
  The register values to select the timer which performs INTENCA0IEC/INTTPTMU10 trigger and pulse width measurement. (**Figure 41.23**, unit (1))

Register Value			
PIC1REG31			
1	0	TAUJ0.TIN00	TAUJ0.TIN01
0	0	Not selected.	
0	1	INTENCA0IEC/INTTPTMU10	—
1	0	—	INTENCA0IEC/INTTPTMU10
1	1	INTENCA0IEC/INTTPTMU10	INTENCA0IEC/INTTPTMU10

Note: Write 0 (value after reset) to PIC1REG30[22,17:16]

- ENCAT0EQ0/INTTPTMU00 and ENCAT0EQ1/INTTPTMU01 trigger and pulse width measurement
  The register values to select the timer which performs ENCAT0EQ0/INTTPTMU00 and ENCAT0EQ1/INTTPTMU01 trigger and pulse width measurement. (**Figure 41.23**, unit (2))

Register Value										
PIC1REG31										
13	12	11	10	9	8	7	6	TAUD0.TIN00	TAUD0.TIN01	TAUD0.TIN02
0	0	0	0	0	0	0	0	Not selected.		
0	0	1	1	0	0	0	1	ENCAT0EQ0/ INTTPTMU00	ENCAT0EQ1/ INTTPTMU01	—
0	1	0	0	0	0	0	1	ENCAT0EQ1/ INTTPTMU01	—	ENCAT0EQ0/ INTTPTMU00
0	1	1	0	0	0	0	1	ENCAT0EQ1/ INTTPTMU01	ENCAT0EQ1/ INTTPTMU01	ENCAT0EQ0/ INTTPTMU00
0	1	1	1	0	0	0	1	ENCAT0EQ0/ INTTPTMU00	ENCAT0EQ1/ INTTPTMU01	ENCAT0EQ0/ INTTPTMU00

Note: Do not set the values other than those listed above for this function. Write 0 (value after reset) to PIC1TAUD0SEL[5:0] and PIC1REG30[22,17:16,1:0].

- Enable initialization of the DT02 to DT00 circuits
  The register values to enable initialization of the DT02 to DT00 circuits.
  PIC1INI01[2:0] = 111_B (initialized)

ENCA1 and TPTM PEn (n = 1, 2, 3)

INTENCA1IEC/INTTPTMU30 trigger and pulse width measurement

The register values to select the timer which performs INTENCA1IEC/INTTPTMU30 trigger and pulse width measurement. (**Figure 41.23**, unit (3))

Register Value			
PIC1REG31			
4	3	TAUJ0.TIN02	TAUJ0.TIN03
0	0	Not selected.	
0	1	INTENCA1IEC/ INTTPTMU30	—
1	0	—	INTENCA1IEC/ INTTPTMU30
1	1	INTENCA1IEC/ INTTPTMU30	INTENCA1IEC/ INTTPTMU30

Note: Write 0 (value after reset) to PIC1REG30[20:19,11:10]

- ENCAT1EQ0/INTTPTMU11 and ENCAT1EQ1/INTTPTMU20 trigger and pulse width measurement
  The register values to select the timer which performs ENCAT1EQ0/INTTPTMU11 and ENCAT1EQ1/INTTPTMU20 trigger and pulse width measurement. (**Figure 41.23**, unit (4))

Register Value										
PIC1REG31										
22	21	20	19	18	17	16	15	TAUD1.TIN00	TAUD1.TIN01	TAUD1.TIN02
0	0	0	0	0	0	0	0	Not selected.		
0	0	1	1	0	0	0	1	ENCAT1EQ0/ INTTPTMU11	ENCAT1EQ1/ INTTPTMU20	—
0	1	0	0	0	0	0	1	ENCAT1EQ1/ INTTPTMU20	—	ENCAT1EQ0/ INTTPTMU11
0	1	1	0	0	0	0	1	ENCAT1EQ1/ INTTPTMU20	ENCAT1EQ1/ INTTPTMU20	ENCAT1EQ0/ INTTPTMU11
0	1	1	1	0	0	0	1	ENCAT1EQ0/ INTTPTMU11	ENCAT1EQ1/ INTTPTMU20	ENCAT1EQ0/ INTTPTMU11

Note: Do not set the values other than those listed above for this function. Write 0 (value after reset) to PIC1TAUD1SEL[5:0] and PIC1REG30[20:19,9:6].

- Enable initialization of DT12 to DT10 circuits
  The register values to enable initialization of the DT12 to DT10 circuits.
  PIC1INI11[2:0] = 111_B (initialized)

(4) Function

Details of this function are described as follows using ENCAn as an example.

The following figure shows the timing diagram.

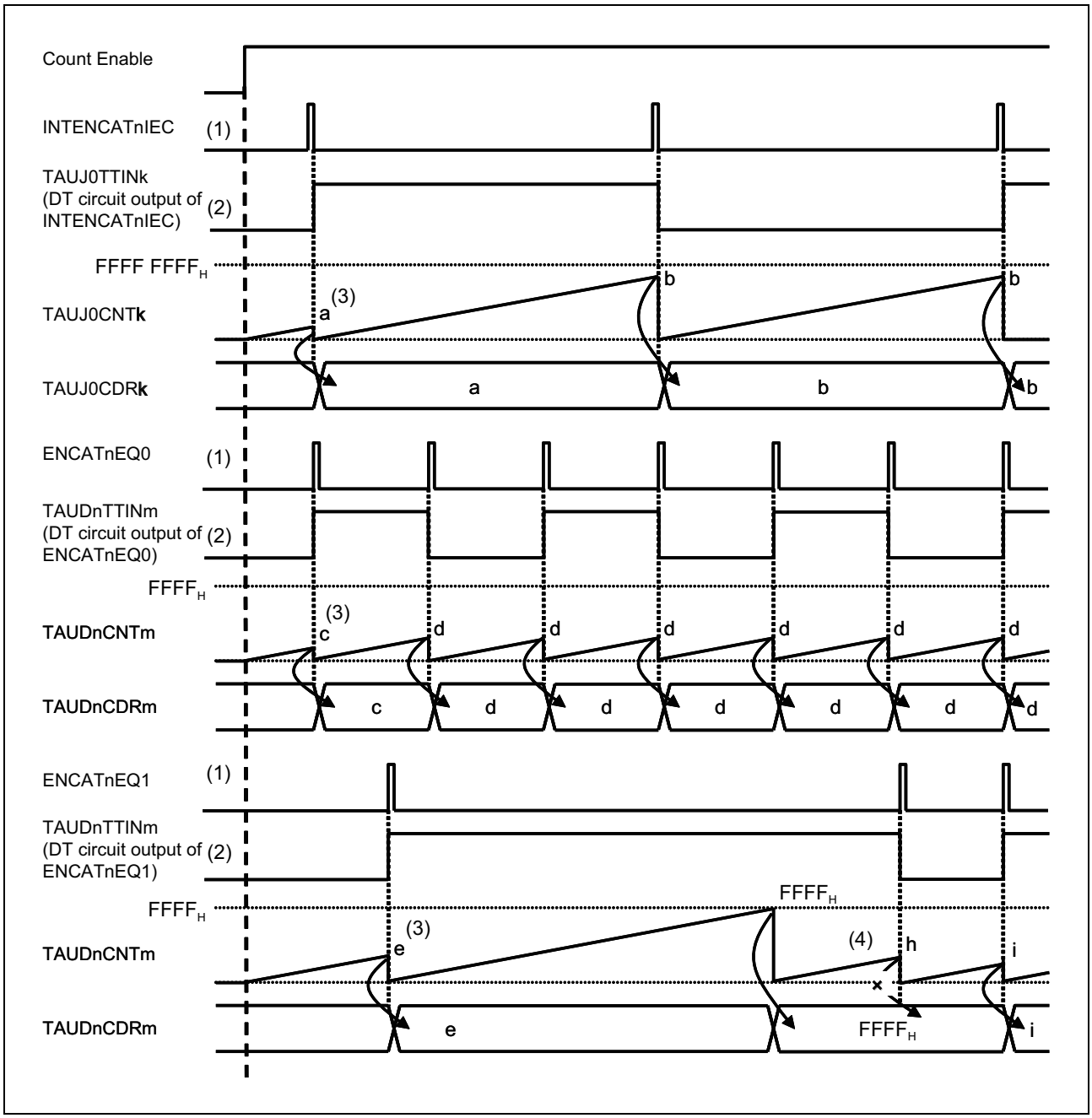


Figure 41.24 Timing Diagram of Trigger and Pulse Width Measurement

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.

(2) The interrupt trigger signal output from ENCAn is converted to the level-sensitive toggle signal by the DT circuit and is output to TINm of TAUJ0 and TINm of TAUDn.

(3) The CNTm value is captured into CDRm on the TINm toggle timing and cleared.

(4) When an overflow occurs, the greatest count value ($FFFF_H$ for TAUDn and FFFF $FFFF_H$ for TAUJ0) is captured and the counter is cleared at the same time. The count value is not captured on the first trigger after the overflow. (When TAUDnCMORm.TAUDnCOS[1] = 1_B)

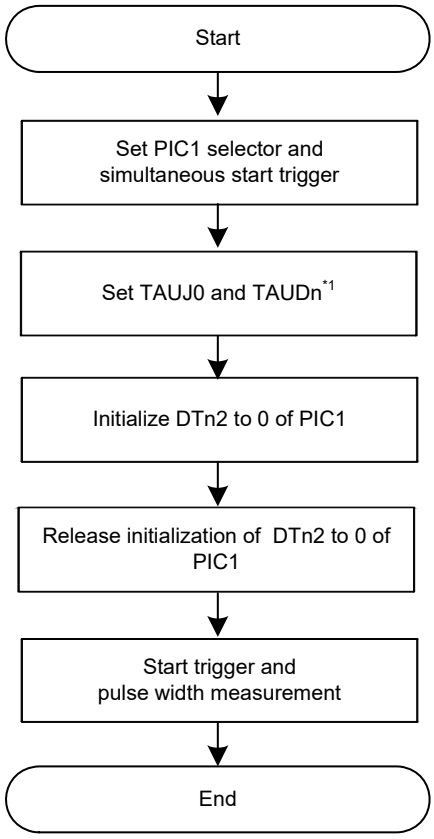
CAUTION

Operation at an overflow varies depending on the setting of TAUJ and TAUD. For details of the TAUJ setting, see Section 34.4.9.3, TAUJnTTINm Input Pulse Interval Measurement Function, see Section 33.12.7, TAUDnTTINm Input Pulse Interval Measurement Function. In this function, set the effective edge to be detected by TAUJ0 and TINm of TAUDn as both edges (rising and falling).

(5) Flow Chart

The following figure shows the setting flow of this function.

This flow chart can be set at both during ENCAn, TPTMPEn operating and while for simultaneous start trigger.



Note 1. For the setting of TAUDn, see **Section 33.12.7, TAUDnTTINm Input Pulse Interval Measurement Function**, for the setting of TAUJ0, see **Section 34.4.9.3, TAUJnTTINm Input Pulse Interval Measurement Function**.

The ENCAn settings to use this function are as follows.

ENCAnCTL[15:0] = xx00 0000 x00x $xxxx_B$

ENCAnIOC0[7:0] = 0000 0000_B

ENCAnIOC1[7:0] = (set any value)

"x" can be set arbitrarily. For the registers specifications, see **Section 40, Encoder Timer A (ENCA)**.

(6) Setting Examples for Operation Functions

This section provides example setting for each register.

The set up example shown in this section describes how to set up measurement of the pulse interval for all the combinations below.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	INTENCA0IEC	TAUJ0 CH0, TAUJ0 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1

Table 41.64 ENCA0 Setting

Register	Bit Position	Bit Name	Setting Value	Note
ENCA0CTL	15	ENCA0CME	Don't care	Enables or disables compare match interrupt detection mask
	14	ENCA0MCS	Don't care	Selects a cancelation trigger for compare match interrupt detection mask
	13 to 10		0	Fixed to 0
	9	ENCA0CRM1	Don't care	Selects the ENCA0CCR1 register function
	8	ENCA0CRM0	Don't care	Selects the ENCA0CCR0 register function
	7	ENCA0CTS	Don't care	Selects trigger for capture operation of ENCA0CCR1.
	6, 5		0	Fixed to 0
	4	ENCA0CLDE	Don't care	Enables or disables reload operation when underflow is generated
	3	ENCA0ECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR1
	2	ENCA0ECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR0
	1, 0	ENCA0UDS[1:0]	Don't care	Selects the counter up/down control by ENCA0E0 and ENCA0E1
ENCA0IOC0	7 to 4		0	Fixed to 0
	3, 2	ENCA0TIS[3:2]	Don't care	Selects the valid edge for capture trigger 1 (ENCA0I1)
	1, 0	ENCA0TIS[1:0]	Don't care	Selects the valid edge for capture trigger 0 (ENCA0I0)
ENCA0IOC1	7	ENCA0SCE	Don't care	Enables encoder special-clear
	6	ENCA0ZCL	Don't care	Selects the clear level of Z phase for encoder special-clear
	5	ENCA0BCL	Don't care	Selects the clear level of B phase for encoder special-clear
	4	ENCA0ACL	Don't care	Selects the clear level of A phase for encoder special-clear
	3, 2	ENCA0ECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCA0EIS[1:0]	Don't care	Selects encoder input (A or B phase) edge

Table 41.65 TAUJ0 Setting (k = 0, 1)
TAUJ0 (TAUJnTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUJ0CMORk	15, 14	TAUJ0CKS[1:0]	Don't care	Operation clock setting
	13, 12	TAUJ0CCS[1:0]	00	
	11	TAUJ0MAS	0	
	10, 9, 8	TAUJ0STS[2:0]	001	
	7, 6	TAUJ0COS[1:0]	11	
	5		0	Fixed to 0
	4, 3, 2, 1	TAUJ0MD[4:1]	0010	
	0	TAUJ0MD0	Don't care	
TAUJ0CMURk	1, 0	TAUJ0TIS[1:0]	10	

NOTE

When TAUJ0CMORk is used for the TAUDnTTINm input pulse interval measurement function, the TAUJ0CKS[1:0] (operating clock selection) and TAUJ0MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUJ0COS[1:0] (overflow mode selection) bits can also set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 34, Timer Array Unit J (TAUJ)**.

For TAUJ common registers (TAUJ0TOE, TAUJ0TO, TAUJ0TOM, TAUJ0TOC, TAUJ0TOL, TAUJ0RDE, and TAUJ0RDM), only set the bits corresponding to the used channels to 0.

Table 41.66 TAUD0 Setting (m = 0 to 2)
TAUD0 (TAUDnTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUD0CMORm	15, 14	TAUD0CKS[1:0]	Don't care	Operation clock setting
	13, 12	TAUD0CCS[1:0]	00	
	11	TAUD0MAS	0	
	10, 9, 8	TAUD0STS[2:0]	001	
	7, 6	TAUD0COS[1:0]	11	
	5		0	Fixed to 0
	4, 3, 2, 1	TAUD0MD[4:1]	0010	
	0	TAUD0MD0	Don't care	
TAUD0CMURm	1, 0	TAUD0TIS[1:0]	10	

NOTE

When TAUD0CMORm is used for the TAUDnTTINm input pulse interval measurement function, the TAUD0CKS[1:0] (operating clock selection) and TAUD0MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Although the TAUD0COS[1:0] (overflow mode selection) bits can also set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 33, Timer Array Unit D (TAUD)**.

For TAUD common registers (TAUD0TOE, TAUD0TO, TAUD0TOM, TAUD0TOC, TAUD0TOL, TAUD0TDE, TAUD0TDM, TAUD0TDL, TAUD0TRE, TAUD0TRO, TAUD0TRC, TAUD0TME, TAUD0RDE, TAUD0RDS, TAUD0RDM, and TAUD0RDC), only set the bits corresponding to the used channels to 0.

Table 41.67 PIC Setting

Register	Bit Position	Bit Name	Setting Value	Note
PIC1REG31	13, 12	PIC1REG3113 PIC1REG3112	0 1	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN2 input signal
	11	PIC1REG3111	1	Selects the signal selected with PIC1REG3106 to PIC1REG3108 (DT output signal from ENCAT0EQ1) as TAUD0TTIN1 input signal
	10, 9	PIC1REG3110 PIC1REG3109	1 0	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN0 input signal
	8 to 6	PIC1REG3108 PIC1REG3107 PIC1REG3106	0 0 1	Selects the DT output signal from ENCAT0EQ1 as TAUD0TTIN1 or TAUD0TTIN0 input signal
	1	PIC1REG3101	1	Selects the DT output signal from INTENCA0IEC as TAUJ0TTIN1 input signal
	0	PIC1REG3100	1	Selects the DT output signal from INTENCA0IEC as TAUJ0TTIN0 input signal

41.2.3.7 Encoder Capture Trigger Select Function

(1) Overview

The function selects any of the ADCJTTOUTn[j] (ADCJn conversion start trigger signal j), INTTAUDnIm (TAUDn-CHm interrupt signal), and ENCAnI1 (signal 1 of the ENCAn external pin input) as the ENCAn capture trigger signal.

(2) Configuration

The encoder capture trigger select function is realized by using ADCJTTOUTn[j] (ADCJn conversion start trigger signal j), INTTAUDnIm (TAUDn CHm interrupt signal), ENCAnI1 (ENCAn external pin input 1 signal), and PIC1.

Block diagram of this function is shown in the following figure.

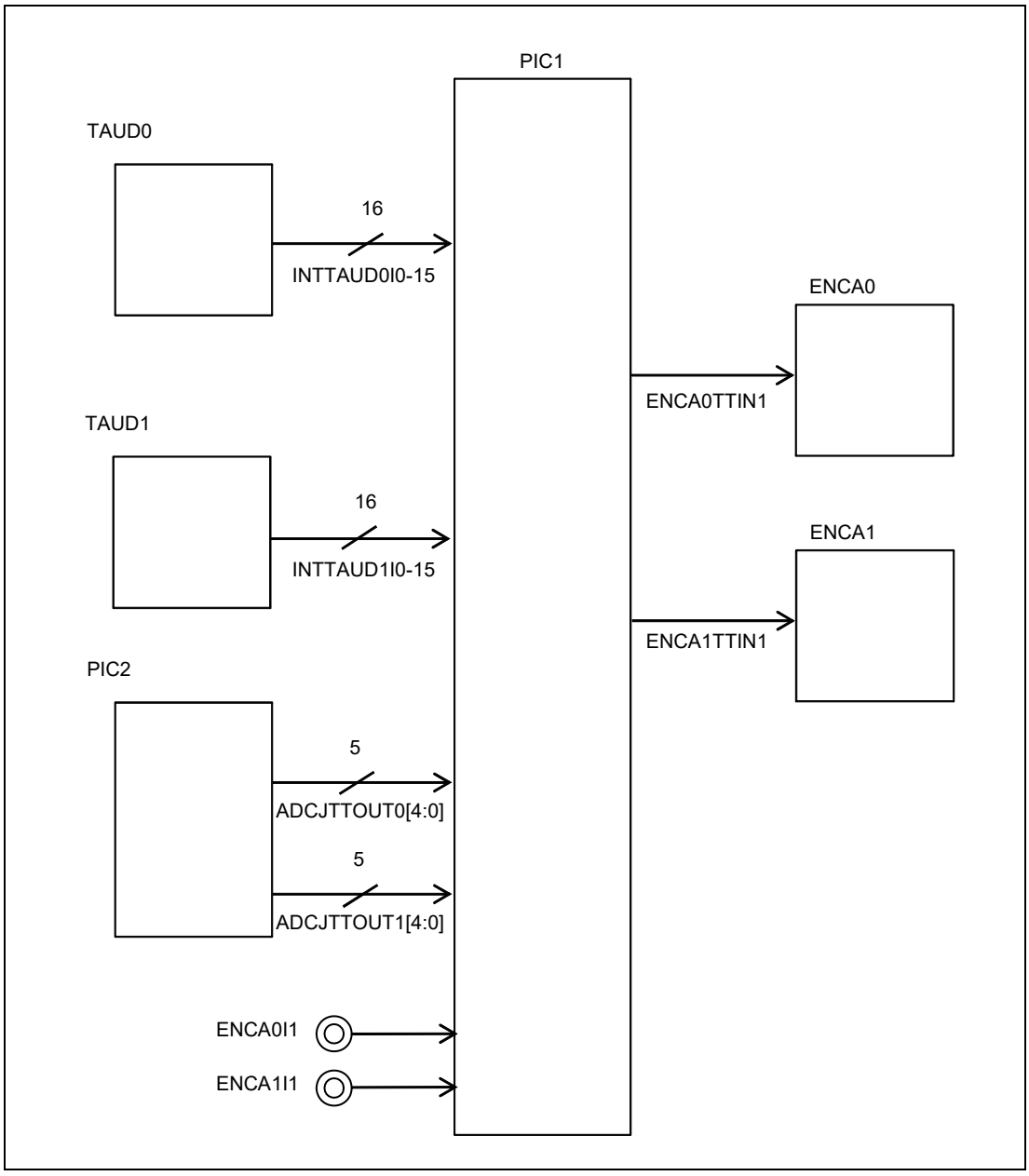


Figure 41.25 Block Diagram of Encoder Capture Trigger Select Function

An example of selecting CH0 of TAUD0 as a capture trigger input of ENCA0 is described below.

PIC1ENCSEL400[7] = 1_B

PIC1ENCSEL400[3:0] = 0000_B

PIC1REG30[18] = 1_B

PIC1REG30[5:2] = 0000_B

(3) Registers

Block diagram of PIC1 is shown in the following figure.

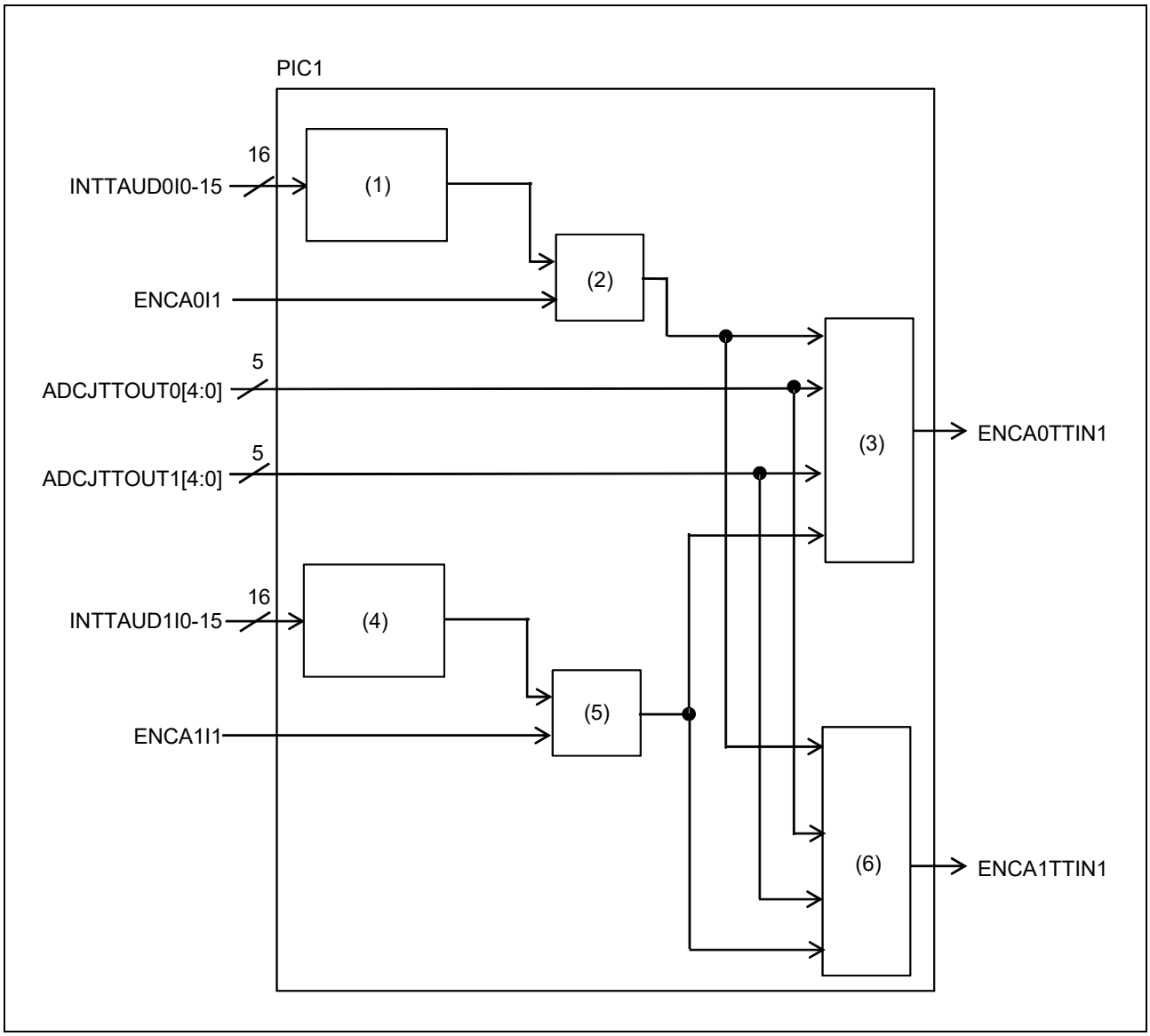


Figure 41.26 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

ENCA0

(1) INTTAUD0Im selection
The register values to select INTTAUD0Im. Set 1 to the PIC1ENCSEL400[7] to select INTTAUD0Im.

Register Values					
PIC1ENCSEL400					
7	3	2	1	0	Output of (1)
1	0	0	0	0	INTTAUD0I0
1	0	0	0	1	INTTAUD0I1
1	0	0	1	0	INTTAUD0I2
1	0	0	1	1	INTTAUD0I3
1	0	1	0	0	INTTAUD0I4
1	0	1	0	1	INTTAUD0I5
1	0	1	1	0	INTTAUD0I6
1	0	1	1	1	INTTAUD0I7
1	1	0	0	0	INTTAUD0I8
1	1	0	0	1	INTTAUD0I9
1	1	0	1	0	INTTAUD0I10
1	1	0	1	1	INTTAUD0I11
1	1	1	0	0	INTTAUD0I12
1	1	1	0	1	INTTAUD0I13
1	1	1	1	0	INTTAUD0I14
1	1	1	1	1	INTTAUD0I15

(2) INTTAUD0Im and ENCA0I1 pins selection
The register values to select either the output (1) or ENCA0I1.

Register Values	
PIC1REG30	
18	Output (2)
1	Output (1)
0	ENCA0I1

(3) ENCA0I1 selection

The register values to select any of the output (2), output (5), ADCJTTOUT0[4:0], or ADCJTTOUT1[4:0].

Register Values				
PIC1REG30				
5	4	3	2	ENCA0I1
0	0	0	0	Output (2)
0	0	0	1	Output (5)
0	0	1	0	ADCJTTOUT0[4]
0	0	1	1	ADCJTTOUT0[3]
0	1	0	0	ADCJTTOUT0[2]
0	1	0	1	ADCJTTOUT0[1]
0	1	1	0	ADCJTTOUT0[0]
0	1	1	1	ADCJTTOUT1[4]
1	0	0	0	ADCJTTOUT1[3]
1	0	0	1	ADCJTTOUT1[2]
1	0	1	0	ADCJTTOUT1[1]
1	0	1	1	ADCJTTOUT1[0]

Note: Do not set the values other than those listed above for this function.

ENCA1

(4) INTTAUD1Im selection

The register values to select INTTAUD1Im. Set 1 to the PIC1ENCSEL410[7] to select INTTAUD1Im.

Register Values					
PIC1ENCSEL410					
7	3	2	1	0	Output (4)
1	0	0	0	0	INTTAUD1I0
1	0	0	0	1	INTTAUD1I1
1	0	0	1	0	INTTAUD1I2
1	0	0	1	1	INTTAUD1I3
1	0	1	0	0	INTTAUD1I4
1	0	1	0	1	INTTAUD1I5
1	0	1	1	0	INTTAUD1I6
1	0	1	1	1	INTTAUD1I7
1	1	0	0	0	INTTAUD1I8
1	1	0	0	1	INTTAUD1I9
1	1	0	1	0	INTTAUD1I10
1	1	0	1	1	INTTAUD1I11
1	1	1	0	0	INTTAUD1I12
1	1	1	0	1	INTTAUD1I13
1	1	1	1	0	INTTAUD1I14
1	1	1	1	1	INTTAUD1I15

(5) INTTAUD1Im and ENCA1I1 pins selection

The register values to select either the output (4) or ENCA1I1.

Register Values	
PIC1REG30	
21	Output (5)
1	Output (4)
0	ENCA1I1

(6) ENCA1I1 selection

The register values to select any of the output (2), output (5), ADCJTTOUT0[4:0], or ADCJTTOUT1[4:0].

Register Values				
PIC1REG30				
15	14	13	12	ENCA1I1
0	0	0	0	Output (5)
0	0	0	1	Output (2)
0	0	1	0	ADCJTTOUT0[4]
0	0	1	1	ADCJTTOUT0[3]
0	1	0	0	ADCJTTOUT0[2]
0	1	0	1	ADCJTTOUT0[1]
0	1	1	0	ADCJTTOUT0[0]
0	1	1	1	ADCJTTOUT0[4]
1	0	0	0	ADCJTTOUT0[3]
1	0	0	1	ADCJTTOUT0[2]
1	0	1	0	ADCJTTOUT0[1]
1	0	1	1	ADCJTTOUT0[0]

Note: Do not set the values other than those listed above for this function.

(4) Function

Details of this function is described with an example of selecting INTTAUDnIm as a capture trigger signal.

The following figure shows the timing diagram.

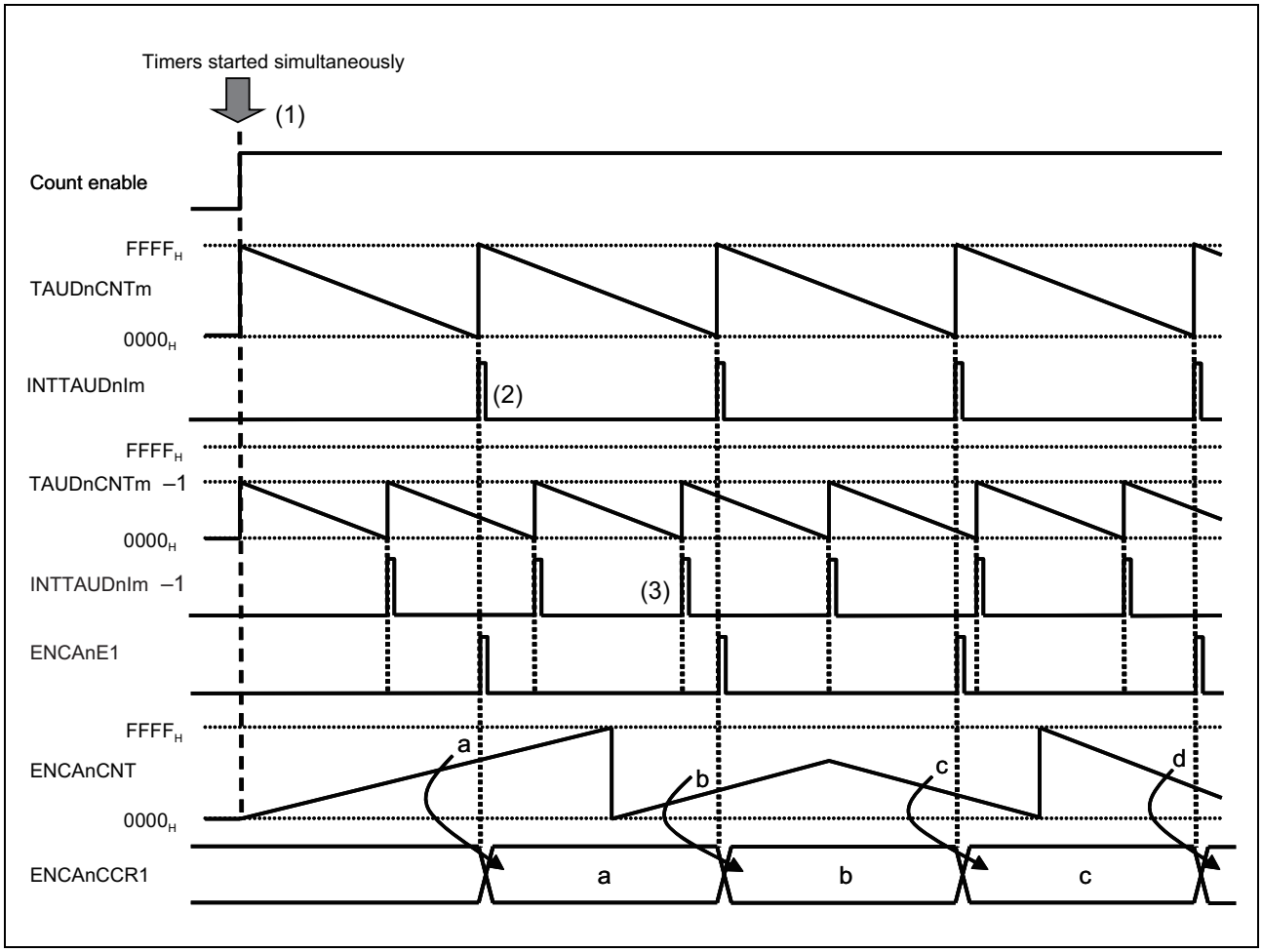


Figure 41.27 Timing Diagram of Encoder Capture Trigger Select Function (INTTAUDnIm)

(1) Using the simultaneous start trigger function, the timers to be used are started simultaneously.

(2) On generation of an effective edge of INTTAUDnIm, ENCAn captures ENCAnCNT.

Do not select ENCAn interrupt trigger signal (INTENCATnI1) as the ADCJn trigger described in **Section 41.3.3.1, ADCJ Trigger Select Function**, the correct operation cannot be performed because the following loop occurs: ADCJTTOUTn[j] generation → ENCAn capture operation → INTENCAnI1 generation by capture operation → ADCJTTOUTn[j] generation.

The following figure shows the loop paths of PIC1, PIC2, and ENCAn.

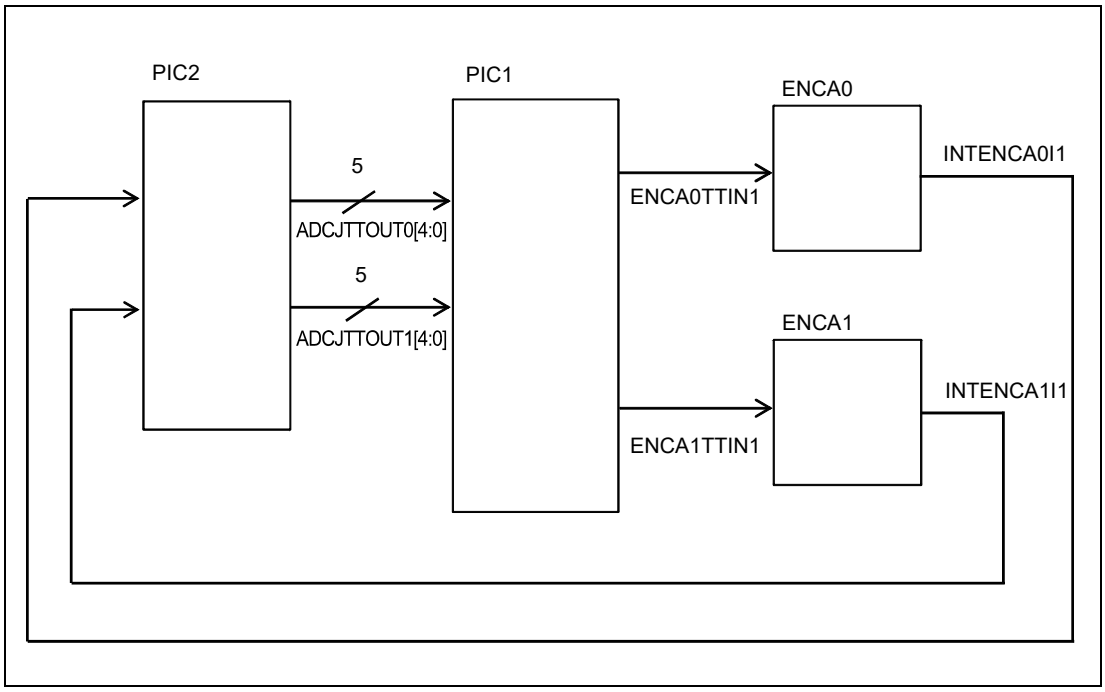


Figure 41.28 Timing Chart of Loop Paths of PIC1, PIC2, and ENCAn

(5) Flow Chart

Select the encoder capture trigger before starting the encoder timer.

ENCAn settings to use this function are as follows.

ENCAnCTL[15:0] = 0000 001x 000x $xxxx_B$

ENCAnIOC0[7:0] = 0000 $01xx_B$

ENCAnIOC1[7:0] = (set any value)

"x" can be set arbitrarily. For the registers specifications, see **Section 40, Encoder Timer A (ENCA)**.

(6) Setting Examples for Operation Functions

This section provides example setting for each register.

The set up example shown in this section describes how to set up capture operation of encoder timer ENCA0 base on the ADCJTTOUT0[4:0] signal. Change the settings depending on the hardware trigger to perform the capture operation.

Table 41.68 ENCAn Setting

Register	Bit Position	Bit Name	Setting Value	Note
ENCAnCTL	15	ENCAnCME	0	Disables compare match interrupt detection masking
	14	ENCAnMCS	0	Selects release trigger for compare match interrupt detection masking
	13 to 10		0	Fixed to 0
	9	ENCAnCRM1	1	Sets the ENCAnCCR1 register for capture operation
	8	ENCAnCRM0	Don't care	Selects the function of ENCAnCCR0 register
	7	ENCAnCTS	0	Selects ENCAnTTIN1 as trigger for capture operation
	6, 5		0	Fixed to 0
	4	ENCAnLDE	Don't care	Enables or disables reload operation when ENCAnCCR0 register underflow occurs
	3	ENCAnECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR1 register
	2	ENCAnECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR0 register
	1, 0	ENCAnUDS[1:0]	Don't care	Selects the counter up/down control by ENCAnE0 and ENCAnE1
ENCAnIOC0	7 to 4		0	Fixed to 0
	3, 2	ENCAnTIS[3:2]	0*1 1*1	Select the valid edge of capture trigger 1 (ENCAnTTIN1) for the rising edge detection
	1, 0	ENCAnTIS[1:0]	Don't care	Select the valid edge of capture trigger 0 (ENCAnTTIN0)
ENCAnIOC1	7	ENCAnSCE	Don't care	Enables encoder special-clear
	6	ENCAnZCL	Don't care	Selects the clear level (input level) of Z phase for encoder special-clear
	5	ENCAnBCL	Don't care	Selects the clear level (input level) of B phase for encoder special-clear
	4	ENCAnACL	Don't care	Selects the clear level (input level) of A phase for encoder special-clear
	3, 2	ENCAnECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCAnEIS[1:0]	Don't care	Selects encoder input (A or B phase) edge
PIC1REG30	4	PIC1REG3004	Don't care	Selects ADCJ0 trigger signal of ENCAnTTIN1
	3	PIC1REG3003	Don't care	
	2	PIC1REG3002	Don't care	

Note 1. Change the setting depending on the hardware trigger to perform the capture operation.

NOTE

Bits ENCA0CRM1 and ENCA0CTS in ENCA0CTL are fixed: ENCA0CRM1 = 1 (ENCA0CCR1 register function) and ENCA0CTS = 0 (trigger source of capture to the ENCA0CCR1 register). All the other bits can be set arbitrarily.

41.2.3.8 Two-Phase Encoder Control Function (Control Method 1)

(1) Overview

This function allows switching of the output patterns of the motor control function (TSG3n) in 120-DC mode using the two-phase encoder control function (ENCAn).

(2) Configuration

Switching of output pattern in 120-DC mode by encoder result is realized by using ENCAn and TSG3n, and PIC1 in combination. The following figure describes the block diagram of two-phase encoder control function (control method 1).

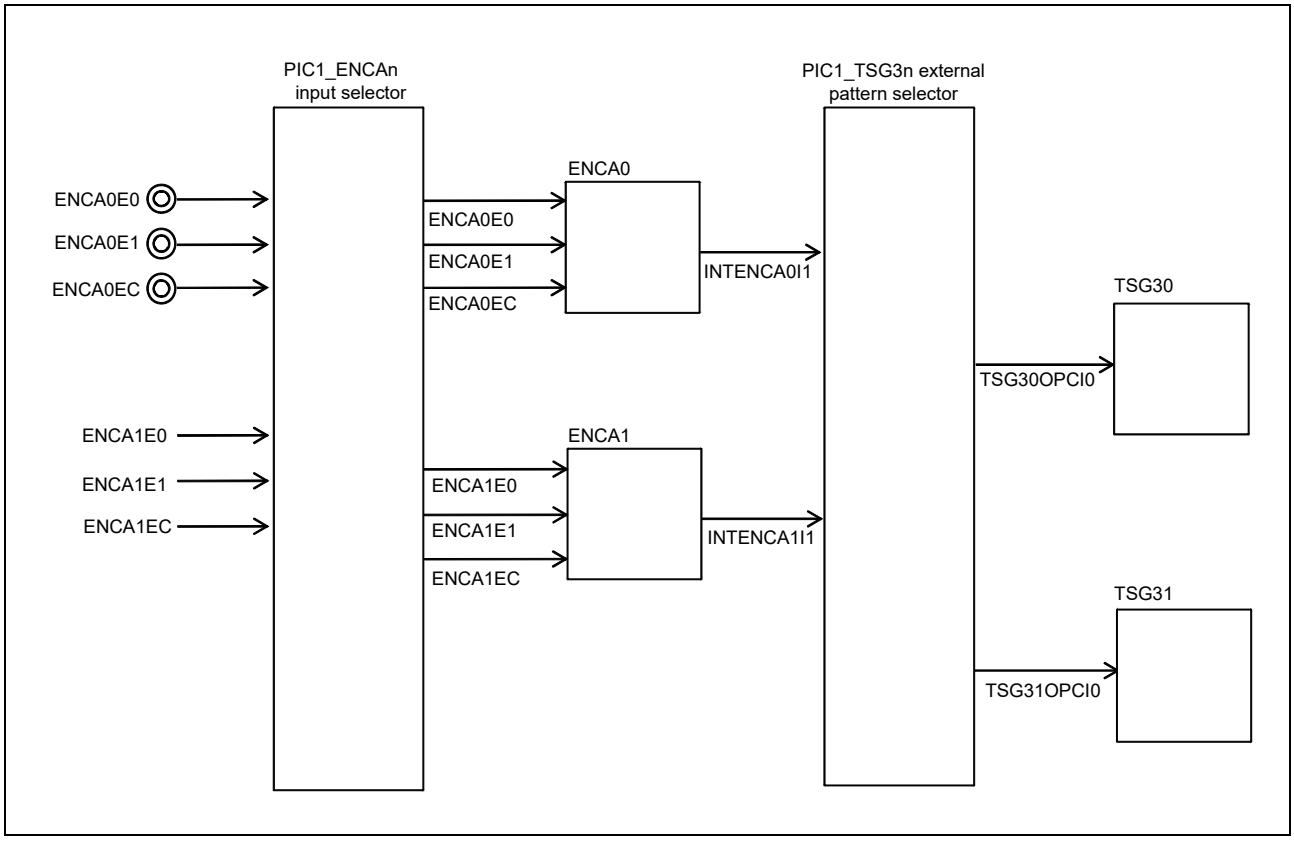


Figure 41.29 Block Diagram of Two-Phase Encoder Control Function (Control Method 1)

The configuration of two-phase encoder control function (control method 1) is described below.

- [PIC1_ENCAn input selector]
  ENCAnE0, ENCAnE1, and ENCAnEC pin inputs are selected and output as ENCAnE0, ENCAnE1, and ENCAnEC.
- [ENCAn]
  INTENCAnI1 is output by two-phase encoder processing.
- [PIC1_TSG3n external pattern selector]
  INTENCAnI1 is selected and output to TSG30OPCI0 or TSG31OPCI0.
- [TSG3n]
  Output pattern in 120-DC mode is switched by TSG3nOPCI0.

(3) Registers

The values of PIC1 registers used in this function are as follows.

PIC1_ENCAn input selector

The register values to output ENCAn pin inputs (ENCAnE0, ENCAnE1, and ENCAnEC) as ENCAnE0, ENCAnE1, and ENCAnEC.

PIC1REG30[22] = 0_B

PIC1REG30[20:19] = 00_B

PIC1REG30[17:16] = 00_B

PIC1REG30[11:6] = 000000_B

PIC1REG30[1:0] = 00_B

PIC1_TSG3n external pattern selection

The register values to select the interrupt signal to be input as TSG30 external pattern.

Register Values				
PIC1REG50				
10	8	6	5	TSG30OPCI0
X	0	0	1	INTENCA0I1
0	X	1	0	INTENCA1I1

Note: Do not set the values other than those listed above for this function.

The register values to select the interrupt signal to be input as TSG31 external pattern.

Register Values				
PIC1REG51				
10	8	6	5	TSG31OPCI0
X	0	0	1	INTENCA0I1
0	X	1	0	INTENCA1I1

Note: Do not set the values other than those listed above for this function.

(4) Function

Details of this function are described using the two-phase encoder control function (method 1) at up count (normal rotation) as an example.

The following figure shows the timing diagram.

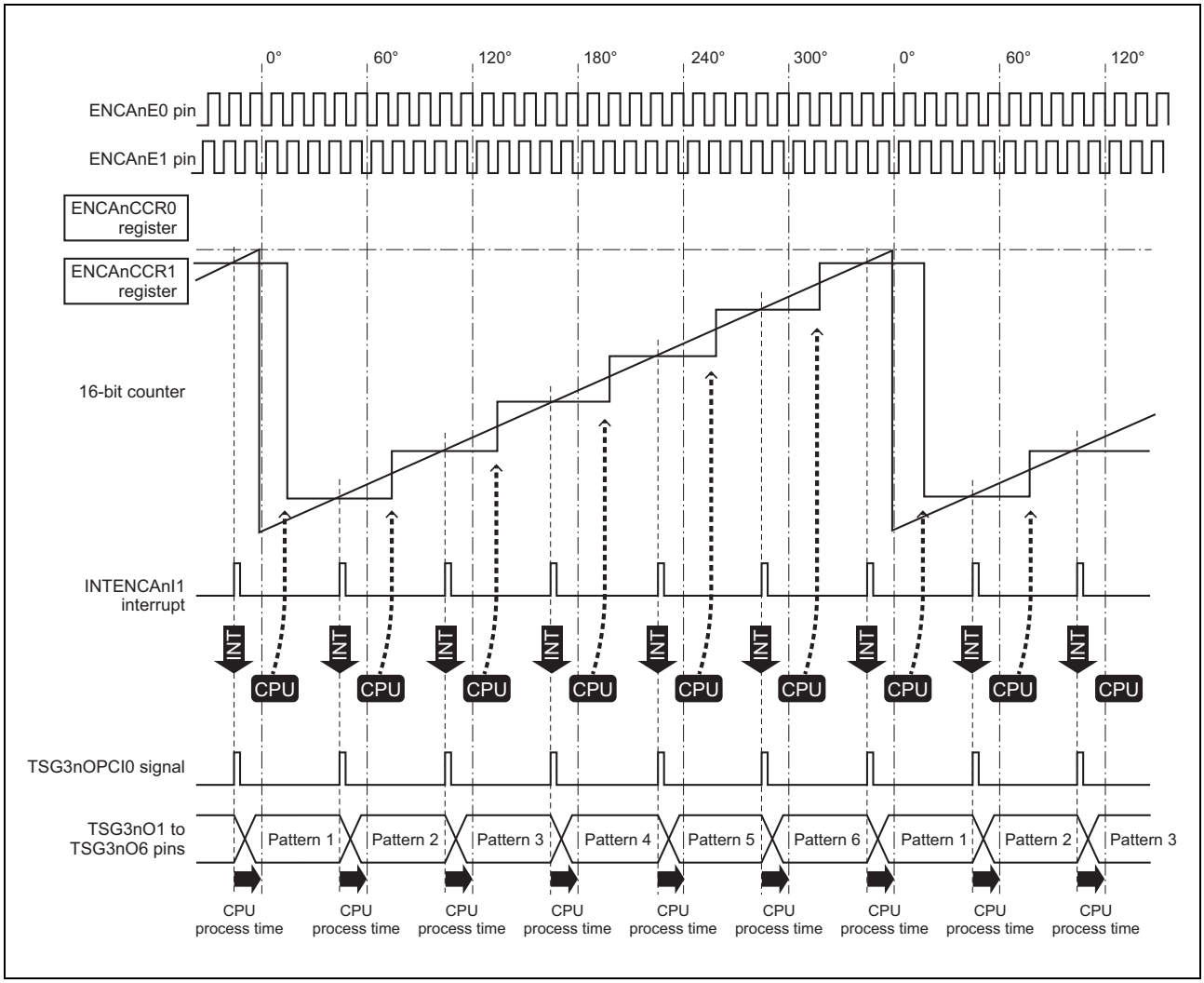


Figure 41.30 Timing Diagram of Two-Phase Encoder Control Function (Control Method 1) at Up Count (Normal Rotation)

(1) The encoder counter value matches the ENCAnCCR1, INTENCAnI1 is generated, and the set pattern is output from the TSG3nO1 to TSG3nO6 pins.

(2) CPU calculates the next timing to switch output patterns by an interrupt processing and sets the value to ENCAnCCR1.

(3) Upon a match of the encoder counter value and ENCAnCCR0, the encoder counter is cleared.

CAUTION

It is necessary to set ENCAnCCR1 at each pattern switch (each INTENCAnI1 interrupt). It is necessary to match the initial output pattern of timer TSG3n to the set value of ENCAnCCR1 before starting, because this value is not cleared by the encoder clear input.

Switching between normal and reverse rotations of output patterns is set with the TSG3nPSC bit of the TSG3nOPT0 register.

The following figure shows the timing diagram at down count (reverse rotation).

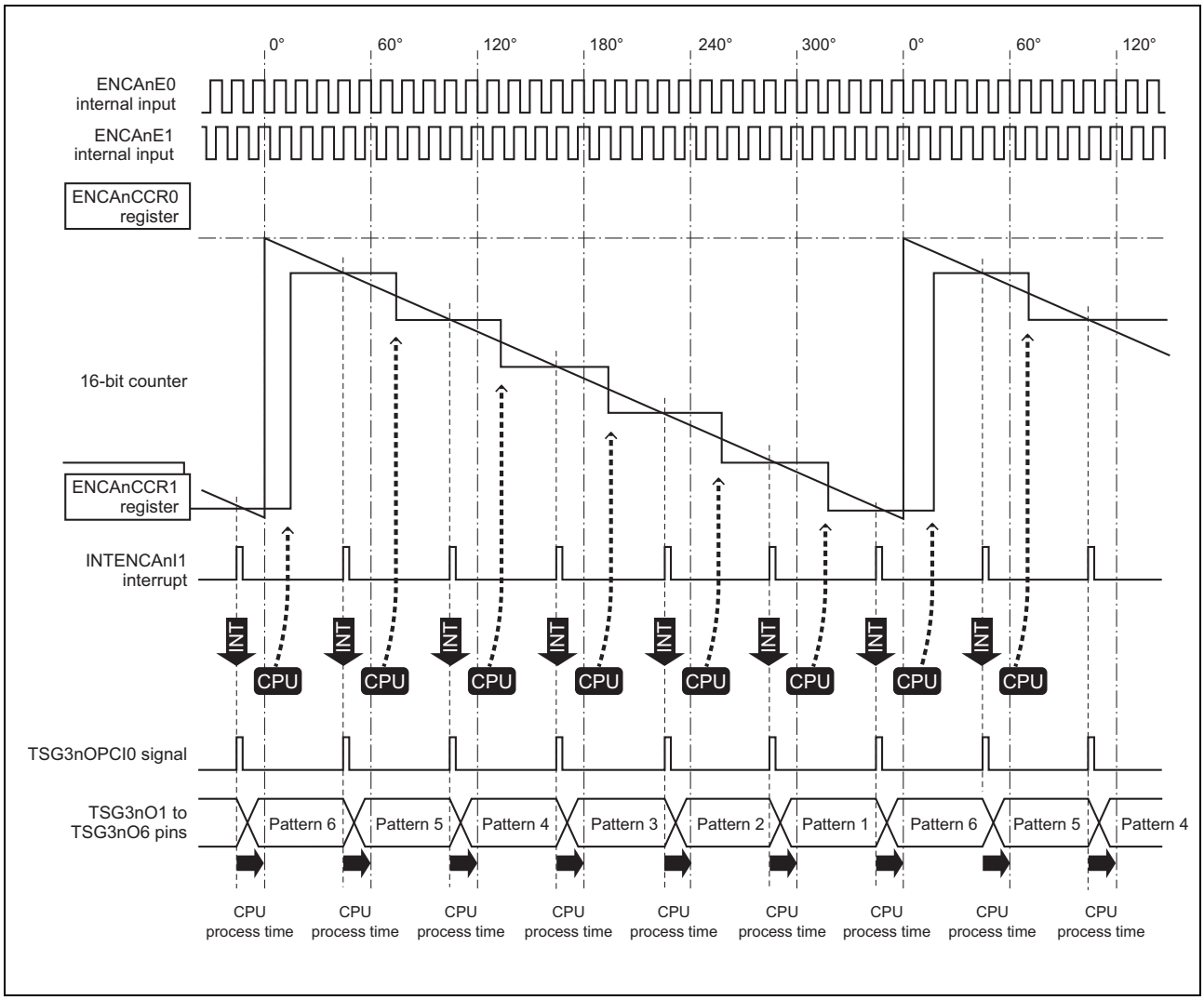
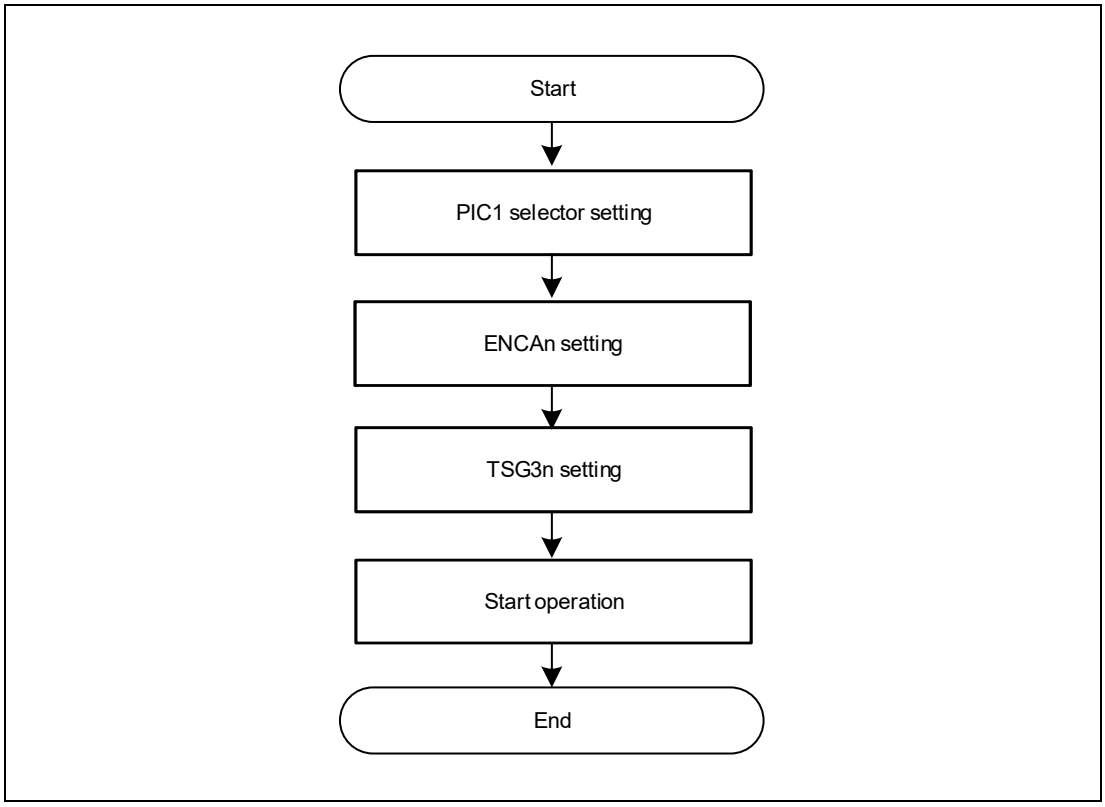


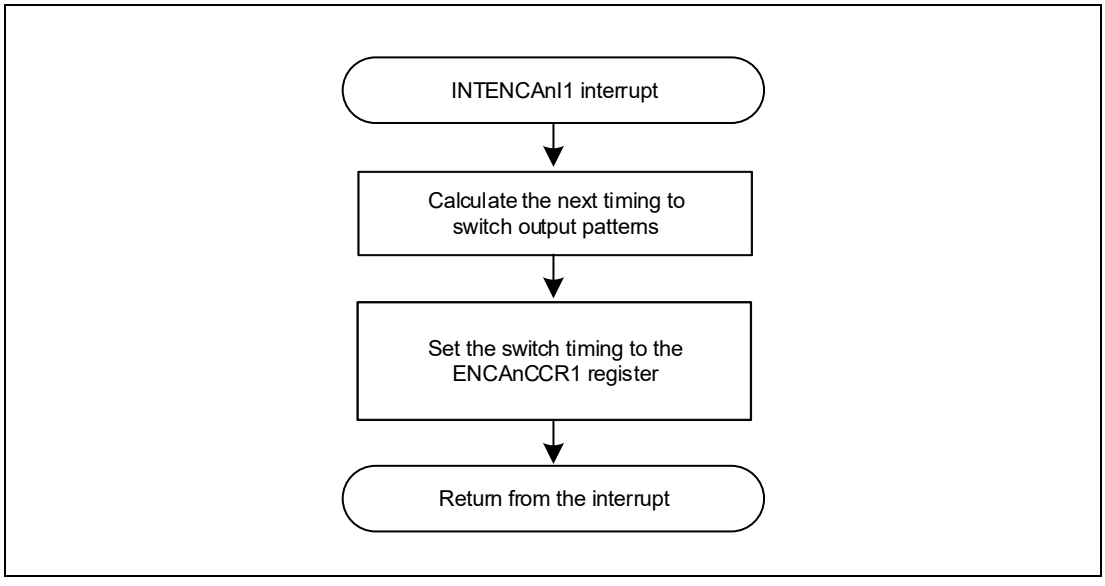
Figure 41.31 Timing Diagram of Two-Phase Encoder Control Function (Control Method 1) at Down Count (Reverse Rotation)

(5) Flow Chart

The following figure shows the setting flow of this function.



The following figure shows the flow chart after interrupt processing.



The values of ENCAn registers used in this function are as follows.

ENCAnCTL[15:0] = 1000 0000 000x 01xx_B

ENCAnIOC1[7:0] = 0000 00xx_B

ENCAnCCR0= (set any value)

ENCAnCCR1= (set any value)

ENCAnCNT= (set any value)

"x" can be set arbitrarily. For the registers specifications, see **Section 40, Encoder Timer A (ENCA)**.

The values of TSG3n registers used in this function are as follows.

TSG3nCTL0[7:0] = 000x 0011_B

TSG3nCTL3[7:0] = 0000 00xx_B

TSG3nCTL4[15:0] = 0000 0001 xxx0 0000_B

TSG3nIOC0[7:0] = 0111 1110_B

TSG3nIOC2[15:0] = 0xxx xxx0 0000 0000_B

TSG3nOPT0[7:0] = 0011 1xx0_B

TSG3nOPT1[7:0] = 0000 0xxx$_B$

TSG3nCMP0 = (set any value)

TSG3nCMP1W,5W,9W = (set any value)

TSG3nCMP1,5,9 = (set any value)

TSG3nPAT0W,1W = (set any value)

TSG3nDTC0W,1W = (set any value)

"x" can be set arbitrarily. For the registers specifications, see **Section 35, Motor Control Timer (TSG3)**.

41.2.3.9 Two-Phase Encoder Control Function (Control Method 2)

(1) Overview

This function allows switching the phase lead and phase lag control of the motor control function (TSG3n) output patterns in 120-DC mode using the two-phase encoder control function (ENCAn).

(2) Configuration

The same configuration as the **Section 41.2.3.8, Two-Phase Encoder Control Function (Control Method 1)** applies to this function. See **(2) Configuration** of **Section 41.2.3.8, Two-Phase Encoder Control Function (Control Method 1)**.

(3) Registers

The same register values as the **Section 41.2.3.8, Two-Phase Encoder Control Function (Control Method 1)** applies to this function. See **(3) Registers** in **Section 41.2.3.8, Two-Phase Encoder Control Function (Control Method 1)**.

(4) Function

Details of this function are described using **Section 41.2.3.9, Two-Phase Encoder Control Function (Control Method 2)** with phase lead (normal rotation) as an example.

The following figure shows the timing diagram.

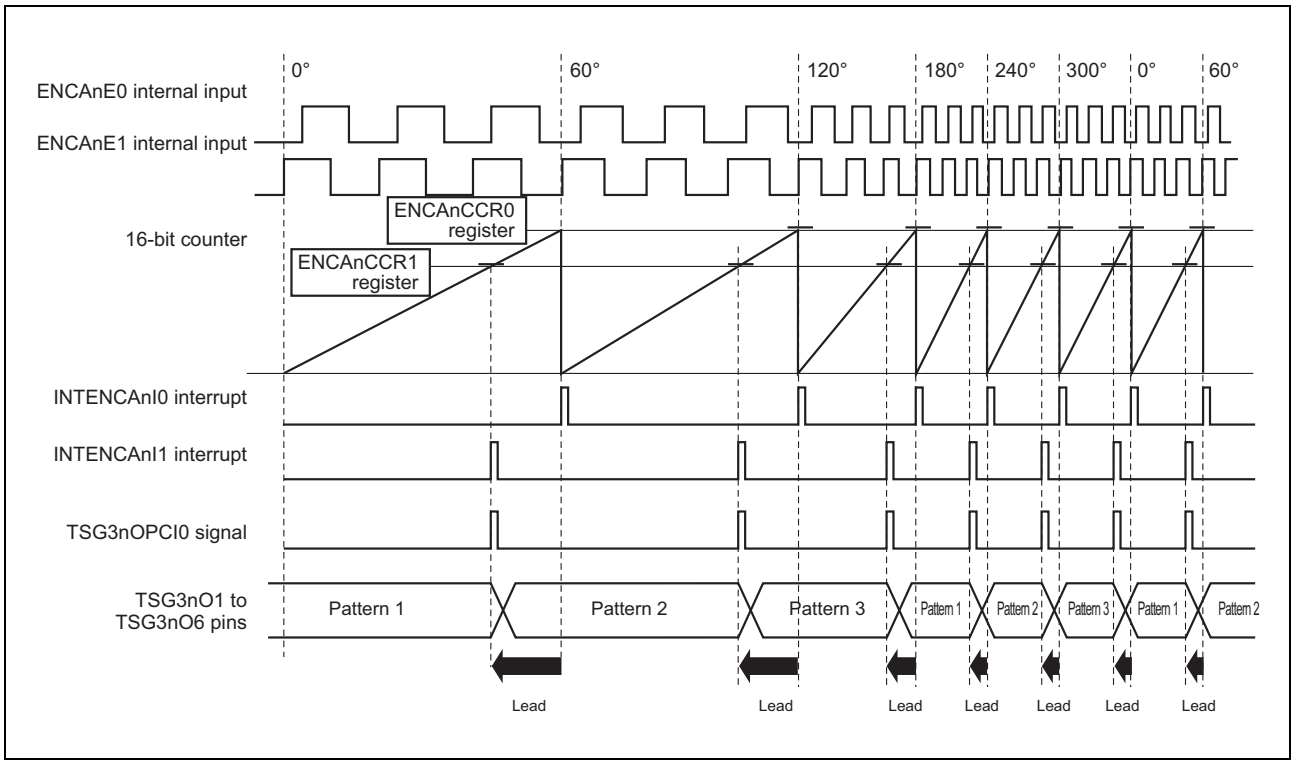


Figure 41.32 Timing Diagram of Two-Phase Encoder Control Function (Method 2) with Phase Lead (Normal Rotation)

(1) Upon a match of the encoder counter value and ENCAnCCR1 (corresponds to the output pattern switch position for TSG3n), INTENCAnI1 is generated, and the set pattern is output from the TSG3nO1 to 6 pins.

(2) Upon a match of the encoder counter value and ENCAnCCR0 (corresponds to the phase lead or lag of the switch position), INTENCAnI0 is generated and the encoder counter is cleared.

CAUTION

It is necessary to set ENCAnCCR1 at each pattern switch (each INTENCAnI1 interrupt). It is necessary to match the initial output pattern of timer TSG3n to the set value of ENCAnCCR0 before starting, because this value is not cleared by the encoder clear input.

Switching between normal and reverse rotations of output patterns is set with the TSG3nPSC bit of the TSG3nOPT0 register.

The following figure shows the timing diagram with phase lag (normal rotation).

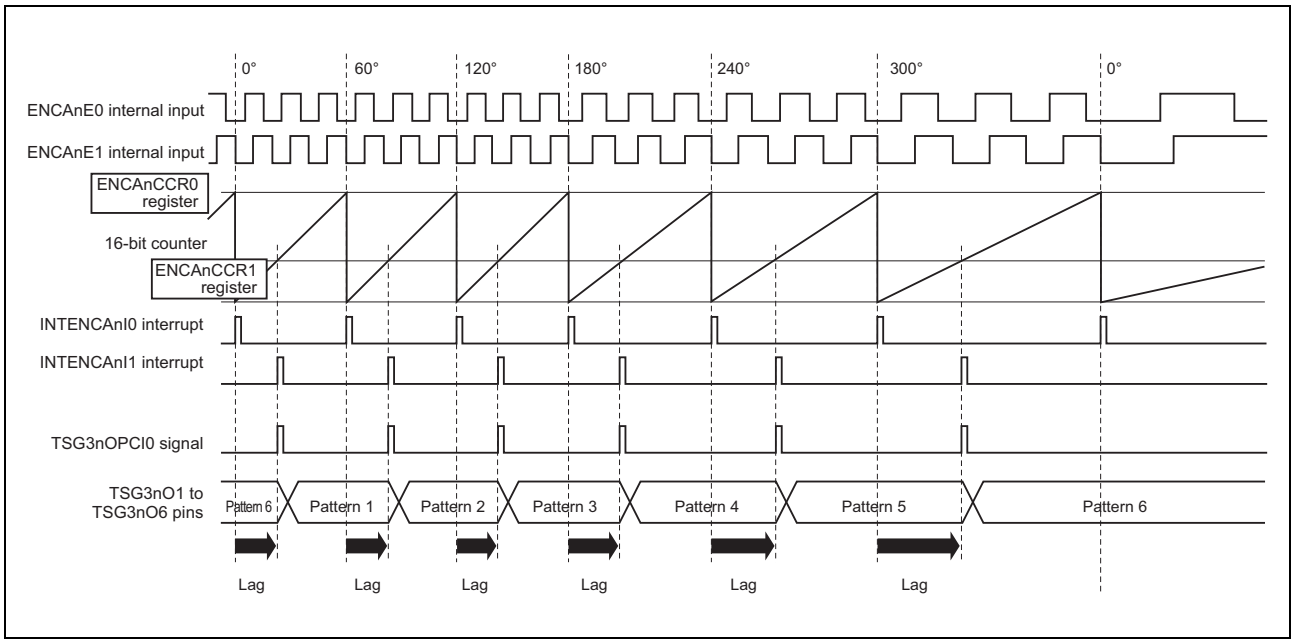


Figure 41.33 Timing Diagram of Two-Phase Encoder Control Function (Method 2) with Phase Lag (Normal Rotation)

By setting greater value to ENCAnCCR1 than that of ENCAnCCR0, TSG3n output pattern phase can be made to lag.

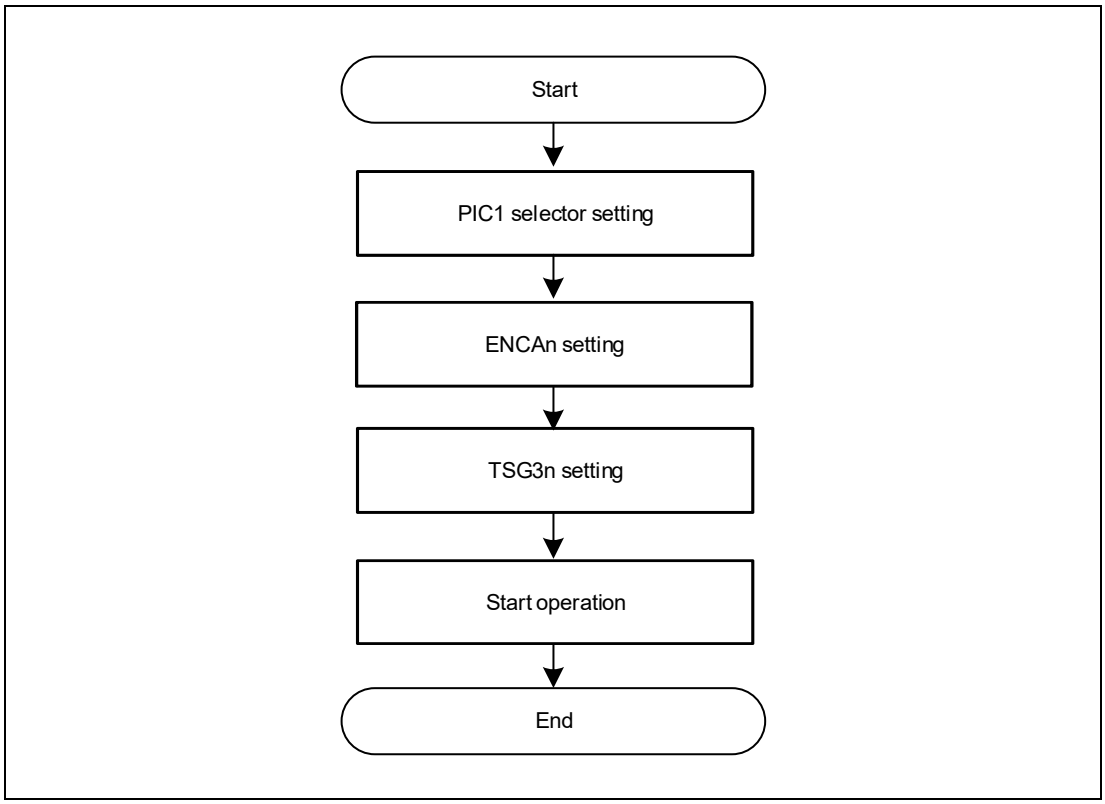
NOTE

This function can be used for phase lead and phase lag control both at up count and down count.

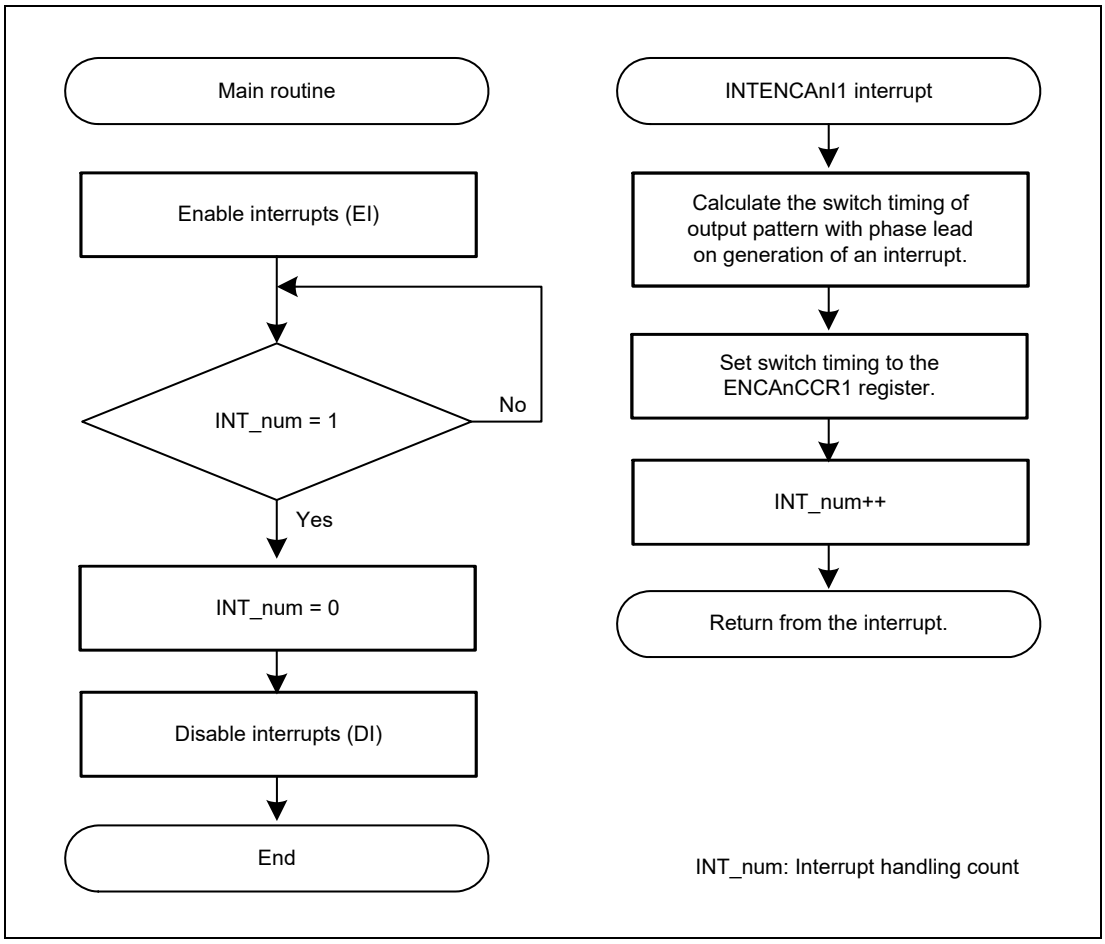
(5) Flow Chart

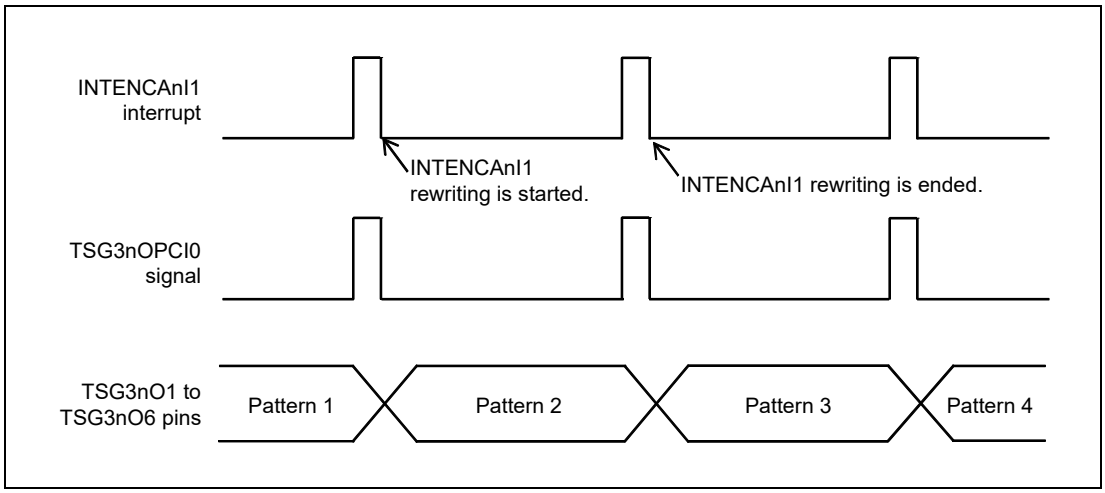
The flow charts for this function are shown below.

Flow chart of main routine.

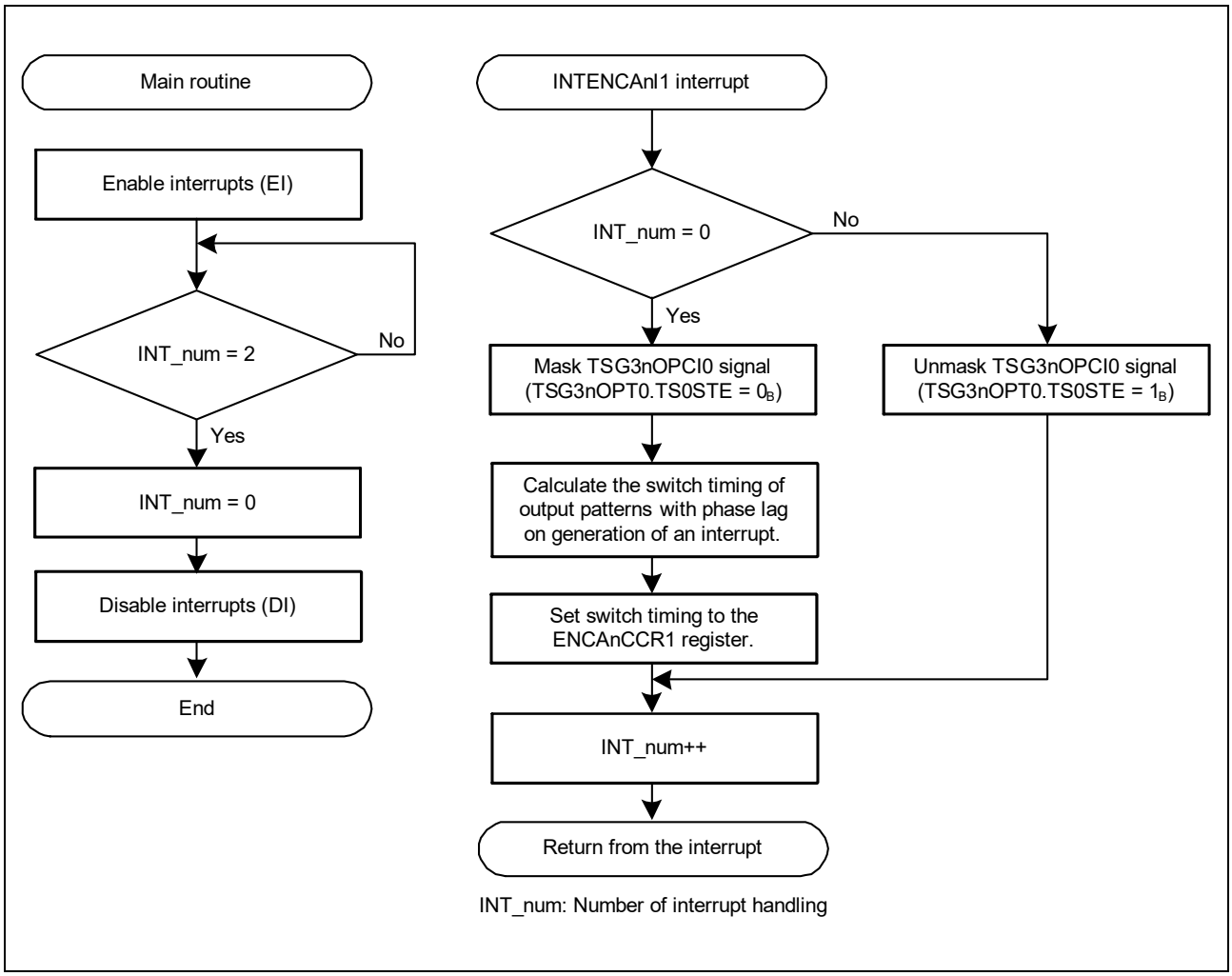


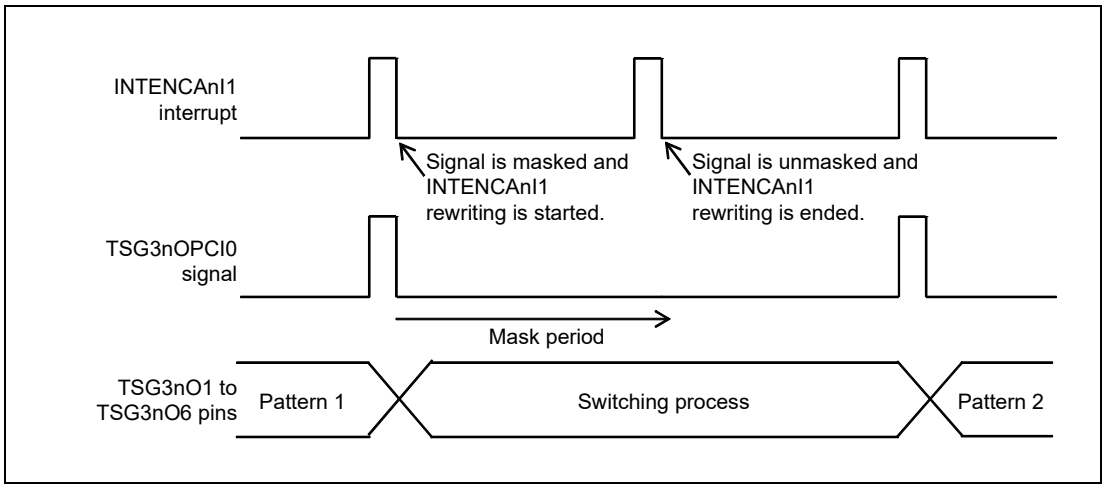
Flow chart of ENCAnCCR1 rewrite processing with phase lead.





Flow chart of ENCAnCCR1 rewrite processing with phase lag.





The register settings for ENCAn to use this function are as follows.

ENCAnCTL[15:0] = 1000 0000 000x $01xx_B$

ENCAnIOC1[7:0] = 0000 $00xx_B$

ENCAnCCR0 = (set any value)

ENCAnCCR1 = (set any value)

ENCAnCNT = (set any value)

"x" can be set arbitrarily. For the registers specifications, **Section 40, Encoder Timer A (ENCA)**.

The register settings for TSG3n to use this function are as follows.

TSG3nCTL0[7:0] = 000x 0011_B

TSG3nCTL3[7:0] =0000 $00xx_B$

TSG3nCTL4[15:0] = 0000 0001 xxx0 0000_B

TSG3nIOC0[7:0] = 0111 1110_B

TSG3nIOC2[15:0] = 0xxx xxx0 0000 0000_B

TSG3nOPT0[7:0] = 0011 $1xx0_B$

TSG3nOPT1[7:0] = 0000 $0xxx_B$

TSG3nCMP0 = (set any value)

TSG3nCMP1W, 5W, 9W = (set any value)

TSG3nCMP1, 5, 9 = (set any value)

TSG3nPAT0W, 1W = (set any value)

TSG3nDTC0W, 1W = (set any value)

"x" can be set arbitrarily. For the registers specifications, **Section 35, Motor Control Timer (TSG3)**.

41.2.3.10 Three-Phase Pulse Input Control Function

(1) Overview

This function allows variable phase control of TSG3n pattern output in 120-DC mode using TSG3n and TAUDn.

The following diagram shows the method of three-phase pulse input control.

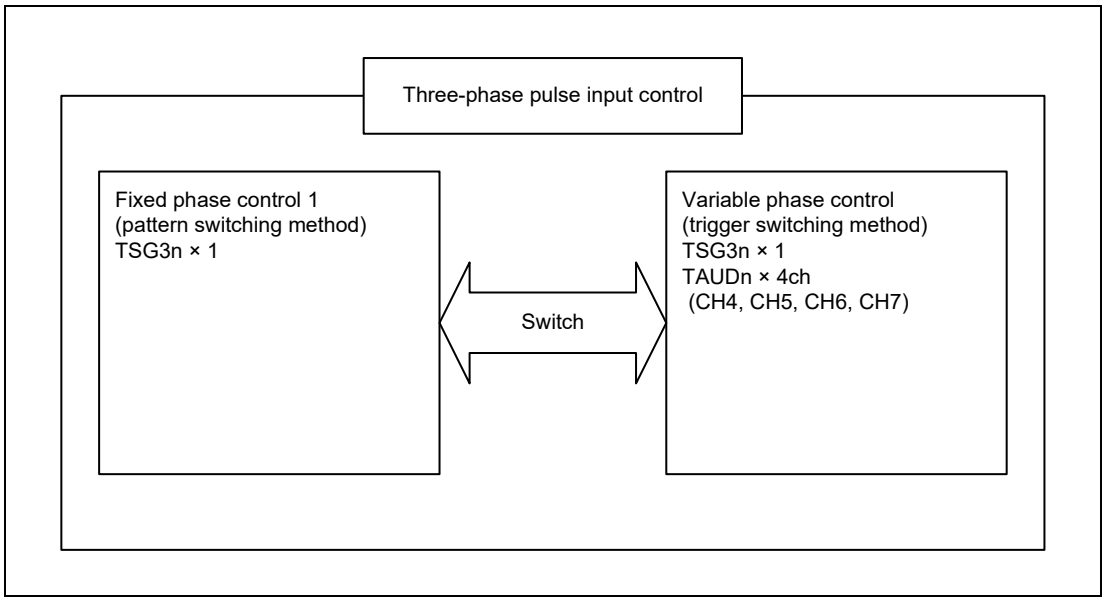


Figure 41.34 Control Method of Three-Phase Pulse Input Control Function

Control Method	Function
Fixed phase control 1 (pattern switching method)	Outputs a fixed pattern at constant rotation angle.
Variable phase control (trigger switching method)	Varies the phase by arbitrary angle (or time) up to ±60 degrees with reference to the rotation angle and outputs the pattern.

(2) Configuration

Three-phase pulse input control function is realized by using three-phase pulse input and TAUDn offset trigger mode, and PIC1 in combination.

The following figure shows the block diagram of three-phase pulse input control function.

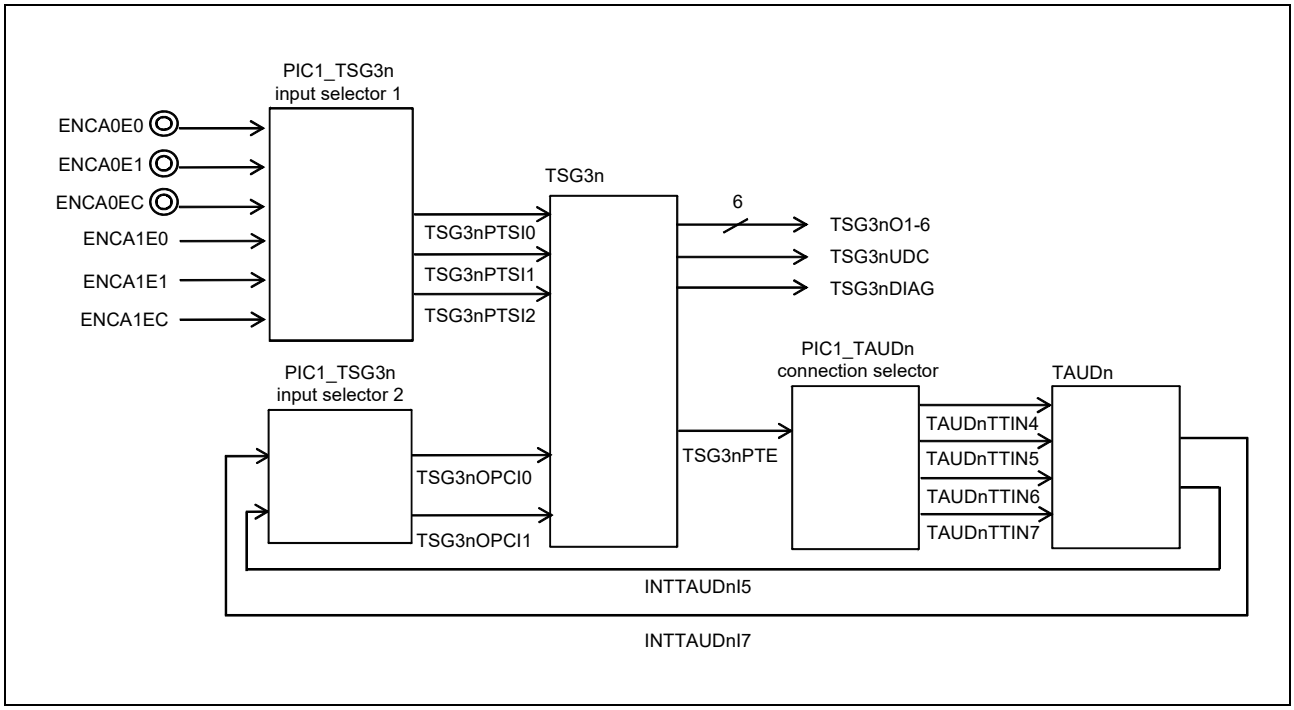


Figure 41.35 Block Diagram of Three-Phase Pulse Input Control

The configuration of this function is described below.

- [PIC1_TSG3n input selector 1]
  ENCAnE0, ENCAnE1, and ENCAnEC pin inputs are selected and output to TSG3nPTSI0 to TSG3nPTSI2.
- [TSG3n]
  Patterns set in TSG3nO1 to TSG3nO6 are output in response to the input of TSG3nPTSI0 to TSG3nPTSI2 signals. TSG3nPTE is toggled each time the output patterns are switched.
- [PIC1_TAUDn connection selector]
  TSG3nPTE input is selected and output to TAUDnTTIN4 to TAUDnTTIN7.
- [TAUDn]
  Interrupt signals INTTAUDnI5 and INTTAUDnI7 for output pattern phase generation are output with the offset trigger mode.
- [PIC1_TSG3n input selector 2]
  INTTAUDnI5 and INTTAUDnI7 inputs are selected and output as TSG3nOPCI0 and TSG3nOPCI1.

(3) Registers

Block diagram of PIC1 is shown in the following figure.

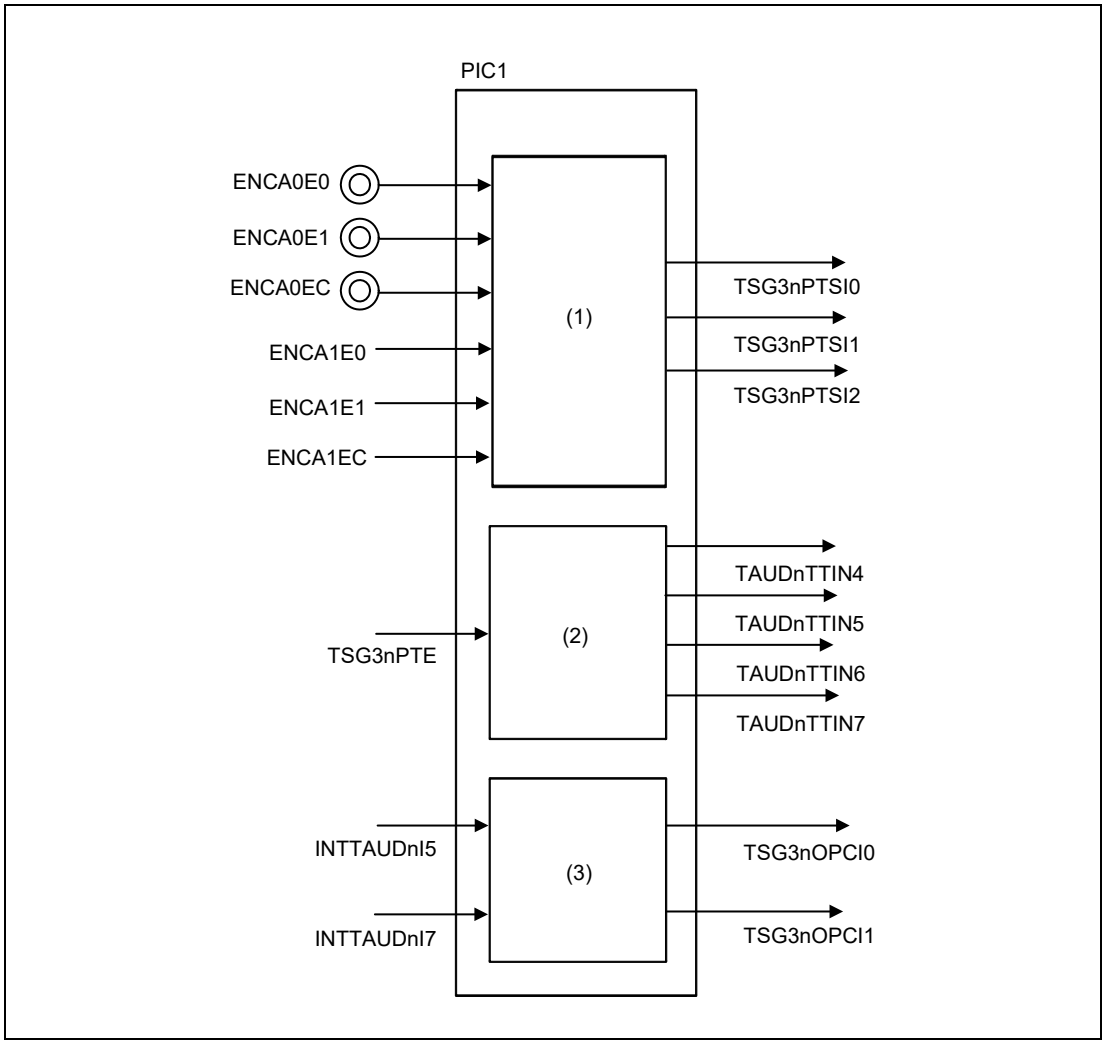


Figure 41.36 Block Diagram of PIC1

The values of PIC1 registers used in this function are as follows.

Unit (1): PIC1_TSG3n input selector 1

The values to output ENCA0E0, ENCA0E1, and ENCA0EC as TSG30PTSI0 to TSG30PTSI2

PIC1TSGHALLSEL[0] = 1_B

PIC1REG50[0] = 0_B

The values to output ENCA1E0, ENCA1E1, and ENCA1EC as TSG31PTSI0 to TSG31PTSI2

PIC1TSGHALLSEL[1] = 1_B

PIC1REG51[0] = 1_B

Unit (2): PIC1_TAUDn connection selector

The values to output TSG3nPTE to TAUDnTTIN4 to TAUDnTTIN7.

PIC1REG2n0[11:8] =1010_B

PIC1REG2n0[3:0] = 1111_B

PIC1TAUDnSEL[15:8] = 00_H

Unit (3): PIC1_ENCAn input selector 2

The values to output INTTAUDnI5 and INTTAUDnI7 to TSG3nOPCI0 and TSG3nOPCI1.

PIC1REG5n[7:5] = 011_B

(4) Function

Details of the three-phase pulse input control function are described here.

The following figure shows the timing diagram.

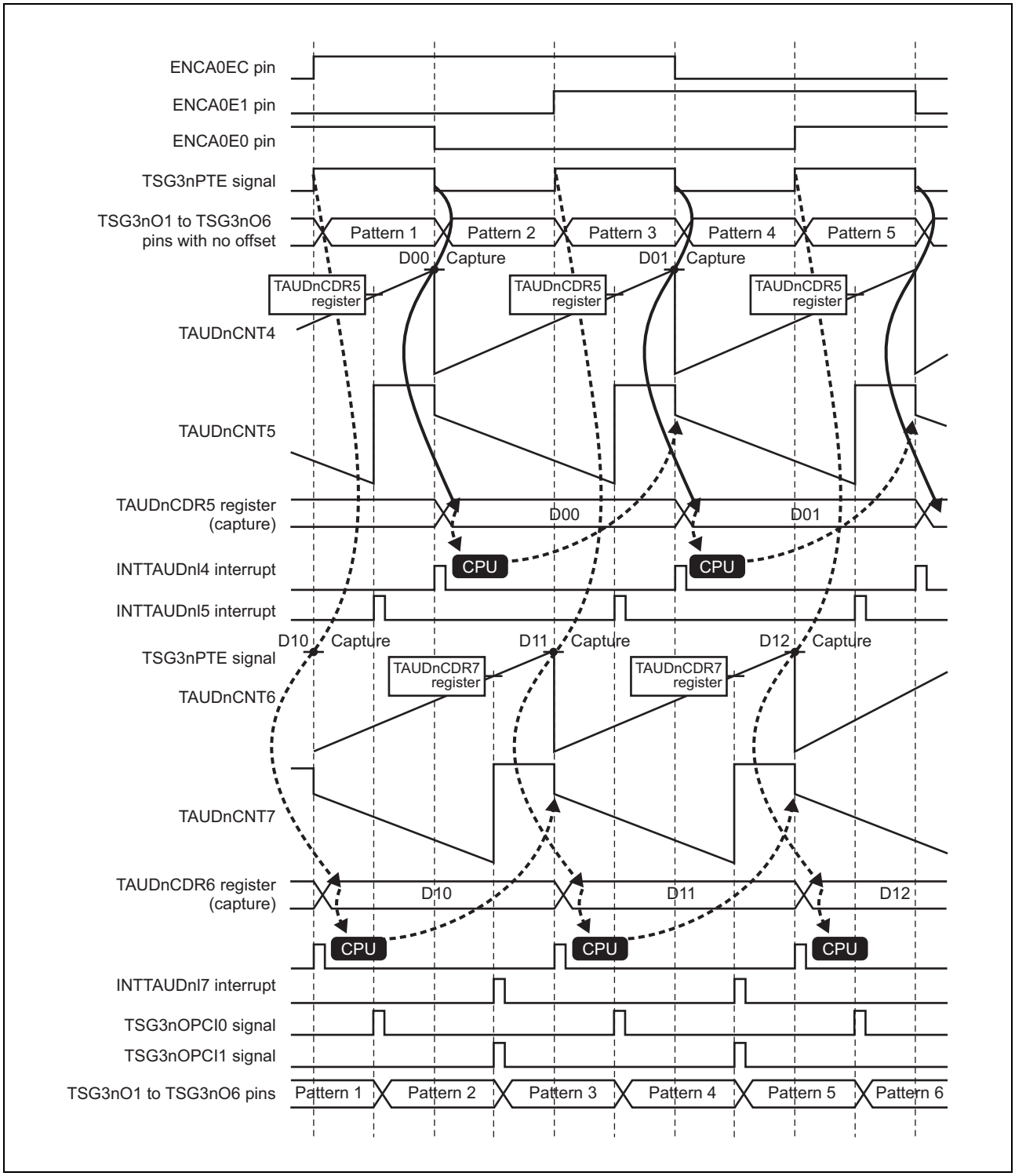


Figure 41.37 Three-Phase Pulse Input Control Function

(1) The output patterns are switched by the TSG pattern switching method on detection of ENCAn pin signal. The TSG3nPTE signal toggles each time the patterns are switched.

(2) TAUDnCNT6 and TAUDnCNT4 are captured at the rising edge and falling edge of the TSG3nPTE signal. The interval to switch patterns is calculated according to the captured value.

(3) CPU calculates the phase of the next output pattern and set the values to TAUDnCNT5 and TAUDnCNT7. The signals corresponding to the values are output as TSG3nOPCI0 and

TSG3nOPCI1. At this time, the patterns delayed for the set phase are output by switching the output patterns by the trigger switching method.

The following table lists the relation between the value set to TAUDnCNTm and the captured value in TAUDnCDR (n = 0, 1) (m = 5, 7).

TAUDnCNTm Register Setting	TSG3n Pattern Output Switch Timing
TAUDnCNTm = 0000_H	The patterns are switched on detection of an edge of TSG3nPTE signal with a delay of up to one cycle of the clock signal being counted by TAUDn is generated.
TAUDnCNTm = Captured value	The patterns are switched on detection of an edge of TSG3nPTE signal.
TAUDnCNTm < Captured value	The patterns are switched on the timing after the phase delayed from detection of an edge of TSG3nPTE signal.
TAUDnCNTm > Captured value	Setting prohibited

An example of switch operation from fixed phase control 1 to variable phase control

The following figure shows an example of switch operation from fixed phase control 1 to variable phase control.

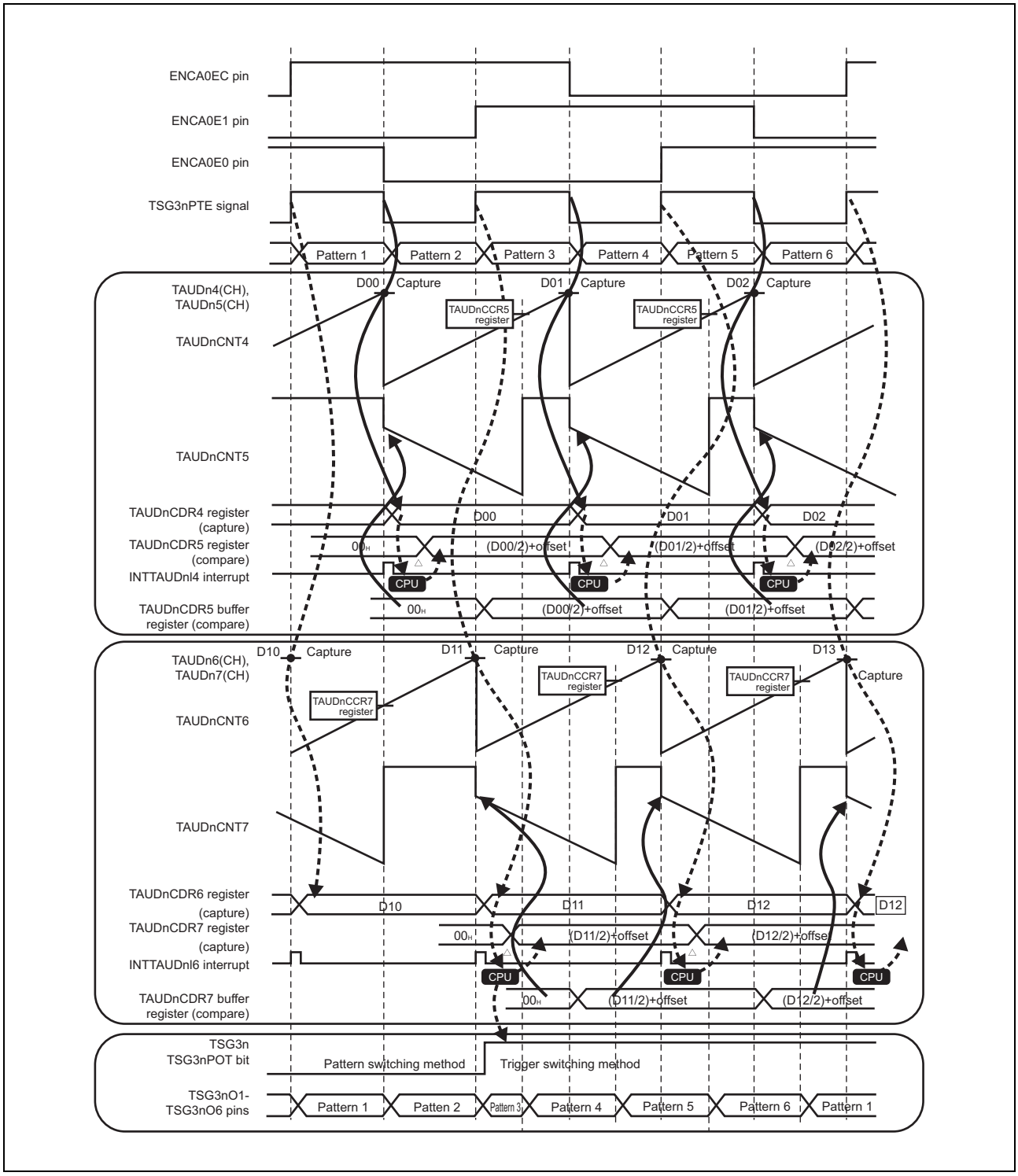


Figure 41.38 An Example of Switch Operation from Fixed Phase Control 1 to Variable Phase Control

By changing the TSG3nPOT bit from low level to high level, the output pattern switching method is changed to the trigger switching method, and the variable phase control is enabled.

An example of switch operation from variable phase control to fixed phase control 1

The following figure shows an example of switch operation from variable phase control to fixed phase control 1.

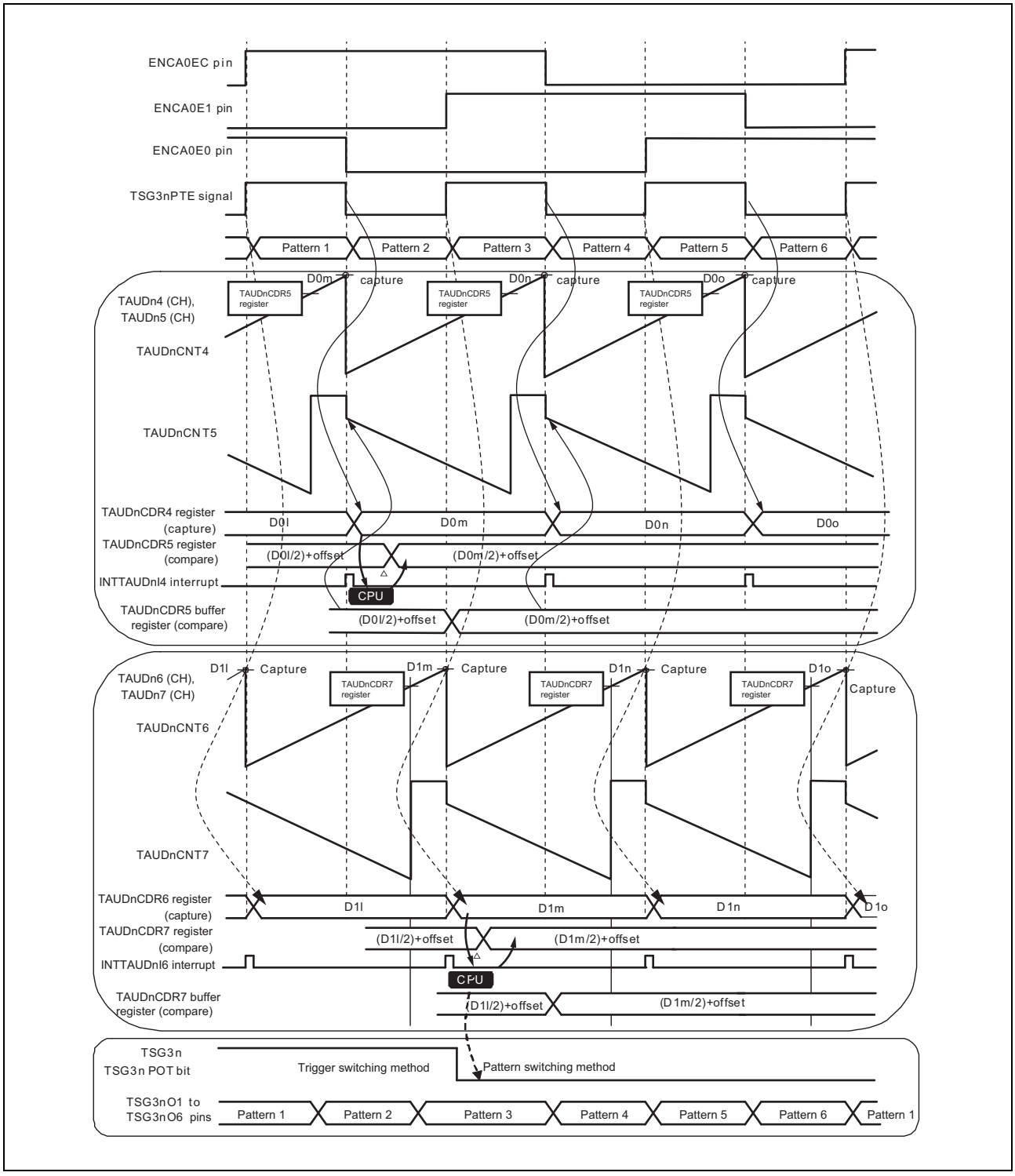


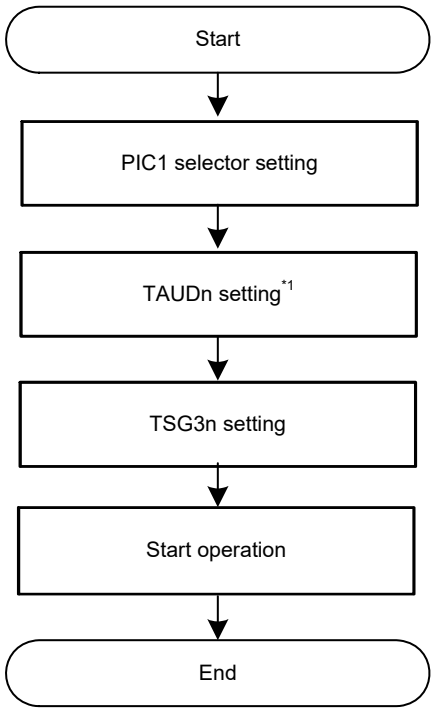
Figure 41.39 Example of Switch Operation from Variable Phase Control to Fixed Phase Control 1

By changing the TSG3nPOT bit from high level to low level, the output pattern switching method is changed to the pattern switching method, and the fixed phase control 1 is enabled.

(5) Flow Chart

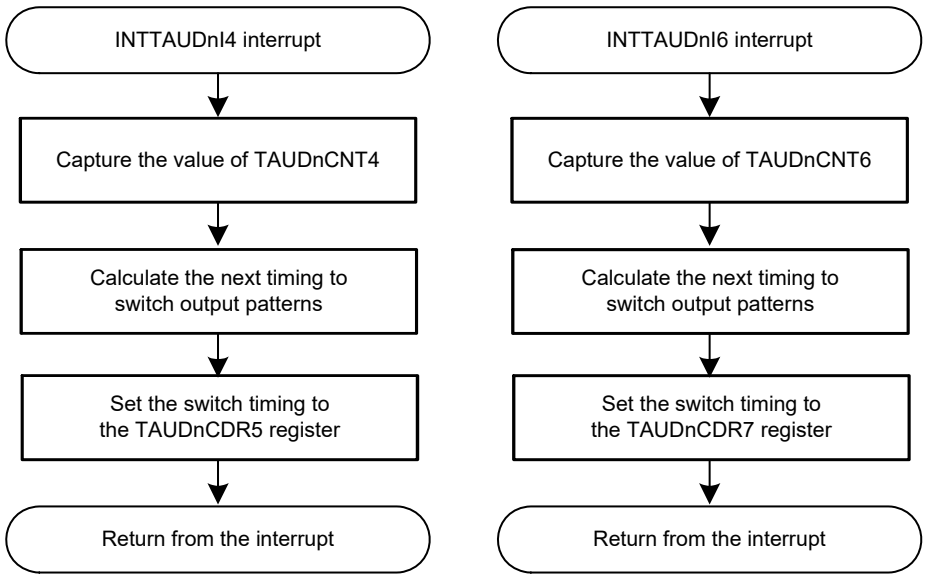
The flow charts for this function are shown below.

Flow chart of main routine



Note 1. For the settings of TAUDn, see **Section 33.15.5, Offset Trigger Output Function**.

Flow chart for rewriting of TAUDnCDR5 and TAUDnCDR7 during operation



Note: For the settings of TAUDn, see **Section 33.15.5, Offset Trigger Output Function**.

The values of TSG3n registers used in this function are as follows.

TSG3nCTL0[7:0] = $0000\ 0011_B$

TSG3nCTL3[7:0] = $0000\ 00xx_B$

TSG3nCTL4[15:0] = $0000\ 0001\ xxx0\ 0000_B$

TSG3nIOC0[7:0] = $0111\ 1110_B$

TSG3nIOC1[7:0] = $0001\ xxxx_B$

TSG3nIOC2[15:0] = $0xxx\ xxx0\ 0000\ 0000_B$

TSG3nOPT0[7:0] = $0011\ 1xx0_B$

TSG3nOPT1[7:0] = $0000\ 0xxx_B$

TSG3nCMP0 = (set any value)

TSG3nCMP1W,5W,9W = (set any value)

TSG3nCMP1,5,9 = (set any value)

TSG3nPAT0W,1W = (set any value)

TSG3nDTC0W,1W = (set any value)

"x" can be set arbitrarily. For the registers specifications, see **Section 35, Motor Control Timer (TSG3)**.

41.2.3.11 Three-Phase Encoder Control Function

(1) Overview

The function allows three-phase external pattern inputs (TSG3nPTSI0 to TSG3nPTSI2) to be encoded using ENCAn

(2) Configuration

The three-phase encoder control function is realized by using TSG3n, ENCAn, and PIC1 in combination. The following figure shows the block diagram of three-phase encoder control function.

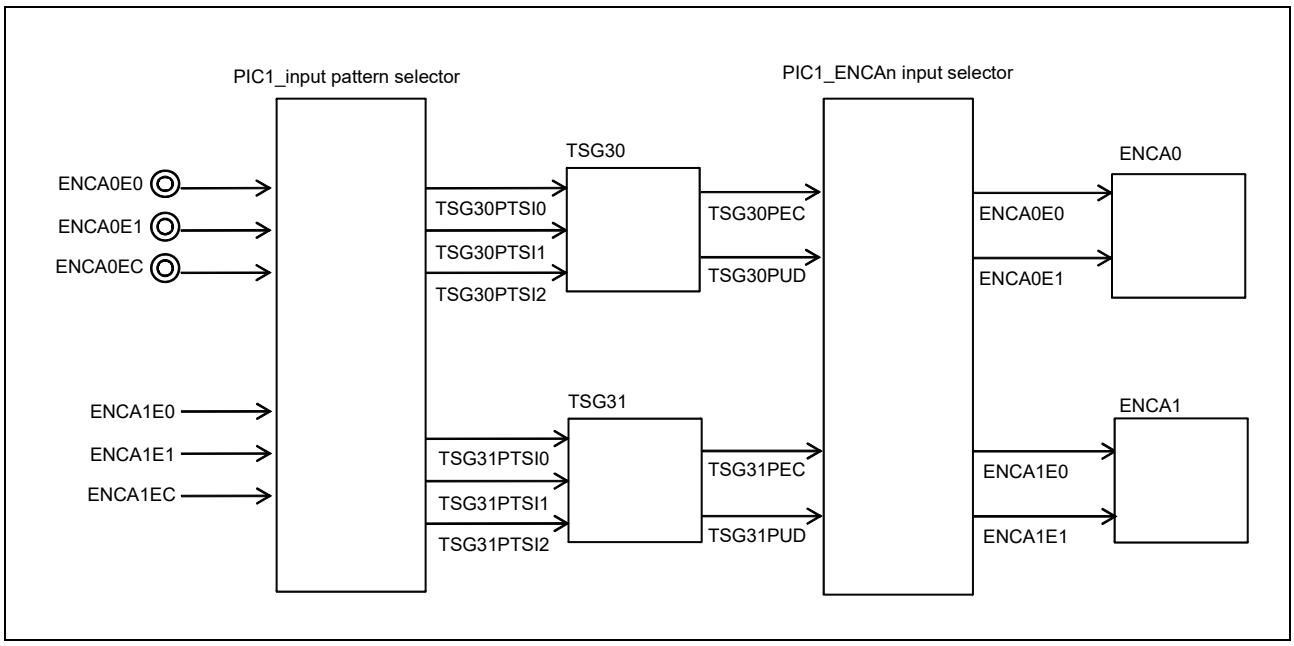


Figure 41.40 Block Diagram of Three-Phase Encoder Control Function

The configuration of this function is described below.

- [PIC1_input pattern selector]
  ENCAnE0, ENCAnE1, and ENCAnEC pin inputs are selected and output to TSG3nPTSI0 to TSG3nPTSI2.
- [TSG3n]
  Patterns set in TSG3nPEC are output in response to the input of the TSG3nPTSI0 to TSG3nPTSI2 signals. TSG3nPUD is output depending on the rotation (normal or reverse).
- [PIC1_ENCAn input selector]
  TSG3nPEC is selected and output to ENCAnE0. TSG3nPUD is selected and output to ENCAnE1.
- [ENCAn]
  ENCAnE0 and ENCAnE1 are encoded.

(3) Registers

Block diagram of PIC1 is shown in the following figure.

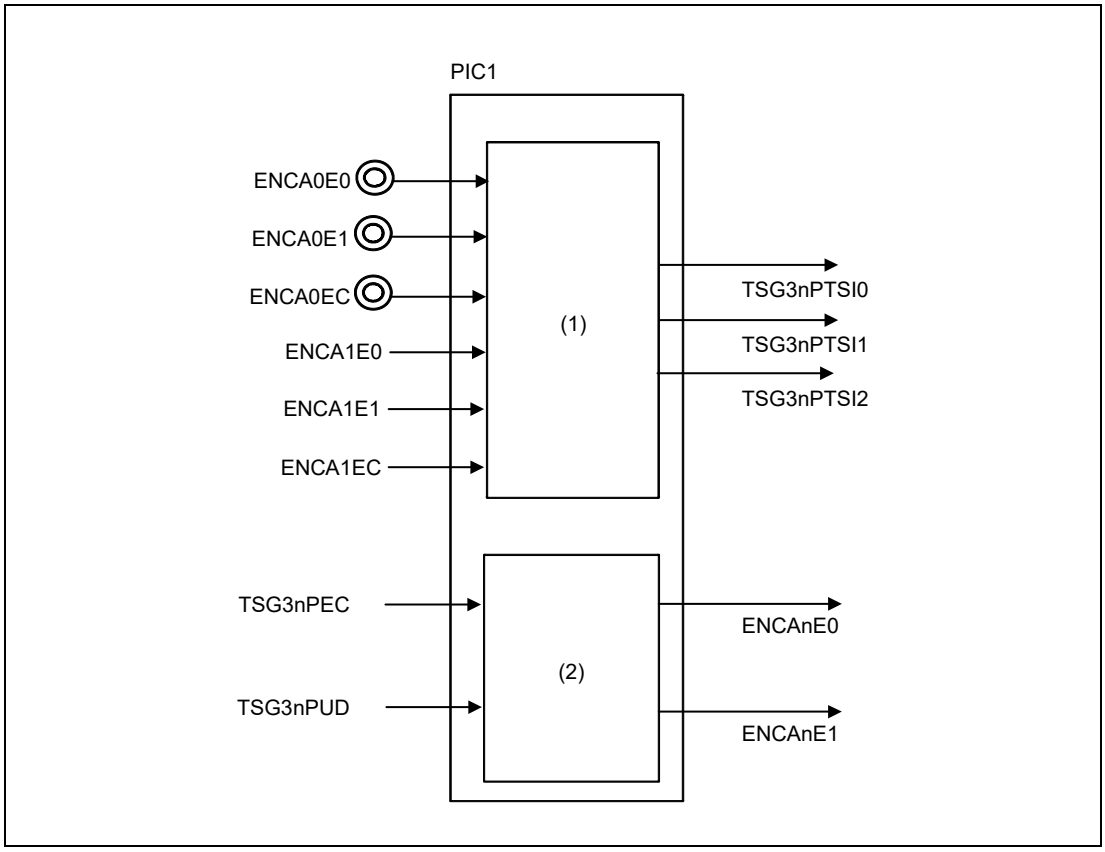


Figure 41.41 Block Diagram of PIC1

The register settings for PIC1 to use this function are as follows.

Unit (1): PIC1_ input pattern selector

The values to output ENCA0E0, ENCA0E1, and ENCA0EC as TSG30PTSI0 to TSG30PTSI2

PIC1TSGHALLSEL[0] = 1_B

PIC1REG50[0] = 0_B

The value to output ENCA1E0, ENCA1E1, and ENCA1EC as TSG31PTSI0 to TSG31PTSI2

PIC1TSGHALLSEL[1] = 1_B

PIC1REG51[0] = 1_B

Unit (2): PIC1_ENCAn input selector

The values to output the TSG30PEC and TSG30PUD as ENCA0E0 and ENCA0E1, respectively

PIC1REG30[22] = 0_B

PIC1REG30[1:0] = 11_B

The values to output the TSG31PEC and TSG31PUD as ENCA1E0 and ENCA1E1, respectively

PIC1REG30[9:8] = 10_B

PIC1REG30[7:6] = 10_B

(4) Function

Details of the three-phase encoder control function are described below.

The following figure shows the timing diagram.

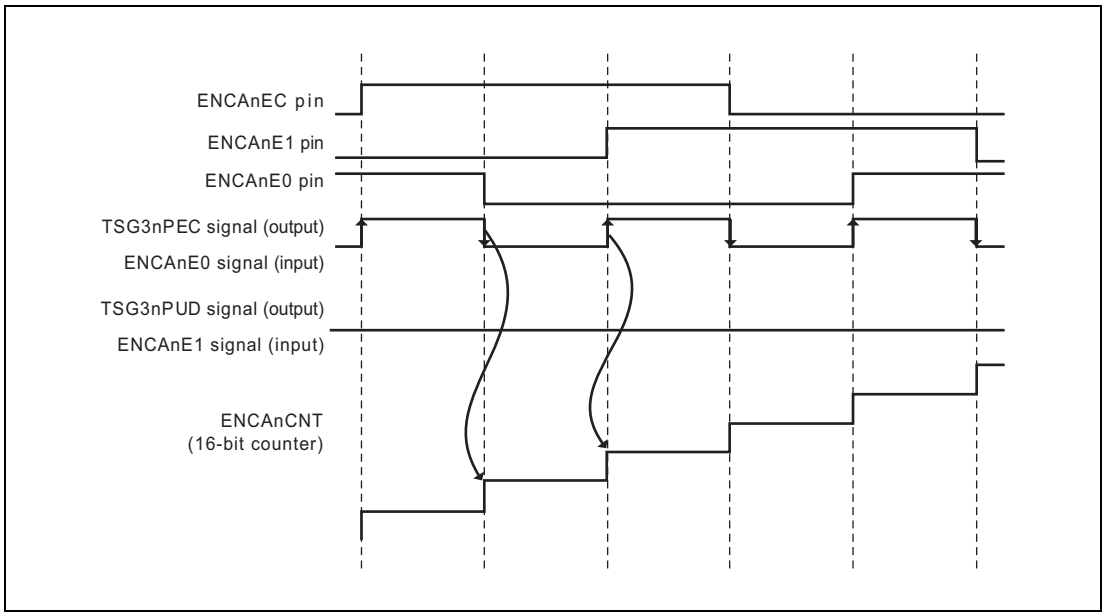


Figure 41.42 Timing Diagram of Three-Phase Encoder Control Function_ENCAnUDS1 and ENCAnUDS0 = 00_B

(1) When the low level is input to ENCAnE1, the count is incremented each time an active edge is input to ENCAnE0.

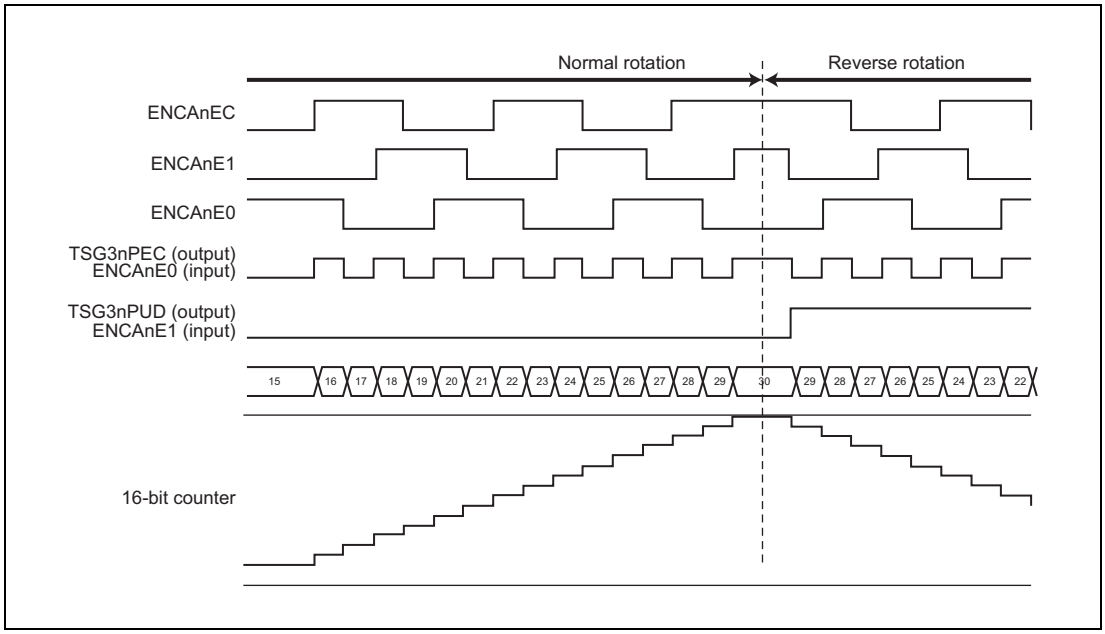


Figure 41.43 Timing Diagram of Three-Phase Encoder Control Function_Normal Rotation/Reverse Rotation

(5) Flow Chart

Set PIC before using the three-phase encoder control function.

The values of ENCAn registers used in this function are as follows.

ENCAnCTL[15:0] = xx00 00xx 000x $xx00_B$

ENCAnIOC1[7:0] = 0000 $00xx_B$*1

ENCAnCCR0 = (set any value)

ENCAnCCR1 = (set any value)

ENCAnCNT = (set any value)

"x" can be set arbitrarily. For the registers specifications, **Section 40, Encoder Timer A (ENCA)**.

Note 1. Except for ENCAnIOC1[1:0] = 00_B (no edge detected) because edge detection is necessary.

The values of TSG3n registers used in this function are as follows.

TSG3nCTL0[7:0] = 0000 0001_B

TSG3nCTL3[7:0] = 0000 $00xx_B$

TSG3nCTL4[15:0] = 0000 000x xxxx $xxxx_B$

TSG3nIOC0[7:0] = 0xxx $xxx0_B$

TSG3nIOC1[7:0] = 0001 $xxxx_B$

TSG3nIOC2[15:0] = 0xxx xxx0 0000 0000_B

TSG3nOPT0[7:0] = 0xxx $xxx0_B$

TSG3nOPT1[7:0] = 0000 $0xxx_B$

TSG3nCMP0 = (set any value)

TSG3nCMP1W,5W,9W = (set any value)

TSG3nCMP1,5,9 = (set any value)

TSG3nPAT0W,1W = (set any value)

TSG3nDTC0W,1W = (set any value)

"x" can be set arbitrarily. For the registers specifications, **Section 35, Motor Control Timer (TSG3)**.

41.2.3.12 Timer Input Select Function

(1) Overview

This function allows to select input signal for TAUDn and TAUJn.

(2) Configuration

TAUD0/TAUD1

The function selects TAUDn input signal to be input as TAUDnTTINm/m+1 signal from either TAUD0Im/m+1 signal or TAUD1Im/m+1 signal (n = 0, 1 and m is an even number between 0 and 15).

The TAUD input select function is realized by using TAUDn input signals and PIC1 in combination.

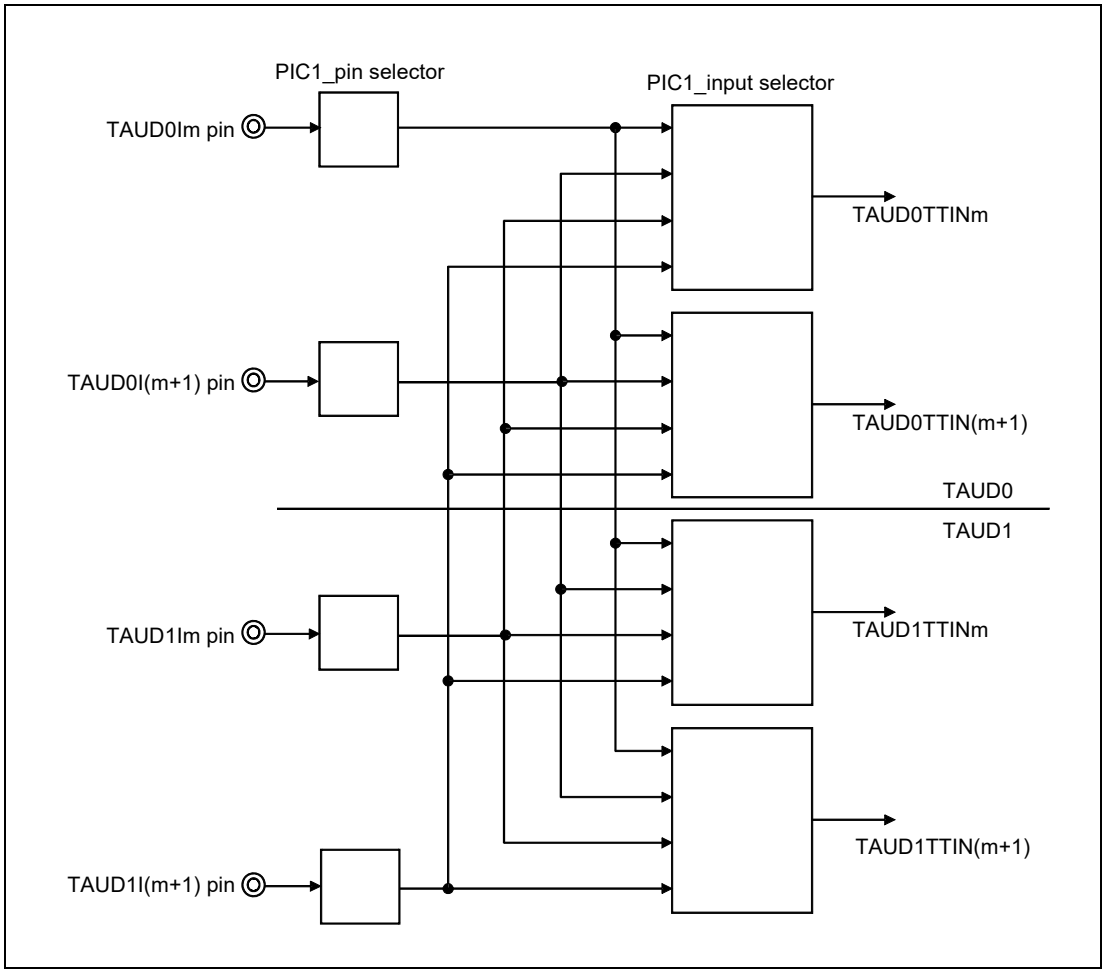


Figure 41.44 Block Diagram of TAUD Input Select Function

TAUD2

The output from port TAUD2Im (m = 0 to 15) can be input to TAUD2TTINm (m = 0 to 15) as shown in the following figure.

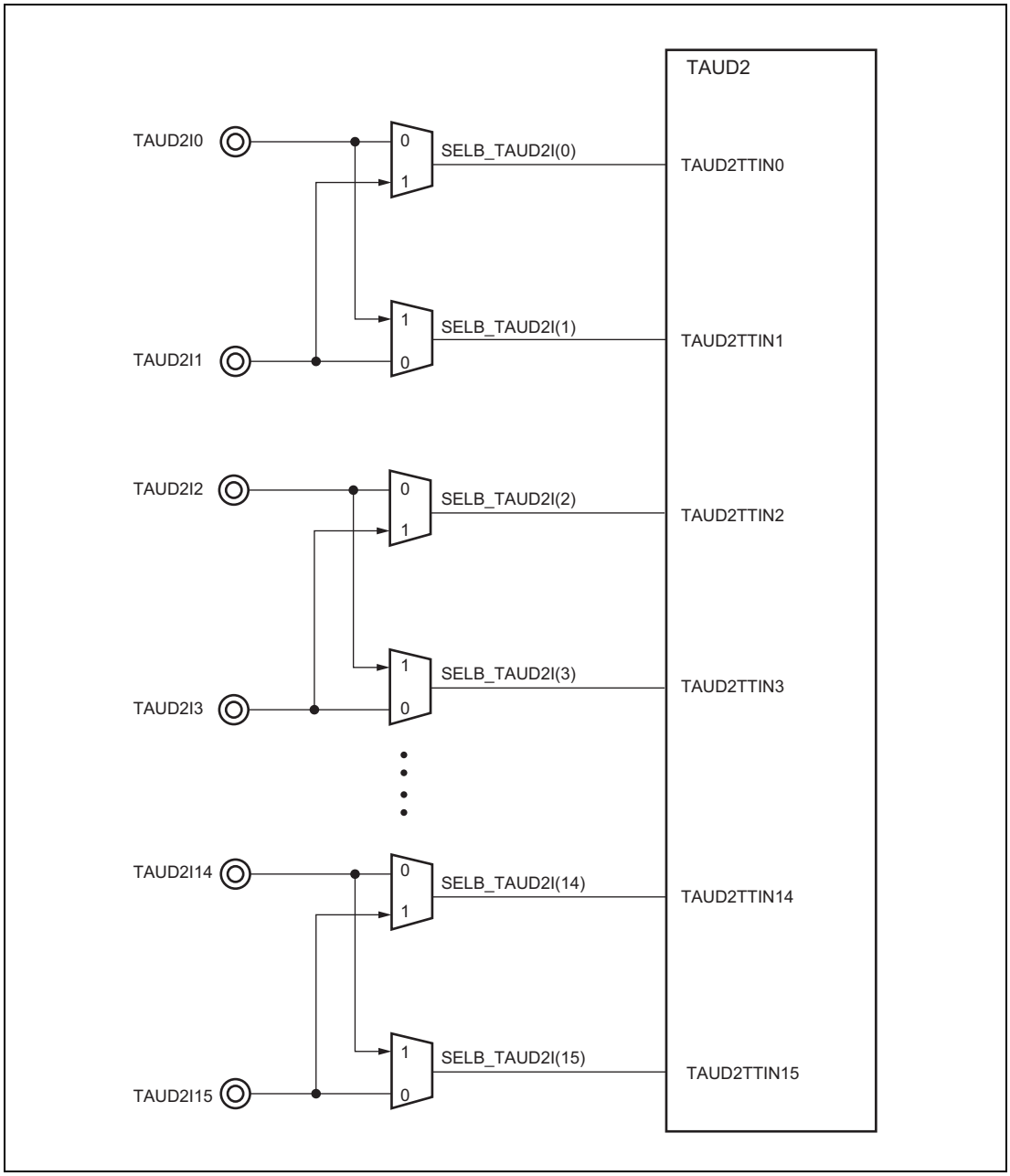


Figure 41.45 Selection of Signals Input to TAUD2

TAUJ2/TAUJ3

The output from TAUJ1 (TAUJ1TTOUT0) can be input to TAUJ2TTIN2 and TAUJ2TTIN3 as shown in the following figure.

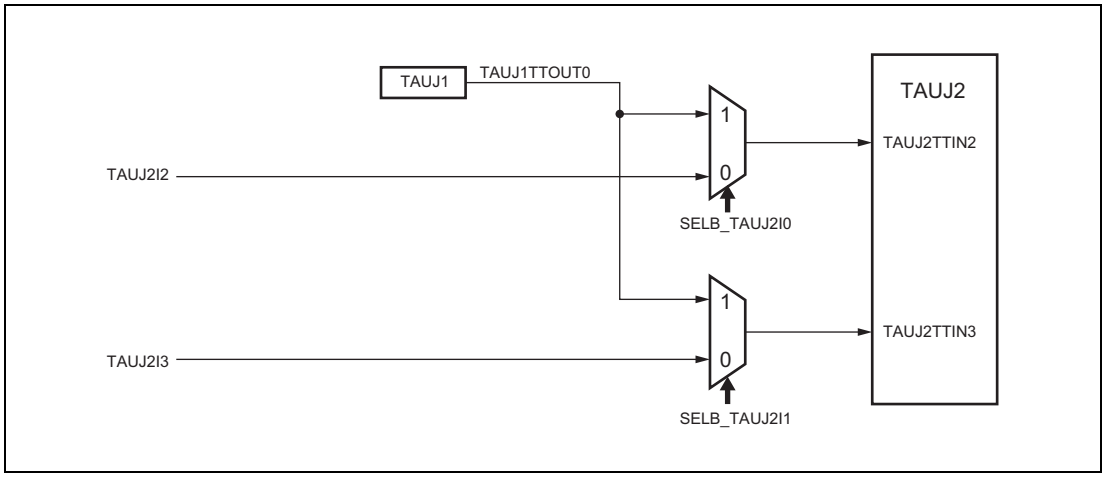


Figure 41.46 Selection of Signals Input to TAUJ2

The 1-Hz pulse output (RTCA0OUT) from RTCA0 and the output (TAUJ0TTOUT0) from TAUJ0 can be input to TAUJ3TTIN2 and TAUJ3TTIN3 as shown in the following figure.

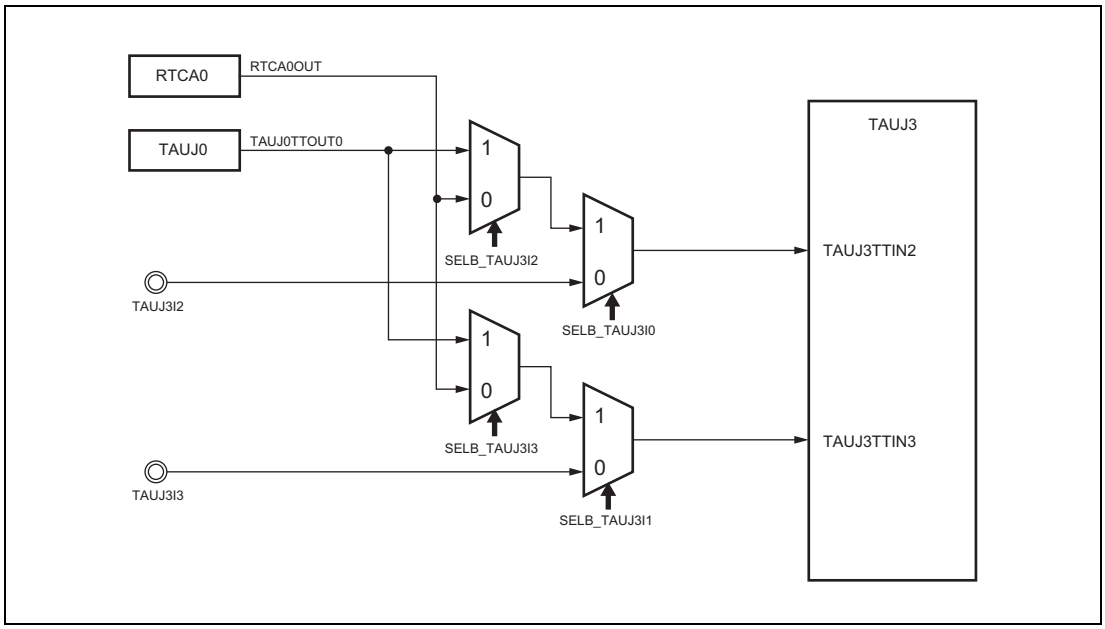


Figure 41.47 Selection of Signals Input to TAUJ3

(3) Registers

The register settings for PIC1 to use this function are described as follows.

PIC1_ pin selector

Select TAUDn external channel input pin for output from PIC1 pin selector.

PIC1_input selector

Select by the registers listed as follows.

PIC1TAUD0SEL

PIC1TAUD1SEL

PIC1REG2n0

PIC1REG2n2

PIC1REG31

For details of the register settings, **Section 41.2.2.10, Section 41.2.2.11, Section 41.2.2.18, Section 41.2.2.19, Section 41.2.2.21**, and **Section 41.2.2.24**.

(4) Function

Detail of this function is described using selection of the TAUD0TTIN[1:0] signal as an example.

The following table lists an example of selection of the TAUD0TTIN[1:0] signals. Setting 000000 to PIC1REG31[11:6], and setting 01 to PIC1TAUD0SEL[3:2] and PIC1TAUD0SEL[1:0] allows TAUD0TTIN0 and TAUD0TTIN1 signals to be input to TAUD0TTIN1 and TAUD0TTIN0 input pins of the TAUD0 timer. Setting 1 to PIC1TAUD0SEL[3] and PIC1TAUD0SEL[1] selects TIN pin signal of TAUD1.

Register Setting PIC1TAUD0SEL [1:0]	TAUD0TTIN0
00_B	TAUD0I0 pin
01_B	TAUD0I1 pin
10_B	TAUD1I0 pin
11_B	TAUD1I1 pin

Register Setting PIC1TAUD0SEL [3:2]	TAUD0TTIN1
00_B	TAUD0I1 pin
01_B	TAUD0I0 pin
10_B	TAUD1I1 pin
11_B	TAUD1I0 pin

The following table shows the method of selecting input signals to several TAUD2 inputs.

Table 41.69 TAUD2 Input Selection

Input Signal	Function	Settings
TAUD2TTIN[m]	Port TAUD2I[m]	SELB_TAUD2I[m] = 0
	Port TAUD2I[m+1]	SELB_TAUD2I[m] = 1
TAUD2TTIN[m+1]	Port TAUD2I[m+1]	SELB_TAUD2I[m+1] = 0
	Port TAUD2I[m]	SELB_TAUD2I[m+1] = 1

The following table shows how to select signals input to the TAUJ.

Table 41.70 TAUJ2 Input Selections

Input Signal	Function	Settings	
		SELB_ TAUJ2I1	SELB_ TAUJ2I0
TAUJ2TTIN2	Port TAUJ2I2	—	0
	TAUJ1TTOUT0	—	1
TAUJ2TTIN3	Port TAUJ2I3	0	—
	TAUJ1TTOUT0	1	—

Table 41.71 TAUJ3 Input Selections

Input Signal	Function	Settings			
		SELB_ TAUJ3I3	SELB_ TAUJ3I2	SELB_ TAUJ3I1	SELB_ TAUJ3I0
TAUJ3TTIN2	Port TAUJ3I2	—	—	—	0
	RTCA0OUT (Real-time clock 1-Hz output)	—	0	—	1
	TAUJ0TTOUT0	—	1	—	1
TAUJ3TTIN3	Port TAUJ3I3	—	—	0	—
	RTCA0OUT (Real-time clock 1-Hz output)	0	—	1	—
	TAUJ0TTOUT0	1	—	1	—

(5) Flow Chart

Set PIC1 before starting the TAUDn timer.

41.2.3.13 Hi-Z Control Function

(1) Overview

The function disconnects three-phase output signal and changes to Hi-Z state.

For details of the purpose and operation of the Hi-Z control function, see **Section 36.4.1, Asynchronous Hi-Z Control Function**.

(2) Configuration

The TAPAnESO, ERROROUTZ, INTTSG30IER, INTTSG31IER, WDTBnTNMI (n=0, 1, 2, 3), INTADCJ0ERR, and INTADCJ1ERR signals are masked and OR'ed in PIC1, and output to TAPAn as the signal for Hi-Z control.

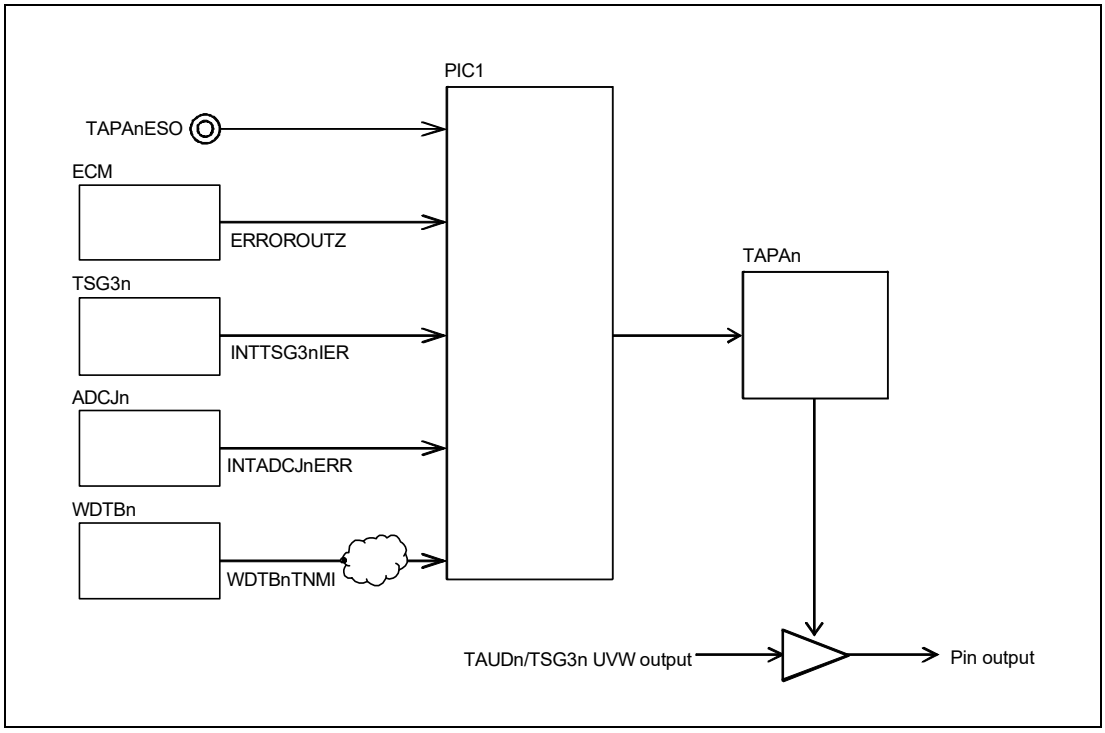


Figure 41.48 Block Diagram of Hi-Z Control

(3) Registers

The block diagram of PIC1 is shown in the following figure.

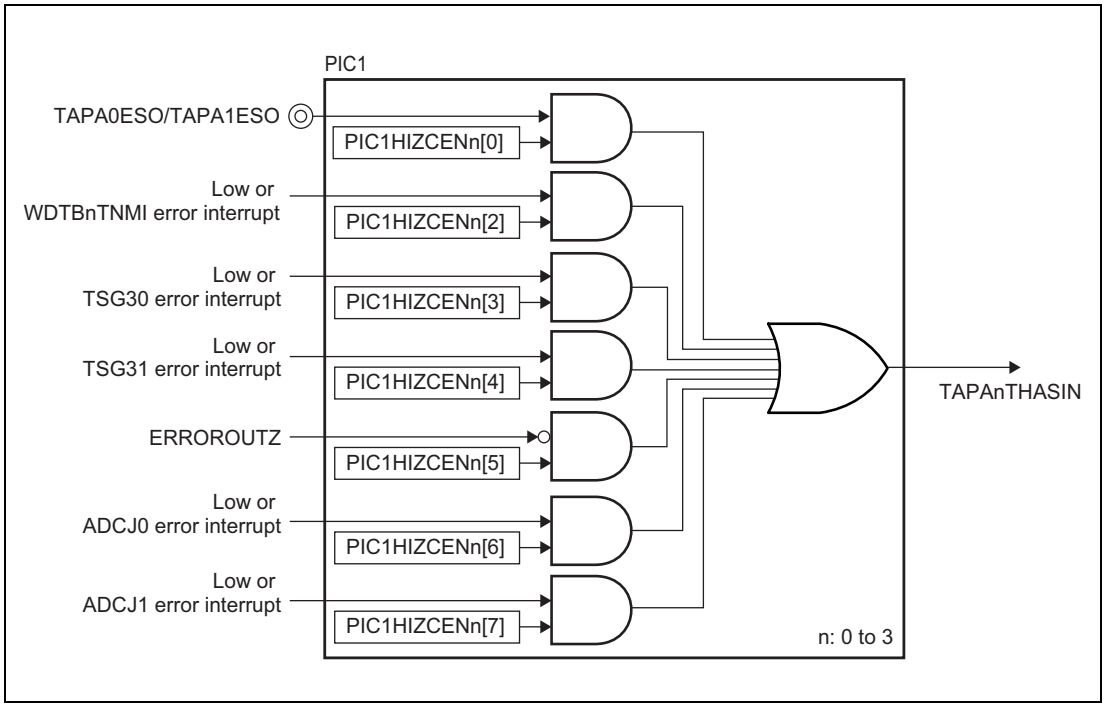


Figure 41.49 Block Diagram of PIC1

The register settings for PIC1 to use this function are as follows.

PIC1HIZCENn[5] = 1 (enabled), 0 (disabled): ERROROUTZ

PIC1HIZCENn[0] = 1 (enabled), 0 (disabled): TAPA0ESO or TAPA1ESO

PIC1HIZCENn[2] = 1 (enabled), 0 (disabled): WDTBnTNMI

PIC1HIZCENn[3] = 1 (enabled), 0 (disabled): INTTSG30IER

PIC1HIZCENn[4] = 1 (enabled), 0 (disabled): INTTSG31IER

PIC1HIZCENn[6] = 1 (enabled), 0 (disabled): INTADCJ0ERR

PIC1HIZCENn[7] = 1 (enabled), 0 (disabled): INTADCJ1ERR

n = 0, 1 is available for TAUD0/TAUD1 and not available for PIC1HIZCENn[3] and PIC1HIZCENn[4].

n = 2 is available for TSG30 and not available for PIC1HIZCENn[4].

n = 3 is available for TSG31 and not available for PIC1HIZCENn[3].

Table 41.72 Correspondence between the Hi-Z Control Function Input Pins and Control Pins

Input Pin Name	Control Register	TAPA Unit Number	Target Timers for Hi-Z Control
TAPA0ESO	PIC1HIZCEN0	TAPA0	TAUD0
TAPA1ESO	PIC1HIZCEN1	TAPA1	TAUD1
TAPA0ESO	PIC1HIZCEN2	TAPA2	TSG30
TAPA1ESO	PIC1HIZCEN3	TAPA3	TSG31

(4) Function

The TAPA0ESO/TAPA1ESO pins, WDTBnTNMI, ERROROUTZ, INTTSG30IER, INTTSG31IER, INTADCJ0ERR, and INTADCJ1ERR for Hi-Z control are masked and OR'ed in PIC1, and output to TAPAn. For Hi-Z control by TAPA, see **Section 36.4.2.1, Basic Operation**.

(5) Flow Chart

Set PIC1 before starting Hi-Z control.

For the operation flow of TAPA, see **Section 36.4.1, Asynchronous Hi-Z Control Function**.

41.2.3.14 Timer Output Monitor Function

(1) Overview

With this function, output signals of TAUD0, TAUD1, TSG30, TSG31, OSTM8, and OSTM9 are monitored using TAUD2.

(2) Configuration

Monitoring of output signals of TAUD0, TAUD1, TSG30, TSG31, OSTM8, and OSTM9 is realized using TAUD2 and PIC1 in combination.

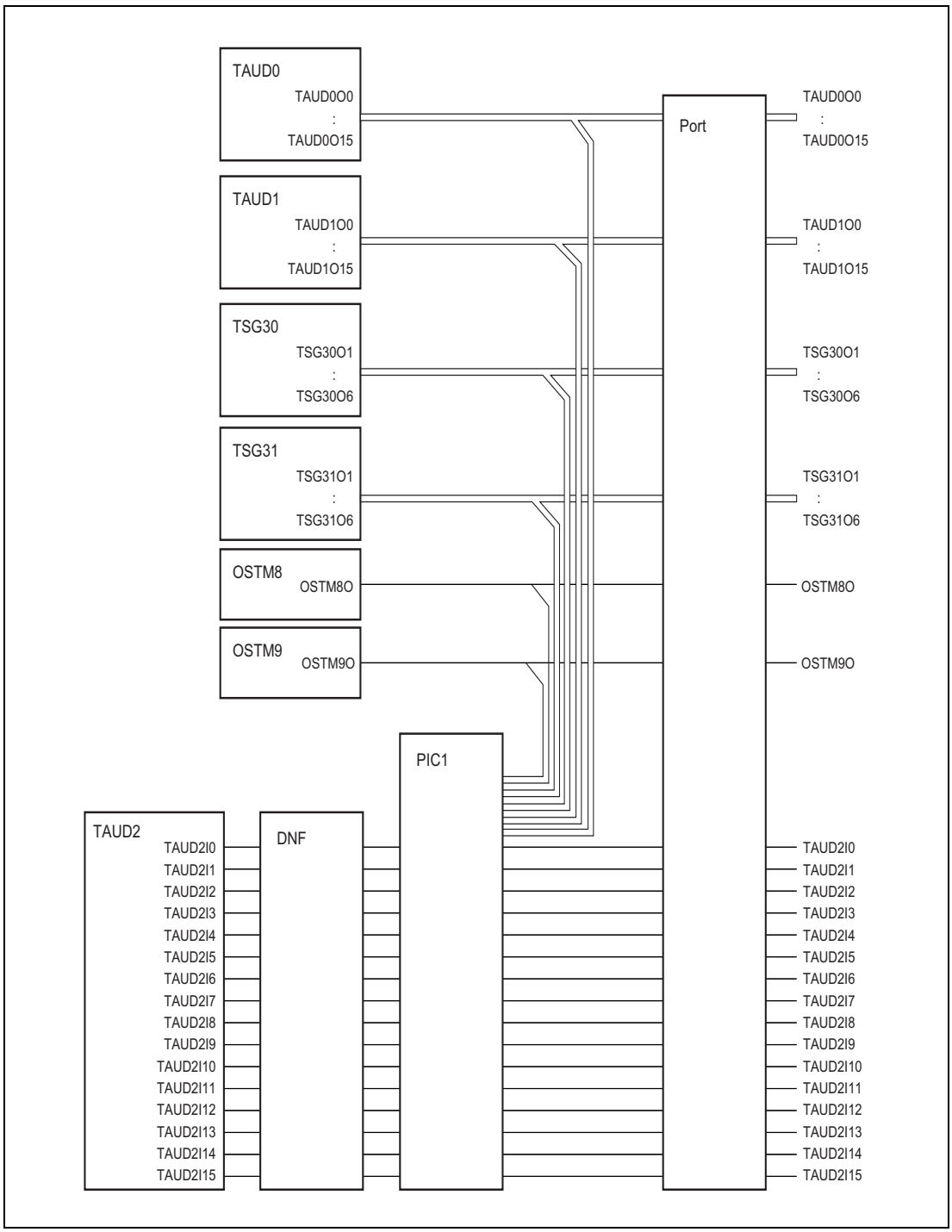


Figure 41.50 Block Diagram of Timer Output Monitor Function

(3) Registers

- The PIC1POMONSEL register selects the output signal to be monitored.
- Set the alternative output level loopback function for the port of the output signal to be monitored. For details, see **Section 2.5.14, PBDCn/APBDCn/JPBDC0 — Port Bi-Direction Control Register**.

41.2.3.15 Timer Input Monitor Function

(1) Overview

With this function, input signals of TAUJ0 and TAUJ1 are monitored using TAUJ2.

(2) Configuration

Monitoring of input signals of TAUJ0 and TAUJ1 is realized using TAUJ2 and PIC1 in combination.

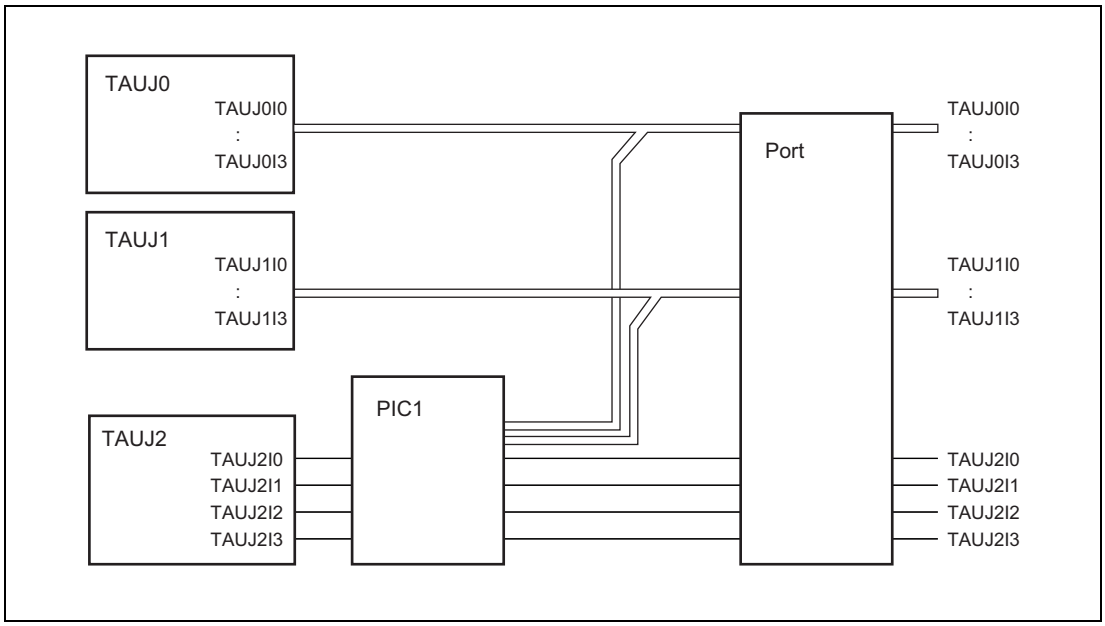


Figure 41.51 Block Diagram of Timer Input Monitor Function

(3) Registers

The PIC1PIMONSEL register selects the input signals to be monitored.

41.2.3.16 TSG3 Synchronous Start and Clear Function

(1) Overview

This function allows synchronous starting and clearing of TSG30 and TSG31.

(2) Configuration

The interrupt signals from TAUD0 channels 14 and 15 are used as synchronous start triggers to initiate synchronous starting or clearing of TSG30 and TSG31.

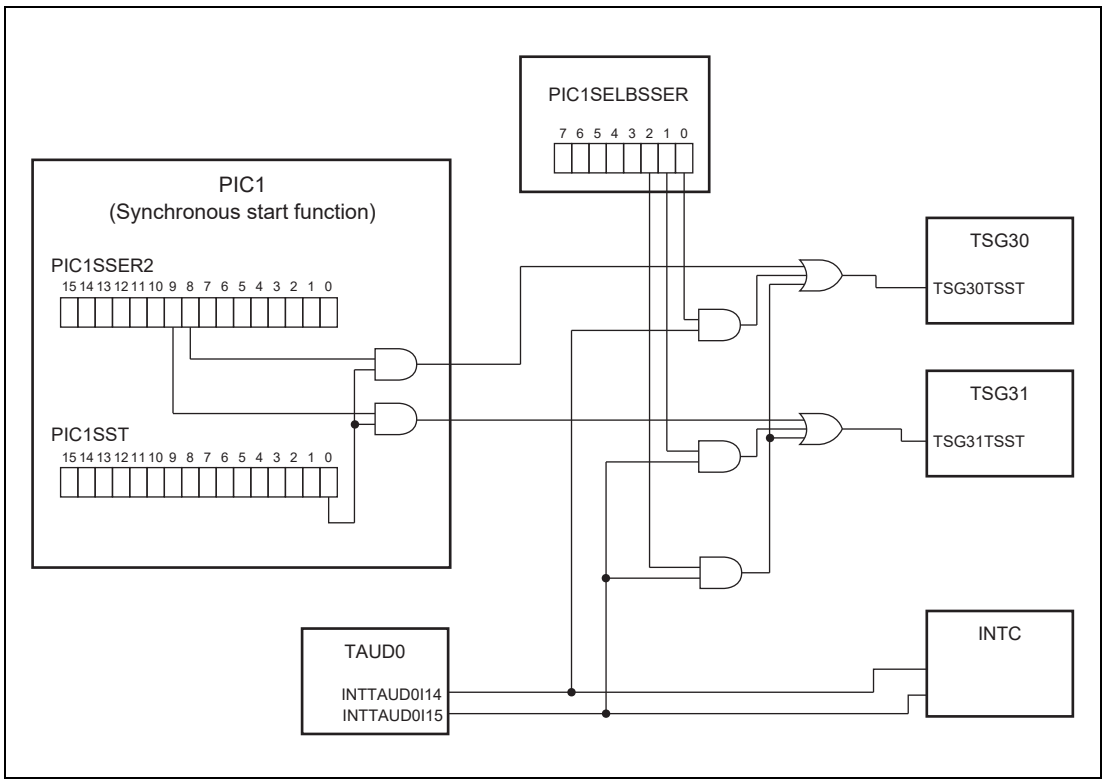


Figure 41.52 Block Diagram of TSG3 Synchronous Clear Function

41.3 Peripheral Interconnect — 2 (PIC2)

41.3.1 Overview

41.3.1.1 Functional Overview

Peripheral interconnect-2 (PIC2) handles configuration of triggers to ADCJn, ENCAn, PSI5-Sn (n = 0, 1) and GTM from various peripheral modules.

The PIC2 has following functions.

- ADCJ Trigger Select Function
- Baud rate measurement for an UART (RLIN3)
- Hi-Z control function over external pin for GTM output
- GTM output monitor for PWM diagnostic
- ENCA Trigger Selection Function
- PSI5-S Timestamp and the Sync Pulse Signal Selection Function
- GTM Timer Input (TIM) Selection Function
- ENCA Encoder Input Selection

41.3.2 Registers

41.3.2.1 List of Registers

PIC2 registers are listed in the following table.

Table 41.73 List of Registers (1/3)

Register Name	Symbol	Address	Access size	Access Protection	
				PBG	Other
ADC trigger selection control register					
A/D converter 0 trigger select control register 0	PIC2ADCJ0TSEL0	<PIC20_base> + 000_H	16, 32	PBG52#1	—
A/D converter 0 trigger select control register 1	PIC2ADCJ0TSEL1	<PIC20_base> + 004_H	16, 32	PBG52#1	—
A/D converter 0 trigger select control register 2	PIC2ADCJ0TSEL2	<PIC20_base> + 008_H	16, 32	PBG52#1	—
A/D converter 0 trigger select control register 3	PIC2ADCJ0TSEL3	<PIC20_base> + $00C_H$	16, 32	PBG52#1	—
A/D converter 0 trigger select control register 4	PIC2ADCJ0TSEL4	<PIC20_base> + 010_H	16, 32	PBG52#1	—
A/D converter 0 trigger edge select control register	PIC2ADCJ0EDGSEL	<PIC20_base> + $01C_H$	16	PBG52#1	—
A/D converter 1 trigger select control register 0	PIC2ADCJ1TSEL0	<PIC20_base> + 020_H	16, 32	PBG52#1	—
A/D converter 1 trigger select control register 1	PIC2ADCJ1TSEL1	<PIC20_base> + 024_H	16, 32	PBG52#1	—
A/D converter 1 trigger select control register 2	PIC2ADCJ1TSEL2	<PIC20_base> + 028_H	16, 32	PBG52#1	—
A/D converter 1 trigger select control register 3	PIC2ADCJ1TSEL3	<PIC20_base> + $02C_H$	16, 32	PBG52#1	—
A/D converter 1 trigger select control register 4	PIC2ADCJ1TSEL4	<PIC20_base> + 030_H	16, 32	PBG52#1	—
A/D converter 1 trigger edge select control register	PIC2ADCJ1EDGSEL	<PIC20_base> + $03C_H$	16	PBG52#1	—
A/D converter trigger output control register 400	PIC2ADTEN400	<PIC20_base> + 080_H	16	PBG52#1	—
A/D converter trigger output control register 401	PIC2ADTEN401	<PIC20_base> + 084_H	16	PBG52#1	—
A/D converter trigger output control register 402	PIC2ADTEN402	<PIC20_base> + 088_H	16	PBG52#1	—
A/D converter trigger output control register 403	PIC2ADTEN403	<PIC20_base> + $08C_H$	16	PBG52#1	—
A/D converter trigger output control register 404	PIC2ADTEN404	<PIC20_base> + 090_H	16	PBG52#1	—
A/D converter trigger output control register 410	PIC2ADTEN410	<PIC20_base> + $0A0_H$	16	PBG52#1	—
A/D converter trigger output control register 411	PIC2ADTEN411	<PIC20_base> + $0A4_H$	16	PBG52#1	—
A/D converter trigger output control register 412	PIC2ADTEN412	<PIC20_base> + $0A8_H$	16	PBG52#1	—
A/D converter trigger output control register 413	PIC2ADTEN413	<PIC20_base> + $0AC_H$	16	PBG52#1	—
A/D converter trigger output control register 414	PIC2ADTEN414	<PIC20_base> + $0B0_H$	16	PBG52#1	—
A/D converter trigger output control register 420	PIC2ADTEN420	<PIC20_base> + $0C0_H$	16	PBG52#1	—
A/D converter trigger output control register 421	PIC2ADTEN421	<PIC20_base> + $0C4_H$	16	PBG52#1	—
A/D converter trigger output control register 422	PIC2ADTEN422	<PIC20_base> + $0C8_H$	16	PBG52#1	—
A/D converter trigger output control register 423	PIC2ADTEN423	<PIC20_base> + $0CC_H$	16	PBG52#1	—
A/D converter trigger output control register 424	PIC2ADTEN424	<PIC20_base> + $0D0_H$	16	PBG52#1	—
A/D converter trigger output control register 500	PIC2ADTEN500	<PIC20_base> + 180_H	16, 32	PBG52#1	—
A/D converter trigger output control register 501	PIC2ADTEN501	<PIC20_base> + 184_H	16, 32	PBG52#1	—
A/D converter trigger output control register 502	PIC2ADTEN502	<PIC20_base> + 188_H	16, 32	PBG52#1	—
A/D converter trigger output control register 503	PIC2ADTEN503	<PIC20_base> + $18C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 504	PIC2ADTEN504	<PIC20_base> + 190_H	16, 32	PBG52#1	—
A/D converter trigger output control register 510	PIC2ADTEN510	<PIC20_base> + $1A0_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 511	PIC2ADTEN511	<PIC20_base> + $1A4_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 512	PIC2ADTEN512	<PIC20_base> + $1A8_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 513	PIC2ADTEN513	<PIC20_base> + $1AC_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 514	PIC2ADTEN514	<PIC20_base> + $1B0_H$	16, 32	PBG52#1	—

Table 41.73 List of Registers (2/3)

Register Name	Symbol	Address	Access size	Access Protection	
				PBG	Other
A/D converter trigger output control register 600	PIC2ADTEN600	<PIC20_base> + 200_H	16, 32	PBG52#1	—
A/D converter trigger output control register 601	PIC2ADTEN601	<PIC20_base> + 204_H	16, 32	PBG52#1	—
A/D converter trigger output control register 602	PIC2ADTEN602	<PIC20_base> + 208_H	16, 32	PBG52#1	—
A/D converter trigger output control register 603	PIC2ADTEN603	<PIC20_base> + $20C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 604	PIC2ADTEN604	<PIC20_base> + 210_H	16, 32	PBG52#1	—
A/D converter trigger output control register 610	PIC2ADTEN610	<PIC20_base> + 220_H	16, 32	PBG52#1	—
A/D converter trigger output control register 611	PIC2ADTEN611	<PIC20_base> + 224_H	16, 32	PBG52#1	—
A/D converter trigger output control register 612	PIC2ADTEN612	<PIC20_base> + 228_H	16, 32	PBG52#1	—
A/D converter trigger output control register 613	PIC2ADTEN613	<PIC20_base> + $22C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 614	PIC2ADTEN614	<PIC20_base> + 230_H	16, 32	PBG52#1	—
A/D converter trigger output control register 700	PIC2ADTEN700	<PIC20_base> + 280_H	16, 32	PBG52#1	—
A/D converter trigger output control register 701	PIC2ADTEN701	<PIC20_base> + 284_H	16, 32	PBG52#1	—
A/D converter trigger output control register 702	PIC2ADTEN702	<PIC20_base> + 288_H	16, 32	PBG52#1	—
A/D converter trigger output control register 703	PIC2ADTEN703	<PIC20_base> + $28C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 704	PIC2ADTEN704	<PIC20_base> + 290_H	16, 32	PBG52#1	—
A/D converter trigger output control register 710	PIC2ADTEN710	<PIC20_base> + $2A0_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 711	PIC2ADTEN711	<PIC20_base> + $2A4_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 712	PIC2ADTEN712	<PIC20_base> + $2A8_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 713	PIC2ADTEN713	<PIC20_base> + $2AC_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 714	PIC2ADTEN714	<PIC20_base> + $2B0_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 800	PIC2ADTEN800	<PIC20_base> + 300_H	16, 32	PBG52#1	—
A/D converter trigger output control register 801	PIC2ADTEN801	<PIC20_base> + 304_H	16, 32	PBG52#1	—
A/D converter trigger output control register 802	PIC2ADTEN802	<PIC20_base> + 308_H	16, 32	PBG52#1	—
A/D converter trigger output control register 803	PIC2ADTEN803	<PIC20_base> + $30C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 804	PIC2ADTEN804	<PIC20_base> + 310_H	16, 32	PBG52#1	—
A/D converter trigger output control register 810	PIC2ADTEN810	<PIC20_base> + 320_H	16, 32	PBG52#1	—
A/D converter trigger output control register 811	PIC2ADTEN811	<PIC20_base> + 324_H	16, 32	PBG52#1	—
A/D converter trigger output control register 812	PIC2ADTEN812	<PIC20_base> + 328_H	16, 32	PBG52#1	—
A/D converter trigger output control register 813	PIC2ADTEN813	<PIC20_base> + $32C_H$	16, 32	PBG52#1	—
A/D converter trigger output control register 814	PIC2ADTEN814	<PIC20_base> + 330_H	16, 32	PBG52#1	—
GTM Hi-Z control register					
HI-Z function for GTM output enable control register	PIC2ENHIZDTM	<PIC20_base> + $A00_H$	8	PBG52#1	—
Encoder timer trigger selection control register					
Encoder timer 0 trigger selection control register	PIC2ENCA0TSEL	<PIC21_base> + 000_H	16, 32	PBG52#1	—
Encoder timer 1 trigger selection control register	PIC2ENCA1TSEL	<PIC21_base> + 020_H	16, 32	PBG52#1	—
Encoder timer trigger output configuration register 0	PIC2ENCATCFG0	<PIC22_base> + $0F0_H$	16	PBG52#1	—
Encoder timer trigger output configuration register 1	PIC2ENCATCFG1	<PIC22_base> + $0F4_H$	16	PBG52#1	—
Encoder timer input select register					
Encoder input selection control register	PIC2ENCAISEN	<PIC22_base> + $0F8_H$	8	PBG52#1	—
PSI5S0 sync output control register					
PSI5S0 sync output control register 0	PIC2PSI5S0EN0	<PIC22_base> + 070_H	16, 32	PBG52#1	—
PSI5S0 sync output control register 1	PIC2PSI5S0EN1	<PIC22_base> + 074_H	16, 32	PBG52#1	—

Table 41.73 List of Registers (3/3)

Register Name	Symbol	Address	Access size	Access Protection	
				PBG	Other
PSI5S0 sync output control register 2	PIC2PSI5S0EN2	<PIC22_base> + 078_H	16, 32	PBG52#1	—
PSI5S0 sync output control register 3	PIC2PSI5S0EN3	<PIC22_base> + $07C_H$	16, 32	PBG52#1	—
PSI5S1 sync output control register					
PSI5S1 sync output control register 0	PIC2PSI5S1EN0	<PIC22_base> + $0A0_H$	16, 32	PBG52#1	—
PSI5S1 sync output control register 1	PIC2PSI5S1EN1	<PIC22_base> + $0A4_H$	16, 32	PBG52#1	—
PSI5S1 sync output control register 2	PIC2PSI5S1EN2	<PIC22_base> + $0A8_H$	16, 32	PBG52#1	—
PSI5S1 sync output control register 3	PIC2PSI5S1EN3	<PIC22_base> + $0AC_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register					
GTM timer input module (TIM) source select register 0	PIC2GTMINEN0	<PIC22_base> + $0B0_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 1	PIC2GTMINEN1	<PIC22_base> + $0B4_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 2	PIC2GTMINEN2	<PIC22_base> + $0B8_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 3	PIC2GTMINEN3	<PIC22_base> + $0BC_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 4	PIC2GTMINEN4	<PIC22_base> + $0C0_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 5	PIC2GTMINEN5	<PIC22_base> + $0C4_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 6	PIC2GTMINEN6	<PIC22_base> + $0C8_H$	16, 32	PBG52#1	—
GTM timer input module (TIM) source select register 7	PIC2GTMINEN7	<PIC22_base> + $0CC_H$	16, 32	PBG52#1	—

41.3.2.2 PIC2ADCJnTSELj — A/D Converter n Trigger Select Control Register j (n = 0, 1; j = 0 to 4)

The PIC2ADCJnTSELj register selects triggers for ADCJn scan group j (n = 0, 1; j = 0 to 4).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC20_base> + 000_H (n = 0, j = 0), <PIC20_base> + 004_H (n = 0, j = 1),
<PIC20_base> + 008_H (n = 0, j = 2), <PIC20_base> + $00C_H$ (n = 0, j = 3),
<PIC20_base> + 010_H (n = 0, j = 4),
<PIC20_base> + 020_H (n = 1, j = 0), <PIC20_base> + 024_H (n = 1, j = 1),
<PIC20_base> + 028_H (n = 1, j = 2), <PIC20_base> + $02C_H$ (n = 1, j = 3),
<PIC20_base> + 030_H (n = 1, j = 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	ADCJnTSELj 28	ADCJnTSELj 27	ADCJnTSELj 26	ADCJnTSELj 25	ADCJnTSELj 24	ADCJnTSELj 23	ADCJnTSELj 22	ADCJnTSELj 21	ADCJnTSELj 20	ADCJnTSELj 19	ADCJnTSELj 18	ADCJnTSELj 17	ADCJnTSELj 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADCJnTSELj 15	ADCJnTSELj 14	ADCJnTSELj 13	ADCJnTSELj 12	ADCJnTSELj 11	ADCJnTSELj 10	ADCJnTSELj 09	ADCJnTSELj 08	ADCJnTSELj 07	ADCJnTSELj 06	ADCJnTSELj 05	ADCJnTSELj 04	ADCJnTSELj 03	ADCJnTSELj 02	ADCJnTSELj 01	ADCJnTSELj 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.74 PIC2ADCJnTSELj Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	ADCJn TSELj28	Output value of ENCAn interrupt signal selected by PIC2ADTEN42j. 0: Selection of the above signal as an ADCJn SGj trigger is disabled. 1: Selection of the above signal as an ADCJn SGj trigger is enabled.
27	ADCJn TSELj27	Selects the INTTSG31I12 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I12 is not selected. 1: INTTSG31I12 is selected.
26	ADCJn TSELj26	Selects the INTTSG31I11 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I11 is not selected. 1: INTTSG31I11 is selected.
25	ADCJn TSELj25	Selects the INTTSG31I8 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I8 is not selected. 1: INTTSG31I8 is selected.
24	ADCJn TSELj24	Selects the INTTSG31I7 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I7 is not selected. 1: INTTSG31I7 is selected.
23	ADCJn TSELj23	Selects the INTTSG31I4 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I4 is not selected. 1: INTTSG31I4 is selected.
22	ADCJn TSELj22	Selects the INTTSG31I3 signal of TSG31 as the trigger source for ADCJn SGj. 0: INTTSG31I3 is not selected. 1: INTTSG31I3 is selected.
21	ADCJn TSELj21	Selects the TAPATADOUT1 signal of TAPA0 as the trigger source for ADCJn SGj. 0: TAPA0TADOUT1 is not selected. 1: TAPA0TADOUT1 is selected.

Table 41.74 PIC2ADCJnTSELj Register Contents (2/3)

Bit Position	Bit Name	Function
20	ADCJn TSELj20	Selects the TAPATADOUT0 signal of TAPA0 as the trigger source for ADCJn SGj. 0: TAPA0TADOUT0 is not selected. 1: TAPA0TADOUT0 is selected.
19	ADCJn TSELj19	Selects the INTTSG30I12 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I8 is not selected. 1: INTTSG30I8 is selected.
18	ADCJn TSELj18	Selects the INTTSG30I11 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I11 is not selected. 1: INTTSG30I11 is selected.
17	ADCJn TSELj17	Selects the INTTSG30I8 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I8 is not selected. 1: INTTSG30I8 is selected.
16	ADCJn TSELj16	Selects the INTTSG30I7 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I7 is not selected. 1: INTTSG30I7 is selected.
15	ADCJn TSELj15	Selects the INTTSG30I4 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I4 is not selected. 1: INTTSG30I4 is selected.
14	ADCJn TSELj14	Selects the INTTSG30I3 signal of TSG30 as the trigger source for ADCJn SGj. 0: INTTSG30I3 is not selected. 1: INTTSG30I3 is selected.
13	ADCJn TSELj13	Selects the ADCJnTRG4 signal of ADCJn as the trigger source for ADCJn SGj. 0: ADCJnTRG4 is not selected. 1: ADCJnTRG4 is selected.
12	ADCJn TSELj12	Selects the ADCJnTRG3 signal of ADCJn as the trigger source for ADCJn SGj. 0: ADCJnTRG3 is not selected. 1: ADCJnTRG3 is selected.
11	ADCJn TSELj11	Selects the ADCJnTRG2 signal of ADCJn as the trigger source for ADCJn SGj. 0: ADCJnTRG2 is not selected. 1: ADCJnTRG2 is selected.
10	ADCJn TSELj10	Selects the ADCJnTRG1 signal of ADCJn as the trigger source for ADCJn SGj. 0: ADCJnTRG1 is not selected. 1: ADCJnTRG1 is selected.
9	ADCJn TSELj09	Selects the ADCJnTRG0 signal of ADCJn as the trigger source for ADCJn SGj. 0: ADCJnTRG0 is not selected. 1: ADCJnTRG0 is selected.
8	ADCJn TSELj08	Selects the TSG31ADTRG1 signal of TSG31 as the trigger source for ADCJn SGj. 0: TSG31ADTRG1 is not selected. 1: TSG31ADTRG1 is selected.
7	ADCJn TSELj07	Selects the TSG31ADTRG0 signal of TSG31 as the trigger source for ADCJn SGj. 0: TSG31ADTRG0 is not selected. 1: TSG31ADTRG0 is selected.
6	ADCJn TSELj06	Selects the TSG30ADTRG1 signal of TSG30 as the trigger source for ADCJn SGj. 0: TSG30ADTRG1 is not selected. 1: TSG30ADTRG1 is selected.
5	ADCJn TSELj05	Selects the TSG30ADTRG0 signal of TSG30 as the trigger source for ADCJn SGj. 0: TSG30ADTRG0 is not selected. 1: TSG30ADTRG0 is selected.
4	ADCJn TSELj04	Selects the INTTAUJ1I3 signal of TAUJ1 as the trigger source for ADCJn SGj. 0: INTTAUJ1I3 is not selected. 1: INTTAUJ1I3 is selected.

Table 41.74 PIC2ADCJnTSELj Register Contents (3/3)

Bit Position	Bit Name	Function
3	ADCJn TSELj03	Selects the INTTAUJ0I3 signal of TAUJ0 as the trigger source for ADCJn SGj. 0: INTTAUJ0I3 is not selected. 1: INTTAUJ0I3 is selected.
2	ADCJn TSELj02	Output value of GTM timer output selected by PIC2ADTEN50j, PIC2ADTEN60j, PIC2ADTEN70j, PIC2ADTEN80j. 0: Selection of the above signal as an ADCJn SGj trigger is disabled. 1: Selection of the above signal as an ADCJn SGj trigger is enabled.
1	ADCJn TSELj01	Selects the TAUD1 interrupt signal selected by the PIC2ADTEN41j register as the trigger source for ADCJn SGj. 0: TAUD1 interrupt is not selected. 1: TAUD1 interrupt is selected.
0	ADCJn TSELj00	Selects the TAUD0 interrupt signal selected by the PIC2ADTEN40j register as the trigger source for ADCJn SGj. 0: TAUD0 interrupt is not selected. 1: TAUD0 interrupt is selected.

41.3.2.3 PIC2ADCJnEDGSEL — A/D Converter Trigger Edge Control Register (n = 0, 1)

The PIC2ADCJnEDGSEL register selects an effective edge for the one-shot pulse generation circuit which generates an ADCJn trigger.

Access: This register can be read or written in 16-bit units.

Address: <PIC20_base> + $01C_H$ (n = 0), <PIC20_base> + $03C_H$ (n = 1)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIC2ADCJnEDGSEL[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.75 PIC2ADCJnEDGSEL Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	PIC2ADCJnEDGSEL[9:8]	Selects an effective edge of ADCJn scan group 4. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.)
7, 6	PIC2ADCJnEDGSEL[7:6]	Selects an effective edge of ADCJn scan group 3. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.)
5, 4	PIC2ADCJnEDGSEL[5:4]	Selects an effective edge of ADCJn scan group 2. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.)
3, 2	PIC2ADCJnEDGSEL[3:2]	Selects an effective edge of ADCJn scan group 1. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.)
1, 0	PIC2ADCJnEDGSEL[1:0]	Selects an effective edge of ADCJn scan group 0. 00: Rising edge is selected. 01: Falling edge is selected. 10: Both edges are selected. 11: — (setting prohibited.)

NOTE

The selection of “falling edge” and “both edges” is not available for ADC trigger signals from GTM.

For procedure to use a falling edge or both edges of a GTM output signal, refer to **Section 41.3.3.1, ADCJ Trigger Select Function** for more details.

41.3.2.4 PIC2ADTEN4kj — A/D Converter Trigger Output Select Control Register (k = 0, 1; j = 0 to 4)

The PIC2ADTEN4kj register enables selecting a trigger source from TAUDn channel m (n = 0, 1; m = 0 to 15) as the ADCJn trigger (n = 0, 1; j = 0 to 4). This register is common to ADCJ0 and ADCJ1.

Access: This register can be read or written in 16-bit units.

Address: <PIC20_base> + 080_H (k = 0, j = 0), <PIC20_base> + 084_H (k = 0, j = 1),
<PIC20_base> + 088_H (k = 0, j = 2), <PIC20_base> + $08C_H$ (k = 0, j = 3),
<PIC20_base> + 090_H (k = 0, j = 4),
<PIC20_base> + $0A0_H$ (k = 1, j = 0), <PIC20_base> + $0A4_H$ (k = 1, j = 1),
<PIC20_base> + $0A8_H$ (k = 1, j = 2), <PIC20_base> + $0AC_H$ (k = 1, j = 3),
<PIC20_base> + $0B0_H$ (k = 1, j = 4)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN4kj[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.76 PIC2ADTEN4kj Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC2ADTEN4kjm (m = 0 to 15)	Set a trigger source from TAUDn channel m. 0: Trigger source of TAUDn channel m cannot be selected as the ADCJ trigger. 1: Trigger source of TAUDn channel m can be selected as the ADCJ trigger.

41.3.2.5 PIC2ADTEN4kj — A/D Converter Trigger Output Select Control Register (k = 2; j = 0 to 4)

The PIC2ADTEN4kj register enables selecting a trigger source from ENCAn interrupt as the ADCJ trigger.

This register is common to ADCJ0 and ADCJ1.

Access: This register can be read or written in 16-bit units.

Address: <PIC20_base> + $0C0_H$ (k = 2, j = 0), <PIC20_base> + $0C4_H$ (k = 2, j = 1), <PIC20_base> + $0C8_H$ (k = 2, j = 2), <PIC20_base> + $0CC_H$ (k = 2, j = 3), <PIC20_base> + $0D0_H$ (k = 2, j = 4)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PIC2ADTEN4kj[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.77 PIC2ADTEN4kj Register Contents

Bit Position	Bit Name	Function
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	PIC2ADTEN4kj5	ENCA1 Clear interrupt signal by encoder input (phase Z) 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
4	PIC2ADTEN4kj4	ENCA1 Compare 1 match or Capture 1 interrupt signal 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
3	PIC2ADTEN4kj3	ENCA1 Compare 0 match or Capture 0 interrupt signal 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
2	PIC2ADTEN4kj2	ENCA0 Clear interrupt signal by encoder input (phase Z) 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
1	PIC2ADTEN4kj1	ENCA0 Compare 1 match or Capture 1 interrupt signal 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
0	PIC2ADTEN4kj0	ENCA0 Compare 0 match or Capture 0 interrupt signal 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

41.3.2.6 PIC2ADTEN5nj — A/D Converter Trigger Output Select Control Register 5nj (n = 0, 1; j = 0 to 4)

The PIC2ADTEN5nj register enables a trigger source from GTM TIM shared interrupt to be selected as an ADCJn trigger (n = 0, 1; j = 0 to 4).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC20_base> + 180_H (n = 0, j = 0), <PIC20_base> + 184_H (n = 0, j = 1),
<PIC20_base> + 188_H (n = 0, j = 2), <PIC20_base> + $18C_H$ (n = 0, j = 3),
<PIC20_base> + 190_H (n = 0, j = 4),
<PIC20_base> + $1A0_H$ (n = 1, j = 0), <PIC20_base> + $1A4_H$ (n = 1, j = 1),
<PIC20_base> + $1A8_H$ (n = 1, j = 2), <PIC20_base> + $1AC_H$ (n =1, j = 3),
<PIC20_base> + $1B0_H$ (n = 1, j = 4)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN5nj31	PIC2ADTEN5nj30	PIC2ADTEN5nj29	PIC2ADTEN5nj28	PIC2ADTEN5nj27	PIC2ADTEN5nj26	PIC2ADTEN5nj25	PIC2ADTEN5nj24	PIC2ADTEN5nj23	PIC2ADTEN5nj22	PIC2ADTEN5nj21	PIC2ADTEN5nj20	PIC2ADTEN5nj19	PIC2ADTEN5nj18	PIC2ADTEN5nj17	PIC2ADTEN5nj16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN5nj15	PIC2ADTEN5nj14	PIC2ADTEN5nj13	PIC2ADTEN5nj12	PIC2ADTEN5nj11	PIC2ADTEN5nj10	PIC2ADTEN5nj09	PIC2ADTEN5nj08	PIC2ADTEN5nj07	PIC2ADTEN5nj06	PIC2ADTEN5nj05	PIC2ADTEN5nj04	PIC2ADTEN5nj03	PIC2ADTEN5nj02	PIC2ADTEN5nj01	PIC2ADTEN5nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.78 PIC2ADTEN5nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN5nj31	TIM3 shared interrupts for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
30	PIC2ADTEN5nj30	TIM3 shared interrupts for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
29	PIC2ADTEN5nj29	TIM3 Shared interrupts for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
28	PIC2ADTEN5nj28	TIM3 shared interrupts for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
27	PIC2ADTEN5nj27	TIM3 shared interrupts for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
26	PIC2ADTEN5nj26	TIM3 shared interrupts for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
25	PIC2ADTEN5nj25	TIM3 shared interrupts for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
24	PIC2ADTEN5nj24	TIM3 shared interrupts for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
23	PIC2ADTEN5nj23	TIM2 shared interrupts for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.78 PIC2ADTEN5nj Register Contents (2/3)

Bit Position	Bit Name	Function
22	PIC2ADTEN5nj 22	TIM2 shared interrupts for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
21	PIC2ADTEN5nj 21	TIM2 shared interrupts for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
20	PIC2ADTEN5nj 20	TIM2 shared interrupts for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
19	PIC2ADTEN5nj 19	TIM2 shared interrupts for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
18	PIC2ADTEN5nj 18	TIM2 shared interrupts for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
17	PIC2ADTEN5nj 17	TIM2 shared interrupts for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
16	PIC2ADTEN5nj 16	TIM2 shared interrupts for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
15	PIC2ADTEN5nj 15	TIM1 shared interrupts for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
14	PIC2ADTEN5nj 14	TIM1 shared interrupts for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
13	PIC2ADTEN5nj 13	TIM1 shared interrupts for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
12	PIC2ADTEN5nj 12	TIM1 shared interrupts for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
11	PIC2ADTEN5nj 11	TIM1 shared interrupts for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
10	PIC2ADTEN5nj 10	TIM1 shared interrupts for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
9	PIC2ADTEN5nj 09	TIM1 shared interrupts for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
8	PIC2ADTEN5nj 08	TIM1 shared interrupts for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
7	PIC2ADTEN5nj 07	TIM0 shared interrupts for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
6	PIC2ADTEN5nj 06	TIM0 shared interrupts for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
5	PIC2ADTEN5nj 05	TIM0 shared interrupts for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
4	PIC2ADTEN5nj 04	TIM0 shared interrupts for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.78 PIC2ADTEN5nj Register Contents (3/3)

Bit Position	Bit Name	Function
3	PIC2ADTEN5nj03	TIM0 shared interrupts for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
2	PIC2ADTEN5nj02	TIM0 shared interrupts for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
1	PIC2ADTEN5nj01	TIM0 shared interrupts for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
0	PIC2ADTEN5nj00	TIM0 shared interrupts for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

41.3.2.7 PIC2ADTEN6nj — A/D Converter Trigger Output Select Control Register 6nj (n = 0, 1; j = 0 to 4)

The PIC2ADTEN6nj register enables a trigger source from GTM timer output to be selected as an ADCJn trigger (n = 0, 1; j = 0 to 4).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC20_base> + 200_H (n = 0, j = 0), <PIC20_base> + 204_H (n = 0, j = 1),
<PIC20_base> + 208_H (n = 0, j = 2), <PIC20_base> + $20C_H$ (n = 0, j = 3),
<PIC20_base> + 210_H (n = 0, j = 4),
<PIC20_base> + 220_H (n = 1, j = 0), <PIC20_base> + 224_H (n = 1, j = 1),
<PIC20_base> + 228_H (n = 1, j = 2), <PIC20_base> + $22C_H$ (n =1, j = 3),
<PIC20_base> + 230_H (n = 1, j = 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN6nj31	PIC2ADTEN6nj30	PIC2ADTEN6nj29	PIC2ADTEN6nj28	PIC2ADTEN6nj27	PIC2ADTEN6nj26	PIC2ADTEN6nj25	PIC2ADTEN6nj24	PIC2ADTEN6nj23	PIC2ADTEN6nj22	PIC2ADTEN6nj21	PIC2ADTEN6nj20	PIC2ADTEN6nj19	PIC2ADTEN6nj18	PIC2ADTEN6nj17	PIC2ADTEN6nj16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN6nj15	PIC2ADTEN6nj14	PIC2ADTEN6nj13	PIC2ADTEN6nj12	PIC2ADTEN6nj11	PIC2ADTEN6nj10	PIC2ADTEN6nj09	PIC2ADTEN6nj08	PIC2ADTEN6nj07	PIC2ADTEN6nj06	PIC2ADTEN6nj05	PIC2ADTEN6nj04	PIC2ADTEN6nj03	PIC2ADTEN6nj02	PIC2ADTEN6nj01	PIC2ADTEN6nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.79 PIC2ADTEN6nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN6nj31	Inverted timer output signal for ATOM1 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
30	PIC2ADTEN6nj30	Timer output signals for ATOM1 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
29	PIC2ADTEN6nj29	Inverted timer output signal for ATOM1 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
28	PIC2ADTEN6nj28	Timer output signals for ATOM1 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
27	PIC2ADTEN6nj27	Inverted timer output signal for ATOM1 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
26	PIC2ADTEN6nj26	Timer output signals for ATOM1 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
25	PIC2ADTEN6nj25	Inverted timer output signal for ATOM1 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
24	PIC2ADTEN6nj24	Timer output signals for ATOM1 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
23	PIC2ADTEN6nj23	Inverted timer output signal for ATOM1 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.79 PIC2ADTEN6nj Register Contents (2/3)

Bit Position	Bit Name	Function
22	PIC2ADTEN6nj 22	Timer output signals for ATOM1 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
21	PIC2ADTEN6nj 21	Inverted timer output signal for ATOM1 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
20	PIC2ADTEN6nj 20	Timer output signals for ATOM1 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
19	PIC2ADTEN6nj 19	Inverted timer output signal for ATOM1 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
18	PIC2ADTEN6nj 18	Timer output signals for ATOM1 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
17	PIC2ADTEN6nj 17	Inverted timer output signal for ATOM1 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
16	PIC2ADTEN6nj 16	Timer output signals for ATOM1 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
15	PIC2ADTEN6nj 15	Inverted timer output signal for ATOM0 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
14	PIC2ADTEN6nj 14	Timer output signals for ATOM0 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
13	PIC2ADTEN6nj 13	Inverted timer output signal for ATOM0 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
12	PIC2ADTEN6nj 12	Timer output signals for ATOM0 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
11	PIC2ADTEN6nj 11	Inverted timer output signal for ATOM0 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
10	PIC2ADTEN6nj 10	Timer output signals for ATOM0 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
9	PIC2ADTEN6nj 09	Inverted timer output signal for ATOM0 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
8	PIC2ADTEN6nj 08	Timer output signals for ATOM0 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
7	PIC2ADTEN6nj 07	Inverted timer output signal for ATOM0 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
6	PIC2ADTEN6nj 06	Timer output signals for ATOM0 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
5	PIC2ADTEN6nj 05	Inverted timer output signal for ATOM0 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
4	PIC2ADTEN6nj 04	Timer output signals for ATOM0 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.79 PIC2ADTEN6nj Register Contents (3/3)

Bit Position	Bit Name	Function
3	PIC2ADTEN6nj03	Inverted timer output signal for ATOM0 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
2	PIC2ADTEN6nj02	Timer output signals for ATOM0 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
1	PIC2ADTEN6nj01	Inverted timer output signal for ATOM0 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
0	PIC2ADTEN6nj00	Timer output signals for ATOM0 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

41.3.2.8 PIC2ADTEN7nj — A/D Converter Trigger Output Select Control Register 7nj (n = 0, 1; j = 0 to 4)

The PIC2ADTEN7nj register enables a trigger source from GTM timer output to be selected as an ADCJn trigger (n = 0, 1; j = 0 to 4).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC20_base> + 280_H (n = 0, j = 0), <PIC20_base> + 284_H (n = 0, j = 1),
<PIC20_base> + 288_H (n = 0, j = 2), <PIC20_base> + $28C_H$ (n = 0, j = 3),
<PIC20_base> + 290_H (n = 0, j = 4),
<PIC20_base> + $2A0_H$ (n = 1, j = 0), <PIC20_base> + $2A4_H$ (n = 1, j = 1),
<PIC20_base> + $2A8_H$ (n = 1, j = 2), <PIC20_base> + $2AC_H$ (n =1, j = 3),
<PIC20_base> + $2B0_H$ (n = 1, j = 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN7nj 31	PIC2ADTEN7nj 30	PIC2ADTEN7nj 29	PIC2ADTEN7nj 28	PIC2ADTEN7nj 27	PIC2ADTEN7nj 26	PIC2ADTEN7nj 25	PIC2ADTEN7nj 24	PIC2ADTEN7nj 23	PIC2ADTEN7nj 22	PIC2ADTEN7nj 21	PIC2ADTEN7nj 20	PIC2ADTEN7nj 19	PIC2ADTEN7nj 18	PIC2ADTEN7nj 17	PIC2ADTEN7nj 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN7nj 15	PIC2ADTEN7nj 14	PIC2ADTEN7nj 13	PIC2ADTEN7nj 12	PIC2ADTEN7nj 11	PIC2ADTEN7nj 10	PIC2ADTEN7nj 09	PIC2ADTEN7nj 08	PIC2ADTEN7nj 07	PIC2ADTEN7nj 06	PIC2ADTEN7nj 05	PIC2ADTEN7nj 04	PIC2ADTEN7nj 03	PIC2ADTEN7nj 02	PIC2ADTEN7nj 01	PIC2ADTEN7nj 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.80 PIC2ADTEN7nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN7nj 31	Inverted timer output signal for ATOM3 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
30	PIC2ADTEN7nj 30	Timer output signals for ATOM3 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
29	PIC2ADTEN7nj 29	Inverted timer output signal for ATOM3 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
28	PIC2ADTEN7nj 28	Timer output signals for ATOM3 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
27	PIC2ADTEN7nj 27	Inverted timer output signal for ATOM3 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
26	PIC2ADTEN7nj 26	Timer output signals for ATOM3 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
25	PIC2ADTEN7nj 25	Inverted timer output signal for ATOM3 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
24	PIC2ADTEN7nj 24	Timer output signals for ATOM3 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
23	PIC2ADTEN7nj 23	Inverted timer output signal for ATOM3 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.80 PIC2ADTEN7nj Register Contents (2/3)

Bit Position	Bit Name	Function
22	PIC2ADTEN7nj 22	Timer output signals for ATOM3 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
21	PIC2ADTEN7nj 21	Inverted timer output signal for ATOM3 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
20	PIC2ADTEN7nj 20	Timer output signals for ATOM3 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
19	PIC2ADTEN7nj 19	Inverted timer output signal for ATOM3 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
18	PIC2ADTEN7nj 18	Timer output signals for ATOM3 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
17	PIC2ADTEN7nj 17	Inverted timer output signal for ATOM3 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
16	PIC2ADTEN7nj 16	Timer output signals for ATOM3 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
15	PIC2ADTEN7nj 15	Inverted timer output signal for ATOM2 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
14	PIC2ADTEN7nj 14	Timer output signals for ATOM2 for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
13	PIC2ADTEN7nj 13	Inverted timer output signal for ATOM2 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
12	PIC2ADTEN7nj 12	Timer output signals for ATOM2 for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
11	PIC2ADTEN7nj 11	Inverted timer output signal for ATOM2 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
10	PIC2ADTEN7nj 10	Timer output signals for ATOM2 for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
9	PIC2ADTEN7nj 09	Inverted timer output signal for ATOM2 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
8	PIC2ADTEN7nj 08	Timer output signals for ATOM2 for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
7	PIC2ADTEN7nj 07	Inverted timer output signal for ATOM2 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
6	PIC2ADTEN7nj 06	Timer output signals for ATOM2 for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
5	PIC2ADTEN7nj 05	Inverted timer output signal for ATOM2 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
4	PIC2ADTEN7nj 04	Timer output signals for ATOM2 for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.80 PIC2ADTEN7nj Register Contents (3/3)

Bit Position	Bit Name	Function
3	PIC2ADTEN7nj03	Inverted timer output signal for ATOM2 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
2	PIC2ADTEN7nj02	Timer output signals for ATOM2 for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
1	PIC2ADTEN7nj01	Inverted timer output signal for ATOM2 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
0	PIC2ADTEN7nj00	Timer output signals for ATOM2 for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

41.3.2.9 PIC2ADTEN8nj — A/D Converter Trigger Output Select Control Register 8nj (n = 0, 1; j = 0 to 4)

The PIC2ADTEN8nj register enables a trigger source from GTM timer output to be selected as an ADCJn trigger (n = 0, 1; j = 0 to 4).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC20_base> + 300_H (n = 0, j = 0), <PIC20_base> + 304_H (n = 0, j = 1),
<PIC20_base> + 308_H (n = 0, j = 2), <PIC20_base> + $30C_H$ (n = 0, j = 3),
<PIC20_base> + 310_H (n = 0, j = 4),
<PIC20_base> + 320_H (n = 1, j = 0), <PIC20_base> + 324_H (n = 1, j = 1),
<PIC20_base> + 328_H (n = 1, j = 2), <PIC20_base> + $32C_H$ (n =1, j = 3),
<PIC20_base> + 330_H (n = 1, j = 4)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ADTEN8nj31	PIC2ADTEN8nj30	PIC2ADTEN8nj29	PIC2ADTEN8nj28	PIC2ADTEN8nj27	PIC2ADTEN8nj26	PIC2ADTEN8nj25	PIC2ADTEN8nj24	PIC2ADTEN8nj23	PIC2ADTEN8nj22	PIC2ADTEN8nj21	PIC2ADTEN8nj20	PIC2ADTEN8nj19	PIC2ADTEN8nj18	PIC2ADTEN8nj17	PIC2ADTEN8nj16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ADTEN8nj15	PIC2ADTEN8nj14	PIC2ADTEN8nj13	PIC2ADTEN8nj12	PIC2ADTEN8nj11	PIC2ADTEN8nj10	PIC2ADTEN8nj09	PIC2ADTEN8nj08	PIC2ADTEN8nj07	PIC2ADTEN8nj06	PIC2ADTEN8nj05	PIC2ADTEN8nj04	PIC2ADTEN8nj03	PIC2ADTEN8nj02	PIC2ADTEN8nj01	PIC2ADTEN8nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.81 PIC2ADTEN8nj Register Contents (1/3)

Bit Position	Bit Name	Function
31	PIC2ADTEN8nj31	MCS3 interrupt for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
30	PIC2ADTEN8nj30	MCS3 interrupt for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
29	PIC2ADTEN8nj29	MCS3 interrupt for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
28	PIC2ADTEN8nj28	MCS3 interrupt for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
27	PIC2ADTEN8nj27	MCS3 interrupt for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
26	PIC2ADTEN8nj26	MCS3 interrupt for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
25	PIC2ADTEN8nj25	MCS3 interrupt for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
24	PIC2ADTEN8nj24	MCS3 interrupt for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
23	PIC2ADTEN8nj23	MCS2 interrupt for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.81 PIC2ADTEN8nj Register Contents (2/3)

Bit Position	Bit Name	Function
22	PIC2ADTEN8nj 22	MCS2 interrupt for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
21	PIC2ADTEN8nj 21	MCS2 interrupt for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
20	PIC2ADTEN8nj 20	MCS2 interrupt for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
19	PIC2ADTEN8nj 19	MCS2 interrupt for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
18	PIC2ADTEN8nj 18	MCS2 interrupt for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
17	PIC2ADTEN8nj 17	MCS2 interrupt for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
16	PIC2ADTEN8nj 16	MCS2 interrupt for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
15	PIC2ADTEN8nj 15	MCS1 interrupt for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
14	PIC2ADTEN8nj 14	MCS1 interrupt for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
13	PIC2ADTEN8nj 13	MCS1 interrupt for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
12	PIC2ADTEN8nj 12	MCS1 interrupt for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
11	PIC2ADTEN8nj 11	MCS1 interrupt for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
10	PIC2ADTEN8nj 10	MCS1 interrupt for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
9	PIC2ADTEN8nj 09	MCS1 interrupt for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
8	PIC2ADTEN8nj 08	MCS1 interrupt for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
7	PIC2ADTEN8nj 07	MCS0 interrupt for channel 7 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
6	PIC2ADTEN8nj 06	MCS0 interrupt for channel 6 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
5	PIC2ADTEN8nj 05	MCS0 interrupt for channel 5 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
4	PIC2ADTEN8nj 04	MCS0 interrupt for channel 4 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

Table 41.81 PIC2ADTEN8nj Register Contents (3/3)

Bit Position	Bit Name	Function
3	PIC2ADTEN8nj03	MCS0 interrupt for channel 3 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
2	PIC2ADTEN8nj02	MCS0 interrupt for channel 2 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
1	PIC2ADTEN8nj01	MCS0 interrupt for channel 1 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.
0	PIC2ADTEN8nj00	MCS0 interrupt for channel 0 0: Selection of the above signal as an ADCJ trigger is disabled. 1: Selection of the above signal as an ADCJ trigger is enabled.

41.3.2.10 PIC2ENHIZDTM — Hi-Z Function for GTM Output Enable Control Register

The PIC2ENHIZDTM register enables Hi-Z function for GTM outputs.

Access: This register can be read or written in 8-bit units.

Address: <PIC20_base> + $A00_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HIZ3ECM	HIZ2ECM	HIZ1ECM	HIZ0ECM	HIZ3ESO	HIZ2ESO	HIZ1ESO	HIZ0ESO
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.82 PIC2ENHIZDTM Register Contents

Bit Position	Bit Name	Function
7	HIZ3ECM	Enables/disables Hi-Z control by ECM for GTM ATOM3 outputs 0: Disabled 1: Enabled
6	HIZ2ECM	Enables/disables Hi-Z control by ECM for GTM ATOM2 outputs 0: Disabled 1: Enabled
5	HIZ1ECM	Enables/disables Hi-Z control by ECM for GTM ATOM1 outputs 0: Disabled 1: Enabled
4	HIZ0ECM	Enables/disables Hi-Z control by ECM for GTM ATOM0 outputs 0: Disabled 1: Enabled
3	HIZ3ESO	Enables/disables Hi-Z control by $\overline{ESO3}$ for GTM ATOM3 outputs 0: Disabled 1: Enabled
2	HIZ2ESO	Enables/disables Hi-Z control by $\overline{ESO2}$ for GTM ATOM2 outputs 0: Disabled 1: Enabled
1	HIZ1ESO	Enables/disables Hi-Z control by $\overline{ESO1}$ for GTM ATOM1 outputs 0: Disabled 1: Enabled
0	HIZ0ESO	Enables/disables Hi-Z control by $\overline{ESO0}$ for GTM ATOM0 outputs 0: Disabled 1: Enabled

41.3.2.11 PIC2ENCAnTSEL — Encoder Timer n Trigger Selection Control Register (n = 0, 1)

The PIC2ENCAnTSEL register selects a trigger for ENCAnTTIN0 (n = 0, 1).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC21_base> + 000_H (n = 0),
<PIC21_base> + 020_H (n = 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2ENCAnTSEL31	PIC2ENCAnTSEL30	PIC2ENCAnTSEL29	PIC2ENCAnTSEL28	PIC2ENCAnTSEL27	PIC2ENCAnTSEL26	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PIC2ENCAnTSEL05	PIC2ENCAnTSEL04	PIC2ENCAnTSEL03	PIC2ENCAnTSEL02	PIC2ENCAnTSEL01	PIC2ENCAnTSEL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.83 PIC2ENCAnTSEL Register Contents (1/2)

Bit Position	Bit Name	Function
31	PIC2ENCAnTSEL31	Selects the INTENCA1IEC as a trigger source for ENCAnTTIN0. (INTENCA1IEC clear interrupt signal by encoder) 0: INTENCA1IEC is not selected. 1: INTENCA1IEC is selected.
30	PIC2ENCAnTSEL30	Selects the INTENCA1I1 as a trigger source for ENCAnTTIN0. (INTENCA1I1 compare 1 match or capture 1 interrupt) 0: INTENCA1I1 is not selected. 1: INTENCA1I1 is selected.
29	PIC2ENCAnTSEL29	Selects the INTENCA1I0 as a trigger source for ENCAnTTIN0. (INTENCA1I0 compare 0 match or capture 0 interrupt) 0: INTENCA1I0 is not selected. 1: INTENCA1I0 is selected.
28	PIC2ENCAnTSEL28	Selects the INTENCA0IEC as a trigger source for ENCAnTTIN0. (INTENCA0IEC clear interrupt signal by encoder) 0: INTENCA0IEC is not selected. 1: INTENCA0IEC is selected.
27	PIC2ENCAnTSEL27	Selects the INTENCA0I1 as a trigger source for ENCAnTTIN0. (INTENCA0I1 compare 1 match or capture 1 interrupt) 0: INTENCA0I1 is not selected. 1: INTENCA0I1 is selected.
26	PIC2ENCAnTSEL26	Selects the INTENCA0I0 as a trigger source for ENCAnTTIN0. (INTENCA0I0 compare 0 match or capture 0 interrupt) 0: INTENCA0I0 is not selected. 1: INTENCA0I0 is selected.
25 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	PIC2ENCAnTSEL05	Selects the ENCA1I0 pin as a trigger source for ENCAnTTIN0. 0: ENCA1I0 is not selected. 1: ENCA1I0 is selected.
4	PIC2ENCAnTSEL04	Selects the ENCA0I0 pin as a trigger source for ENCAnTTIN0. 0: ENCA0I0 is not selected. 1: ENCA0I0 is selected.

Table 41.83 PIC2ENCAnTSEL Register Contents (2/2)

Bit Position	Bit Name	Function
3	PIC2ENCAn TSEL03	Output value of GTM timer output and TIM interrupt selected by PIC2ENCATCFG1 bit 15 to 8. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.
2	PIC2ENCAn TSEL02	Output value of GTM timer output and TIM interrupt selected by PIC2ENCATCFG1 bit 7 to 0. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.
1	PIC2ENCAn TSEL01	Output value of GTM timer output and TIM interrupt selected by PIC2ENCATCFG0 bit 15 to 8. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.
0	PIC2ENCAn TSEL00	Output value of GTM timer output and TIM interrupt selected by PIC2ENCATCFG0 bit 7 to 0. 0: Selection of the above signal as an ENCA trigger is disabled. 1: Selection of the above signal as an ENCA trigger is enabled.

41.3.2.12 PIC2PSI5SnENk — PSI5-Sn Sync Output Control Register k (n = 0, 1; k = 0 to 3)

The PIC2PSI5SnENk register selects a source of a sync trigger of PSI5-Sn (n = 0, 1; k = 0 to 3).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC22_base> + 070_H (n = 0, k = 0), <PIC22_base> + 074_H (n = 0, k = 1),
<PIC22_base> + 078_H (n = 0, k = 2), <PIC22_base> + $07C_H$ (n = 0, k = 3),
<PIC22_base> + $0A0_H$ (n = 1, k = 0), <PIC22_base> + $0A4_H$ (n = 1, k = 1),
<PIC22_base> + $0A8_H$ (n = 1, k = 2), <PIC22_base> + $0AC_H$ (n = 1, k = 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2PSI5SnENk[31:24]								PIC2PSI5SnENk[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2PSI5SnENk[15:8]								PIC2PSI5SnENk[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.84 PIC2PSI5SnENk Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2PSI5SnENk [31:24]	Selects the GTM ATOM output as a signal source for PSI5-Sn. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected.
23 to 16	PIC2PSI5SnENk [23:16]	Selects the GTM ATOM output as a signal source for PSI5-Sn. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected.
15 to 8	PIC2PSI5SnENk [15:8]	Selects the GTM ATOM output as a signal source for PSI5-Sn. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected.
7 to 0	PIC2PSI5SnENk [7:0]	Selects the GTM ATOM output as a signal source for PSI5-Sn. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected.

Note: PSI5-S destination is determined by k and listed in **Table 41.85**.

Table 41.85 List of Destination (1/2)

PIC2PSI5SnENk	Bit Position	Destination
PIC2PSI5SnEN0	31 to 24	PSI5-Sn Synchronous trigger Ch3
PIC2PSI5SnEN0	23 to 16	PSI5-Sn Synchronous trigger Ch2
PIC2PSI5SnEN0	15 to 8	PSI5-Sn Synchronous trigger Ch1
PIC2PSI5SnEN0	7 to 0	Reserved
PIC2PSI5SnEN1	31 to 24	PSI5-Sn Synchronous trigger Ch7
PIC2PSI5SnEN1	23 to 16	PSI5-Sn Synchronous trigger Ch6
PIC2PSI5SnEN1	15 to 8	PSI5-Sn Synchronous trigger Ch5
PIC2PSI5SnEN1	7 to 0	PSI5-Sn Synchronous trigger Ch4
PIC2PSI5SnEN2	31 to 24	PSI5-Sn Timestamp counter B clear
PIC2PSI5SnEN2	23 to 16	PSI5-Sn Timestamp counter A clear
PIC2PSI5SnEN2	15 to 8	PSI5-Sn Timestamp counter B clock
PIC2PSI5SnEN2	7 to 0	PSI5-Sn Timestamp counter A clock

Table 41.85 List of Destination (2/2)

PIC2PSI5SnENk	Bit Position	Destination
PIC2PSI5SnEN3	31 to 24	Reserved
PIC2PSI5SnEN3	23 to 16	Reserved
PIC2PSI5SnEN3	15 to 8	PSI5-Sn Timestamp counter B enable
PIC2PSI5SnEN3	7 to 0	PSI5-Sn Timestamp counter A enable

41.3.2.13 PIC2GTMINENk — GTM Timer Input Module (TIM) Source Select Register k (k = 0, 2, 4, 6)

The PIC2GTMINENk register selects a source of timer input of GTM (TIM) (k = 0, 2, 4, 6).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC22_base> + $0B0_H$ (k = 0), <PIC22_base> + $0B8_H$ (k = 2),
<PIC22_base> + $0C0_H$ (k = 4), <PIC22_base> + $0C8_H$ (k = 6)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2GTMINENk[31:24]								PIC2GTMINENk[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2GTMINENk[15:8]								PIC2GTMINENk[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.86 PIC2GTMINENk Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2GTMINENk [31:24]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(k / 2)]_IN3) 0 to 127: Refer to **Table 41.87 PIC2GTMINENk List of Function**
23 to 16	PIC2GTMINENk [23:16]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(k / 2)]_IN2) 0 to 127: Refer to **Table 41.87 PIC2GTMINENk List of Function**
15 to 8	PIC2GTMINENk [15:8]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(k/ 2)]_IN1) 0 to 127: Refer to **Table 41.87 PIC2GTMINENk List of Function**
7 to 0	PIC2GTMINENk [7:0]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[(k / 2)]_IN0) 0 to 127: Refer to **Table 41.87 PIC2GTMINENk List of Function**

Table 41.87 PIC2GTMINENk List of Function (1/2)

	Function
0	GPIO selected. (via port functions)
1	ADCJ0 scan group 0 (SG0) end interrupt
2	ADCJ0 scan group 1 (SG1) end interrupt
3	ADCJ0 scan group 2 (SG2) end interrupt
4	ADCJ0 scan group 3 (SG3) end interrupt
5	ADCJ0 scan group 4 (SG4) end interrupt
6	ADCJ1 scan group 0 (SG0) end interrupt
7	ADCJ1 scan group 1 (SG1) end interrupt
8	ADCJ1 scan group 2 (SG2) end interrupt
9	ADCJ1 scan group 3 (SG3) end interrupt
10	ADCJ1 scan group 4 (SG4) end interrupt
11 to 32	Reserved
33	RLIN30 receive data input
34	RLIN31 receive data input
35	RLIN32 receive data input
36	RLIN33 receive data input
37 to 64	Reserved

Table 41.87 PIC2GTMINENk List of Function (2/2)

	Function
65	ENCA0 encoder input (phase A)
66	ENCA0 encoder input (phase B)
67	ENCA0 encoder input (phase Z)
68	ENCA0 compare 0 match or capture 0 interrupt signal
69	ENCA0 compare 1 match or capture 1 interrupt signal
70	ENCA0 clear interrupt signal by encoder input (phase Z)
71	ENCA0 count clock
72	ENCA0 down-count enable signal
73	ENCA1 encoder input (phase A)
74	ENCA1 encoder input (phase B)
75	ENCA1 encoder input (phase Z)
76	ENCA1 compare 0 match or capture 0 interrupt signal
77	ENCA1 compare 1 match or capture 1 interrupt signal
78	ENCA1 clear interrupt signal by encoder input (phase Z)
79	ENCA1 count clock
80	ENCA1 down-count enable signal
81 to 127	Reserved

41.3.2.14 PIC2GTMINENk — GTM Timer Input Module (TIM) Source Select Register k (k = 1, 3, 5, 7)

The PIC2GTMINENk register selects a source of timer input of GTM (TIM) (k = 1, 3, 5, 7).

Access: This register can be read or written in 16-bit or 32-bit units.

Address: <PIC22_base> + $0B4_H$ (k = 1), <PIC22_base> + $0BC_H$ (k = 3),
<PIC22_base> + $0C4_H$ (k = 5), <PIC22_base> + $0CC_H$ (k = 7)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PIC2GTMINENk[31:24]								PIC2GTMINENk[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2GTMINENk[15:8]								PIC2GTMINENk[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.88 PIC2GTMINENk Register Contents

Bit Position	Bit Name	Function
31 to 24	PIC2GTMINENk [31:24]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((k-1) / 2)]_IN7) 0 to 127: Refer to **Table 41.89 PIC2GTMINENk List of Function**
23 to 16	PIC2GTMINENk [23:16]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((k-1) / 2)]_IN6) 0 to 127: Refer to **Table 41.89 PIC2GTMINENk List of Function**
15 to 8	PIC2GTMINENk [15:8]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((k-1) / 2)]_IN5) 0 to 127: Refer to **Table 41.89 PIC2GTMINENk List of Function**
7 to 0	PIC2GTMINENk [7:0]	Selects a GTM timer input signal from peripheral IPs. (GTM_TIM[((k-1) / 2)]_IN4) 0 to 127: Refer to **Table 41.89 PIC2GTMINENk List of Function**

Table 41.89 PIC2GTMINENk List of Function (1/2)

	Function
0	GPIO selected. (via port functions)
1	ADCJ0 scan group 0 (SG0) end interrupt
2	ADCJ0 scan group 1 (SG1) end interrupt
3	ADCJ0 scan group 2 (SG2) end interrupt
4	ADCJ0 scan group 3 (SG3) end interrupt
5	ADCJ0 scan group 4 (SG4) end interrupt
6	ADCJ1 scan group 0 (SG0) end interrupt
7	ADCJ1 scan group 1 (SG1) end interrupt
8	ADCJ1 scan group 2 (SG2) end interrupt
9	ADCJ1 scan group 3 (SG3) end interrupt
10	ADCJ1 scan group 4 (SG4) end interrupt
11 to 32	Reserved
33	RLIN30 receive data input
34	RLIN31 receive data input

Table 41.89 PIC2GTMINENk List of Function (2/2)

	Function
35	RLIN32 receive data input
36	RLIN33 receive data input
37 to 64	Reserved
65	ENCA0 encoder input (phase A)
66	ENCA0 encoder input (phase B)
67	ENCA0 encoder input (phase Z)
68	ENCA0 compare 0 match or capture 0 interrupt signal
69	ENCA0 compare 1 match or capture 1 interrupt signal
70	ENCA0 clear interrupt signal by encoder input (phase Z)
71	ENCA0 count clock
72	ENCA0 down-count enable signal
73	ENCA1 encoder input (phase A)
74	ENCA1 encoder input (phase B)
75	ENCA1 encoder input (phase Z)
76	ENCA1 compare 0 match or capture 0 interrupt signal
77	ENCA1 compare 1 match or capture 1 interrupt signal
78	ENCA1 clear interrupt signal by encoder input (phase Z)
79	ENCA1 count clock
80	ENCA1 down-count enable signal
81 to 127	Reserved

41.3.2.15 PIC2ENCATCFG0 — Encoder Timer Trigger Output Configuration Register 0

The PIC2ENCATCFG0 register assigns timer output of GTM (ATOM) and interrupt of TIM to PIC2ENCAnTSEL (n = 0, 1).

Access: This register can be read or written in 16-bit units.

Address: <PIC22_base> + $0F0_H$

Value after reset: $0000\ 0000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ENCATCFG0[15:8]								PIC2ENCATCFG0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.90 PIC2ENCATCFG0 Register Contents

Bit Position	Bit Name	Function
15 to 8	PIC2ENCAT CFG0[15:8]	Selects the GTM ATOM output and TIM interrupt as a PIC2ENCAnTSEL Bit 1 register configuration. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected. 177 + (i x 8) + x: INTGTMA0TIM[i][x] is selected.
7 to 0	PIC2ENCAT CFG0[7:0]	Selects the GTM ATOM output and TIM interrupt as a PIC2ENCAnTSEL Bit 0 register configuration. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected. 177 + (i x 8) + x: INTGTMA0TIM[i][x] is selected.

41.3.2.16 PIC2ENCATCFG1 — Encoder Timer Trigger Output Configuration Register 1

The PIC2ENCATCFG1 register assigns timer output of GTM (ATOM) and interrupt of TIM to PIC2ENCAnTSEL (n = 0, 1).

Access: This register can be read or written in 16-bit units.

Address: <PIC22_base> + $0F4_H$

Value after reset: $0000\ 0000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC2ENCATCFG1[15:8]								PIC2ENCATCFG1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.91 PIC2ENCATCFG1 Register Contents

Bit Position	Bit Name	Function
15 to 8	PIC2ENCAT CFG1[15:8]	Selects the GTM ATOM output and TIM interrupt as a PIC2ENCAnTSEL Bit 3 register configuration. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected. 177 + (i x 8) + x: INTGTMA0TIM[i][x] is selected.
7 to 0	PIC2ENCAT CFG1[7:0]	Selects the GTM ATOM output and TIM interrupt as a PIC2ENCAnTSEL Bit 2 register configuration. 0 to 96: GTM signal is not selected. 97 + (i x 8) + x: GTM_ATOM[i]_OUT[x] is selected. 177 + (i x 8) + x: INTGTMA0TIM[i][x] is selected.

41.3.2.17 PIC2ENCAISEN — Encoder Input Selection Control Register

The PIC2ENCAISEN register selects ENCAn input signals (n = 0, 1).

Access: This register can be read or written in 8-bit units.

Address: <PIC22_base> + $0F8_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIC2ENCAISEN[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 41.92 PIC2ENCAISEN Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PIC2ENCAISEN [1:0]	Selects the input pins (ENCAnE0, ENCAnE1, ENCAnEC) of ENCAn timer. (n = 0, 1) 00: Series connection 01: Parallel connection 10: Cascade connection 11: Reserved

41.3.3 Function

41.3.3.1 ADCJ Trigger Select Function

(1) Overview

The function allows generation of ADCJ hardware trigger signal for individual scan groups by the signals from each IP. The IPs can be selected from TAUD0, TAUD1, GTM, TAUJ0, TAUJ1, TAPA0, ENCA0, ENCA1, TSG30, TSG31 and ADCJn external trigger pins.

(2) Configuration

The ADCJ trigger select function is realized by using individual IPs and PIC2 in combination. The following figure shows the block diagram of ADCJ trigger select function.

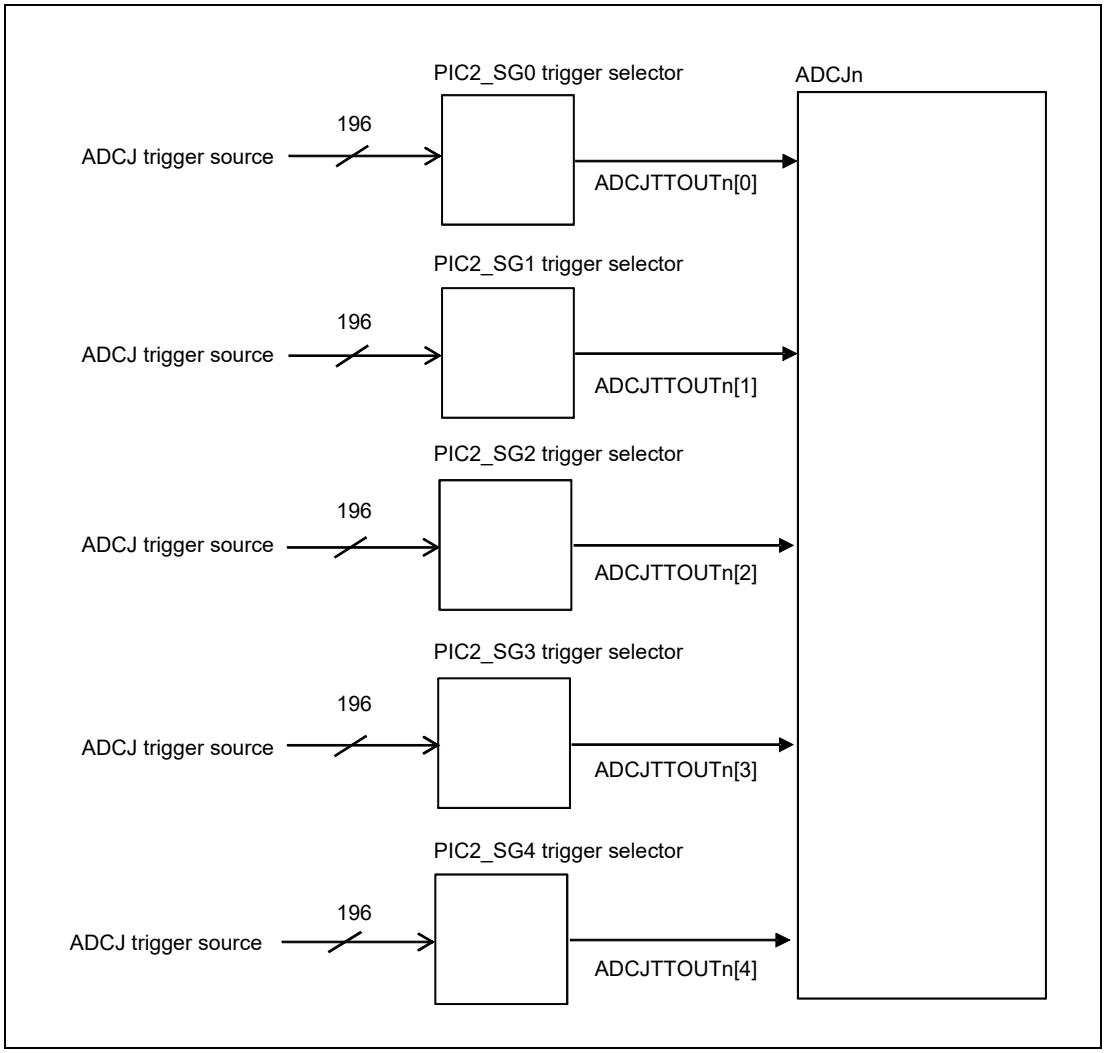
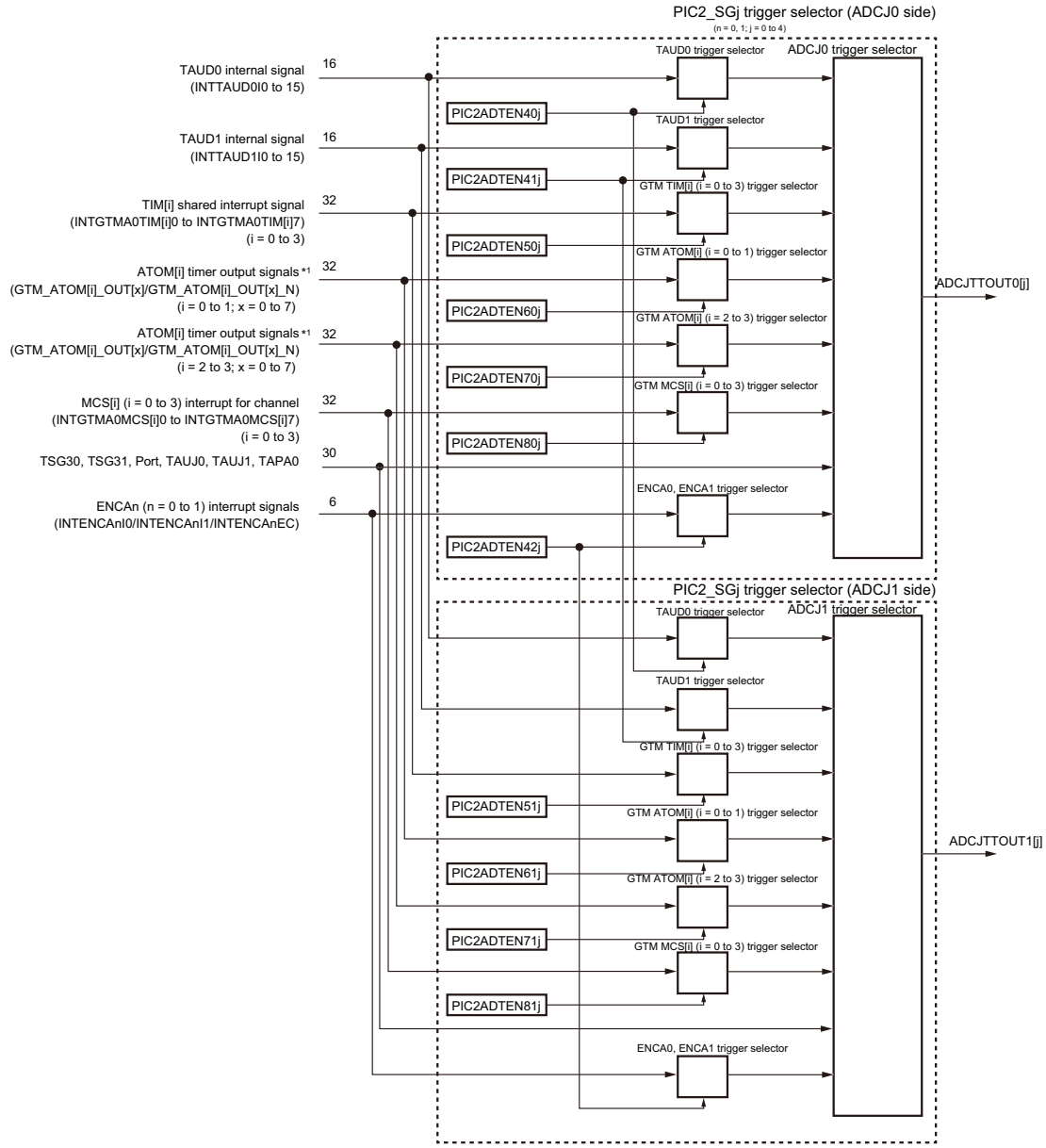


Figure 41.53 Block Diagram of ADCJ Trigger Select Function



Note 1. PIC2 detects only the rising edge from the above GTM signals and generates a pulse at the time. Each pulse is used as the trigger source for each ADCJn (n=0-1) instead of the above GTM signals. Therefore, to use falling edges of the above GTM signals to trigger the ADCJn scan group, select the rising edge detection by the PIC2ADCJnEDGSEL register, and generate a pulse at the falling edge with reference to one of the examples shown below.

- Invert the polarity of the GTM signal in the GTM, and select the inverted signal as the trigger source in the PIC2.
- Use the DTM in the GTM to generate a pulse synchronized with the falling edge of the GTM signal.
- Use a TIM in the GTM, route the ATOM signal to the TIM auxiliary input (TIM_AUX_IN), and generate an interrupt pulse (INTGTMA0TIM[i][x]) by detecting the falling edge of the ATOM with the TIM.

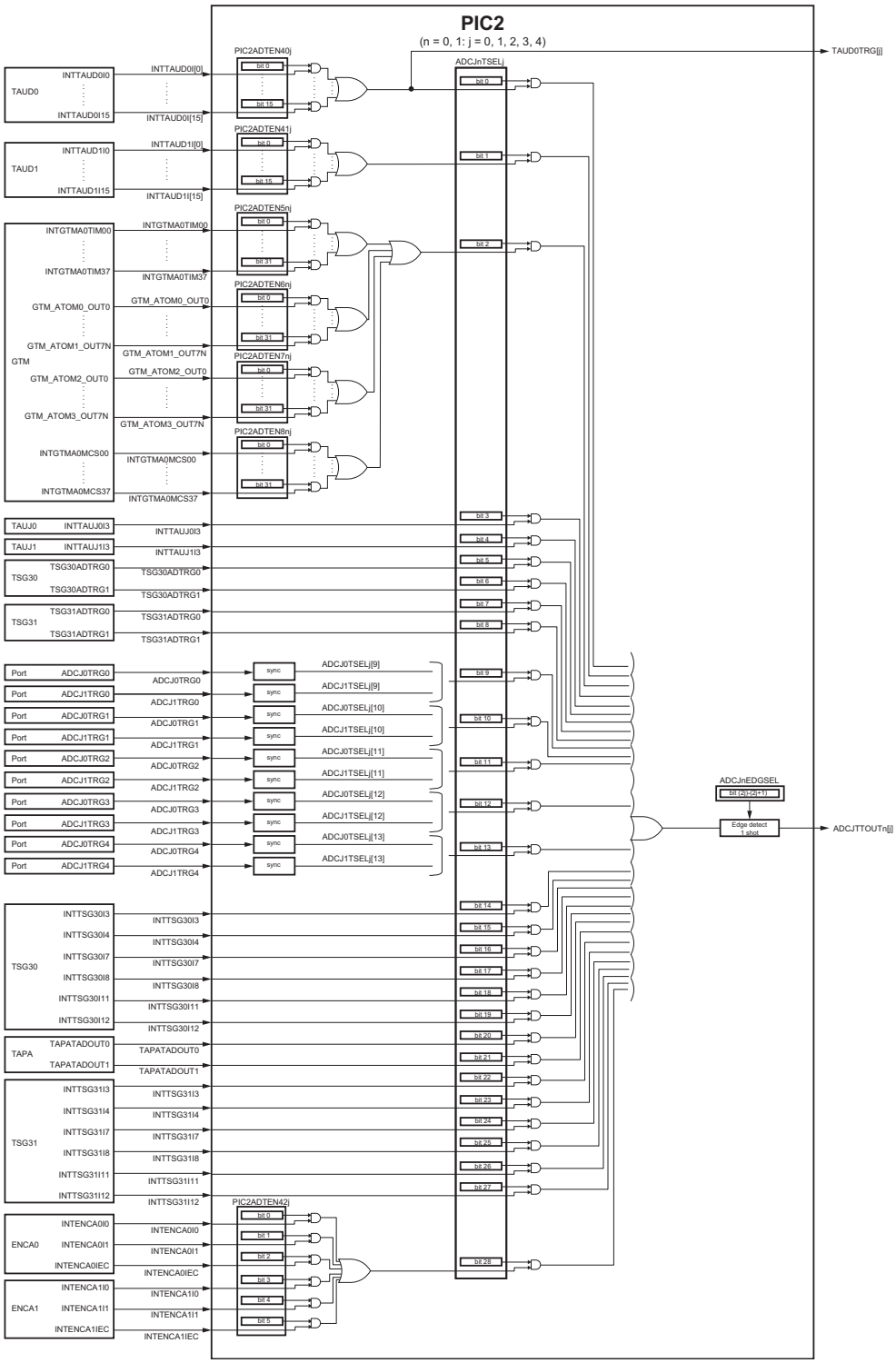
When using both rising and falling edges to trigger the ADCJn scan group, choose one of the following options:

- Generate the trigger pulses at the rising edge and the falling edge individually and merge them by using PIC2.
- Generate the pulses at both edges by the DTM in the GTM.

Figure 41.54 Block Diagram of PIC2_SGj Trigger Selection

(3) Registers

For the settings of the registers used in this function, see **Figure 41.55, Block Diagram of PIC2**, and from **Section 41.3.2.2** to **Section 41.3.2.4.**



Note 1. For external trigger signal, a delay of up to two clock cycles (12.5ns (80 MHz) @ 1clk) is generated because of the synchronization circuit (Sync) which needs to be considered when selecting an external trigger signal as the ADCJ trigger.

Figure 41.55 Block Diagram of PIC2

(4) Function

A/D trigger signal can be selected for individual ADCJn scan groups and output can be selected from rising edge, falling edge, and both edges. The TAUD trigger for the scan group with the same number is shared between ADCJ0 and ADCJ1.

(5) Flow

Set this function before starting A/D converter.

41.3.3.2 Baud Rate Measurement for an UART (RLIN3)

To measure the baud rate of the received data a connection from the RLIN3 soft macro data input (RX) to GTM input is made. The connection can be selected by a control register.

Table 41.93 Connections between RLIN3 and GTM

Signal source	GTM input	Supporting device									
		RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
RLIN30RX	GTM_TIM0_IN5	√	√	√	√	√	√	√	√	√	√
RLIN31RX	GTM_TIM0_IN6	√	√	√	√	√	√	√	√	—	—
RLIN32RX	GTM_TIM1_IN5	√	√	√	√	√	√	√	√	—	√
RLIN33RX	GTM_TIM1_IN6	√	√	√	√	√	√	√	√	√	√

41.3.3.3 Hi-Z Control Function Over External Pin for GTM Output

The I/O driven by GTM output can be set to Hi-Z over an external pin but also by the ECM module within 50 ns. The path to Hi-Z control of the I/O buffers can be enabled by a register in the PIC module.

The signals from the ECM and from the ESO pin can be masked independently from each other.

For this device, GTM has four output groups and each group has an independent Hi-Z control signal.

The controlled output signals are GTMAT[i]O[x] and GTMAT[i]O[x]N with i = 0 to 3 and x = 0 to 7.

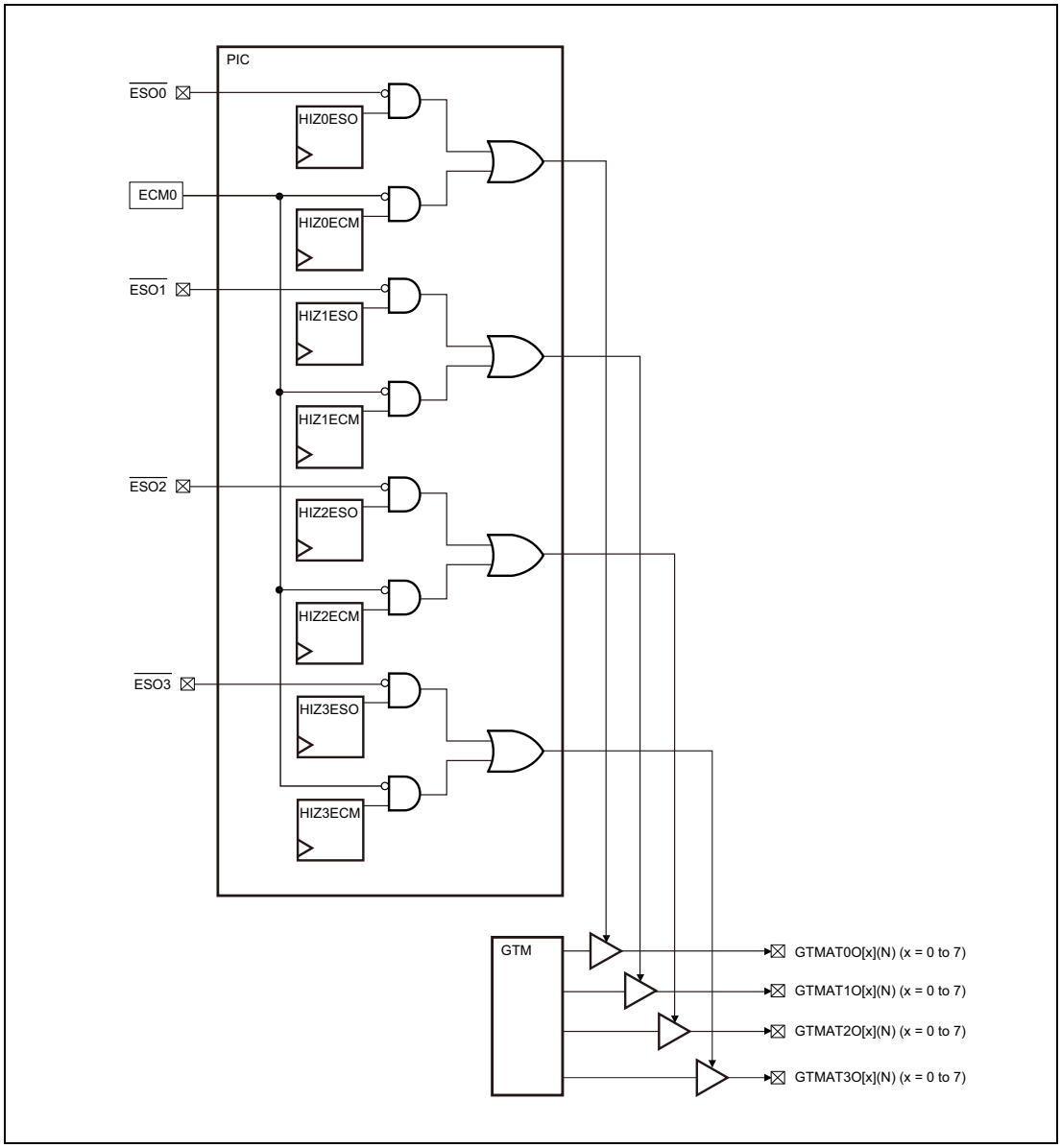


Figure 41.56 Hi-Z Control of GTM Output

41.3.3.4 GTM Output Monitor for PWM Diagnostic

This function helps to verify the output of the GTM by loop-back. The monitoring point is in the IO-buffer itself were the input path must be enabled over the port function. The signal is routed back to GTM input over the PIC. In the PIC this path can be enabled over a register. The block diagram is shown in **Figure 41.57**.

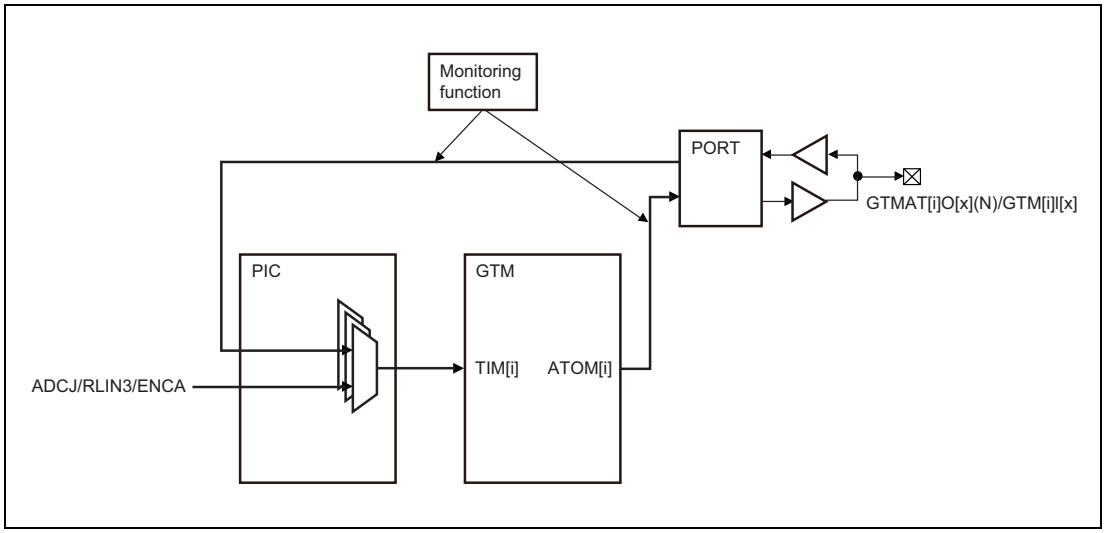


Figure 41.57 Block Diagram for GTM Output Monitoring Function

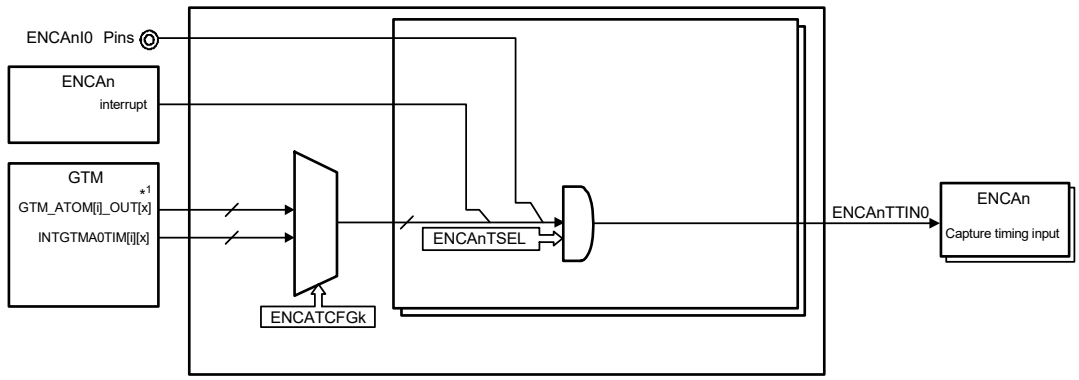
The assignment of the GTM outputs and the corresponding GTM inputs is bound to the pin multiplexing since the monitoring point is the pin itself. For details of the assignment, refer to **Section 2, Pin Functions**.

Note: For this function the respective port has to be set into bi-directional mode and TIM[i] inputs have to be configured accordingly.

41.3.3.5 ENCA Trigger Selection Function

ENCA has the trigger which captures the encoder counter value to a capture register. It's possible to map GTM output to capture trigger of ENCA. In this case GTM outputs are multiplexed.

- GTM trigger selection
  ENCA trigger can be generated from GTM_ATOM[i]_OUT[x]*1 and INTGTMA0TIM[i][x] (i = 0 to 3; x = 0 to 7).



Note 1. PIC2 detects only the rising edge from the above GTM signals and generates a pulse at the time.
Each pulse is used as the trigger source for each ENCAn instead of the above ATOM signals.

Therefore, to use falling edges of the above ATOM signals to capture the encoder counter value, select the rising edge detection by the ENCAnIOC0 register, and generate a pulse at the falling edge with reference to one of the example as shown below.
- Invert the polarity of the ATOM signal in the GTM, and select the inverted signal as the trigger source in the PIC2.
- Use the DTM in the GTM to generate a pulse synchronized with the falling edge of the GTM signal.
- Use a TIM in the GTM, route the ATOM signal to the TIM auxiliary input (TIM_AUX_IN), and generate an interrupt pulse (INTGTMA0TIM[i][x]) by detecting the falling edge of the ATOM signal with the TIM.

When using both rising and falling edges to capture the encoder counter value, choose one of the following options:
- Generate the trigger pulses at the rising edge and the falling edge individually and merge them by using PIC2.
- Generate the pulses at both edges by the DTM from ATOM signals.
- Generate TIM interrupts at the rising and falling edges of ATOM signals.

Figure 41.58 Block Diagram of Trigger Selection Function of ENCA

41.3.3.6 PSI5-S Timestamp and the Sync Pulse Signal Selection Function

PSI5-S has Time stamp and Sync Pulse Generation. GTM outputs can be mapped to Time stamp (clock, enable, clear) and Sync Pulse of PSI5-S. GTM outputs in this case are multiplexed.

- GTM trigger selection
  PSI5-S time stamp trigger and sync pulse can be generated from GTM_ATOM[i]_OUT[x] (i = 0 to 3; x = 0 to 7).

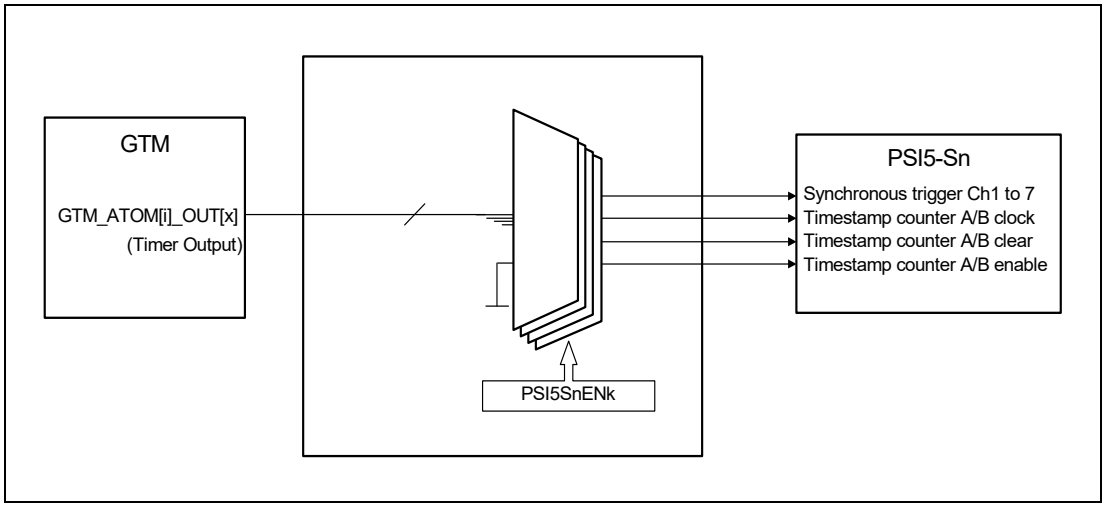


Figure 41.59 Block Diagram of Trigger Selection Function of PSI5-S

41.3.3.7 GTM Timer Input (TIM) Selection Function

Source of GTM Timer Input signals for TIM[i] (i: 0 to 3) can be selected from

- PORT
  It's possible to select the GTM timer input chosen in GPIO. (Defaults)
- ADCJ
  It's possible to select AD conversion end interrupt as timer input of GTM.
- ENCA
  It's possible to select count clock and down-count enable signal from encoder input (phase A, phase B, phase Z) and compare capture interrupt and phase Z edge detection as timer input of GTM.
- RLIN
  It's possible to select RLIN3 soft macro data input (RX) as timer input of GTM.

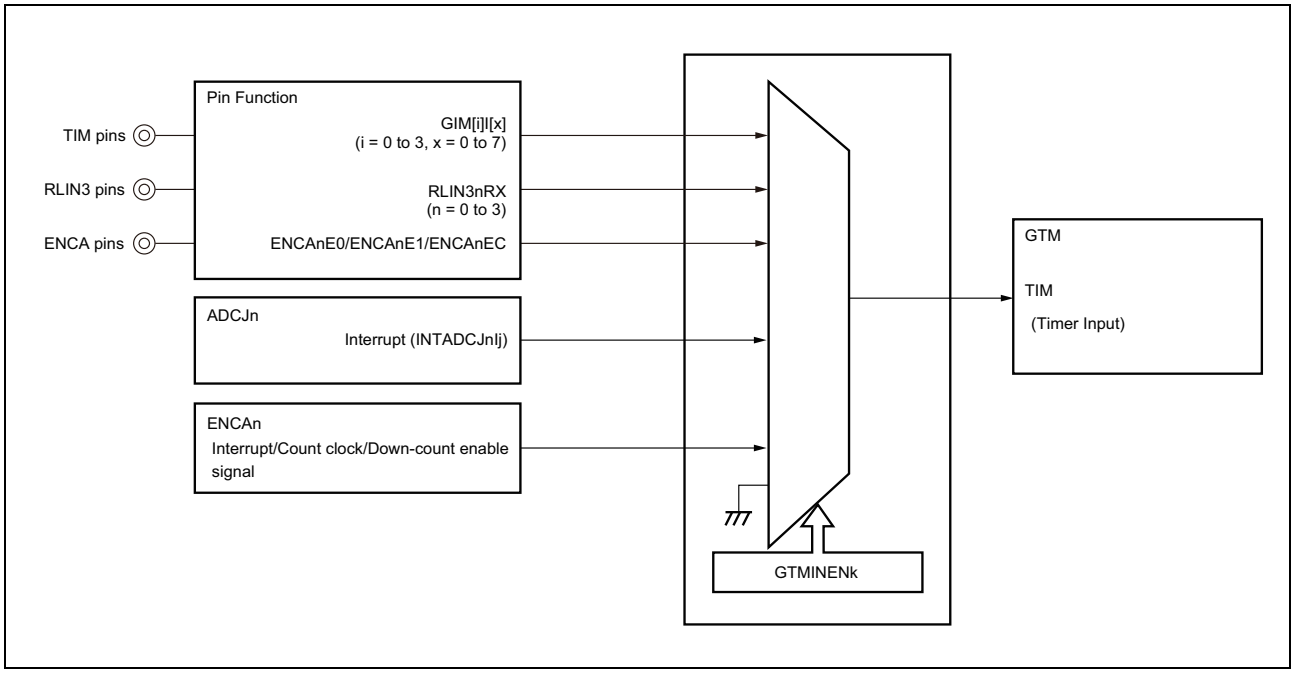


Figure 41.60 Block Diagram of Trigger Selection Function of GTM

41.3.3.8 ENCA Encoder Input Selection

It's possible to select encoder input of an encoder timer (phase A, phase B, phase Z) to ENCAn.

Encoder input can be selected as series, parallel or cascade connection.

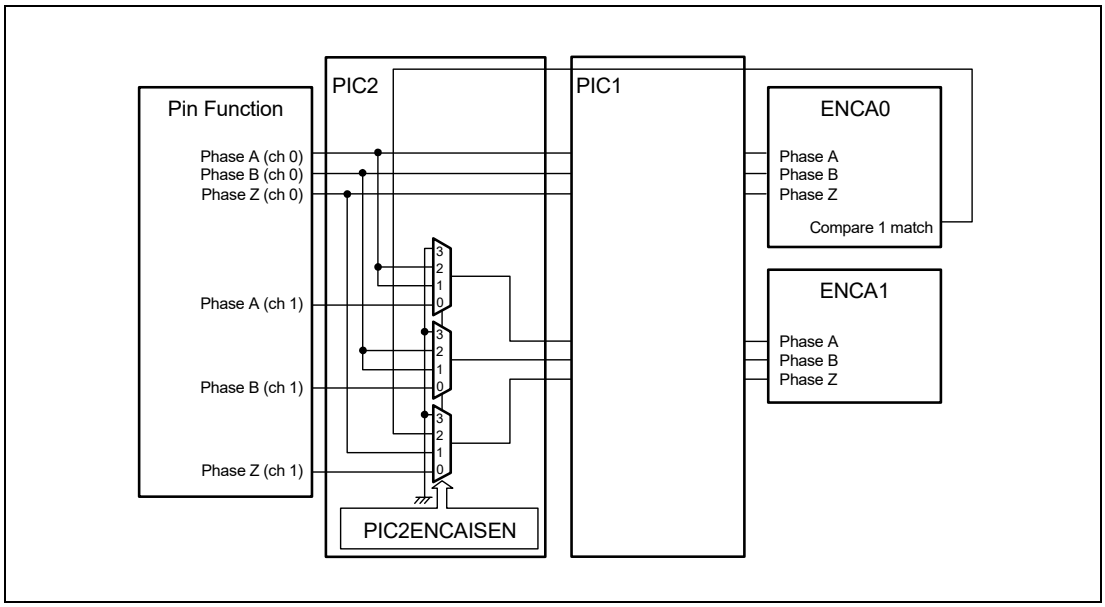


Figure 41.61 Block Diagram of ENCA Input Selection Function

Section 42 PWM Output/Diagnostic (PWM-Diag)

This section contains a generic description of the PWM output/diagnostic function (PWM-Diag).

The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the units constituting PWM-Diag.

42.1 Features PWM-Diag for RH850/U2A-EVA

42.1.1 Number of Units and Channels

The PWM-Diag unit consists of a PWBA block for generating clock signals, PWGC blocks that generate PWM signals, and a PWSD block for generating triggers for A/D conversion. The numbers of individual units are listed below.

Each PWGC unit has one PWM channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 42.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	PWBAn									
Number of Units	96 (n = 0 to 95)	96 (n = 0 to 95)	96 (n = 0 to 95)	96 (n = 0 to 95)	96 (n = 0 to 95)	96 (n = 0 to 95)	96 (n = 0 to 95)	76 (n = 0 to 8, 11 to 14, 19 to 21, 26 to 69, 71 to 86)	50 (n = 0, 14, 20, 21, 27, 28, 30, 31, 33 to 46, 49, 52 to 63, 65, 66, 68,69, 73, 74, 77 to 80, 82 to 86)	64 (n = 0, 2, 11 to 21, 23, 26 to 31, 34 to 40, 44, 45, 48 to 69, 71 to 83)
Name	PWGCn									
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	PWSDn									

Table 42.2 Indices (1/2)

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock and PWGC period setting registers are identified by the index “m” (m = 0 to 3); for example, PWBAnBRSm indicates the PWMCLKm clock cycle configuration register.
x	An A/D converter configuration register number or A/D conversion result corresponding to a PWM-Diag channel is identified by the index “x”; for example, PWSDnPVCRx (x = 00 to 95).
j	Registers storing trigger channel numbers (encoded value) from PWGCn are identified by the index “j”; for example, the PWSDnQUEj register (j = 0 to 7).
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGCk register (k = 0 to 2).

Table 42.2 Indices (2/2)

Index	Description
h	Sets of registers where each has the same function are identified by the index “h”; for example, the PWGCINTFhk register (h = 0 to 3).
v	Throughout this section, individual units of A/D converter are identified by the index “v”; for example, ADCJvPWDVCR register (v = 0 to 2).
q	Sets of registers where each has the same function are identified by the index “q”; for example, the PWGCPRDSLq register (q = 0 to 5).

The following table shows values indicated by the indices of each product.

Table 42.3 Indices of Products

Indices of each product									
RH850/U2A-EVA (516 pins)	**RH850/ U2A16 (516 pins)**	**RH850/ U2A16 (373 pins)**	**RH850/ U2A16 (292 pins)**	**RH850/ U2A8 (373 pins)**	**RH850/ U2A8 (292 pins)**	**RH850/ U2A6 (292 pins)**	**RH850/ U2A6 (176 pins)**	**RH850/ U2A6 (156 pins)**	**RH850/ U2A6 (144 pins)**
m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3	m = 0 to 3
x = 00 to 95	x = 00 to 95	x = 00 to 95	x = 00 to 95	x = 00 to 95	x = 00 to 95	x = 00 to 95	x = 0 to 8, 11 to 14, 19 to 21, 26 to 69, 71 to 86	x = 0, 14, 20, 21, 27, 28, 30, 31, 33 to 46, 49, 52 to 63, 65, 66, 68,69, 73, 74, 77 to 80, 82 to 86	x = 0, 2, 11 to 21, 23, 26 to 31, 34 to 40, 44, 45, 48 to 69, 71 to 83
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2	k = 0 to 2
h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3	h = 0 to 3
v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 2	v = 0 to 1	v = 0 to 2
q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5	q = 0 to 5

42.1.2 Register Base Addresses

PWM-Diag base addresses are listed in the following table.

PWM-Diag register addresses are given as offsets from the base addresses.

Table 42.4 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PWBAn_base>	FFEF F000$_H$	P-Bus Group 6L
<PWGCn_base>	FFF0 0000$_H$ + 200$_H$ × n	P-Bus Group 6L
<PWSDn_base>	FFF0 E000$_H$	P-Bus Group 6L
<SLPW_base>	FFF0 C000$_H$	P-Bus Group 6L
<PWGCINTF_base>	FFF0 C200$_H$	P-Bus Group 6L

42.1.3 Clock Supply

The PWM-Diag clock supply is shown in the following table.

Table 42.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
PWBAn	PCLK	CLK_LSB	Peripheral low speed clock
PWGCn	PCLK	CLK_LSB	Peripheral low speed clock
PWSDn	PCLK	CLK_LSB	Peripheral low speed clock
SLPW	PCLK	CLK_LSB	Peripheral low speed clock
PWGCINTFhk	PCLK	CLK_LSB	Peripheral low speed clock

42.1.4 Interrupt Requests and Error Notifications

PWM-Diag interrupt requests are listed in the following table.

Table 42.6 Interrupt Requests (1/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA/DTS Trigger Number
PWGC_INT0	PWGC0 interrupt	660, 663, 666, 669	—
PWGC_INT1	PWGC1 interrupt	660, 663, 666, 669	—
PWGC_INT2	PWGC2 interrupt	660, 663, 666, 669	—
PWGC_INT3	PWGC3 interrupt	660, 663, 666, 669	—
PWGC_INT4	PWGC4 interrupt	660, 663, 666, 669	—
PWGC_INT5	PWGC5 interrupt	660, 663, 666, 669	—
PWGC_INT6	PWGC6 interrupt	660, 663, 666, 669	—
PWGC_INT7	PWGC7 interrupt	660, 663, 666, 669	—
PWGC_INT8	PWGC8 interrupt	660, 663, 666, 669	—
PWGC_INT9	PWGC9 interrupt	660, 663, 666, 669	—
PWGC_INT10	PWGC10 interrupt	660, 663, 666, 669	—
PWGC_INT11	PWGC11 interrupt	660, 663, 666, 669	—
PWGC_INT12	PWGC12 interrupt	660, 663, 666, 669	—
PWGC_INT13	PWGC13 interrupt	660, 663, 666, 669	—
PWGC_INT14	PWGC14 interrupt	660, 663, 666, 669	—
PWGC_INT15	PWGC15 interrupt	660, 663, 666, 669	—
PWGC_INT16	PWGC16 interrupt	660, 663, 666, 669	—
PWGC_INT17	PWGC17 interrupt	660, 663, 666, 669	—
PWGC_INT18	PWGC18 interrupt	660, 663, 666, 669	—
PWGC_INT19	PWGC19 interrupt	660, 663, 666, 669	—
PWGC_INT20	PWGC20 interrupt	660, 663, 666, 669	—
PWGC_INT21	PWGC21 interrupt	660, 663, 666, 669	—
PWGC_INT22	PWGC22 interrupt	660, 663, 666, 669	—
PWGC_INT23	PWGC23 interrupt	660, 663, 666, 669	—
PWGC_INT24	PWGC24 interrupt	660, 663, 666, 669	—
PWGC_INT25	PWGC25 interrupt	660, 663, 666, 669	—
PWGC_INT26	PWGC26 interrupt	660, 663, 666, 669	—
PWGC_INT27	PWGC27 interrupt	660, 663, 666, 669	—

Table 42.6 Interrupt Requests (2/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA/DTS Trigger Number
PWGC_INT28	PWGC28 interrupt	660, 663, 666, 669	—
PWGC_INT29	PWGC29 interrupt	660, 663, 666, 669	—
PWGC_INT30	PWGC30 interrupt	660, 663, 666, 669	—
PWGC_INT31	PWGC31 interrupt	660, 663, 666, 669	—
PWGC_INT32	PWGC32 interrupt	661, 664, 667, 670	—
PWGC_INT33	PWGC33 interrupt	661, 664, 667, 670	—
PWGC_INT34	PWGC34 interrupt	661, 664, 667, 670	—
PWGC_INT35	PWGC35 interrupt	661, 664, 667, 670	—
PWGC_INT36	PWGC36 interrupt	661, 664, 667, 670	—
PWGC_INT37	PWGC37 interrupt	661, 664, 667, 670	—
PWGC_INT38	PWGC38 interrupt	661, 664, 667, 670	—
PWGC_INT39	PWGC39 interrupt	661, 664, 667, 670	—
PWGC_INT40	PWGC40 interrupt	661, 664, 667, 670	—
PWGC_INT41	PWGC41 interrupt	661, 664, 667, 670	—
PWGC_INT42	PWGC42 interrupt	661, 664, 667, 670	—
PWGC_INT43	PWGC43 interrupt	661, 664, 667, 670	—
PWGC_INT44	PWGC44 interrupt	661, 664, 667, 670	—
PWGC_INT45	PWGC45 interrupt	661, 664, 667, 670	—
PWGC_INT46	PWGC46 interrupt	661, 664, 667, 670	—
PWGC_INT47	PWGC47 interrupt	661, 664, 667, 670	—
PWGC_INT48	PWGC48 interrupt	661, 664, 667, 670	—
PWGC_INT49	PWGC49 interrupt	661, 664, 667, 670	—
PWGC_INT50	PWGC50 interrupt	661, 664, 667, 670	—
PWGC_INT51	PWGC51 interrupt	661, 664, 667, 670	—
PWGC_INT52	PWGC52 interrupt	661, 664, 667, 670	—
PWGC_INT53	PWGC53 interrupt	661, 664, 667, 670	—
PWGC_INT54	PWGC54 interrupt	661, 664, 667, 670	—
PWGC_INT55	PWGC55 interrupt	661, 664, 667, 670	—
PWGC_INT56	PWGC56 interrupt	661, 664, 667, 670	—
PWGC_INT57	PWGC57 interrupt	661, 664, 667, 670	—
PWGC_INT58	PWGC58 interrupt	661, 664, 667, 670	—
PWGC_INT59	PWGC59 interrupt	661, 664, 667, 670	—
PWGC_INT60	PWGC60 interrupt	661, 664, 667, 670	—
PWGC_INT61	PWGC61 interrupt	661, 664, 667, 670	—
PWGC_INT62	PWGC62 interrupt	661, 664, 667, 670	—
PWGC_INT63	PWGC63 interrupt	661, 664, 667, 670	—
PWGC_INT64	PWGC64 interrupt	662, 665, 668, 671	—
PWGC_INT65	PWGC65 interrupt	662, 665, 668, 671	—
PWGC_INT66	PWGC66 interrupt	662, 665, 668, 671	—
PWGC_INT67	PWGC67 interrupt	662, 665, 668, 671	—
PWGC_INT68	PWGC68 interrupt	662, 665, 668, 671	—
PWGC_INT69	PWGC69 interrupt	662, 665, 668, 671	—
PWGC_INT70	PWGC70 interrupt	662, 665, 668, 671	—

Table 42.6 Interrupt Requests (3/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA/DTS Trigger Number
PWGC_INT71	PWGC71 interrupt	662, 665, 668, 671	—
PWGC_INT72	PWGC72 interrupt	662, 665, 668, 671	—
PWGC_INT73	PWGC73 interrupt	662, 665, 668, 671	—
PWGC_INT74	PWGC74 interrupt	662, 665, 668, 671	—
PWGC_INT75	PWGC75 interrupt	662, 665, 668, 671	—
PWGC_INT76	PWGC76 interrupt	662, 665, 668, 671	—
PWGC_INT77	PWGC77 interrupt	662, 665, 668, 671	—
PWGC_INT78	PWGC78 interrupt	662, 665, 668, 671	—
PWGC_INT79	PWGC79 interrupt	662, 665, 668, 671	—
PWGC_INT80	PWGC80 interrupt	662, 665, 668, 671	—
PWGC_INT81	PWGC81 interrupt	662, 665, 668, 671	—
PWGC_INT82	PWGC82 interrupt	662, 665, 668, 671	—
PWGC_INT83	PWGC83 interrupt	662, 665, 668, 671	—
PWGC_INT84	PWGC84 interrupt	662, 665, 668, 671	—
PWGC_INT85	PWGC85 interrupt	662, 665, 668, 671	—
PWGC_INT86	PWGC86 interrupt	662, 665, 668, 671	—
PWGC_INT87	PWGC87 interrupt	662, 665, 668, 671	—
PWGC_INT88	PWGC88 interrupt	662, 665, 668, 671	—
PWGC_INT89	PWGC89 interrupt	662, 665, 668, 671	—
PWGC_INT90	PWGC90 interrupt	662, 665, 668, 671	—
PWGC_INT91	PWGC91 interrupt	662, 665, 668, 671	—
PWGC_INT92	PWGC92 interrupt	662, 665, 668, 671	—
PWGC_INT93	PWGC93 interrupt	662, 665, 668, 671	—
PWGC_INT94	PWGC94 interrupt	662, 665, 668, 671	—
PWGC_INT95	PWGC95 interrupt	662, 665, 668, 671	—
PWSD_INT_QFULL	PWSD queue full interrupt	672	—

This module has no error notifications.

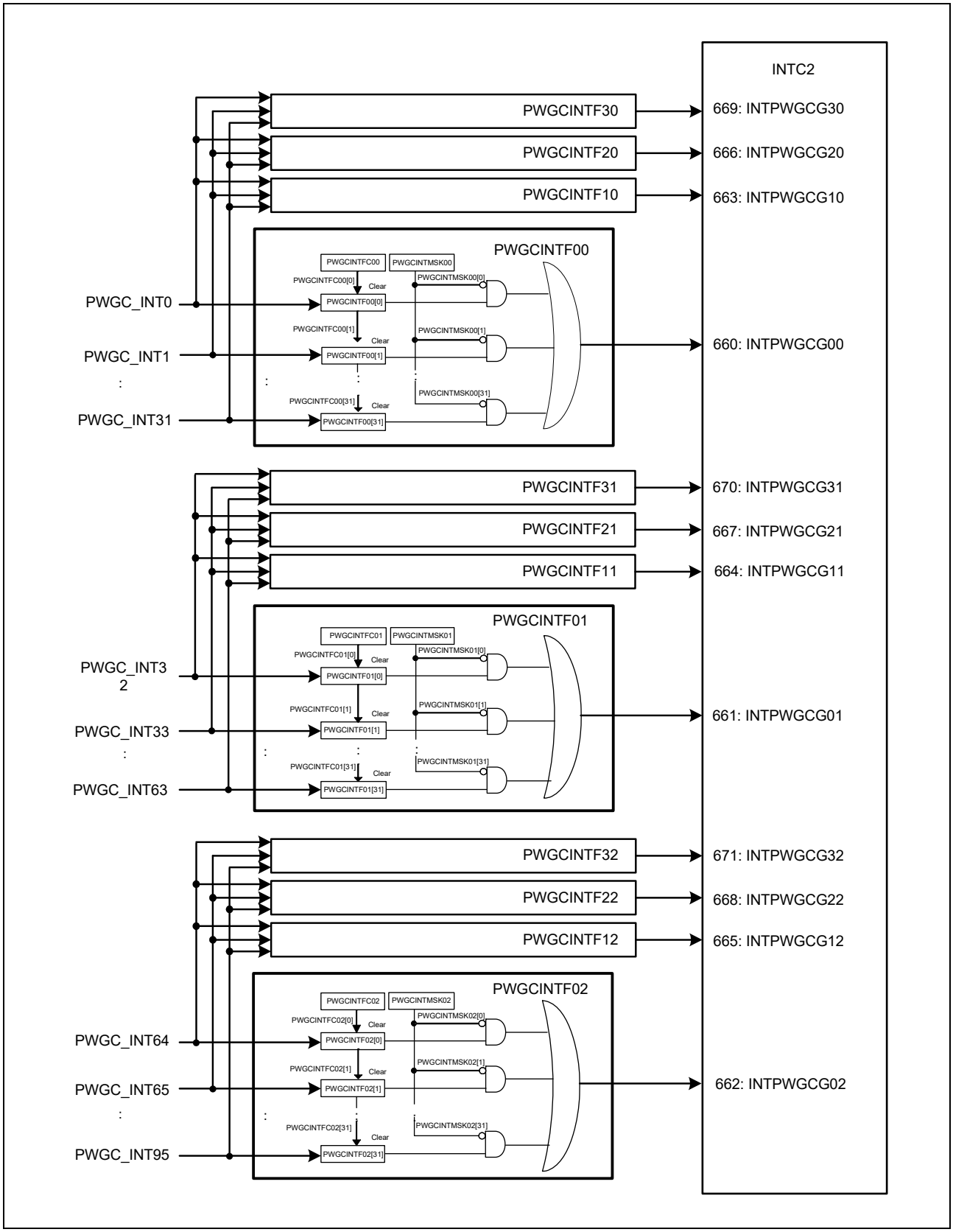


Figure 42.1 PWGC Interrupt Connection Image

42.1.5 Reset Sources

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

Table 42.7 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
PWBA	All registers	√	√	√	√	√	√	—
PWGCn	All registers	√	√	√	√	√	√	—
PWSD	All registers	√	√	√	√	√	√	—
SLPW	All registers	√	√	√	√	√	√	—
PWGCINTF	All registers	√	√	√	√	√	√	—

42.1.6 External Input/Output Signals

External input/output signals of the PWM-Diag are listed below.

Table 42.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
PWGC_TOUTn (Unit: PWGC)	PWGC unit n output	PWGCnO

42.1.7 Internal Output Signals

The I/O signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

Table 42.9 Internal Output Signals

Unit Signal Name	Description	Connected to
PWBA		
PWMCLK0	PWGC count clock 0	PWGCn
PWMCLK1	PWGC count clock 1	PWGCn
PWMCLK2	PWGC count clock 2	PWGCn
PWMCLK3	PWGC count clock 3	PWGCn
PWGCn		
PWGC_TRGOUTn	PWGCn trigger	PWSD0
PWSD0		
PWSD_ADTRGv	A/D converter unit select signal	ADCJ0, ADCJ1, ADCJ2
PWSD_PVCR_VALUE[17:0]	A/D converter control signal	ADCJ0, ADCJ1, ADCJ2
ADCJv		
ADC_CONV_ENDv	A/D conversion completion signal	PWSD0
ADC_PWDDRv[15:0]	A/D conversion result signal	PWSD0

42.2 Overview

42.2.1 Functional Overview

This function is comprised of four types of units: clock divider (PWBA), PWM generator (PWGC), A/D conversion trigger select function (PWSD), and A/D converter (ADCJ).

PWBA

- Clock divider
  PWBA generates a PWMCLKm count clock signal by frequency division of PCLK and supplies it to the PWM generator PWGC.
  The cycle of the PWMCLKm count clock signal can be calculated from the setting of the PWBAnBRSm register by the equation below.
  - When the PWBAnBRSm = 1 to 2047
    PWMCLKm count clock cycle = (PWBAnBRSm value × 2) × PCLK cycle
  - When the PWBAnBRSm = 0
    PWMCLKm count clock cycle = PCLK cycle

PWGC

PWGC outputs PWM waveforms and A/D conversion trigger to PWSD by using the clock PWMCLKm input from PWBA.

- PWM waveform output PWGC_TOUTn
  This generator outputs PWM waveforms from the PWGC_TOUTn pin. The PWM cycle is controlled by the match timing of PWGCnCNT register value and the PWGC_PERIOD input value can be set by PWGCPRDm, PWGCPRDSLq register setting. Set the high-level period of PWM output in the PWGCnCSDR and PWGCnCRDR registers.

  The PWM waveform cycle and duty can be calculated by the equations below.

  PWM waveform cycle = (PWGC_PERIOD + 1) × PWMCLKm count clock cycle

  When PWGCnCRDR[15:0] > PWGCnCSDR[15:0],

  High-level period of PWM waveform =
  (PWGCnCRDR register value – PWGCnCSDR register value)
  × PWMCLKm count clock cycle

  PWM waveform duty (%) = High-level period of PWM waveform/PWM waveform cycle × 100 = (PWGCnCRDR register value – PWGCnCSDR register value)/(PWGC_PERIOD + 1) × 100

  Note that the PWM output is fixed to the low level when the PWGCnCRDR register value is equal to the PWGCnCSDR register value.

  When 1_xxxx_H or value higher than PWGC_PERIOD is set in the PWGCnCRDR register (i.e. bit 16 is set to 1), the PWM output is fixed to the high level.

  During counting state, when PWGCnFOT is set to 1, PWGC_TOUTn is forcibly fixed to the level of selected by PWGCnFOS from the timing of writing to PWGCnFOT to the timing of PWGCnCNT = PWGCnCSBR.

  When counter is stopped during forcible PWGC_TOUTn output level fix, PWGC_TOUTn becomes low level at the timing of stop.

The output level during forcible PWGC_TOUTn output level fix is not affected by changing the value of PWGCnFOS.

- A/D conversion trigger output PWGC_TRGOUTn
  The A/D conversion trigger signal PWGC_TRGOUTn for PWSD is generated when the PWGCnCTDR register value and the PWGCnCNT register value match and its timing can be set by PWGCnOCL.
  The output enable/disable of PWGC_TRGOUTn is controlled by PWGCnTCR register.
  The timing can be calculated by the equation below.

  A/D conversion trigger signal generation timing = PWGCnCTDR register value × PWMCLKm count clock cycle

- PWGC interrupt request signal PWGC_INTn
  PWGC generates the interrupt request signal PWGC_INTn at the timing which PWGCn_CNT value matches with PWGCnCSBR value (regardless of the state of PWGC_TOUT) or PWGCn_CNT value matches with PWGCn_CRBR or PWGCnCNT matches with 0 during PWM 100% output. The time of generation of PWGC_INTn depends on PWGCnCTL[15:14].

PWSD

PWSD transmits the required setting information to the A/D converter and outputs the A/D conversion start trigger, based on the A/D conversion trigger signal PWGC_TRGOUTn from the PWM generator (PWGC).

- A/D conversion control by PWSD
  PWSD outputs the information required for the A/D conversion, which is set in the corresponding PWSDnPVCRx register for the channel number of the trigger input from PWGCn, (i.e., information on ADC physical channel, external MPX control, and error detection level selection) to the A/D converter.
  At the same timing, A/D conversion trigger (PWSD_ADTRGv) is output to ADCJ0, ADCJ1 or ADCJ2. (A maximum of eight input trigger signal PWGC_TRGOUTn data received during A/D conversion are stored and kept in PWSDnQUE.)
  The setting information to be output to the A/D converter is kept until the next trigger is generated.
  When the A/D conversion triggered by the PWM-Diag function is completed in the A/D converter, PWSD triggers the next A/D conversion based on the data stored in the PWSDnQUE register.

- Queuing of A/D conversion triggers from PWGC
  The A/D conversion trigger signal (PWGC_TRGOUTn) input from PWGCn is stored in the PWSDnQUEj register as a channel number. The PWSDnQUEj register stores a maximum of eight channel numbers of the A/D conversion trigger signal PWGC_TRGOUTn received during A/D conversion in a queue structure.
  A PWSD queue full interrupt occurs in the following states, when the queue of the PWSDnQUEj register becomes full:
  - A trigger number is written to PWSDnQUE7
  - A trigger number has already been written to PWSDnQUE7 and cannot be written when PWGC_TRGOUTn is input.

- Storing A/D conversion result
  The A/D conversion result is stored in PWSDnPWDDIRx register.

ADCJ

A/D conversion is executed upon receipt of information required for A/D conversion and A/D conversion trigger from PWSD.

A/D conversion is executed using the PWM-Diag-dedicated scan group; on completion of the A/D conversion, it is reported to the PWSD.

For the basic operation of the A/D converter, see **Section 43, Analog to Digital Converter (ADCJ)**.

For the A/D converter operation with the PWM-Diag function, see **Section 43.4.16, PWM-Diag [Example of PWM-Diag]**.

42.2.2 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.

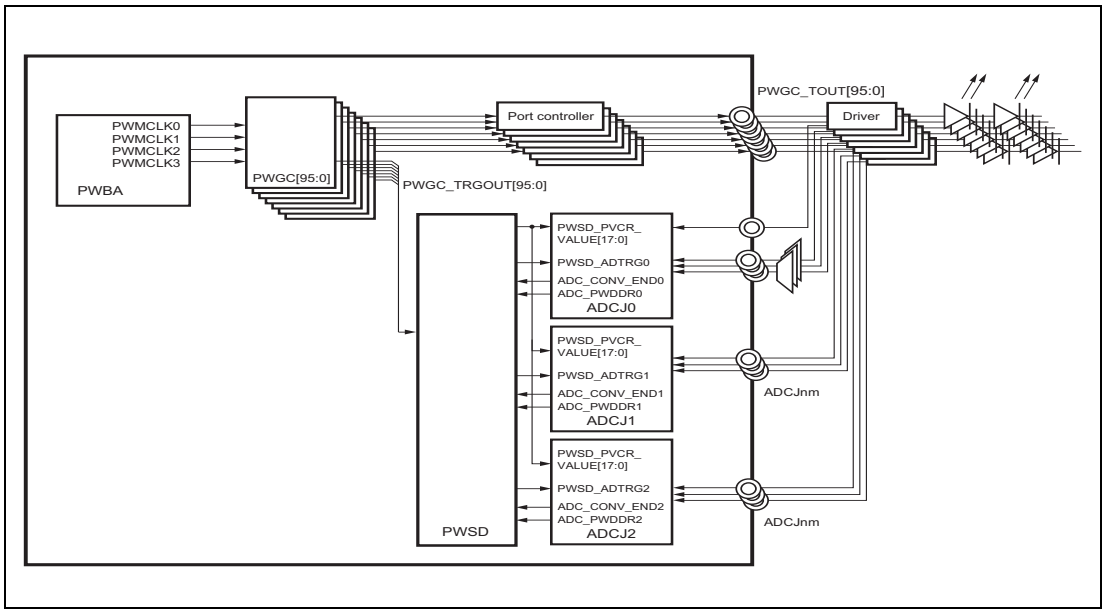


Figure 42.2 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter

42.3 Registers

42.3.1 List of Registers

PWM Output/Diagnostic registers are listed in the following table.

<PWBAn_base>, <PWSDn_base>, and <PWGCn_base> are defined in **Section 42.1.2, Register Base Addresses**.

Table 42.10 List of Registers (1/2)

Module name	Register name	Symbol	Address	Access size	Access Protection	
					PBG	Other
PWBAn	PWMCLKm cycle configuration register	PWBAnBRSm	<PWBAn_base> + 0004_H × m	16	PBG6L0#5	—
	PWMCLKm enable status register	PWBAnTE	<PWBAn_base> + 0010_H	8	PBG6L0#5	—
	PWMCLKm start trigger register	PWBAnTS	<PWBAn_base> + 0014_H	8	PBG6L0#5	—
	PWMCLKm stop trigger register	PWBAnTT	<PWBAn_base> + 0018_H	8	PBG6L0#5	—
PWGCn	PWM output set condition register	PWGCnCSDR	<PWGCn_base> + 0000_H	16	PBG6L0#5	—
	PWM output reset condition register	PWGCnCRDR	<PWGCn_base> + 0004_H	32	PBG6L0#5	—
	PWGC_TRGOUTn generation condition register	PWGCnCTDR	<PWGCn_base> + 0008_H	16	PBG6L0#5	—
	Buffer register reload trigger register	PWGCnRDT	<PWGCn_base> + $000C_H$	8	PBG6L0#5	—
	PWGC status register	PWGCnST	<PWGCn_base> + 0010_H	16	PBG6L0#5	—
	PWM cycle count register	PWGCnCNT	<PWGCn_base> + 0014_H	16	PBG6L0#5	—
	PWGC_TRGOUTn control register	PWGCnTCR	<PWGCn_base> + 0018_H	8	PBG6L0#5	—
	PWGCnTCR buffer register	PWGCnTCBR	<PWGCn_base> + $001C_H$	8	PBG6L0#5	—
	PWGC control register	PWGCnCTL	<PWGCn_base> + 0020_H	16	PBG6L0#5	—
	PWGCnCSDR buffer register	PWGCnCSBR	<PWGCn_base> + 0024_H	16	PBG6L0#5	—
	PWGCnCRDR buffer register	PWGCnCRBR	<PWGCn_base> + 0028_H	32	PBG6L0#5	—
	PWGCnCTDR buffer register	PWGCnCTBR	<PWGCn_base> + $002C_H$	16	PBG6L0#5	—
	Forcible PWGC_TOUTn output level fix trigger register	PWGCnFOT	<PWGCn_base> + $003C_H$	8	PBG6L0#5	—
SLPWG	PWGC synchronous trigger register	SLPWGCk	<SLPW_base> + k × 4_H	32	PBG6L0#5	—
	PWGC period setting register	PWGCPRDm	<SLPW_base> + $000C_H$ + m × 4_H	16	PBG6L0#5	—
	PWGC period selection register	PWGCPRDSLq	<SLPW_base> + $001C_H$ + q × 4_H	32	PBG6L0#5	—
PWSDn	PWSD control register	PWSDnCTL	<PWSDn_base> + 0000_H	8	PBG6L0#5	—
	Trigger queue status register	PWSDnSTR	<PWSDn_base> + 0004_H	8	PBG6L0#5	—
	Trigger queue status clear register	PWSDnSTC	<PWSDn_base> + 0008_H	8	PBG6L0#5	—
	Trigger queue register	PWSDnQUEj	<PWSDn_base> + 0020_H + j × 4_H	8	PBG6L0#5	—
	PWM-Diag mode A/D setting register	PWSDnPVCRx	<PWSDn_base> + 0100_H + x × 4_H	32	PBG6L0#5	—
	PWM-Diag data supplementary information register	PWSDnPWDDIRx	<PWSDn_base> + 0300_H + x × 4_H	32	PBG6L0#5	—

Table 42.10 List of Registers (2/2)

Module name	Register name	Symbol	Address	Access size	Access Protection	
					PBG	Other
PWGC_INTF	PWGC interrupt factor register	PWGCINTFhk	<PWGCINTFhk_base> + h × 600_H + k × 200_H + 0_H	32	PBG6L0#5	—
	PWGC interrupt mask register	PWGCINTMSKhk	<PWGCINTFhk_base> + h × 600_H + k × 200_H + 4_H	32	PBG6L0#5	—
	PWGC interrupt factor clear register	PWGCINTFChk	<PWGCINTFhk_base> + h × 600_H + k × 200_H + 8_H	32	PBG6L0#5	—

42.3.2 PWBAnBRSm — PWMCLKm Cycle Configuration Register

This register sets the clock cycle of PWMCLKm.

Access: This register can be read or written in 16-bit units.

Address: <PWBAn_base> + 0004_H × m

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWBAnBRSm[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.11 PWBAnBRSm Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	PWBAnBRSm [10:0]	Register for setting the clock cycle of PWMCLKm. – PWBAnBRSm = 0: PWMCLKm = PCLK – PWBAnBRSm = 1: PWMCLKm = PCLK × (2 × 1) – PWBAnBRSm = 2: PWMCLKm = PCLK × (2 × 2) ... – PWBAnBRSm = n: PWMCLKm = PCLK × (2 × n) (n = 1 to 2047) These bits can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).

42.3.3 PWBAnTE — PWMCLKm Enable Status Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWBAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTE3	PWBAnTE2	PWBAnTE1	PWBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 42.12 PWBAnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	PWBAnTE3	A status flag indicating the operation status of PWMCLK3 0: Not operating 1: Operating
2	PWBAnTE2	A status flag indicating the operation status of PWMCLK2 0: Not operating 1: Operating
1	PWBAnTE1	A status flag indicating the operation status of PWMCLK1 0: Not operating 1: Operating
0	PWBAnTE0	A status flag indicating the operation status of PWMCLK0 0: Not operating 1: Operating

42.3.4 PWBAnTS — PWMCLKm Start Trigger Register

This register is a start trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0014_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTS3	PWBAnTS2	PWBAnTS1	PWBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 42.13 PWBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTS3	Start Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Starts the output of PWMCLK0.

42.3.5 PWBAnTT — PWMCLKm Stop Trigger Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTT3	PWBAnTT2	PWBAnTT1	PWBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 42.14 PWBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTT3	Stop Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Stops the output of PWMCLK0.

42.3.6 PWGCnCTL — PWGC Control Register

PWGCnCTL is used to select the count clock from PWBA, setting output condition of PWGC_TRGOUTn, output level of forcible PWGC_TOUTn output level fix and output permission for interrupt.

Access: This register can be read or written in 16-bit units.

Address: <PWGCn_base> + 0020_H

Value after reset: 8000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnTFIE	PWGCnTRIE	—	—	—	—	—	PWGCnFOS	PWGCnOCL	—	PWGCnTCUT[1:0]		—	—	PWGCnCKS[1:0]	
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W

Table 42.15 PWGCnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	PWGCnTFIE	This bit controls interrupt of PWGC_INTn at timing PWGCnCNT value matches with PWGCn_CRBR or PWGCn_CNT value is 0 during PWGC_TOUT 100% output. 0: PWGC_INTn is disabled 1: PWGC_INTn is enabled Note 1. When PWGCnTFIE = 1, after compare register simultaneous rewrite, when 1st PWGCnCNT = PWGCnCRBR occurs before PWGCnCNT = PWGCnCSBR, PWGC_INTn is not generated. It is generated at the timing of 2nd PWGCnCNT = PWGCnCRBR after PWGCnCNT = PWGCnCSBR. Note 2. When PWGCnTFIE = 1, PWGC_INTn is generated at falling edge of PWGC_TOUTn (PWGCnCNT = PWGCnCRBR) or PWGCnCNT is match with 0 during PWM 100% output.
14	PWGCnTRIE	This bit controls interrupt of PWGC_INTn at timing PWGCnCNT value matches with PWGCn_CSBR value. 0: PWGC_INTn is disabled 1: PWGC_INTn is enabled **Note:** When forcible PWGC_TOUTn output fix level function is used (PWGCnFOT = 1) and PWGCnTRIE = 1, PWGC_INTn is generated when PWGCnCNT matches with PWGCnCSBR regardless of state of PWGC_TOUTn.
13 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	PWGCnFOS	This bit selects the output level for forcible PWGC_TOUTn output fix level. 0: PWGC_TOUTn is forcibly fixed to low level 1: PWGC_TOUTn is forcibly fixed to high level
7	PWGCnOCL	This bit selects the PWGC_TRGOUTn output condition related with PWGC_TOUTn*1. 0: PWGC_TRGOUTn output at the condition of high level of PWGC_TOUTn 1: PWGC_TRGOUTn output at the condition of both high and low level of PWGC_TOUTn Note 1. When PWGCnOCL = 0, PWGC_TRGOUTn is generated with checking PWGC_TOUTn output level before applying forcible output level fix even if PWGC_TOUTn output level is forcibly fixed.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 42.15 PWGCnCTL Register Contents (2/2)

Bit Position	Bit Name	Function
5, 4	PWGCnTCUT [1:0]	These bits select the update timing of PWGCnTCBR register*1. 00: No update 01: Update at rising edge of PWGC_TOUTn at the condition of PWGCnTCBR rewrite in progress (PWGCnRSFT = 1) 10: Update at falling edge of PWGC_TOUTn at the condition of PWGCnTCBR rewrite in progress (PWGCnRSFT = 1) 11: Update immediately (PWGCnTCBR rewrite request trigger of PWGCnRDTT = 1 setting is invalid) Note 1. When PWGCnTCUT[1:0] = 01_B or 10_B, forcible PWGC_TOUTn output level fix does not cause updating of PWGCnTCBR. The changing of output level before applying forcible output level fix is used to update PWGCnTCBR.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PWGCnCKS [1:0]	Count Clock Enable Input PWMCLK3 to PWMCLK0 Select 00: Uses PWMCLK0 as count clock 01: Uses PWMCLK1 as count clock 10: Uses PWMCLK2 as count clock 11: Uses PWMCLK3 as count clock These bits can only be rewritten when the PWGCn operation is stopped (SLPWGCk.SLPWGC[31:0] = 0).

42.3.7 PWGCnCNT — PWM Cycle Count Register

This is a count register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGCn_base> + 0014_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.16 PWGCnCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCnCNT [15:0]	16-bit counter value

42.3.8 PWGCnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGC_TOUTn output.

Access: This register can be read or written in 16-bit units.

Address: <PWGCn_base> + 0000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCSDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.17 PWGCnCSDR Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCnCSDR [15:0]	These bits set the setting condition for PWM output. The set value is reflected to the PWGCnCSBR register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1).

NOTE

The setting of PWGCnCSDR > PWGC_PERIOD is prohibited.

42.3.9 PWGCnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGC_TOUTn output.

Access: This register can be read or written in 32-bit units.

Address: <PWGCn_base> + 0004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWGCnCRDR [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCRDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.18 PWGCnCRDR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16 to 0	PWGCnCRDR [16:0]	These bits set the reset condition for PWM output. The set value is reflected to the PWGCnCRBR register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1).

42.3.10 PWGCnCTDR — PWGC_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGC_TRGOUTn.

Access: This register can be read or written in 16-bit units.

Address: <PWGCn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCTDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.19 PWGCnCTDR Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCnCTDR [15:0]	These bits set the A/D conversion trigger generation condition for PWSDn. The set value is reflected to the PWGCnCTBR register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1).

42.3.11 PWGCnCSBR — PWGCnCSDR Buffer Register

This is a buffer register for the PWGCnCSDR register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGCn_base> + 0024_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCSBR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.20 PWGCnCSBR Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCnCSBR [15:0]	The PWGCnCSDR register value is reflected to this register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1). When the value matches the PWGCnCNT register value and does not match the PWGCnCRBR, the pin output is driven high. When PWGC_TOUTn is forcibly fixed, forcible PWGC_TOUTn output level fix is continued until PWGCnCNT = PWGCnCSBR.

42.3.12 PWGCnCRBR — PWGCnCRDR Buffer Register

This is a buffer register for the PWGC_TOUTn reset condition.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PWGCn_base> + 0028_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWGCnCRBR[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCRBR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.21 PWGCnCRBR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16 to 0	PWGCnCRBR [16:0]	The PWGCnCRDR register value is reflected to this register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1). When the value matches the PWGCnCNT register value, the pin output is driven low except for during forcible PWGC_TOUTn output level fix and case that PWGCnCNT = PWGCnCRBR occurs by PWGCnCNT = PWGCnCSBR after compare buffer register simultaneous rewrite.

42.3.13 PWGCnCTBR — PWGCnCTDR Buffer Register

This is a buffer register for the PWGC_TRGOUTn generation condition.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGCn_base> + $002C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCnCTBR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.22 PWGCnCTBR Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCnCTBR [15:0]	The PWGCnCTDR register value is reflected to this register at the start of PWGCn operation (SLPWGCk.SLPWGC of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGCnRDT.PWGCnRDT = 1). When the value matches the PWGCnCNT register value, a trigger is transmitted to PWSDn.

42.3.14 PWGCnST — PWGC Status Register

This register is a status register for compare buffer register rewrite control and forcible PWGC_TOUTn output level fix.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGCn_base> + 0010_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGCnFOF	—	—	—	—	—	—	PWGCnRSFT	PWGCnRSF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.23 PWGCnST Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8	PWGCnFOF	This bit indicates forcible PWGC_TOUTn output level fix status. 0: PWGC_TOUTn output PWM waveforms. 1: PWGC_TOUTn output level is forcibly fixed.
7 to 2	Reserved	When read, the value after reset is returned.
1	PWGCnRSFT	PWGCnTCBR Rewrite Control Status
0	PWGCnRSF	Simultaneous Rewrite Control Status 0: Simultaneous rewrite is enabled. This value indicates the completion of simultaneous rewrite after the generation of a simultaneous rewrite trigger signal. 1: Simultaneous rewrite is in progress. This value indicates the waiting state for completion.

42.3.15 PWGCnTCR — PWGC_TRGOUTn Control Register

This register controls enable/disable of PWGC_TRGOUTn.

Access: This register can be read or written in 8-bit units.

Address: <PWGCn_base> + 0018_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGCnTOE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Table 42.24 PWGCnTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWGCnTOE	This bit sets the output enable/disable of PWGC_TRGOUTn. 0: Output disable of PWGC_TRGOUTn 1: Output enable of PWGC_TRGOUTn This register value is reflected to the PWGCnTCBR at the update timing selected by PWGCnTCUT[1:0].

NOTE

PWGCnTOE bit has to be set before count operation. In addition, in case of PWGCnTCUT = 01_B or 10_B, PWGCnTOE bit has to be set before PWGCnTCBR rewrite request.

42.3.16 PWGCnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWGCn_base> + $000C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWGCnRDTT	PWGCnRDT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 42.25 PWGCnRDT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWGCnRDTT	PWGCnTCBR Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the rewrite request for PWGCnTCBR, and sets PWGCnST.PWGCnRSFT to 1. In case of PWGCnTCUT = 00_B or 11_B, write to this bit “1” is invalid.
0	PWGCnRDT	Compare Buffer Register Simultaneous Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the simultaneous rewrite request for the compare registers (PWGCnCSDR, PWGCnCRDR, and PWGCnCTDR), and sets PWGCnST.PWGCnRSF to 1.

42.3.17 PWGCnTCBR — PWGCnTCR Buffer Register

This is a buffer register for the PWGCnTCR register.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWGCn_base> + $001C_H$

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGCnTOBE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 42.26 PWGCnTCBR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	PWGCnTOBE	The PWGCnTCR register value is reflected to this register at the update timing selected by PWGCnCTL.PWGCnTCUT[1:0]. 0: Output disable of PWGC_TRGOUTn. 1: Output enable of PWGC_TRGOUTn.

42.3.18 PWGCnFOT — Forcible PWGC_TOUTn Output level Fix Trigger Register

This register is a trigger register to set PWGC_TOUTn output level forcibly.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWGCn_base> + $003C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGCnFOT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 42.27 PWGCnFOT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	PWGCnFOT	Trigger bit to set PWGC_TOUTn output level forcibly. 0: Writing 0 has no effect. 1: Set PWGC_TOUTn to level selected by PWGCnFOS bit forcibly.

Note: During counting state, setting this bit is available.

42.3.19 SLPWGCk — PWGC Synchronous Trigger Register (k = 0 to 2)

This register triggers start and stop for multiple channels simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <SLPW_base> + k × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SLPWGC[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLPWGC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.28 SLPWGCk Register Contents

Bit Position	Bit Name	Function
31 to 0	SLPWGC [31:0]	Trigger start and stop to multiple channels simultaneously. 0: Stops the corresponding channels. 1: Starts the corresponding channels. The bits correspond to the following channels. SLPWGC0.SLPWGC[31:0]: PWGC31 - PWGC0 SLPWGC1.SLPWGC[31:0]: PWGC63 - PWGC32 SLPWGC2.SLPWGC[31:0]: PWGC95 - PWGC64

42.3.20 PWSDnCTL — PWSD Control Register

This register is used to control operations of PWSD.

Access: This register can be read or written in 8-bit units.

Address: <PWSDn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PWSDnARSE	—	—	—	—	—	—	PWSDnENBL
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W

Table 42.29 PWSDnCTL Register Contents

Bit Position	Bit Name	Function
7	PWSDnARSE	A/D conversion result storing control 0: Storing the ADCJvPWDDIR register to PWSDnPWDDIRx register is disabled. 1: Storing the ADCJvPWDDIR register to PWSDnPWDDIRx register is enabled.
6 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSDnENBL	Operation Permission Control 0: Operation is prohibited (initial state). Writing 0 initializes PWSDnSTR and PWSDnQUEj, PWSDnOWE and PWSDnWFLG. 1: Operation is enabled.

NOTE

PWSDnARSE setting has to be changed during the state of trigger has not been input (PWSDnQUE0 = $7F_H$) and condition of not input the trigger from PWGC.

42.3.21 PWSDnSTR — Trigger Queue Status Register

This is a status register that indicates whether the number of a channel for which an A/D conversion trigger has been generated is stored in a PWSDnQUEj register.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSDn_base> + 0004_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSDnQFL	PWSDnQNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 42.30 PWSDnSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWSDnQFL	Indicates the queuing state of the A/D conversion trigger. 0: Some PWSDnQUEj registers do not store a channel number. 1: All of the PWSDnQUEj registers store a channel number.
0	PWSDnQNE	Bit indicating that there is a trigger in the trigger queue 0: A channel number is not stored in a PWSDnQUEj register, or A/D conversion is in progress while only PWSDnQUE0 stores a channel number. 1: The number of the channel waiting for conversion is stored in j = 1 and subsequent PWSDnQUEj registers.

42.3.22 PWSDnSTC — Trigger Queue Status Clear Register

This register clears the status of the PWSDnSTR register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWSDn_base> + 0008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSDnCLFL	PWSDnCLNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 42.31 PWSDnSTC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWSDnCLFL	PWSDnQFL Clear 0: PWSDnQFL retains the status (Writing 0 has no effect). 1: PWSDnQFL is cleared to 0.
0	PWSDnCLNE	PWSDnQNE Clear 0: PWSDnQNE retains the status (Writing 0 has no effect). 1: PWSDnQNE is cleared to 0.

42.3.23 PWSDnQUEj (j = 0 to 7) — Trigger Queue Register

This register stores the channel number that received the trigger from PWGCn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSDn_base> + $0020_H + j \times 4_H$

Value after reset: $7F_H$

Bit	7	6	5	4	3	2	1	0
	—	PWSDnQUEj[6:0]						
Value after reset	0	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R

Table 42.32 PWSDnQUEj Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	PWSDnQUEj [6:0]	These bits hold the channel number (0 to 95) of the PWGC for which a trigger was generated in order from PWSDnQUE0 to PWSDnQUE7. After the A/D conversion of PWSDnQUE0 is completed, the values in PWSDnQUE1 to PWSDnQUE7 shift to PWSDnQUE0 to PWSDnQUE6.

NOTE

If a trigger occurs simultaneously for multiple channels, the trigger with the smaller channel number has priority.

42.3.24 PWSDnPVCRx — PWM-Diag Mode A/D Setting Register

This register is used to set the corresponding A/D converter for each channel.

At the generation of a trigger, the set value is transmitted to the ADCJvPWDVCR register of the A/D converter.

For the ADCJvPWDVCR register, see **Section 43, Analog to Digital Converter (ADCJ)**.

Access: This register can be read or written in 32-bit units.

Address: <PWSDn_base> + 0100_H + x × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PWSDnWTTSx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWSDnMPXEx	PWSDnMPXVx[2:0]			PWSDnVCULLMTBSx[3:0]				PWSDnSLADx[1:0]		PWSDnGCTRLx[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.33 PWSDnPVCRx Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	PWSDnWTTSx [3:0]	These bits indicate the setting value of ADCJvPWDVCR.WTTS[3:0] that selects wait time table in ADCTL.
15	PWSDnMPXEx	This bit indicates the setting value of ADCJvPWDVCR.MPXE that selects the use of MPX in ADCTL.
14 to 12	PWSDnMPXVx [2:0]	These bits indicate the setting value of ADCJvPWDVCR.MPXV[2:0] that selects MPX channel in ADCTL.
11 to 8	PWSDnVCULL MTBSx[3:0]	These bits indicate the setting value of ADCJvPWDVCR.VCULLMTBS[3:0] that selects upper limit/lower limit table.
7, 6	PWSDnSLADx [1:0]	ADCJ Select: 00: Output to ADCJ0 01: Output to ADCJ1 10: Output to ADCJ2 11: Setting prohibited
5 to 0	PWSDnGCTRLx [5:0]	These bits indicate the setting value of the ADCAnPWDVCR.GCTRL[5:0] that selects physical channel in ADCTL.

42.3.25 PWSDnPWDDIRx — PWM-Diag Data Supplementary Information Register

This register stores the A/D conversion result correspond to PWSDnPVCRx register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PWSDn_base> + 0300_H + x × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWSDnMPXE	PWSDnMPXV[2:0]			—	PWSDnOWE	PWSDnWFLG	—	—	—	PWSDnID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWSDnPWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.34 PWSDnPWDDIRx Register Contents

Bit Position	Bit Name	Function
31	PWSDnMPXE	This bit mirrors PWSDnPVCRx.PWSDnMPXEx.
30 to 28	PWSDnMPXV [2:0]	These bits mirror PWSDnPVCRx.PWSDnMPXVx[2:0]
27	Reserved	When read, the value after reset is returned.
26	PWSDnOWE	This bit indicates overwrite error flag of the A/D conversion result storing. Set timing is same as PWSDnPWDDR stored timing at the condition of PWSDnWFLG = 1. 0: An overwrite error is not detected. 1: An overwrite error is detected. Setting condition: PWSDnWFLG = 1, and new A/D conversion result is written to PWSDnPWDDR[15:0]. Clearing condition: Reading this register, or clearing the PWSDnENBL bit.
25	PWSDnWFLG	This bit indicates write flag of the A/D conversion result storing. Set timing is same as PWSDnPWDDR stored timing. 0: This register is read, or A/D conversion is not finished. 1: The A/D conversion result is stored (not read yet). Setting condition: The A/D conversion result is written to PWSDnPWDDR[15:0]. Clearing condition: Reading this register, or clearing the PWSDnENBL bit.
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	PWSDnID[5:0]	These bits mirror PWSDnPVCRx.PWSDnGCTRLx[5:0]
15 to 0	PWSDnPWDDR [15:0]	These bits indicate the A/D conversion result which is sent by ADC_PWDDRv[15:0] pins from ADCJv. The data format is same as ADCJvPWDTSNDR.PWDDR[15:0].

42.3.26 PWGCPRDm — PWGC Period Setting Register (m = 0 to 3)

This register is used to set the PWGC counter period.

Access: This register can be read or written in 16-bit units.

Address: <SLPW_base> + $000C_H$ + m × 4_H

Value after reset: m = 0: $0FFF_H$
m = 1 to 3: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCPRD[15:0]															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.35 PWGCPRDm Register Contents

Bit Position	Bit Name	Function
15 to 0	PWGCPRD [15:0]	These bits set the PWGC counter period (= PWGC_PERIOD). The period of PWGC_TOUTn can be flexible changed by setting this register and PWGCPRDSLq register. Note 1. – m = 0: PWGCPRD initial value = $0FFF_H$ – m = 1 to 3: PWGCPRD initial value = $FFFF_H$

NOTE

PWGCPRDm register setting change is prohibited during count operation.

42.3.27 PWGCPRDSLq — PWGC Period Selection Register (q = 0 to 5)

This register is used to select the PWGC counter period from PWGCPRDm register setting.

Access: This register can be read or written in 32-bit units.

Address: <SLPW_base> + $001C_H$ + q × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGCPRDSLq[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCPRDSLq[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.36 PWGCPRDSLq Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGCPRDSLq [31:0]	Select the PWGC counter period by PWSDPRDm register setting. PWGCPRDSL0[1:0] select counter period for PWGC0 PWGCPRDSL0[3:2] select counter period for PWGC1 PWGCPRDSL0[5:4] select counter period for PWGC2 - - - - - PWGCPRDSL5[31:30] select counter period for PWGC95 Below is example for PWGCPRDSL0[1:0] select counter period for PWGC0: **PWGCPRDSL0[1:0] / Description** 00 / PWGCPRD0 (default) 01 / PWGCPRD1 10 / PWGCPRD2 11 / PWGCPRD3 The bits correspond to the following channels. PWGCPRDSL0: PWGC15 to PWGC0 PWGCPRDSL1: PWGC31 to PWGC16 PWGCPRDSL2: PWGC47 to PWGC32 PWGCPRDSL3: PWGC63 to PWGC48 PWGCPRDSL4: PWGC79 to PWGC64 PWGCPRDSL5: PWGC95 to PWGC80

42.3.28 PWGCINTFhk — PWGC Interrupt Factor Register (k = 0 to 2, h = 0 to 3)

These registers contain information about which PWGCn interrupt (PWGC_INTn) has been generated without depending on PWGCINTMSKhk setting. Regarding block diagram image, see **Figure 42.1, PWGC Interrupt Connection Image**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PWGCINTF_base> + h × 600_H + k × 200_H + 0_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGCINTFhk[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCINTFhk[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 42.37 PWGCINTFhk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGCINTFhk [31:0]	PWGC_INTn interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred The bits correspond to the following channels. PWGCINTFh0[31:0]: PWGC31 to PWGC0 PWGCINTFh1[31:0]: PWGC63 to PWGC32 PWGCINTFh2[31:0]: PWGC95 to PWGC64

42.3.29 PWGCINTMSKhk — PWGC Interrupt Mask Register (k = 0 to 2, h = 0 to 3)

These registers mask PWGCn interrupt output to INTC2 by each channels. Regarding block diagram image, see **Figure 42.1, PWGC Interrupt Connection Image**.

Access: This register can be read or written in 32-bit units.

Address: <PWGCINTF_base> + h × 600_H + k × 200_H + 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGCINTMSKhk[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCINTMSKhk[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 42.38 PWGCINTMSKhk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGCINTMSKhk [31:0]	PWGC interrupt output to INTC2 mask 0: Not masked 1: Masked The bits correspond to the following channels. PWGCINTMSKh0[31:0]: PWGC31 - PWGC0 PWGCINTMSKh1[31:0]: PWGC63 - PWGC32 PWGCINTMSKh2[31:0]: PWGC95 - PWGC64

42.3.30 PWGCINTFChk — PWGC Interrupt Factor Clear Register (k = 0 to 2, h = 0 to 3)

These registers clear the bits of PWGC interrupt factor register (PWGCINTFhk). Regarding block diagram image, see **Figure 42.1, PWGC Interrupt Connection Image**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <PWGCINTF_base> + h × 600_H + k × 200_H + 8_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWGCINTFChk[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWGCINTFChk[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

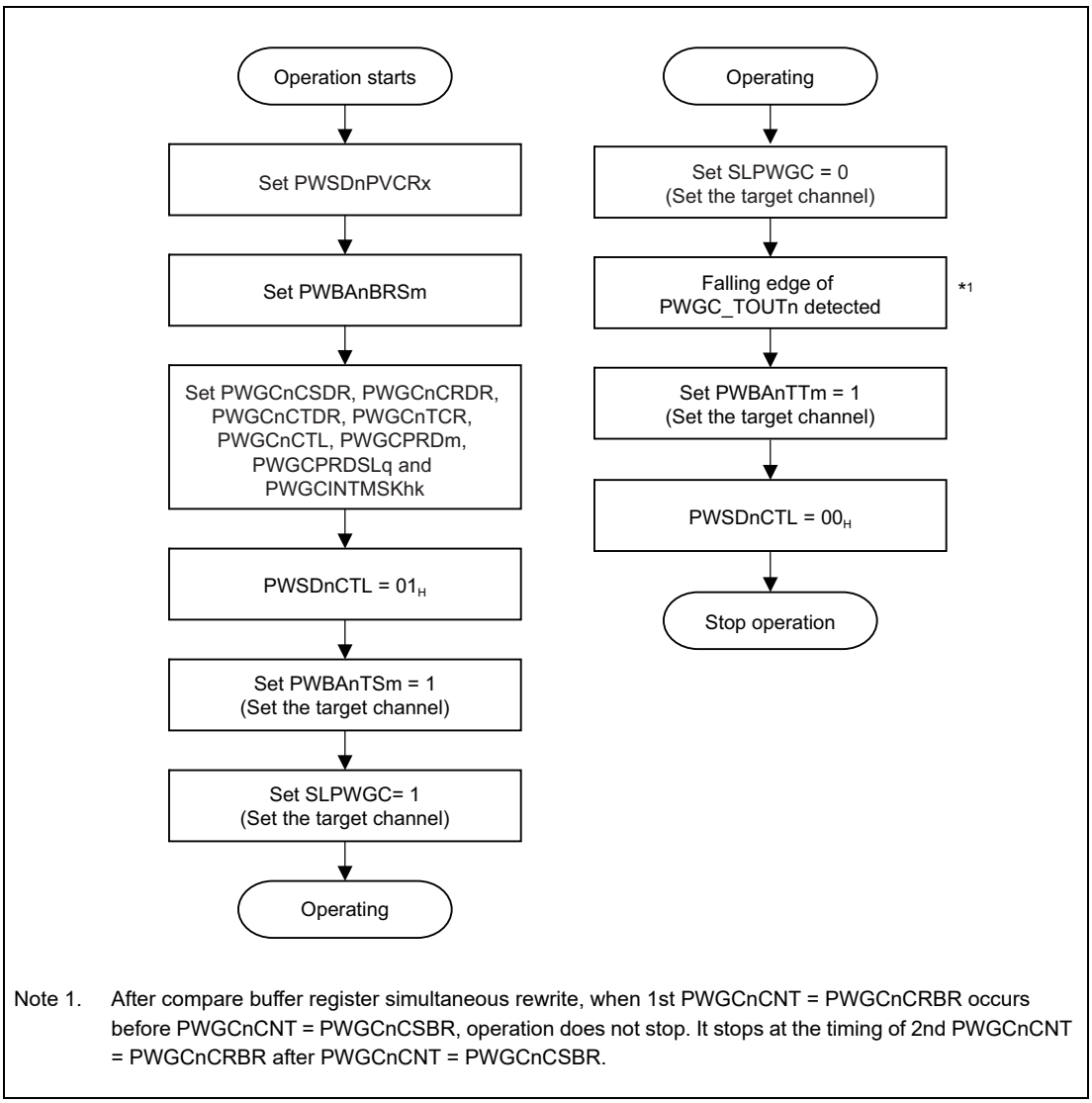
Table 42.39 PWGCINTFChk Register Contents

Bit Position	Bit Name	Function
31 to 0	PWGCINTFChk [31:0]	PWGCINTFhk[31:0] flag clear 0: — 1: Clear The bits correspond to the following channels. PWGCINTFCh0[31:0]: PWGC31 to PWGC0 PWGCINTFCh1[31:0]: PWGC63 to PWGC32 PWGCINTFCh2[31:0]: PWGC95 to PWGC64

42.4 Operation

42.4.1 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.



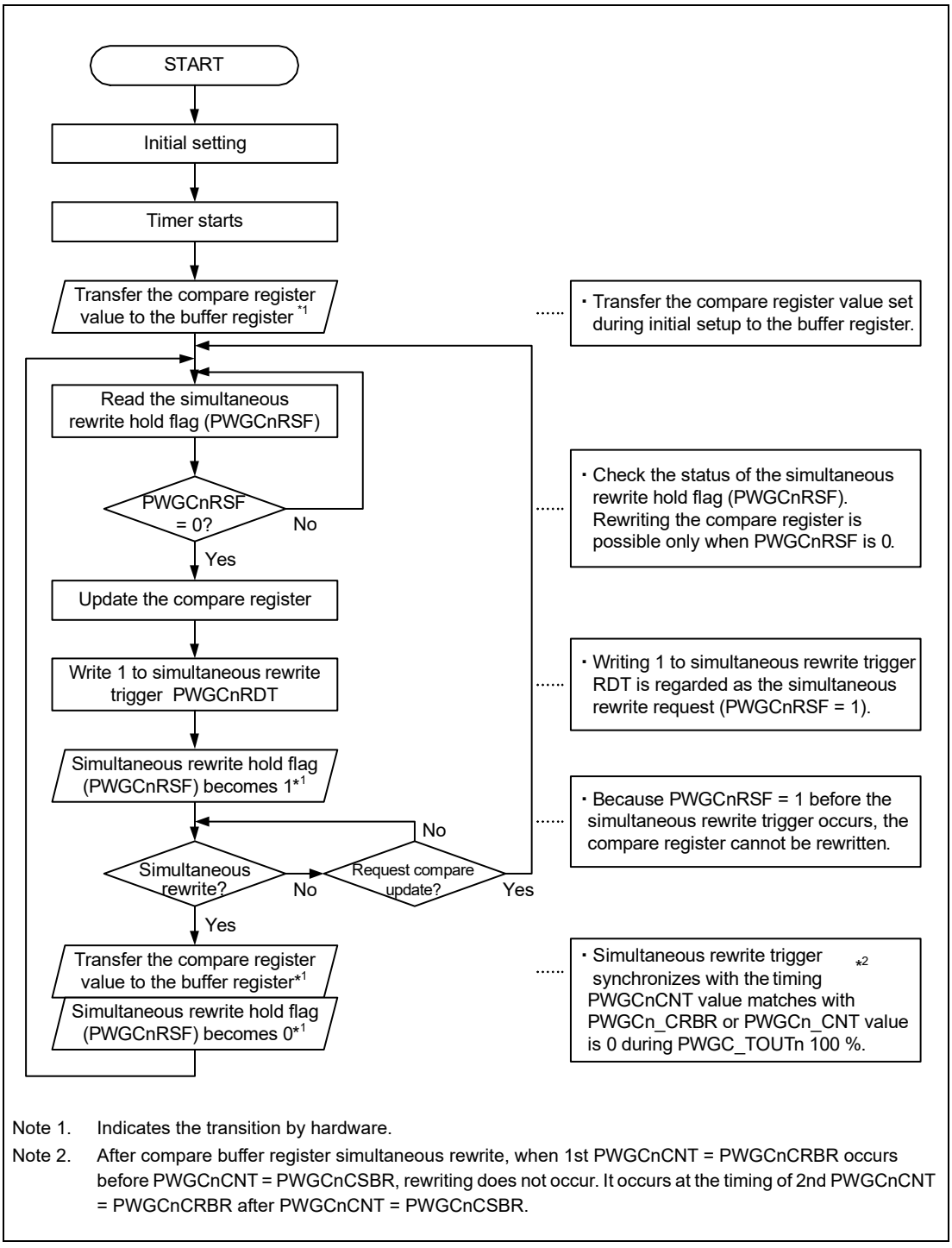
Note 1. After compare buffer register simultaneous rewrite, when 1st PWGCnCNT = PWGCnCRBR occurs before PWGCnCNT = PWGCnCSBR, operation does not stop. It stops at the timing of 2nd PWGCnCNT = PWGCnCRBR after PWGCnCNT = PWGCnCSBR.

Figure 42.3 PWM-Diag Operating Procedure

Procedures for simultaneous rewrite of PWGC are illustrated below.

The described term “compare register” indicates PWGCnCSDR, PWGCnCRDR, or PWGCnCTDR.

In addition, the described term “buffer register” indicates PWGCnCSBR, PWGCnCRBR, or PWGCnCTBR.



Note 1. Indicates the transition by hardware.

Note 2. After compare buffer register simultaneous rewrite, when 1st PWGCnCNT = PWGCnCRBR occurs before PWGCnCNT = PWGCnCSBR, rewriting does not occur. It occurs at the timing of 2nd PWGCnCNT = PWGCnCRBR after PWGCnCNT = PWGCnCSBR.

Figure 42.4 Simultaneous Rewrite Procedure

42.4.2 Operation Waveform of PWM-Diag

42.4.2.1 PWM Waveform Output by PWGC and Operation Waveform for A/D Conversion Trigger Output

(1) Basic Operation Waveform of PWGC

The basic operation waveforms of PWGC are illustrated below.

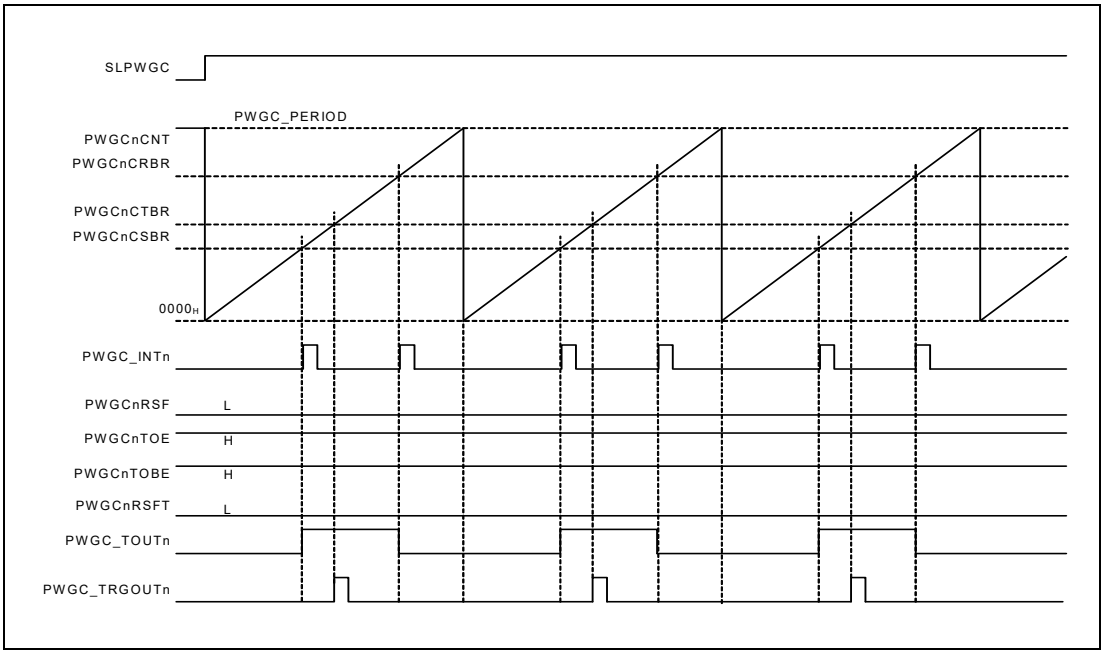


Figure 42.5 Basic Waveform (when PWGCnTFIE = 1 and PWGCnTRIE = 1)

(2) Operation Waveform when Simultaneous Rewrite for PWGC is Executed

The following figure illustrates the operation waveforms when simultaneous rewrite for PWGC is executed.

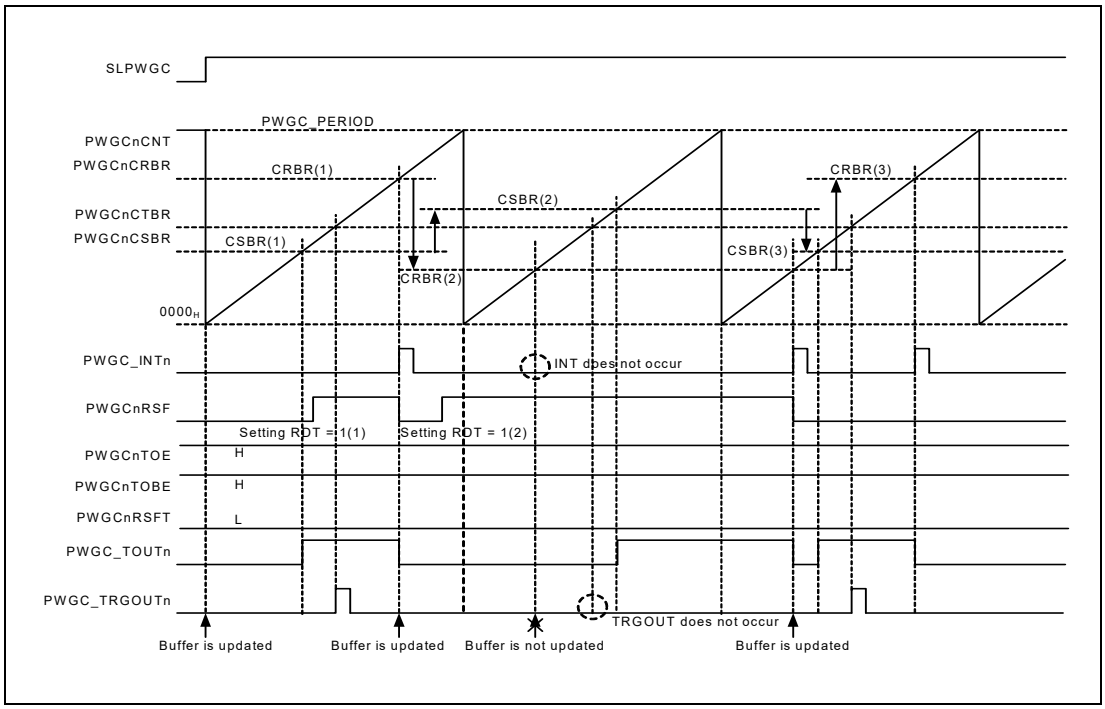


Figure 42.6 Waveform when compare buffer register simultaneous rewrite is executed (PWGCnOCL = 0, PWGCnTFIE = 1 and PWGCnTRIE = 0)

Compare buffer register simultaneous rewrite is executed by re-setting the PWGCnCSDR and PWGCnCRDR registers, then setting either the PWGCnRDT or SLPWGCk register.

Moreover, if the relationship between set values in one interval is PWGCnCSDR > PWGCnCRDR, a falling edge in that interval is meaningless, and the falling edge in the next interval is valid.

In case of PWGCnOCL = 1, PWGCn_TRGOUTn is also occurred at the timing of “TRGOUT does not occur” in **Figure 42.6, Waveform when compare buffer register simultaneous rewrite is executed (PWGCnOCL = 0, PWGCnTFIE = 1 and PWGCnTRIE = 0)**.

(3) Operation Waveform when Stopping and Resuming PWGC Operation

The following figure illustrates the operation waveforms when stopping and resuming PWGC operation.

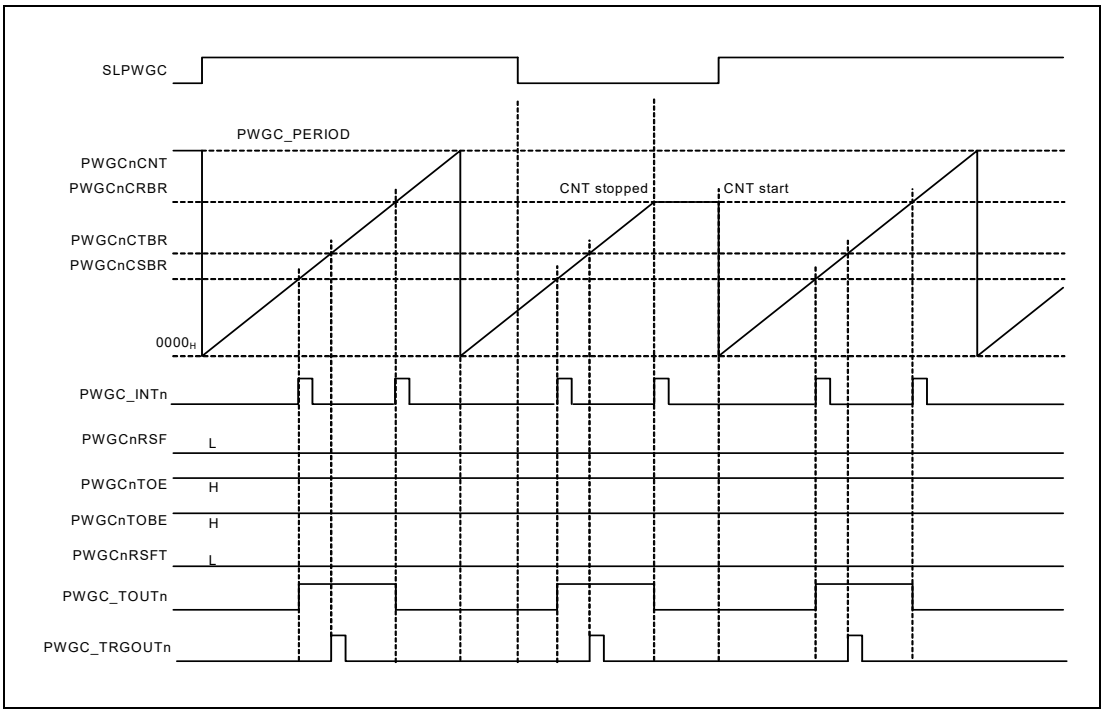


Figure 42.7 Stopping and Resuming Operation when PWGCnTFIE = 1 and PWGCnTRIE = 1 (1)

After the setting of SLPWGC has been changed from 1 to 0, PWGCnCNT stops operation because PWGC_INTn is generated.

After PWGC_INTn has been generated, by changing the setting of SLPWGC from 0 to 1, PWGCnCNT resumes counting from 0000_H.

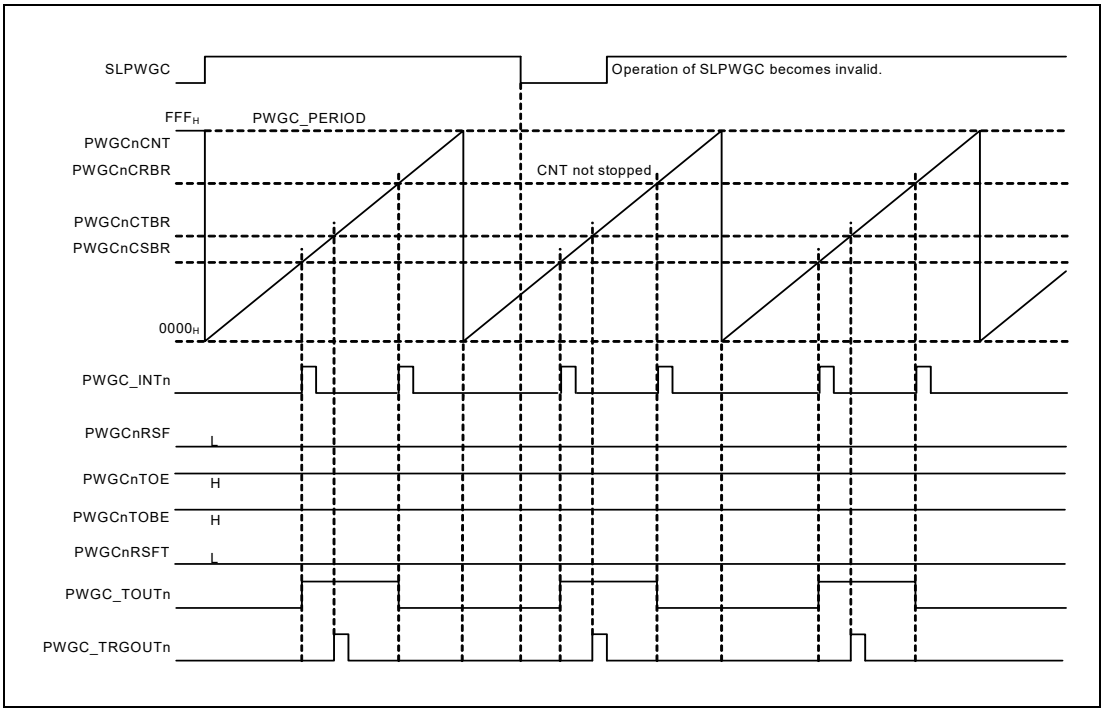


Figure 42.8 Stopping and Resuming Operation when PWGCnTFIE = 1 and PWGCnTRIE = 1 (2)

After the setting of SLPWGC has been changed from 1 to 0, if the setting of SLPWGC is changed from 0 to 1 before PWGC_INTn is generated, operations of SLPWGC become invalid, and PWGCnCNT continues counting.

(4) Waveforms of PWGC Operation with Specific Settings

The following figures illustrate the waveforms of PWGC operation with specific settings.

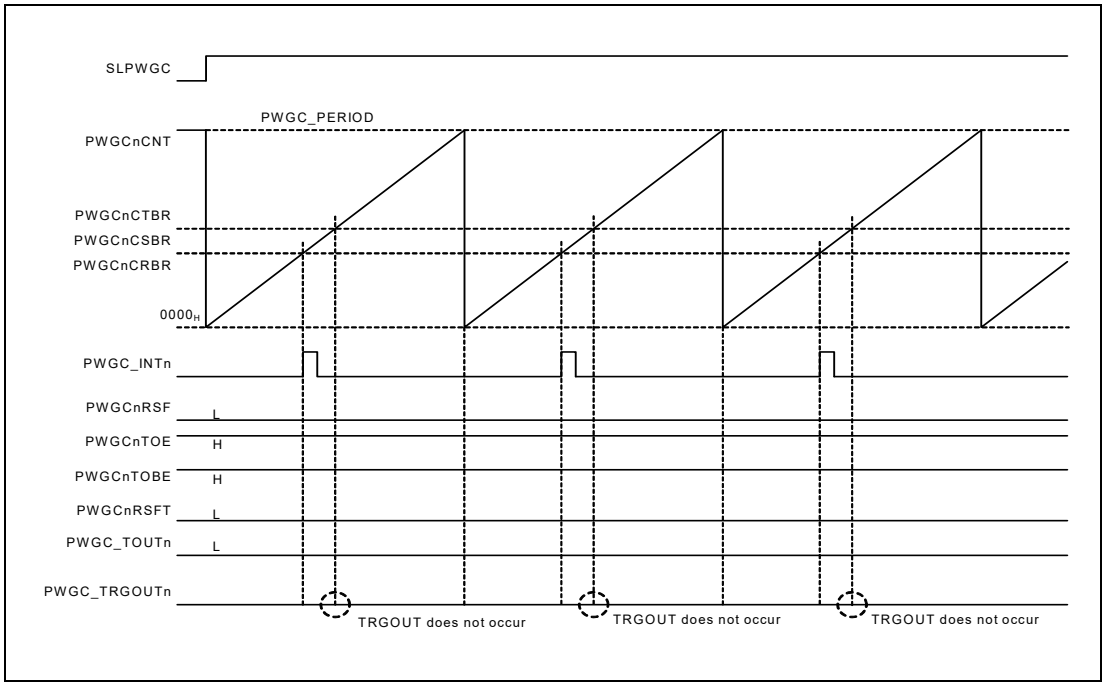


Figure 42.9 PWGC_TOUTn = 0% Output Waveform (PWGCnTFIE = 1 and PWGCnTRIE = 1)

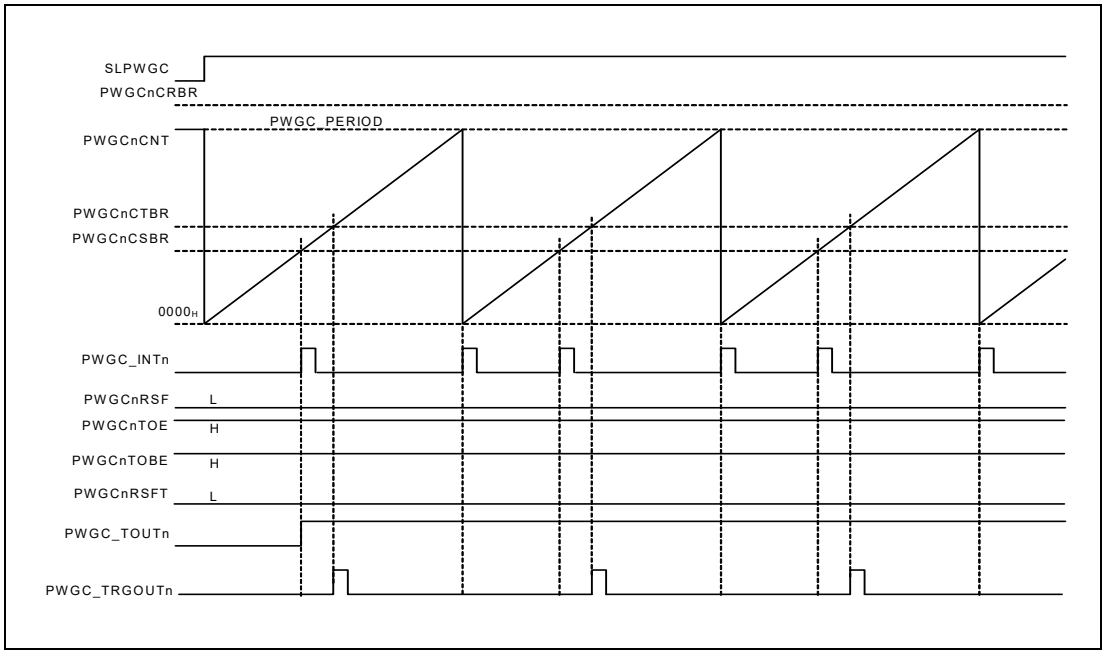


Figure 42.10 PWGC_TOUTn = 100% Output Waveform (PWGCnTFIE = 1 and PWGCnTRIE = 1)

42.4.2.2 Operating wave form when forcible PWGC_TOUTn output level fix

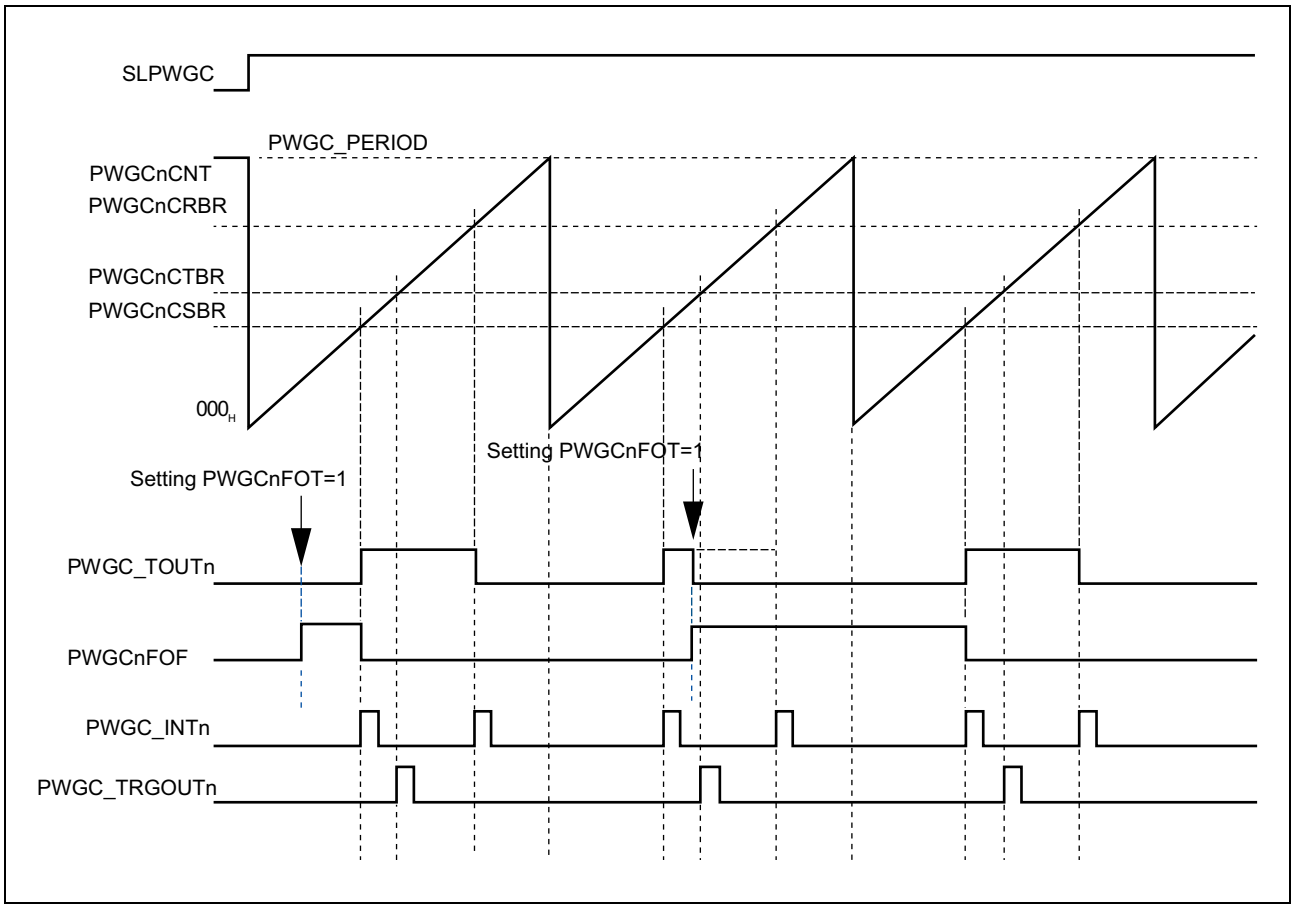


Figure 42.11 PWGC_TOUTn Output Fixed to Low Level Waveform

When PWGCnFOT is set to 1 in the state of PWGCnFOS=0, PWGC_TOUTn is forcibly set to low level immediately until the next PWGCnCSBR matching.

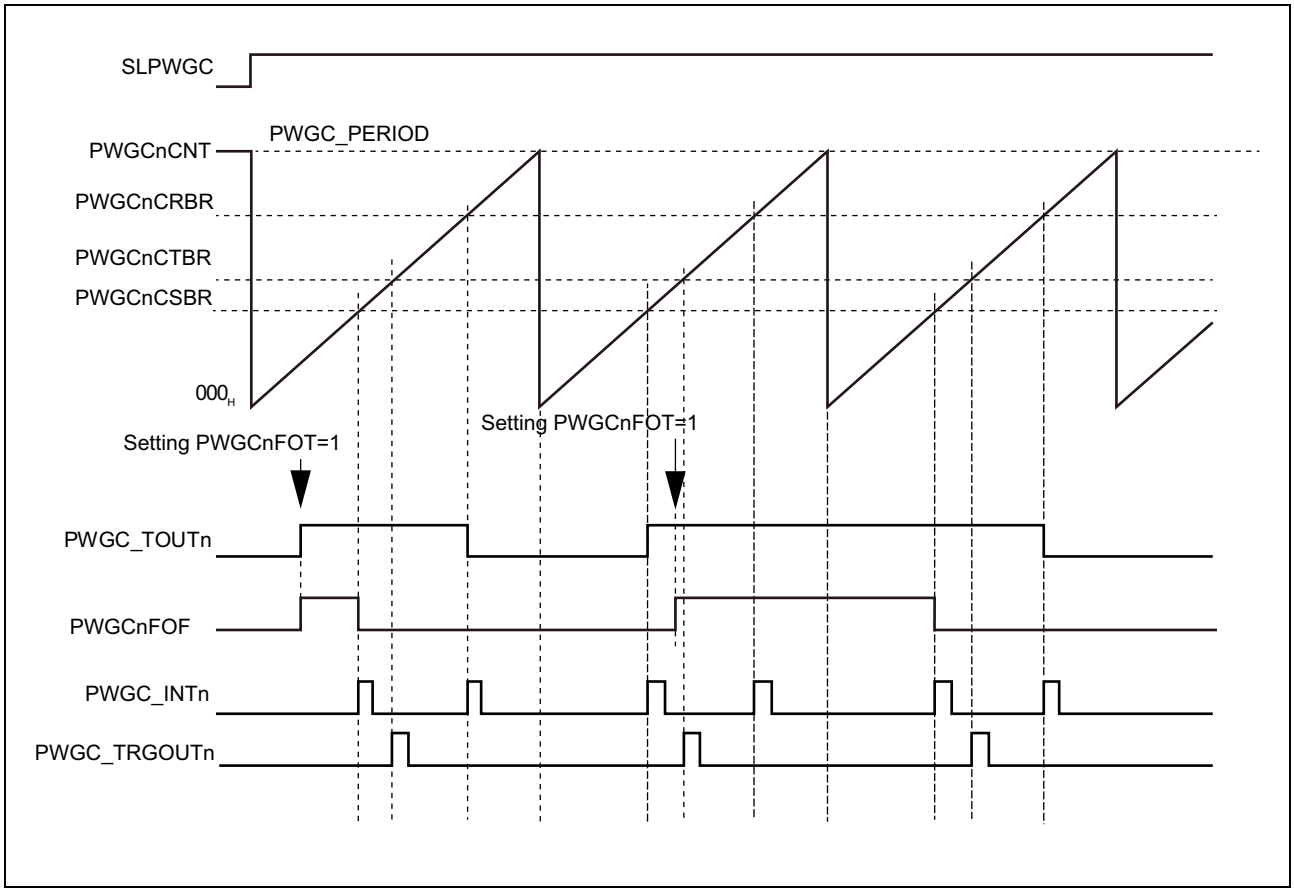


Figure 42.12 PWGC_TOUTn Output Fixed to High Level Waveform

When PWGCnFOT is set to 1 in the state of PWGCnFOS=1, PWGC_TOUTn is forcibly set to high level immediately until the next PWGCnCSBR matching.

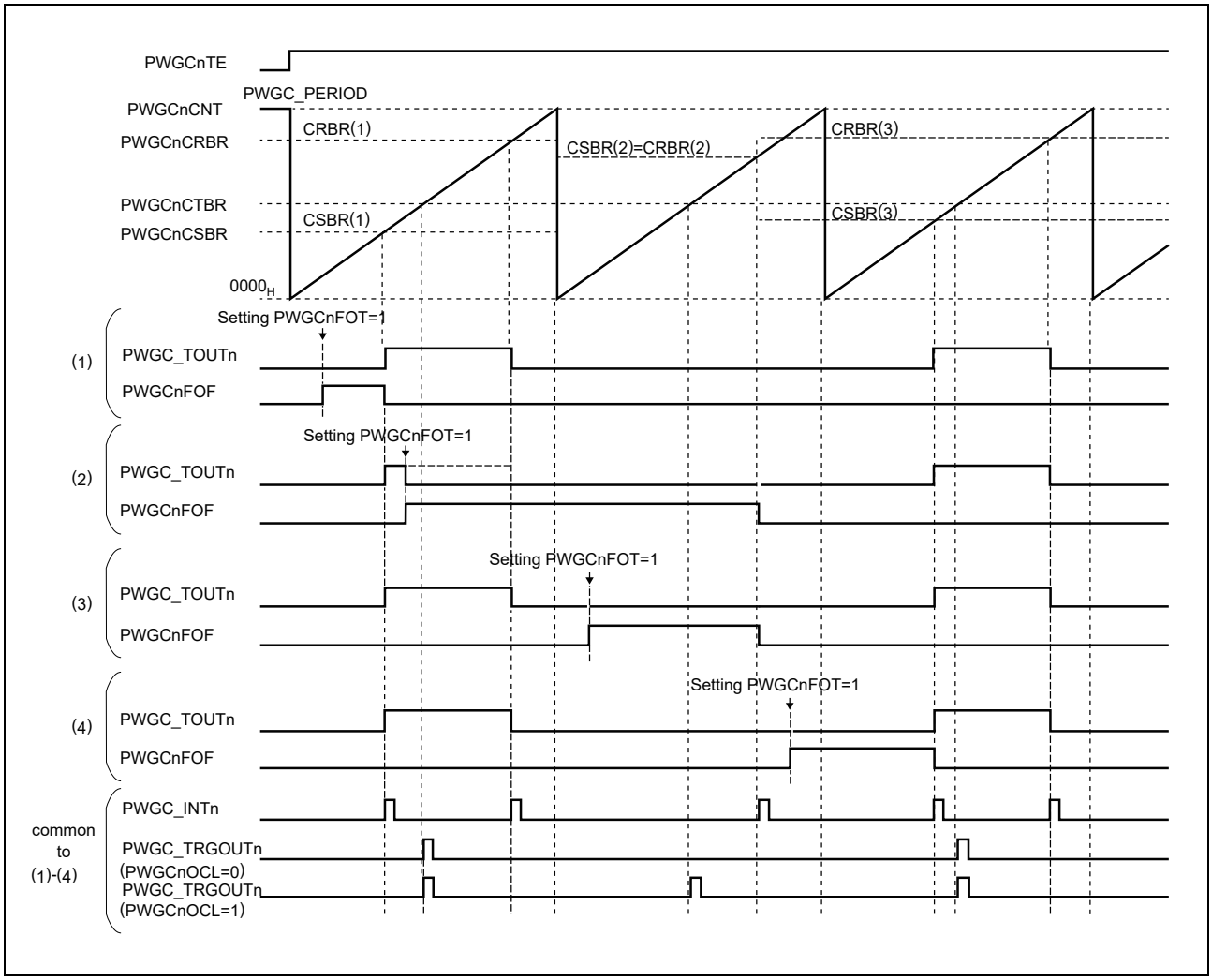


Figure 42.13 PWGC_TOUTn Output Fixed to Low Level for PWM 0% Output Waveform

(1) shows the case of forcible low level fix during the low level period of PWM normal output before PWM 0% output.

(2) shows the case of forcible low level fix during the high level period of PWM normal output before PWM 0% output.

(3) shows the case of forcible low level fix during PWM 0% output.

(4) shows the case of forcible low level fix during the low level period of PWM normal output after PWM 0% output.

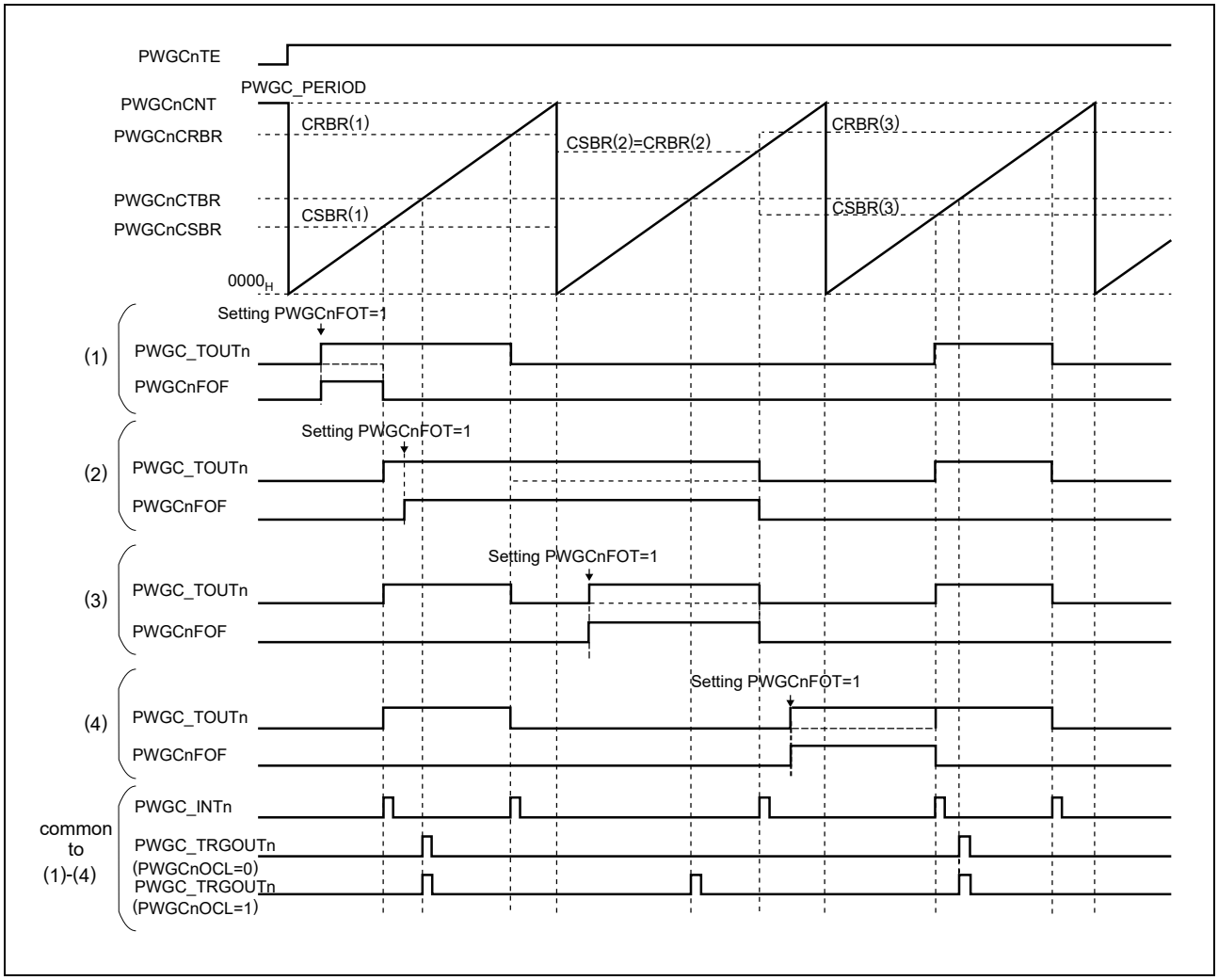


Figure 42.14 PWGC_TOUTn Output Fixed to High Level for PWM 0% Output Waveform

(1) shows the case of forcible high level fix during the low level period of PWM normal output before PWM 0% output.

(2) shows the case of forcible high level fix during the high level period of PWM normal output before PWM 0% output.

(3) shows the case of forcible high level fix during PWM 0% output.

(4) shows the case of forcible high level fix during the low level period of PWM normal output after PWM 0% output.

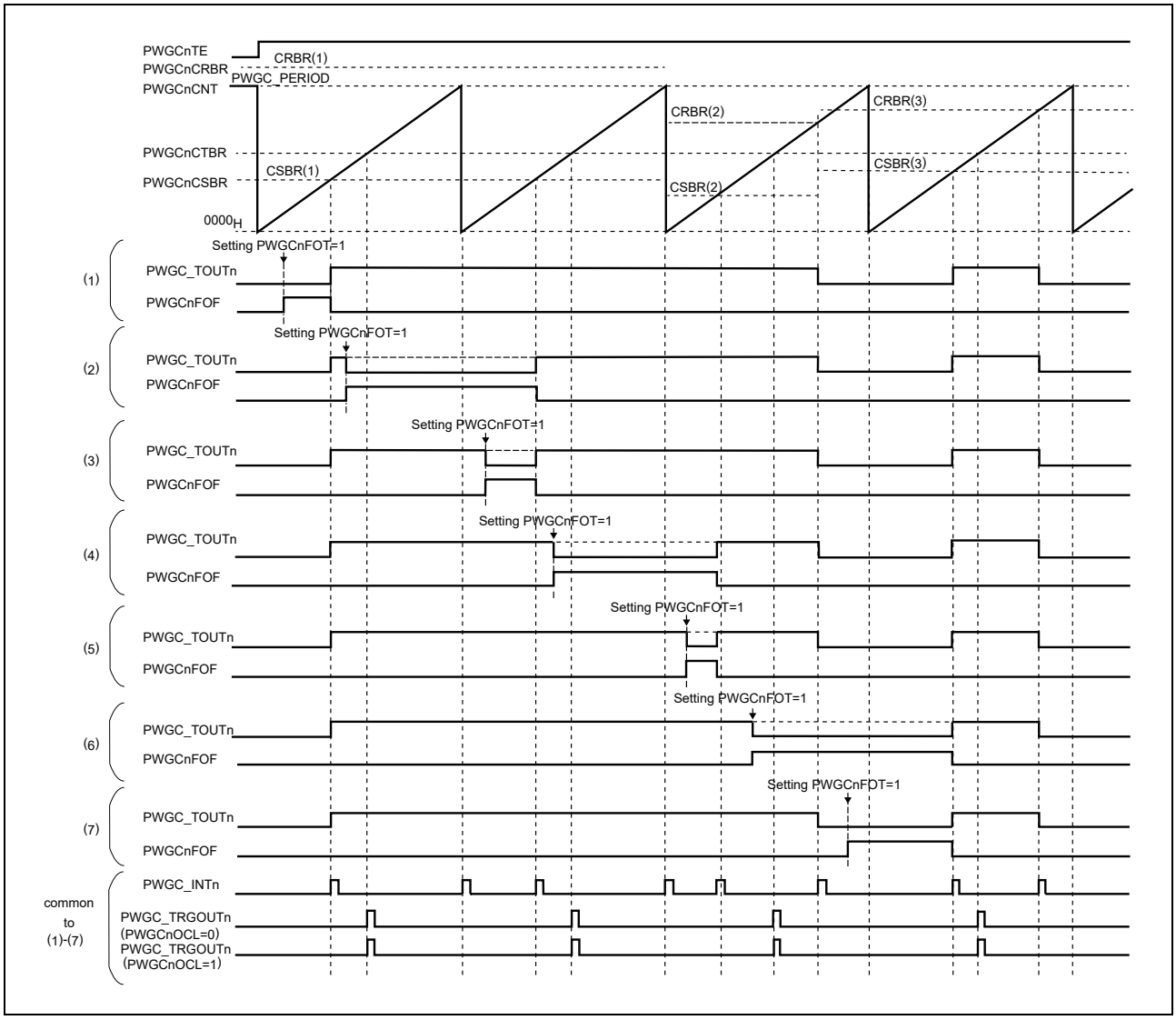


Figure 42.15 PWGC_TOUTn Output Fixed to Low Level for PWM 100% Output Waveform

(1) shows the case of forcible low level fix during the low level period of transmission to PWM 100% output.

(2) shows the case of forcible low level fix during the high level period of transmission to PWM 100% output.

(3) shows the case of forcible low level fix during PWGCnCNT<=PWGCnCSBR period of PWM 100% output.

(4) shows the case of forcible low level fix during PWGCnCSBR<PWGCnCNT<=PWGC_PERIOD period of PWM 100% output.

(5) shows the case of forcible low level fix during PWGCnCNT<=PWGCnCSBR period of transmission from PWM 100% output to PWM normal output.

(6) shows the case of forcible low level fix during PWGCnCSBR<PWGCnCNT<=PWGCnCRBR period of transmission from PWM 100% output to PWM normal output.

(7) shows the case of forcible low level fix during the low level period of PWM normal output after PWM 100% output.

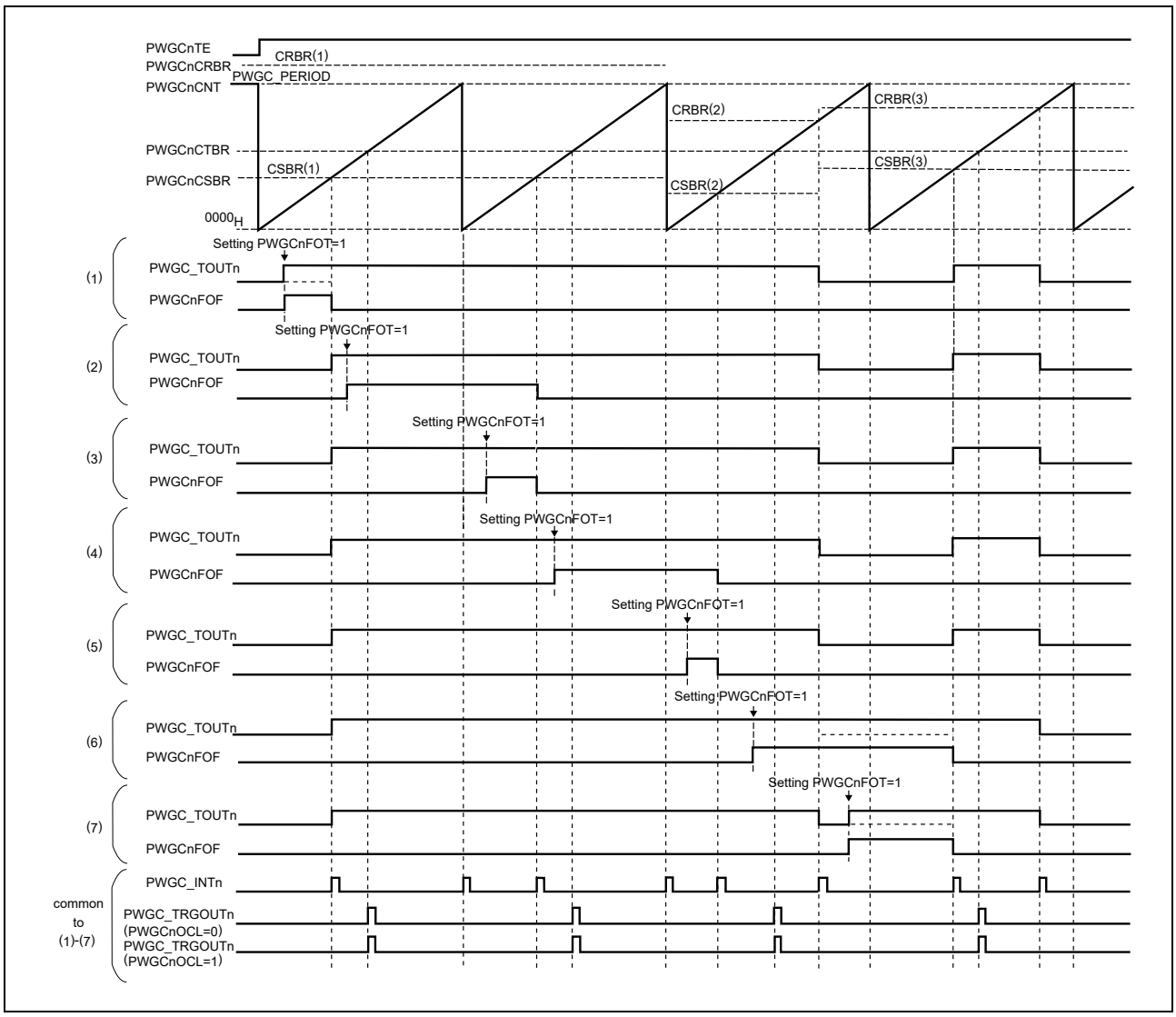


Figure 42.16 PWGC_TOUTn Output Fixed to High Level for PWM 100% Output Waveform

(1) shows the case of forcible high level fix during the low level period of transmission to PWM 100% output.

(2) shows the case of forcible high level fix during the high level period of transmission to PWM 100% output.

(3) shows the case of forcible high level fix during PWGCnCNT<=PWGCnCSBR period of PWM 100% output.

(4) shows the case of forcible high level fix during PWGCnCSBR<PWGCnCNT<=PWGC_PERIOD period of PWM 100% output.

(5) shows the case of forcible high level fix during PWGCnCNT<=PWGCnCSBR period of transmission from PWM 100% output to PWM normal output.

(6) shows the case of forcible high level fix during PWGCnCSBR<PWGCnCNT<=PWGCnCRBR period of transmission from PWM 100% output to PWM normal output.

(7) shows the case of forcible high level fix during the low level period of PWM normal output after PWM 100% output.

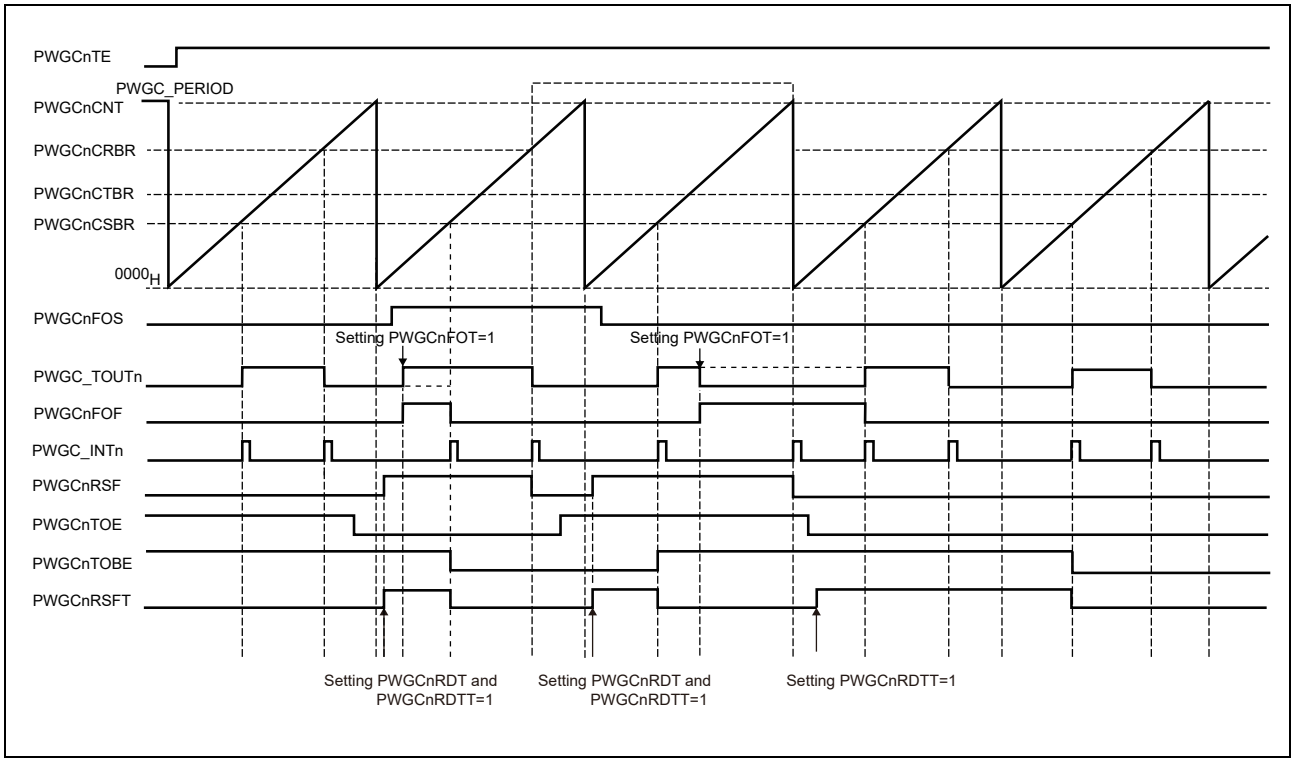


Figure 42.17 PWGCnTCBR Rewrite Waveform for Forcible PWGC_TOUTn Output Level Fix (PWGCnTCUT=01_B)

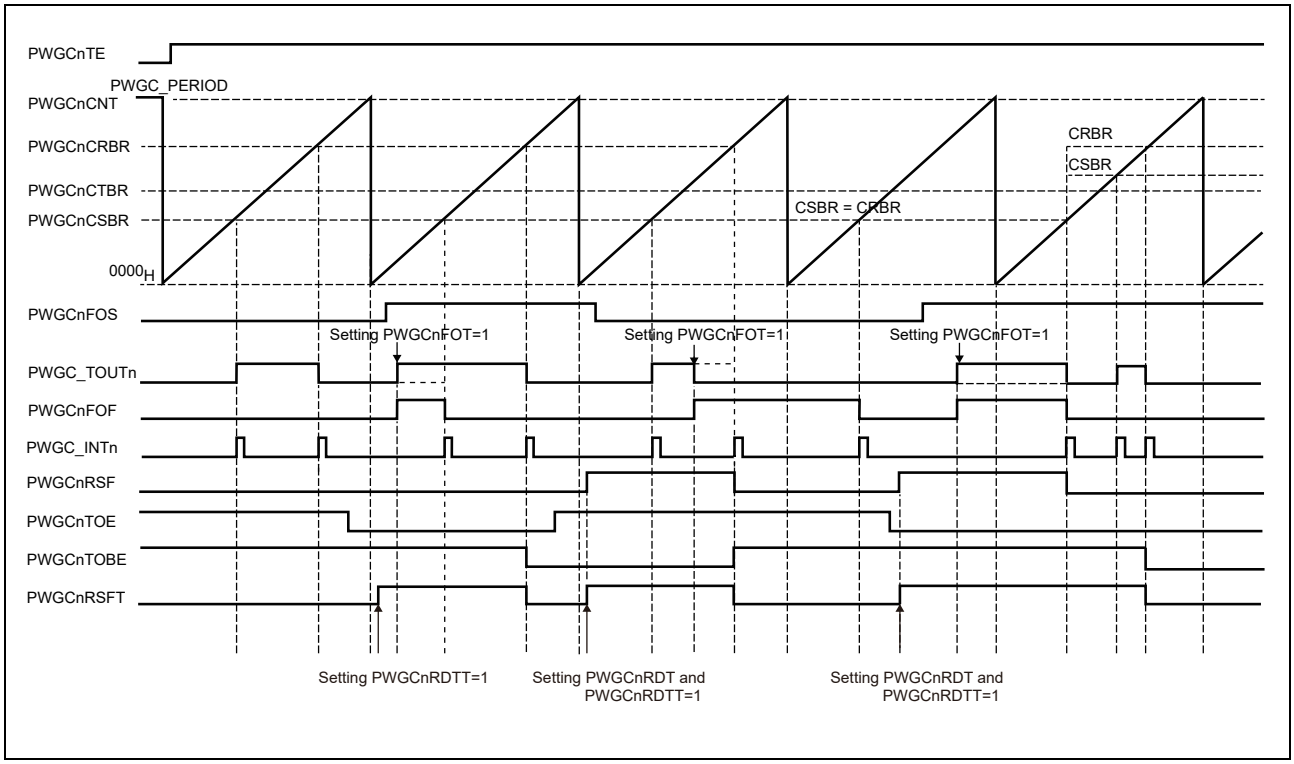
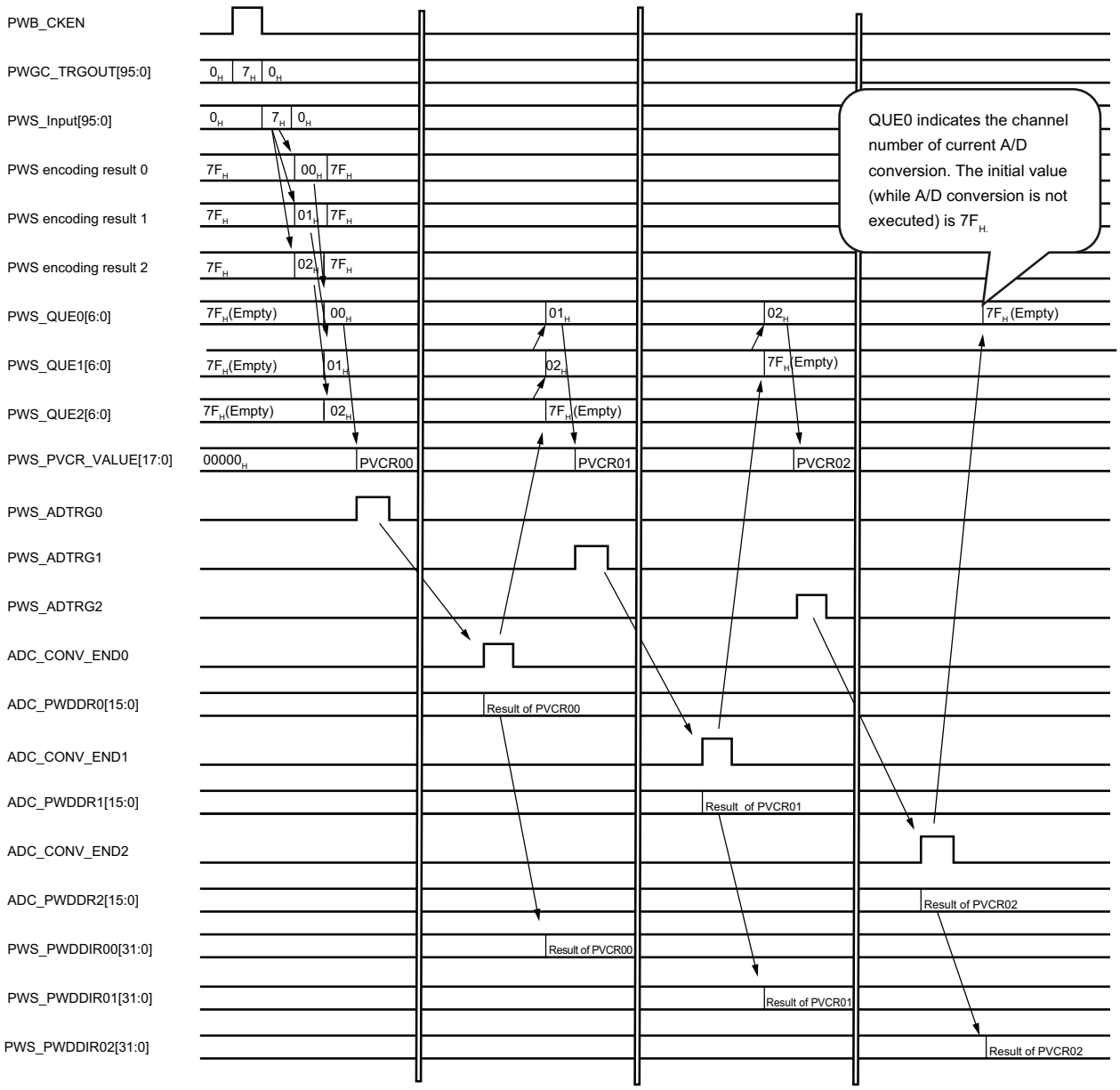


Figure 42.18 PWGCnTCBR Rewrite Waveform for Forcible PWGC_TOUTn Output Level Fix (PWGCnTCUT=10_B)

42.4.2.3 Operation Waveform when A/D Conversion Trigger Occurs in PWSD

An example of the PWSD operation is shown below.



Note 1. QUE0 indicates the PWGC_TRG number while the A/D conversion is executed. The initial value (while the A/D conversion is not executed) is $7F_H$.

Figure 42.19 Example of PWSD Operation

(1) Triggers occur simultaneously in channels 0, 1 and 2 of PWGC. Channel 0 with the smallest channel number is stored in PWSDnQUE0, channel 1 with the larger channel number is stored in PWSDnQUE1, and channel 2 with the largest channel number is stored in PWSDnQUE2. The value of PWSDnPVCR00 corresponding to the value stored in PWSDnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.
The A/D converter ADCJ0, ADCJ1 or ADCJ2 is selected by PWSDnSLAD00 bit of PWSDnPVCR00. This example shows that ADCJ0 is selected when PWSDnSLAD00 is 0.
At this time, as the A/D conversion for channel 1 and channel 2 is in the waiting state, the PWSDnSTR.PWSDnQNE bit is set.

(2) On completion of A/D conversion executed in step (1), the result of A/D conversion is stored in the lower 16 bits of PWSDnPWDDIR00. The higher 16 bits of PWSDnPWDDIR00 mirrors the information of PWSDnPVCR00. The channel number of PWSDnQUE1 shifts to PWSDnQUE0, PWSDnQUE2 shifts to PWSDnQUE1and PWSDnQUE2 enter empty state.
After that, as similar to step (1), the value of PWSDnPVCR01 corresponding to the value stored in PWSDnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter. This example shows that ADCJ1 is selected when PWSDnSLAD01 is 1.

(3) On completion of A/D conversion executed in step (2), the result of A/D conversion is stored in the lower 16 bits of PWSDnPWDDIR01. The higher 16 bits of PWSDnPWDDIR01 mirrors the information of PWSDnPVCR01. The channel number of PWSDnQUE1 shifts to PWSDnQUE0, PWSDnQUE1 enters empty state.
After that, as similar to step (1) or step (2), the value of PWSDnPVCR02 corresponding to the value stored in PWSDnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter. This example shows that ADCJ2 is selected when PWSDnSLAD02 is 2.

(4) On completion of A/D conversion executed in step (3), the result of A/D conversion is stored in the lower 16 bits of PWSDnPWDDIR02. The higher 16 bits of PWSDnPWDDIR02 mirrors the information of PWSDnPVCR02. PWSDnQUE0 enters the empty state, PWSDnSTR.PWSDnQNE bit is cleared.

42.4.3 PWM-Diag Related Functions in A/D Converter (ADCJ)

This section describes the A/D converter used for the PWM-Diag function.

42.4.3.1 ADCJ registers when the PWM-Diag function is used

- Before starting PWSD operation, the A/D converter must be set using the following register.
  - PWM-Diag scan group control register (ADCJvPWDSGCR)
- When the PWM-Diag is running, the PWSDnPVCRx value corresponding to the channel under conversion is set in the following register of the A/D converter.
  - PWM-Diag virtual channel register (ADCJvPWDVCR)
- After completion of A/D conversion, the conversion result can be checked by reading the following registers.
  - PWM-Diag data register (ADCJvPWDDR)
  - PWM-Diag data supplementary information register (ADCJvPWDDIR, PWSDnPWDDIRx*1)
- When A/D conversion result is outside the expected range, it can be confirmed using the upper/lower limit error detection function. The upper/lower limit error detection function is set by the following register.
  - Upper limit/lower limit error register (ADCJvSGULCR4)

Note 1. PWSDnPWDDIRx register is in PWSD.

Section 43 Analog to Digital Converter (ADCJ)

This section contains a description of the A/D Converter (ADCJ).
The first part of this section describes all of this products specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ADCJ.

43.1 Features of RH850/U2A-EVA ADCJ

43.1.1 Number of Units

This microcontroller has the following number of ADCJ units.

Table 43.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Name	ADCJn									
Number of ADCJ Units	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	3 (n = 0 to 2)	2 (n = 0, 1)	3 (n = 0 to 2)
Name	AVSEG									
Number of AVSEG Units	1	1	1	1	1	1	1	1	1	1

The number of physical channels and virtual channels on individual product are listed below.

Table 43.2 Unit Configurations and Physical Channels

Unit Name (Number of Channels) ADCJn		RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
ADCJ0	High accuracy inputs	20	20	20	20	20	20	20	14	8	10
	Low accuracy inputs	10	10	10	10	10	10	10	10	6	3
ADCJ1	High accuracy inputs	20	20	20	20	20	20	20	14	8	10
	Low accuracy inputs	14	14	14	14	14	14	14	14	7	10
ADCJ2	High accuracy inputs	20	20	20	5	20	5	5	5	0	0
	Low accuracy inputs	10	10	10	10	10	10	10	7	0	6

Table 43.3 Unit Configurations and Virtual Channels

Unit Name (Number of Channels) ADCJn	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
ADCJ0	64	64	64	64	64	64	64	64	64	64
ADCJ1	64	64	64	64	64	64	64	64	64	64
ADCJ2	64	64	64	64	64	64	64	64	0	64

Table 43.4 Indices of Products

Index	Meaning
n	Throughout this section, the individual ADCJ units are identified by the index “n” (n = 0 to 2); for example, ADCJnADHALTR indicates the A/D force halt register.
p	Throughout this section, the individual physical channel groups (channel group in the unit) of ADCJn are identified by the index “p”; for example, ANnpq.
q	Throughout this section, the individual physical sub channels (sub channel in the unit) of ADCJn are identified by the index “q”; for example, ANnpq.
j	Throughout this section, the individual virtual channels of ADCJn are identified by the index “j”; for example, ADCJnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCJn are identified by the index “x” (x = 0 to 4); for example, ADCJnSGSTCRx indicates the scan group x start control register.
y	Throughout this section, the individual MPX wait time setting or upper/lower limit setting registers are identified by the index “y” (y = 0 to 7); for example, ADCJnVCULLMTBRy indicates the Upper/Lower Limit Check Table Register y.
z	Throughout this section, the individual T&H unit are identified by the index “z” (z = 0 to 3); for example, THzE indicates the T&H unit z.
v	Throughout this section, the A/D conversion monitor channel pointer is indicated by the index “v” (v = 0 to 4); for example, ADCJnADENDPv indicates the A/D Conversion Monitor Virtual Channel Pointer v.

43.1.2 Register Base Addresses

The ADCJn base addresses are listed in the following table.

The ADCJn register addresses are given as offsets from the base address.

Table 43.5 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ADCJ0_base>	FFCA 0000$_H$	P-Bus Group 6L
<AVSEG_base>	FFCA 2000$_H$	P-Bus Group 6L
<ADCJ1_base>	FFF2 0000$_H$	P-Bus Group 7
<ADCJ2_base>	FF9A 2000$_H$	P-Bus Group 2L
<ADCJ2_SELB_base>	FF9A 3800$_H$	P-Bus Group 2L

43.1.3 Clock Supply

The ADCJn clock supplies are shown in the following table.

Table 43.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ADCJn (n = 0, 1)	CLKAD	CLK_ADC
	Register access clock	CLK_LSB
ADCJ2	CLKAD	CLKA_ADC
	Register access clock	CLKA_ADC
AVSEG	Register access clock	CLK_LSB

43.1.4 Interrupt Requests and Error Notifications

ADCJ interrupt requests are listed in the following table.

Table 43.7 Interrupt and DMA/DTS Requests

Interrupt Symbol Name	Unit Interrupt Name	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number	Other Trigger Signals
ADCJ0						
	INTADCJ0ERR (merged INT_ADE0 & INT_UL0)	A/D error interrupt upper/lower limit error interrupt	226	—	—	TAPA*2
	INTADCJ0SEC	Secondary voltage monitor error interrupt	225	—	—	—
	INTADCJ0I0	Scan group 0 (SG0) end interrupt	227	Group0-0	Group0-0	GTM*1
	INTADCJ0I1	Scan group 1 (SG1) end interrupt	228	Group0-1	Group0-1	GTM*1
	INTADCJ0I2	Scan group 2 (SG2) end interrupt	229	Group0-2	Group0-2	GTM*1
	INTADCJ0I3	Scan group 3 (SG3) end interrupt	230	Group0-3	Group0-3	GTM*1
	INTADCJ0I4	Scan group 4 (SG4) end interrupt	231	Group0-4	Group0-4	GTM*1
	ADMPXI0	MPX DMA trigger request	—	Group0-5	—	—
	ADC_CONV_END0	PWM A/D conversion end signal	—	Group0-6	Group0-5	—
ADCJ1						
	INTADCJ1ERR (merged INT_ADE1 & INT_UL1)	A/D error interrupt upper/lower limit error interrupt	232	—	—	TAPA*2
	INTADCJ1I0	Scan group 0 (SG0) end interrupt	233	Group0-7	Group0-6	GTM*1
	INTADCJ1I1	Scan group 1 (SG1) end interrupt	234	Group0-8	Group0-7	GTM*1
	INTADCJ1I2	Scan group 2 (SG2) end interrupt	235	Group0-9	Group0-8	GTM*1
	INTADCJ1I3	Scan group 3 (SG3) end interrupt	236	Group0-10	Group0-9	GTM*1
	INTADCJ1I4	Scan group 4 (SG4) end interrupt	237	Group0-11	Group0-10	GTM*1
	ADMPXI1	MPX DMA trigger request	—	Group0-12	—	—
	ADC_CONV_END1	PWM A/D conversion end signal	—	Group0-13	Group0-11	—
ADCJ2						
	INTADCJ2ERR (merged INT_ADE2 & INT_UL2)	A/D error interrupt upper/lower limit error interrupt	238	—	—	—
	INTADCJ2I0	Scan group 0 (SG0) end interrupt	239	Group0-14	Group0-12	LPS, WUP2
	INTADCJ2I1	Scan group 1 (SG1) end interrupt	240	Group0-15	Group0-13	LPS, WUP2
	INTADCJ2I2	Scan group 2 (SG2) end interrupt	241	Group0-16	Group0-14	LPS, WUP2
	INTADCJ2I3	Scan group 3 (SG3) end interrupt	242	Group0-17	Group0-15	LPS, WUP2
	INTADCJ2I4	Scan group 4 (SG4) end interrupt	243	Group0-18	Group0-16	LPS, WUP2
	ADMPXI2	MPX DMA trigger request	—	Group0-19	—	—
	ADC_CONV_END2	PWM A/D conversion end signal	—	Group0-20	Group0-17	—

Note 1. INTADCJ0I0 to INTADCJ0I4, INTADCJ1I0 to INTADCJ1I4 are chosen by the PIC (Peripheral Interconnect), and they are forwarded to the GTM. For details, refer to **Section 41, Peripheral Interconnect (PIC)**.

Note 2. INTADCJ0ERR and INTADCJ1ERR are chosen by the PIC (Peripheral Interconnect), and they are forwarded to the TAPA. For details, refer to **Section 41, Peripheral Interconnect (PIC)**.

ADCJ error notifications are listed in the following table.

Table 43.8 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
A/D converter parity error	ADCJ0 parity error	72	—
	ADCJ1 parity error	72	—
	ADCJ2 parity error	72	—

43.1.5 Reset Sources

ADCJn reset sources are listed in the following table. ADCJn is initialized by these reset sources.

Table 43.9 Reset Sources (For RH850/U2A-EVA)

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ADCJn (n = 0, 1)	All registers	√	√	√	√	√	√	—
ADCJ2	All registers	√	√	√	√	—	—	—
AVSEG	All registers	√	√	√	√	√	√	—

43.1.6 External Input/Output Signals

External input/output signals of ADCJ are listed in the following table.

Table 43.10 External Input/Output Signals (1/4)*2

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCJ0		
A0VCC*3	Power supply pin for the analog part	A0VCC
A0VSS	Ground pin for the analog part	A0VSS
A0VREFH*3	Reference voltage pin for the analog part	A0VREFH
AN000	Analog input pin	ADCJ0I0
AN001	Analog input pin	ADCJ0I1
AN002	Analog input pin	ADCJ0I2
AN003	Analog input pin	ADCJ0I3
AN010	Analog input pin	ADCJ0I4
AN011	Analog input pin	ADCJ0I5
AN012	Analog input pin	ADCJ0I6
AN013	Analog input pin	ADCJ0I7
AN020	Analog input pin	ADCJ0I8
AN021	Analog input pin	ADCJ0I9
AN022	Analog input pin	ADCJ0I10
AN023	Analog input pin	ADCJ0I11
AN030	Analog input pin	ADCJ0I12
AN031	Analog input pin	ADCJ0I13
AN032	Analog input pin	ADCJ0I14
AN033	Analog input pin	ADCJ0I15
AN040	Analog input pin	ADCJ0I16
AN041	Analog input pin	ADCJ0I17
AN042	Analog input pin	ADCJ0I18
AN043	Analog input pin	ADCJ0I19
AN060*1	Analog input pin	ADCJ0I0S
AN061*1	Analog input pin	ADCJ0I1S
AN062*1	Analog input pin	ADCJ0I2S
AN063*1	Analog input pin	ADCJ0I3S

Table 43.10 External Input/Output Signals (2/4)*2

Unit Signal Name	Outline	Alternative Port Pin Signal
AN070*1	Analog input pin	ADCJ0I4S
AN071*1	Analog input pin	ADCJ0I5S
AN072*1	Analog input pin	ADCJ0I6S
AN073*1	Analog input pin	ADCJ0I7S
AN080*1	Analog input pin	ADCJ0I8S
AN081*1	Analog input pin	ADCJ0I9S
ADCJ0TRG0	External hardware trigger pin 0	ADCJ0TRG0
ADCJ0TRG1	External hardware trigger pin 1	ADCJ0TRG1
ADCJ0TRG2	External hardware trigger pin 2	ADCJ0TRG2
ADCJ0TRG3	External hardware trigger pin 3	ADCJ0TRG3
ADCJ0TRG4	External hardware trigger pin 4	ADCJ0TRG4
MPXCUR0	External analog multiplexer (MPX) output pin 0	ADCJ0SEL0
MPXCUR1	External analog multiplexer (MPX) output pin 1	ADCJ0SEL1
MPXCUR2	External analog multiplexer (MPX) output pin 2	ADCJ0SEL2
ADEND0	A/D conversion timing monitor pin 0	ADCJ0CNV0
ADEND1	A/D conversion timing monitor pin 1	ADCJ0CNV1
ADEND2	A/D conversion timing monitor pin 2	ADCJ0CNV2
ADEND3	A/D conversion timing monitor pin 3	ADCJ0CNV3
ADEND4	A/D conversion timing monitor pin 4	ADCJ0CNV4
ADCJ1		
A1VCC*3	Power supply pin for the analog part	A1VCC
A1VSS	Ground pin for the analog part	A1VSS
A1VREFH*3	Reference voltage pin for the analog part	A1VREFH
AN100	Analog input pin	ADCJ1I0
AN101	Analog input pin	ADCJ1I1
AN102	Analog input pin	ADCJ1I2
AN103	Analog input pin	ADCJ1I3
AN110	Analog input pin	ADCJ1I4
AN111	Analog input pin	ADCJ1I5
AN112	Analog input pin	ADCJ1I6
AN113	Analog input pin	ADCJ1I7
AN120	Analog input pin	ADCJ1I8
AN121	Analog input pin	ADCJ1I9
AN122	Analog input pin	ADCJ1I10
AN123	Analog input pin	ADCJ1I11
AN130	Analog input pin	ADCJ1I12
AN131	Analog input pin	ADCJ1I13
AN132	Analog input pin	ADCJ1I14
AN133	Analog input pin	ADCJ1I15
AN140	Analog input pin	ADCJ1I16
AN141	Analog input pin	ADCJ1I17
AN142	Analog input pin	ADCJ1I18
AN143	Analog input pin	ADCJ1I19

Table 43.10 External Input/Output Signals (3/4)*2

Unit Signal Name	Outline	Alternative Port Pin Signal
AN150*1	Analog input pin	ADCJ1I0S
AN151*1	Analog input pin	ADCJ1I1S
AN152*1	Analog input pin	ADCJ1I2S
AN153*1	Analog input pin	ADCJ1I3S
AN160*1	Analog input pin	ADCJ1I4S
AN161*1	Analog input pin	ADCJ1I5S
AN162*1	Analog input pin	ADCJ1I6S
AN163*1	Analog input pin	ADCJ1I7S
AN170*1	Analog input pin	ADCJ1I8S
AN171*1	Analog input pin	ADCJ1I9S
AN172*1	Analog input pin	ADCJ1I10S
AN173*1	Analog input pin	ADCJ1I11S
AN180*1	Analog input pin	ADCJ1I12S
AN181*1	Analog input pin	ADCJ1I13S
ADCJ1TRG0	External hardware trigger pin 0	ADCJ1TRG0
ADCJ1TRG1	External hardware trigger pin 1	ADCJ1TRG1
ADCJ1TRG2	External hardware trigger pin 2	ADCJ1TRG2
ADCJ1TRG3	External hardware trigger pin 3	ADCJ1TRG3
ADCJ1TRG4	External hardware trigger pin 4	ADCJ1TRG4
MPXCUR0	External analog multiplexer (MPX) output pin 0	ADCJ1SEL0
MPXCUR1	External analog multiplexer (MPX) output pin 1	ADCJ1SEL1
MPXCUR2	External analog multiplexer (MPX) output pin 2	ADCJ1SEL2
ADEND0	A/D conversion timing monitor pin 0	ADCJ1CNV0
ADEND1	A/D conversion timing monitor pin 1	ADCJ1CNV1
ADEND2	A/D conversion timing monitor pin 2	ADCJ1CNV2
ADEND3	A/D conversion timing monitor pin 3	ADCJ1CNV3
ADEND4	A/D conversion timing monitor pin 4	ADCJ1CNV4
ADCJ2		
A2VCC*3	Power supply pin for the analog part	A2VCC
A2VSS	Ground pin for the analog part	A2VSS
A2VREFH*3	Reference voltage pin for the analog part	A2VREFH
AN200	Analog input pin	ADCJ2I0
AN201	Analog input pin	ADCJ2I1
AN202	Analog input pin	ADCJ2I2
AN203	Analog input pin	ADCJ2I3
AN210	Analog input pin	ADCJ2I4
AN211	Analog input pin	ADCJ2I5
AN212	Analog input pin	ADCJ2I6
AN213	Analog input pin	ADCJ2I7
AN220	Analog input pin	ADCJ2I8
AN221	Analog input pin	ADCJ2I9
AN222	Analog input pin	ADCJ2I10
AN223	Analog input pin	ADCJ2I11

Table 43.10 External Input/Output Signals (4/4)*2

Unit Signal Name	Outline	Alternative Port Pin Signal
AN230	Analog input pin	ADCJ2I12
AN231	Analog input pin	ADCJ2I13
AN232	Analog input pin	ADCJ2I14
AN233	Analog input pin	ADCJ2I15
AN240	Analog input pin	ADCJ2I16
AN241	Analog input pin	ADCJ2I17
AN242	Analog input pin	ADCJ2I18
AN243	Analog input pin	ADCJ2I19
AN250*1	Analog input pin	ADCJ2I0S
AN251*1	Analog input pin	ADCJ2I1S
AN252*1	Analog input pin	ADCJ2I2S
AN253*1	Analog input pin	ADCJ2I3S
AN260*1	Analog input pin	ADCJ2I4S
AN261*1	Analog input pin	ADCJ2I5S
AN262*1	Analog input pin	ADCJ2I6S
AN263*1	Analog input pin	ADCJ2I7S
AN270*1	Analog input pin	ADCJ2I8S
AN271*1	Analog input pin	ADCJ2I9S
ADCJ2TRG0	External trigger pin (Scan group 0)	ADCJ2TRG0
ADCJ2TRG1	External trigger pin (Scan group 1)	ADCJ2TRG1
ADCJ2TRG2	External trigger pin (Scan group 2)	ADCJ2TRG2
ADCJ2TRG3	External trigger pin (Scan group 3)	ADCJ2TRG3
ADCJ2TRG4	External trigger pin (Scan group 4)	ADCJ2TRG4
MPXCUR0	External analog multiplexer (MPX) output pin 0	ADCJ2SEL0
MPXCUR1	External analog multiplexer (MPX) output pin 1	ADCJ2SEL1
MPXCUR2	External analog multiplexer (MPX) output pin 2	ADCJ2SEL2
ADEND0	A/D conversion timing monitor pin 0	ADCJ2CNV0
ADEND1	A/D conversion timing monitor pin 1	ADCJ2CNV1
ADEND2	A/D conversion timing monitor pin 2	ADCJ2CNV2
ADEND3	A/D conversion timing monitor pin 3	ADCJ2CNV3
ADEND4	A/D conversion timing monitor pin 4	ADCJ2CNV4

Note 1. Low accuracy inputs.
Note 2. For available pins of each product, see **Section 2, Pin Functions**.
Note 3. For U2A6, the AnVREFH should be connected with AnVCC.

For ADCJn, it's possible to perform an A/D conversion of an analog input with the assistance of RRAMP.

For ADCJ0, it's possible to perform an A/D conversion of VCC, E0VCC, ISOVDD and AWOVDD as secondary power supply voltage monitor.

For analog inputs and VMON (Secondary power supply voltage monitor), the correspondence list of physical channel, physical channel groups, physical sub channel and RRAMP is shown in the following table.

Table 43.11 ADCJ0 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input or VMON	Physical Channel	Physical Channel Group	Physical Sub Channel	Pin-Level Self-Diagnosis Voltage Group	T&H Connection	RRAMP Correspondence	
						RRAMP00	RRAMP01
AN000	0	0	0	Even	T&H0	√	—
AN001	1		1	Odd	T&H1	—	√
AN002	2		2	Even	T&H2	√	—
AN003	3		3	Odd	T&H3	—	√
AN010	4	1	0	Even	—	√	—
AN011	5		1	Odd	—	—	√
AN012	6		2	Even	—	√	—
AN013	7		3	Odd	—	—	√
AN020	8	2	0	Even	—	√	—
AN021	9		1	Odd	—	—	√
AN022	10		2	Even	—	√	—
AN023	11		3	Odd	—	—	√
AN030	12	3	0	Even	—	√	—
AN031	13		1	Odd	—	—	√
AN032	14		2	Even	—	√	—
AN033	15		3	Odd	—	—	√
AN040	16	4	0	Even	—	√	—
AN041	17		1	Odd	—	—	√
AN042	18		2	Even	—	√	—
AN043	19		3	Odd	—	—	√
VMON_VCC*1	20	5	0	—	—	—	—
VMON_E0VCC*2	21		1	—	—	—	—
VMON_ISOVDD*3	22		2	—	—	—	—
VMON_AWOVDD*4	23		3	—	—	—	—
AN060*5	24	6	0	Even	—	√	—
AN061*5	25		1	Odd	—	√	—
AN062*5	26		2	Even	—	√	—
AN063*5	27		3	Odd	—	√	—
AN070*5	28	7	0	Even	—	√	—
AN071*5	29		1	Odd	—	√	—
AN072*5	30		2	Even	—	√	—
AN073*5	31		3	Odd	—	√	—
AN080*5	32	8	0	Even	—	√	—
AN081*5	33		1	Odd	—	√	—

Note 1. This is the secondary supply voltage monitor for VCC.
Note 2. This is the secondary supply voltage monitor for E0VCC.
Note 3. This is the secondary supply voltage monitor for ISOVDD.
Note 4. This is the secondary supply voltage monitor for AWOVDD.
Note 5. Low accuracy inputs.

Table 43.12 ADCJ1 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input or VMON	Physical Channel	Physical Channel Group	Physical Sub Channel	Pin-Level Self-Diagnosis Voltage Group	T&H Connection	RRAMP Correspondence	
						RRAMP00	RRAMP01
AN100	0	0	0	Even	T&H0	√	—
AN101	1		1	Odd	T&H1	—	√
AN102	2		2	Even	T&H2	√	—
AN103	3		3	Odd	T&H3	—	√
AN110	4	1	0	Even	—	√	—
AN111	5		1	Odd	—	—	√
AN112	6		2	Even	—	√	—
AN113	7		3	Odd	—	—	√
AN120	8	2	0	Even	—	√	—
AN121	9		1	Odd	—	—	√
AN122	10		2	Even	—	√	—
AN123	11		3	Odd	—	—	√
AN130	12	3	0	Even	—	√	—
AN131	13		1	Odd	—	—	√
AN132	14		2	Even	—	√	—
AN133	15		3	Odd	—	—	√
AN140	16	4	0	Even	—	√	—
AN141	17		1	Odd	—	—	√
AN142	18		2	Even	—	√	—
AN143	19		3	Odd	—	—	√
AN150*1	20	5	0	Even	—	√	—
AN151*1	21		1	Odd	—	√	—
AN152*1	22		2	Even	—	√	—
AN153*1	23		3	Odd	—	√	—
AN160*1	24	6	0	Even	—	√	—
AN161*1	25		1	Odd	—	√	—
AN162*1	26		2	Even	—	√	—
AN163*1	27		3	Odd	—	√	—
AN170*1	28	7	0	Even	—	√	—
AN171*1	29		1	Odd	—	√	—
AN172*1	30		2	Even	—	√	—
AN173*1	31		3	Odd	—	√	—
AN180*1	32	8	0	Even	—	√	—
AN181*1	33		1	Odd	—	√	—

Note 1. Low accuracy inputs.

Table 43.13 ADCJ2 Physical Channel, Physical Channel Group, Physical Sub Channel and RRAMP

Analog Input or VMON	Physical Channel	Physical Channel Group	Physical Sub Channel	Pin-Level Self-Diagnosis Voltage Group	T&H Connection	RRAMP Correspondence	
						RRAMP00	RRAMP01
AN200	0	0	0	Even	—	√	—
AN201	1		1	Odd	—	—	√
AN202	2		2	Even	—	√	—
AN203	3		3	Odd	—	—	√
AN210	4	1	0	Even	—	√	—
AN211	5		1	Odd	—	—	√
AN212	6		2	Even	—	√	—
AN213	7		3	Odd	—	—	√
AN220	8	2	0	Even	—	√	—
AN221	9		1	Odd	—	—	√
AN222	10		2	Even	—	√	—
AN223	11		3	Odd	—	—	√
AN230	12	3	0	Even	—	√	—
AN231	13		1	Odd	—	—	√
AN232	14		2	Even	—	√	—
AN233	15		3	Odd	—	—	√
AN240	16	4	0	Even	—	√	—
AN241	17		1	Odd	—	—	√
AN242	18		2	Even	—	√	—
AN243	19		3	Odd	—	—	√
AN250*1	20	5	0	Even	—	√	—
AN251*1	21		1	Odd	—	√	—
AN252*1	22		2	Even	—	√	—
AN253*1	23		3	Odd	—	√	—
AN260*1	24	6	0	Even	—	√	—
AN261*1	25		1	Odd	—	√	—
AN262*1	26		2	Even	—	√	—
AN263*1	27		3	Odd	—	√	—
AN270*1	28	7	0	Even	—	√	—
AN271*1	29		1	Odd	—	√	—

Note 1. Low accuracy inputs.

43.2 Overview

43.2.1 Functional Overview

(1) Basic functions

- A/D converter
  Resolution: 12-bit
  A/D conversion method: Successive approximation
  Conversion speed: 1.0 µs (min)
- Sample-and-hold function
  Each ADCJ has an internal sample-and-hold circuit that enables each module to perform an A/D conversion independently. The sampling time can be selected from several options.
- Extended physical channels (MPX function)
  Each ADCJ can extend physical channels by using an external analog multiplexer.
- Track & Hold (T&H) input channels
  Several channel inputs can select T&H circuit for synchronize conversion.
- A/D-converted value adding function
  The ADCJ performs A/D conversion sequentially twice or four times for a channel, and stores the addition result in the data register.
- A/D conversion monitor output
  The processing timing of a desired virtual channel can be output to the A/D conversion monitor output pin.

(2) Virtual channels

- Virtual channel concept
  The number of virtual channels is larger than the number of physical channels.
  Each virtual channel can be freely assigned to each physical conversion channel.
- Data registers
  Data registers corresponding to virtual channels are provided.

(3) Scan group

- Supporting 5 scan groups
  Each ADCJ has 5 scan groups (SG0, SG1, SG2, SG3, and SG4).
  SG4 can select PWM-Diag function.
  Scan settings can be made independently for each scan group.
- ADCJ has two scan modes.
  - Multicycle scan mode executes the specified number of scans.
  - Continuous scan mode executes scans repeatedly without limit.
- Start trigger for each scan group
  Hardware triggers and software triggers can start the processing of each scan group. Only scan groups 3(SG3) and 4(SG4) can start processing by an A/D timer trigger.
- Asynchronous/synchronous suspend and resume function
  A processing for a scan group can be interrupted to run the processing for another scan group.
- Scan end interrupt

Each scan group can generate an interrupt request to the INTC and activate the DMAC/DTS each time a processing for the virtual channel indicated by the end virtual channel pointer ends or a virtual channel ends.

- Interval function
  ADCJ can start scan groups in any cycle by using the A/D timer equipped in the scan groups 3 and 4. This enables scans with intervals inserted.

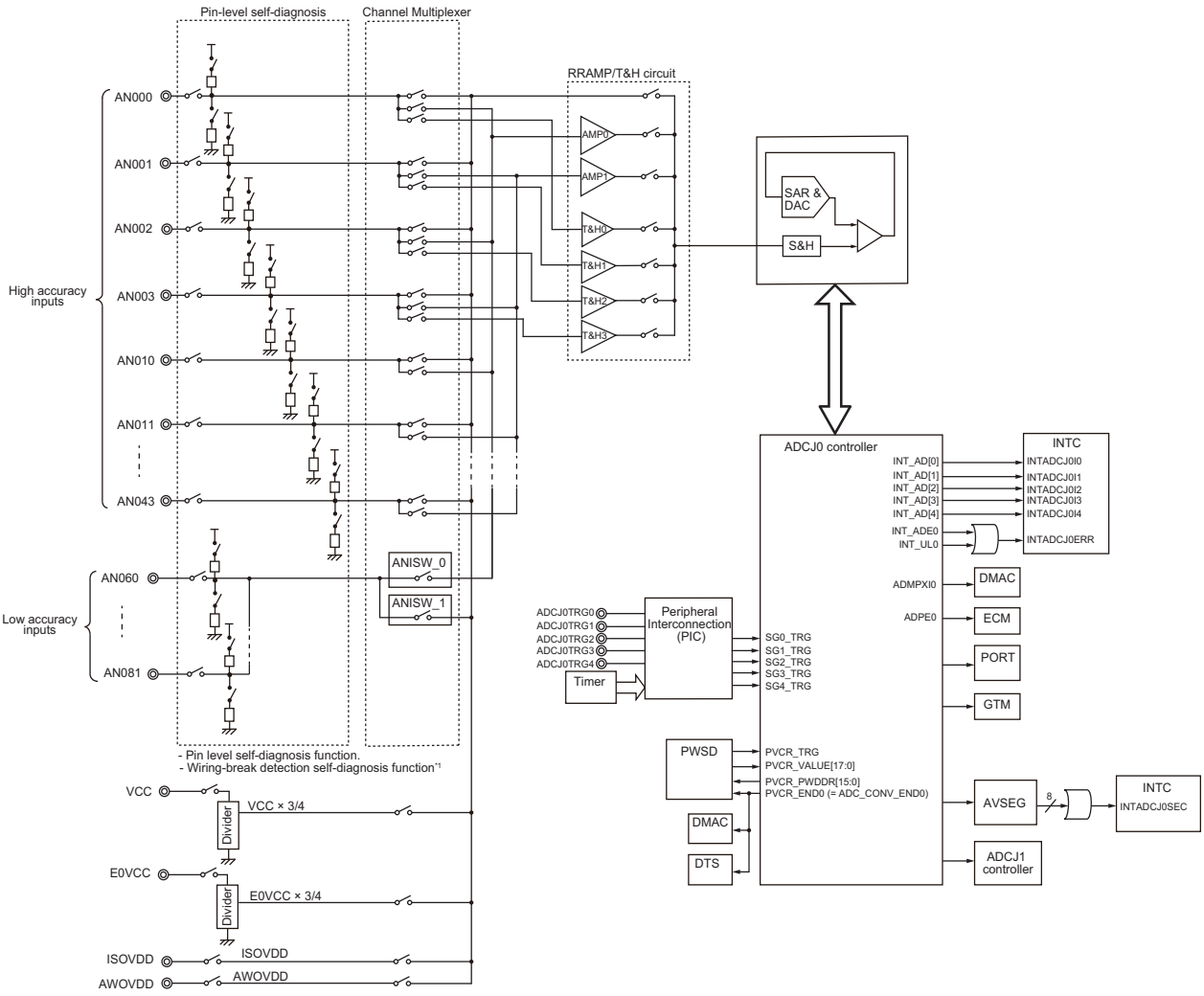
(4) Safety-related functions

- Diagnosis
  - wiring-break detection
  - voltage monitor
- Self-diagnosis
  - Self-diagnosis of the wiring-break detection
  - A/D converter self-diagnosis
  - Pin-level self-diagnosis
  - Self-diagnosis of the T&H path
  - Self-diagnosis of the voltage monitor function
  - A/D conversion of A/D conversion data path diagnosis
- Error Detection
  - Read and clear functions for the data registers
  - Parity check for the data registers
  - Overwrite check for the data registers
  - ID error
  - Upper/Lower Limit Check (8 tables)
  - Trigger overlap Check
- Upper/lower-limit-excess-notice-function to ADC VMON secondary error generator (AVSEG) in each virtual channel
  The ADCJ0 can output to the ADC-VMON-secondary-error-generator (AVSEG) a signal to notify that an A/D conversion result has increased above the upper limit value of the designated table in each virtual channel or that an A/D conversion result has decreased below the lower limit value of the designated table in each virtual channel. Once the Table register for the upper/lower limit value is set, it can be rewritten at any time.
  For details of the AVSEG, see **Section 43.7, ADC VMON Secondary Error Generator (AVSEG)**.
- Secondary power supply voltage monitor (VMON)
  The ADCJ0 can convert the voltage of VCC, E0VCC, ISOVDD and AWOVDD power supply in AD. Secondary HDET, Secondary LDET of each power supply (VCC, E0VCC, ISOVDD and AWOVDD) can generate an interrupt request to INTC by using the upper/lower-limit-execution-notice-function of ADCJ0 and AVSEG.
  For details of the AVSEG, see **Section 43.7, ADC VMON Secondary Error Generator (AVSEG)**.
  For the whole description of power supply voltage monitor, refer to **Section 11, Power Supply Voltage Monitor**.

43.2.2 Block Diagram

Figure 43.1 show the ADCJn block diagram.

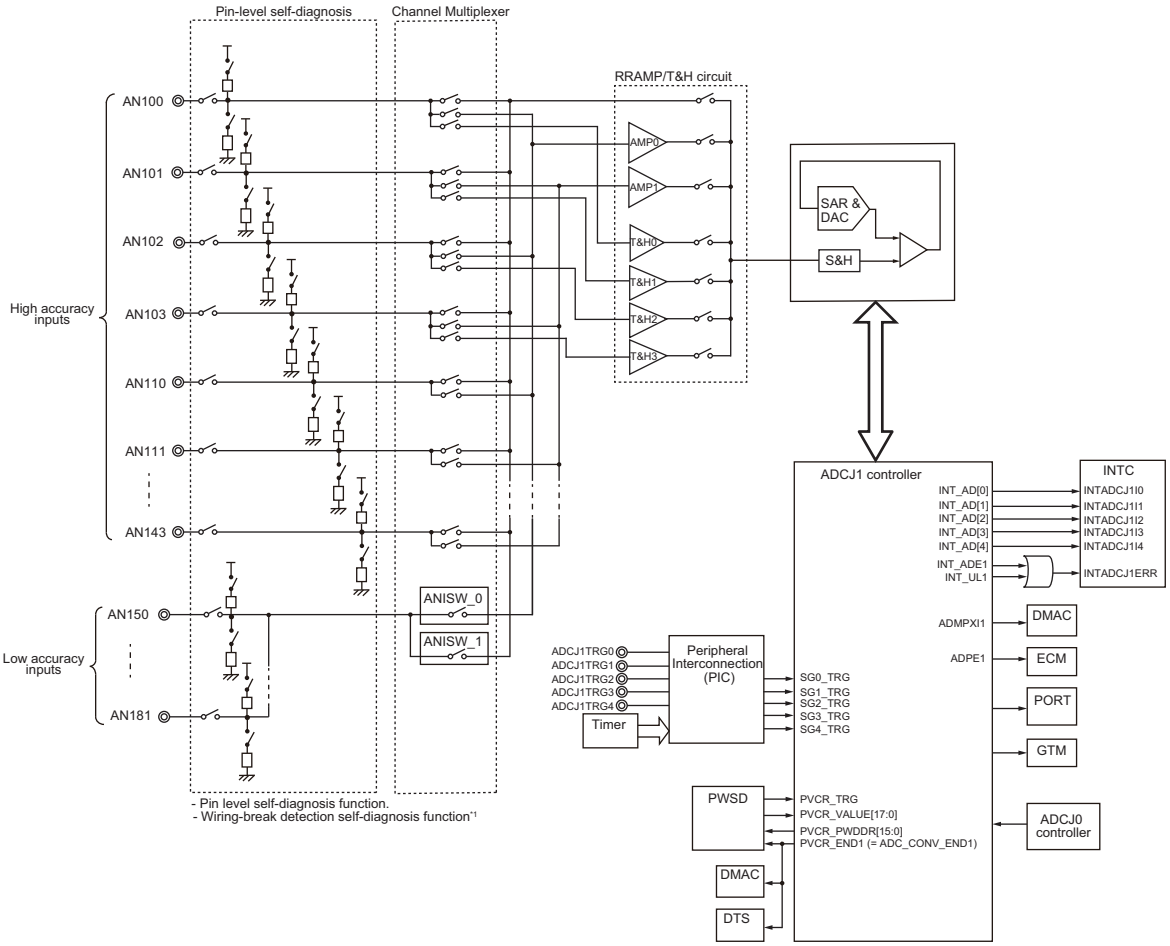
(1) ADCJ0 block diagram



Note 1. For details, see **Section 43.4.10, Self-Diagnostic Functions**.

Figure 43.1 Block Diagram of ADCJ0

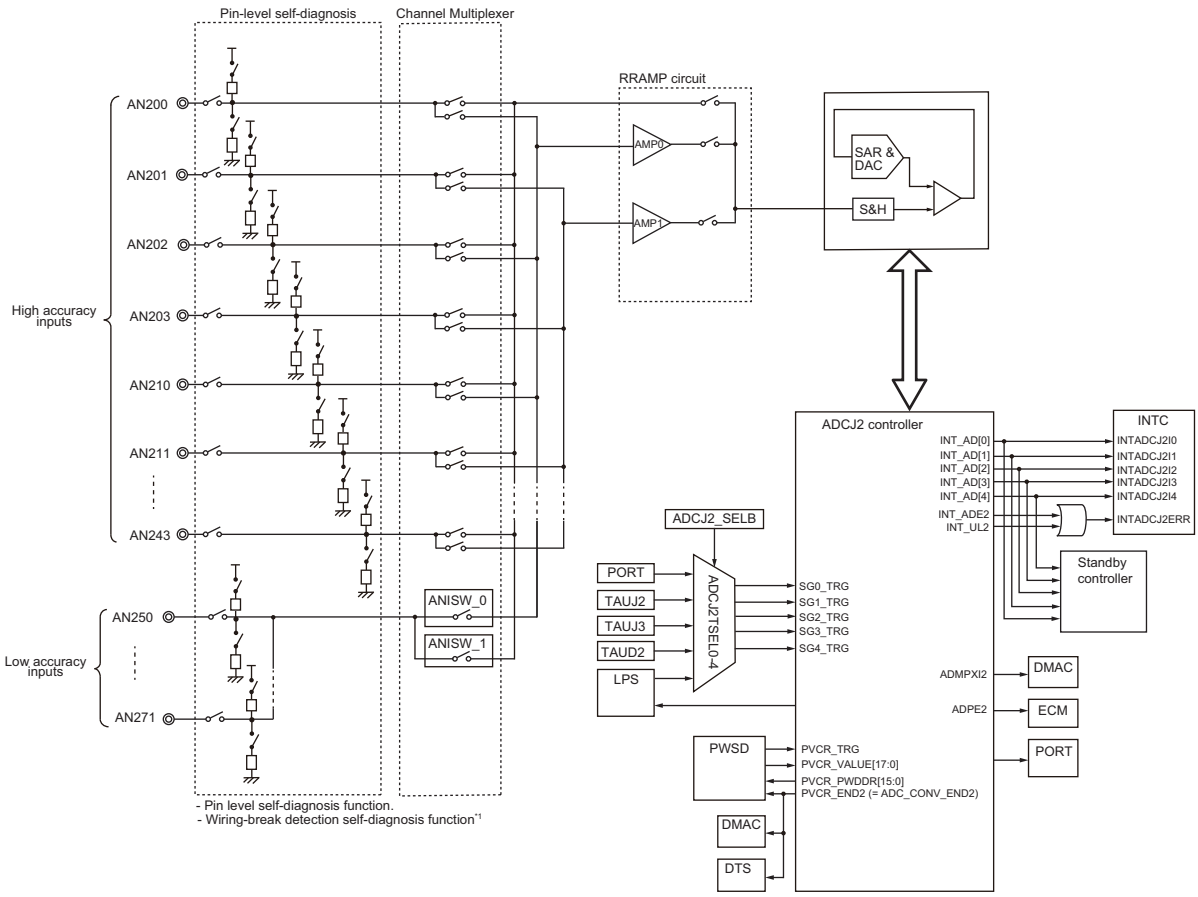
(2) ADCJ1 block diagram



Note 1. For details, see **Section 43.4.10, Self-Diagnostic Functions**.

Figure 43.2 Block Diagram of ADCJ1

(3) ADCJ2 block diagram



Note 1. For details, see **Section 43.4.10, Self-Diagnostic Functions**.

Figure 43.3 Block Diagram of ADCJ2

43.2.3 Explanation of Terms

43.2.3.1 Physical channel

Physical channel means ANnpq that is an A/D conversion target of the 12-bit A/D converter.

43.2.3.2 Virtual Channel

Virtual channel means a structure to enable settings (such as A/D conversion type and physical channel) necessary for A/D conversion by setting a single register. Physical channel numbers can be freely selected for each virtual channel (ADCJnVCRj) by setting the GCTRL[5:0] bits in ADCJnVCRj.

Arbitrary consecutive virtual channel numbers can be set for each scan group (SG0-SG4). Set the starting virtual channel number and the end virtual channel number in the scan group x virtual channel pointer register (ADCJnSGVCPRx) provided for each scan group.
Because the ADCJ is configured to input the virtual channel setting one by one to 12-bit A/D converter for execution, multiple A/D conversion is not processed concurrently.

Starting virtual channel pointer	ADCJnSGVCPRx.VCSP[5:0]
End virtual channel pointer	ADCJnSGVCPRx.VCEP[5:0]

A/D conversion result is stored in the data register (ADCJnDRj) provided for each virtual channel.

A/D conversion result data for consecutive two channels (16 bits × 2 = 32 bits) is stored in ADCJnDRj.

A/D conversion result is also stored in the data supplementary information register (ADCJnDIRj). This register includes supplementary information in addition to A/D conversion result.

43.2.3.3 Scan Group

Set arbitrary number of consecutive virtual channels and activate a scan group by a startup trigger to perform A/D conversion continuously in ascending order with the virtual channel setting.

The priority is set for each scan group as follows:

SG0 < SG1 < SG2 < SG3 < SG4

If a high-priority scan group is activated while scan group x is operating, the scan group in progress is suspended and the high-priority scan group is executed instead. After all virtual channels in the high-priority scan group have been processed, processing of scan group x is resumed.

For details, see **Section 43.4.17, Example of Suspend and Resume Operation**.

Figure 43.4 shows that the relationship between virtual channel transition, virtual channel and Data Register, virtual channel and scan group. This figure shows that scan group 2 is processed from virtual channel 16 to 20, and scan group 1 is processed from virtual channel 4 and virtual channel 7 is being processed now.
The number of Data Register corresponds with the number of virtual channel. The number of physical channel can be set each virtual channel, same physical channel number can be set different virtual channels.

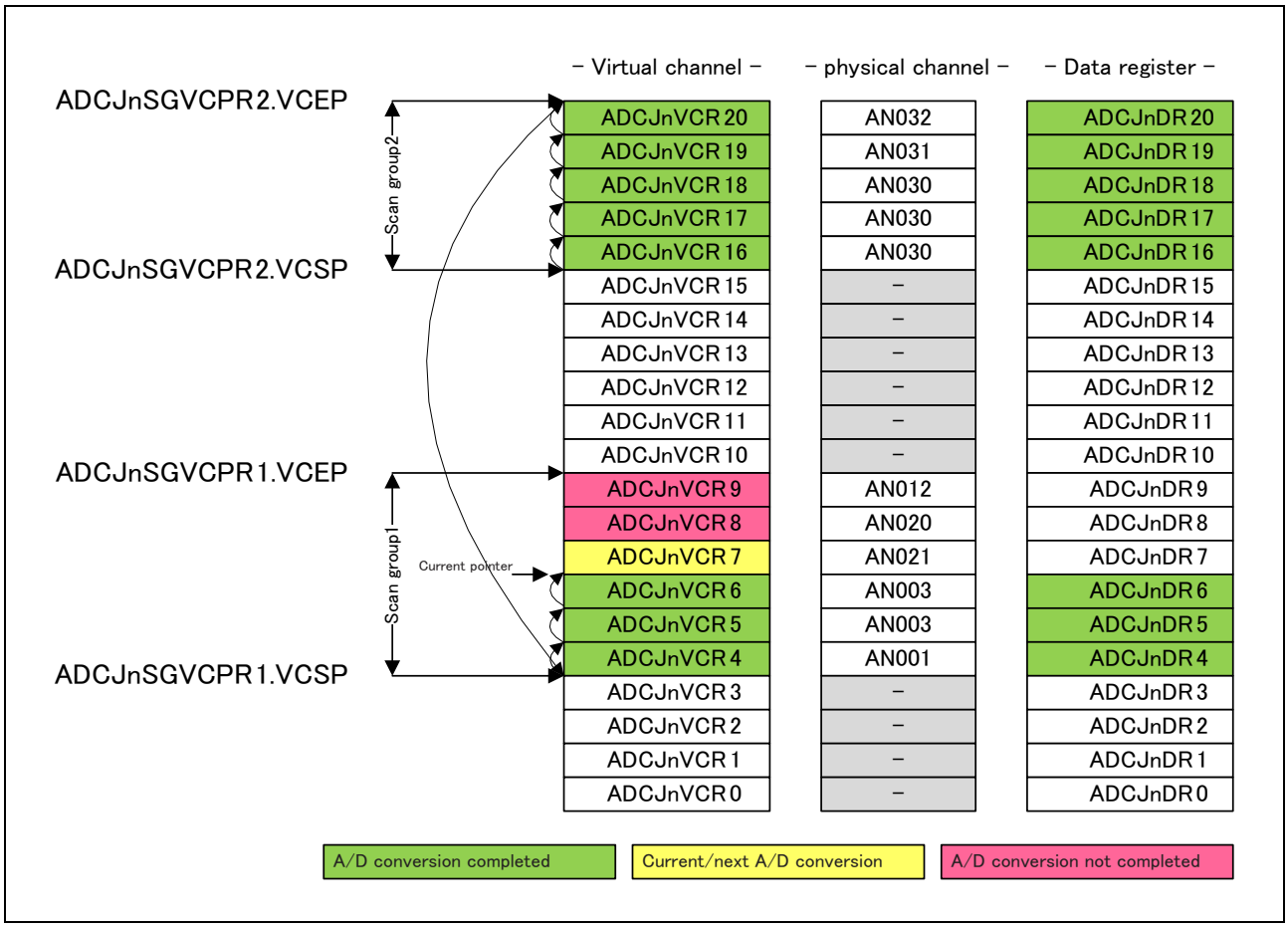


Figure 43.4 Relationship among physical channels, Virtual Channels, and Scan Groups

43.2.3.4 Data Format

This section describes data formats of the ADCJ.

(1) Data Formats of Data Registers

Formats in combination of data format (ADCJnADCR2.DFMT[2:0]), addition mode (ADCJnVCRj.CNVCLS[3:0]), and addition count (ADCJnADCR2.ADDNT, ADCJnODCR.WADDE) stored in data registers (ADCJnDRj, ADCJnPWDDR) are shown below.

(1) For 12bit signed fixed-point format (DFMT[2:0] = 000_B)

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S													0	0	0
Convert twice	S														0	0
Convert 4 times	S															0

↑ Position of decimal point

(2) For 12-bit signed integer format (DFMT[2:0] = 001_B)

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S	S	S	S												
Convert twice	S	S	S													
Convert 4 times	S	S														

↗ Position of decimal point

(3) For 12-bit unsigned fixed-point format (DFMT[2:0] = 010_B)

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once													0	0	0	0
Convert twice														0	0	0
Convert 4 times															0	0

↑ Position of decimal point

(4) For 12-bit unsigned integer format (Right) (DFMT[2:0] = 100_B)

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	0	0	0	0												
Convert twice	0	0	0													
Convert 4 times	0	0														

↗ Position of decimal point

(5) For 12-bit unsigned integer format (Left) (DFMT[2:0] = 101_B)

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once													0	0	0	0
Convert twice														0	0	0
Convert 4 times															0	0

Position of decimal point

S	: Sign bit (always 0)
0	: Zero extension

The format setting in ADDNT is valid when CNVCLS[3:0] = 4_H or 6_H.

If CNVCLS[3:0] is not 4_H or 6_H, the format is “convert once”.

CAUTION

Setting any value other than above values for ADCJnADCR2.DFMT[2:0] is prohibited.

(2) Data Format of Upper/Lower Limit Check Table Register for data register

When upper/lower limit check is used for data register (ADCJnDRj, ADCJnPWDDR), set the value in the upper/lower limit check table register as shown below.

(1) When DFMT[2:0] = 000_B, DFMT[2:0] = 010_B

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S													0	0	0
Convert twice	S														0	0
Convert 4 times	S															0

(2) When DFMT[2:0] = 001_B

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S	S	S	S												
Convert twice	S	S	S													
Convert 4 times	S	S														

(3) When DFMT[2:0] = 100_B, DFMT[2:0] = 101_B

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S	S	S	S												
Convert twice	S	S	S													
Convert 4 times	S	S														

(3) Data Format for Output to GTM

The data format for output to GTM is independent of the ADCJnADCR2.DFMT[2:0] setting. Data is output in the 12-bit signed fixed-point format as shown below.

- Data is output in the 12-bit (resolution) signed fixed-point format.

Number of addition	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Convert once	S													0	0	0
Convert twice	S														0	0
Convert 4 times	S															0

↑ Position of decimal point

Shaded part: Fixed to 0 when DFMT = 011_B

S	: Sign bit (always 0)
0	: Zero extension

The format setting in ADDNT is valid when CNVCLS[3:0] = 4_H or 6_H.

If CNVCLS[3:0] is not 4_H or 6_H, the format is “convert once”.

43.3 Registers

43.3.1 List of Registers

ADCJ registers are listed in the following table.

Table 43.14 List of Registers (1/3)

Register Name	Symbol	Address	Access Protection	
			PBG	Other
ADC specific registers (virtual channel)				
Virtual channel register j	ADCJnVCRj	<ADCJn_base> + j × 4_H	*1	—
PWM-Diag Virtual Channel Register	ADCJnPWDVCR	<ADCJn_base> + 180_H	*1	—
Data register j (j = only even numbers)	ADCJnDRj	<ADCJn_base> + $1A0_H$ + j × 2_H	*1	—
PWM-Diag Data Register	ADCJnPWDDR	<ADCJn_base>+260_H	*1	—
Data supplementary information register j	ADCJnDIRj	<ADCJn_base> + 280_H + j × 4_H	*1	—
PWM-Diag data supplementary information register	ADCJnPWDDIR	<ADCJn_base>+400_H	*1	—
Scan group 0 start control register	ADCJnSGSTCR0	<ADCJn_base>+440_H	*1	—
Scan group 0 stop control register	ADCJnSGSTPCR0	<ADCJn_base>+444_H	*1	—
Scan group 0 control register	ADCJnSGCR0	<ADCJn_base>+450_H	*1	—
Scan group 0 virtual channel pointer	ADCJnSGVCPR0	<ADCJn_base>+454_H	*1	—
Scan group 0 multicycle register	ADCJnSGMCYCR0	<ADCJn_base>+458_H	*1	—
Scan group 0 status register	ADCJnSGSR0	<ADCJn_base>+460_H	*1	—
Scan group 1 start control register	ADCJnSGSTCR1	<ADCJn_base>+480_H	*1	—
Scan group 1 stop control register	ADCJnSGSTPCR1	<ADCJn_base>+484_H	*1	—
Scan group 1 control register	ADCJnSGCR1	<ADCJn_base>+490_H	*1	—
Scan group 1 virtual channel pointer	ADCJnSGVCPR1	<ADCJn_base>+494_H	*1	—
Scan group 1 multicycle register	ADCJnSGMCYCR1	<ADCJn_base>+498_H	*1	—
Scan group 1 status register	ADCJnSGSR1	<ADCJn_base>+$4A0_H$	*1	—
Scan group 2 start control register	ADCJnSGSTCR2	<ADCJn_base>+$4C0_H$	*1	—
Scan group 2 stop control register	ADCJnSGSTPCR2	<ADCJn_base>+$4C4_H$	*1	—
Scan group 2 control register	ADCJnSGCR2	<ADCJn_base>+$4D0_H$	*1	—
Scan group 2 virtual channel pointer	ADCJnSGVCPR2	<ADCJn_base>+$4D4_H$	*1	—
Scan group 2 multicycle register	ADCJnSGMCYCR2	<ADCJn_base>+$4D8_H$	*1	—
Scan group 2 status register	ADCJnSGSR2	<ADCJn_base>+$4E0_H$	*1	—
Scan group 3 start control register	ADCJnSGSTCR3	<ADCJn_base>+500_H	*1	—
Scan group 3 stop control register	ADCJnSGSTPCR3	<ADCJn_base>+504_H	*1	—
A/D timer 3 start control register	ADCJnADTSTCR3	<ADCJn_base>+508_H	*1	—
A/D timer 3 end control register	ADCJnADTENDCR3	<ADCJn_base>+$50C_H$	*1	—
Scan group 3 control register	ADCJnSGCR3	<ADCJn_base>+510_H	*1	—
Scan group 3 virtual channel pointer	ADCJnSGVCPR3	<ADCJn_base>+514_H	*1	—
Scan group 3 multicycle register	ADCJnSGMCYCR3	<ADCJn_base>+518_H	*1	—
Scan group 3 status register	ADCJnSGSR3	<ADCJn_base>+520_H	*1	—
A/D timer 3 initial phase register	ADCJnADTIPR3	<ADCJn_base>+524_H	*1	—
A/D timer 3 cycle register	ADCJnADTPRR3	<ADCJn_base>+528_H	*1	—
Scan group 4 start control register	ADCJnSGSTCR4	<ADCJn_base>+540_H	*1	—
Scan group 4 stop control register	ADCJnSGSTPCR4	<ADCJn_base>+544_H	*1	—
A/D timer 4 start control register	ADCJnADTSTCR4	<ADCJn_base>+548_H	*1	—
A/D timer 4 end control register	ADCJnADTENDCR4	<ADCJn_base>+$54C_H$	*1	—
Scan group 4 control register	ADCJnSGCR4	<ADCJn_base>+550_H	*1	—
Scan group 4 virtual channel pointer	ADCJnSGVCPR4	<ADCJn_base>+554_H	*1	—

Table 43.14 List of Registers (2/3)

Register Name	Symbol	Address	Access Protection	
			PBG	Other
Scan group 4 multicycle register	ADCJnSGMCYCR4	<ADCJn_base>+558_H	*1	—
Scan group 4 status register	ADCJnSGSR4	<ADCJn_base>+560_H	*1	—
A/D timer 4 initial phase register	ADCJnADTIPR4	<ADCJn_base>+564_H	*1	—
A/D timer 4 cycle register	ADCJnADTPRR4	<ADCJn_base>+568_H	*1	—
PWM-Diag control register	ADCJnPWDCR	<ADCJn_base>+580_H	*1	—
PWM-Diag scan group control register	ADCJnPWDSGCR	<ADCJn_base>+584_H	*1	—
Scan Group Common Status Register	ADCJnSGSTR	<ADCJn_base>+600_H	*1	—
A/D synchronization start control register	ADCJnADSYNSTCR	<ADCJn_base>+610_H	*1, *2	—
A/D timer synchronization start control register	ADCJnADTSYNSTCR	<ADCJn_base>+614_H	*1, *2	—
A/D halt register	ADCJnADHALTR	<ADCJn_base>+640_H	*1	—
A/D control register 1	ADCJnADCR1	<ADCJn_base>+644_H	*1	—
A/D control register 2	ADCJnADCR2	<ADCJn_base>+648_H	*1	—
Sampling Control Register	ADCJnSMPCR	<ADCJn_base>+$64C_H$	*1	—
MPX current control register	ADCJnMPXCURCR	<ADCJn_base>+650_H	*1	—
MPX interrupt enable register	ADCJnMPXINTER	<ADCJn_base>+654_H	*1	—
MPX current register 1	ADCJnMPXCURR1	<ADCJn_base>+658_H	*1	—
MPX current register 2	ADCJnMPXCURR2	<ADCJn_base>+$65C_H$	*1	—
MPX command information register	ADCJnMPXCMDR	<ADCJn_base>+660_H	*1	—
GTM entry scan group enable register	ADCJnGTMENTSGER	<ADCJn_base>+670_H	*1, *3	—
A/D conversion monitor virtual channel pointer 0	ADCJnADENDP0	<ADCJn_base>+674_H	*1	—
A/D conversion monitor virtual channel pointer 1	ADCJnADENDP1	<ADCJn_base>+678_H	*1	—
A/D conversion monitor virtual channel pointer 2	ADCJnADENDP2	<ADCJn_base>+$67C_H$	*1	—
A/D conversion monitor virtual channel pointer 3	ADCJnADENDP3	<ADCJn_base>+680_H	*1	—
A/D conversion monitor virtual channel pointer 4	ADCJnADENDP4	<ADCJn_base>+684_H	*1	—
T&H Sampling Start Control Register	ADCJnTHSMPSTCR	<ADCJn_base>+690_H	*1, *3	—
T&H Stop Control Register	ADCJnTHSTPCR	<ADCJn_base>+694_H	*1, *3	—
T&H Control Register	ADCJnTHCR	<ADCJn_base>+698_H	*1, *3	—
T&H Group A hold start control register	ADCJnTHAHLDSTCR	<ADCJn_base>+$6A0_H$	*1, *3	—
T&H Enable Register	ADCJnTHER	<ADCJn_base>+$6B4_H$	*1, *3	—
T&H Group A Control Register	ADCJnTHACR	<ADCJn_base>+$6C0_H$	*1, *3	—
Wait setting register 0	ADCJnWAITTR0	<ADCJn_base>+700_H	*1	—
Wait setting register 1	ADCJnWAITTR1	<ADCJn_base>+704_H	*1	—
Wait setting register 2	ADCJnWAITTR2	<ADCJn_base>+708_H	*1	—
Wait setting register 3	ADCJnWAITTR3	<ADCJn_base>+$70C_H$	*1	—
Wait setting register 4	ADCJnWAITTR4	<ADCJn_base>+710_H	*1	—
Wait setting register 5	ADCJnWAITTR5	<ADCJn_base>+714_H	*1	—
Wait setting register 6	ADCJnWAITTR6	<ADCJn_base>+718_H	*1	—
Wait setting register 7	ADCJnWAITTR7	<ADCJn_base>+$71C_H$	*1	—
Voltage Monitoring Voltage Divider Control Register 1	ADCJnVMONVDCR1	<ADCJn_base>+740_H	*1, *2	—
Voltage Monitoring Voltage Divider Control Register 2	ADCJnVMONVDCR2	<ADCJn_base>+744_H	*1, *2	—
Pin-level self-diagnosis control register	ADCJnTDCR	<ADCJn_base>+760_H	*1	—
Wiring-break detection control register	ADCJnODCR	<ADCJn_base>+764_H	*1	—
Safety control register	ADCJnSFTCR	<ADCJn_base>+770_H	*1	—
Trigger overlap check control register	ADCJnTOCCR	<ADCJn_base>+774_H	*1	—
Trigger overlap check error status register	ADCJnTOCER	<ADCJn_base>+790_H	*1	—
Overwrite error status register	ADCJnOWER	<ADCJn_base>+$79C_H$	*1	—

Table 43.14 List of Registers (3/3)

Register Name	Symbol	Address	Access Protection	
			PBG	Other
Parity error status register	ADCJnPER	<ADCJn_base>+$7A0_H$	*1	—
ID Error Register	ADCJnIDER	<ADCJn_base>+$7A4_H$	*1	—
Error clear register	ADCJnECR	<ADCJn_base>+$7A8_H$	*1	—
Upper/Lower Limit Check Table Register 0	ADCJnVCULLMTBR0	<ADCJn_base>+800_H	*1	—
Upper/Lower Limit Check Table Register 1	ADCJnVCULLMTBR1	<ADCJn_base>+804_H	*1	—
Upper/Lower Limit Check Table Register 2	ADCJnVCULLMTBR2	<ADCJn_base>+808_H	*1	—
Upper/Lower Limit Check Table Register 3	ADCJnVCULLMTBR3	<ADCJn_base>+$80C_H$	*1	—
Upper/Lower Limit Check Table Register 4	ADCJnVCULLMTBR4	<ADCJn_base>+810_H	*1	—
Upper/Lower Limit Check Table Register 5	ADCJnVCULLMTBR5	<ADCJn_base>+814_H	*1	—
Upper/Lower Limit Check Table Register 6	ADCJnVCULLMTBR6	<ADCJn_base>+818_H	*1	—
Upper/Lower Limit Check Table Register 7	ADCJnVCULLMTBR7	<ADCJn_base>+$81C_H$	*1	—
Upper/Lower Limit Check Interrupt Enable Register 1	ADCJnVCLMINTER1	<ADCJn_base>+840_H	*1	—
Upper/Lower Limit Check Interrupt Enable Register 2	ADCJnVCLMINTER2	<ADCJn_base>+844_H	*1	—
PWM-Diag Upper/Lower Limit Check Interrupt Enable Register	ADCJnPWVCLMINTER	<ADCJn_base>+$84C_H$	*1	—
Upper/Lower Limit Check Status Register 1	ADCJnVCLMSR1	<ADCJn_base>+860_H	*1	—
Upper/Lower Limit Check Status Register 2	ADCJnVCLMSR2	<ADCJn_base>+864_H	*1	—
PWM-Diag Upper/Lower Limit Check Status Register	ADCJnPWVCLMSR	<ADCJn_base>+$86C_H$	*1	—
Scan group 0 Upper/Lower Limit Check Status Register	ADCJnSGULCR0	<ADCJn_base>+880_H	*1	—
Scan group 1 Upper/Lower Limit Check Status Register	ADCJnSGULCR1	<ADCJn_base>+884_H	*1	—
Scan group 2 Upper/Lower Limit Check Status Register	ADCJnSGULCR2	<ADCJn_base>+888_H	*1	—
Scan group 3 Upper/Lower Limit Check Status Register	ADCJnSGULCR3	<ADCJn_base>+$88C_H$	*1	—
Scan group 4 Upper/Lower Limit Check Status Register	ADCJnSGULCR4	<ADCJn_base>+890_H	*1	—
Upper/Lower Limit Check Status Clear Register 1	ADCJnVCLMSCR1	<ADCJn_base>+$8A0_H$	*1	—
Upper/Lower Limit Check Status Clear Register 2	ADCJnVCLMSCR2	<ADCJn_base>+$8A4_H$	*1	—
PWM-Diag Upper/Lower Limit Check Status Clear Register	ADCJnPWVCLMSCR	<ADCJn_base>+$8AC_H$	*1	—
Scan Group Upper/Lower Limit Check Status Clear Register	ADCJnSGULCCR	<ADCJn_base>+$8B4_H$	*1	—
Virtual Channel Upper/Lower Limit Check Status Clear Register	ADCJnVCLMASCR	<ADCJn_base>+$8B8_H$	*1	—
Scan Group x Start Trigger Control Register	ADCJnSGTSELx	<ADCJ2_SELB_base> + x × 04_H	PBG20#9 *4	—

Note 1. n = 0: PBG6L0#4
n = 1: PBG70#2
n = 2: PBG20#9

Note 2. These registers are supported for ADCJ0 only.

Note 3. These registers are supported for ADCJ0 & ADCJ1 only.

Note 4. This register is supported for ADCJ2 only.

NOTE

For the number of units and indices, see **Section 43.1.1, Number of Units**.

43.3.2 ADC Specific Registers (Virtual Channel)

43.3.2.1 ADCJnVCRj — Virtual Channel Register j

ADCJnVCRj is a readable and writable register to make settings for each virtual channel.

Access: This register can be read or written in 32-bit units.

Address: <ADCJn_base> + j × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VCULLMTBS[3:0]				WTTS[3:0]				—	—	—	GTM ENT	GTMTAG[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CNVCLS[3:0]				MPXV[2:0]			ADIE	—	GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.15 ADCJnVCRj Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	VCULLMTBS [3:0]	Upper/Lower Limit Check Table Register Select Select an upper/lower limit check table register to be compared. 0_H: Disabled 1_H: ADCJnVCULLMTBR0 is chosen 2_H: ADCJnVCULLMTBR1 is chosen 3_H: ADCJnVCULLMTBR2 is chosen 4_H: ADCJnVCULLMTBR3 is chosen 5_H: ADCJnVCULLMTBR4 is chosen 6_H: ADCJnVCULLMTBR5 is chosen 7_H: ADCJnVCULLMTBR6 is chosen 8_H: ADCJnVCULLMTBR7 is chosen Other than above: Setting prohibited (same as 0_H).
27 to 24	WTTS[3:0]	Wait Time Table Select 0_H: Disabled 1_H: ADCJnWAITTR0 is chosen. 2_H: ADCJnWAITTR1 is chosen. 3_H: ADCJnWAITTR2 is chosen. 4_H: ADCJnWAITTR3 is chosen. 5_H: ADCJnWAITTR4 is chosen. 6_H: ADCJnWAITTR5 is chosen. 7_H: ADCJnWAITTR6 is chosen. 8_H: ADCJnWAITTR7 is chosen. Other than above: Setting prohibited (same as 0_H).
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20	GTMENT	GTM Entry 0: Entry disabled 1: Entry enabled Select whether to perform entry to the GTM. Entry is available only in scan groups enabled by GTMENTSGxE in ADCJnGTMENTSGER.
19 to 16	GTMTAG[3:0]	GTM-TAG When entry is requested to GTM, entry is performed to the channel of GTM for which the value of TAG is same as GTMTAG[3:0]. If multiple channels match, entry is performed t to them.

Table 43.15 ADCJnVCRj Register Contents (2/3)

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 11	CNVCLS[3:0]	Conversion Type 0_H: Normal A/D conversion 1_H: Hold value A/D conversion 2_H: Normal A/D conversion at extended sampling cycle 3_H: ADcore self-diagnosis A/D conversion 4_H: Addition mode A/D conversion 5_H: MPX normal A/D conversion 6_H: MPX addition mode A/D conversion 7_H: Pin level self-diagnosis A/D conversion 8_H: A/D conversion in wiring-break detection mode 1 9_H: A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) A_H: A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) B_H: Self-diagnosis A/D conversion in wiring-break detection mode 1 C_H: Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) D_H: Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) E_H: A/D conversion of A/D conversion data path diagnosis (ADVAL mode) F_H: Reserved
10 to 8	MPXV[2:0]	MPX channel setting These bits are used to set the channel of external MPX to be transferred to the external analog multiplexer.
7	ADIE	Virtual Channel End Interrupt Enable 0: INT_ADx is not output at the end of virtual channel n in SGx. 1: INT_ADx is output at the end of virtual channel n in SGx. For details of the relationship between ADIE in ADCJnSGCRx and ADIE in ADCJnVCRj, see **Section 43.4.19.1, Scan End Interrupt Request**.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 43.15 ADCJnVCRj Register Contents (3/3)

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	General Control Set A/D conversion for each conversion mode. Normal A/D conversion (CNVCLS[3:0] = 0_H) GCTRL[5:2]: Set the physical channel group number. 0_H: Group number of physical channels ANn00 to ANn03 1_H: Group number of physical channels ANn10 to ANn13 : 7_H: Group number of physical channels ANn70 to ANn73 8_H: Group number of physical channels ANn80 to ANn81 GCTRL[1:0]: Set the physical sub channel number. 0_H: Sub channel 0 of the physical channel group specified by GCTRL[5:2] 1_H: Sub channel 1 of the physical channel group specified by GCTRL[5:2] 2_H: Sub channel 2 of the physical channel group specified by GCTRL[5:2] 3_H: Sub channel 3 of the physical channel group specified by GCTRL[5:2] Example of setting: To specify ANn10, set GCTRL[5:2]: = 1_H and GCTRL[1:0] = 0_H. To specify ANn72, set GCTRL[5:2]: = 7_H and GCTRL[1:0] = 2_H. Hold value A/D conversion (CNVCLS[3:0] = 1_H) GCTRL[5:3]: Set 000_B. GCTRL[2:0]: Set the T&H channel number. 0_H: The T&H0 value is AD converted. 1_H: The T&H1 value is AD converted. 2_H: The T&H2 value is AD converted. 3_H: The T&H3 value is AD converted. Others: Setting prohibited. Normal A/D conversion with extended Sampling time (CNVCLS[3:0] = 2_H) Make the same settings as normal A/D conversion (CNVCLS[3:0] = 0_H). ADcore self-diagnosis A/D conversion (CNVCLS[3:0] = 3_H) GCTRL[5]: Set 0. GCTRL[4:0]: Set the ADcore self-diagnosis voltage level. 10_H: AnVREFH × 1 $0C_H$: AnVREFH × 3/4 08_H: AnVREFH × 1/2 04_H: AnVREFH × 1/4 00_H: AnVREFH × 0 Other than above: Setting prohibited When the CNVCLS[3:0] value is set to the following value (CNVCLS[3:0] = 4_H to D_H), make the same settings as normal A/D conversion (CNVCLS[3:0] = 0_H). For addition mode A/D conversion (CNVCLS[3:0] = 4_H) For MPX normal A/D conversion (CNVCLS[3:0] = 5_H) For MPX addition mode A/D conversion (CNVCLS[3:0] = 6_H) For pin level self-diagnosis A/D conversion (CNVCLS[2:0] = 7_H) For A/D conversion in wiring-break detection mode 1 (CNVCLS[3:0] = 8_H) For A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) (CNVCLS[3:0] = 9_H) For A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) (CNVCLS[3:0] = A_H) For self-diagnosis A/D conversion in wiring-break detection mode 1 (CNVCLS[3:0] = B_H) For self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) (CNVCLS[3:0] = C_H) For self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) (CNVCLS[3:0] = D_H) A/D conversion data path diagnosis A/D conversion (ADVAL mode) (CNVCLS[3:0] = E_H) 00: Fix the A/D conversion data from ADCore to AAA_H. 01: Fix the A/D conversion data from ADCore to 555_H. Other than above: Setting prohibited.

CAUTIONS

1. To prevent a malfunction, this register has restrictions on updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".
2. When not used MPX normal A/D conversion or MPX addition mode A/D conversion, be sure to set MPXV[2:0] = 000_B.
3. When using MPX normal A/D conversion or MPX addition mode A/D conversion, be sure to insert a wait.
4. It is prohibited to set the value corresponding to no implemented physical channel or T&H channel to GCTRL[5:0]. In this case, A/D conversion result of these channels is not guaranteed because A/D conversion is executed with no selecting I/O or T&H.

43.3.2.2 ADCJnPWDVCR — PWM-Diag Virtual Channel Register

ADCJnPWDVCR is a read-only register to control PWM-Diag virtual channels.

Access: This register can be read in 32-bit units.

Address: <ADCJn_base> + 180_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	WTTS[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE	MPXV[2:0]			VCULLMTBS[3:0]				—	—	GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.16 ADCJnPWDVCR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 16	WTTS[3:0]	Wait Time Table Select 0_H: Disabled 1_H: ADCJnWAITTR0 is chosen. 2_H: ADCJnWAITTR1 is chosen. 3_H: ADCJnWAITTR2 is chosen. 4_H: ADCJnWAITTR3 is chosen. 5_H: ADCJnWAITTR4 is chosen. 6_H: ADCJnWAITTR5 is chosen. 7_H: ADCJnWAITTR6 is chosen. 8_H: ADCJnWAITTR7 is chosen. Other than above: Setting prohibited (same as 0_H).
15	MPXE	MPX Enable When operating this bit, set MPXE in the register of the PWSD module. 0: Normal A/D conversion is performed. The MPXV[2:0] value is not transferred to the output pin PVCR_MUXCUR[2:0]. No wait is inserted before A/D conversion starts. 1: MPX normal A/D conversion is performed. The MPXV[2:0] value is transferred to the output pin PVCR_MUXCUR[2:0]. A wait is inserted before A/D conversion starts.
14 to 12	MPXV[2:0]	MPX Channel Value When operating these bits, set MPXV[2:0] in the register of the PWSD module. Use this bit to select an MPX channel of the external analog multiplexer.
11 to 8	VCULLMTBS [3:0]	Upper/Lower Limit Check Table Register Select When operating these bits, set VCULLMTBS[3:0] in the register of the PWSD module. Select the upper/lower limit check table register to be compared. 0_H: Disabled 1_H: ADCJnVCULLMTBR0 is chosen. 2_H: ADCJnVCULLMTBR1 is chosen. 3_H: ADCJnVCULLMTBR2 is chosen. 4_H: ADCJnVCULLMTBR3 is chosen. 5_H: ADCJnVCULLMTBR4 is chosen. 6_H: ADCJnVCULLMTBR5 is chosen. 7_H: ADCJnVCULLMTBR6 is chosen. 8_H: ADCJnVCULLMTBR7 is chosen. Other than above: Setting prohibited (same as 0_H).
7, 6	Reserved	When read, the value after reset is returned.

Table 43.16 ADCJnPWDVCR Register Contents (2/2)

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	When operating these bits, set GCTRL[5:0] in the register of the PWSD module. GCTRL[5:2]: Set the physical channel group number. GCTRL[1:0]: Set the physical sub channel number. Example of setting: To specify ANn10, set GCTRL[5:2]: = 1_H and GCTRL[1:0] = 0_H. To specify ANn81, set GCTRL[5:2]: = 8_H and GCTRL[1:0] = 1_H.

CAUTIONS

1. When using MPX normal A/D conversion, be sure to insert a wait.
2. When not used MPX normal A/D conversion or MPX addition mode A/D conversion, be sure to set MPXV[2:0] to 000_B.
3. It is prohibited to set the value corresponding to no implemented physical channel to GCTRL[5:0]. In this case, A/D conversion result of these channels is not guaranteed because A/D conversion is executed with no selecting I/O.

NOTES

1. VCULLMTBS[3:0] in ADCJnPWDVCR and VCULLMTBS[3:0] in ADCJnVCRj are the same function.
2. PVCR_VALUE[17:0] is allocated to the following bits.
   PVCR_VALUE[17:14] = WTTS[3:0]
   PVCR_VALUE[13] = MPXE
   PVCR_VALUE[12:10] = MPXV[2:0]
   PVCR_VALUE[9:6] = VCULLMTBS[3:0]
   PVCR_VALUE[5:0] = GCTRL[5:0]

43.3.2.3 ADCJnDRj— Data Register j

ADCJnDRj is a read-only register to store A/D conversion value.

Access: This register is a read-only register that can be read in 32- or 16-bit units.

Address: <ADCJn_base> + $1A0_H$ + j × 2_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRj[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.17 ADCJnDRj Register Contents

Bit Position	Bit Name	Function
15 to 0	DRj[15:0]	Data Register These bits store A/D conversion result data ADCJnVCRj.

Data format of ADCJnDRj depends on the settings of data format bit (ADCJnADCR2.DFMT[2:0]). Regardless of the setting of the data format, data format translated to GTM is 12-bit (resolution) signed fixed-point format. Refer to **Section 43.2.3.4, Data Format** for details.

When the read and clear enable bit is valid (ADCJnSFTCR.RDCLRE = 1), ADCJnDRj is cleared by reading ADCJnDRj. And when ADCJnDRj is read, both ADCJnDRj and ADCJnDRj+1 are read at the same time because ADCJnDRj and ADCJnDRj+1 are stored in 32-bit border.

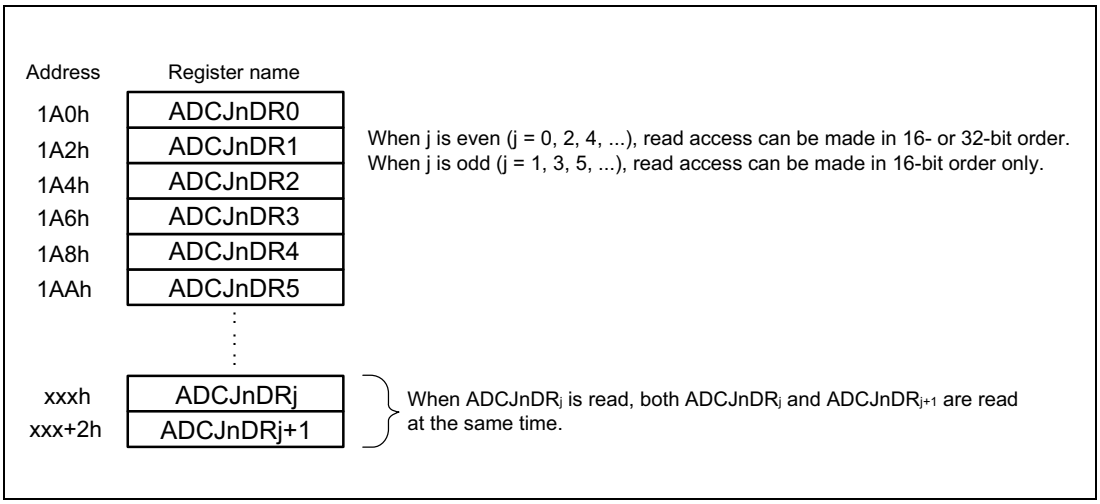


Figure 43.5 ADCJnDRj Read Specifications

When whether ADCJnDRj or ADCJnDRj+1 is read by CPU order, the other ADCJnDRj or ADCJnDRj+1 is cleared. So, set the read and clear enable bit invalid (ADCJnSFTCR.RDCLRE = 0).

And when whether ADCJnDRj or ADCJnDRj+1 is read by CPU order, it is prohibited to use the overwrite function*1. So, disable the overwrite function*2.

Note 1. Refer to NOTE 6 in **Section 43.3.2.5, ADCJnDIRj — Data Supplementary Information Register j** for details.

Note 2. When all of the following three conditions are met, the overwrite function is disabled.
(1) Do not compare the expected value of the WFLAG bit in ADCJnDIRj.
(2) Do not judge an error using the OWE bit in ADCJnOWER.
(3) Set the OWEIE bit in ADCJnSFTCR to 0 (disabled).

CAUTION

When the effective bit range of data is changed without reading ADCJnDRj after storing A/D conversion result, pseudo parity error may occur. Refer to CAUTION of Section 43.4.19.3, Parity Error Interrupt Request for details.

NOTES

1. When the read and clear enable bit is valid (ADCJnSFTCR.RDCLRE = 1), and when either ADCJnDRj or ADCJnDRj+1 is read as 16-bit data by a CPU instruction, both ADCJnDRj and ADCJnDRj+1 are cleared at the same time.
2. When the read and clear enable bit is valid (ADCJnSFTCR.RDCLRE = 1), ADCJnDRj is cleared by reading ADCJnDIRj.

When read and clear is valid (ADCJnSFTCR.RDCLRE = 1), set virtual channel number assignment constraint between scan groups to recommended settings 1 and 2 of **Figure 43.6**.

When virtual channel numbers of two scan groups are consecutively like a prohibited setting of **Figure 43.6**, by reading ADCJnDRj of one scan group target, ADCJnDRj+1 of the other scan group target is cleared, too. So this setting is prohibited.

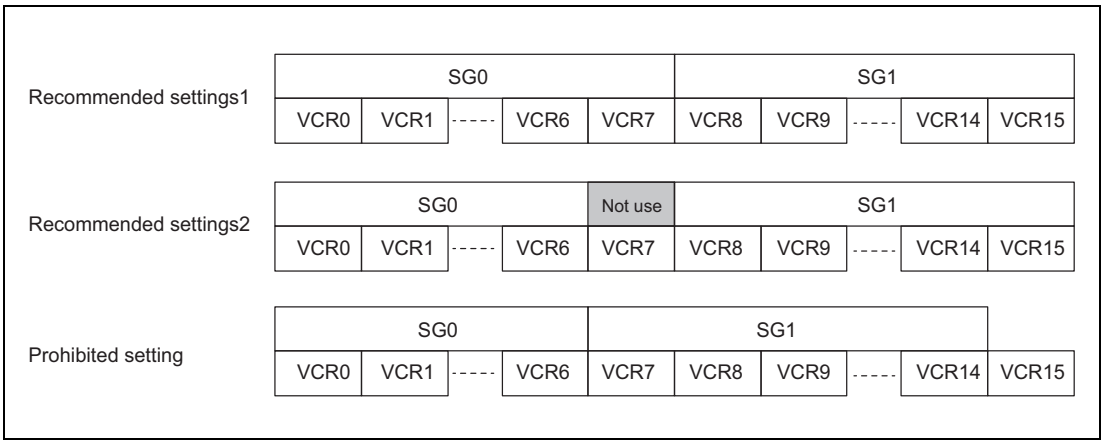


Figure 43.6 Restriction of Virtual channel number assignment constraint between scan groups

43.3.2.4 ADCJnPWDDR — PWM-Diag Data Register

ADCJnPWDDR is a read-only register to store A/D conversion value of PWM-Diag.

No overwrite error of ADCJnPWDDR is generated.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <ADCJn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.18 ADCJnPWDDR Register Contents

Bit Position	Bit Name	Function
15 to 0	PWDDR[15:0]	PWM-Diag Data Register These bits are used to store the A/D conversion result data for the PWM-Diag.

CAUTION

When the effective bit range of data is changed without reading ADCJnPWDDR after storing A/D conversion result, pseudo parity error may occur. Refer to CAUTION of **Section 43.4.19.3, Parity Error Interrupt Request** for details.

NOTES

1. The ADCJnPWDDR format varies with the setting of data format (ADCJnADCR2.DFMT[2:0]). For details, check **(1), Data Formats of Data Registers**.
2. When the read and clear enable bit is valid (ADCJnSFTCR.RDCLRE = 1), ADCJnPWDDR is cleared by reading ADCJnPWDDIR.

43.3.2.5 ADCJnDIRj — Data Supplementary Information Register j

ADCJnDIRj is a read-only register to store supplementary information of ADCJnDRj and A/D converted value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + 280_H + j × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE	MPXV[2:0]			—	IDEF	WFLAG	PRTY	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRj[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.19 ADCJnDIRj Register Contents (1/2)

Bit Position	Bit Name	Function
31	MPXE	MPX Enable Flag 0: The MPX function is not used. 1: The MPX function is used.
30 to 28	MPXV[2:0]	MPX Channel Value These bits are used to store the MPX channel number transferred to the external analog multiplexer.
27	Reserved	When read, the value after reset is returned.
26	IDEF	ID Error 0: There is an error. 1: There is no error.
25	WFLAG	Write Flag Setting and clearing conditions 0: Clearing conditions Reading ADCJnDRj or ADCJnDIRj 1: Setting condition Storing A/D conversion value in ADCJnDRj
24	PRTY	Even parity bit of DRj[15:0], ID[5:0], IDEF, MPXV[2:0], and MPXE
23, 22	Reserved	When read, the value after reset is returned.

Table 43.19 ADCJnDIRj Register Contents (2/2)

Bit Position	Bit Name	Function
21 to 16	ID[5:0]	ID Information The channel number that A/D conversion has been executed is stored. When normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 0_H) is performed, the ID[5:2] value means physical channel group and the ID[1:0] value means physical sub channel. When hold value A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 1_H) is performed, the T&H whose hold value was A/D converted is stored in ID[2:0]. 0_H: The hold value of T&H0 is A/D converted. 1_H: The hold value of T&H1 is A/D converted. 2_H: The hold value of T&H2 is A/D converted. 3_H: The hold value of T&H3 is A/D converted. The ID bits value other than above is always 0. When normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 2_H) at the extended sampling cycle is performed, the same information as normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 0_H) is stored. When ADcore self-diagnosis A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 3_H) is performed, the ADcore self-diagnosis level is stored in ID[4:0]. 10_H: AnVREFH × 1 $0C_H$: AnVREFH × 3/4 08_H: AnVREFH × 1/2 04_H: AnVREFH × 1/4 00_H: AnVREFH × 0 The ID bits value other than above is always 0. When the ADCJnVCRj.CNVCLS[3:0] value is set to the following value (ADCJnVCRj.CNVCLS[3:0] = 4_H to E_H), the same information as normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 0_H) is stored. For addition mode A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 4_H) For MPX normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 5_H) For MPX addition mode A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 6_H) For pin level self-diagnosis A/D conversion (ADCJnVCRj.CNVCLS[2:0] = 7_H) For A/D conversion in wiring-break detection mode 1 (ADCJnVCRj.CNVCLS[3:0] = 8_H) For A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) (ADCJnVCRj.CNVCLS[3:0] = 9_H) For A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) (ADCJnVCRj.CNVCLS[3:0] = A_H) For self-diagnosis A/D conversion in wiring-break detection mode 1 (ADCJnVCRj.CNVCLS[3:0] = B_H) For self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down) (ADCJnVCRj.CNVCLS[3:0] = C_H) For self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up) (ADCJnVCRj.CNVCLS[3:0] = D_H) When A/D conversion data path diagnosis (ADVAL mode) (ADCJnVCRj.CNVCLS[3:0] = E_H) is performed, the executed GCTRL[5:0] value in ADCJnVCRj is stored.
15 to 0	DRj[15:0]	Data Register It is ADCJnDRj itself. Refer to **Section 43.3.2.3, ADCJnDRj— Data Register j** for details.

CAUTION

When the effective bit range of data is changed without reading ADCJnDIRj after storing A/D conversion result, pseudo parity error may occur. Refer to CAUTION of **Section 43.4.19.3, Parity Error Interrupt Request** for details.

NOTES

1. When read & clear is valid (ADCJnSFTCR.RDCLRE = 1), value of IDEF bit is changed by the following condition.

Condition	Value of IDEF
Initial state (reset value)	0
Reading ADCJnDRj or reading ADCJnDIRj	
Detection of ID error	
No detection of ID error	1

When read & clear is invalid (ADCJnSFTCR.RDCLRE = 0), value of IDEF bit is changed by the following condition.

Condition	Value of IDEF
Initial state (reset value)	0
Detection of ID error	
No detection of ID error	1

2. The MPXE bit is set to 1 when the MPX is in use (ADCJnVCRj.CNVCLS = 5_H or 6_H) in the executed virtual channel.
3. The MPXV bit of the executed virtual channel is stored in the MPXV[2:0] bits.
4. When reading ADCJnDRj or ADCJnDIRj, ADCJnDIRj is cleared by the following condition.

Condition	ADCJnDIRj	
	Cleared	Not cleared
Read & clear is enabled (ADCJnSFTCR.RDCLRE = 1)	MPXE, MPXV, IDEF, WFLAG, PRTY, ID, DRj	Nothing
Read & clear is disabled (ADCJnSFTCR.RDCLRE = 0)	WFLAG	MPXE, MPXV, IDEF, PRTY, ID, DRj

5. DRj bit is ADCJnDRj register itself. So, when the read and clear is valid (ADCJnSFTCR.RDCLRE = 1), all bits of ADCJnDIRj are cleared by read ADCJnDRj register.
6. The overwrite is the function that is independent of Read & Clear. When either ADCJnDRj or ADCJnDRj+1 is read by CPU order, the other WFLAG bit register is also cleared. Therefore, when either ADCJnDRj or ADCJnDRj+1 is read by CPU order, it is prohibited to use the overwrite function.
7. When the read & clear is valid (ADCJnSFTCR.RDCLRE = 1), and when either ADCJnDRj or ADCJnDRj+1 is read by CPU order, both all bits of ADCJnDIRj and ADCJnDIRj+1 are cleared at the same time.

43.3.2.6 ADCJnPWDDIR — PWM-Diag Data Supplementary Information Register

ADCJnPWDDIR is a read-only register to store supplementary information of ADCJnPWDDR and A/D conversion values.

PWDDR in this bit is ADCJnPWDDR.PWDDR[15:0] itself. Supplementary information of A/D conversion result is transferred from MPX enable (MPXE), MPX channel value (MPXV[2:0]) and physical channel number (ID[5:0]) in ADCJnPWDVCR.

No overwrite error of ADCJnPWDDIR is generated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + 400_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE	MPXV[2:0]			—	IDEF	WFLAG	PRTY	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.20 ADCJnPWDDIR Register Contents

Bit Position	Bit Name	Function
31	MPXE	MPX Enable Flag 0: The MPX function is not used. 1: The MPX function is used.
30 to 28	MPXV[2:0]	MPX Channel Value These bits are used to store the MPX channel number transferred to the external analog multiplexer.
27	Reserved	When read, the value after reset is returned.
26	IDEF	ID Error 0: There is an error. 1: There is no error.
25	WFLAG	Write Flag Setting and clearing conditions 0: Clearing conditions Reading ADCJnPWDDR or ADCJnPWDDIR 1: Setting condition Storing A/D conversion value in ADCJnPWDDR
24	PRTY	Even parity bit of PWDDR[15:0], ID[5:0], IDEF, MPXV[2:0], and MPXE
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	ID Information The value of ADCJnPWDVCR.GCTRL that has been executed is stored.
15 to 0	PWDDR[15:0]	PWM-Diag Data Register It is ADCJnPWDDR itself. Refer to **Section 43.3.2.4, ADCJnPWDDR — PWM-Diag Data Register** for details.

CAUTION

When the effective bit range of data is changed without reading ADCJnPWDDIR after storing A/D conversion result, pseudo parity error may occur. Refer to CAUTION of **Section 43.4.19.3, Parity Error Interrupt Request** for details.

NOTES

1. When read & clear is valid (ADCJnSFTCR.RDCLRE = 1), value of IDEF bit is changed by the following condition.

Condition	Value of IDEF
Initial state (reset value)	0
Reading ADCJnPWDDR or reading ADCJnPWWDIR	
Detection of ID error	
No detection of ID error	1

When read & clear is invalid (ADCJnSFTCR.RDCLRE = 0), value of IDEF bit is changed by the following condition.

Condition	Value of IDEF
Initial state (reset value)	0
Detection of ID error	
No detection of ID error	1

2. The MPXE bit is set to 1 when the MPX is in use (MPXE = 1) in the executed ADCJnPWDVCR.
3. The MPXV bit value in ADCJnPWDVCR is stored in the MPXV[2:0] bits.
4. When reading ADCJnPWDDR or ADCJnPWDDIR, ADCJnPWDDIR is cleared by the following condition.

Condition	ADCJnPWDDIR	
	Cleared	Not cleared
Read & clear is enabled (ADCJnSFTCR.RDCLRE = 1)	MPXE, MPXV, IDEF, WFLAG, PRTY, ID, PWDDR	Nothing
Read & clear is disabled (ADCJnSFTCR.RDCLRE = 0)	WFLAG	MPXE, MPXV, IDEF, PRTY, ID, PWDDR

5. PWDDR bit is ADCJnPWDDR register itself. So, when the read and clear is valid (ADCJnSFTCR.RDCLRE = 1), all bits of ADCJnPWDDIR is cleared by read ADCJnPWDDR register.

43.3.3 ADC Scan Group Specific Registers

43.3.3.1 ADCJnSGCRx — Scan Group x Control Register

ADCJnSGCRx is a readable and writable register to control scan group x.

Access: This register can be read or written in 8-bit units.

Address: ADCJnSGCR0: <ADCJn_base> + 450_H
ADCJnSGCR1: <ADCJn_base> + 490_H
ADCJnSGCR2: <ADCJn_base> + $4D0_H$
ADCJnSGCR3: <ADCJn_base> + 510_H
ADCJnSGCR4: <ADCJn_base> + 550_H

Value after reset: 00_H

When x = 0 to 2

Bit	7	6	5	4	3	2	1	0
	—	ADSTARTE	SCANMD	ADIE	—	—	—	TRGMD[0]
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W

Table 43.21 ADCJnSGCRx Register Contents (x = 0 to 2)

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	ADSTARTE	A/D Conversion Simultaneous Start Enable This bit is used to set whether to enable or disable the A/D conversion simultaneous start bit (ADCJnADSYNSTCR.ADSTART). 0: Disabled 1: Enabled
5	SCANMD	Scan Mode This bit is used to set scan mode of scan group x. 0: Multicycle scan mode 1: Continuous scan mode
4	ADIE	Scan End Interrupt Enable This bit is used to set whether to enable or disable output of the end interrupt signal (INT_ADx) of scan group x. 0: Output disabled 1: Output enabled For details of relationship between ADIE in ADCJnVCRj and ADIE in ADCJnSGCRx, see “**Section 43.4.19.1, Scan End Interrupt Request**”.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD[0]	Trigger Mode This bit is used to set whether to enable or disable hardware triggers of scan group x. 0: Disabled 1: Enabled

TRGMD[0] is a enable bit that controls the start of scan group x by hardware trigger.

TRGMD[1:0]	Scan group x (x = 0, 1, 2)			
	Input			Output
	Hardware trigger (sgx_trg)	Software trigger (ADCJnSGSTCRx.SGST)	A/D conversion simultaneous start trigger (ADCJnADSYNSTCR.ADSTART)	Scan group Start
0_H	—	√	√	valid
1_H	√	√	√	valid

When x = 3, 4

Bit	7	6	5	4	3	2	1	0
	ADTSTARTE	ADSTARTE	SCANMD	ADIE	—	—	TRGMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 43.22 ADCJnSGCRx Register Contents (x = 3, 4)

Bit Position	Bit Name	Function
7	ADTSTARTE	A/D Timer Simultaneous Start Enable This bit is used to set whether to enable or disable the A/D timer simultaneous start bit (ADCJnADTSYNSTCR.ADTSTART). 0: Disabled 1: Enabled
6	ADSTARTE	A/D Conversion Simultaneous Start Enable This bit is used to set whether to enable or disable the A/D conversion simultaneous start bit (ADCJnADSYNSTCR.ADSTART). 0: Disabled 1: Enabled
5	SCANMD	Scan Mode This bit is used to set scan mode of scan group x. 0: Multicycle scan mode 1: Continuous scan mode
4	ADIE	Scan End Interrupt Enable This bit is used to set whether to enable or disable output of the end interrupt signal of scan group x. 0: Output disabled 1: Output enabled
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	TRGMD[1]	A/D Timer Trigger x Enable This bit is used to set whether to enable or disable A/D timer trigger x. 0: A/D timer trigger x is not requested 1: It is permitted for scan group x to start by A/D timer trigger x. * A/D timer trigger is internal trigger of ADCJ that activates scan group x at initial phase and every setting period of A/D timer x.
0	TRGMD[0]	Trigger Mode This bit is used to set whether to enable or disable hardware triggers of scan group x. 0: Disabled 1: Enabled

TRGMD[1:0] is a enable bit that controls both the start of scan group x and the start of A/D timer x by hardware trigger

- A/D timer x
  When TRGMD[1] is set to 1, A/D timer trigger x that A/D timer x outputs is valid. A/D timer x outputs A/D timer trigger x after initial phase or every period. And scan group x starts by A/D timer trigger x.
  When A/D timer x is started by hardware trigger, set TRGMD[1:0] to 3_H.

TRGMD [1:0]	A/D timer x (x = 3, 4)				
	Input			Output	
	Hardware trigger (sgx_trg)	Software trigger (ADCJnADTSTCRx.ADTST)	A/D timer simultaneous start trigger (ADCJnADTSYNSTCR.ADTSTART)	A/D timer Start	A/D timer trigger x
0_H	—	√	√	valid	Not output
1_H	—	√	√	valid	Not output
2_H	—	√	√	valid	Output
3_H	√	√	√	valid	Output

- Scan group x
  When scan group x is started by hardware trigger, set TRGMD[1:0] to 1_H.
  When A/D timer x is started by hardware trigger, set TRGMD[1:0] to 3_H.

TRGMD [1:0]	Scan group x (x = 3, 4)				
	Input				Output
	Hardware trigger (sgx_trg)	Software trigger (ADCJnADTSTCRx.ADTST)	A/D timer simultaneous start trigger (ADCJnADTSYNSTCR.ADTSTART)	A/D timer trigger x	Scan group x start
0_H	—	√	√	—	valid
1_H	√	√	√	—	valid
2_H	—	√	√	√	valid
3_H	—	√	√	√	valid

NOTES

1. x = 0 to 4
2. x = 0 to 2
   When ADCJnSGCRx.TRGMD is set to 1_H, it is enable to start scan group x by hardware trigger.
3. x = 3, 4
   When ADCJnSGCRx.TRGMD[1:0] is set to 1_H, it is enable to start scan group x by hardware trigger. When ADCJnSGCRx.TRGMD[1:0] is set to 3_H, it is enable to start A/D timer x by hardware trigger. And scan group x is started by A/D timer x in period of initial phase and in every period of each period.

CAUTIONS

1. A/D conversion start trigger input of the same scan group x occurred during A/D conversion of scan group x is ignored.
2. When continuous scan mode is set (SCANMD = 1) for any scan group, scan groups of lower priority than that of the scan group are disabled.
3. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**". However, clearing the ADIE and TRGMD bits to 0 when stopping a scan group is excluded.

43.3.3.2 ADCJnSGVCPRx — Scan Group x Virtual Channel Pointer Register

ADCJnSGVCPRx is a readable and writable register to specify the starting pointer and end pointer of virtual channels.

Write this register with a 16-bit width only.

Access: This register can be read or written in 16-bit units.

Address: ADCJnSGVCPR0: <ADCJn_base> + 454_H
ADCJnSGVCPR1: <ADCJn_base> + 494_H
ADCJnSGVCPR2: <ADCJn_base> + $4D4_H$
ADCJnSGVCPR3: <ADCJn_base> + 514_H
ADCJnSGVCPR4: <ADCJn_base> + 554_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	VCEP[5:0]						—	—	VCSP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.23 ADCJnSGVCPRx Register Contents

Bit Position	Bit Name	Function
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	VCEP[5:0]	End Virtual Channel Pointer
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0]	Starting Virtual Channel Pointer

CAUTIONS

1. Do not set a value larger than the number of implemented virtual channels.
2. Set this register to satisfy ADCJnSGVCPRx.VCSP ≤ ADCJnSGVCPRx.VCEP.
3. Setting a virtual channel (ADCJnVCRj) for multiple scan groups x is prohibited by setting the starting pointer and end pointer.
4. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.

43.3.3.3 ADCJnSGMCYCRx — Scan Group x Multicycle Register

ADCJnSGMCYCRx is a readable and writable register to specify the scan count in multicycle scan mode.

Access: This register can be read or written in 8-bit units.

Address: ADCJnSGMCYCR0: <ADCJn_base> + 458_H
ADCJnSGMCYCR1: <ADCJn_base> + 498_H
ADCJnSGMCYCR2: <ADCJn_base> + $4D8_H$
ADCJnSGMCYCR3: <ADCJn_base> + 518_H
ADCJnSGMCYCR4: <ADCJn_base> + 558_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	MCYC[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.24 ADCJnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
7 to 0	MCYC[7:0]	Multicycle These bits are used to set the scan count in multicycle scan mode. Scan count = MCYC[7:0] + 1 When operating the PWM-Diag in SG4 with PWE in ADCJnPWDCR set to 1, ADCJnSMGCYCR4 can be set only to 00_H.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.3.4 ADCJnSGSTCRx — Scan Group x Start Control Register

ADCJnSGSTCRx is a write-only register to control start of scan group x. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: ADCJnSGSTCR0: <ADCJn_base> + 440_H
ADCJnSGSTCR1: <ADCJn_base> + 480_H
ADCJnSGSTCR2: <ADCJn_base> + $4C0_H$
ADCJnSGSTCR3: <ADCJn_base> + 500_H
ADCJnSGSTCR4: <ADCJn_base> + 540_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SGST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.25 ADCJnSGSTCRx Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SGST	Scan Group Start This software trigger is used to start A/D conversion of scan group x. 0: No function (Writing 0 to this bit is ignored.) 1: A/D conversion start Start A/D conversion when the scan group status (ADCJnSGSRx.SGACT) = 0.

43.3.3.5 ADCJnSGSTPCRx — Scan Group x Stop Control Register

ADCJnSGSTPCRx is a write-only register to control stop of scan group x. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: ADCJnSGSTPCR0: <ADCJn_base> + 444_H
ADCJnSGSTPCR1: <ADCJn_base> + 484_H
ADCJnSGSTPCR2: <ADCJn_base> + $4C4_H$
ADCJnSGSTPCR3: <ADCJn_base> + 504_H
ADCJnSGSTPCR4: <ADCJn_base> + 544_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SGSTP
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.26 ADCJnSGSTPCRx Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SGSTP	Scan Group Stop This software trigger is used to stop A/D conversion of scan group x. 0: No function (Writing 0 to this bit is ignored.) 1: A/D conversion stop Writing 1 to this bit is ignored when the scan group status (ADCJnSGSRx.SGACT) = 0.

43.3.3.6 ADCJnADTSTCRx — A/D Timer x Start Control Register

ADCJnADTSTCRx is a write-only register to control start of A/D timer x. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: ADCJnADTSTCR3: <ADCJn_base> + 508_H
ADCJnADTSTCR4: <ADCJn_base> + 548_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.27 ADCJnADTSTCRx Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ADTST	A/D Timer Start This software trigger is used to start A/D timer x. 0: No function (Writing 0 to this bit is ignored.) 1: A/D timer start. Start the A/D timer when the A/D timer status (ADCJnSGSRx.ADTACT) = 0.

NOTE

x = 3, 4

43.3.3.7 ADCJnADTENDCRx — A/D Timer x End Control Register

ADCJnADTENDCRx is a write-only register to control end of A/D timer x. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: ADCJnADTENDCR3: <ADCJn_base> + $50C_H$
ADCJnADTENDCR4: <ADCJn_base> + $54C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTEND
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.28 ADCJnADTENDCRx Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ADTEND	A/D Timer End This software trigger is used to terminate the A/D timer. 0: No function (Writing 0 to this bit is ignored.) 1: A/D timer end

NOTE

x = 3, 4

43.3.3.8 ADCJnADTIPRx — A/D Timer x Initial Phase Register

ADCJnADTIPRx is a readable and writable register to set the initial phase of A/D timer x.

Access: This register can be read or written in 32-bit units.

Address: ADCJnADTIPR3: <ADCJn_base> + 524_H
ADCJnADTIPR4: <ADCJn_base> + 564_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADTIP[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADTIP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.29 ADCJnADTIPRx Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	ADTIP[20:0]	A/D Timer Initial Phase These bits are used to set the initial phase of A/D timer x. Setting range: 00_0000_H to $1F\_FFFF_H$

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTES

1. For details, see "**Section 43.4.13, Example of A/D Timer Operation**".
2. x = 3, 4

43.3.3.9 ADCJnADTPRRx — A/D Timer x Cycle Register

ADCJnADTPRRx is a readable and writable register to set the cycle of A/D timer x.

Access: This register can be read or written in 32-bit units.

Address: ADCJnADTPRR3: <ADCJn_base> + 528_H
ADCJnADTPRR4: <ADCJn_base> + 568_H

Value after reset: 001F FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADTPR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADTPR[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.30 ADCJnADTPRRx Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	ADTPR[20:0]	A/D Timer Cycle These bits are used to set the cycle of A/D timer x. Setting range: 00_0000$_H$ to 1F_FFFF$_H$

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTES

1. For details, see "**Section 43.4.13, Example of A/D Timer Operation**".
2. x = 3, 4

43.3.3.10 ADCJnSGSRx — Scan Group x Status Register

ADCJnSGSRx is a read-only register that indicates the status of scan group x.

Access: This register is a read-only register that can be read in 8-bit units.

Address: ADCJnSGSR0: <ADCJn_base> + 460_H
ADCJnSGSR1: <ADCJn_base> + $4A0_H$
ADCJnSGSR2: <ADCJn_base> + $4E0_H$
ADCJnSGSR3: <ADCJn_base> + 520_H
ADCJnSGSR4: <ADCJn_base> + 560_H

Value after reset: 00_H

When x = 0 to 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SGACT	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.31 ADCJnSGSRx Register Contents (x = 0 to 2)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	SGACT	Scan Group Status This bit indicates the A/D conversion status of scan group x. 0: Scan group x is in idle state. 1: A/D conversion of scan group x is in progress.
0	Reserved	When read, the value after reset is returned.

When x = 3, 4

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADTACT	SGACT	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.32 ADCJnSGSRx Register Contents (x = 3, 4)

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	ADTACT	A/D Timer Status This bit indicates the operating status of A/D timer x. 0: A/D timer x is in idle state. 1: A/D timer x is running
1	SGACT	Scan Group Status This bit indicates the A/D conversion status of scan group x. 0: Scan group x is in idle state. 1: A/D conversion of scan group x is in progress.
0	Reserved	When read, the value after reset is returned.

NOTES

1. x = 0 to 4
2. ADCJnSGSRx.SGACT bit is set to 1 after the start trigger of scan group x is input. There are the following 3 conditions that SGACT bit is set to 0 except reset.
   1) When the A/D conversion of virtual channel j that is set in ADCJnSGVCPRx.VCEP is finished.
   2) When Forced Termination (ADCJnADHALTR.HALT) is written to 1.
   3) When Scan Group Stop (ADCJnSGSTPCRx.SGSTP) is written to 1.

43.3.3.11 ADCJnPWDCR — PWM-Diag Control Register

ADCJnPWDCR is a readable and writable register to control the PWM-Diag.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 580_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.33 ADCJnPWDCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWE	PWM-Diag Enable 0: The PWM-Diag is disabled. 1: The PWM-Diag is enabled.

NOTE

When ADCJnPWDCR.PWE is set to 1, SG4 executes the PWM-Diag function.

CAUTIONS

1. When ADCJnPWDCR.PWE is set to 1, setting of the A/D timer of SG4, multicycle setting, and the setting of the virtual channel pointer are disabled. Use the PWM-Diag scan group control register (ADCJnPWDSGCR) to control (ADCJnSGCR4, ADCJnSGVCPR4, ADCJnSGMCYCR4, ADTSTCR4, and ADTENDCR4) SG4.
2. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.

43.3.3.12 ADCJnPWDSGCR — PWM-Diag Scan Group Control Register

ADCJnPWDSGCR is a readable and writable register to control the PWM-Diag hardware trigger.

When the PWM-Diag is used, a scan end interrupt is always output.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 584_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TRGMD
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.34 ADCJnPWDSGCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD	Trigger Mode This bit is used to set whether to enable or disable hardware triggers of the PWM-Diag. 0: Disabled 1: Enabled

CAUTIONS

1. When ADCJnPWDCR.PWE = 1, this register setting is enabled.
2. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see **Section 43.4.4, Restrictions on Updating Settings**.

43.3.3.13 ADCJnSGSTR — Scan Group Common Status Register

ADCJnSGSTR is a read-only register that indicates the scan group status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <ADCJn_base> + 600_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ADTACT[4:3]		—	—	—	—	SHACT	SVSACT	SGACT[4:0]					—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.35 ADCJnSGSTR Register Contents

Bit Position	Bit Name	Function
15, 14	Reserved	When read, the value after reset is returned.
13	ADTACT[4]	A/D Timer 4 Status 0: A/D timer 4 is in idle state. 1: A/D timer 4 is running.
12	ADTACT[3]	A/D Timer 3 Status 0: A/D timer 3 is in idle state. 1: A/D timer 3 is running.
11 to 8	Reserved	When read, the value after reset is returned.
7	SHACT	T&H Status Flag 0: T&H is stopped. 1: T&H conversion or sampling is in progress.
6	SVSACT	SVSTOP Status Flag 0: SVSTOP is canceled. 1: SVSTOP is accepted.
5	SGACT[4]	SGACT[4] Scan Group 4 / PWM-Diag (SG4 / PWM-Diag) Status Flag 0: A/D conversion for SG4 / PWM-Diag is completed. 1: A/D conversion for SG4 / PWM-Diag is in processing or suspension.
4	SGACT[3]	SGACT[3] Scan Group 3 (SG3) Status Flag 0: A/D conversion for SG3 is completed. 1: A/D conversion for SG3 is in processing or suspension.
3	SGACT[2]	SGACT[2] Scan Group 2 (SG2) Status Flag 0: A/D conversion for SG2 is completed. 1: A/D conversion for SG2 is in processing or suspension.
2	SGACT[1]	SGACT[1] Scan Group 1 (SG1) Status Flag 0: A/D conversion for SG1 is completed. 1: A/D conversion for SG1 is in processing or suspension.
1	SGACT[0]	SGACT[0] Scan Group 0 (SG0) Status Flag 0: A/D conversion for SG0 is completed. 1: A/D conversion for SG0 is in processing or suspension.
0	Reserved	When read, the value after reset is returned.

43.3.3.14 ADCJnADSYNSTCR — A/D Synchronization Start Control Register

ADCJnADSYNSTCR is a write-only register to control simultaneous start of A/D conversion of scan groups ADCJ0 and ADCJ1. This register is always read as 0.

This register setting is available only for ADCJ0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJ0_base> + 610_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADSTART
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.36 ADCJnADSYNSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ADSTART	A/D Conversion Simultaneous Start This software trigger is used to simultaneously start A/D conversion of scan groups ADCJ0 and ADCJ1. 0: No function (Writing 0 to this bit is ignored.) 1: A/D conversion start Set the scan group synchronization start enable bit (ADCJnSGCRx.ADSTARTE bit of ADCJ0 and ADCJ1) of desired scan group x in advance.

43.3.3.15 ADCJnADTSYNSTCR — A/D Timer Synchronization Start Control Register

ADCJnADTSYNSTCR is a write-only register to control simultaneous count operation start of each A/D timer of ADCJ0 and ADCJ1. This register is always read as 0.

This register setting is available only for ADCJ0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJ0_base> + 614_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADTSTART
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.37 ADCJnADTSYNSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ADTSTART	A/D Timer Simultaneous Start This software trigger is used to simultaneously start A/D timer count operation of ADCJ0 and ADCJ1. 0: No function (Writing 0 to this bit is ignored.) 1: A/D timer count start The A/D timer is implemented in scan groups 3 and 4. Set the A/D timer synchronization start enable bit (ADCJnSGCRx.ADTSTARTE bit of ADCJ0 and ADCJ1) of desired scan group x in advance. A/D conversion of the scan group starts at the end of A/D timer count.

43.3.4 Hardware Trigger Specific Register

43.3.4.1 ADCJnSGTSELx — Scan Group x Start Trigger Control Register

This register is used to select the A/D conversion trigger (hardware trigger) for SGx (x = 0 to 4).

This register setting is available only for ADCJ2.

Access: This register can be read or written in 32-bit units.

Address: <ADCJ2_SELB_base> + x × 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TxSEL5	TxSEL4	TxSEL3	TxSEL2	TxSEL1	TxSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.38 ADCJnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	TxSELp (p = 0 to 5)	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled. **CAUTION** When setting TxSELp to 1, set only one of the bits to 1.

The list below shows the hardware triggers to be selected.

Table 43.39 List of A/D Conversion Hardware Triggers

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCJ2	ADCJ2SGTSEL0	T0SEL0	INTTAUD2I7	TAUD2
		T0SEL1	INTTAUD2I15	TAUD2
		T0SEL2	INTTAUJ2I3	TAUJ2
		T0SEL3	INTTAUJ3I3	TAUJ3
		T0SEL4	ADCJ2TRG0	External trigger pin
		T0SEL5	SEQADTRG	LPS
	ADCJ2SGTSEL1	T1SEL0	INTTAUD2I7	TAUD2
		T1SEL1	INTTAUD2I15	TAUD2
		T1SEL2	INTTAUJ2I3	TAUJ2
		T1SEL3	INTTAUJ3I3	TAUJ3
		T1SEL4	ADCJ2TRG1	External trigger pin
		T1SEL5	SEQADTRG	LPS
	ADCJ2SGTSEL2	T2SEL0	INTTAUD2I7	TAUD2
		T2SEL1	INTTAUD2I15	TAUD2
		T2SEL2	INTTAUJ2I3	TAUJ2
		T2SEL3	INTTAUJ3I3	TAUJ3
		T2SEL4	ADCJ2TRG2	External trigger pin
		T2SEL5	SEQADTRG	LPS
	ADCJ2SGTSEL3	T3SEL0	INTTAUD2I7	TAUD2
		T3SEL1	INTTAUD2I15	TAUD2
		T3SEL2	INTTAUJ2I3	TAUJ2
		T3SEL3	INTTAUJ3I3	TAUJ3
		T3SEL4	ADCJ2TRG3	External trigger pin
		T3SEL5	SEQADTRG	LPS
	ADCJ2SGTSEL4	T4SEL0	INTTAUD2I7	TAUD2
		T4SEL1	INTTAUD2I15	TAUD2
		T4SEL2	INTTAUJ2I3	TAUJ2
		T4SEL3	INTTAUJ3I3	TAUJ3
		T4SEL4	ADCJ2TRG4	External trigger pin
		T4SEL5	SEQADTRG	LPS

CAUTIONS

1. **To prevent malfunction, ADCJnSGTSELx should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.**
2. **When SEQADTRG (LPS) is selected as hardware trigger, perform AD conversion in 1 Scan group only (It's prohibited to select SEQADTRG for multiple Scan groups).**

43.3.5 ADC Specific Registers (Control)

43.3.5.1 ADCJnADHALTR — A/D Halt Register

ADCJnADHALTR is a write-only register to forcibly terminate the ADCJ. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + 640_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HALT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.40 ADCJnADHALTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	HALT	Forced Termination This bit is used to forcibly terminate and initialize all scan groups and all A/D timers to drive the ADC to idle state. 0: No function (Writing 0 to this bit is ignored). 1: All scan groups and all A/D timers are forcibly terminated.

43.3.5.2 ADCJnADCR1 — A/D Control Register 1

ADCJnADCR1 is a readable and writable register for common control of the ADCJ.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 644_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SUSMTD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 43.41 ADCJnADCR1 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SUSMTD[1:0]	Suspending Method These bits are used to set the suspending method between scan groups. 0_H: Synchronous suspend 1_H: Synchronous/asynchronous mixed suspend 2_H: Asynchronous suspend 3_H: Setting prohibited (synchronous suspend)

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.5.3 ADCJnADCR2 — A/D Control Register 2

ADCJnADCR2 is a readable and writable register for common control of the ADCJ.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 648_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	DFMT[2:0]			—	—	—	ADDNT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W

Table 43.42 ADCJnADCR2 Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	DFMT[2:0]	Data Format These bits are used to specify the data format of ADCJnDRj, ADCJnPWDDR. 000_B: Resolution 12-bit signed fixed-point format 001_B: Resolution 12-bit signed integer format 010_B: Resolution 12-bit unsigned fixed-point format 100_B: Resolution 12-bit unsigned integer format (right-justified) 101_B: Resolution 12-bit unsigned integer format (left-justified) For GTM Data is output to the GTM in the format described in "**Section 43.2.3.4, (3) Data Format for Output to GTM**" regardless of the DFMT[2:0] bits setting. For the data format for ADCJnVCULLMTBRy specified by this register setting value, see "**Section 43.2.3.4, (2) Data Format of Upper/Lower Limit Check Table Register for data register**".
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ADDNT	Addition Count Select This bit is used to select the addition times of addition mode A/D conversion 0: Addition twice 1: Addition four times ADCJ has the following 3 types of addition mode A/D conversion. (1) Addition mode A/D conversion. (2) MPX addition mode A/D conversion. (3) Addition mode Wiring-Break detection*1. Note 1. Refer to WADDE bit in **Section 43.3.6.2, ADCJnODCR — Wiring Break Detection Control Register** for details.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.5.4 ADCJnSMPCR — Sampling Control Register

ADCJnSMPCR is a readable and writable register to control sampling.

Access: This register can be read or written in 32-bit units.

Address: <ADCJn_base> + $64C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	EXSMPT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUFAMPD	—	—	SMPTS	—	—	—	—	SMPT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.43 ADCJnSMPCR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	EXSMPT[7:0]	Extended Sampling Period These bits are selected when CNVCLS[3:0] = extended sampling A/D conversion mode. 00_H: Set to 60 states. 63_H: Set to 99 states. $8A_H$: Set to 138 states. FC_H: Set to 252 states. Other than above: Setting prohibited
15	BUFAMPD	Buffer Amplifier Disable 0: A buffer amplifier is used. 1: No buffer amplifier is used.
14, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SMPTS	Sampling Period Select 0: The SMPT[7:0] bits are not selected. 1: The SMPT[7:0] bits are selected.
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	SMPT[7:0]	Sampling Period 00_H: Set to 18 states. 63_H: Set to 99 states. CC_H: Set to 204 states. FC_H: Set to 252 states. Other than above: Setting prohibited

For the settings of sampling period and buffer amplifier, see "**Section 43.4.11.3, Sampling Period Setting**".

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.5.5 ADCJnMPXCURCR — MPX Current Control Register

ADCJnMPXCURCR is a readable and writable register to control the ADCJnMPXCURR1 format.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 650_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MSKCFMT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.44 ADCJnMPXCURCR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MSKCFMT[3:0]	Mask Control Format These bits are used to set the format of the mask control bits (ADCJnMPXCURR1.MSKC[15:0]) in the MPX current register. MSKCFMT[3] 0: MSKC[15:12] = 0000 1: MSKC[15:12] = 1111 MSKCFMT[2] 0: MSKC[11:8] = 0000 1: MSKC[11:8] = 1111 MSKCFMT[1] 0: MSKC[7:4] = 0000 1: Setting prohibited MSKCFMT[0] 0: MSKC[3:0] = 0000 1: MSKC[3:0] = 0111

CAUTIONS

1. Decide the setting value referring to the I/O port specifications.
2. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTE

ADCJnMPXCURCR is the register that is arranged in 32-bit area. When ADCJnMPXCURCR is read, the 32-bit read data is only ADCJnMPXCURCR.

43.3.5.6 ADCJnMPXINTER — MPX Interrupt Enable Register

ADCJnMPXINTER is a readable and writable register to control MPX interrupt INT_MPX output.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 654_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADMPXIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.45 ADCJnMPXINTER Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ADMPXIE	MPX Interrupt Enable This bit is used to control the output of MPX interrupt INT_MPX. 0: INT_MPX is not output. 1: INT_MPX is output.

NOTE

ADCJnMPXINTER is the register that is arranged in 32-bit area. When ADCJnMPXINTER is read, the 32-bit read data is only ADCJnMPXINTER.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.

43.3.5.7 ADCJnMPXCURR1 — MPX Current Register 1

ADCJnMPXCURR1 is a read-only register to store the MPX value.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + 658_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MSKC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXCMD[7:0]								—	—	—	—	—	MPXCUR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.46 ADCJnMPXCURR1 Register Contents

Bit Position	Bit Name	Function
31 to 16	MSKC[15:0]	Mask Control The mask format is stored according to the MSKCFMT[3:0] setting in ADCJnMPXCURCR.
15 to 8	MPXCMD[7:0]	SPI Communication Command Information The MPX command information register (ADCJnMPXCMDR) value is stored at the start of MPX normal A/D conversion or MPX addition mode A/D conversion.
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	MPXCUR[2:0]	Current MPX Value The MPXV[2:0] value in ADCJnVCRj is stored at the start of MPX normal A/D conversion or MPX addition mode A/D conversion.

NOTE

For details of operation, see “**Section 43.4.8, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode**”.

43.3.5.8 ADCJnMPXCURR2 — MPX Current Register 2

ADCJnMPXCURR2 is a read-only register to store the MPX value for an external analog multiplexer.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ADCJn_base> + $65C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MPXCUR[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.47 ADCJnMPXCURR2 Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	MPXCUR[2:0]	Current MPX Value These bits store the MPXV[2:0] value in ADCJnPWDVCR (if a virtual channel is started with MPXE in ADCJnPWDVCR set to 1_H) or store the MPXV[2:0] value in ADCJnVCRj (if a virtual channel is started with CNVCLS[3:0] in ADCJnVCRj set to 5_H or 6_H).

NOTE

For details of operation, see "**Section 43.4.8, Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode**".

43.3.5.9 ADCJnMPXCMDR — MPX Command Information Register

ADCJnMPXCMDR is a readable and writable register to store the SPI communication command information to be transferred to an external analog multiplexer.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 660_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	MPXCMD[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.48 ADCJnMPXCMDR Register Contents

Bit Position	Bit Name	Function
7 to 0	MPXCMD[7:0]	SPI Communication Command Information These bits are used to set the MSPI (SPI I/F) command to control MPX. When MPX normal A/D conversion or MPX addition mode A/D conversion starts, this command information is transferred to the SPI communication command information bits (ADCJnMPXCURR1.MPXCMD[7:0]) in the MPX current register.

CAUTIONS

1. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".
2. Determine the setting value according to the MSPI specifications.

43.3.5.10 ADCJnGTMENTSGER — GTM Entry Scan Group Enable Register

ADCJnGTMENTSGER is a readable and writable register to control whether to enable or disable scan groups to be entered in the GTM.

Access: This register can be read or written in 16-bit units.

Address: <ADCJn_base> + 670_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GTMENTSG4E	GTMENTSG3E	GTMENTSG2E	GTMENTSG1E	GTMENTSG0E	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 43.49 ADCJnGTMENTSGER Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	GTMENTSGxE	GTM Entry Scan Group Enable 0: GTM entry when SGx is activated is disabled. 1: GTM entry when SGx is activated is enabled. Entry is made in a virtual channel for which GTMENT in ADCJnVCRj is set to 1.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.5.11 ADCJnADENDPv — A/D Conversion Monitor Virtual Channel Pointer v

ADCJnADENDPv is a readable and writable register to select a virtual channel that outputs the A/D conversion timing signal to ADENDv.

Access: This register can be read or written in 8-bit units.

Address: ADCJnADENDP0: <ADCJn_base> + 674_H
ADCJnADENDP1: <ADCJn_base> + 678_H
ADCJnADENDP2: <ADCJn_base> + $67C_H$
ADCJnADENDP3: <ADCJn_base> + 680_H
ADCJnADENDP4: <ADCJn_base> + 684_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	ENDP[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.50 ADCJnADENDPv Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	ENDP[5:0]	A/D Conversion Monitor Virtual Channel Pointer These bits are used to set virtual channel numbers that monitor ongoing A/D conversion. 00_H to $3F_H$: Virtual channels 00 to 63 are set. The ADENDv pin is set to high level at the start of A/D conversion of the set virtual channel, and is set to low level at the end of A/D conversion.

CAUTION

1. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.
2. PWM-Diag Virtual Channel Register (ADCJnPWDVCR) is not set to ENDP[5:0] of ADCJnADENDPv.

43.3.5.12 ADCJnTHSMPSTCR — T&H Sampling Start Control Register

ADCJnTHSMPSTCR is a write-only register to control sampling start of all T&H. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + 690_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SMPST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.51 ADCJnTHSMPSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SMPST	T&H Sampling Start This software trigger is used to start sampling of all T&H. 0: No function (Writing 0 to this bit is ignored.) 1: Sampling start

NOTE

The T&H sampling processing is activated only by a software trigger (ADCJnTHSMPSTCR.SMPT), but is not activated by a hardware trigger.

CAUTIONS

1. **To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see 43.4.4, Restrictions on Updating Settings. For relevant setting flow, see 43.4.2.2, SG1/SG2 Startup Flow and 43.4.2.3, SG3/SG4 Startup Flow.**
   - **When T&H is used, it is recommended to set 0 to ADCJnTHCR.ASMPMSK (Automatic sampling is effective).**
   - **When the automatic sampling is effective, the setting of T&H sampling start is needed only first startup-time. Set the first T&H sampling start while all scan groups are stopped.**
2. **ADCJnTHSMPSTCR.SMPST is forced to start sampling processing of all T&Hs. It is prohibited to write this bit to 1 when the scan group of T&H group A is processing.**

43.3.5.13 ADCJnTHSTPCR — T&H Stop Control Register

ADCJnTHSTPCR is a write-only register to control stop of all T&H. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + 694_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	THSTP
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.52 ADCJnTHSTPCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	THSTP	T&H Sampling Stop This software trigger is used to stop all T&H. 0: No function (Writing 0 to this bit is ignored.) 1: All T&H is stopped

CAUTION

Use the THSTP bit after all scan groups have been forcibly stopped by ADCJnADHALTR.HALT.

Furthermore, after T&H is stopped by the THSTP bit, set the THzE (z = 0 to 3) bits to all 0 to prevent re-sampling by the automatic sampling function (ADCJnTHCR.ASMPMASK).

43.3.5.14 ADCJnTHCR — T&H Control Register

ADCJnTHCR is a readable and writable register to control the automatic sampling function of T&H.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 698_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ASMPMSK
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.53 ADCJnTHCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ASMPMSK	Automatic Sampling Mask This bit is used to set whether to start sampling automatically at the end of hold value A/D conversion. 0: Sampling starts automatically. 1: Sampling does not start automatically.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.

43.3.5.15 ADCJnTHAHLDSTCR— T&H Group A Hold Start Control Register

ADCJnTHAHLDSTCR is a write-only register to control hold start of T&H group A. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + $6A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.54 ADCJnTHAHLDSTCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Hold Start This software trigger is used to start hold of T&H group A. 0: No function (Writing 0 to this bit is ignored.) 1: Hold start

43.3.5.16 ADCJnTHER — T&H Enable Register

ADCJnTHER is a readable and writable register to control whether to enable or disable each T&H.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + $6B4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TH3E	TH2E	TH1E	TH0E
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.55 ADCJnTHER Register Contents

Bit Position	Bit Name	Function
7to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	THzE	T&H Enable These bits are used to set whether to enable or disable track and hold operation of the T&Hz circuit. 0: Disabled 1: Enabled

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”.

43.3.5.17 ADCJnTHACR — T&H Group A Control Register

ADCJnTHACR is a readable and writable register to control T&H group A.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + $6C0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 43.56 ADCJnTHACR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	Hold Control Enable This bit is used to specify whether to enable or disable the hold control. 0: Hold control is disabled. 1: Hold control is enabled.
4	HLDTE	Hold Trigger Enable This bit is used to specify whether to enable or disable hardware trigger signals. 0: Disabled 1: Hardware trigger signals of the scan group selected by the scan group select bits (ADCJnTHACR.SGS[1:0]) are enabled.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SGS[1:0]	Selecting Scan Group These bits are used to select scan group x of T&H group A. 0_H: SG1 1_H: SG2 2_H: SG3 3_H: SG4 **CAUTION** SG0 is not selectable. When running the PWM-Diag in SG4 with the PWE bit in ADCJnPWDCR set to 1, setting SG4 is prohibited.

CAUTIONS

1. SG0 is not selectable for the T&H target scan group.
2. When the PWE bit in ADCJnPWDCR is set to 1, and when the PWM-Diag is operated with SG4, it is prohibited to select the T&H target scan group to SG4.
3. When using T&H, set SUSMTD[1:0] in ADCJnADCR1 to 2_H. Setting any other value is prohibited.
4. Set TRGMD in ADCJnSGCRx in the T&H target scan group to 1_H. Setting any other value is prohibited.
5. When using T&H, select multicycle scan mode (ADCJnSGCRx.SCANMD = 0_H) of the T&H target scan group and set the scan count to one (ADCJnSGMCYCRx.MCYC[7:0] = 0_H). Any other combination of settings is prohibited.
6. When using hardware triggers by the T&H function, set ADCJnTHACR.HLDCTE = 1 and ADCJnTHACR.HLDTE = 1.
7. When using software triggers by the T&H function, set ADCJnTHACR.HLDCTE = 1 and ADCJnTHACR.HLDTE = 0.
8. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see “**Section 43.4.4, Restrictions on Updating Settings**”. However, clearing the HLDTE bit to 0 when stopping a scan group is excluded.

43.3.5.18 ADCJnWAITTRy — Wait Setting Register y

ADCJnWAITTRy is a readable and writable register to set arbitrary wait time before executing a virtual channel.

Access: This register can be read or written in 16-bit units.

Address: ADCJnWAITTR0: <ADCJn_base> + 700_H
ADCJnWAITTR1: <ADCJn_base> + 704_H
ADCJnWAITTR2: <ADCJn_base> + 708_H
ADCJnWAITTR3: <ADCJn_base> + $70C_H$
ADCJnWAITTR4: <ADCJn_base> + 710_H
ADCJnWAITTR5: <ADCJn_base> + 714_H
ADCJnWAITTR6: <ADCJn_base> + 718_H
ADCJnWAITTR7: <ADCJn_base> + $71C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	WAITTIME[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.57 ADCJnWAITTRy Register Contents

Bit Position	Bit Name	Function
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	WAITTIME[13:0]	These bits are used to set arbitrary wait time to be inserted before executing a virtual channel. 0000_H: No wait is inserted before executing a virtual channel. Other than 0000_H: A wait for the set time period is inserted before executing a virtual channel. Calculating formula: Wait time (μs) = WAITTIME[13:0] × CLKAD cycle (μs) * An example of cycle period setting is shown below. * CLKAD cycle (μs) = 1/CLKAD frequency (MHz)

Example of WAITTIME[13:0] Setting

Stabilization Time (μs)	CLKAD Frequency (MHz)					
	10	12	16	20	24	40
1	$000A_H$	$000C_H$	0010_H	0014_H	0018_H	0028_H
100	$03E8_H$	$04B0_H$	0640_H	$07D0_H$	0960_H	$0FA0_H$
250	$09C4_H$	$0BB8_H$	$0FA0_H$	1388_H	1770_H	2710_H
400	$0FA0_H$	$12C0_H$	1900_H	$1F40_H$	2580_H	$3E80_H$

CAUTIONS

1. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".
2. When using a multiplexer in the first virtual channel of each scan group (ADCJnVCRj.CNVCLS[3:0] = 5_H or 6_H or ADCJnPWDVCR.MPXE = 1), insert a wait to the first virtual channel. When using no multiplexer, do not insert a wait to the first virtual channel of each scan group.

43.3.5.19 ADCJnVMONVDCR1 — Voltage Monitoring Voltage Divider Control Register 1

ADCJnVMONVDCR1 is a readable and writable register to control the voltage monitoring voltage divider of the power supply. This register setting is available only for ADCJ0.

Access: This register can be read or written in 8-bit units.

Address: <ADCJ0_base> + 740_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VDE1
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.58 ADCJnVMONVDCR1 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	VDE1	Voltage Divider Enable (Voltage Dividing Resistor Control) This bit is used to control the voltage dividing resistor together with the VDE2 bit in ADCJnVMONVDCR2. For details, see "**Table 43.60, Resistance voltage divide and Pull-Down Control Setting Table**".

CAUTIONS

1. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".
2. Do not perform A/D conversion with the setting of ADCJnVMONVDCR1.VDE1 or ADCJnVMONVDCR2.VDE2 = 01_B or 10_B.

43.3.5.20 ADCJnVMONVDCR2 — Voltage Monitoring Voltage Divider Control Register 2

ADCJnVMONVDCR2 is a readable and writable register to control the voltage monitoring voltage divider of the power supply. This register setting is available only for ADCJ0.

Access: This register can be read or written in 8-bit units.

Address: <ADCJ0_base> + 744_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	VDE2
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.59 ADCJnVMONVDCR2 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	VDE2	Voltage Divider Enable (Voltage Dividing Resistor Control) This bit is used to control the voltage dividing resistor together with the VDE1 bit in ADCJnVMONVDCR1. For details, see "**Table 43.60, Resistance voltage divide and Pull-Down Control Setting Table**".

CAUTIONS

1. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".
2. Do not perform A/D conversion with the setting of ADCJnVMONVDCR1.VDE1 or ADCJnVMONVDCR2.VDE2 = 01_B or 10_B.

Table 43.60 Resistance voltage divide and Pull-Down Control Setting Table

VDE1	VDE2	Resistance voltage divide	Pull-Down
0	0	OFF	ON
0	1	OFF	OFF
1	0	OFF	OFF
1	1	ON	OFF

43.3.6 ADC Specific Registers (Safety-Related)

43.3.6.1 ADCJnTDCR — Pin Level Self-Diagnostic Control Register

ADCJnTDCR is a readable and writable register to control T&H path self-diagnosis and pin-level self-diagnosis.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 760_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	THSDE	—	—	—	—	—	TDLV[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W

Table 43.61 ADCJnTDCR Register Contents

Bit Position	Bit Name	Function
7	THSDE	T&H Path Self-Diagnosis Enable 0: T&H path self-diagnosis is disabled. 1: T&H path self-diagnosis is enabled.
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TDLV[1:0]	Pin-Level Self-Diagnosis Voltage Level These bits are used to set a voltage level applied at the time of pin-level self-diagnosis. • Target physical channel is high accuracy input 0_H: Even groups of Pin-Level Self-Diagnosis Voltage are discharged to AnVSS and odd groups are charged to AnVCC. 1_H: Even groups of Pin-Level Self-Diagnosis Voltage are charged to AnVCC and odd groups are discharged to AnVSS. 2_H: Even groups of Pin-Level Self-Diagnosis Voltage are discharged to AnVSS and odd groups are charged to 1/2 × AnVCC. 3_H: Even groups of Pin-Level Self-Diagnosis Voltage are charged to 1/2 × AnVCC and odd groups are discharged to AnVSS. • Target physical channel is low accuracy input 0_H: Even groups of Pin-Level Self-Diagnosis Voltage are discharged to AnVSS and odd groups are charged to EnVCC. 1_H: Even groups of Pin-Level Self-Diagnosis Voltage are charged to EnVCC and odd groups are discharged to AnVSS. 2_H: Setting prohibited. 3_H: Setting prohibited.

CAUTIONS

1. When T&H path self-diagnosis is executed, set registers by seeing "**Section 43.4.10.2, Self-Diagnosis of the T&H Path**".
2. To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.6.2 ADCJnODCR — Wiring Break Detection Control Register

ADCJnODCR is a readable and writable register to control wiring-break detection.

Access: This register can be read or written in 16-bit units.

Address: <ADCJn_base> + 764_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WADDE	—	—	—	ODPW[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 43.62 ADCJnODCR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	WADDE	Addition Mode Wiring-Break Detection Enable 0: A/D conversion of wiring-break detection mode is performed in normal mode. 1: A/D conversion of wiring-break detection mode is performed in addition mode.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	ODPW[4:0]	Wiring-Break Detection Pulse Width for Wiring-Break Detection Mode 1 04_H: 1 state (of internal clock) 05_H: 2 states (of internal clock) : 13_H: 16 states (of internal clock) 14_H: 17 states (of internal clock) Setting a value equal to or less than 03_H and equal to or more than 15_H is prohibited.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTES

1. A/D conversion target analog pins are discharged with the pulse width specified by ODPW[4:0] after the A/D conversion sampling has been completed.
2. A/D conversion of wiring-break detection mode means the following setting.

   Virtual channel j of Scan group x

ADCJnVCRj.CNVCLS[3:0]	Function
8_H	A/D conversion in wiring-break detection mode 1
9_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)
A_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)
B_H	Self-diagnosis A/D conversion in wiring-break detection mode 1
C_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)
D_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)

43.3.6.3 ADCJnSFTCR — Safety Control Register

ADCJnSFTCR is a readable and writable register for safety control.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 770_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	TOCEIE	—	RDCLRE	—	OWEIE	PEIE	IDEIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R/W	R	R/W	R/W	R/W

Table 43.63 ADCJnSFTCR Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TOCEIE	Trigger Overlap Check Error Interrupt Enable This bit specifies whether to enable or disable trigger overlap check error interrupts. 0: Disabled 1: Enabled
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	RDCLRE	Read and Clear Enable This bit sets read and clear enable operation. This bit specifies whether to clear the clearing target register by reading the following target registers. Clearing target registers: ADCJnDRj, ADCJnDIRj, ADCJnPWDDR, and ADCJnPWDDIR. 0: Not cleared by reading. 1: Cleared by reading.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	OWEIE	Overwrite Error Interrupt Enable This bit specifies whether to enable or disable overwrite error interrupts. 0: Disabled 1: Enabled
1	PEIE	Parity Error Interrupt Enable This bit specifies whether to enable or disable parity error interrupts. 0: Disabled 1: Enabled
0	IDEIE	ID Error Interrupt Enable This bit specifies whether to enable or disable ID error interrupts. 0: Disabled 1: Enabled

NOTE

Refer to Note in the following chapter for clear condition of each bits of Data Supplementary Register.
Section 43.3.2.5, ADCJnDIRj — Data Supplementary Information Register j
Section 43.3.2.6, ADCJnPWDDIR — PWM-Diag Data Supplementary Information Register

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.6.4 ADCJnTOCCR — Trigger Overlap Check Control Register

ADCJnTOCCR is a readable and writable register to control trigger overlap check.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + 774_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TOCE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.64 ADCJnTOCCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TOCE	Trigger Overlap Check Function Enable 0: Disabled 1: Enabled

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.6.5 ADCJnTOCER — Trigger Overlap Check Error Status Register

ADCJnTOCER is a read-only register that indicates a trigger overlap check error.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ADCJn_base> + 790_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	TOCESG[4:0]					—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.65 ADCJnTOCER Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 1	TOCESG[4:0]	These bits indicate that a trigger overlap check error has occurred in SG0 to SG4. TOCESG[x] 0: No error in SGx 1: An error in SGx
0	Reserved	When read, the value after reset is returned.

CAUTIONS

1. The TOCESG[x] value is retained until a value of 1 is written to the SGx trigger overlap check error clear bit (ADCJnECR.TOCESGC[x]).
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when TOCESG[x] = 0, setting takes precedence (TOCESG[x] = 1).
   If a conflict occurs when TOCESG[x] = 1, clearing takes precedence (TOCESG[x] = 0).

43.3.6.6 ADCJnOWER — Overwrite Error Register

ADCJnOWER is a read-only register that indicates an overwrite error.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ADCJn_base> + $79C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	OWE	—	OWECAP[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.66 ADCJnOWER Register Contents

Bit Position	Bit Name	Function
7	OWE	Overwrite Error 0: No error is present. 1: An error is present.
6	Reserved	When read, the value after reset is returned.
5 to 0	OWECAP[5:0]	Overwrite Error Capture These bits indicate the first virtual channel number in which an overwrite error has occurred. 00_H-$3F_H$: An overwrite error has occurred in ADCJnVCRj. No value other than above is indicated.

CAUTIONS

1. This register value is retained until a value of 1 is written to the overwrite error clear bit (ADCJnECR.OWEC). This register is set simultaneously when the A/D conversion result is stored in the data register (ADCJnDRj).
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when OWE = 0, setting takes precedence (OWE = 1, OWECAP capture).
   If a conflict occurs when OWE = 1, clearing takes precedence (OWE = 0, OWECAP clear).

43.3.6.7 ADCJnPER — Parity Error Status Register

ADCJnPER is a read-only register that indicates a parity error.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ADCJn_base> + $7A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PE	PECAP[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.67 ADCJnPER Register Contents

Bit Position	Bit Name	Function
7	PE	Parity Error 0: No error is present. 1: An error is present.
6 to 0	PECAP[6:0]	Parity Error Capture These bits indicate the first virtual channel number in which a parity error has occurred. 00_H-$3F_H$: A parity error has occurred in ADCJnVCRj. 70_H: A parity error has occurred in ADCJnPWDVCR. No value other than the above values is indicated.

CAUTIONS

1. This register value is retained until a value of 1 is written to the parity error clear bit (ADCJnECR.PEC). This register is set simultaneously when reading ADCJnDRj, ADCJnDIRj, ADCJnPWDDR, or ADCJnPWDDIR has been completed.
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when PE = 0, setting takes precedence (PE = 1, PECAP capture).
   If a conflict occurs when PE = 1, clearing takes precedence (PE = 0, PECAP clear).

43.3.6.8 ADCJnIDER — ID Error Register

ADCJnIDER is a read-only register that indicates an ID error.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ADCJn_base> + $7A4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDE	IDECAP[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 43.68 ADCJnIDER Register Contents

Bit Position	Bit Name	Function
7	IDE	ID Error 0: No error is present. 1: An error is present.
6 to 0	IDECAP[6:0]	ID Error Capture These bits indicate the first virtual channel number in which an ID error has occurred. 00_H-$3F_H$: An ID error has occurred in ADCJnVCRj. 70_H: An ID error has occurred in ADCJnPWDVCR. No value other than the above values is indicated.

CAUTIONS

1. This register value is retained until a value of 1 is written to the ID error clear bit (ADCJnECR.IDEC). This register is set simultaneously when storing the A/D conversion result in the data register (ADCJnDRj, or ADCJnPWDDR).
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when IDE = 0, setting takes precedence (IDE = 1, IDECAP capture).
   If a conflict occurs when IDE = 1, clearing takes precedence (IDE = 0, IDECAP clear).

43.3.6.9 ADCJnECR — Error Clear Register

ADCJnECR is a write-only register to control error clear. This register is always read as 0.

Access: This register is a write-only register that can be written in 16-bit units.

Address: <ADCJn_base> + $7A8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TOCESGC[4:0]					—	—	—	—	—	—	OWEC	PEC	IDEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	R	R	R	R	R	R	W	W	W

Table 43.69 ADCJnECR Register Contents

Bit Position	Bit Name	Function
15, 14	Reserved	When writing, write the value after reset.
13 to 9	TOCESGC[4:0]	SGx Trigger Overlap Check Error Clear 0: TOCESG[x] is not cleared. 1: TOCESG[x] is cleared.
8 to 3	Reserved	When writing, write the value after reset.
2	OWEC	Overwrite Error Clear 0: Overwrite error is not cleared. 1: Overwrite error is cleared.
1	PEC	Parity Error Clear 0: Parity error is not cleared. 1: Parity error is cleared.
0	IDEC	ID Error Clear 0: ID error is not cleared. 1: ID error is cleared.

43.3.6.10 ADCJnVCULLMTBRy — Upper/Lower Limit Check Table Register y

ADCJnVCULLMTBRy is a readable and writable register to set the upper-limit threshold value and the lower-limit threshold value of A/D conversion values.

Be sure to write a 32-bit value to this register.

Access: This register can be read or written in 32-bit units.

Address: ADCJnVCULLMTBR0: <ADCJn_base> + 800_H
ADCJnVCULLMTBR1: <ADCJn_base> + 804_H
ADCJnVCULLMTBR2: <ADCJn_base> + 808_H
ADCJnVCULLMTBR3: <ADCJn_base> + $80C_H$
ADCJnVCULLMTBR4: <ADCJn_base> + 810_H
ADCJnVCULLMTBR5: <ADCJn_base> + 814_H
ADCJnVCULLMTBR6: <ADCJn_base> + 818_H
ADCJnVCULLMTBR7: <ADCJn_base> + $81C_H$

Value after reset: 7FFF 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VCULMTB[15:0]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VCLLMTB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.70 ADCJnVCULLMTBRy Register Contents

Bit Position	Bit Name	Function
31 to 16	VCULMTB[15:0]	Upper-Limit Threshold Value Table These bits are used to specify the upper-limit threshold value of A/D conversion values. For the data format to be set in this register, see **Section 43.2.3.4, (2) Data Format of Upper/Lower Limit Check Table Register for data register**.
15 to 0	VCLLMTB[15:0]	Lower-Limit Threshold Value Table These bits are used to specify the lower-limit threshold value of A/D conversion values. For the data format to be set in this register, see **Section 43.2.3.4, (2) Data Format of Upper/Lower Limit Check Table Register for data register**.

NOTE

ADCJnVCULLMTBRy is selected bye the following registers.

Register bit name	Category
ADCJnVCRj.VCULLMTBS[3:0]	Scan group x
ADCJnPWDVCR.VCULLMTBS[3:0]	PWM-Diag scan group

43.3.6.11 ADCJnVCLMINTER1 — Upper/Lower Limit Check Interrupt Enable Register 1

ADCJnVCLMINTER1 is a readable and writable register to control upper/lower limit check interrupt output of virtual channels 0 to 31.

Access: This register can be read or written in 32-bit units.

Address: <ADCJn_base> + 840_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADUL31IE	ADUL30IE	ADUL29IE	ADUL28IE	ADUL27IE	ADUL26IE	ADUL25IE	ADUL24IE	ADUL23IE	ADUL22IE	ADUL21IE	ADUL20IE	ADUL19IE	ADUL18IE	ADUL17IE	ADUL16IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADUL15IE	ADUL14IE	ADUL13IE	ADUL12IE	ADUL11IE	ADUL10IE	ADUL09IE	ADUL08IE	ADUL07IE	ADUL06IE	ADUL05IE	ADUL04IE	ADUL03IE	ADUL02IE	ADUL01IE	ADUL00IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.71 ADCJnVCLMINTER1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ADULjIE	Upper/Lower Limit Check Interrupt Enable 0: An upper/lower limit check interrupt (INT_UL) is not output. 1: An upper/lower limit check interrupt (INT_UL) is output.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTE

j = 00 to 31

43.3.6.12 ADCJnVCLMINTER2 — Upper/Lower Limit Check Interrupt Enable Register 2

ADCJnVCLMINTER2 is a readable and writable register to control upper/lower limit check interrupt output of virtual channels 32 to 63.

Access: This register can be read or written in 32-bit units.

Address: <ADCJn_base> + 844_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADUL63IE	ADUL62IE	ADUL61IE	ADUL60IE	ADUL59IE	ADUL58IE	ADUL57IE	ADUL56IE	ADUL55IE	ADUL54IE	ADUL53IE	ADUL52IE	ADUL51IE	ADUL50IE	ADUL49IE	ADUL48IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADUL47IE	ADUL46IE	ADUL45IE	ADUL44IE	ADUL43IE	ADUL42IE	ADUL41IE	ADUL40IE	ADUL39IE	ADUL38IE	ADUL37IE	ADUL36IE	ADUL35IE	ADUL34IE	ADUL33IE	ADUL32IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.72 ADCJnVCLMINTER2 Register Contents

Bit Position	Bit Name	Function
31 to 0	ADULjIE	Upper/Lower Limit Check Interrupt Enable 0: An upper/lower limit check interrupt (INT_UL) is not output. 1: An upper/lower limit check interrupt (INT_UL) is output.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

NOTE

j = 32 to 63

43.3.6.13 ADCJnPWVCLMINTER — PWM-Diag Upper/Lower Limit Check Interrupt Enable Register

ADCJnPWVCLMINTER is a readable and writable register to control upper/lower limit check interrupt output of PWM-Diag virtual channels.

Access: This register can be read or written in 8-bit units.

Address: <ADCJn_base> + $84C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWADULIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 43.73 ADCJnPWVCLMINTER Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWADULIE	PWM-Diag Upper/Lower Limit Check Interrupt Enable 0: An upper/lower limit check interrupt (INT_UL) is not output. 1: An upper/lower limit check interrupt (INT_UL) is output.

CAUTION

To prevent a malfunction, this register has restrictions for updating settings. For restrictions on updating settings, see "**Section 43.4.4, Restrictions on Updating Settings**".

43.3.6.14 ADCJnVCLMSR1 — Upper/Lower Limit Check Status Register 1

ADCJnVCLMSR1 is a read-only register that indicates the upper/lower limit decision result of virtual channels 0 to 31.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + 860_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC31LMS	VC30LMS	VC29LMS	VC28LMS	VC27LMS	VC26LMS	VC25LMS	VC24LMS	VC23LMS	VC22LMS	VC21LMS	VC20LMS	VC19LMS	VC18LMS	VC17LMS	VC16LMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC15LMS	VC14LMS	VC13LMS	VC12LMS	VC11LMS	VC10LMS	VC09LMS	VC08LMS	VC07LMS	VC06LMS	VC05LMS	VC04LMS	VC03LMS	VC02LMS	VC01LMS	VC00LMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.74 ADCJnVCLMSR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMS	Upper/Lower Limit Check Status of Virtual Channels 0: Status is not reported. 1: Status is reported. Setting condition Each status bit is set to 1 when the A/D conversion value is larger than the upper-limit check table value or smaller than the lower-limit check table value. Clearing condition (1) Each status bit is cleared to 0 by writing 1 to ADCJnVCLMSCR1.VCjLMSC (bit corresponding to each status). (2) Each status bit is cleared to 0 by writing 1 to ADCJnVCLMASCR.ALLMSC.

CAUTIONS

1. These register bits are set simultaneously when storing the A/D conversion result in the data register.
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when VCjLMS = 0, setting takes precedence (VCjLMS = 1).
   If a conflict occurs when VCjLMS = 1, clearing takes precedence (VCjLMS = 0).

NOTE

j = 00 to 31

43.3.6.15 ADCJnVCLMSR2 — Upper/Lower Limit Check Status Register 2

ADCJnVCLMSR2 is a read-only register that indicates the upper/lower limit decision result of virtual channels 32 to 63.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + 864_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC63LMS	VC62LMS	VC61LMS	VC60LMS	VC59LMS	VC58LMS	VC57LMS	VC56LMS	VC55LMS	VC54LMS	VC53LMS	VC52LMS	VC51LMS	VC50LMS	VC49LMS	VC48LMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC47LMS	VC46LMS	VC45LMS	VC44LMS	VC43LMS	VC42LMS	VC41LMS	VC40LMS	VC39LMS	VC38LMS	VC37LMS	VC36LMS	VC35LMS	VC34LMS	VC33LMS	VC32LMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.75 ADCJnVCLMSR2 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMS	Upper/Lower Limit Check Status of Virtual Channels 0: Status is not reported. 1: Status is reported. Setting condition Each status bit is set to 1 when the A/D conversion value is larger than the upper-limit check table value or smaller than the lower-limit check table value. Clearing condition (1) Each status bit is cleared to 0 by writing 1 to ADCJnVCLMSCR2.VCjLMSC (bit corresponding to each status). (2) Each status bit is cleared to 0 by writing 1 to ADCJnVCLMASCR.ALLMSC.

CAUTIONS

1. These register bits are set simultaneously when storing the A/D conversion result in the data register.
2. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when VCjLMS = 0, setting takes precedence (VCjLMS = 1).
   If a conflict occurs when VCjLMS = 1, clearing takes precedence (VCjLMS = 0).

NOTE

j = 32 to 63

43.3.6.16 ADCJnPWVCLMSR — PWM-Diag Upper/Lower Limit Check Status Register

ADCJnPWVCLMSR is a read-only register that indicates the upper/lower limit check result of PWM-Diag virtual channel.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCJn_base> + $86C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWVCLMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.76 ADCJnPWVCLMSR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PWVCLMS	PWM-Diag Upper/Lower Limit Check Status 0: Status is not reported. 1: Status is reported. Setting condition PWVCLMS is set to 1 when the A/D conversion value is larger than the upper limit or smaller than the lower limit. Clearing condition (1) This bit is cleared to 0 by writing 1 to the PWVCLMSC bit in ADCJnPWVCLMSCR. (2) This bit is cleared to 0 by writing 1 to the ALLMSC bit in ADCJnVCLMASCR.

CAUTION

Bit 0 is set to 1 simultaneously when the A/D conversion result is stored in the PWM-Diag data register (ADCJnPWDDR). When setting timing conflicts with clearing timing, this register operates as follows:

If a conflict occurs when PWVCLMS = 0, setting takes precedence (PWVCLMS = 1).

If a conflict occurs when PWVCLMS = 1, clearing takes precedence (PWVCLMS = 0).

43.3.6.17 ADCJnSGULCRx — Scan Group x Upper/Lower Limit Check Status Register

ADCJnSGULCRx is a read-only register that indicates the upper/lower limit decision result of each scan group.

If upper-limit or lower-limit status is reported again without clearing ADCJnSGULCRx, the latest information is discarded and the previous value is retained.

Access: This register is a read-only register that can be read in 16-bit units.

Address: ADCJnSGULCR0: <ADCJn_base> + 880_H
ADCJnSGULCR1: <ADCJn_base> + 884_H
ADCJnSGULCR2: <ADCJn_base> + 888_H
ADCJnSGULCR3: <ADCJn_base> + $88C_H$
ADCJnSGULCR4: <ADCJn_base> + 890_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UE	LE	—	—	MPXE	MPXV[2:0]			ULE	—	ULPC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.77 ADCJnSGULCRx Register Contents

Bit Position	Bit Name	Function
15	UE	Upper-Limit Check Status 0: The upper-limit status is not reported. 1: The upper-limit status is reported.
14	LE	Lower-Limit Check Status 0: The lower-limit status is not reported. 1: The lower-limit status is reported.
13, 12	Reserved	When read, the value after reset is returned.
11	MPXE	Use of MPX 0: When upper-limit or lower-limit status is reported, the MPX function is not used. 1: When upper-limit or lower-limit status is reported, the MPX function is used.
10 to 8	MPXV[2:0]	MPX Value when Upper/Lower Limit Check Status Is Reported 000 to 111: The MPX value when upper-limit or lower-limit status is reported is stored.
7	ULE	Upper/Lower Limit Check Status 0: The upper-limit or lower-limit status is not reported. 1: The upper-limit or lower-limit status is reported.
6	Reserved	When read, the value after reset is returned.
5 to 0	ULPC[5:0]	Upper/Lower Limit Check Status Capture The first physical channel number in which upper-limit or lower-limit status was reported is output.

CAUTIONS

1. The upper/lower limit threshold value check result is incorporated in this register.
2. This register value is retained until 1 is written to the scan group x upper/lower limit check status clear bit (ADCJnSGULCCR.SGULCC[x]). This register is set simultaneously when the A/D conversion result is stored in the data register (ADCJnDRj or ADCJnPWDDR).
3. When setting timing conflicts with clearing timing, this register operates as follows:
   If a conflict occurs when UE = 0, setting takes precedence (UE = 1, all status set).
   If a conflict occurs when UE = 1, clearing takes precedence (UE = 0, all status clear).
   These operations also apply to LE/MPXE/MPXV[2:0], and ULE/ULPC[6:0].
4. When PWM-Diag is used, ADCJnSGULCRx of target scan group is upper/lower limit check status of PWM-Diag.

43.3.6.18 ADCJnVCLMSCR1 — Upper/Lower Limit Check Status Clear Register 1

ADCJnVCLMSCR1 is a write-only register to clear the upper/lower limit check status register 1 (ADCJnVCLMSR1). This register is always read as 0.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <ADCJn_base> + $8A0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC31LMSC	VC30LMSC	VC29LMSC	VC28LMSC	VC27LMSC	VC26LMSC	VC25LMSC	VC24LMSC	VC23LMSC	VC22LMSC	VC21LMSC	VC20LMSC	VC19LMSC	VC18LMSC	VC17LMSC	VC16LMSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC15LMSC	VC14LMSC	VC13LMSC	VC12LMSC	VC11LMSC	VC10LMSC	VC09LMSC	VC08LMSC	VC07LMSC	VC06LMSC	VC05LMSC	VC04LMSC	VC03LMSC	VC02LMSC	VC01LMSC	VC00LMSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 43.78 ADCJnVCLMSCR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMSC	Upper/Lower Limit Check Status Clear 0: No function (Writing 0 to this bit is ignored.) 1: Corresponding status bits in the upper/lower limit check status register 1 (ADCJnVCLMSR1) are cleared.

NOTE

j = 00 to 31

43.3.6.19 ADCJnVCLMSCR2 — Upper/Lower Limit Check Status Clear Register 2

ADCJnVCLMSCR2 is a write-only register to clear the upper/lower limit check status register 2 (ADCJnVCLMSR2). This register is always read as 0.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <ADCJn_base> + $8A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VC63LMSC	VC62LMSC	VC61LMSC	VC60LMSC	VC59LMSC	VC58LMSC	VC57LMSC	VC56LMSC	VC55LMSC	VC54LMSC	VC53LMSC	VC52LMSC	VC51LMSC	VC50LMSC	VC49LMSC	VC48LMSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VC47LMSC	VC46LMSC	VC45LMSC	VC44LMSC	VC43LMSC	VC42LMSC	VC41LMSC	VC40LMSC	VC39LMSC	VC38LMSC	VC37LMSC	VC36LMSC	VC35LMSC	VC34LMSC	VC33LMSC	VC32LMSC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 43.79 ADCJnVCLMSCR2 Register Contents

Bit Position	Bit Name	Function
31 to 0	VCjLMSC	Upper/Lower Limit Check Status Clear 0: No function (Writing 0 to this bit is ignored.) 1: Corresponding status bits in the upper/lower limit check status register 2 (ADCJnVCLMSR2) are cleared.

NOTE

j = 32 to 63

43.3.6.20 ADCJnPWVCLMSCR — PWM-Diag Upper/Lower Limit Check Status Clear Register

ADCJnPWVCLMSCR is a write-only register to clear the PWM-Diag upper/lower limit check status register (ADCJnPWVCLMSR). This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + $8AC_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWVCLMSC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.80 ADCJnPWVCLMSCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	PWVCLMSC	PWM-Diag Upper/Lower Limit Check Status Clear 0: No function (Writing 0 to this bit is ignored.) 1: The PWVCLMS bit in ADCJnPWVCLMSR is cleared.

43.3.6.21 ADCJnSGULCCR — Scan Group Upper/Lower Limit Check Status Clear Register

ADCJnSGULCCR is a write-only register to clear upper/lower limit check status register of each scan group.
This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + $8B4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	SGULCC[4:0]					—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	W	W	W	W	W	R

Table 43.81 ADCJnSGULCCR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When writing, write the value after reset.
5 to 1	SGULCC[4:0]	Scan Group x Upper/Lower Limit Check Status Clear SGULCC[x] 0: No function (Writing 0 to this bit is ignored.) 1: Scan Group x upper/lower limit check status register (SGULCRx) is cleared.
0	Reserved	When writing, write the value after reset.

43.3.6.22 ADCJnVCLMASCR — Virtual Channel Upper/Lower Limit Check All Status Clear Register

ADCJnVCLMASCR is a write-only register to clear virtual channel upper/lower limit check status registers. This register is always read as 0.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <ADCJn_base> + $8B8_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ALLMSC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 43.82 ADCJnVCLMASCR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ALLMSC	Virtual Channel Upper/Lower Limit Check Status (*) Clear 0: No function (Writing 0 to this bit is ignored.) 1: All upper/lower limit check status registers are cleared. (*) All upper/lower limit check status registers are listed below. **Bit name** ADCJnVCLMSR1-2.VCnLMS ADCJnPWVCLMSR.PWVCLMS

43.4 Function

43.4.1 Setting Procedure

43.4.1.1 Initial Setting Procedure

Figure 43.7 shows the initial setting flow of this product. Settings of each register (except for ADCJnSGCR0 to ADCJnSGCR4, and ADCJnPWDSGCR) after reset until startup trigger input are arranged in a random order.

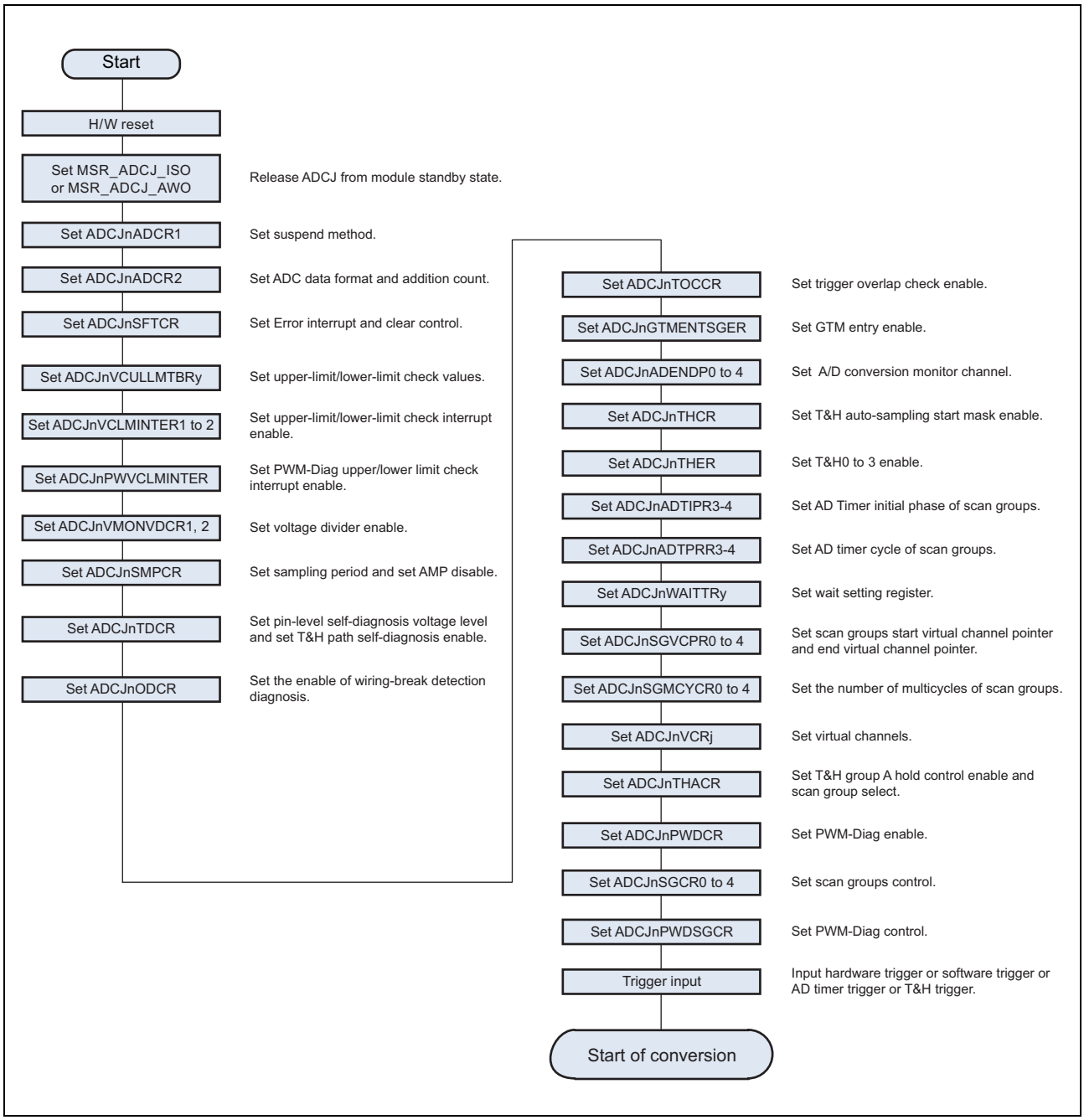


Figure 43.7 Initial Setting Flow

43.4.2 Startup Method

43.4.2.1 SG0 Startup Flow

Figure 43.8 shows the startup flow of SG0.

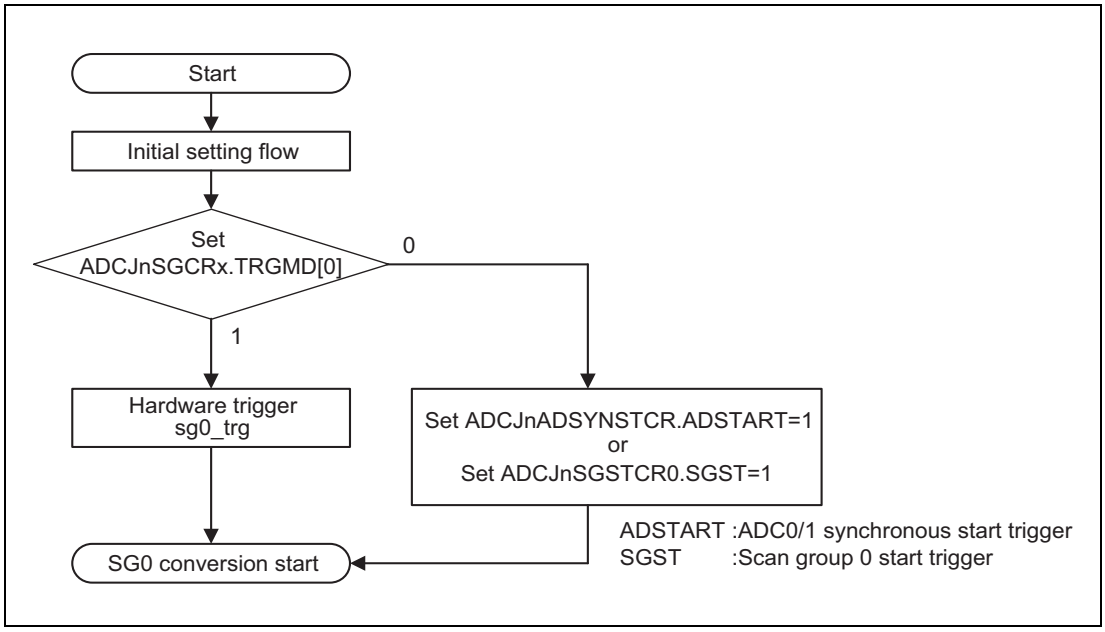


Figure 43.8 Startup Flow (SG0)

43.4.2.2 SG1/SG2 Startup Flow

Figure 43.9 shows the startup flow of SG1 and SG2.

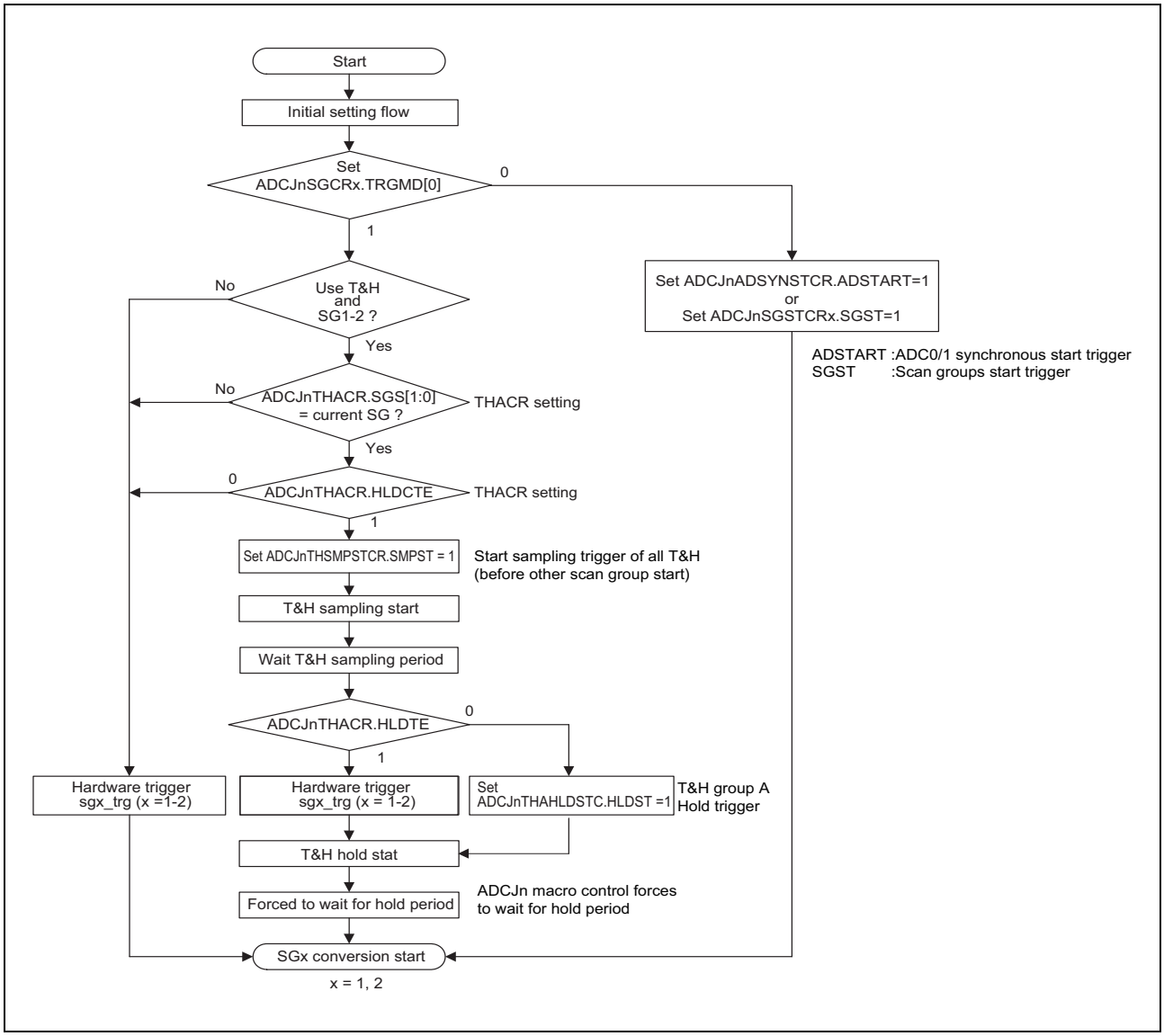


Figure 43.9 Startup Flow (SG1 and SG2)

NOTE

When SGx is started individually, There are specification restrictions in ADCJnTHSMPSTCR.SMPST. Refer to **Section 43.3.5.12, ADCJnTHSMPSTCR — T&H Sampling Start Control Register** for details.

43.4.2.3 SG3/SG4 Startup Flow

Figure 43.10 shows the startup flow of SG3 and SG4.

When the PWM-Diag is used, the following figure shows the startup flow of SG3.

For the PWM-Diag startup flow, see "**Section 43.4.2.4, PWM-Diag Startup Flow**".

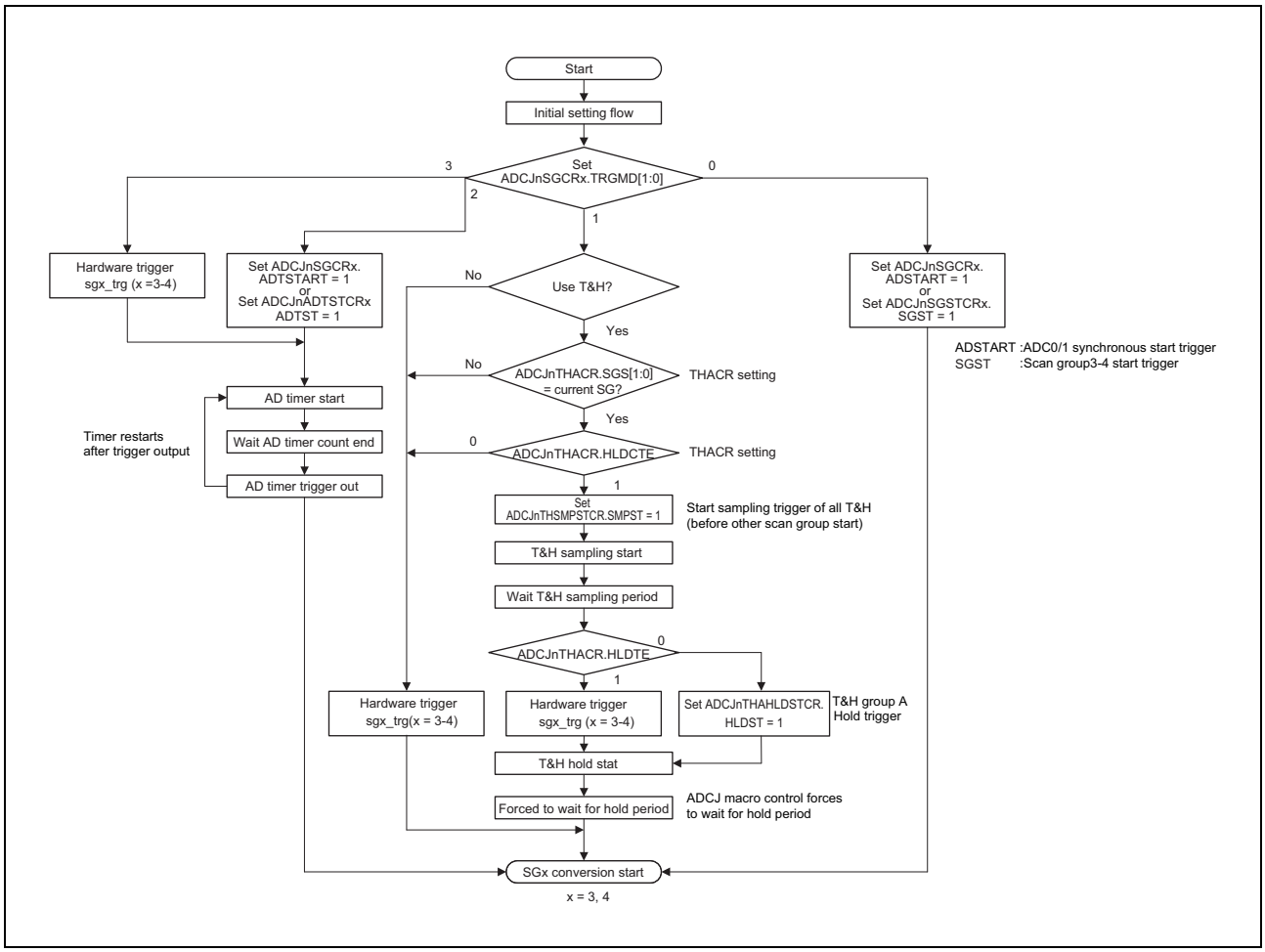


Figure 43.10 Startup Flow (SG3 and SG4)

NOTES

1. When using T&H, be sure to set ADCJnSGCR3-4.TRGMD to 1.
2. When using T&H, setting ADCJnSGCR3-4.TRGMD to 0, 2, or 3 is prohibited.
3. When SGx is started individually, there are specification restrictions in ADCJnTHSMPSTCR.SMPST.
   Refer to **Section 43.3.5.12, ADCJnTHSMPSTCR — T&H Sampling Start Control Register** for details.

43.4.2.4 PWM-Diag Startup Flow

Figure 43.11 shows the startup flow of the PWM-Diag.

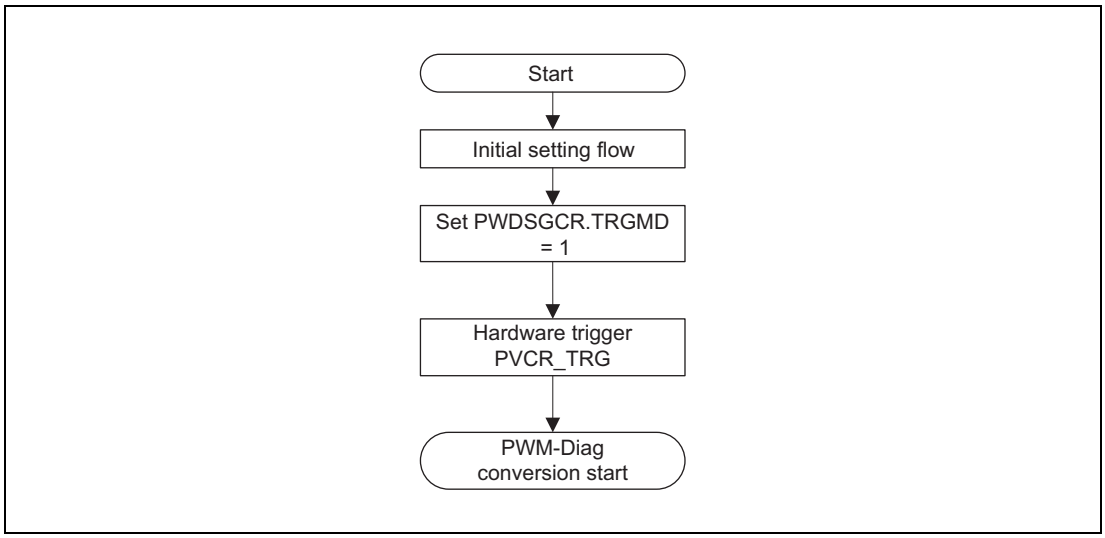


Figure 43.11 Startup Flow (PWM-Diag)

43.4.3 Termination Procedure

43.4.3.1 Forced Termination Flow of All Scan Groups

Figure 43.12 shows the forced termination flow of all scan groups.

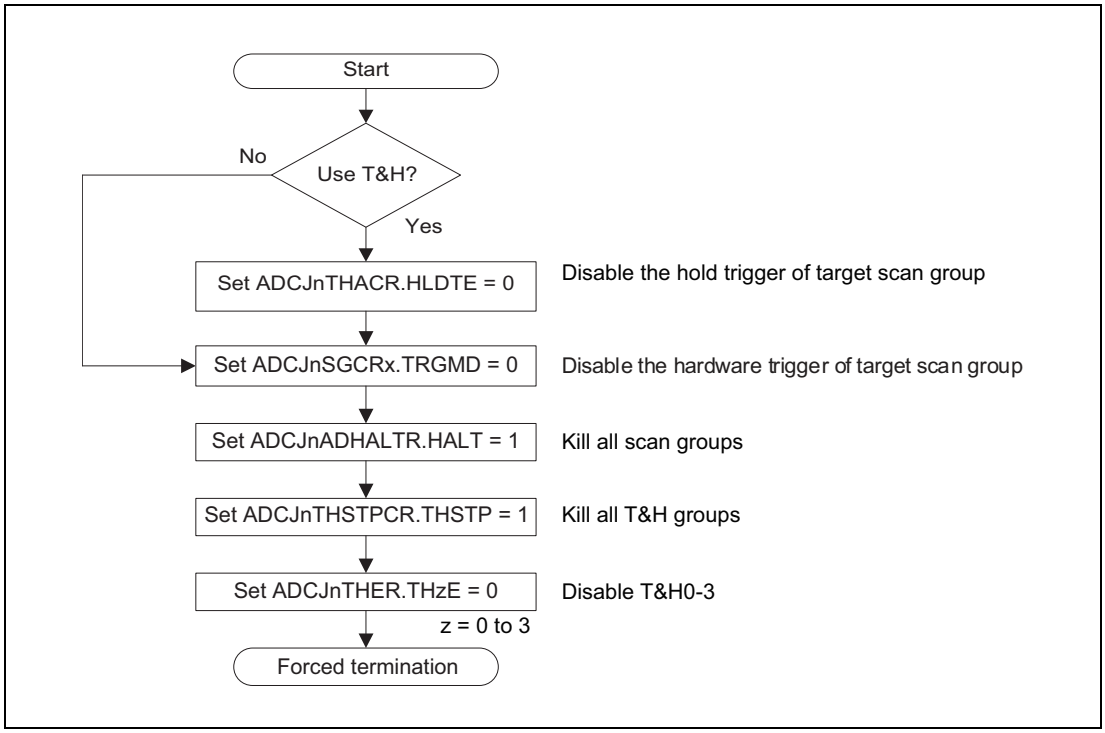


Figure 43.12 Forced Termination Flow of All Scan Groups

43.4.3.2 Individual Stopping Flow of Scan Group x

This section describes the flow to individually stop scan group x. To stop two or more scan groups, perform the following stopping flow for each scan group.

When scan group x is set for group A of T&H, the hold trigger of the target T&H group and the T&H channel of the target T&H group must be disabled.

(1) Individual Stopping Flow of Scan Group x (1)

Only multicycle scan mode is available in the stopping flow below. When an activate scan group is stopped in continuous scan mode, use the flow of **Section 43.4.3.1, Forced Termination Flow of All Scan Groups** or **Section (2), Individual Stopping Flow of Scan Group x (2)**.

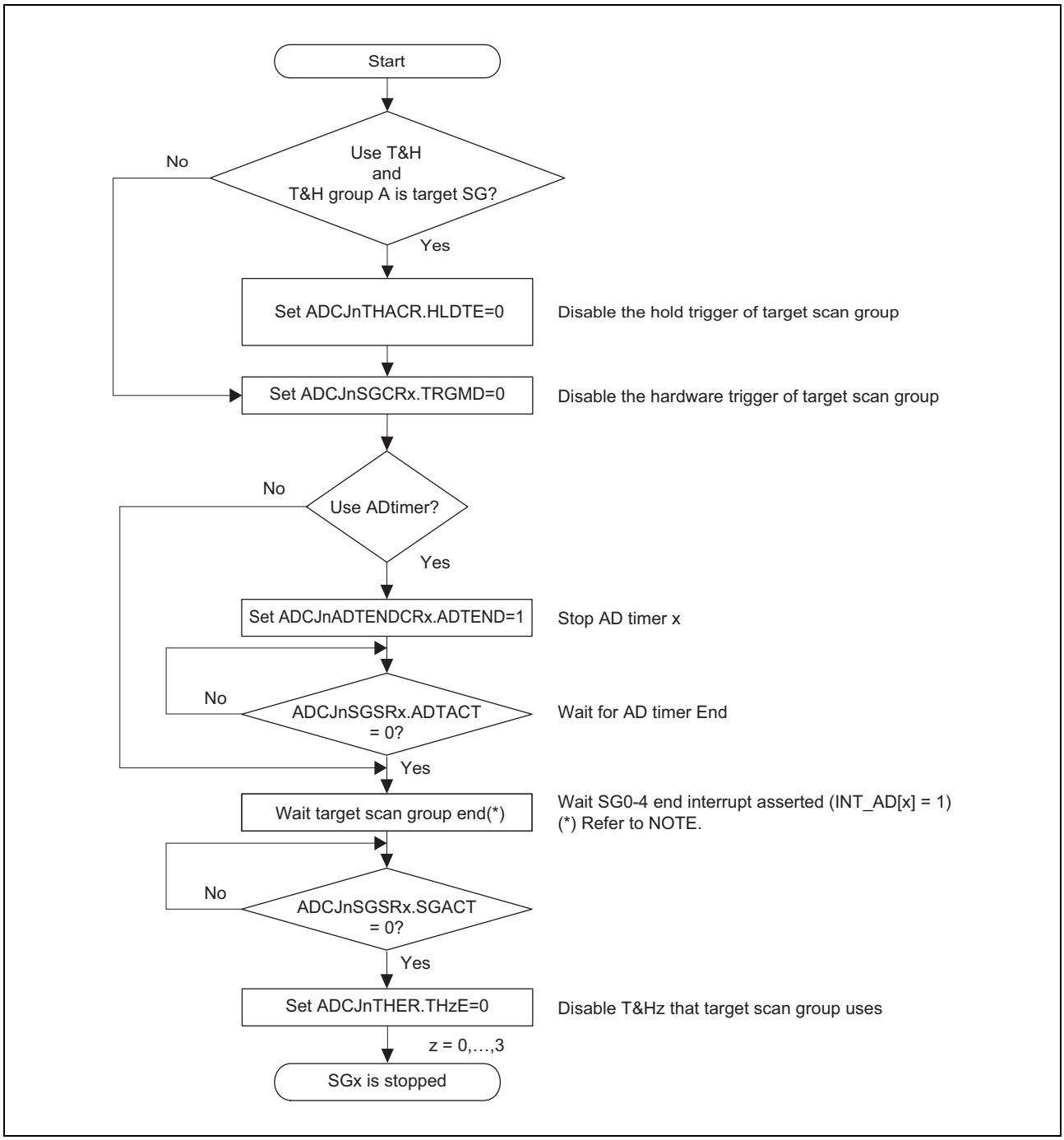


Figure 43.13 Individual Stopping Flow of Scan Groups (1)

NOTES

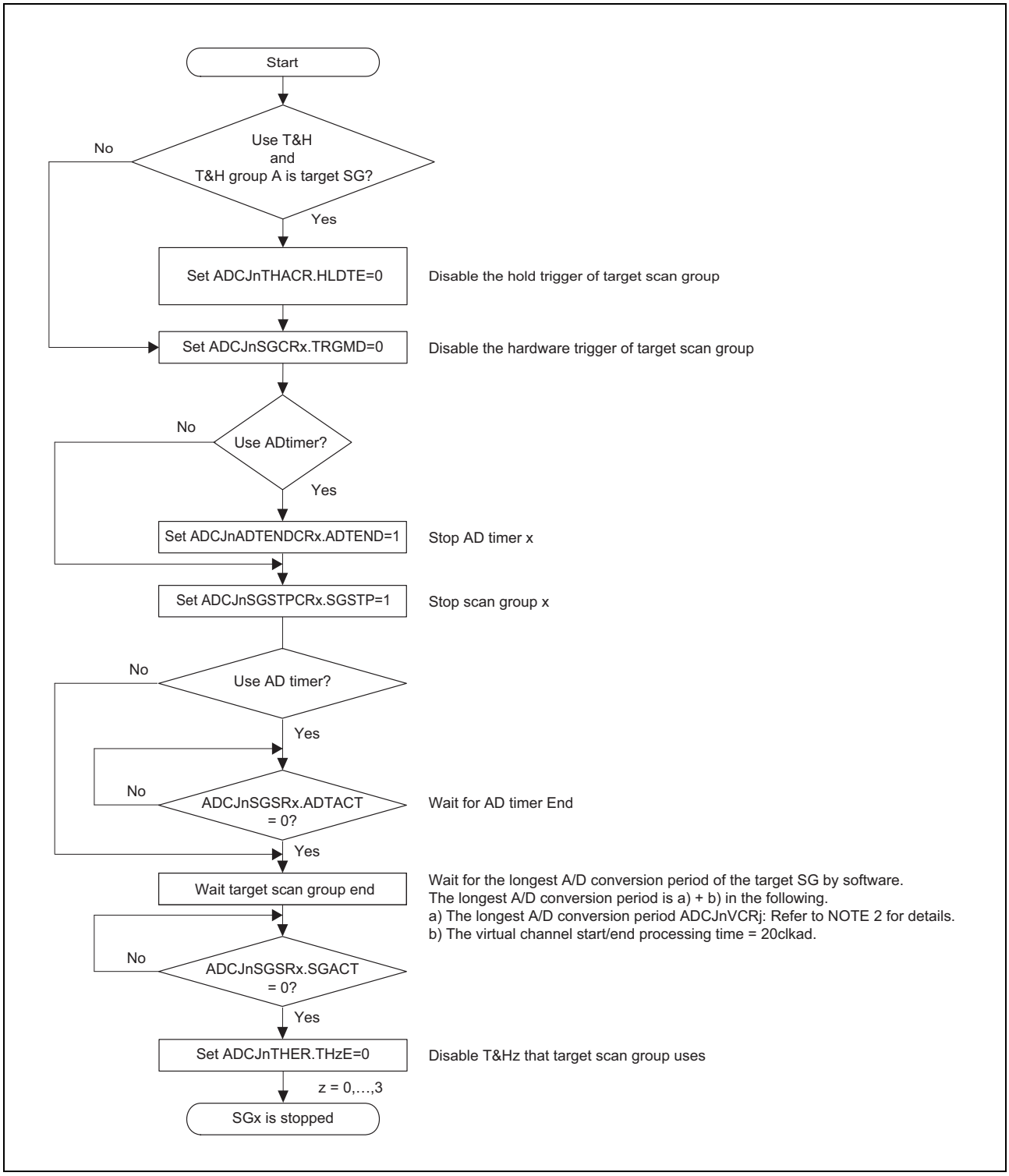
1. When the scan group x end interrupt is not used, wait time for the end of target scan group depends on user system. Because scan group has the priority. For example, in the case of scan group 4, when you read ADCJnSGSR4.SGACT=0 after you wait the period of all virtual channels set by virtual channel pointer, it indicates the end of scan group 4. But in the case of low priority scan group, when a high priority scan group is interrupted, the wait time is the time until the high scan group is end, and further more, the processing time of stopped virtual channel of the low priority scan group is added to the wait time in the case of asynchronous suspend mode. And when a high priority scan group is started again after the low priority can group is resumed, the wait time is added more. So, when this individual stopping flow of scan group x is selected, take into account the input interval of start triggers of each scan group.
2. About operation of suspend processing, refer to **Section 43.4.17, Example of Suspend and Resume Operation** for details.

(2) Individual Stopping Flow of Scan Group x (2)

Both multicycle scan mode and continuous scan mode are available in the stopping flow below.

When stopping scan group x in this flow, write 1 to SGSTP to stop the ongoing virtual channel processing.

Then the scan group x is stopped.



NOTES

1. If SGSTP is set to 1 before A/D conversion is completed, the target scan group enters the IDLE state but an A/D conversion end interrupt or a scan end interrupt may be output depending on register settings. In case interrupt output affects the system, set 0 to interrupt enable bits (ADCJnVCRj.ADIE, ADCJnSGCRx.ADIE) and then, set SGSTP to 1.
2. About the longest A/D conversion period of ADCJnVCRj.
   A/D conversion period is defined as tWAIT + tSPL + tSAR.

   tWAIT: Wait time of "wait (1) and wait (2)". About "wait (1) and wait (2)", refer to **Section 43.4.11, Wait Function**.
   tSPL, tSAR: Refer to **Section 43.4.24, Analog Input Sampling and Scan Group Processing Time**.

43.4.3.3 Individual Stopping Flow of PWM-Diag

Figure 43.15 shows the flow to stop the PWM-Diag.

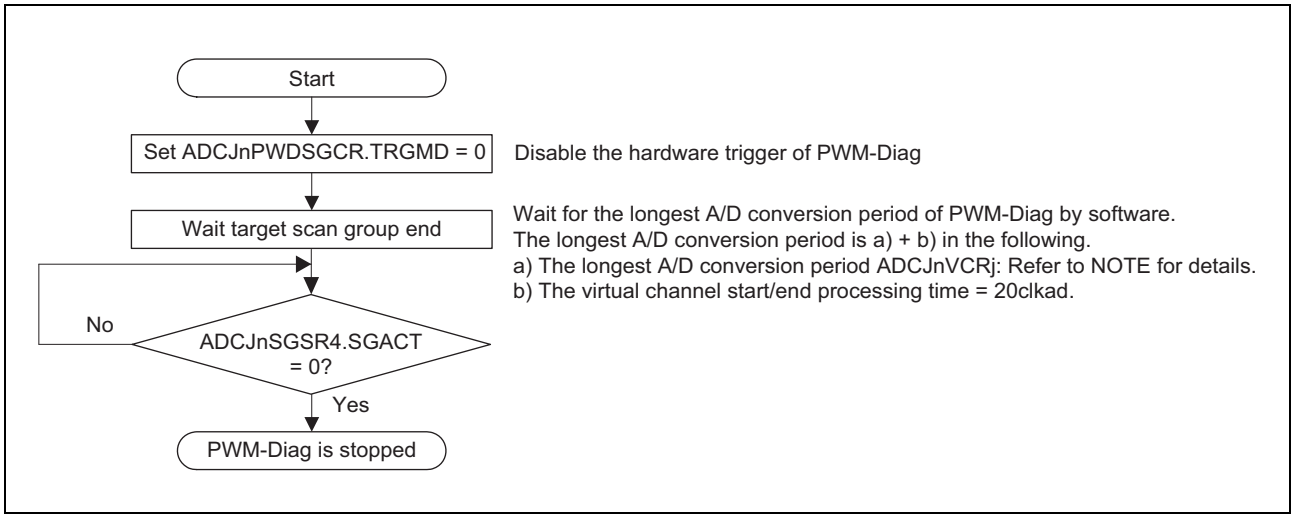


Figure 43.15 Individual Stopping Flow of PWM-Diag

NOTE

About the longest A/D conversion period of ADCJnVCRj, refer to **Section 43.4.24, Analog Input Sampling and Scan Group Processing Time** for details.

43.4.4 Restrictions on Updating Settings

Restrictions on setting updatable period are provided for registers shown in the initial setting flow.

Updating method

Stop scan group → Update register settings → Start scan group

If a scan group to be stopped uses hardware triggers, disable them before stopping the scan group.

Updatable periods vary with register as shown in the table below.

Perform the specified stopping procedure, and then update register settings according to the table below.

Table 43.83 Register Updatable Period for Initial Setting Flow (1/2)

Register	Bit	Updatable Period
ADCJnADCR1	All bits	While all scan groups are stopped
ADCJnADCR2		
ADCJnSFTCR		
ADCJnTOCCR		
ADCJnODCR		
ADCJnSMPCR		
ADCJnMPXCURCR		
ADCJnADENDPv		
ADCJnTDCR	All bits	While all scan groups are stopped (Only in the flow of Pin-level Self-Diagnosis of the T & H path which explained in **Section 43.4.10.2, Self-Diagnosis of the T&H Path**, TDLV is updatable in spite of scan group operating)
ADCJnWAITTRy	All bits	While all scan groups are stopped
ADCJnVCLMINTER1/2		
ADCJnSGCRx	SCANMD, ADSTARTE, ADTSTARTE, ADIE/TRGMD set	While relative scan group x is stopped
	ADIE clear	Always updatable
	TRGMD clear	
ADCJnTHER	THzE (Used by T&H group that is set to target scan group x)	While relative scan group x is stopped
ADCJnTHACR	HLDCTE, SGS[1:0], HLDTE set	
	HLDTE clear	Always updatable
ADCJnSGMCYCRx	All bits	While relative scan group x is stopped
ADCJnADTIPRx	All bits	While relative A/D timer x is stopped
ADCJnADTPRRx		
ADCJnVCRj	VCULLMTBS, WTTS, GTMENT, GTMTAG, CNVCLS, MPXV, GCTRL, ADIE set	While relative scan group x is stopped
	ADIE clear	While relative scan group x is stopped. (There is an exception. Refer below paragrapth)
ADCJnTHCR	All bits	While relative scan group x is stopped
ADCJnSGVCPRx	All bits	Always updatable
ADCJnVCULLMTBRy		

Table 43.83 Register Updatable Period for Initial Setting Flow (2/2)

Register	Bit	Updatable Period
ADCJnGTMENTSGER	All bits	While all scan groups are stopped
ADCJnVMONVDCR1/2		
ADCJnMPXINTER		
ADCJnMPXCMDR	All bits	Always updatable
ADCJnPWVCLMINTER	All bits	While all scan groups are stopped
ADCJnPWDCR		
ADCJnPWDSGCR	TRGMD set	While relative scan group x is stopped
	TRGMD clear	Always updatable
ADCJnTHSMPSTCR	SMPST	While all scan groups are stopped

ADCJnSGVCPRx, ADCJnVCULLMTBRy, ADCJnMPXCMDR, ADCJnMPXINTER, ADCJnVCLMINTER1-2 and ADCJnPWVCLMINTER are always updatable regardless of whether scan groups are operating or stopped.

Updated setting values are incorporated at the following timing.

Register	Updated setting timing
ADCJnSGVCPRx	Next startup of the relevant scan group x
ADCJnVCULLMTBRy, ADCJnMPXINTER, ADCJnVCLMINTER1-2, ADCJnPWVCLMINTER	Immediately
ADCJnMPXCMDR	At the beginning of MPX mode A/D conversion or MPX addition mode A/D conversion

There is a exception to constraints clearing ADCJnVCRj.ADIE.

When a scan group is stopped according to **Section 43.4.3.2, (2) Individual Stopping Flow of Scan Group x (2)**, and when it is necessary to clear ADCJnVCRj.ADIE, clear ADCJnVCRj.ADIE according to NOTE 1 of **Section 43.4.3.2, (2) Individual Stopping Flow of Scan Group x (2)**.

43.4.5 Conversion Class

There are 15 conversion types. Set them for each virtual channel.

Table 43.84 Conversion Class List (1/2)

ADCJnVCRj.CNVCLS	Conversion Class	Description	Reference Section
0_H	Normal A/D conversion	Performs A/D conversion of analog inputs of physical channels.	**43.4.6**
1_H	Hold value A/D conversion	Performs A/D conversion of analog inputs held by the T&H function. Use this A/D conversion together with the T&H function.	**43.4.18**
2_H	Normal A/D conversion at extended sampling cycle	Performs A/D conversion of analog inputs of physical channels at the sampling cycle specified by the EXSMPT[7:0] bits in ADCJnSMPCR.	**43.4.11.3**
3_H	ADcore self-diagnosis A/D conversion	This A/D conversion is one of safety functions, which isolates physical channels and applies the predetermined voltage to analog input pins of the ADcore hardware macro to perform A/D conversion.	**43.4.10.3**
4_H	Addition mode A/D conversion	Performs A/D conversion of analog inputs of physical channels twice or four times in series, and then stores added values in the data register. The addition count (twice or four times) are common to all virtual channels.	**43.4.7**
5_H	MPX normal A/D conversion	This function selects an MPX input channel (MPX value) and performs A/D conversion in physical channels connected to an external MPX.	**43.4.8**
6_H	MPX addition mode A/D conversion	Selects an MPX input channel (MPX value) and performs A/D conversion twice or four times in series, and then stores added values in the data register in physical channels connected to an external MPX.	**43.4.8**
7_H	Pin level self-diagnosis A/D conversion	This function applies different voltage values to even channel groups and odd channel groups to perform A/D conversion. Thus this function diagnoses errors in the analog input path.	**43.4.10.1**
8_H	A/D conversion in wiring-break detection mode 1	This function detects solder stripping. This function pulls down physical channels ANnpq and performs normal A/D conversion to decide whether wiring-break is present with the pull-down result of nearly 0 V or not.	**43.4.9.1**
9_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)	This function detects solder stripping. This function pulls down physical channels ANnpq and performs normal A/D conversion to decide whether wiring-break is present with the pull-down result of nearly 0 V or not.	**43.4.9.2**
A_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)	This function detects solder stripping. This function pulls up physical channels ANnpq and performs normal A/D conversion to decide whether wiring-break is present with the pull-up result of nearly 5 V or not.	**(2)**
B_H	Self-diagnosis A/D conversion in wiring-break detection mode 1	This A/D conversion is a self-diagnostic function to verify that wiring-break detection mode 1 operates normally.	**(1)**
C_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)	This A/D conversion is a self-diagnostic function to verify that wiring-break detection mode 2 A/D conversion (physical channel IO pull-down) operates normally.	**(2)**

Table 43.84 Conversion Class List (2/2)

ADCJnVCRj.CNVCLS	Conversion Class	Description	Reference Section
D_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)	This A/D conversion is a self-diagnostic function to verify that wiring-break detection mode 2 A/D conversion (physical channel IO pull-up) operates normally.	**(2)**
E_H	A/D conversion of A/D conversion data path diagnosis (ADVAL mode)	The A/D conversion data path diagnosis checks the data path until data is stored in the data register from the ADCore.	**43.4.10.6**
F_H	Reserved	—	—

43.4.6 Examples of Normal A/D Conversion Operation

Analog inputs of physical channels are A/D-converted in normal A/D conversion.

For A/D conversion operations, see **Section 43.4.12, Scan Mode**.

43.4.7 Example of Addition Mode A/D Conversion Operation

In addition mode A/D conversion, analog signals of physical channels are A/D-converted twice or four times in series, and added values are stored in the data register. The addition count (twice or four times) is common to all virtual channels.

Figure 43.16 shows an example of A/D conversion operation.

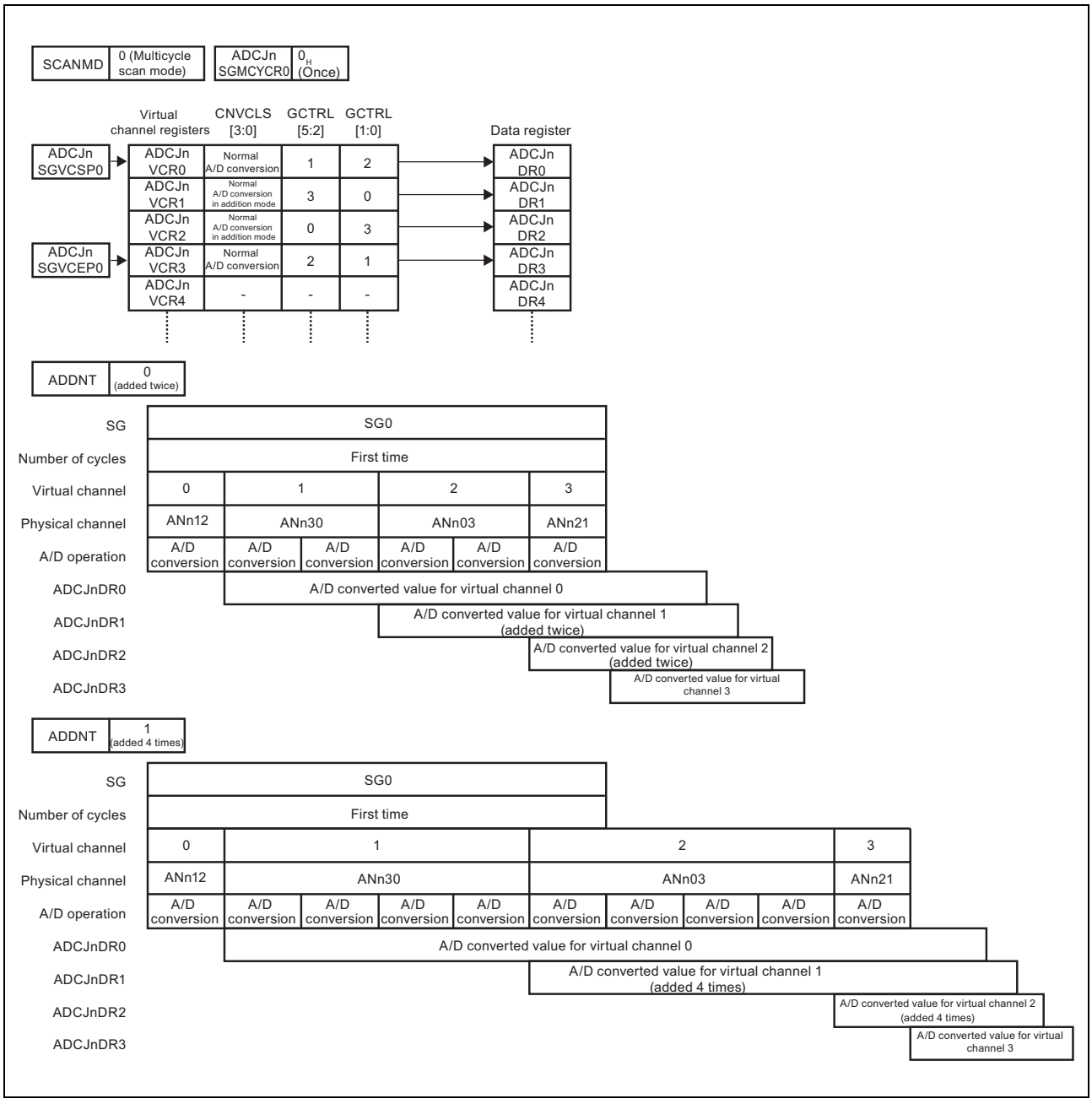


Figure 43.16 Example of Addition Mode Normal A/D Conversion Operation

For operation if a synchronous suspension or asynchronous suspension occurs while addition mode A/D conversion is in progress, see "**Section 43.4.17, Example of Suspend and Resume Operation**" For operation of suspension caused by a hold trigger, see case 4 and case 5 in "**Section 43.4.18, Example of Track and Hold Operation (T&H)**."

43.4.8 Normal A/D Conversion with the MPX and Normal A/D Conversion with the MPX in Addition Mode

This function selects an MPX input channel (MPX value) in physical channels connected to an external MPX and performs A/D conversion. This function includes MPX normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 5_H) and MPX addition mode A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 6_H).

To transfer the MPX value to the external MPX, functions (such as DMAC and I/O port MSPI) and the output port MPXCUR[2:0] are used.

Set the wait time to ensure the MPX value transfer period and the MPX output settling period.

For how to set the wait time, see "**Section 43.3.5.18, ADCJnWAITTRy — Wait Setting Register y**".

When MPX normal A/D conversion or MPX addition mode A/D conversion is in progress or when a synchronous suspension or asynchronous suspension occurs during the wait state, ADCJ operates in the same way as when normal A/D conversion or addition mode A/D conversion is in progress or when a synchronous suspension or asynchronous suspension occurs during the wait state in normal A/D conversion. For details, see "**Section 43.4.17, Example of Suspend and Resume Operation**."

For operation of suspension caused by a hold trigger, see case 4 and case 5 in "**Section 43.4.18, Example of Track and Hold Operation (T&H)**"

NOTES

1. MPX value transfer period is the period from starting MPX A/D conversion to output INT_MPX.
2. MPX output setting period is the period from the INT_MPX output to the MPX value is fixed at the external Multiplexer and the analog voltage input is stabilized. Specify the period after evaluating by user.

43.4.8.1 Using I/O Port Output

An example of configuration using DMA and I/O port output and an example of operation are shown below.

Set the following registers according to each function and IP specifications.

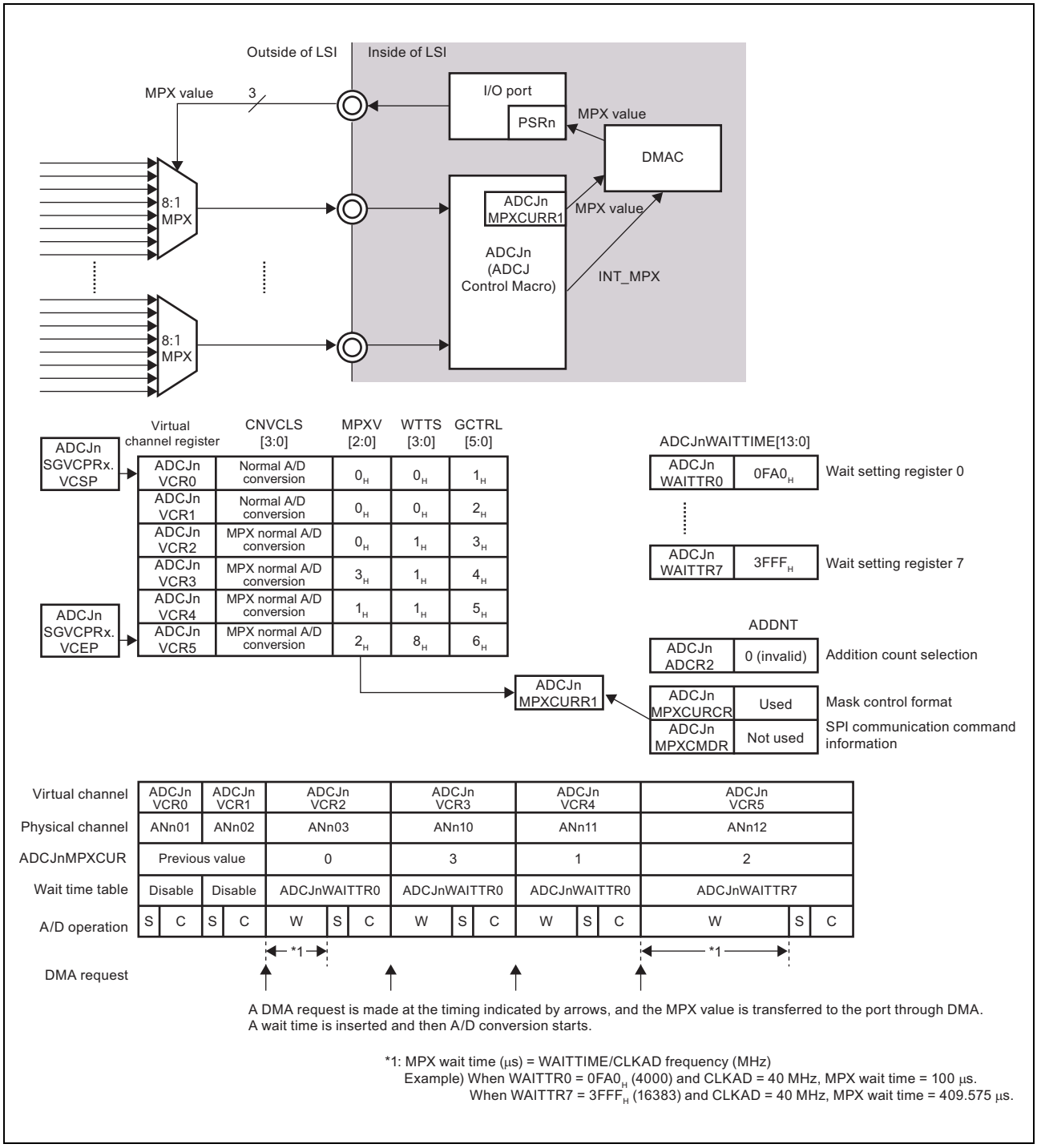
- MPX current control register (ADCJnMPXCURCR)
- External MPX channel (ADCJnVCRj.MPXV[2:0])
- Wait time table selection value (ADCJnVCRj.WTTS[3:0])
- Physical channel group/sub channel number (ADCJnVCRj.GCTRL[5:0])
- Wait setting register y (ADCJnWAITTRy.WAITTIME[13:0])
- MPX interrupt enable register (ADCJnMPXINTER)

The ADCJ performs the following operations.

1. Stores the ADCJnMPXCURCR and ADCJnVCRj.MPXV[2:0] values in the MPX current register 1 (ADCJnMPXCURR1) at the beginning of virtual channels and outputs the interrupt signal (INT_MPX).
2. Inserts arbitrary wait time using the wait setting value (ADCJnWAITTRy.WAITTIME[13:0]) selected by the WTTS[3:0] bits in ADCJnVCRj before executing virtual channels.
   (Waiting for MPX value transfer by the DMAC, I/O port, etc. and settling of MPX output)
3. Perform A/D conversion of physical channels connected to the MPX.

Set the mask setting value of the MPX current control register according to the I/O port specifications.

This set value is used for I/O port output, but is not used for MSPI output.



A DMA request is made at the timing indicated by arrows, and the MPX value is transferred to the port through DMA.
A wait time is inserted and then A/D conversion starts.

*1: MPX wait time (μs) = WAITTIME/CLKAD frequency (MHz)
Example) When WAITTR0 = $0FA0_H$ (4000) and CLKAD = 40 MHz, MPX wait time = 100 μs.
When WAITTR7 = $3FFF_H$ (16383) and CLKAD = 40 MHz, MPX wait time = 409.575 μs.

Figure 43.17 Example of Using MPX (Output to I/O Port)

43.4.8.2 Using MSPI Output

An example of configuration using DMA and MSPI output and an example of operation are shown below.

Set the following registers according to each function and IP specifications.

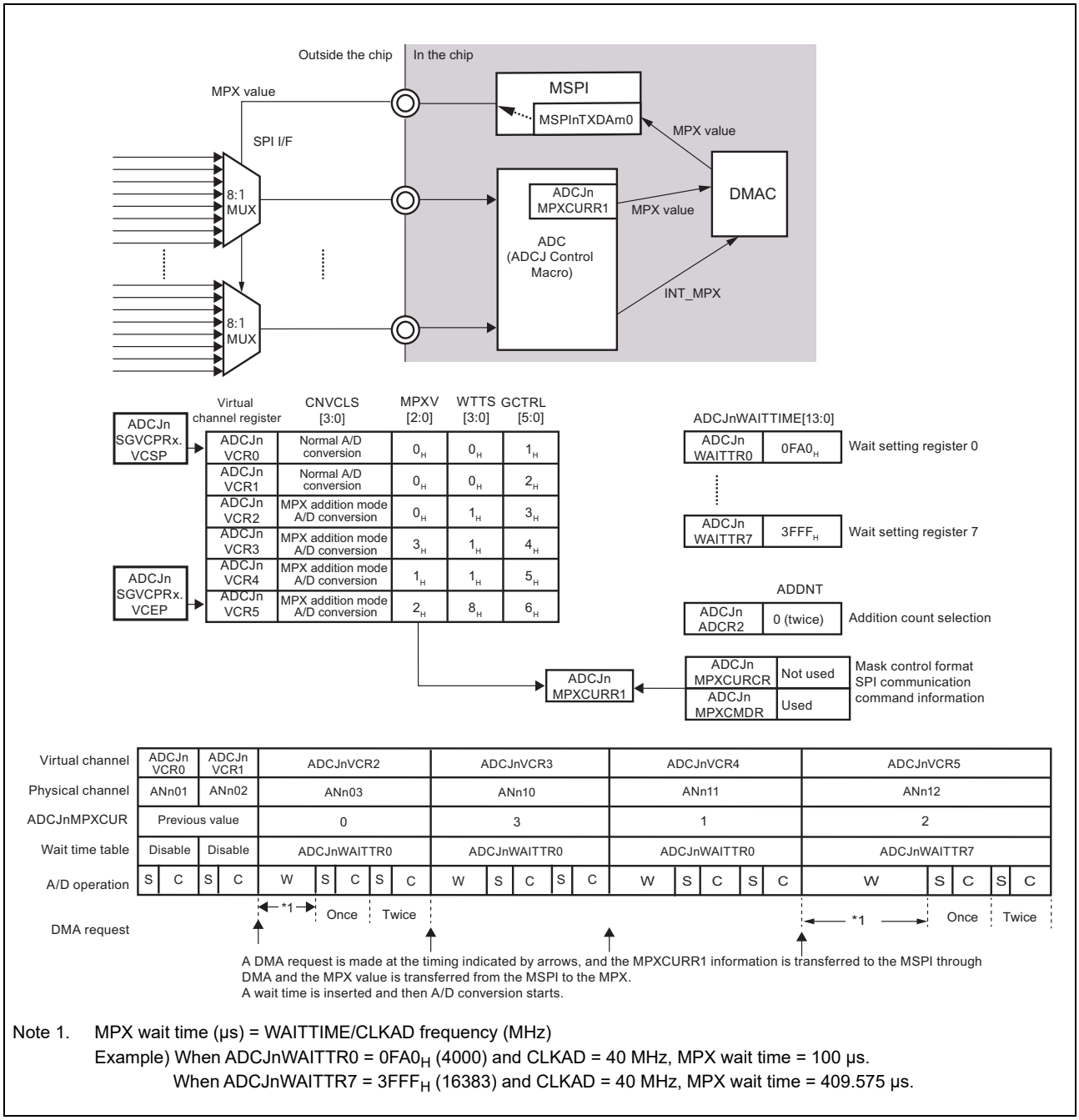
- MPX command information register (ADCJnMPXCMDR)
- External MPX channel (ADCJnVCRj.MPXV[2:0])
- Wait time table selection value (ADCJnVCRj.WTTS[3:0])
- Physical channel group/sub channel number (ADCJnVCRj.GCTRL[5:0])
- Wait setting register y (ADCJnWAITTRy.WAITTIME[13:0])
- MPX interrupt enable register (ADCJnMPXINTER)

The ADCJ performs the following operations.

1. Stores the ADCJnMPXCMDR and ADCJnVCRj.MPXV[2:0] values in the MPX current register 1 (ADCJnMPXCURR1) at the beginning of virtual channels and outputs the interrupt signal (INT_MPX).
2. Inserts arbitrary wait time using the wait setting value (ADCJnWAITTRy.WAITTIME[13:0]) selected by the WTTS[3:0] bits in ADCJnVCRj before executing virtual channels.
(Waiting for MPX value transfer by the DMAC, MSPI, etc. and settling of MPX output)
3. Perform A/D conversion of physical channels connected to the MPX.

Set the SPI communication command information of the MSPI in the MPX command information register.

This set value is used for MSPI output, but is not used for I/O port output.



Note 1. MPX wait time (µs) = WAITTIME/CLKAD frequency (MHz)
Example) When ADCJnWAITTR0 = $0FA0_H$ (4000) and CLKAD = 40 MHz, MPX wait time = 100 µs.
When ADCJnWAITTR7 = $3FFF_H$ (16383) and CLKAD = 40 MHz, MPX wait time = 409.575 µs.

Figure 43.18 Example of Using MPX (Output to MSPI)

43.4.8.3 Using Output Port

An example of configuration using the output port MPXCUR[2:0] and an example of operation are shown below.

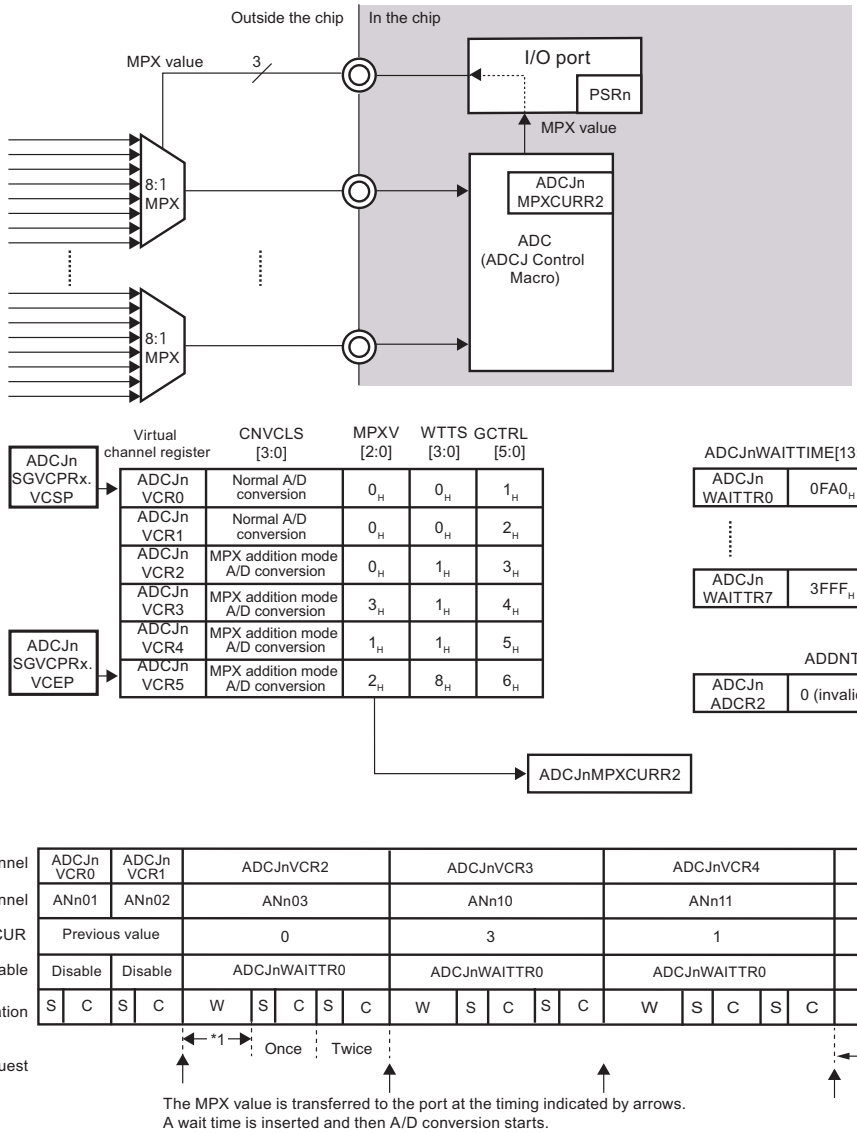
Set the following registers.

- External MPX channel (ADCJnVCRj.MPXV[2:0])
- Wait time table selection value (ADCJnVCRj.WTTS[3:0])
- Physical channel group/sub channel number (ADCJnVCRj.GCTRL[5:0])
- Wait setting register y (ADCJnWAITTRy.WAITTIME[13:0])

The ADCJ performs the following operations.

1. Stores the ADCJnVCRj.MPXV[2:0] value in the MPX current register 2 (ADCJnMPXCURR2) and sets the value for the output port MPXCUR[2:0] at the beginning of virtual channels.
2. Inserts arbitrary wait time using the wait setting value (ADCJnWAITTRy.WAITTIME[13:0]) selected by the WTTS[3:0] bits in ADCJnVCRj before executing virtual channels. (Waiting for setting MPX output)
3. Perform A/D conversion of physical channels connected to the MPX.

For operation when the MPX is enabled in the PWM-Diag, see **Section 43.4.16, PWM-Diag [Example of PWM-Diag]**.



Note 1. MPX wait time (µs) = WAITTIME/CLKAD frequency (MHz)
Example) When ADCJnWAITTR0 = $0FA0_H$ (4000) and CLKAD = 40 MHz, MPX wait time = 100 µs.
When ADCJnWAITTR7 = $3FFF_H$ (16383) and CLKAD = 40 MHz, MPX wait time = 409.575 µs.

Figure 43.19 Example of Using MPX (Output Port)

43.4.9 Diagnosis Functions

The ADCJ is equipped with the following three diagnostic functions.

- Wiring-break detection mode 1 (pull-down)
- Wiring-break detection mode 2 (pull-down or pull-up)
- Voltage monitoring function

43.4.9.1 Wiring-Break Detection Mode 1

Wiring-break detection mode 1 is a function to detect solder stripping.

Analog inputs are pulled down during sequential conversion, A/D conversion of wiring-break detection mode 1 is executed continuously several times for same physical channel. If wiring-break is present, the conversion result attenuates to near 0 V. This can be determined as wiring-break detection. For detailed timing, see **Figure 43.20, Wiring-Break Detection Mode 1 Flow**.

[Recommended flow]

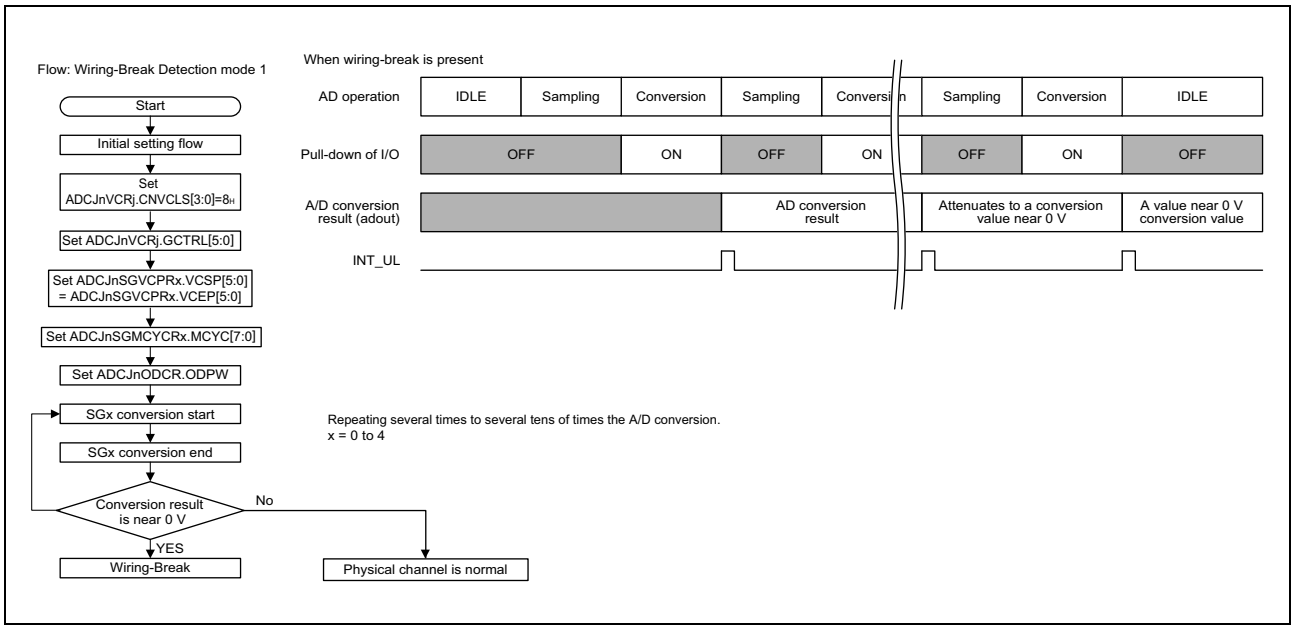


Figure 43.20 Wiring-Break Detection Mode 1 Flow

1. Make settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**).
2. In the Virtual Channel Register j (ADCJnVCRj), set CNVCLS[3:0] to 8H and set any physical channels in GCTRL[5:0].
3. Set only one virtual channel by Scan Group x Virtual Channel Pointer Register (ADCJnSGVCPRx). Set the number of virtual channel j to both VCSP[5:0] and VCEP[5:0].
4. Set the number of A/D conversion of wiring-break detection mode 1 to Scan Group x Multicycle register (ADCJnSGMCYCRx.MCYC).
5. In the wiring-break detection control register (ADCJnODCR), set any wiring-break detection pulse width in ODPW[4:0].
6. Activate the scan group x to perform A/D conversion of wiring-break detection mode 1.

7. Perform A/D conversion of wiring-break detection mode 1 several times. If the A/D conversion results attenuate to near 0 V, a wiring-break may be present.

CAUTIONS

1. **Execute the wiring-break detection for physical channel connected to T&H only when the T&H function is disabled.**
2. **When the normal A/D conversion result is near 0 V, it cannot determine wiring-break.**
3. **When performing normal A/D conversion after the wiring-break detection function is used, provide a sufficient charging period considering external stabilizing capacitor and signal source impedance.**
4. **When buffer amp disable is valid (ADCJnSMPCR.BUFAMPD = 1), it is prohibited to use wiring-break detection mode 1 because I/O pull-down is invalid during conversion time.**

43.4.9.2 Wiring-Break Detection Mode 2

Wiring-break detection mode 2 is a function to detect solder stripping.

Because analog inputs are pulled up or pulled down during sampling, the conversion result is affected by pull-up or pull-down. Therefore, when no wiring-break is present, the A/D conversion result cannot be used as a normal A/D conversion result value.

CAUTIONS

1. **Execute the wiring-break detection for physical channel connected to T&H only when the T&H function is disabled.**
2. **When performing normal A/D conversion after the wiring-break detection function is used, provide a sufficient charging period considering external stabilizing capacitor and signal source impedance.**

(1) Wiring-Break Detection Mode 2 (Pull-Down for physical channel I/O)

[Recommended flow]

The I/O buffer of the ANnpq pin to be detected is driven to the pull-down state.

Make the following settings in the virtual channel register j (ADCJnVCRj) according to the initial setting flow (**Section 43.4.1.1, Initial Setting Procedure**).

ADCJnVCRj number	CNVCLS[3:0]	GCTRL[5:0]
N	9_H	ANnpq number

Assign the virtual channel j shown above to scan group x.

Start scan group x according to the startup flow and perform A/D conversion.

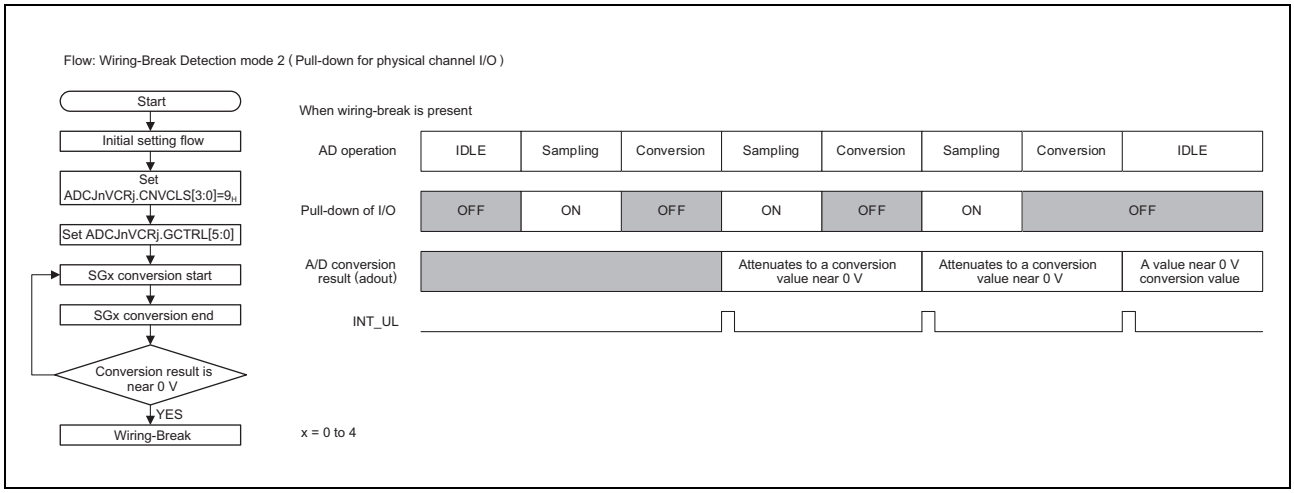


Figure 43.21 Recommended Flow of Wiring-Break Detection Mode 2 (Pull-Down for physical channel I/O)

[Decision method]

If the A/D conversion result attenuates to near 0 V, a wiring-break may be present.

We recommend that you perform A/D conversion several times and check the results.

For detailed timing, see "**Figure 43.21, Recommended Flow of Wiring-Break Detection Mode 2 (Pull-Down for physical channel I/O)**".

(2) Wiring-Break Detection Mode 2 (Difference Comparison)

This mode is usable to compare differences in A/D conversion results to detect wiring-break using wiring-break detection mode 2 (pull-down for physical channel I/O) and wiring-break detection mode 2 (pull-up for physical channel I/O).

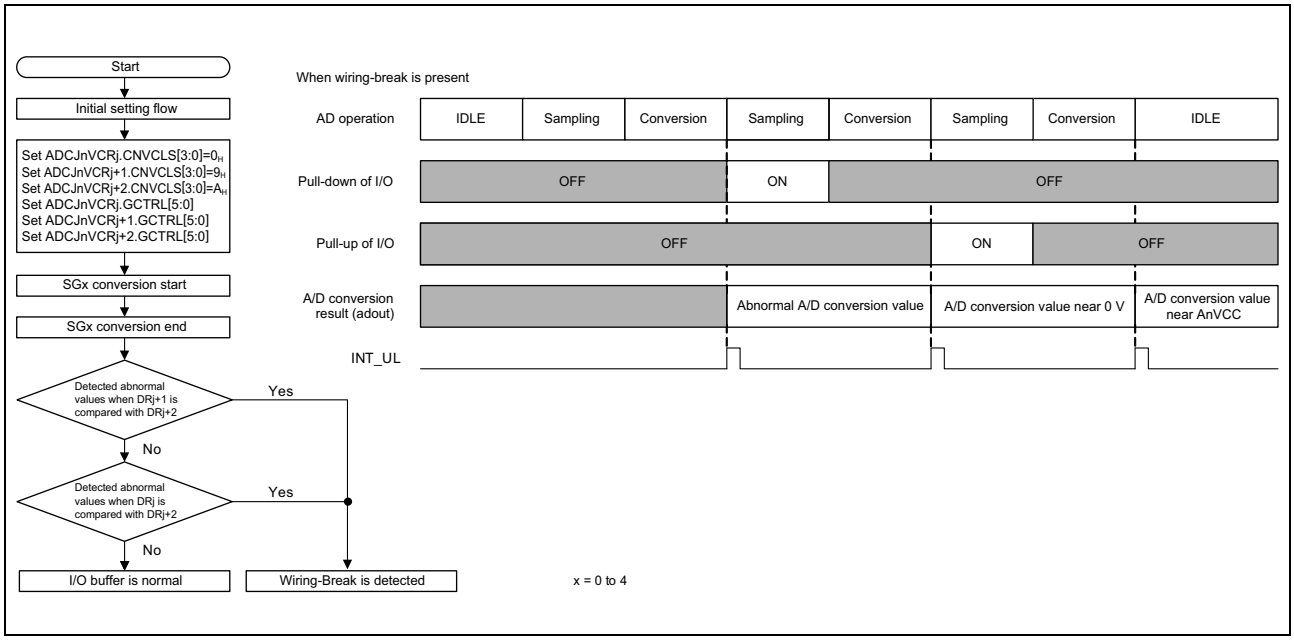


Figure 43.22 Recommended Flow of Wiring-Break Detection Mode 2 (Difference Comparison)

Set the I/O buffer of the ANnpq pin to be detected to the pull-up, pull-down, or pin connection state. Make the following settings for consecutive three virtual channels (ADCJnVCRj) according to "**Section 43.4.1.1, Initial Setting Procedure**".

I/O buffer	ADCJnVCRj number	CNVCLS[3:0]	GCTRL[5:0]
Pin connection	j	0_H	ANnpq number
Pull-down	j+1	9_H	
Pull-up	j+2	A_H	

Assign the set ADCJnVCRj to ADCJnVCRj+2 to the same scan group x.

Start scan group x according to the startup flow and perform A/D conversion.

[An Example of Decision method]

Compare:

1. between ADCJnVCRj +1 and ADCJnVCRj+2
2. between ADCJnVCRj and ADCJnVCRj+2

If wiring-break is present, the A/D conversion result of $ADCJnVCR_{j+1}$ attenuates to near 0 V and the A/D conversion result of $ADCJnVCR_{j+2}$ rises to near AnVCC. For this reason, abnormal values can be detected from the result of comparison 1 and 2 above. Thus, wiring-break can be decided. For detailed timing, see "**Figure 43.22, Recommended Flow of Wiring-Break Detection Mode 2 (Difference Comparison)**".

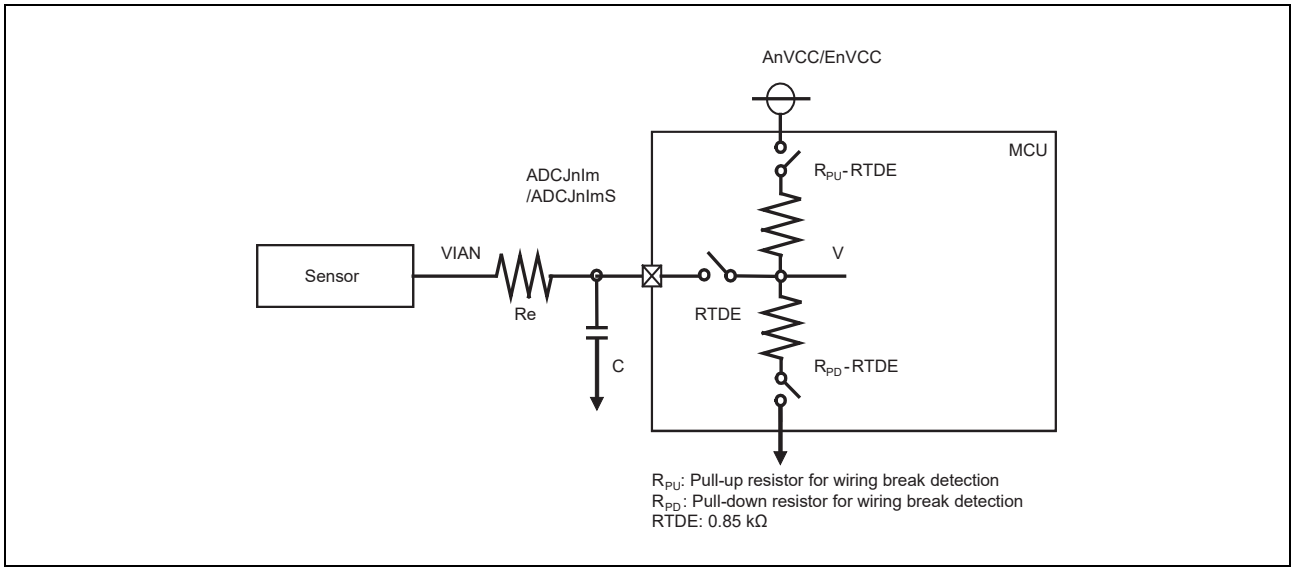


Figure 43.23 Schematic for Wiring Break Detection Formula

- $DR_{j+2} = V \times 4096 / AnVREFH = (AnVCC - (AnVCC - VIAN) * (R_{PU} - RTDE) / (Re + R_{PU})) * 4096 / AnVREFH$
- $DR_{j+1} = V \times 4096 / AnVREFH = (VIAN * (R_{PD} - RTDE) / (Re+R_{PD})) * 4096 / AnVREFH$

Because R_{PU} and R_{PD} depend on VIAN according to **Table 55.94, ADC Characteristics**, DR_{j+2} is maximum when R_{PU} is minimum and DR_{j+1} is minimum when R_{PD} is minimum.

The following is the example of formula for high accuracy pins (ADCJnlm).

If AnVCC = 5 V, AnVREFH = 5 V, VIAN = 2.5 V, Re = 1 kΩ, min $[R_{PU}]$@VIAN = AnVCC/2 is 5 kΩ, min $[R_{PD}]$@VIAN = AnVCC/2 is 5 kΩ,

- max $[DR_{j+2}] = (5 - (5 - 2.5) * (5 - 0.85) / (1 + 5)) * 4096 / 5 = 2680$ [LSB]
- min $[DR_{j+1}] = (2.5 * (5 - 0.85) / (1 + 5)) * 4096 / 5 = 1416$ [LSB]
- $DRj = 2.5 * 4096 / 5 = 2048$ [LSB]
- In comparison1, if $DR_{j+2} - DR_{j+1} > 1264$ (= 2680 - 1416) [LSB], wiring break is detected.
- In comparison2, if $DR_{j+2} - DR_j > 632$ (= 2680 - 2048) [LSB], wiring break is detected.

43.4.9.3 Secondary Power Supply Voltage Monitoring Function

The ADCJ0 can A/D-convert VCC, E0VCC, ISOVDD and AWOVDD power voltages. Setting the voltage divider enable bits 1 and 2 (VDE1, VDE2) in the voltage monitoring voltage divider control registers 1 and 2 (ADCJ0VMONVDCR1, ADCJ0VMONVDCR2) to 1 allows A/D conversion of each voltage. Divided voltages of VCC and E0VCC are A/D-converted.

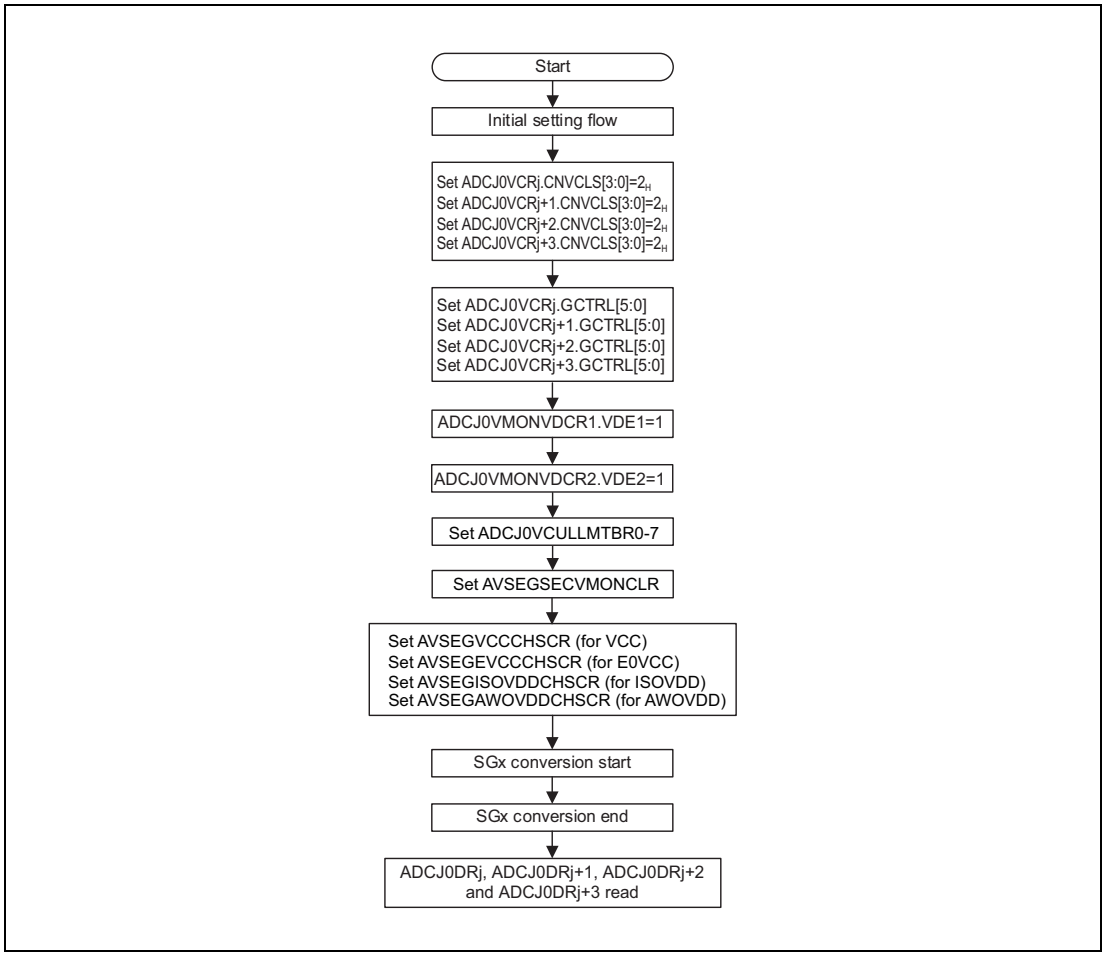
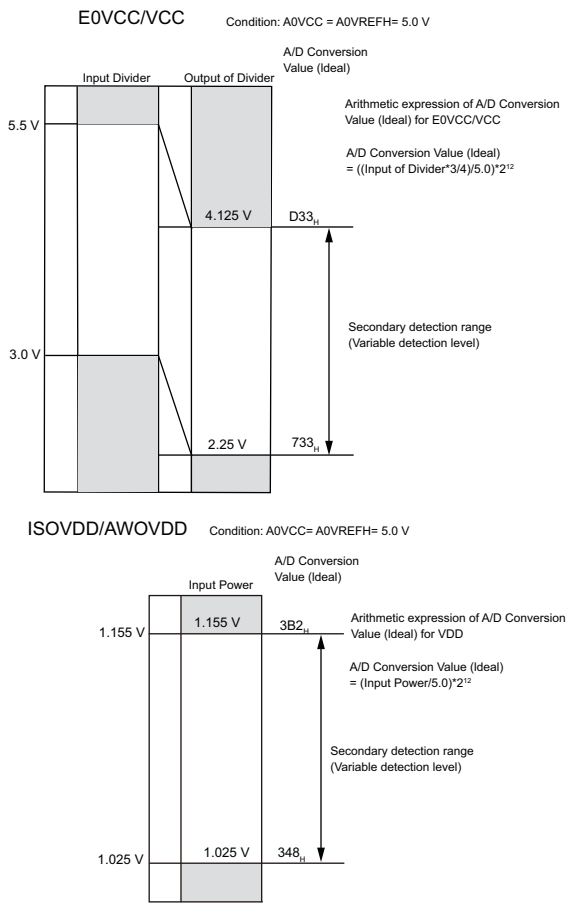


Figure 43.24 Recommended Flow

1. Make settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**).
2. Set CNVCLS[3:0] = 2_H and set each physical channel of power voltages (VCC, E0VCC, ISOVDD and AWOVDD) for GCTRL[5:0] in virtual channel setting registers ADCJ0VCR0, ADCJ0VCR1, ADCJ0VCR2, and ADCJ0VCR3. (* Settings can be made freely if four ADCJ0VCRs are consecutive.)
3. Set VDE1 in the voltage monitoring voltage divider control register 1 to 1, and then set VDE2 in the voltage monitoring voltage divider control register 2 to 1.
4. Set upper limit value and lower limit value of each power supply voltage to ADCJ0VCULLMTBR0 to ADCJ0VCULLMTBR7 (any of ADCJ0VCULLMTBR (ADCJ0VCULLMTBR0 to ADCJ0VCULLMTBR7) are able to be used as voltage monitor). For reference value of upper/lower limit value, refer to **Figure 43.25**.

5. Set AVSEGVCCCHSCR (for VCC), AVSEGEVCCCHSCR (for E0VCC), AVSEGISOVDDCHSCR (for ISOVDD) and AVSEGAWOVDDCHSCR (for AWOVDD) to allocate virtual channels which corresponds to each power source. For details of this setting, refer to **Section 43.7, ADC VMON Secondary Error Generator (AVSEG)**.
6. Set configuration of a filter to AVSEGVCCCNTCR (for VCC), AVSEGEVCCCNTCR (for E0VCC), AVSEGISOVDDCNTCR (for ISOVDD) and AVSEGAWOVDDCNTCR (for AWOVDD). For details of this setting, refer to **Section 43.7, ADC VMON Secondary Error Generator (AVSEG)**.
7. Activate SG, and then perform A/D conversion.

Using the upper/lower limit check function together makes it possible to output an interrupt request (INTADCJ0SEC) if an unexpected voltage is generated. Also, check value of AVSEGSECVMONERR register to confirm error status is kept.



Note 1. Set the upper/lower limit value as the value with which the absolute error of A/D conversion of power supply voltage is added to a reference value of this figure.

Note 2. For more information, refer to **Section 11, Power Supply Voltage Monitor**.

Figure 43.25 Reference Value of Upper/Lower Limit Value of Secondary Power Supply Voltage Monitor

Figure 43.26 shows operating example of voltage monitoring function.

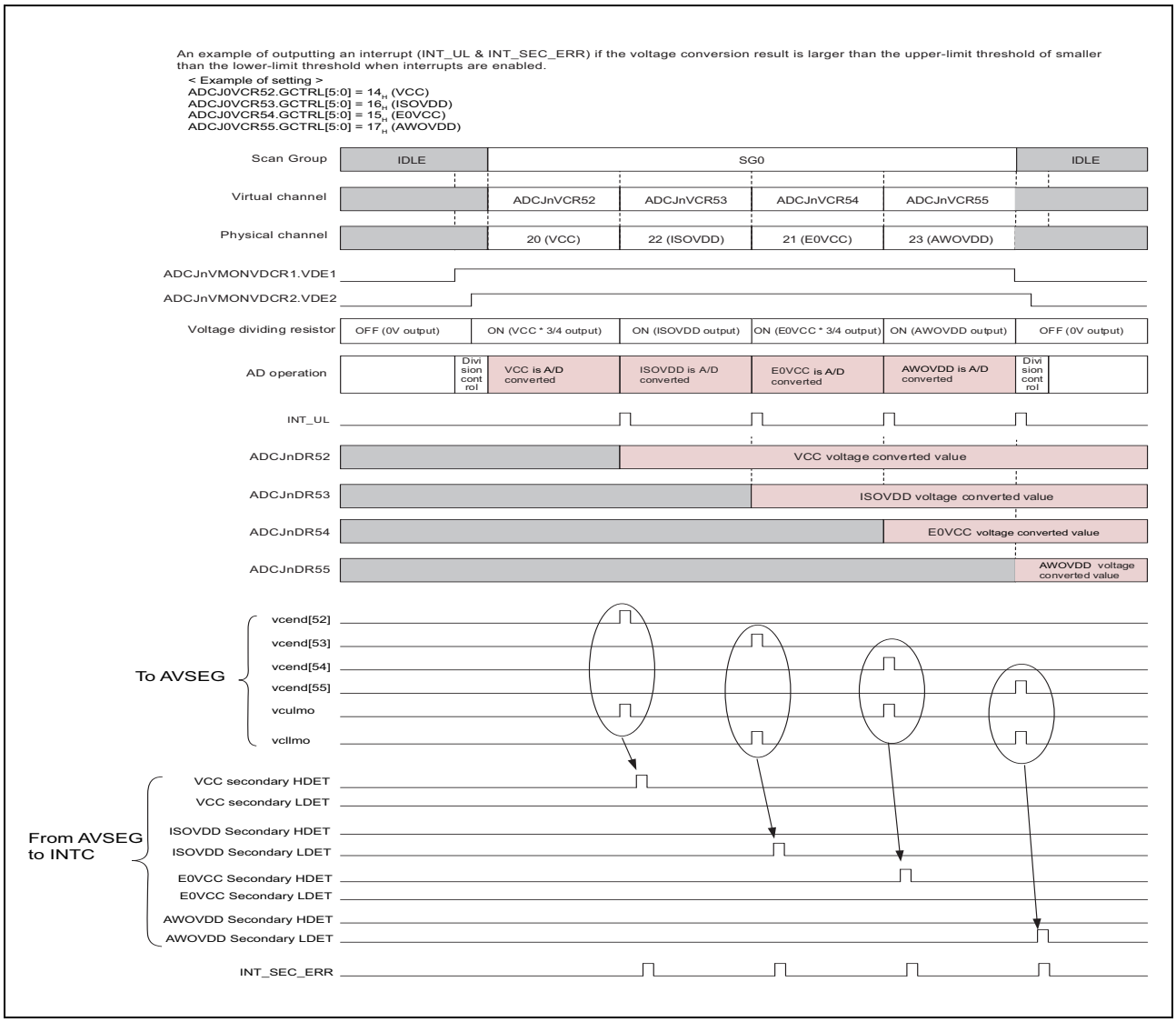


Figure 43.26 Example of Voltage Monitoring Operation

NOTE

For details about register setting and operation of INT_UL, vculmo, vcllmo, refer to **Section 43.4.19.5, Upper/Lower Limit Check**.

Refer to **Section 43.4.3.1, Forced Termination Flow of All Scan Groups**.

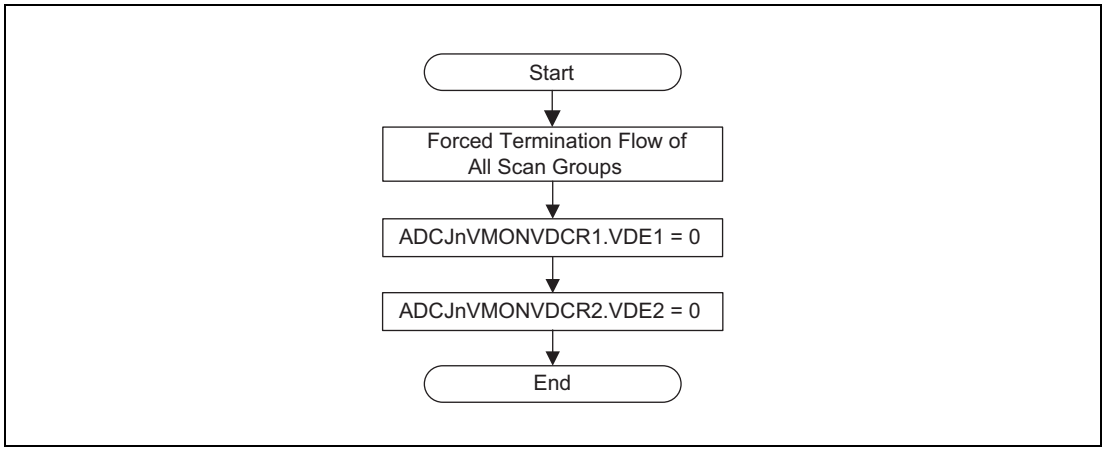


Figure 43.27 Stopping flow of Voltage Divider Enable bit 1/2

CAUTIONS

1. **Before stopping the ADCJ0 by module standby mode, set the voltage divider enable bits 1 and 2 (VDE1, VDE2) to 0.**
2. **Suppress the AnVREFH swing as much as possible.**

43.4.10 Self-Diagnostic Functions

The ADCJ is equipped with the following six self-diagnostic functions.

- Pin-level self-diagnostic function
- T&H path self-diagnostic function
- ADcore self-diagnostic function
- Wiring-break detection self-diagnostic function
- Voltage monitoring self-diagnostic function
- A/D conversion of A/D conversion data path diagnosis

43.4.10.1 Pin-Level Self-Diagnosis

The pin-level self-diagnosis function applies different voltage values to the physical channels of the even group and to the physical channel of the odd group respectively and performs A/D conversion to diagnose errors in the path from the I/O buffer of physical channels to the ADcore. If the A/D conversion result differs from the voltage applied to corresponding physical channel, target analog input path may be damaged.

There are 3 types of path from the I/O buffer of physical channels to the ADCore.

(1) When ADCJnSMPCR.BUFAMPD is set to 0, I/O buffer → RRAMP → ADcore input (shown in this section)

(2) When ADCJnSMPCR.BUFAMPD is set to 1, I/O buffer → ADcore input

(3) I/O buffer → T&H circuit → ADcore input (shown in **Section 43.4.10.2, Self-Diagnosis of the T&H Path**)

When diagnosing the path of (1), (2) or (3), the voltage to be applied to the I/O buffer is determined by the ADCJnTDCR.TDLV[1:0] setting.

	Voltage Applied to High Accuracy Input Pins		Voltage Applied to Low Accuracy Input Pins	
ADCJnTDCR.TDLV[1:0]	Even group	Odd group	Even group	Odd group
0_H	AnVSS	AnVCC	AnVSS	EnVCC
1_H	AnVCC	AnVSS	EnVCC	AnVSS
2_H	AnVSS	1/2×AnVCC	Not available	Not available
3_H	1/2×AnVCC	AnVSS	Not available	Not available

(1) The following describes self-diagnosis mode that diagnoses errors in the path from the I/O buffer through RRAMP to ADcore input.

[Recommended flow]

Set 0 to ADCJnSMPCR.BUFAMPD, and set the virtual channel register j (ADCJnVCRj) and TDLV[1:0] in the pin-level self-diagnosis register ADCJnTDCR according to the initial setting flow (**Section 43.4.1.1, Initial Setting Procedure**).

ADCJnVCRj number	CNVCLS[3:0]	GCTRL[5:0]
N	7_H	ANnpq number

Assign virtual channel n shown above to scan group x.

Start scan group x according to the startup flow and perform A/D conversion.

NOTE

When diagnosing errors in the path from (2) I/O buffer to ADcore input, the buffer amplifier setting has to change to disabled from the setting of [Recommended flow]. So, set SMPCR.BUFAMPD to 1, set other settings same as [Recommended flow], and start scan group x.

43.4.10.2 Self-Diagnosis of the T&H Path

This section describes self-diagnosis mode that diagnoses errors in the path from the I/O buffer through the T&H circuit to ADcore input.

Confirm that both the following (a) and (b) satisfy the diagnosis voltage values of physical channels connected to T&H.

	Route to be diagnosed	Diagnosis method
(a)	IO buffer ⇒ T&H circuit ⇒ ADcore input	Pin-Level Self-Diagnosis of the T&H Path
(b)	IO buffer ⇒ RRAMP ⇒ ADcore input	Pin-Level Self-Diagnosis of physical channel

Perform this diagnosis for all cases of the pin-level self-diagnosis level bits (ADCJnTDCR.TDLV) = 0_H, 1_H, 2_H, and 3_H. When respective diagnosis voltage values are satisfied, it can be decided that the T&H path is normal.

The setting method and judgment method of both (a) and (b) are explained in **(1), Pin-Level Self-Diagnosis of the T&H Path**.

Note: Even/odd group of self-diagnosis of the T&H path means even/odd physical channel number.
Even group = ANn00, ANn02
Odd group = ANn01, ANn03

(1) Pin-Level Self-Diagnosis of the T&H Path

[Recommended flow]

1. Initialize the settings according to "**Section Figure 43.29, Initialization Settings for Self-Diagnosis of the T&H Path**"
2. Write 1 to the sampling start bit (ADCJnTHSMPSTCR.SMPST) to start T&H sampling processing.
3. After waiting the period of T&H sampling*1, write 1 to the hold start bit (ADCJnTHAHLDSTCR.HLDST) to start T&H hold processing.
4. Write 1 to the scan group 1 start bit (ADCJnSGSTCR1.SGST), and scan group 1 is held suspended during the period of T&H hold processing.
5. After T&H hold processing, scan group 1 is started.
6. After generated the first scan End interrupt of virtual channel 0, change the self-diagnosis voltage level by writing pin-level self-diagnosis level bits (ADCJnTDCR.TDLV).
7. After scan group 1 is end, check the A/D conversion result. The expected value of hold value A/D conversion that is from ADCJnVCR2 to ADCJnVCR5 is the self-diagnosis voltage level before changing the self-diagnosis voltage level, and the expected value of pin-level self-diagnosis A/D conversion that is from ADCJnVCR6 to ADCJnVCR9 is the self-diagnosis voltage level after changing the self-diagnosis voltage level.
8. Change the pin-level self-diagnosis level bits (ADCJnTDCR.TDLV) from 0_H to 3_H, and repeat step from 1) to 7). If each value of data register n (ADCJnDR2 to ADCJnDR9) satisfies with the self-diagnosis voltage level of both odd group and even group, T&H path is judged to be normal.

Note 1. Wait for at least 0.45 µs (minimum sampling time of T&H) from the T&H sampling start in step 2), and then carry out the hold start control in step 4).

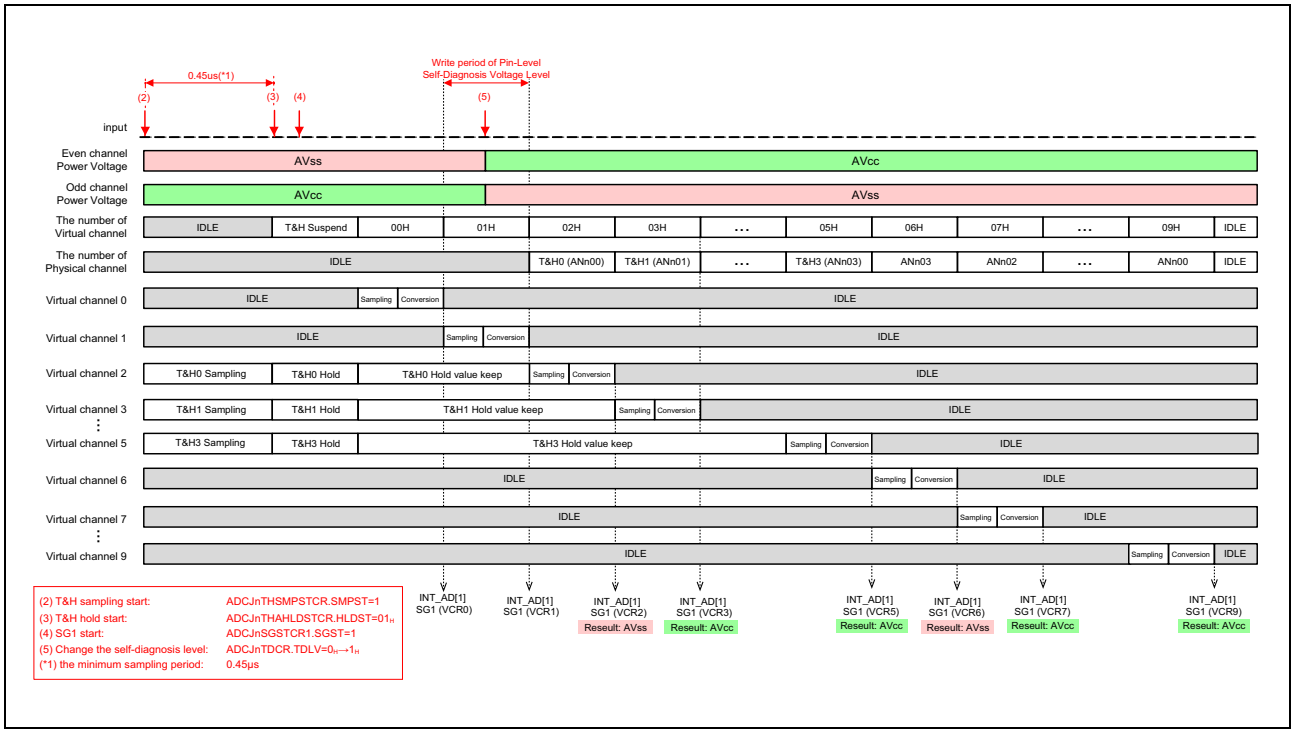


Figure 43.28 Example of Self-Diagnosis of the T&H Path Timing Chart

CAUTION

About the value of A/D conversion of self-diagnosis voltage level, refer to **Section 55, Electrical Characteristics**.

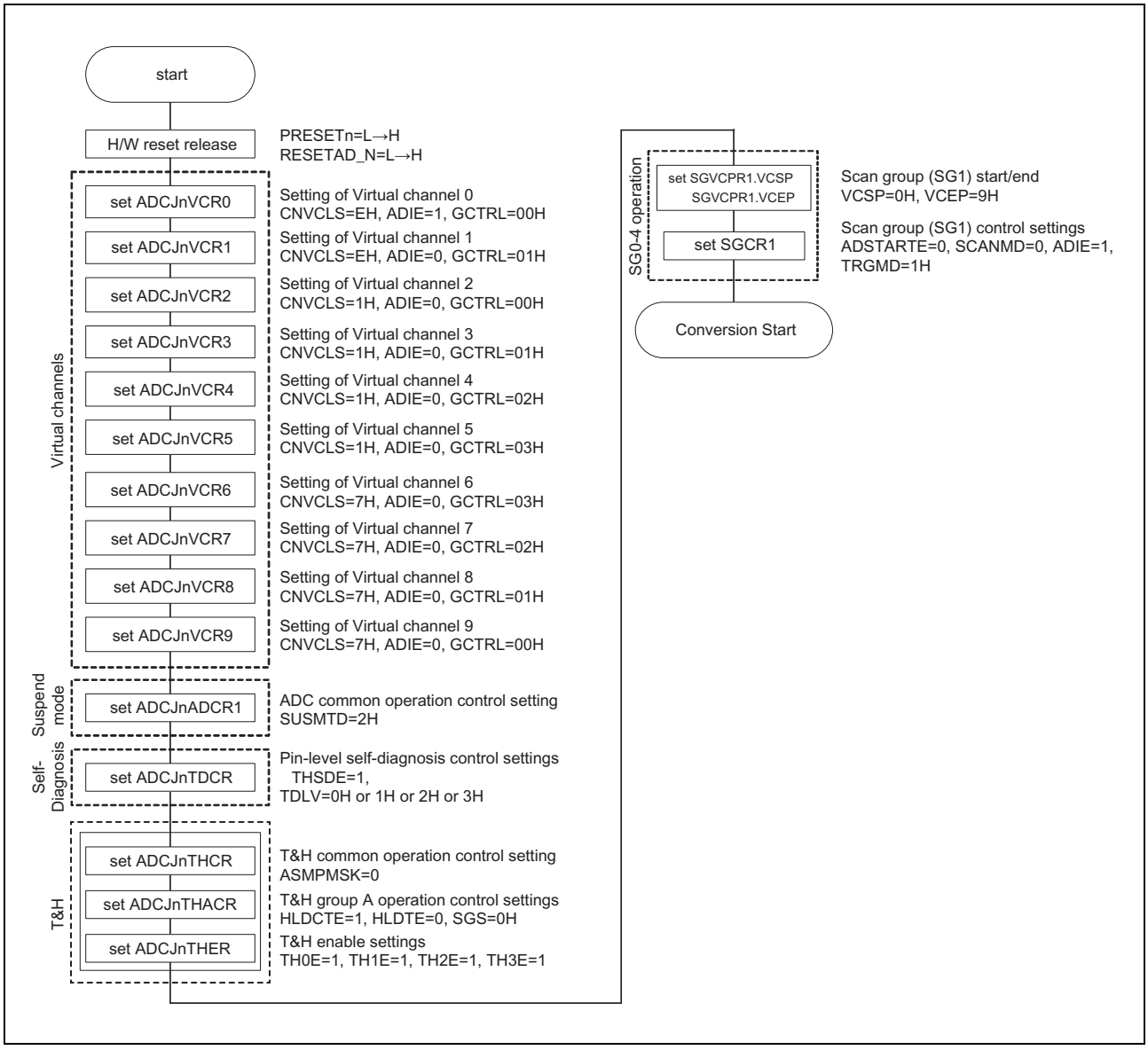


Figure 43.29 Initialization Settings for Self-Diagnosis of the T&H Path

43.4.10.3 A/Dcore Self-Diagnosis Function

The A/D core self-diagnosis function is used to verify that A/D conversion operates normally.

If the A/D conversion result differs from the expected value, the ADcore macro may be damaged.

To set a voltage value, set the CNVCLS[3:0] to 3_H and set GCTRL[5:0] in the virtual channel register j.

Conversions by AnVREFH × 1, AnVREFH × 3/4, AnVREFH × 1/2, AnVREFH × 1/4, and AnVREFH × 0 are available.

[Recommended flow]

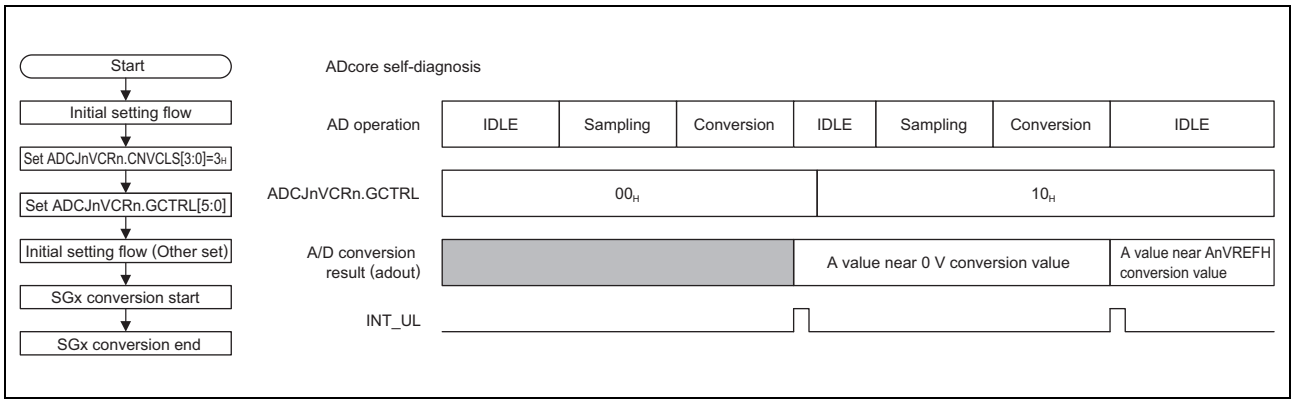


Figure 43.30 A/D core Self-Diagnosis Flow

1. Make settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**).
2. In the virtual channel setting register ADCJnVCRj (j = 0 to 63), set CNVCLS[3:0] to 3_H and set any A/D core self-diagnosis voltage level in GCTRL[5:0].
3. Make other necessary settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**).
4. Assert triggers of SG0 to SG4 and perform A/D conversion.

43.4.10.4 Self-Diagnosis of the Wiring-Break Detection Function

(1) Self-Diagnosis of Wiring-Break Detection Mode 1

The self-diagnosis A/D conversion in wiring-break detection mode 1 is used to check that the wiring-break detection mode 1 works normally. The A/D conversion executes in disabling ANI input and pull-down the ANI by ADCJnVCRj and ADCJnODCR setting. The self-diagnosis A/D conversion in wiring-break detection mode 1 is executed continuously several times for same physical channel. When the A/D conversion result of pull-down attenuates to approximately 0 V, the wiring-break detection mode 1 is judged to be working normally.

[Recommended flow]

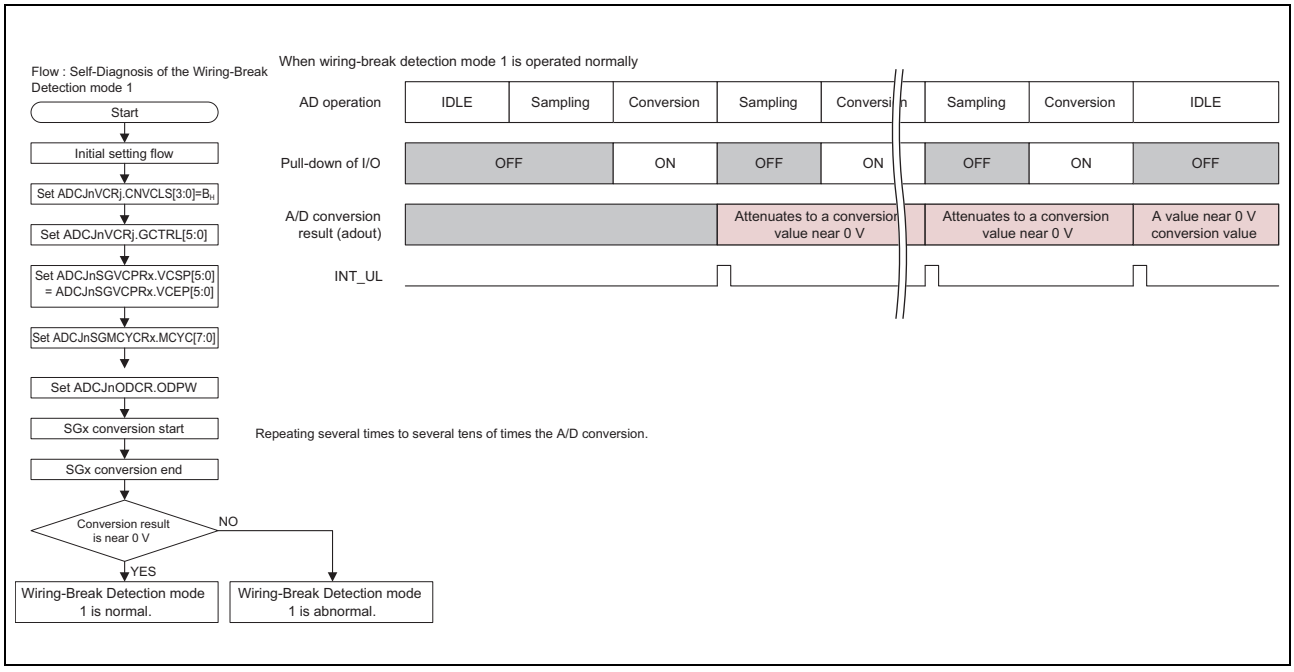


Figure 43.31 Recommended Flow of Self-Diagnosis of Wiring-Break Detection Mode 1

1. Make settings according to the initial setting procedure (**Section 43.4.1.1, Initial Setting Procedure**).
2. In the virtual channel setting register (ADCJnVCRj (j = 0 to 63)), set CNVCLS[3:0] to B_H and set any physical channels in GCTRL[5:0].
3. Set only one virtual channel by Scan Group x Virtual Channel Pointer Register (ADCJnSGVCPRx). Set the number of virtual channel j to both VCSP[5:0] and VCEP[5:0].
4. Set the number of self-diagnosis A/D conversion in wiring-break detection mode 1 to Scan Group x Multicycle register (ADCJnSGMCYCRx.MCYC).
5. In the wiring-break detection control register (ADCJnODCR), set any wiring-break detection pulse width in ODPW[4:0].
6. Activate the scan group x to perform the self-diagnosis A/D conversion in wiring-break detection mode 1.

7. Perform the self-diagnosis A/D conversion in wiring-break detection mode 1 several times.
   If the A/D conversion results attenuate to near 0 V, it is determined that wiring-break detection mode 1 (pull-down function) is operating normally.

(2) Self-Diagnosis of Wiring-Break Detection Mode 2

The wiring-break detection mode 2 self-diagnosis function is used to check that wiring-break detection mode 2 works normally.

The first A/D conversion execute in disabling ANI input and pull-down the ANI, and then the A/D conversion result of pull-down indicate approximately 0 V. The second A/D conversion execute in disabling ANI input and pull-up the ANI, and then the A/D conversion result of pull-up indicate approximately 5 V. The wiring-break detection mode 2 is judged to be working normally from each A/D conversion result.

[Feature]

Users can freely set physical channels to which wiring-break detection mode 2 self-diagnosis is applied.

[Recommended flow]

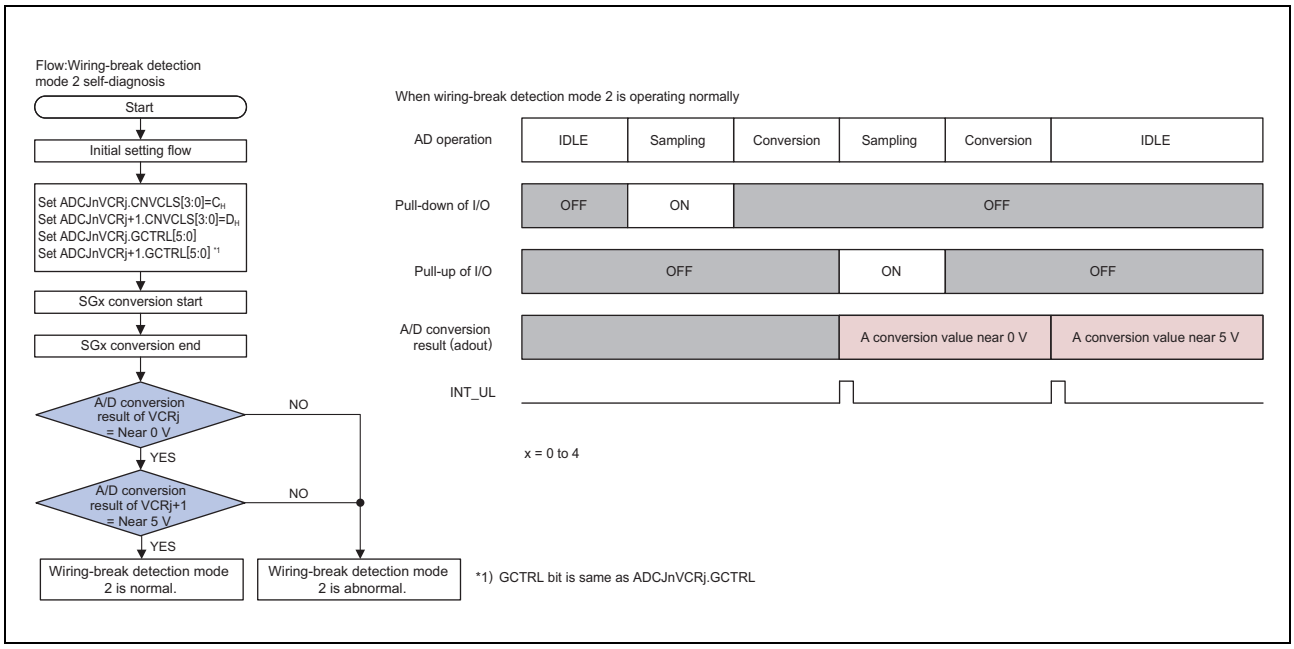


Figure 43.32 Recommended Flow of Self-Diagnosis of Wiring-Break Detection Mode 2

Set the I/O buffer of the ANnpq pin to be detected to the pull-up or pull-down state.

Make the following settings for consecutive two virtual channels (ADCJnVCRj) according to "**Section 43.4.1.1, Initial Setting Procedure**".

I/O buffer	ADCJnVCRj number	CNVCLS[3:0]	GCTRL[5:0]
Pull-down	j	C_H	ANnpq number
Pull-up	j+1	D_H	

Assign the set ADCJnVCRj and ADCJnVCRj+1 to the same scan group x.

Start scan group x according to the startup flow and perform A/D conversion.

[Decision method]

When wiring-break detection mode operates normally, the A/D conversion result of ADCJnVCRj attenuates to near 0 V and the A/D conversion result of ADCJnVCRj+1 rises to near AnVCC. Therefore, it can be diagnosed from each A/D conversion result that wiring-break detection mode 2 is operating normally.

43.4.10.5 Self-Diagnosis of the Voltage Monitoring Function

The voltage monitoring self-diagnosis function performs A/D conversion with the voltage dividing resistor in the power voltage monitoring circuit set to OFF by the voltage monitoring voltage divider control registers 1 and 2 (ADCJnVMONVDCR1, ADCJnVMONVDCR2). When the conversion result becomes near 0 V, it verifies that the voltage dividing resistor for voltage monitoring is normally controlled.

When performing self-diagnosis, perform A/D conversion and self-diagnosis of the power voltage alternately for each channel. For example of operation, see "**Figure 43.33, Example of Operation of Self-Diagnosis of Voltage Monitor**".

[Recommended flow]

1. Make settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**). However, when using the self-diagnosis function of the voltage monitor, operate only one target scan group. It is prohibited to perform A/D conversion of other scan group during using the self-diagnosis function of voltage monitor.
2. In the virtual channel setting register (ADCJnVCRj (j = 0 to 63)), set CNVCLS[3:0] to 0_H and set the channel for power voltage in GCTRL[5:0] for one channel.
3. In the voltage monitoring voltage divider control register 1, set VDE1 to 1, and then set VDE2 in the voltage monitoring voltage divider control register 2 to 1.
4. Perform A/D conversion, and then confirm that the conversion result is the power voltage value.
5. In the voltage monitoring voltage divider control register 1, set VDE1 to 0, and then set VDE2 in the voltage monitoring voltage divider control register 2 to 0.
6. Perform A/D conversion (self-diagnosis), and then confirm that the conversion result is near 0 V.

Figure 43.33 shows example of operation of Self-Diagnosis of Voltage Monitor. Rewrite ADCJnVMONVDCR1.VDE1 and ADCJnVMONVDCR2.VDE2 to 0 according to **Figure 43.27, Stopping flow of Voltage Divider Enable bit 1/2**.

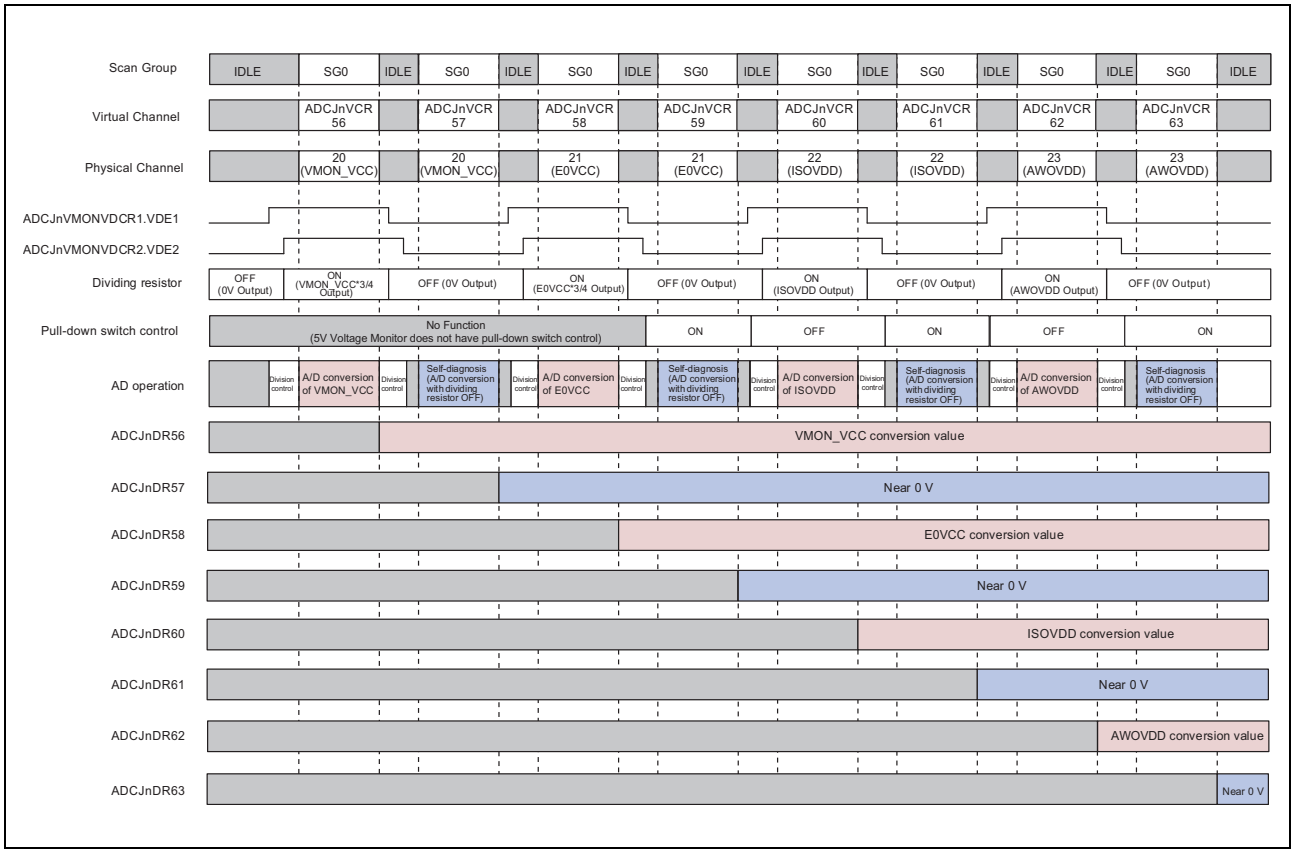


Figure 43.33 Example of Operation of Self-Diagnosis of Voltage Monitor

43.4.10.6 A/D Conversion Data Path Diagnosis

The A/D conversion data path diagnosis is used to diagnose the data path until data is stored in the data register from the ADCore.

Select a fixed value of A/D conversion data from the ADCore from GCTRL[5:0] in the virtual channel register ADCJnVCRj, and then perform A/D conversion.

If the A/D conversion result is equal to the fixed value of A/D conversion data selected from GCTRL[5:0], the data path is decided to be normal.

CAUTION

Because the A/D conversion data fixed value is 12 bits, select the 12-bit data format. A/D conversion data is stored in the data register with the data format selected by the DFMT bit.

[Recommended flow]

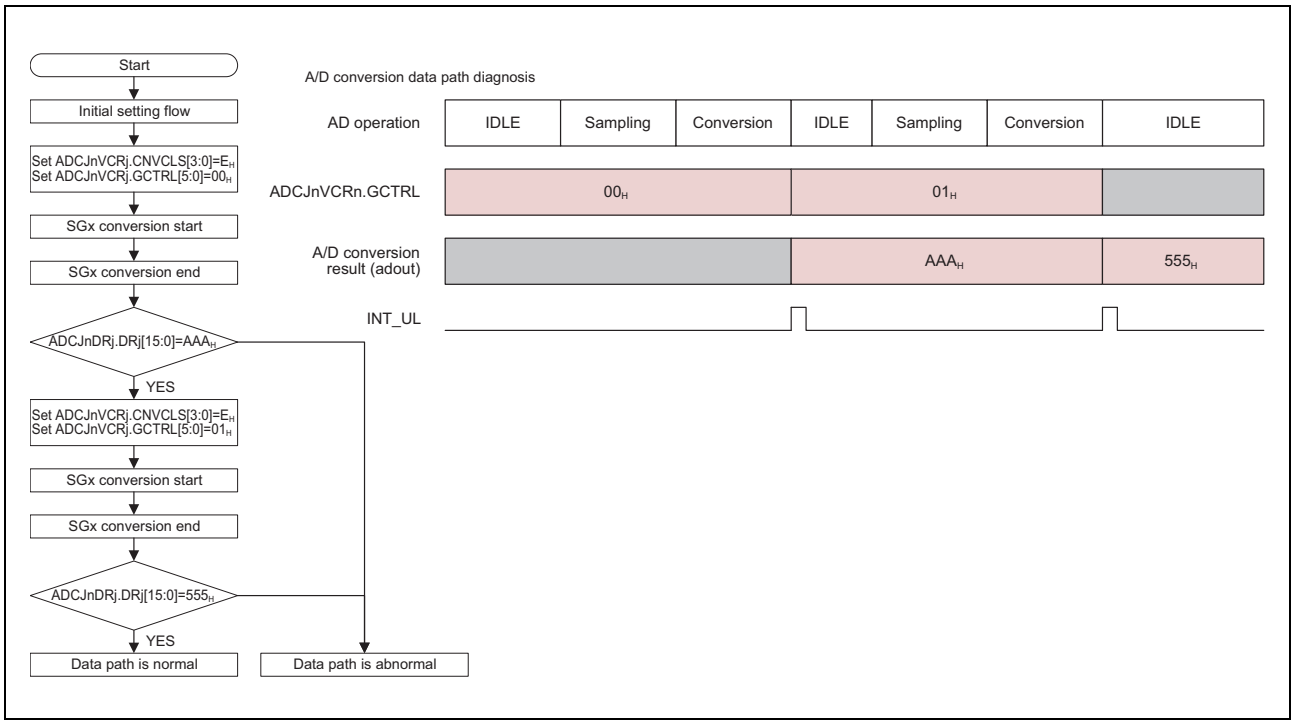


Figure 43.34 Recommended Flow of A/D Conversion Data Path Diagnosis

[Recommended flow]

1. Make settings according to the initial settings (**Section 43.4.1.1, Initial Setting Procedure**).
2. In the virtual channel setting register ADCJnVCRj (j = 0 to 63), set CNVCLS[3:0] = E_H and GCTRL[5:0] = 00_H.
3. Assert triggers of SG0 to SG4 and perform A/D conversion.
4. After the A/D conversion, compare the A/D conversion result with the fixed value (AAA_H) selected by GCTRL[5:0].
5. In the virtual channel setting register ADCJnVCRj (n = 0 to 63), set CNVCLS[3:0] = E_H and GCTRL[5:0] = 01_H.
6. Assert triggers of SG0 to SG4 and perform A/D conversion.
7. After the A/D conversion, compare the A/D conversion result with the fixed value (555_H) selected by GCTRL[5:0].
8. When the comparison results in steps 4) and 7) are equal to the fixed values respectively, the data path is decided to be normal.

43.4.11 Wait Function

The ADCJ can insert a wait before the A/D conversion sampling time. Decide whether to insert a wait by the CNVCLS[3:0] setting in ADCJnVCRj. The table below shows A/D conversion types in which wait (1) and wait (2) can be inserted.

(1) A/D conversion types in which a wait can be inserted into the first virtual channel at the beginning of scan group

(2) A/D conversion types in which a wait can be inserted into channels other than the first virtual channel

Table 43.85 A/D Conversion Types in Which a Wait Can Be Inserted

ADCJnVCRj.CNVCLS	Conversion Type	(1) Insertion of wait into the first channel	(2) Insertion of wait into other channels
0_H	Normal A/D conversion	×	√
1_H	Hold value A/D conversion	×	×
2_H	Normal A/D conversion at extended sampling cycle	×	√
3_H	ADcore self-diagnosis A/D conversion	×	×
4_H	Addition mode A/D conversion	×	√
5_H	MPX normal A/D conversion	√	√
6_H	MPX addition mode A/D conversion	√	√
7_H	Pin-level self-diagnosis A/D conversion	×	√
8_H	A/D conversion in wiring-break detection mode 1	×	√
9_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)	×	√
A_H	A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)	×	√
B_H	Self-diagnosis A/D conversion in wiring-break detection mode 1	×	√
C_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-down)	×	√
D_H	Self-diagnosis A/D conversion in wiring-break detection mode 2 (physical channel IO pull-up)	×	√
E_H	A/D conversion of A/D conversion data path diagnosis (ADVAL mode)	×	×
F_H	Reserved	—	—

√ : Wait inserted
×: Wait not inserted

CAUTION

When the CNVCLS[3:0] value in ADCJnVCRj is 1_H, 3_H, or E_H, no wait is inserted even with the wait insertion setting (ADCJnVCRj.WTTS[3:0] = 1_H to 8_H and the ADCJnWAITTRy.WAITTIME[13:0] value specified by ADCJnVCRj.WTTS[3:0] is 0008_H or more).

The table below shows settable wait time and the number of wait time tables.

Category	Explanation
Wait time	ADCJnWAITTRy.WAITTIME[13:0] × CLKAD period
The number of wait time tables	8
Wait setting target	Each virtual channel

With the following settings, the ADCJ inserts a wait time.

Category	Explanation
Wait enable	ADCJnVCRj.WTTS[3:0] = 1_H to 8_H
Wait table setting	ADCJnVCRj.WTTS[3:0] = 1_H to 8_H
Wait time setting	ADCJnWAITTRy.WAITTIME[13:0]

NOTES

1. In addition mode, the wait is input before first A/D conversion is processed (see **Figure 43.35, Wait function when addition mode**).

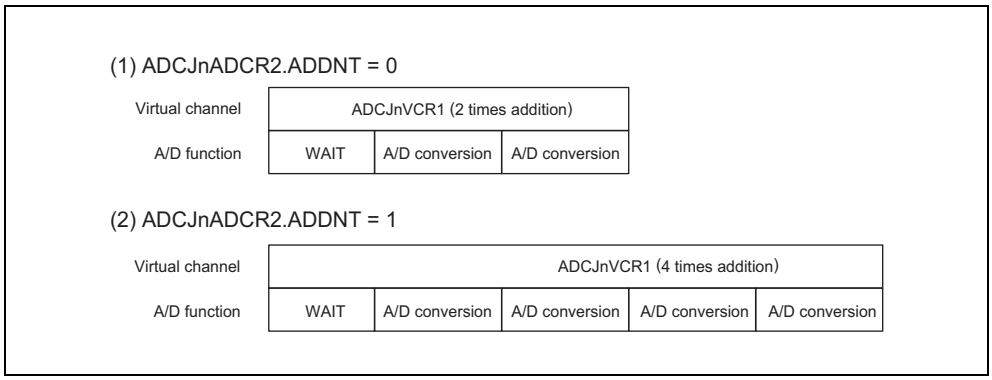


Figure 43.35 Wait function when addition mode

2. After the last virtual channel processing has been completed, no wait is inserted.
3. Even in MPX normal A/D conversion or MPX addition mode A/D conversion, no wait is inserted unless wait is enabled in the setting.

CAUTION

Wait setting is prohibited for all applicable virtual channels in scan groups including virtual channels that perform hold value A/D conversion.

43.4.11.1 Example of Wait Function between Virtual Channels

(1) Example of Wait Function (Multicycle Scan Mode that Scan Count is Once)

Register Name	Setting	Notes
ADCJnSGCR1.SCANMD	0_H	Multicycle scan mode
ADCJnSGMCYCR1.MCYC[7:0]	0_H	The scan count is once
ADCJnSGVCPR1.VCSP[6:0]	0_H	Starting pointer: ADCJnVCR0
ADCJnSGVCPR1.VCEP[6:0]	3_H	End pointer: ADCJnVCR3
ADCJnVCR0.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR0.WTTS[3:0]	1_H	A wait inserted
ADCJnVCR1.CNVCLS[3:0]	4_H	Addition mode A/D conversion
ADCJnVCR1.WTTS[3:0]	1_H	A wait inserted and wait table 0 selected
ADCJnVCR2.CNVCLS[3:0]	5_H	MPX normal A/D conversion
ADCJnVCR2.WTTS[3:0]	2_H	A wait inserted and wait table 1 selected
ADCJnVCR3.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR3.WTTS[3:0]	0_H	No wait inserted

Because the first virtual channel is not MPX normal A/D conversion or MPX addition mode A/D conversion, no wait is inserted in the first virtual channel 0.

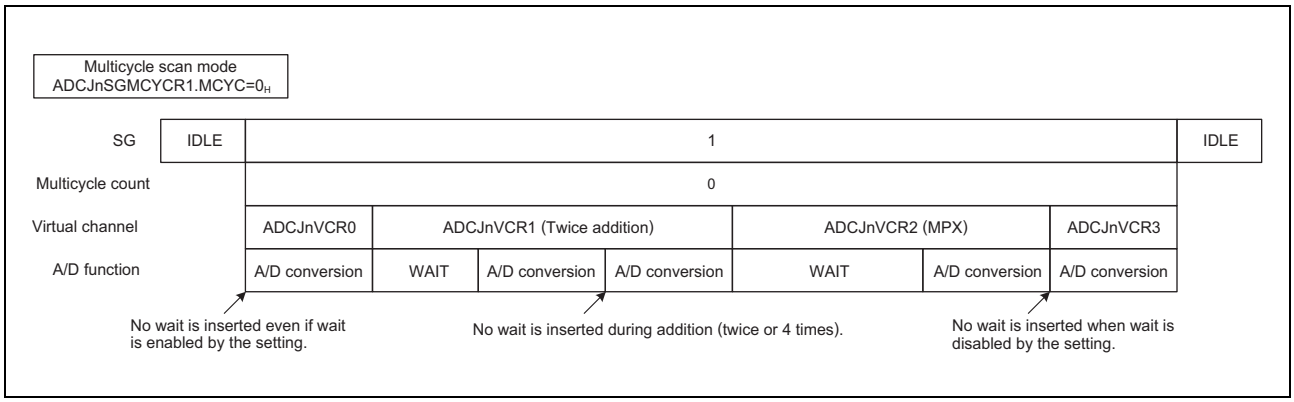


Figure 43.36 Wait Function (Multicycle Scan Mode that Scan Count is Once)

(2) Example of Wait Function (Multicycle Scan Mode that Scan Count is Twice)

Register Name	Setting	Notes
ADCJnSGCR1.SCANMD	0_H	Multicycle scan mode
ADCJnSGMCYCR1.MCYC[7:0]	1_H	The scan count is twice
ADCJnSGVCPR1.VCSP[6:0]	0_H	Starting pointer: ADCJnVCR0
ADCJnSGVCPR1.VCEP[6:0]	3_H	End pointer: ADCJnVCR3
ADCJnVCR0.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR0.WTTS[3:0]	1_H	A wait inserted
ADCJnVCR1.CNVCLS[3:0]	4_H	Addition mode A/D conversion
ADCJnVCR1.WTTS[3:0]	1_H	A wait inserted and wait table 0 selected
ADCJnVCR2.CNVCLS[3:0]	5_H	MPX normal A/D conversion
ADCJnVCR2.WTTS[3:0]	2_H	A wait inserted and wait table 1 selected
ADCJnVCR3.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR3.WTTS[3:0]	0_H	No wait inserted

Because the first virtual channel is not MPX normal A/D conversion or MPX addition mode A/D conversion, no wait is inserted in the first virtual channel 0. However, a wait is inserted in the virtual channel 0 of the second or later scan.

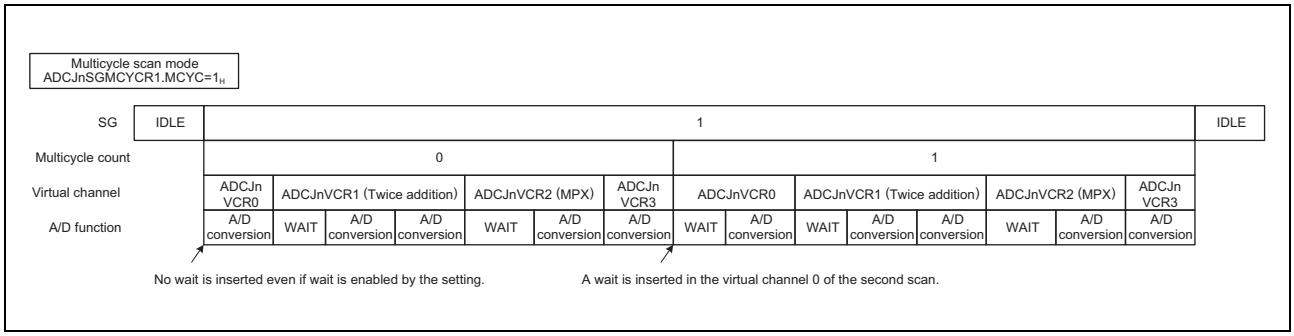


Figure 43.37 Wait Function (Multicycle Scan Mode that Scan Count is Twice)

(3) Example of Wait Function (Continuous Scan Mode)

Register Name	Setting	Notes
ADCJnSGCR1.SCANMD	1_H	Continuous scan mode
ADCJnSGVCPR1.VCSP[6:0]	0_H	Starting pointer: ADCJnVCR0
ADCJnSGVCPR1.VCEP[6:0]	3_H	End pointer: ADCJnVCR3
ADCJnVCR0.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR0.WTTS[3:0]	1_H	A wait inserted
ADCJnVCR1.CNVCLS[3:0]	4_H	Addition mode A/D conversion
ADCJnVCR1.WTTS[3:0]	1_H	A wait inserted and wait table 0 selected
ADCJnVCR2.CNVCLS[3:0]	5_H	MPX normal A/D conversion
ADCJnVCR2.WTTS[3:0]	2_H	A wait inserted and wait table 1 selected
ADCJnVCR3.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR3.WTTS[3:0]	0_H	No wait inserted

Because the first virtual channel is not MPX normal A/D conversion or MPX addition mode A/D conversion, no wait is inserted in the first virtual channel 0. However, a wait is inserted in virtual channel 0 of the starting pointer in the second continuous scan mode or later.

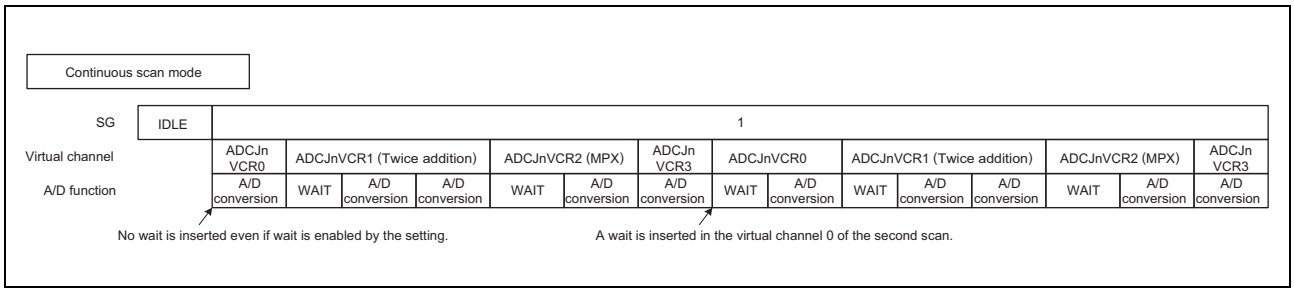


Figure 43.38 Wait Function (Continuous Scan Mode)

(4) Example of Wait Function (Synchronous Suspension)

This example shows operation of interrupt of a high-priority scan group (SG2 in the figure below) into a low-priority scan group (SG1 in the figure below) under the following conditions.

<Case1> An interrupt during A/D conversion of a low-priority scan group

<Case2> An interrupt during the wait time of a virtual channel of a low-priority scan group

<Case3> An interrupt during A/D conversion of a virtual channel with wait of a low-priority scan group

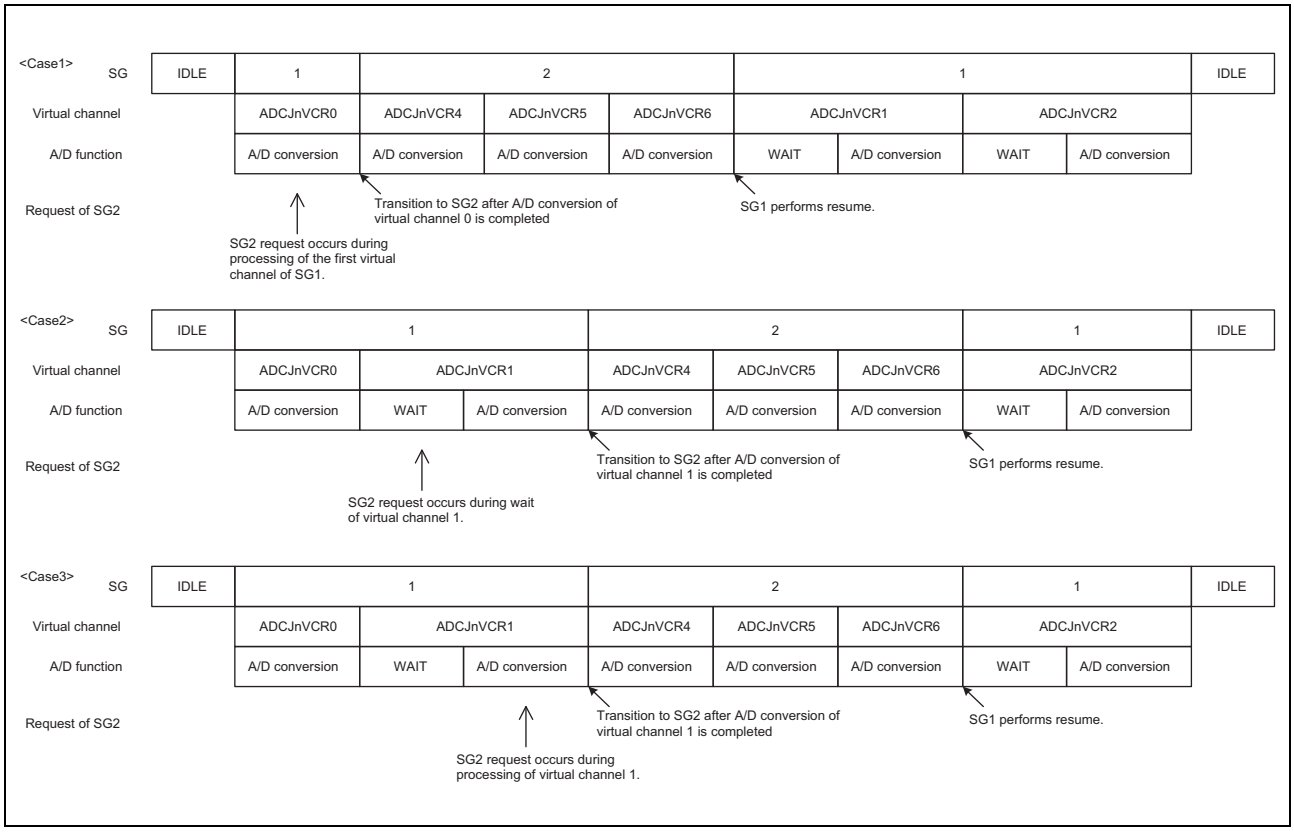


Figure 43.39 Wait Function (Synchronous Suspension)

(5) Example of Wait Function (Asynchronous Suspension)

This example shows operation of interrupt of a high-priority scan group (SG2 in the figure below) into a low-priority scan group (SG1 in the figure below) under the following conditions.

<Case1> An interrupt during A/D conversion of a low-priority scan group

<Case2> An interrupt during the wait time of a virtual channel of a low-priority scan group

<Case3> An interrupt during A/D conversion of a virtual channel with wait of a low-priority scan group

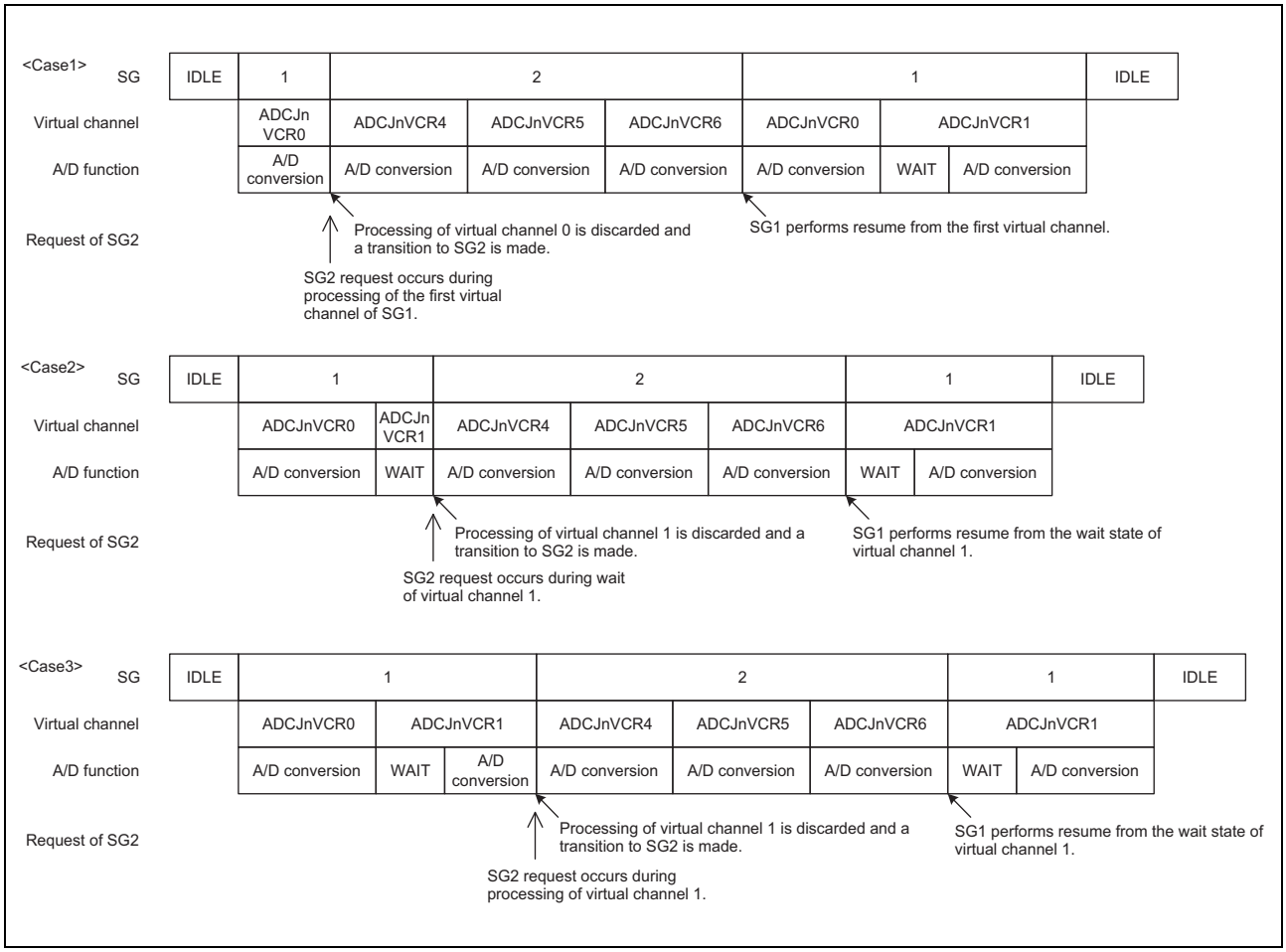


Figure 43.40 Wait Function (Asynchronous Suspension)

43.4.11.2 Example of MPX Wait

Register Name	Setting	Notes
ADCJnSGCR1.SCANMD	0_H	Multicycle scan mode
ADCJnSGMCYCR1.MCYC[7:0]	1_H	The scan count is twice
ADCJnSGVCPR1.VCSP[6:0]	0_H	Starting pointer: ADCJnVCR0
ADCJnVCR0.CNVCLS[3:0]	5_H	MPX normal A/D conversion
ADCJnVCR0.WTTS[3:0]	2_H	A wait inserted and wait table 1 selected
ADCJnVCR1.CNVCLS[3:0]	4_H	Addition mode A/D conversion
ADCJnVCR1.WTTS[3:0]	1_H	A wait inserted and wait table 0 selected
ADCJnVCR2.CNVCLS[3:0]	0_H	Normal A/D conversion
ADCJnVCR2.WTTS[3:0]	0_H	No wait inserted

When the A/D conversion type of a virtual channel of the starting pointer is MPX normal A/D conversion or MPX addition mode A/D conversion and wait is enabled by the setting, a wait is inserted in the first virtual channel 0. A wait is also inserted in virtual channel 0 in the second or later scan.

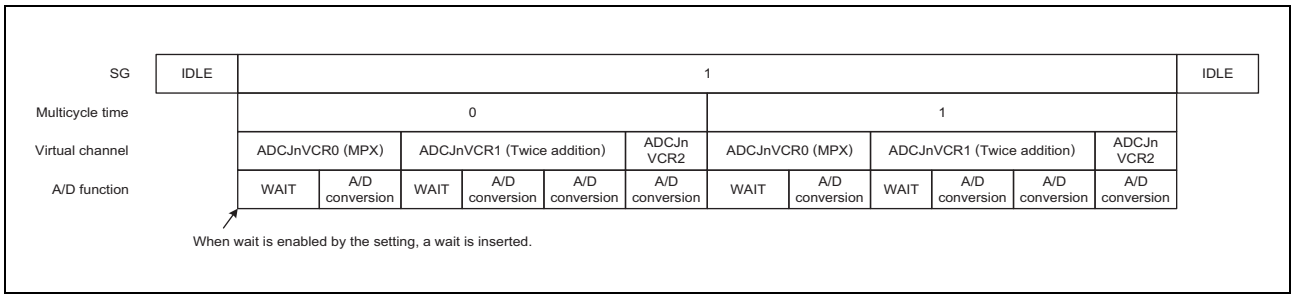


Figure 43.41 Wait Function (MPX A/D Conversion)

43.4.11.3 Sampling Period Setting

The following combinations for setting sampling period and buffer amplifier are supported.

Settings of other combinations are prohibited.

Category	Support Period Setting (Cycle)	Support Buffer AMP Setting (ADCJnSMPCR.BUFAMPD)	Notes
Extended sampling normal A/D conversion	60 or 99 or 138 or 252	0 (RRAMP enabled)	ADCJnVCRj.CNVCLS[3:0] = 2H
Common settings for all scans	99 or 204 or 252	1 (RRAMP disabled)	ADCJnSMPCR.SMPTS = 1
Default	18	0 (RRAMP enabled)	

The logic diagram for sampling period setting is shown below.

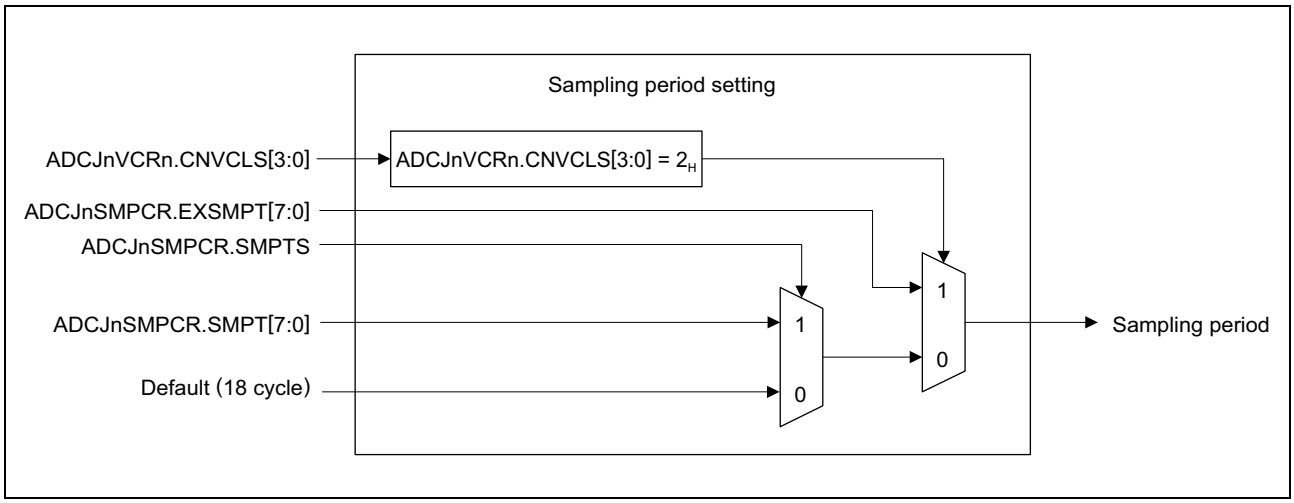


Figure 43.42 Logic Diagram of Sampling Period

43.4.12 Scan Mode

43.4.12.1 Multicycle Scan Mode

Multicycle scan mode is enabled by setting SCANMD in ADCJnSGCRx to 0.

Set the number of scan count (00_H to FF_H) in ADCJnSGMCYCRx. Set a value of (desired scan count - 1).

Figure 43.43 shows an example of multicycle scan mode operation when the scan count = 1 and 2.

- Conversion type: Normal A/D conversion mode (CNVCLS[3:0] = 0_H)
- Scan group 0
- Virtual channels: ADCJnVCR0 to ADCJnVCR3

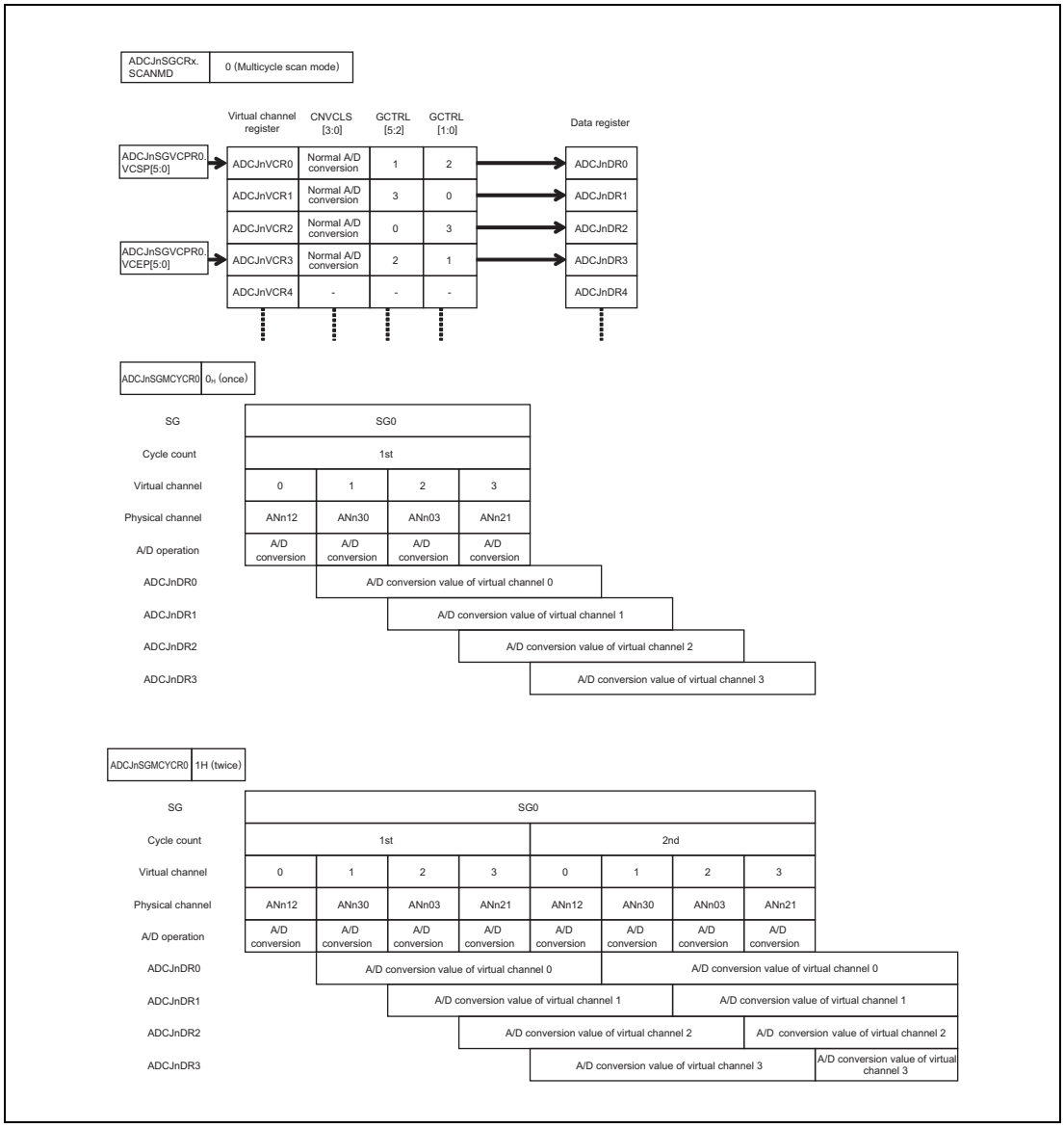


Figure 43.43 Example of Multicycle Scan Mode Operation

43.4.12.2 Continuous Scan Mode

In this mode, A/D conversion of virtual channels assigned to the target SGx is repeated unlimitedly.

Continuous scan mode is enabled by setting SCANMD in ADCJnSGCRx to 1. The set ADCJnSGMCYCRx value is invalid.

Figure 43.44 shows an example of continuous scan mode operation

- Conversion type: Normal A/D conversion mode (CNVCLS[3:0] = 0_H)
- Scan group 0
- Virtual channels: ADCJnVCR0 to ADCJnVCR3

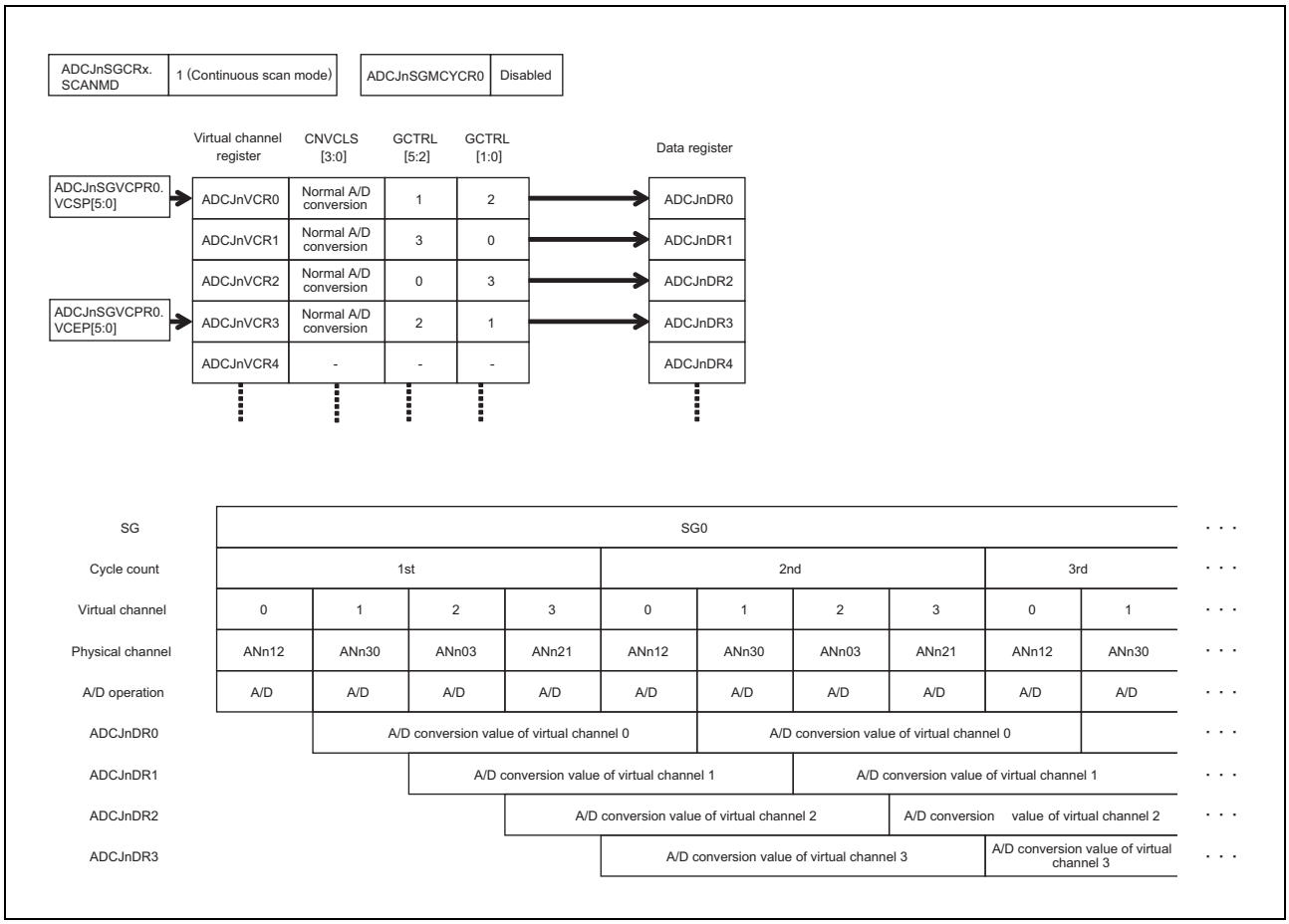


Figure 43.44 Example of Continuous Scan Mode Operation

43.4.13 Example of A/D Timer Operation

Scan group x can be activated at constant intervals by using the A/D timer trigger.

One count of the A/D timer = 1 CLKAD cycle

[Recommended flow]

Set the A/D timer initial phase register (ADCJnADTIPRx) and the A/D timer cycle register (ADCJnADTPRR) according to the initial setting flow (**Section 43.4.1.1, Initial Setting Procedure**).

The settable range is 00_0000_H to $1F\_FFFF_H$.

Set these registers considering the period required for one cycle of scan group x.

Activate A/D timer x with the A/D timer start trigger.

Stop A/D timer x with the A/D timer end trigger or by the A/D conversion forced termination setting (ADCJnADHALTR.HALT = 1).

The following describes an example of operation.

(1) When A/D timer x is activated, ADCJnADTIPRx is loaded to A/D timer x and the timer starts counting down.

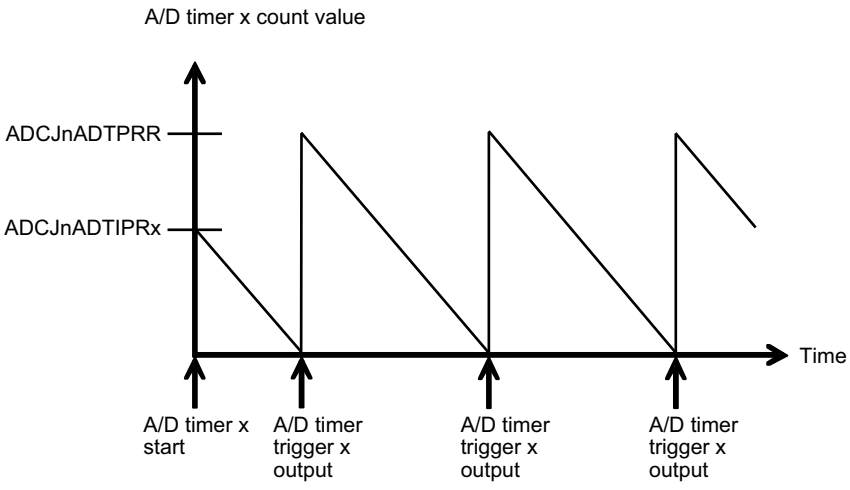
(2) When the A/D timer x becomes 0, the A/D timer trigger x is output for one CLKAD cycle and ADCJnADTPRRx is loaded to A/D timer x to restart down-counting.

After that, step (2) is repeated until the timer is halted.

If the A/D timer start trigger is received again after the A/D timer x is halted, operation starts from step (1).

The A/D timer synchronization start trigger enables A/D timers ADCJ0 and ADCJ1 to start simultaneously.

After synchronization start, A/D timers operate in the same way as above.



Note: x = 3, 4

Figure 43.45 Example of A/D Timer Operation

43.4.14 Trigger Function

The ADCJ has software triggers and hardware triggers.

While a scan group is operating (ADCJnSGSRx.SGACT = 1), if an A/D conversion start trigger of the same scan group is generated, the A/D conversion start trigger is ignored. Furthermore, while a scan group is operating (ADCJnSGSRx.SGACT = 1), a T&H hold start trigger of the same scan group is generated, the T&H hold start trigger is ignored.

At this time, an A/D error interrupt request can be output. For details, see "**Section 43.4.19.2, (1) Trigger Overlap Check Interrupt Request**"

While an A/D timer is operating (ADCJnSGSRx.ADTACT = 1), if an A/D timer start trigger of the same scan group is generated, the A/D timer start trigger is ignored.

A software trigger is generated by writing 1 to the bit in the register. Each software trigger has a single function.

The table below shows whether the trigger function of each scan group is provided or not.

Table 43.86 List of Availability of the Trigger Function in Each Scan Group

Category	Function	Register Bit Name or Pin Name	SG0	SG1	SG2	SG3	SG4	
							*1	*2
Software triggers	Scan group x start	ADCJnSGSTCRx.SGST	√	√	√	√	√	–
	Scan group x stop	ADCJnSGSTPCRx.SGSTP	√	√	√	√	√	–
	A/D conversion synchronization start	ADCJnADSYNSTCR.ADSTART	√	√	√	√	√	–
	AD forced termination	ADCJnADHALTR.HALT	√	√	√	√	√	√
	AD timer start	ADTSTCRx.ADTST	–	–	–	√	√	–
	AD timer synchronization start	ADCJnADTSYNSTCR.ADTSTART	–	–	–	√	√	–
	AD timer end	ADTENDCRx.ADTEND	–	–	–	√	√	–
	T&H sampling start	ADCJnTHSMPSTCR.SMPST	–	√	√	√	√	–
	T&H A hold	ADCJnTHAHLDSTCR.HLDST	–	√	√	√	√	–
	T&H stop	ADCJnTHSTPCR.THSTP	–	√	√	√	√	–
Hardware triggers	SGx trigger	SGx_TRG	√	√	√	√	√	–
	PWM-Diag trigger	PVCR_TRG	–	–	–	–	–	√

Note 1. ADCJnPWDCR.PWE=0

Note 2. ADCJnPWDCR.PWE=1

Note: √: Function provided
–: Function not provided

Note: x: 0 to 4

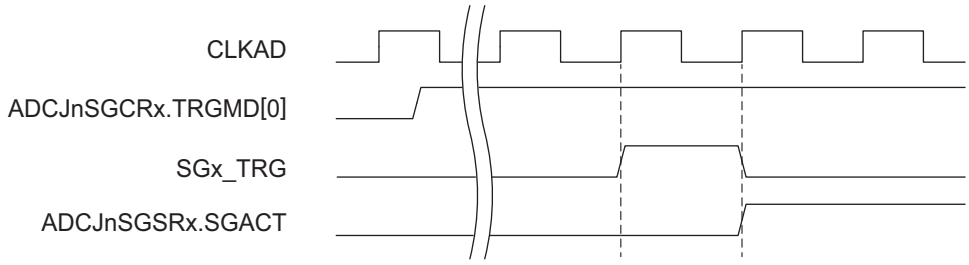
A hardware trigger is a rising edge of SGx_TRG.

For ADCJ0 & ADCJ1, there are external triggers (ADCJnTRGx), other Timers can become as trigger sources of SGx_TRG of each ADCJn, and they are chosen by the PIC (Peripheral Interconnect). For details, refer to **Section 41, Peripheral Interconnect (PIC)**.

For ADCJ2, there are external triggers (ADCJ2TRGx), TAUD2, TAUJ2, TAUJ3 and LPS can become as trigger sources of SGx_TRG of ADCJ2, and they are chosen by the ADCJnSGTSELx register. For details, refer to **Section 43.3.4.1, ADCJnSGTSELx — Scan Group x Start Trigger Control Register**.

43.4.14.1 Starting a Scan Group by Using a Hardware Trigger

When hardware triggers are enabled (ADCJnSGCRx.TRGMD[0] = 1) for trigger mode of scan group x, the scan group x status (ADCJnSGSRx.SGACT) is set to 1 after SGx_TRG is input.



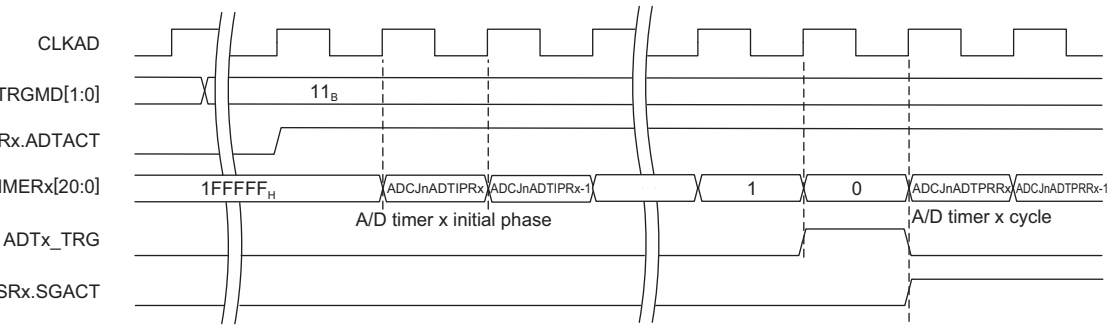
Note: x = 0 to 4

Figure 43.46 Hardware Trigger Timing

43.4.14.2 Starting a Scan Group by Using an A/D Timer Trigger

When scan group x and AD timer trigger x hardware triggers are enabled (ADCJnSGCRx.TRGMD[1:0] = 11_B) for trigger mode of scan group x, the scan group x status (ADCJnSGSRx.SGACT) is set to 1 after the AD timer trigger is input.

For AD timer x operation, see "**Section 43.4.13, Example of A/D Timer Operation**."



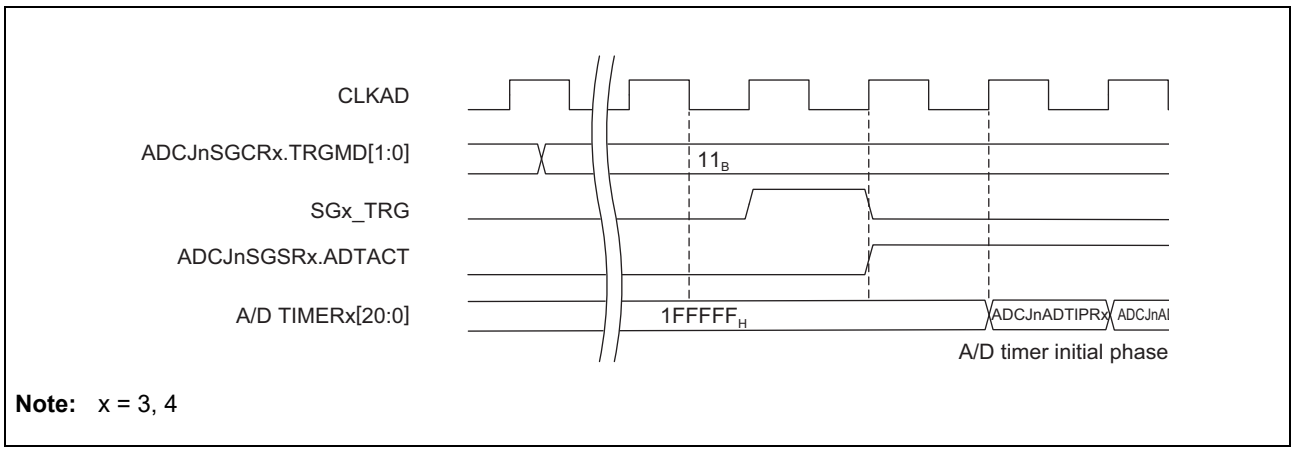
Note: x = 3, 4

Figure 43.47 A/D Timer x Hardware Trigger Timing

43.4.14.3 Starting A/D Timer by Using a Hardware Trigger

When scan group x and AD timer trigger x hardware triggers are enabled (ADCJnSGCRx.TRGMD[1:0] = 11_B) for trigger mode of scan group x, the AD timer status of scan group x (ADCJnSGSRx.ADTACT) is set to 1 (AD timer operating) after SGx_TRG is input.

For AD timer x operation, see "**Section 43.4.13, Example of A/D Timer Operation**."



Note: x = 3, 4

Figure 43.48 Hardware Trigger Timing (AD Timer x)

43.4.15 Example of Scan Group Stop

This section provides an example of concurrent stop operation of all scan groups (ADCJnADHALTR.HALT) and an example of individual stop operation of each scan group (ADCJnSGSTPCRx.SGSTP).

43.4.15.1 Example of Stopping All Scan Groups (A/D Halt)

When the HALT bit in ADCJnADHALTR is set to 1, all running scan groups are immediately stopped and transitioned to the idle state and the SGACT signal of each scan group are cleared to 0 (by at most 4 register access clocks).

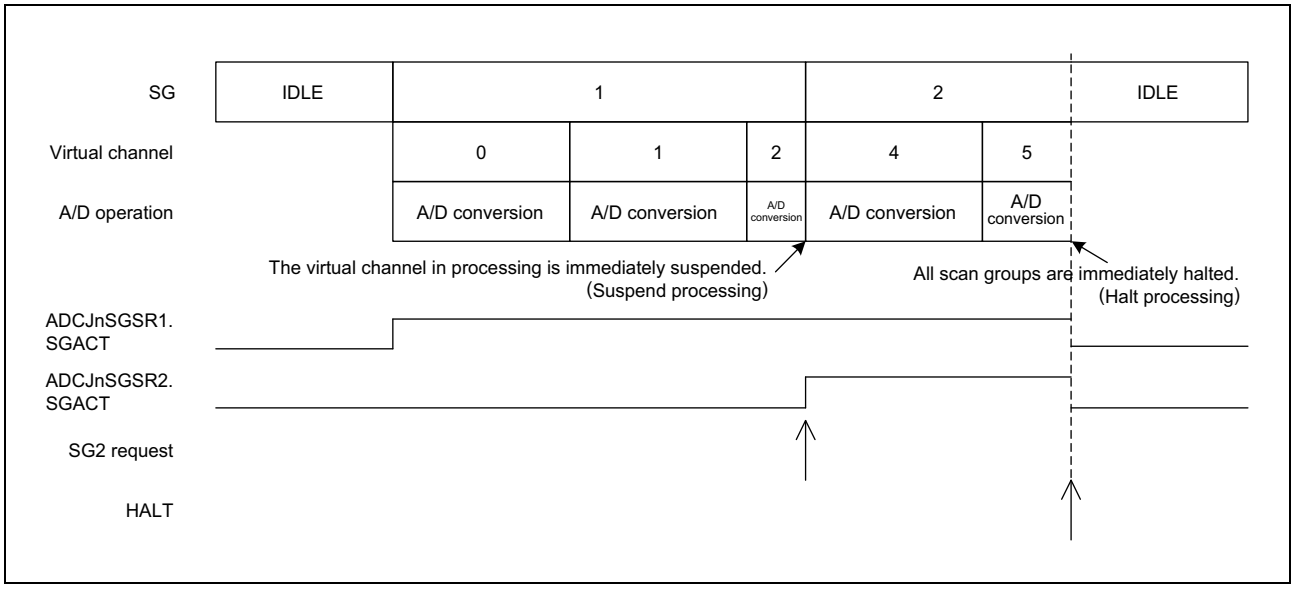


Figure 43.49 Example of Stopping All Scan Groups (Forced Termination)

43.4.15.2 Example of Stopping Scan Group x

When the scan group stop bit (ADCJnSGSTPCRx.SGSTP) is set to 1, the corresponding scan group terminates the processing virtual channel and then enters the idle state and clears the SGACT bit to 0. An example of operation is shown in **Figure 43.50**.

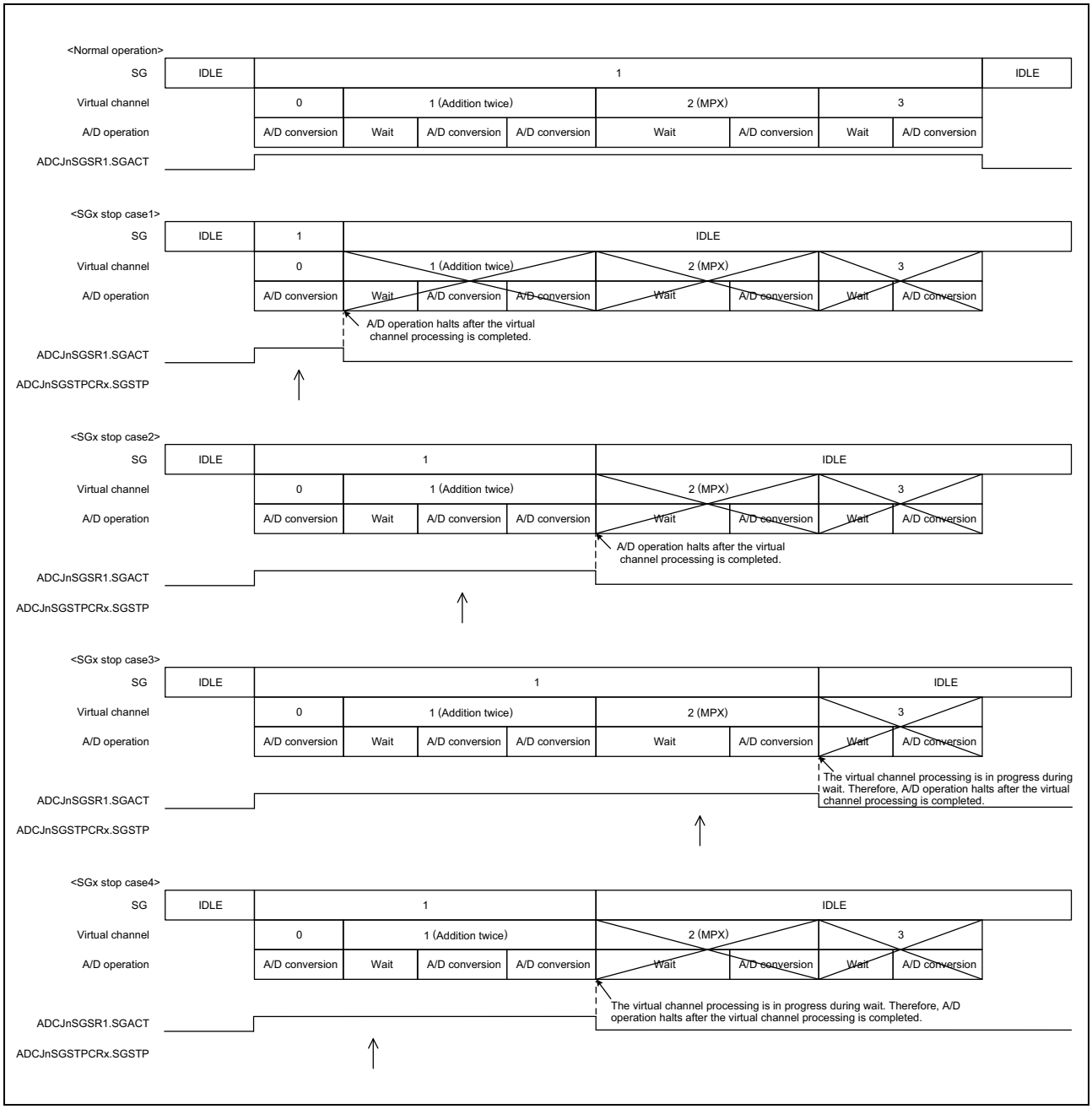


Figure 43.50 Example of Stopping Scan Group x 1

When the scan group stop bit (ADCJnSGSTPCRx.SGSTP) is set to 1 while a scan group applicable to T&H group A is performing T&H hold processing, the T&H hold processing halts after A/D conversion of the first virtual channel is performed.

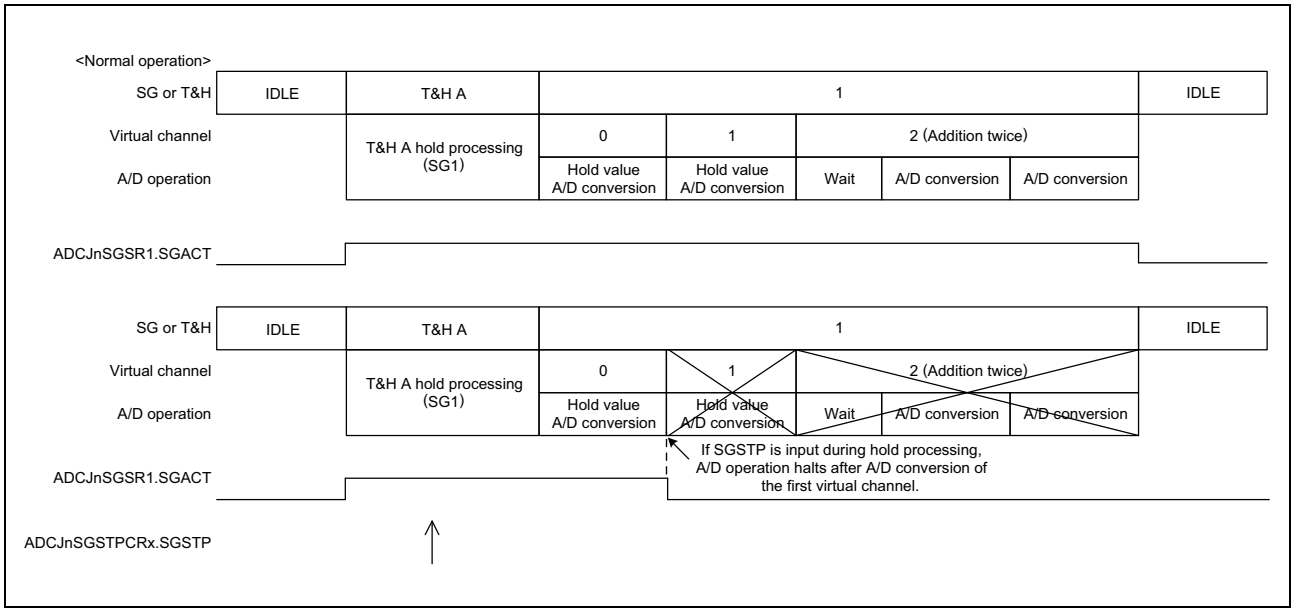


Figure 43.51 Example of Stopping T&H Target Scan Group x 2

When stopping multiple operating scan groups by the SGSTP bit, they halt immediately unless they are in processing (because they are suspended). Scan groups in processing halt after A/D conversion of the virtual channel in processing is completed.

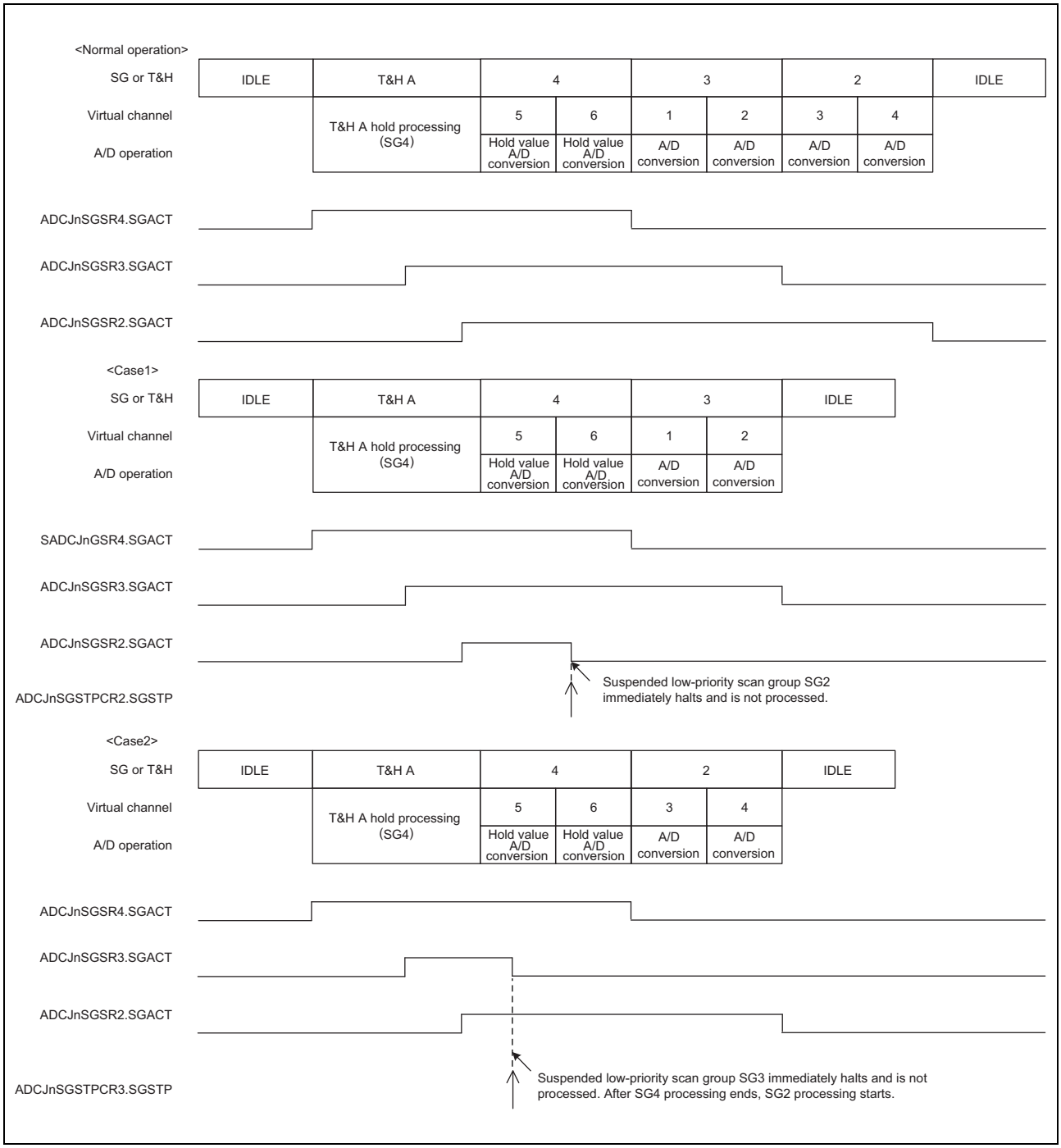


Figure 43.52 Example of Stopping Scan Group x when Multiple Scan Groups Are Operating

After hold processing of T&H that is target of the low priority scan group is executed, when the low priority scan group is stopped by SGSTP during the processing of high priority scan group, the low priority scan group is stopped as follows.

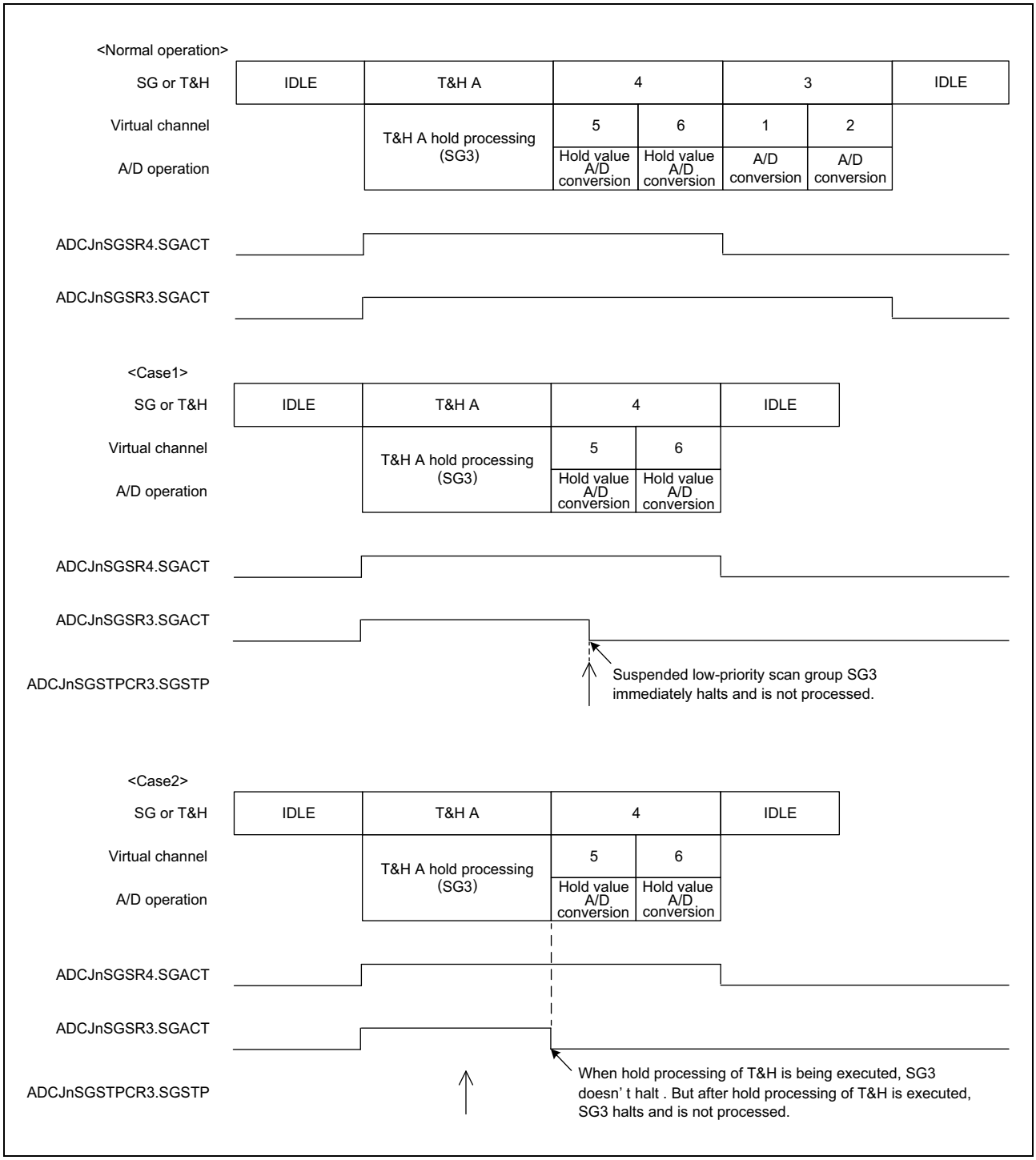


Figure 43.53 Example of Stopping Low Priority Scan Group x after hold processing of T&H is processing

43.4.16 PWM-Diag [Example of PWM-Diag]

The A/D conversion of ADCJ can be started by the request from PWM-Diag. The scan group (SG4) is appropriated for this function. The wait time of PWM-Diag is selected by WTTS in ADCJnPWDVCR. With the PWM-Diag function enabled by PWE in the ADCJnPWDCR register, A/D conversion is performed by the instruction from the PWM-Diag. Therefore, the instruction of A/D conversion for this function must be prepared in the PWM-Diag. At the end of A/D conversion, A/D conversion result is output to PWM-Diag with the data format setting by ADCJnADCR2.DFMT[2:0].
For details, see the PWM-Diag specification.

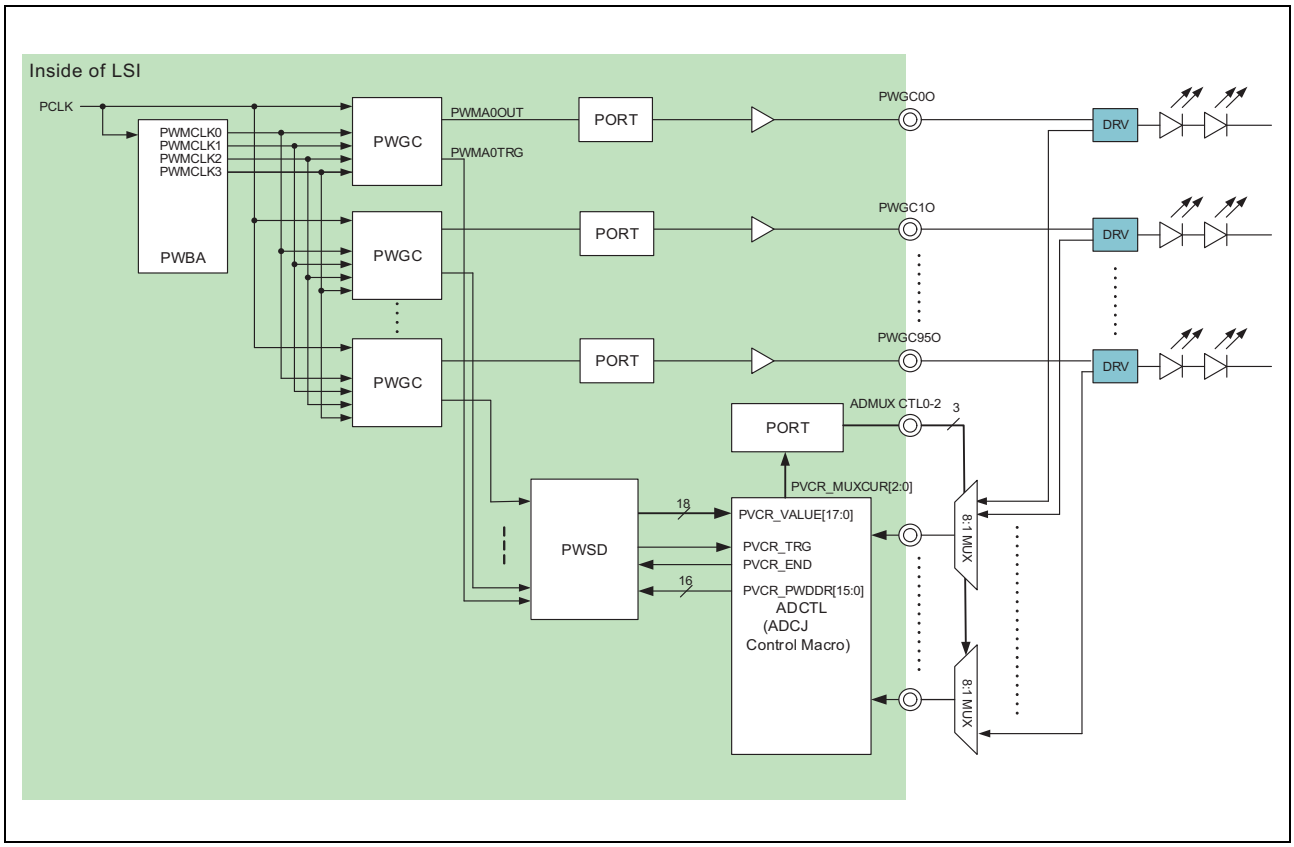


Figure 43.54 PWM-Diag Configuration

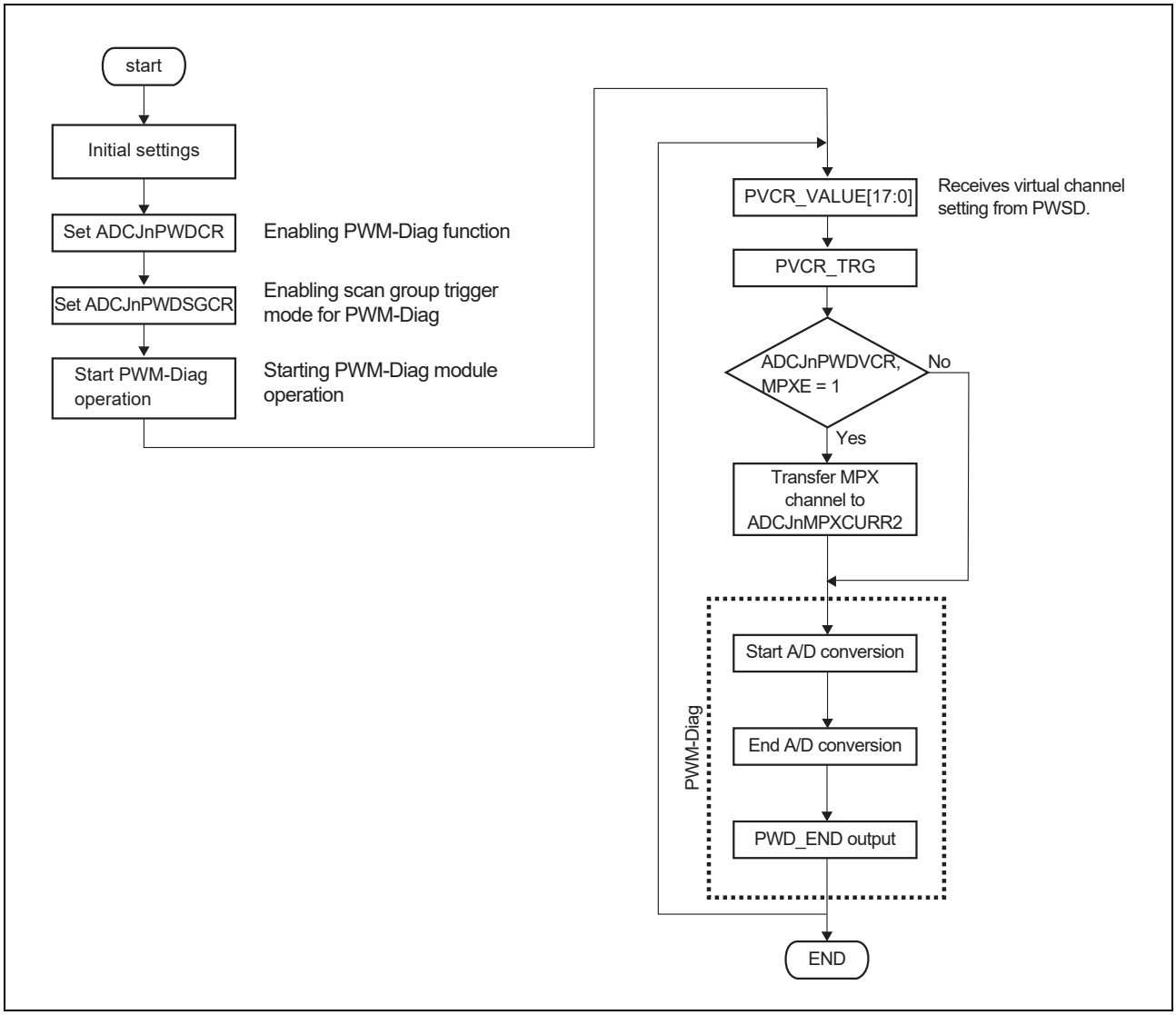


Figure 43.55 PWM-Diag Operation Flow

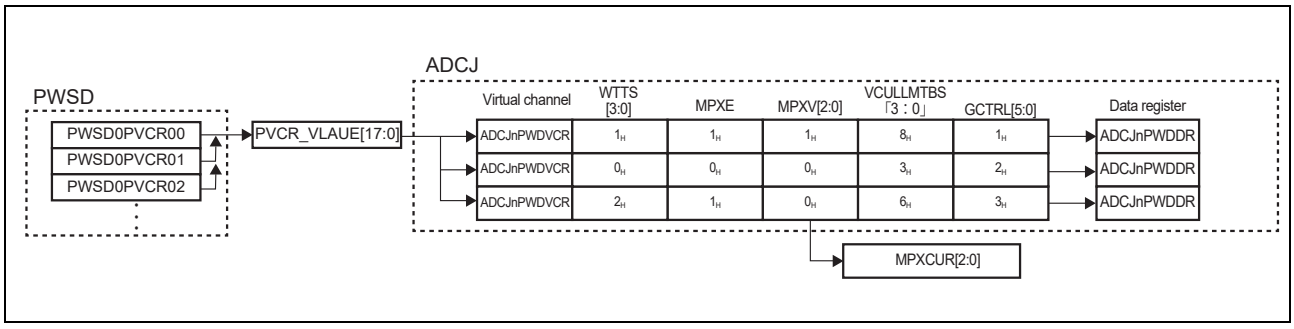


Figure 43.56 Data Transfer and Register Settings

Example 1: Example of operation shown in **Figure 43.56**

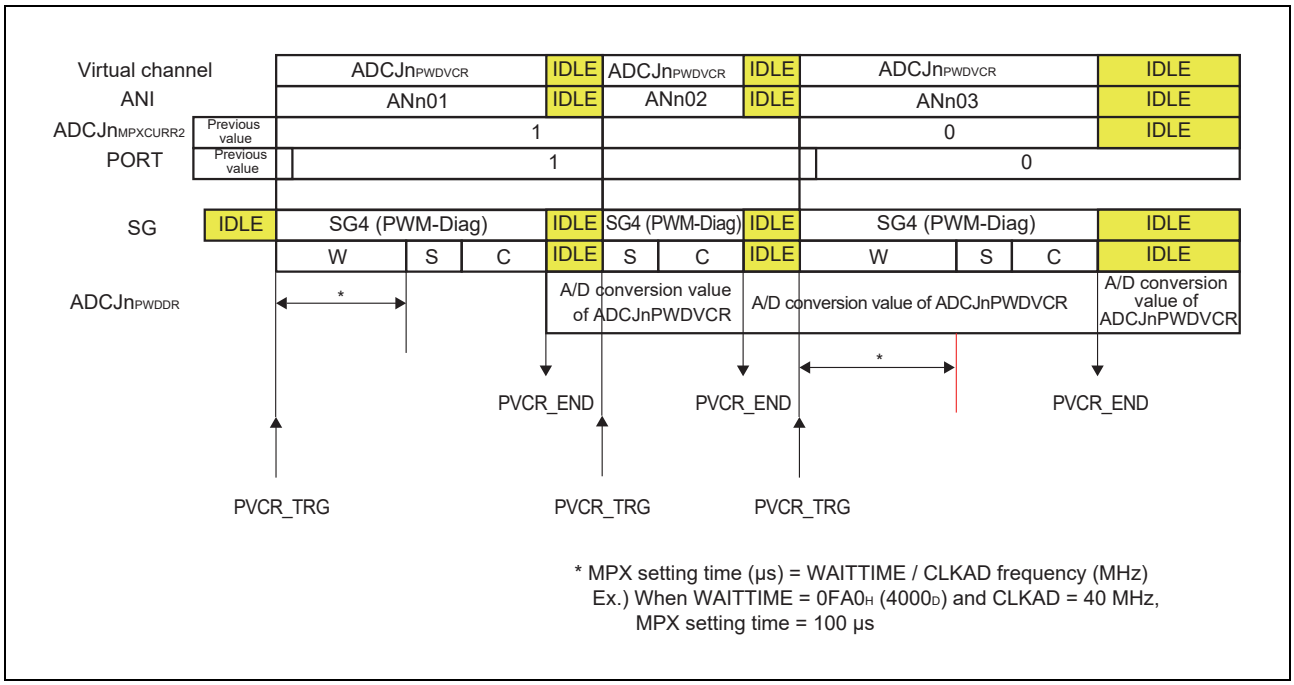


Figure 43.57 PWM-Diag Operation

43.4.17 Example of Suspend and Resume Operation

When a start trigger of a high-priority scan group is accepted during operation of a low-priority scan group, the suspension function suspends the operation of the low-priority scan group and executes the high-priority scan group.

After operation of the high-priority scan group is completed, the suspended low-priority scan group is resumed.

When a start trigger of a low-priority scan group is accepted during operation of a high-priority scan group, operation of the high-priority scan group continues and the low-priority scan group is suspended. After operation of the high-priority scan group is completed, the suspended low-priority scan group is started.

While a scan group is operating or suspended, a start trigger of the same scan group is not accepted.

Three suspension methods are provided.

The priority order of scan groups 0 to 4 is as follows:

Low High

SG0 < SG1 < SG2 < SG3 < SG4 (PWM-Diag*1)

Note 1. When ADCJnPWDCR.PWE = 1, SG4 operates in the PWM-Diag.

43.4.17.1 Example of Synchronous Suspend and Resume Operation

In synchronous suspension, after the A/D conversion operation of a virtual channel in a low priority scan group is completed, the scan group is suspended and then the A/D conversion of a high-priority scan group is started.

Upon completion of the A/D conversion of the high-priority scan group, the low-priority scan group is resumed from the suspended virtual channel.

Figure 43.58 shows an example of synchronous suspend operation.

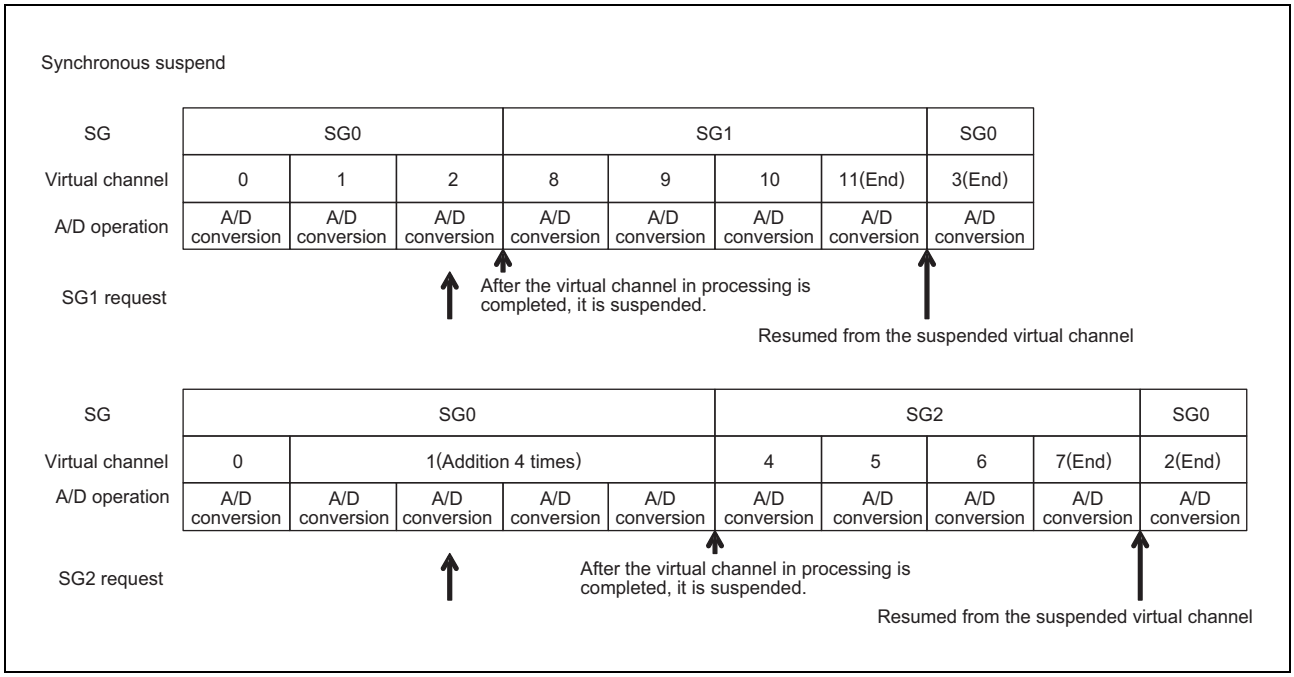


Figure 43.58 Example of Synchronous Suspend and Resume Operation

43.4.17.2 Example of Asynchronous Suspend and Resume Operation

In asynchronous suspension, the A/D conversion operation of a virtual channel in processing is immediately suspended before it is completed and the A/D conversion of a high-priority scan group is started.

Upon completion of the A/D conversion of all virtual channel in the high-priority scan group, the low-priority scan group is resumed from the suspended virtual channel.

Figure 43.59 shows an example of asynchronous suspend operation.

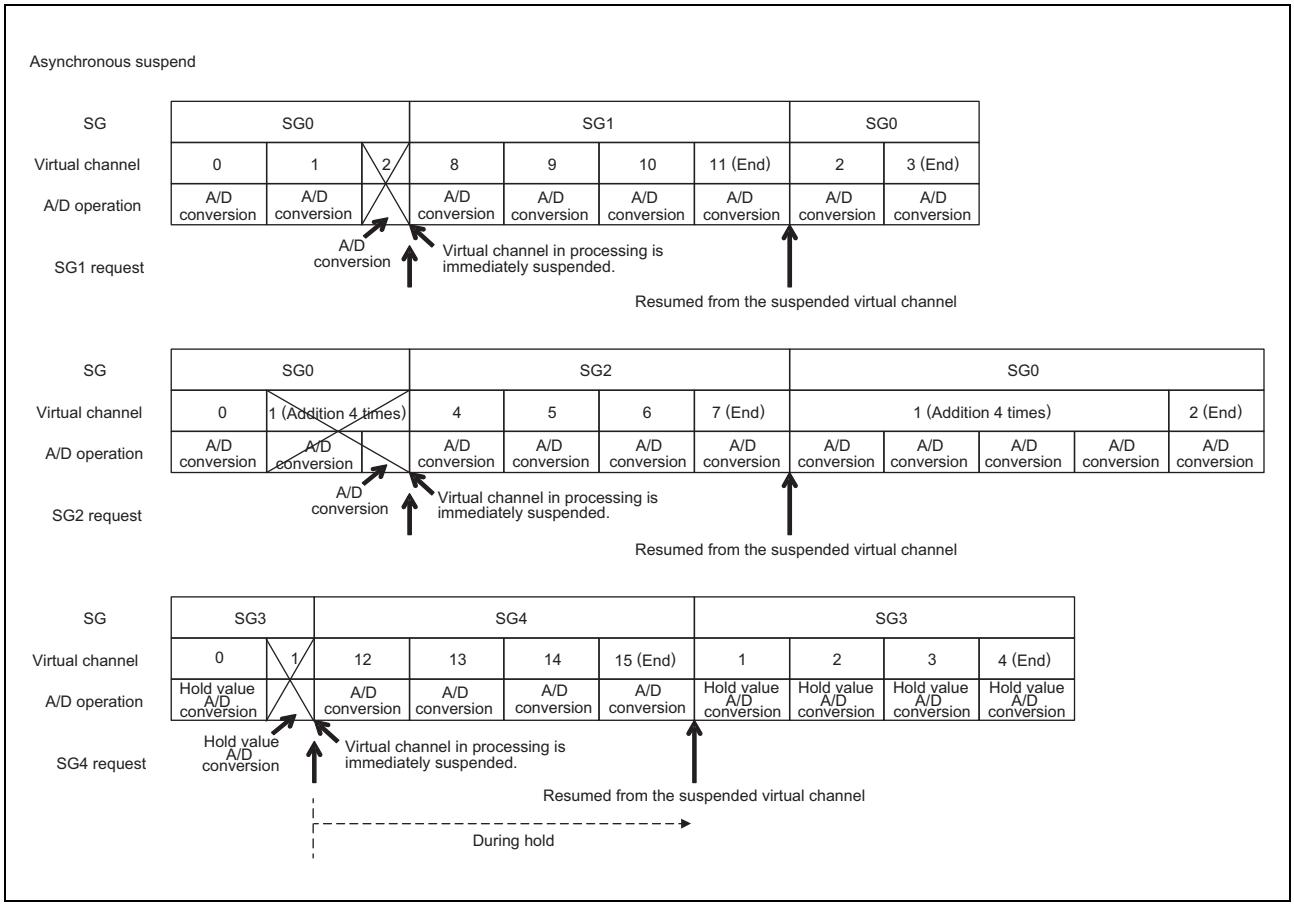


Figure 43.59 Example of Asynchronous Suspend and Resume Operation

43.4.17.3 Synchronous/Asynchronous Mixture Type Suspend

When a high-priority scan group interrupts scan group 0, the asynchronous suspension method is used.

When a high-priority scan group interrupts a scan group other than scan group 0, the synchronous suspension method is used.

Figure 43.60 shows an example of synchronous/asynchronous mixture type suspend operation.

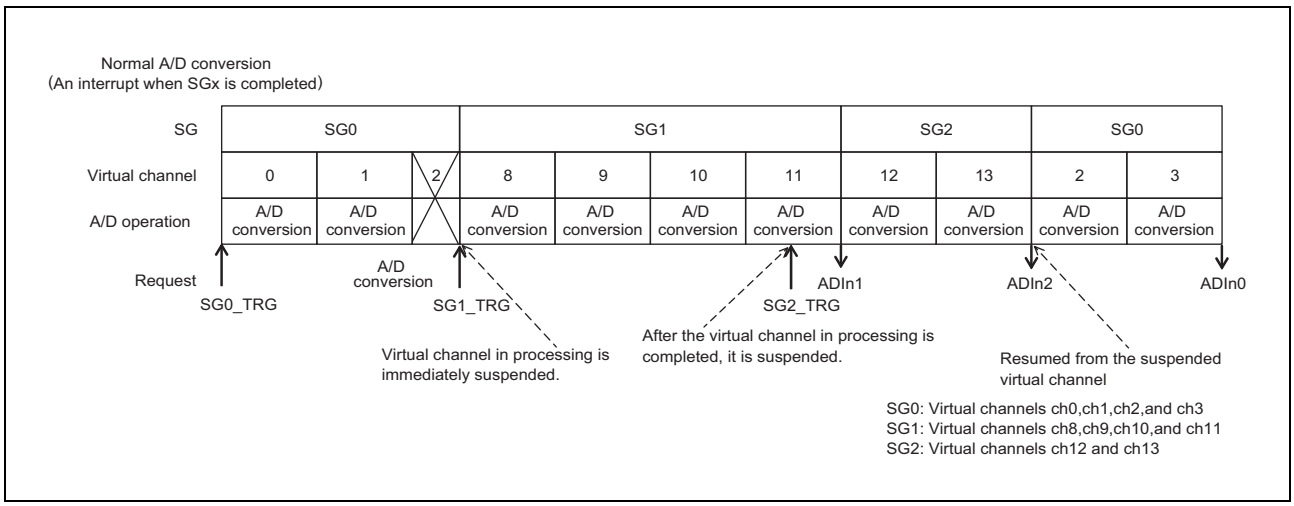


Figure 43.60 Example of Synchronous/Asynchronous Mixture Type Suspend Operation

43.4.18 Example of Track and Hold Operation (T&H)

The T&H processing function performs A/D conversion of multiple analog inputs at the same time. This function is achieved by holding analog input voltages simultaneously in up to four T&H circuits and then performing A/D conversion of outputs from the T&H circuits.

There is 1 group (T&H group A) for T&H processing. T&H sampling is activated by a software trigger for this group. T&H hold is activated by either software trigger or hardware trigger. The hardware trigger (sgx_trg) that activates scan groups also activates T&H hold.
Therefore, select a scan group to be a target of T&H group A from ADCJnTHACR.SGS[1:0]. Scan group 0 cannot be a target of T&H processing.

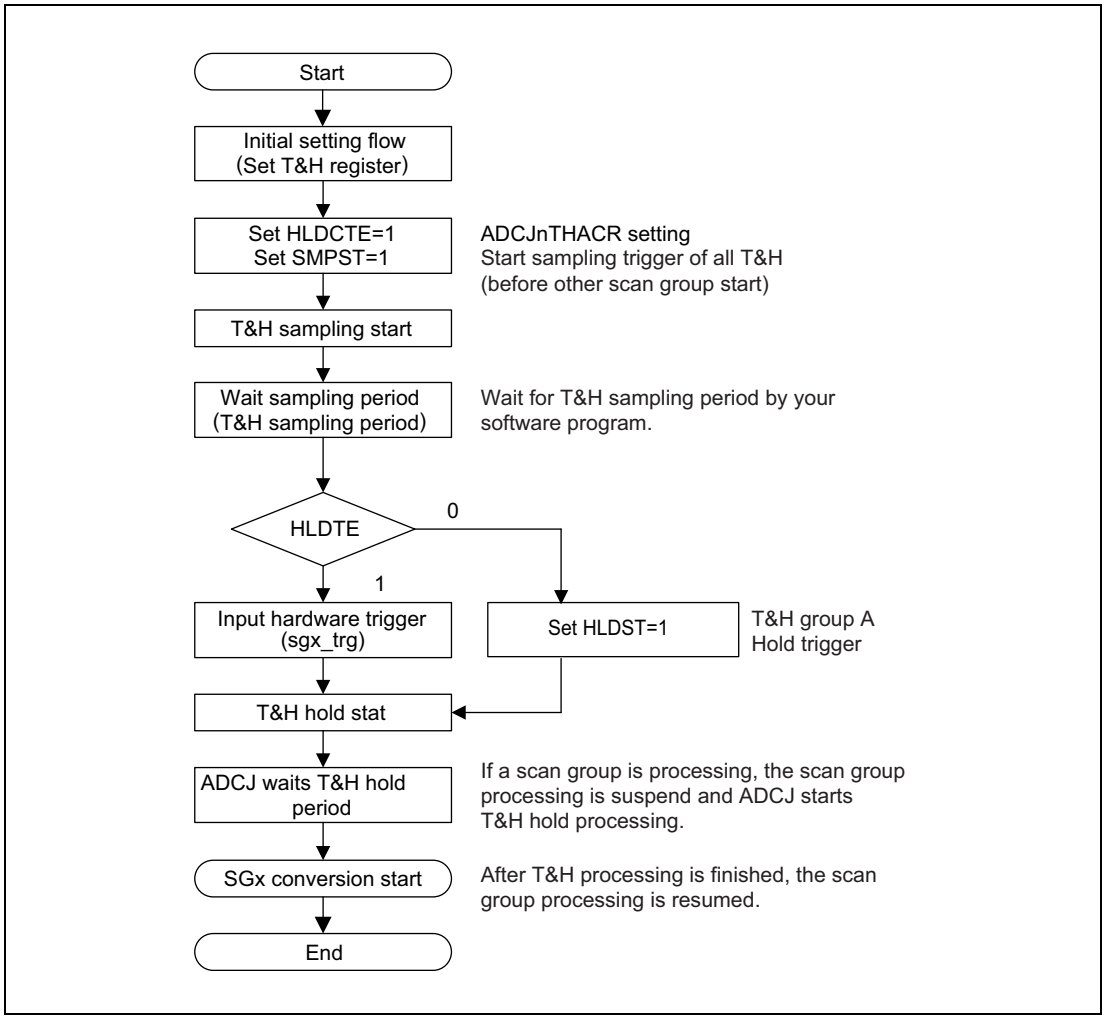


Figure 43.61 Recommended T&H Operation Flow

1. Set the T&H register according to the initial setting flow "**Section 43.4.1.1, Initial Setting Procedure**".
2. When using T&H group A in SG1/SG2 after initial settings have been completed, start T&H sampling and hold processing according to "**Section 43.4.2.2, SG1/SG2 Startup Flow**". When using T&H group A in SG3/SG4, start T&H sampling and hold processing according to "**Section 43.4.2.3, SG3/SG4 Startup Flow**".
3. After the T&H hold processing has been completed, perform A/D conversion.

CAUTIONS

1. **When using T&H, set ADCJnADCR1.SUSMTD[1:0] to 2_H. Other settings are prohibited.**
2. **Set ADCJnSGCRx.TRGMD of T&H target scan groups to 1_H. Other settings are prohibited.**
3. **When using T&H, select multicycle scan mode (ADCJnSGCRx.SCANMD = 0_H) for T&H target scan groups, and specify a single scan count (ADCJnSGMCYCRx.MCYC[7:0] = 0_H). Settings other than this combination are prohibited.**

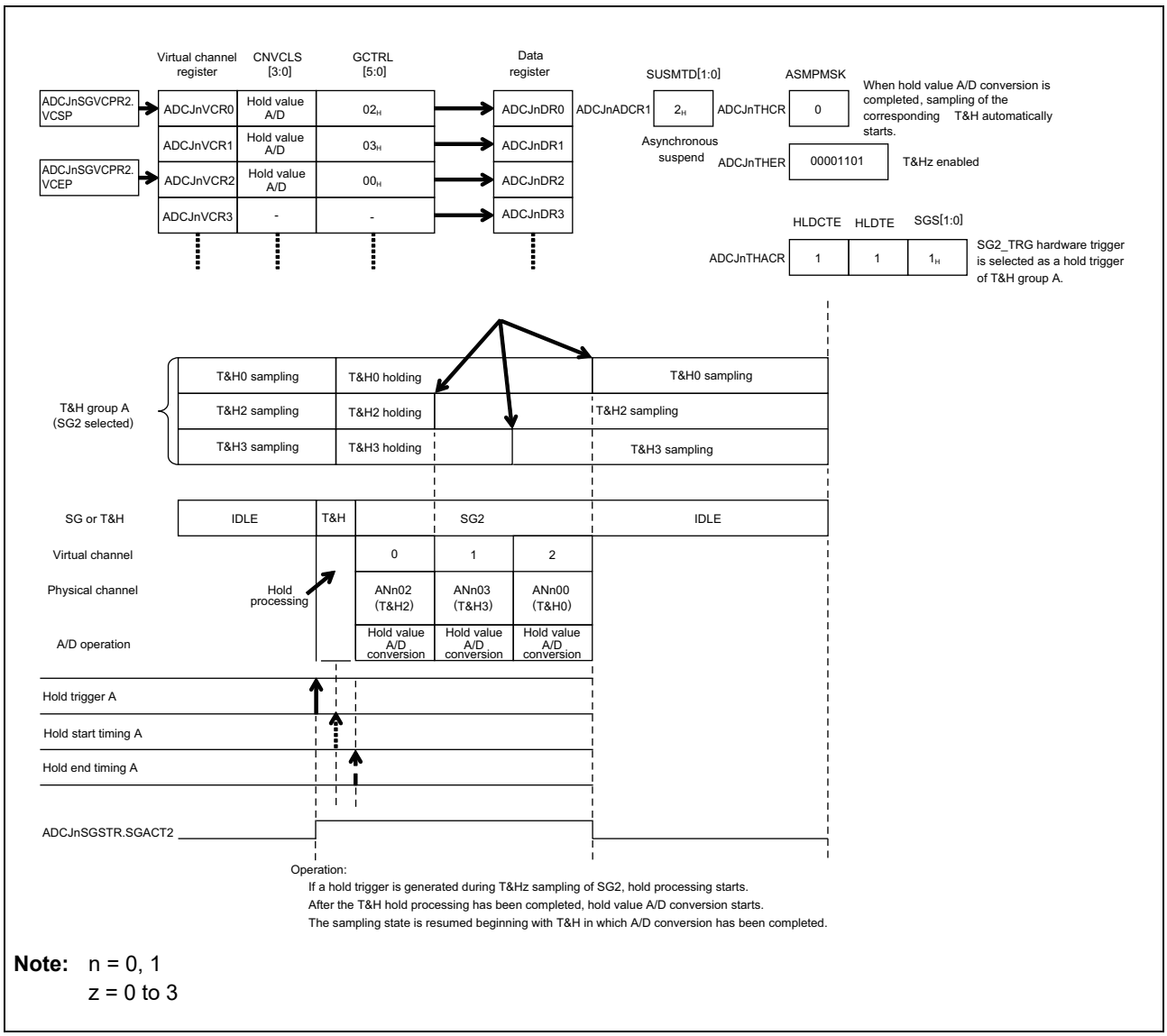
[Asynchronous suspend operation by T&H]

When starting T&H hold processing during scan group processing, T&H hold processing is started after suspension without waiting until the A/D conversion operation of the scan group is completed. After the T&H hold processing is completed, A/D conversion of the highest-priority scan group is resumed.

For details of operation, see Case1 to Case6 below.

- Case1: Using single scan group
- Case2: Asynchronously suspended by T&H hold trigger
- Case3: A trigger of high-priority scan group is input during T&H hold processing

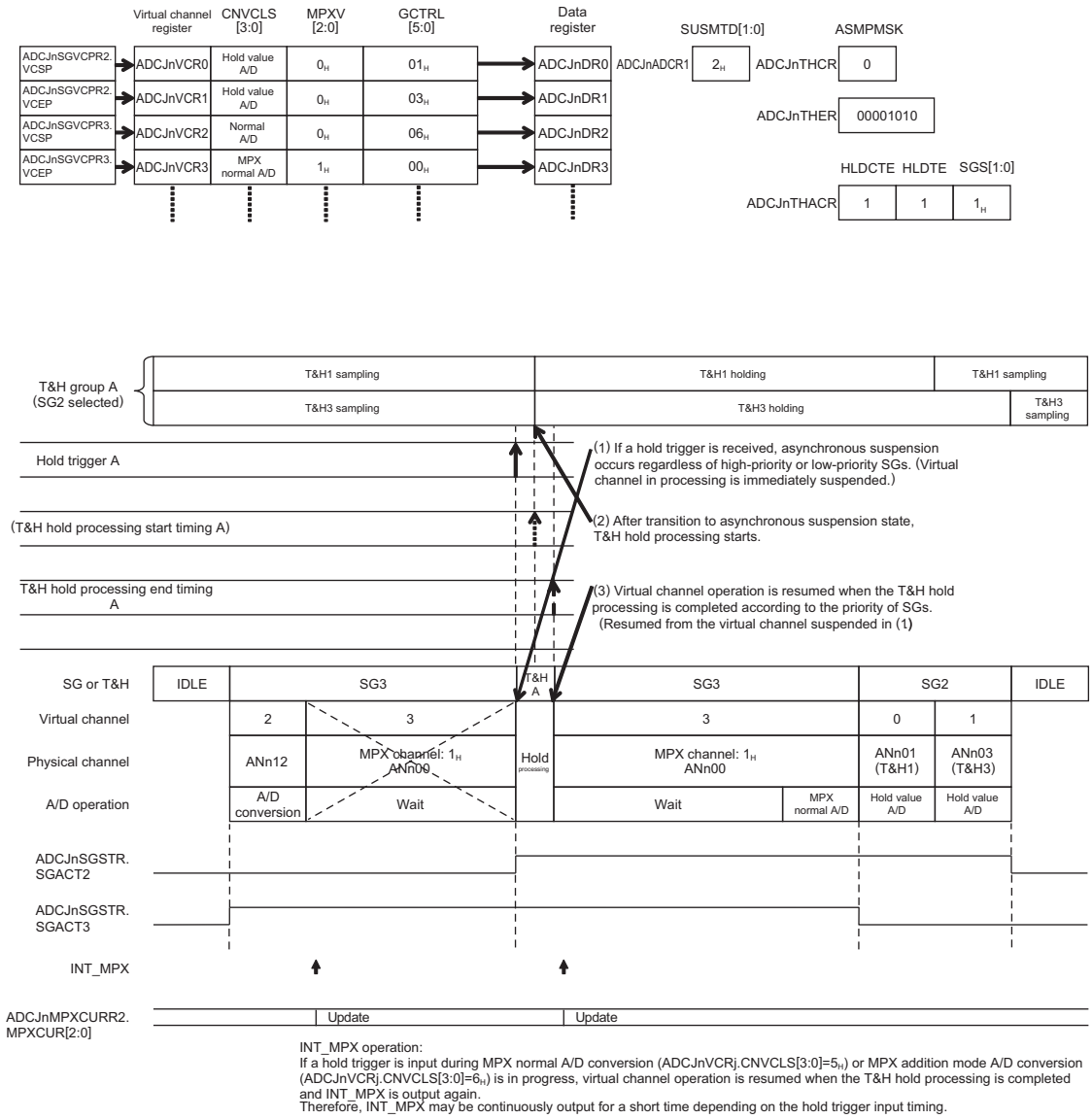
(1) Case1 Using a single group



Note: n = 0, 1
z = 0 to 3

Figure 43.62 Example of Track and Hold Processing (T&H Control / Case1)

(2) Case2 Asynchronously suspended by T&H hold trigger



Note: This example shows suspend and resume operation when a hold trigger is generated during the wait period. The same operation is performed when a hold trigger is generated during A/D conversion of other conversion types. Example: When a hold trigger is generated during addition mode A/D conversion, A/D conversion is resumed from the virtual channel suspended by the hold trigger (first addition) in the resume operation.

Note: n = 0, 1

Figure 43.63 Example of Track and Hold Processing (T&H Control / Case 2)

(3) Case3 A trigger of high-priority scan group is input during T&H hold processing

When a scan group is activated during the T&H hold processing, all scan groups are suspended until the T&H hold processing is completed. After the T&H hold processing has been completed, processing of the high-priority scan group is resumed.

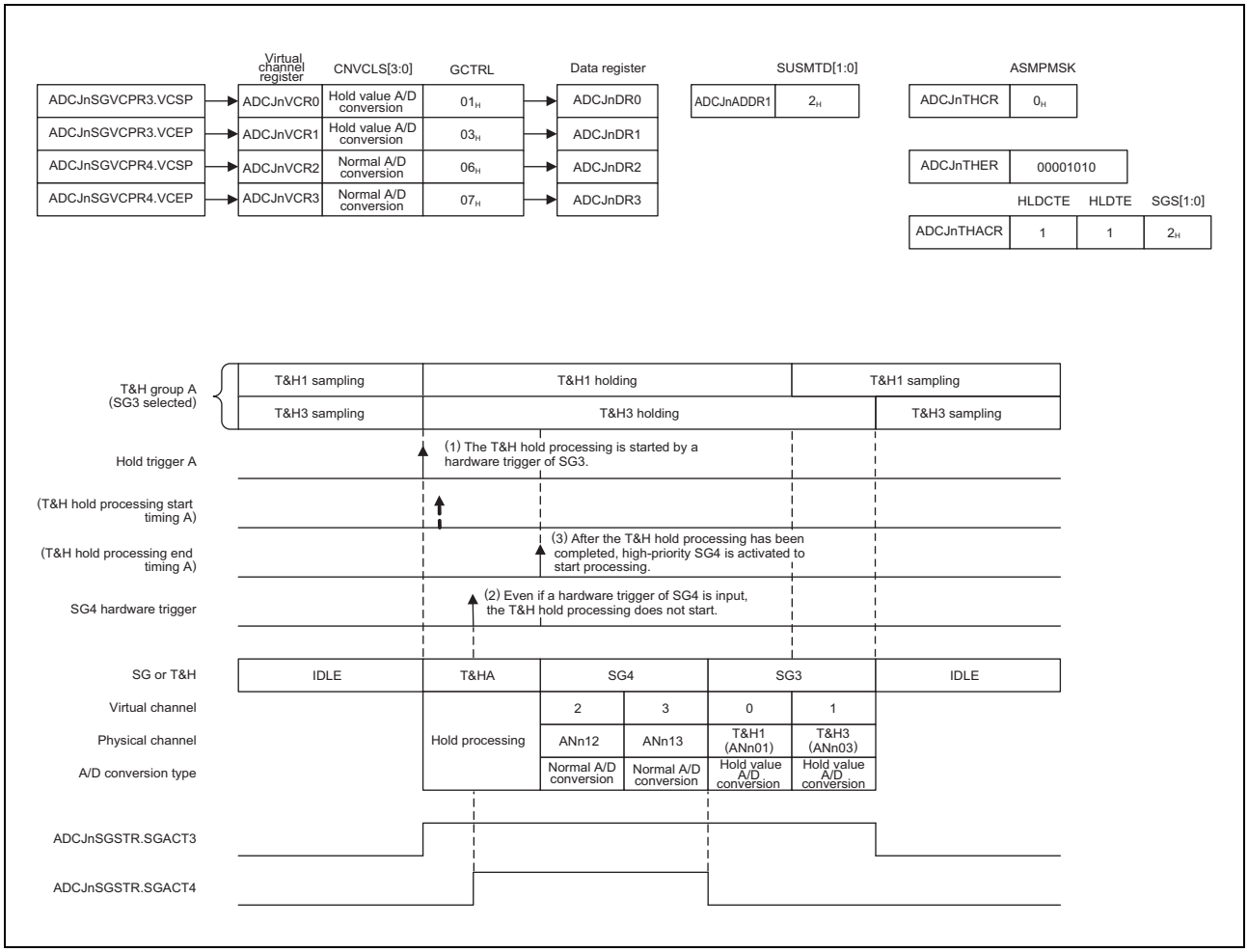


Figure 43.64 Example of Track and Hold Processing (T&H Control / Case 3)

43.4.19 Interrupt Request Functions

Interrupts of this IP include scan end interrupt, A/D error interrupt, upper/lower limit check interrupt (INT_UL), overwrite error interrupt, ID error interrupt, trigger overlap check error interrupt, parity error interrupt, and MPX interrupt.

43.4.19.1 Scan End Interrupt Request

(1) INT_ADx Interrupt Request

The ADCJ generates a scan end interrupt request (INT_ADx) at the end of scan group x*1 and at the end of ADCJnVCRj.

The scan end interrupt can activate the DMAC and DTS.

The INT_ADx output enable/disable setting is made by the following bits.

- Scan end interrupt enable bit (ADCJnSGCRx.ADIE) of scan group x
- Virtual channel end interrupt enable bit (ADCJnVCRj.ADIE) of the virtual channel register n

Table 43.87 INT_ADx Generation Enable/Disable Settings

ADCJnSGCRx.ADIE	ADCJnVCRj.ADIE	INT_ADx Output	Interrupt Request Generation Condition
0	0	Disabled	—
0	1	Enabled	At the end of A/D conversion of virtual channel n
1	0	Enabled	At the end of scan group x*1
1	1	Enabled	At the end of scan group x*1 or at the end of A/D conversion of virtual channel n

Note 1. When A/D conversion of the virtual channel specified by the scan group x virtual channel pointer register (ADCJnSGVCPRx.VCEP[5:0]) is completed

The INT_ADx signal is generated when the A/D conversion result is stored in the data register (ADCJnDRj).

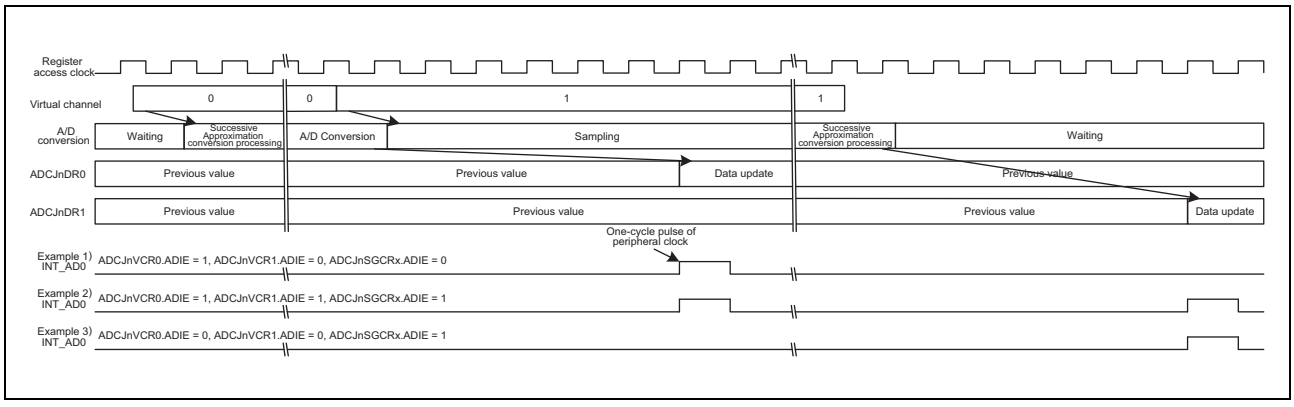


Figure 43.65 Example of Multicycle Scan Mode Operation

43.4.19.2 A/D Error Interrupt Request (INT_ADE)

When AD errors occur, the ADCJ generates an AD error interrupt (INT_ADE).

The generate timing of AD error interrupt is the same as the scan end interrupt generate timing except for trigger overlap check error.

After an interrupt occurs, clear the error status in the interrupt handler.

AD error interrupt is generated with all error factor that is set error interrupt enable bit of safety control register (ADCJnSFTCR) to valid as **Table 43.88**.

Table 43.88 INT_ADE Output Enable/Disable Settings

ADCJnSFTCR			INT_ADE	Interrupt Request Generation Condition
TOCEIE	OWEIE	IDEIE		
0	0	0	Disabled	—
1	*	*	Enabled	Trigger overlap check error
*	1	*	Enabled	Overwrite error
*	*	1	Enabled	ID error

Error status information is stored in the relevant error register.

When a trigger overlap check error is detected, a value of 1 is stored in the bit corresponding to the error scan group.

When an overwrite error are detected, a value of 1 is stored in the error bits and the virtual channel number in which the error occurred is stored in the capture bit.

When an ID error is detected, a value of 0 is stored in IDEF in the data supplementary information register (ADCJnDIRj or ADCJnPWDDIR) corresponding to the virtual channel (ADCJnVCRj or ADCJnPWDVCR) in which the ID error was detected. When no ID error is detected, a value of 1 is stored. Furthermore, when an ID error is detected, a value of 1 is stored in the error bit in the ID error register (ADCJnIDER) and the virtual channel number in which an ID error was detected first is stored in the capture bit.

Error status information except for ADCJnDIRj.IDEF and ADCJnPWDDIR.IDEF is retained until the flag bit of the corresponding error is cleared by the error clear register. Information ADCJnDIRj.IDEF and ADCJnPWDDIR.IDEF is retained until the flag bit of the corresponding error is read with the read and clear enable bit (RDCLRE) set to 1.

If the same error occurs again with the error remaining, the subsequent error information is discarded. However, if the interrupt enable bit is set to 1, an AD error interrupt is output even a subsequent error occurs.

Table 43.89 INT_ADE Error Source Registers

Error Name	Error Register		Error Clear Register
	Flag Bit	Capture Bit	Clear Bit
Trigger overlap check error	SG0 : ADCJnTOCER.TOCESG[0]	None	SG0 : ADCJnECR.TOCESGC[0]
Trigger overlap check error	SG1 : ADCJnTOCER.TOCESG[1]	None	SG1 : ADCJnECR.TOCESGC[1]
Trigger overlap check error	SG2 : ADCJnTOCER.TOCESG[2]	None	SG2 : ADCJnECR.TOCESGC[2]
Trigger overlap check error	SG3 : ADCJnTOCER.TOCESG[3]	None	SG3 : ADCJnECR.TOCESGC[3]
Trigger overlap check error	SG4 : ADCJnTOCER.TOCESG[4]	None	SG4 : ADCJnECR.TOCESGC[4]
Overwrite error	ADCJnOWER.OWE	ADCJnOWER.OWECAP[5:0]	ADCJnECR.OWEC
ID error	ADCJnDIRj.IDEF	None	None*1
ID error	ADCJnPWDDIR.IDEF	None	None*1
ID error	ADCJnIDER.IDE	ADCJnIDER.IDECAP[5:0]	ADCJnECR.IDEC

Note 1. When the data supplementary information registers (ADCJnDIRj, ADCJnPWDDIR) is read with RDCLRE being set 1, the register is cleared to 0.

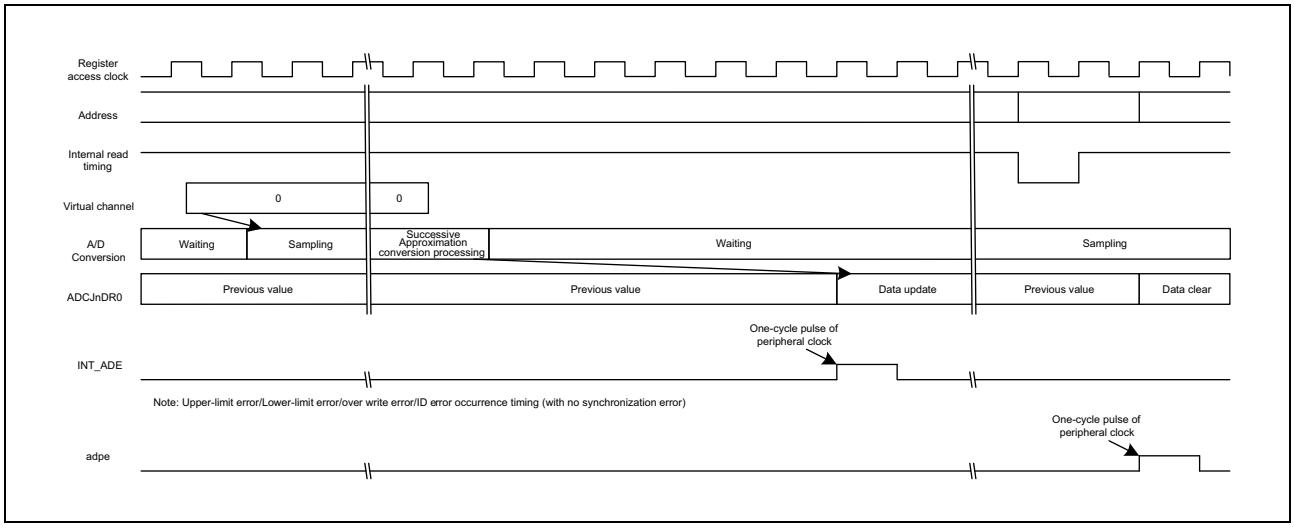


Figure 43.66 A/D Error Interrupt and Parity Error Interrupt Timing

(1) Trigger Overlap Check Interrupt Request

The trigger overlap check function is one of the generation factor of A/D error interrupt request.

When a startup trigger of the same scan group is enabled during scan group operation (ADCJnSGSRx.SGACT = 1), an A/D error interrupt request due to trigger overlap check is output.

When an A/D error interrupt occurs, reading the trigger overlap check error status register (ADCJnTOCER) makes it possible to check scan groups in which a trigger overlap check error occurred.

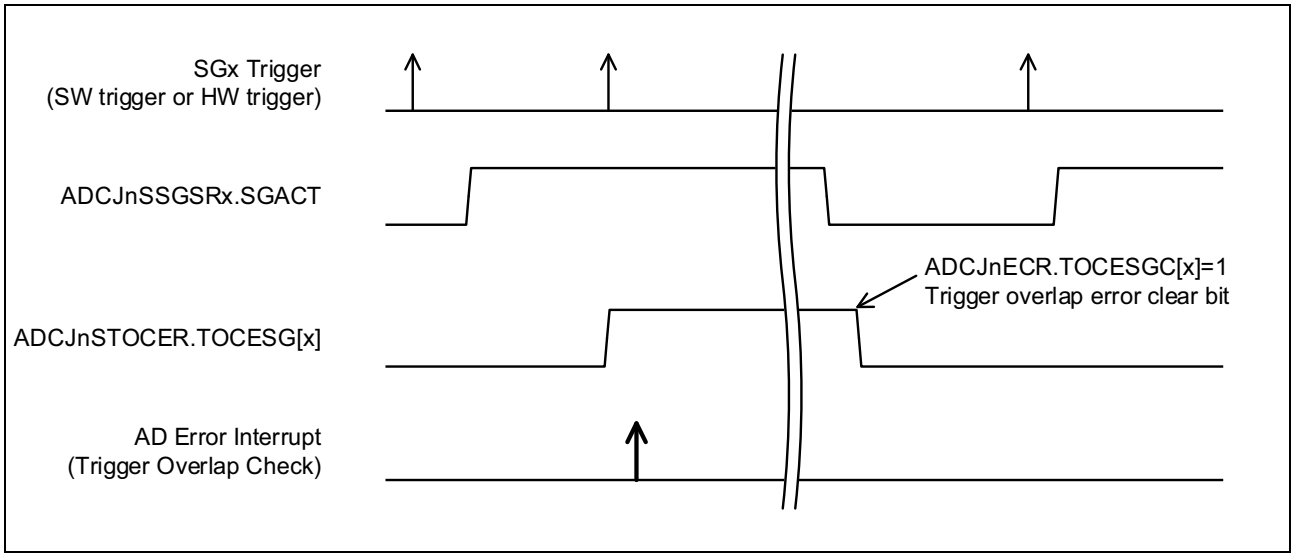


Figure 43.67 Trigger Overlap Check Error Interrupt Request

(2) Overwrite Error

This error indicates that a new A/D conversion result has been overwritten before the data register (ADCJnDRj) is read.

If a new A/D conversion result is stored in the relevant data register (ADCJnDRj) when the write flag (ADCJnDIRj.WFLAG) in the data supplementary information register = 1, an overwrite error is detected.

(3) ID Error

When data is stored in each channel's Data Supplementary Information Register (ADCJnDIRj, ADCJnPWDDIR), ID error is detected when there is inconsistency between feedback signal of I/O buffer/RRAMP/Track and Hold circuit and internal state of ADCJ.

ID error shows that operation failures occurred in the internal analog switch control of I/O buffer/ RRAMP/T&H circuit.

The ID error detection result is stored in the ID error bit in the data supplementary information registers (ADCJnDIRj, ADCJnPWDDIR) of the virtual channel in which an ID error was detected.

Furthermore, the virtual channel number in which an ID error was detected first and ID error information are stored in the ID error register (ADCJnIDER).

43.4.19.3 Parity Error Interrupt Request

When a parity error occurs, the ADCJ can generate a parity error notification (adpe) to the ECM. Parity of ADCJ is even parity. Parity error notification is generated when parity error occurs and when parity error interrupt enable bit of safety control register (ADCJnSFTCR.PEIE) is set to 1. Parity error is detected by the following procedure of **(1)**, **(2)** in ADCJ.

(1) When A/D conversion result is stored, a parity is generated for target bit of data supplementary information registers. After that, the result is stored in parity bit (ADCJnDIRj.PRTY). **Table 43.90** shows the parity target bits of each data supplementary information register.

Table 43.90 Parity Error Target Bits

Parity Bits in Data Supplementary Information Registers	Bits in Data Supplementary Information Registers				A/D Conversion Value	
	MPXE	MPXV[2:0]	ID[5:0]	IDEF	ADCJnDRj.DRj[15:0]	ADCJnPWDDR.PWDDR[15:0]
ADCJnDIRj.PRTY	√*1	√*1	√	√	√	–
ADCJnPWDDIR.PRTY	√*1	√*1	√	√	–	√

Note: √: Applicable, –: Inapplicable

Note 1. MPXE and MPXV[2:0] are target of the parity check only when MPX normal A/D conversion and MPX addition A/D conversion are executed.

(2) When the data register (ADCJnDRj or ADCJnPWDDR) or the data supplementary information register (ADCJnDIRj or ADCJnPWDDIR) is read, the parity check is executed. Parity error is output when this parity check result is abnormal.

The error status information is stored in the parity error register (ADCJnPER).
When a parity error is detected, the parity error flag bit (ADCJnPER.PE) is set to 1 and the number of the virtual channel j in which the parity error occurred is stored in the parity error capture bits (ADCJnPER.PECAP[5:0]).

The error status information is cleared by writing 1 to the parity error clear bit (ADCJnECR.PEC) in the error clear register.

If other virtual channel's parity error occurs without clearing the error, this parity error information is discarded. However, when the parity error interrupt enable bit is set to 1, a parity error notification is output to the ECM also when this parity error occurs.

CAUTION

After the register bits that shows in Table 43.91 are changed after storing A/D conversion result, if data register or data supplementary register is read, pseudo parity error may occur.

Table 43.91 The register bit that causes pseudo parity error and several example settings

Target register bit	Several Example Settings	
	Value Setting	Setting Explanation
ADCJnVCRj.CNVCLS[3:0]	$4_H \Rightarrow 0_H$	Change from addition mode to normal mode
ADCJnADCR2.ADDNT	$1_B \Rightarrow 0_B$	Change from 4 times addition to 2 times addition
ADCJnODCR.WADDE	$1_B \Rightarrow 0_B$	Change from addition mode of wiring-break detection to normal mode of wiring-break detection

43.4.19.4 MPX Interrupt Request

The ADCJ can generate an MPX interrupt request (INT_MPX).

When MPX interrupt is enabled (ADCJnMPXINTER.ADMPXIE = 1), an interrupt request INT_MPX is generated while MPX normal A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 5_H) or MPX addition mode A/D conversion (ADCJnVCRj.CNVCLS[3:0] = 6_H) is in progress.

Occurrence of INT_MPX can activate the DMAC.

When MPX interrupts are disabled (ADCJnMPXINTER.ADMPXIE = 0), no INT_MPX interrupt request is output.

The output port MPXCUR[2:0] value is updated regardless of the ADMPXIE value in ADCJnMPXINTER.

The register (ADCJnMPXCURR1.MPXCUR[2:0]) is updated at the same time as the output port MPXCUR[2:0] update.

43.4.19.5 Upper/Lower Limit Check

The ADCJ can generate an interrupt request signal (INT_UL), over upper-limit threshold value notice signal (vculmo), under lower-limit threshold value notice signal (vcllmo), and end notification signal (Refer to **Table 43.92**) for each virtual channel after detecting that the A/D conversion result is above the upper-limit threshold value or below the lower-limit threshold value.
Also, it's possible to notify the ADC VMON Secondary Error generator (AVSEG).

NOTE

The End Notification Signal indicates the signal name of **Table 43.92**.

Table 43.92 Correspondence Table of End Notification Signal

Signal name	Target scan group	Explanation
vcend[j]	Scan group x	Virtual channel A/D conversion end notice
pvcr_end	PWM-Diag	PWM-Diag A/D conversion end notice

(1) Setting Registers

Functions of INT_UL, vculmo, vcllmo, and end notification signals can be enabled or disabled for each virtual channel by setting.

When they are enabled, the A/D conversion result is compared with the upper/lower limit check table registers ADCJnVCULLMTBRy and signals INT_UL, vculmo, vcllmo, and end notification signals are output.

Table 43.93, **Table 43.94** list settings to enable or disable INT_UL, vculmo, vcllmo, and end notification signals functions.

Table 43.93 Settings to Enable/Disable INT_UL, vculmo, vcllmo, and vcend[j] Functions of ADCJnVCRj

Register Setting		Output Pin			
ADCJnVCLMINTER1/2. ADULnIE	ADCJnVCRj. VCULLMTBS	INT_UL	vculmo	vcllmo	vcend[j]
0	0_H	Disabled	Disabled	Disabled	Disabled
0	1_H to 8_H	Disabled	Enabled	Enabled	Enabled
1	0_H	Disabled	Disabled	Disabled	Disabled
1	1_H to 8_H	Enabled	Enabled	Enabled	Enabled

Table 43.94 Settings to Enable/Disable INT_UL Functions of PWM-Diag

Register Setting		Output Pin
ADCJnPWVCLMINTER.PWADULIE	ADCJnPWDVCR.VCULLMTBS	INT_UL
0	0_H	Disabled
0	1_H to 8_H	Disabled
1	0_H	Disabled
1	1_H to 8_H	Enabled

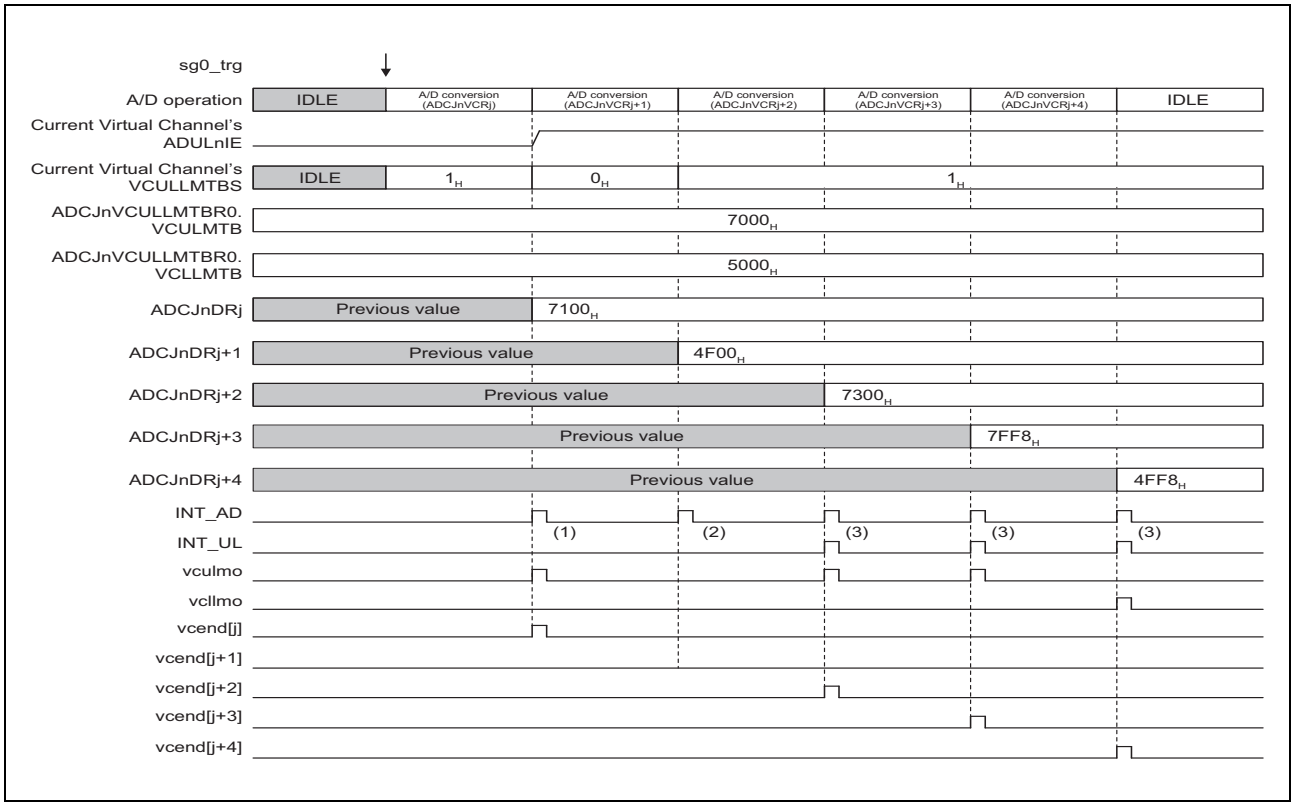


Figure 43.68 Upper/Lower Limit Check Signal Output Conditions

(1) When ADULnIE = 0, INT_UL is not output, but when ADCJnVCRj.VCULLMTBS = 1_H to 8_H, vculmo, vcllmo, and vcend[j] are output.

(2) When ADCJnVCRj.VCULLMTBS = 0_H, INT_UL, vculmo, vcllmo, and vcend[j] are not output regardless of the ADULnIE setting.

(3) When ADULnIE = 1 and ADCJnVCRj.VCULLMTBS = 1_H to 8_H, INT_UL, vculmo, vcllmo, and vcend[j] are output.

When the conversion type is addition mode A/D conversion or MPX addition mode A/D conversion, INT_UL, vculmo, vcllmo, and vcend[j] are output only at the end of the predetermined addition count.

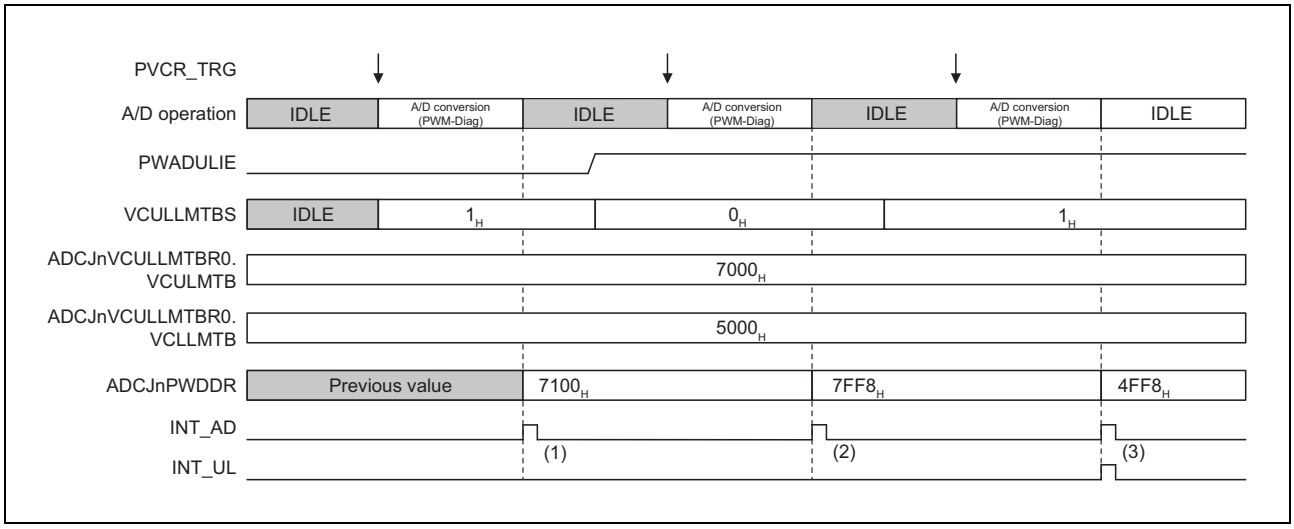


Figure 43.69 Upper/Lower Limit Check Signal Output Conditions (PWM-Diag)

(1) When PWADULIE = 0, INT_UL is not output.

(2) When ADCJnPWDVCR.VCULLMTBS = 0_H, INT_UL is not output regardless of the PWADULIE setting.

(3) When PWADULIE = 1 and ADCJnPWDVCR.VCULLMTBS = 1_H to 8_H, INT_UL is output.

(2) Table Register

ADCJnVCULLMTBRy is selected by upper/lower limit check table register select bits (Refer to **Table 43.96**).

Combinations of upper-limit threshold values and lower-limit threshold values are provided as sets.

No different table register number can be selected for the upper-limit side and lower-limit side of a virtual channel.

Set the upper-limit threshold value in the upper 16 bits and set the lower-limit threshold value in the lower 16 bits of the selected ADCJnVCULLMTBRy.

Write 32-bit values. Writing 16-bit or 8-bit values is prohibited.

ADCJnVCULLMTBRy can be modified at any time regardless of whether the scan group is active or not.

Table 43.95 Relationship between ADCJnVCRj.VCULLMTBS[3:0] and ADCJnVCULLMTBy

VCULLMTBS	Register Name	Upper-Limit Side Notice Condition	Lower-Limit Side Notice Condition
0_H	Upper/lower limit check disabled		
1_H	ADCJnVCULLMTBR0	See **Table 43.99, Output condition (1) of upper/lower limit check notification**.	See **Table 43.99, Output condition (1) of upper/lower limit check notification**.
2_H	ADCJnVCULLMTBR1		
3_H	ADCJnVCULLMTBR2		
4_H	ADCJnVCULLMTBR3		
5_H	ADCJnVCULLMTBR4		
6_H	ADCJnVCULLMTBR5		
7_H	ADCJnVCULLMTBR6		
8_H	ADCJnVCULLMTBR7		
Other than above	Setting prohibited		

NOTES

1. The Upper/Lower Limit Check Table Register Select bit indicates all bit name of **Table 43.96**.

Table 43.96 Correspondence Table of Upper/Lower Limit Check Table Register Select Bit

Bit name	Target scan group	Explanation
ADCJnVCRj.VCULLMTBS[3:0]	Scan group x	Virtual channel j Upper/Lower Limit Check Table Register Select bit
ADCJnPWDVCR.VCULLMTBS[3:0]	PWM-Diag	PWM-Diag Upper/Lower Limit Check Table Register Select bit

2. Data register indicates all bit name of **Table 43.97**.

Table 43.97 Correspondence Table of Data Register

Bit name	Target scan group	Explanation
ADCJnDRj[15:0]	Scan group x	Virtual channel j Data register
ADCJnPWDDR[15:0]	PWM-Diag	PWM-Diag Data register

Table 43.98 Relationship between ADCJnPWDVCR.VCULLMTBS[3:0] and ADCJnVCULLMTBy

VCULLMTBS	Register Name	Upper-Limit Side Notice Condition	Lower-Limit Side Notice Condition
0_H	Upper/lower limit check disabled		
1_H	ADCJnVCULLMTBR0	See **Table 43.99, Output condition (1) of upper/lower limit check notification** and **Table 43.100, Output condition (2) of upper/lower limit check notification**.	See **Table 43.99, Output condition (1) of upper/lower limit check notification** and **Table 43.100, Output condition (2) of upper/lower limit check notification**.
2_H	ADCJnVCULLMTBR1		
3_H	ADCJnVCULLMTBR2		
4_H	ADCJnVCULLMTBR3		
5_H	ADCJnVCULLMTBR4		
6_H	ADCJnVCULLMTBR5		
7_H	ADCJnVCULLMTBR6		
8_H	ADCJnVCULLMTBR7		
Other than above	Setting prohibited		

Table 43.99 shows output condition of upper/lower limit check notification of Data Register. A/D conversion result and the upper/lower limit value are compared according to the data format by ADCJnADCR2.DFMT.

Table 43.99 Output condition (1) of upper/lower limit check notification

DFMT	Number of additions	Output condition of Upper Limit Check	Output condition of Lower Limit Check
000_B	Convert once	Data register[15:0] > VCULMTB[15:0]	Data register[15:0] < VCLLMTB[15:0]
	Convert twice		
	Convert 4 times		
001_B	Convert once	Data register[15:0] > VCULMTB[15:0]	Data register[15:0] < VCLLMTB[15:0]
	Convert twice		
	Convert 4 times		
010_B	Convert once	Data register[15:0] > VCULMTB[15:0]	Data register[15:0] < VCLLMTB[15:0]
	Convert twice		
	Convert 4 times		

Table 43.100 Output condition (2) of upper/lower limit check notification

DFMT	Number of additions	Output condition of Upper Limit Check	Output condition of Lower Limit Check
100_B	Convert once	Data register[15:0] > VCULMTB[15:0]	Data register[15:0] < VCLLMTB[15:0]
	Convert twice		
	Convert 4 times		
101_B	Convert once	{0000, Data register[15:4]} > VCULMTB[15:0]	{0000, Data register[15:4]} < VCLLMTB[15:0]
	Convert twice	{000, Data register[15:3]} > VCULMTB[15:0]	{000, Data register[15:3]} < VCLLMTB[15:0]
	Convert 4 times	{00, Data register[15:2]} > VCULMTB[15:0]	{00, Data register[15:2]} < VCLLMTB[15:0]

43.4.20 Function to Detect Overvoltage in Chip Standby Mode (LPS Operation)

As main assumption, ADCJ2 activates by intermittent trigger from LPS operation during Chip standby mode without CPU operation.
Because the register access clock does not operate during the Chip standby mode, ADCJ2 operates by CLKAD (CLKA_ADC) to continue LPS operation. And all register settings necessary for LPS operation must be completed before transition to Chip standby mode.

ADCJ2 starts A/D conversion by a trigger from LPS. When ADCJ2 outputs conversion end interrupt at the end of A/D conversion, LPS detects this interrupt and enters idle state again. If A/D conversion result exceeds the threshold value set in ADCJ2VCULLMTBRy, outputs the ULE flag directly to standby controller and Wake-Up operation can be realized.

For details about LPS operation, refer to **Section 16, Low-Power Sampler (LPS)**.

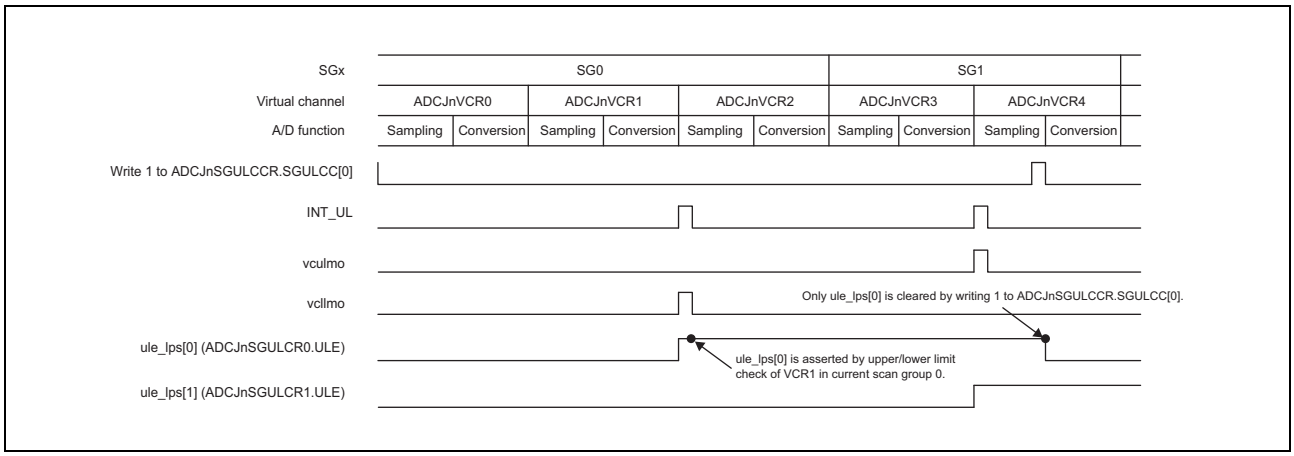


Figure 43.70 ule_lps[4:0] Operation

43.4.21 SVSTOP Function

The SVSTOP is output from the on-chip debugger to operate.

This function masks the scan group start trigger to prevent scan groups from starting during emulation break. The scan group start trigger (including PWM-Diag trigger) that is input during SVSTOP is retained by the hardware until SVSTOP is reset, and retained scan groups are started after SVSTOP is reset. Scan groups for which ADCJnSGSRx.SGACT is set before SVSTOP start are not suspended by SVSTOP. Furthermore, scan groups in continuous scan mode are not suspended even during SVSTOP.

NOTE

During SVSTOP, the A/D timer count is not stopped but the A/D timer trigger is ignored. Therefore new scan group can not started by the A/D timer during SVSTOP.

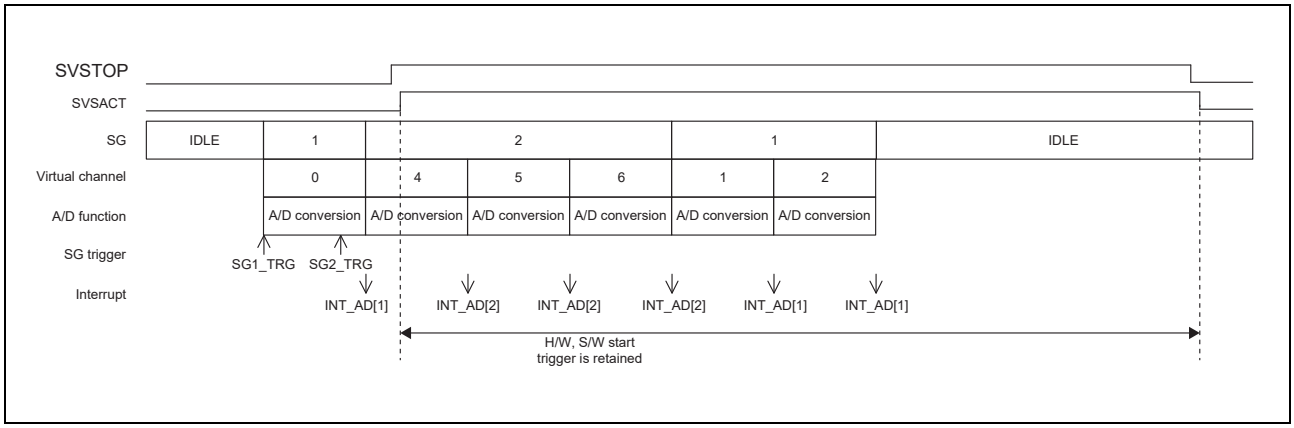


Figure 43.71 SVSTOP Start during A/D Conversion (Synchronous Suspend)

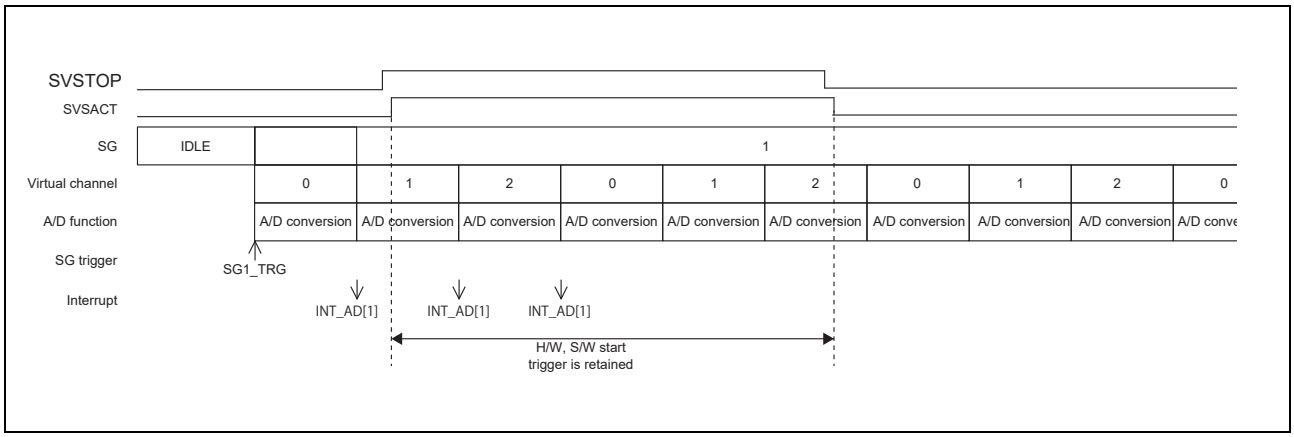


Figure 43.72 SVSTOP Start during A/D Conversion (Continuous Scan Mode)

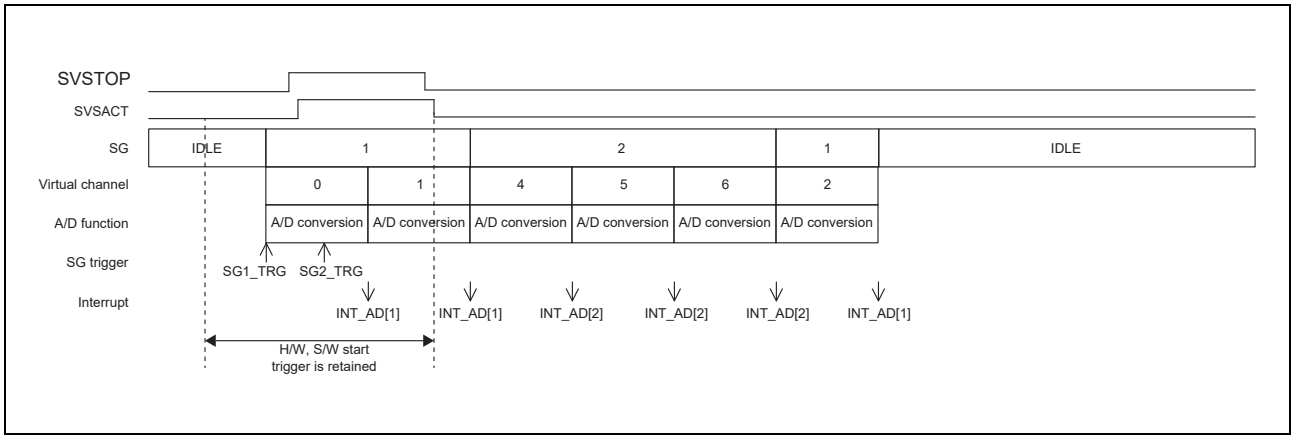


Figure 43.73 SVSTOP Start during A/D Conversion (Synchronous Suspend)

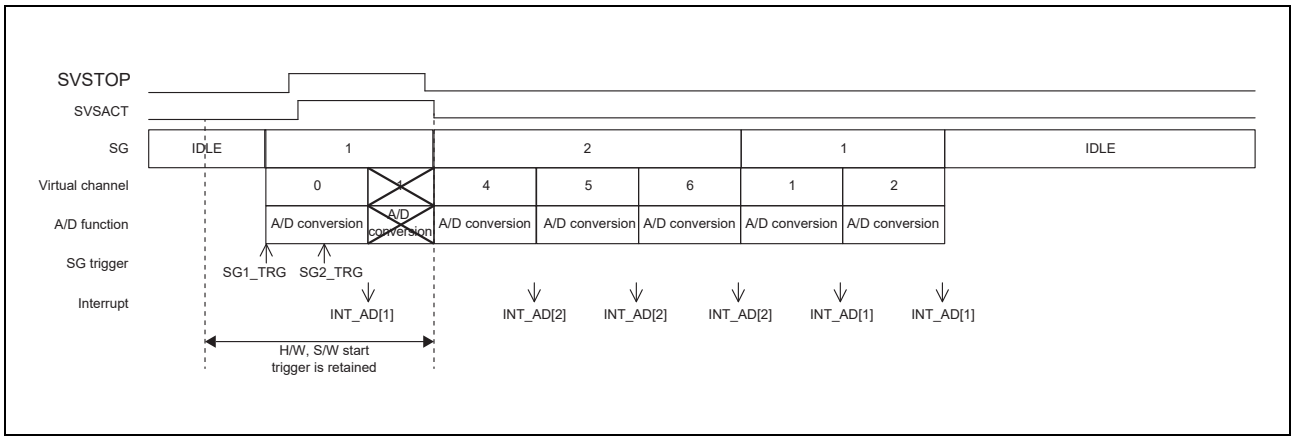


Figure 43.74 Scan Group and SVSTOP Start Alternately (Asynchronous Suspend)

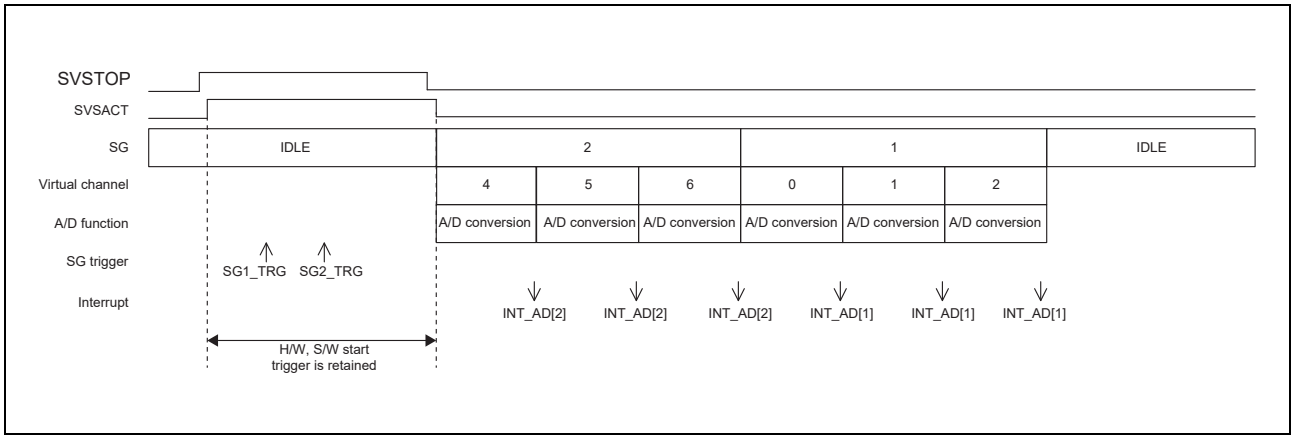


Figure 43.75 Scan Group Start in SVSTOP

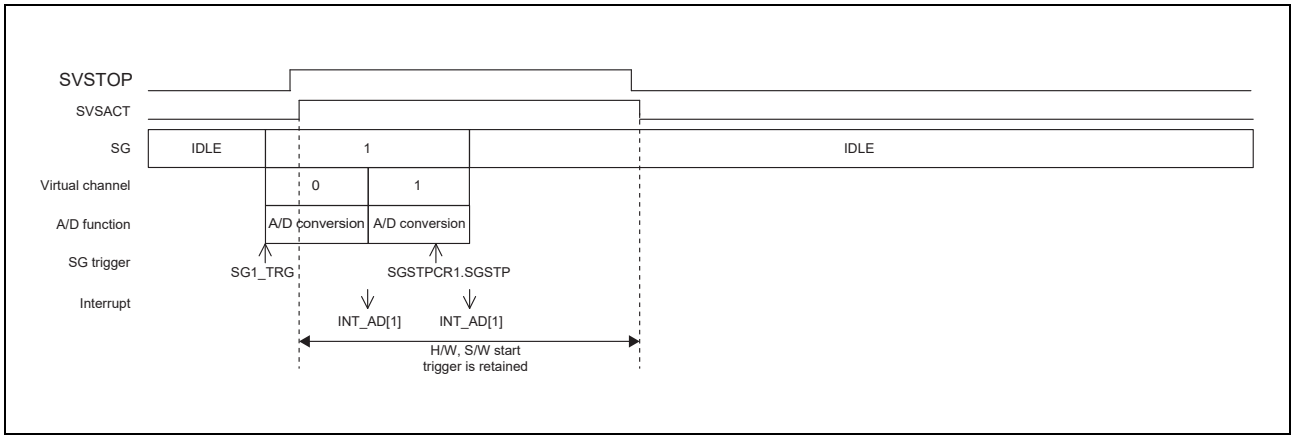


Figure 43.76 Scan Group Stop in SVSTOP (Prohibited Operation)

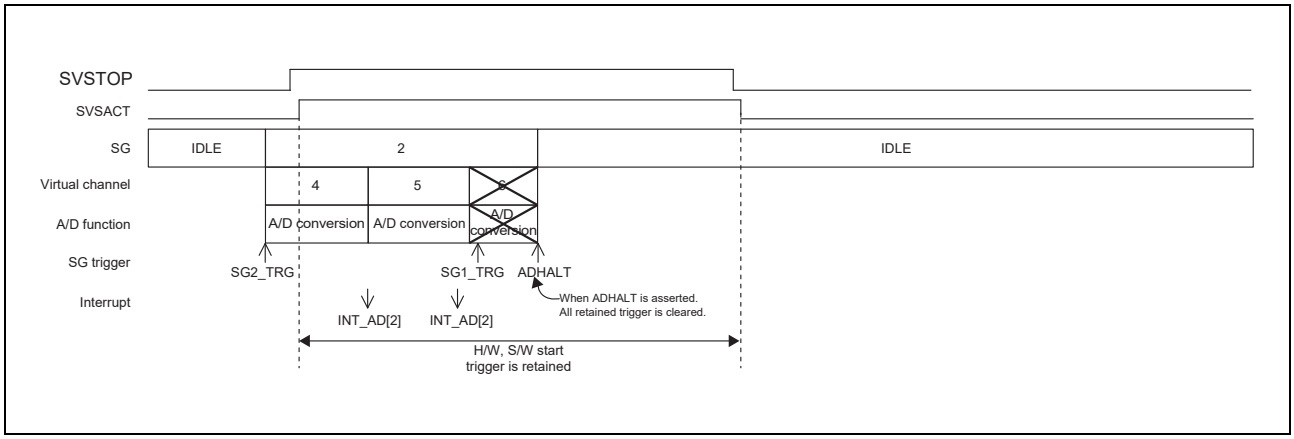


Figure 43.77 AD Forced Termination in SVSTOP (Prohibited Operation)

While SVSTOP operation, the data/flag clear function by reading ADCJnDRj, ADCJnDIRj, ADCJnPWDDR, and ADCJnPWDDIR registers is disabled.

Settings for A/D forced termination (ADCJnADHALTR.HALT = 1) during SVSTOP or stop control (ADCJnSGSTPCRx.SGSTP = 1) for each scan group are prohibited.

Table 43.101 shows differences of ADCJ operation during SVSTOP and in other cases.

Table 43.101 Conditions during SVSTOP Operation

Item	Not during SVSTOP	During SVSTOP
Software trigger/hardware trigger to SG0 to SG4, and PWM-Diag	Enabled	Retained (Disabled if SGACTx has been set)
Software trigger/hardware trigger to T&H group A (Hold trigger of T&H group A)	Enabled	Retained (Disabled if SGACTx has been set)
A/D timer trigger x (x = 3, 4)	Enabled	Disabled (A/D timer trigger is not retained)
SVSTOP status (ADCJnSGSTR:SVSACT)	Cleared	Set
Read and clear of data write flag (ADCJnDIRj.WFLAG, ADCJnPWDDIR.WFLAG)	Enabled	Disabled
Read and clear of data register (ADCJnDRj, ADCJnDIRj, ADCJnPWDDR, and ADCJnPWDDIR)	Enabled	Disabled

43.4.22 Monitoring Function Using the A/D Conversion Monitor Pin

The processing timing of the virtual channel specified by ADCJnADENDPv can be monitored using the ADENDv pin.

The ADENDv pin outputs a high level at the start of A/D conversion of the specified virtual channel, and outputs a low level before the A/D conversion ends.

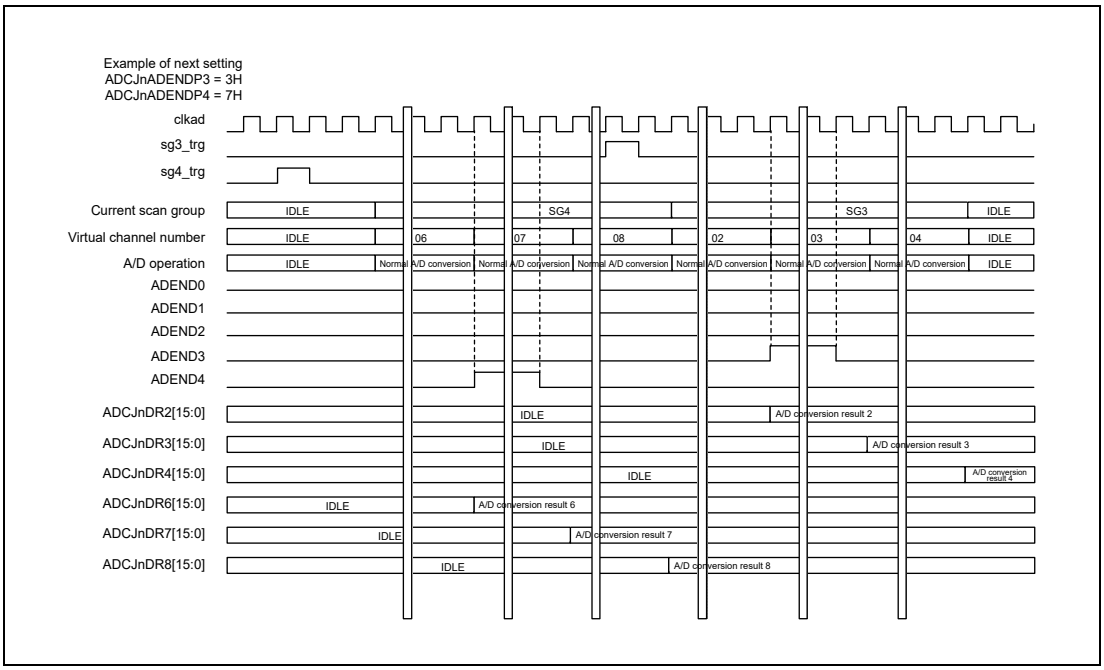


Figure 43.78 A/D Conversion Monitoring Timing

While the ADENDv pin outputs a high level in a low-priority scan group, if operation is suspended (asynchronous suspension) by a high-priority scan group, a low level is output once. When the virtual channel in the suspended low-priority scan group is resumed, the ADENDv pin outputs a high level again.

The ADENDv pin during the asynchronous suspension period due to a hold trigger outputs a low level.

43.4.23 External Module Option Function

43.4.23.1 GTM Entry Function

The ADCJ can output an entry request to the GTM according to the settings of GTMENT in ADCJnVCRj and GTMENTSGxE in ADCJnGTMENTSGER to enter GTM.

At the same time, the ADCJ outputs TAG information set in GTMTAG of ADCJnVCRj and A/D conversion data.

For the format of A/D conversion data to be output, see **Section 43.2.3.4, (3) Data Format for Output to GTM**.

43.4.24 Analog Input Sampling and Scan Group Processing Time

ADCJ has embedded-sample & hold circuit. After SGST bit of ADCJnSGSTCRx is set to 1 and scan group start delay time (tD) has passed, the ADC executes sampling and then starts sequential compare transition processing.

Scan group processing time (tSG) includes scan group start delay time (tD), sampling time (tSPL), sequential compare conversion processing time (tSAR), and scan group end delay time (tED). **Table 43.102, Scan Group Processing Time** shows scan group processing time.

In multicycle scan mode, the scan group processing time (tSG) can be calculated in the formula below when i = the number of virtual channels and j = the number of multicycles.

tSG = tD + (tSPL + tSAR) × i × j + tED

1st cycle scan in continuous scan mode: tD + (tSPL + tSAR) × i

2nd and onward scan in continuous scan mode: (tSPL + tSAR) × i

Table 43.102 Scan Group Processing Time

Item	Symbol	Period	Unit
Scan group start delay time	tD	(2 to 4) × P_Φ + 5 × I_Φ	Refer to *4
Sampling time	tSPL	SMPT × I_Φ *1 or EXSMPT × I_Φ *2	Refer to *4
Sequential compare conversion processing time	tSAR	22 × I_Φ	Refer to *4
Scan group end delay time	tED	(2 to 4) × I_Φ + 3 × P_Φ	Refer to *4
Scan group processing time	tSG	47 × I_Φ + 5 × P_Φ to 49 × I_Φ + 7 × P_Φ *3	Refer to *4

Note 1. Select from SMPT = 18 (default), 99, 204, 252. Set by ADCJnSMPCR register.

Note 2. Select from EXSMPT = 60 (default), 99, 138, 252. Set by ADCJnSMPCR register that valid during CNVCLS[3:0] = 2_H.
(Normal A/D conversion at extended sampling cycle)

Note 3. In case of SMPT = 18 (default) setting.

Note 4. for ADCJ0/1: P_Φ (CLK_LSB), I_Φ (CLK_ADC)
for ADCJ2: P_Φ (CLKA_ADC), I_Φ (CLKA_ADC)

43.5 Notes on Using

43.5.1 Notes on switching clocks

It is prohibited to switch clock frequency of ADCJ during A/D conversion and register access.
So, when clock frequency is switched to use the detect overvoltage function during stand-by mode, user should switch clock frequency after all scan groups halt by setting HALT of ADCJnADHALTR register.

43.5.2 Notes on Using Module Standby Mode

1. When using module standby mode of ADCJ, module standby mode should be set to valid after scan group x stops by using **Section 43.4.3.1, Forced Termination Flow of All Scan Groups** flow because of preventing scan group x activation by unnecessary hardware trigger. Or when the following all status bits are IDLE state, set the module standby mode valid.

Register bit name	Value	Explanation
ADCJnSGSTR.SGACT[4:0]	0_H	All scan groups is IDLE state.
ADCJnSGSTR.ADTACT[4:3]	0_H	All AD timer x is IDLE state.
ADCJnSGSTR.SHACT	0_H	T&H group A is IDLE state.

2. When module standby mode is used setting hardware trigger to valid (ADCJnSGCRx.TRGMD[0]=1), it is prohibited that the hardware trigger is input while ADCJ stop processing starts and module standby mode is valid.

43.5.3 Notes on Using T&H

1. When T&H is effective(ADCJnTHER.THzE=1), the physical channel connected to T&H must be used as Hold value A/D conversion. Exceptionally, the use case of Self-Diagnosis of the T&H Path described in **Section 43.4.10.2, Self-Diagnosis of the T&H Path** is available.
2. It is prohibited to connect MPX to any T&H target physical channel.
3. When T&H is effective(ADCJnTHER.THzE=1), It is prohibited to execute A/D conversion of PWM-Diag to any T&H target physical channel.
4. When using the T&H function, observe the following two T&H analog specifications for products. If a trigger that does not meet the following specifications is asserted, the quality of the ADCJ is not guaranteed.
   a) T&H sampling period from sampling start trigger to hold trigger
   b) T&H hold settling time from hold trigger to hold completion
5. When controlling T&H, use multicycle scan mode (ADCJnSGCRx.SCANMD = 0_H) and select a single scan (ADCJnSGMCYCRx.MCYC[7:0] = 0_H).
6. When using the T&H function, it is prohibited to insert a wait using the WAITTIME[13:0] bits in ADCJnWAITTRy. For wait operation, see **Section 43.4.11.1, Example of Wait Function between Virtual Channels**.

43.5.4 Notes on Using MPX

When using MPX normal A/D conversion or MPX addition mode A/D conversion, be sure to insert a wait.

Also, observe the following notes when using MPX to prevent a system failure.

With the exception, set the number of wait states of the MPX.

Option for [MPX (to I/O port via DMAC)]

Table 43.103 Number of MPX Wait States

MPX Value Transfer Method	Wait State
I/O port	A/D conversion period (default: 1 μsec) or more

Option for [MPX (to MSPI via DMAC)]

Table 43.104 Number of MPX Wait States

MPX Value Transfer Method	Wait State
MSPI(SPI I/F)	A/D conversion period (default: 1 μsec) + MSPI transmission time or more

Option for [MPX (direct output to I/O port)]

Table 43.105 Number of MPX Wait States

MPX Value Transfer Method	Wait State
Output port	A/D conversion period (default: 1 μsec) or more

Exception 1) Set the number of MPX wait states as follows when using MPX in a scan group in any of the following cases.

- In SG0 using the synchronous/asynchronous mixture type suspension method
- In SG0, SG1, SG2, or SG3 using the asynchronous suspension method

Option for [MPX (to I/O port via DMAC)]

Table 43.106 Number of MPX Wait States (Exception 1)

MPX Value Transfer Method	Wait State
I/O port	A/D conversion period (default: 1 μsec) or more

Option for [MPX (to MSPI via DMAC)]

Table 43.107 Number of MPX Wait States (Exception 1)

MPX Value Transfer Method	Wait State
MSPI(SPI I/F)	MSPI transmission time × 2 or more

Option for [MPX (direct output to I/O port)]

Table 43.108 Number of MPX Wait States (Exception 1)

MPX Value Transfer Method	Wait State
Output port	A/D conversion period (default: 1 μsec) or more

Exception 2) Observe the following notes when using MPX in two or more scan groups in any of the following cases.

- Using the synchronous/asynchronous mixture type suspension method
- Using the asynchronous suspension method

Option for [MPX (to I/O port via DMAC)]

Option for [MPX (to MSPI via DMAC)]

Make settings not to use MPX for the first virtual channel of each scan group. (However, there is no problem with the setting to use MPX for the first virtual channel of the lowest-priority scan group among scan groups using MPX.)

Specify one or two scan groups to be transferred by the MSPI (SPI I/F).

Set the number of MPX wait states as follows.

Option for [MPX (to I/O port via DMAC)]

Table 43.109 Number of MPX Wait States (Exception 2)

MPX Value Transfer Method	Wait State
I/O port	A/D conversion period (default: 1 μsec) or more

Option for [MPX (to MSPI via DMAC)]

Table 43.110 Number of MPX Wait States (Exception 2)

MPX Value Transfer Method	Wait State
MSPI(SPI I/F)	MSPI transmission time × 2 or more

Option for [MPX (direct output to I/O port)]

Table 43.111 Number of MPX Wait States (Exception 2)

MPX Value Transfer Method	Wait State
Output port	A/D conversion period (default: 1 μsec) or more

43.5.5 Notes of Upper/Lower Limit Check Function and Status Register

Upper/Lower Limit Check is the function that compare A/D conversion result with the setting value of Upper/Lower Limit Check Table register (ADCJnVCULLMTBRy) after A/D conversion of each virtual channel is end. Status register of compare result has 2 type that are “1. each virtual channel” and “2.each scan group”.

Virtual Channel Upper/Lower Limit Check Status has 2 type clear that are “1.individual clear” and “2.all clear”. When all clear(ADCJnVCLMASCR.ALLMSC) is used during ADCJ operation, the status of each virtual channel that is being operated is cleared. So it is recommended to use individual clear when ADCJ is operating.

Scan Group Upper/Lower Limit Check Status has individual clear. All clear can be used because individual clear of all scan groups is set one register.

Table 43.112 The relation between Upper/Lower Limit Check and Clear Register

Function	Category	Status (bit or register)	Clear register name	
			Cleared individually	Cleared at once
Upper/Lower Limit Check	Each Virtual Channel	ADCJnVCLMSR1.VC00LMS to VC31LMS	ADCJnVCLMSCR1.VC00LMSC to VC31LMSC	ADCJnVCLMASCR .ALLMSC
		ADCJnVCLMSR2.VC32LMS to VC63LMS	ADCJnVCLMSCR2.VC32LMSC to VC63LMSC	
		ADCJnPWVCLMSR.PWVCLMS	ADCJnPWVCLMSCR.PWVCLMSC	
	Each Scan Group	ADCJnSGULCR0	ADCJnSGULCCR.SGULCC[0]	—
		ADCJnSGULCR1	ADCJnSGULCCR.SGULCC[1]	
		ADCJnSGULCR2	ADCJnSGULCCR.SGULCC[2]	
		ADCJnSGULCR3	ADCJnSGULCCR.SGULCC[3]	
		ADCJnSGULCR4	ADCJnSGULCCR.SGULCC[4]	

43.6 Definitions with Respect to A/D Conversion Accuracy

The following describes A/D conversion accuracy. Definitions (3) to (7) are illustrated.

(1) Resolution
Number of digital output codes of A/D converter

(2) Quantization error
An error that A/D converter essentially has, which is given as 1/2LSB

(3) Offset error
Deviation from ideal A/D conversion characteristic of analog input voltage value when digital output changes from the minimum voltage value 000_H to 001_H, excluding quantization error

(4) Full-scale error
Deviation from ideal A/D conversion characteristic of analog input voltage value when digital output changes from FFEH to FFFH, excluding quantization error

(5) DNL (Differential non-linearity error)
Deviation between ideal digital output code width (Vq) and actual digital output code width (Va), which is given as (Va - Vq)/Vq, excluding offset error, full-scale error, and quantization error

(6) INL (Integral non-linearity error)
Deviation from ideal A/D conversion characteristic from zero voltage to full-scale voltage, which is given as integral of DNL from 000_H to an arbitrary digital output code, excluding offset error, full-scale error, and quantization error

(7) Total error
Deviation between digital value and analog input value, including offset error, full-scale error, quantization error, DNL, and INL

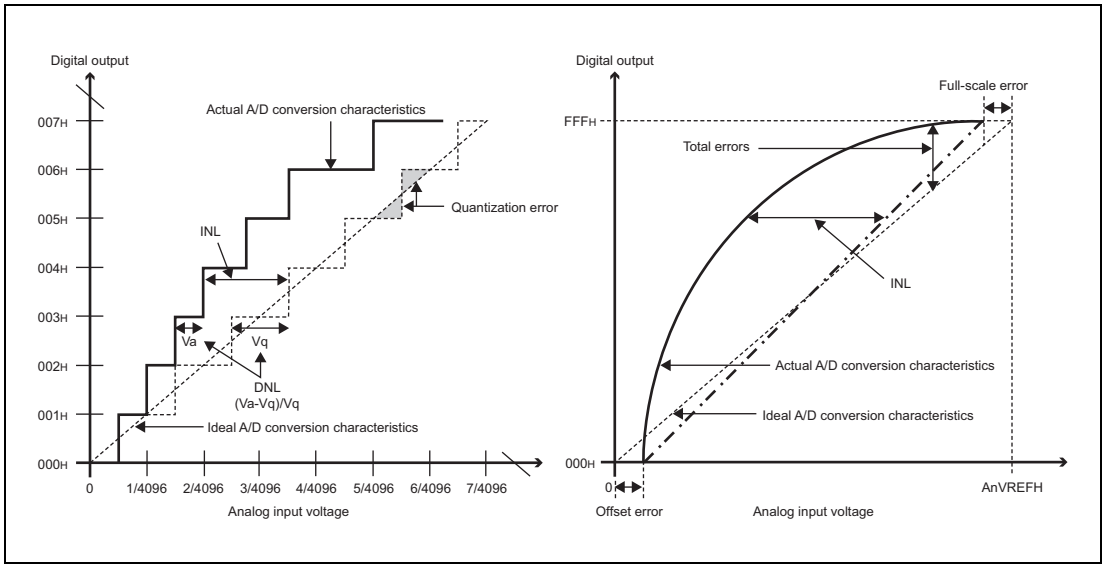


Figure 43.79 Definition of A/D Conversion Accuracy

43.7 ADC VMON Secondary Error Generator (AVSEG)

43.7.1 Overview

43.7.1.1 Function Overview

AVSEG is equivalent to the function which notify an upper error pulse (secondary HDET) and a lower error pulse (Secondary LDET) of each power supply (VCC, E0VCC, ISOVDD and AWOVDD) to INTC in the secondary power supply voltage monitor (Upper error pulse & Lower error pulse are merged together then routing to INTC).

- Upper Error Pulse Control and Lower Error Pulse Control
  This function generates an error signal from ADCJ0 to INTC if an upper or lower bound is exceeded.
- Noise Filter
  The upper and lower error pulses are generated by signals passed through a filter to reduce noise.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

43.7.1.2 Block Diagram

Figure 43.80 illustrates the AVSEG block diagram.

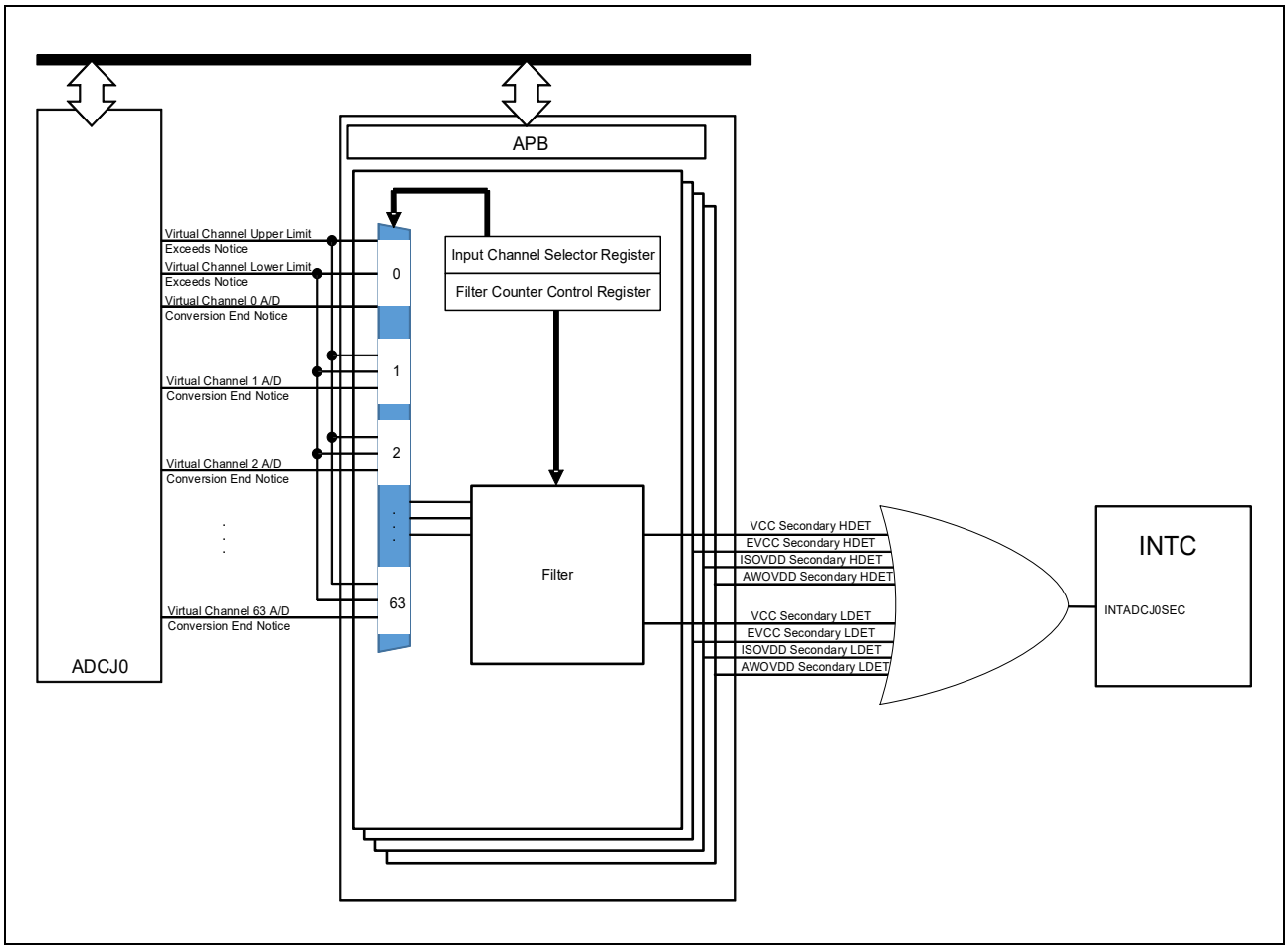


Figure 43.80 AVSEG Block Diagram

43.7.2 Registers

43.7.3 List of Registers

Table 43.113 List of Registers

Module Name	Register Name	Symbol	Address	Access	Access Protection	
					PBG	Other
AVSEG	Input Channel Selector Register VCC	AVSEGVCCCHSCR	<AVSEG_base> + 00_H	32	PBG6L0#4	—
AVSEG	Filter Counter Control Register VCC	AVSEGVCCCNTCR	<AVSEG_base> + 04_H	32	PBG6L0#4	—
AVSEG	Input Channel Selector Register EVCC	AVSEGEVCCCHSCR	<AVSEG_base> + 10_H	32	PBG6L0#4	—
AVSEG	Filter Counter Control Register EVCC	AVSEGEVCCCNTCR	<AVSEG_base> + 14_H	32	PBG6L0#4	—
AVSEG	Input Channel Selector Register AWOVDD	AVSEGAWOVDDCHSCR	<AVSEG_base> + 20_H	32	PBG6L0#4	—
AVSEG	Filter Counter Control Register AWOVDD	AVSEGAWOVDDCNTCR	<AVSEG_base> + 24_H	32	PBG6L0#4	—
AVSEG	Input Channel Selector Register ISOVDD	AVSEGISOVDDCHSCR	<AVSEG_base> + 30_H	32	PBG6L0#4	—
AVSEG	Filter Counter Control Register ISOVDD	AVSEGISOVDDCNTCR	<AVSEG_base> + 34_H	32	PBG6L0#4	—
AVSEG	Secondary voltage monitor error register	AVSEGSECVMONERR	<AVSEG_base> + 40_H	32	PBG6L0#4	—
AVSEG	Secondary voltage monitor error clear register	AVSEGSECVMONCLR	<AVSEG_base> + 44_H	32	PBG6L0#4	—

43.7.3.1 AVSEG Specific Registers

(1) AVSEGVCCCHSCR — Input Channel Selector Register VCC

This register selects the input channel.

Access: This register can be read or written in 32-bit units.

Address: <AVSEG_base> + 00_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	AVSEGVCCCHS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.114 AVSEGVCCCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	AVSEGVCCCHS[5:0]	Input Channel Selector These bits select the channel used for generating error of VCC in the secondary power supply voltage monitor. 00_H: Selects ADCJ0 virtual channel 0 : $3F_H$: Selects ADCJ0 virtual channel 63

(2) AVSEGVCCCNTCR — Filter Counter Control Register VCC

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Access: This register can be read or written in 32-bit units.

Address: <AVSEG_base> + 04_H

Value after reset: $0000\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEGVCCENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGVCCNRMCNT				—	—	—	—	AVSEGVCCERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.115 AVSEGVCCCNTCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	AVSEGVCCENB	Filter Enable This bit enable AVSEG for VCC 0: Disabled 1: Enables the filter. When set to disable all AVSEG for VCC functions are disabled.
27 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	AVSEGVCCNRMCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGVCCNRMCNT. 1_H: Recovery after the signal is within the boundaries for 1 count. : F_H: Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	AVSEGVCCERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to AVSEGVCCERRCNT. 1_H: Out of bounds when the signal violates the boundaries for 1 count. : F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting AVSEGVCCNRMCNT and AVSEGVCCERRCNT to 0_H is prohibited.
If AVSEGVCCNRMCNT is set to 0_H the counter will act as if it was set to 1_H.
The filter will not output any signals if AVSEGVCCERRCNT is set to 0_H.

(3) AVSEGEVCCCHSCR — Input Channel Selector Register EVCC

This register selects the input channel.

Access: This register can be read or written in 32-bit units.
Address: <AVSEG_base> + 10_H
Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	AVSEGEVCCCHS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.116 AVSEGEVCCCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	AVSEGEVCCCHS[5:0]	Input Channel Selector These bits select the channel used for generating error of E0VCC in the secondary power supply voltage monitor 00_H: Selects ADCJ0 virtual channel 0 : $3F_H$: Selects ADCJ0 virtual channel 63

(4) AVSEGEVCCCNTCR — Filter Counter Control Register EVCC

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Access: This register can be read or written in 32-bit units.

Address: <AVSEG_base> + 14_H

Value after reset: 0000 0101_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEGEVCCENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGEVCCNRMCNT				—	—	—	—	AVSEGEVCCERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.117 AVSEGEVCCCNTCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	AVSEGEVCCENB	Filter Enable This bit enable AVSEG for E0VCC 0: Disabled 1: Enables the filter. When set to disable all AVSEG for E0VCC functions are disabled.
27 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	AVSEGEVCCNRMCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGEVCCNRMCNT. 1_H: Recovery after the signal is within the boundaries for 1 count. : F_H: Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	AVSEGEVCCERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to AVSEGEVCCERRCNT. 1_H: Out of bounds when the signal violates the boundaries for 1 count. : F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting AVSEGEVCCNRMCNT and AVSEGEVCCERRCNT to 0H is prohibited.
If AVSEGEVCCNRMCNT is set to 0_H the counter will act as if it was set to 1_H.
The filter will not output any signals if AVSEGEVCCERRCNT is set to 0_H.

(5) AVSEGAWOVDDCHSCR — Input Channel Selector Register AWOVDD

This register selects the input channel.

Access: This register can be read or written in 32-bit units.
Address: <AVSEG_base> + 20_H
Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	AVSEGAWOVDDCHS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.118 AVSEGAWOVDDCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	AVSEGAWOVDDCHS[5:0]	Input Channel Selector These bits select the channel used for generating error of AWOVDD in the secondary power supply voltage monitor. 00_H: Selects ADCJ0 virtual channel 0 : $3F_H$: Selects ADCJ0 virtual channel 63

(6) AVSEGAWOVDDCNTCR — Filter Counter Control Register AWOVDD

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Access: This register can be read or written in 32-bit units.

Address: <AVSEG_base> + 24_H

Value after reset: $0000\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEGAWOVDDENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGAWOVDDNRMCNT				—	—	—	—	AVSEGAWOVDDERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.119 AVSEGAWOVDDCNTCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	AVSEGAWOVDDENB	Filter Enable This bit enable AVSEG for ISOVDD 0: Disabled 1: Enables the filter. When set to disable all AVSEG for AWOVDD functions are disabled.
27 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	AVSEGAWOVDDNRMCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGAWOVDDNRMCNT. 1_H: Recovery after the signal is within the boundaries for 1 count. : F_H: Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	AVSEGAWOVDDERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to AVSEGAWOVDDERRCNT. 1_H: Out of bounds when the signal violates the boundaries for 1 count. : F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting AVSEGAWOVDDNRMCNT and AVSEGAWOVDDERRCNT to 0H is prohibited.
If AVSEGAWOVDDNRMCNT is set to 0_H the counter will act as if it was set to 1_H.
The filter will not output any signals if AVSEGAWOVDDERRCNT is set to 0_H.

(7) AVSEGISOVDDCHSCR — Input Channel Selector Register ISOVDD

This register selects the input channel.

Access: This register can be read or written in 32-bit units.
Address: <AVSEG_base> + 30_H
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	AVSEGISOVDDCHS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.120 AVSEGISOVDDCHSCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	AVSEGISOVDDCHS[5:0]	Input Channel Selector These bits select the channel used for generating error of ISOVDD in the secondary power supply voltage monitor 00_H: Selects ADCJ0 virtual channel 0 : $3F_H$: Selects ADCJ0 virtual channel 63

(8) AVSEGISOVDDCNTCR — Filter Counter Control Register ISOVDD

This register controls the filter counter. The noise filtering interval is controlled by using an error counter and a recovery counter. The internal down counter is reset with the written values when this register is modified.

Access: This register can be read or written in 32-bit units.

Address: <AVSEG_base> + 34_H

Value after reset: $0000\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	AVSEGISOVDDENB	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AVSEGISOVDDNRMCNT				—	—	—	—	AVSEGISOVDDERRCNT			
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.121 AVSEGISOVDDCNTCR Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	AVSEGISOVDDENB	Filter Enable This bit enable AVSEG for ISOVDD 0: DISABLED 1: Enables the filter. When set to DISABLED all AVSEG for ISOVDD functions are disabled.
27 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 8	AVSEGISOVDDNRMCNT	Recovery Counter Settings These bits control the number of counts until the signal is considered recovered. Once the signal is considered out of bounds, it will not be recovered until the set number of consecutive non-error pulses are detected. The counter value is reset when one of the following values is written to AVSEGISOVDDNRMCNT. 1_H: Recovery after the signal is within the boundaries for 1 count. : F_H: Recovery after the signal is within the boundaries for 15 consecutive counts.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	AVSEGISOVDDERRCNT	Error Counter Control These bits control the number of counts until the signal is considered out of bounds. The signal is considered out of bounds once the set number of consecutive error pulses are detected. The counter value is reset when one of the following values is written to AVSEGISOVDDERRCNT. 1_H: Out of bounds when the signal violates the boundaries for 1 count. : F_H: Out of bounds when the signal violates the boundaries for 15 consecutive counts.

CAUTION

Setting AVSEGISOVDDNRMCNT and AVSEGISOVDDERRCNT to 0H is prohibited.
If AVSEGISOVDDNRMCNT is set to 0_H the counter will act as if it was set to 1_H.
The filter will not output any signals if AVSEGISOVDDERRCNT is set to 0_H.

(9) AVSEGSECVMONERR – Secondary voltage monitor error register

AVSEGSECVMONERR is a readable register that indicates Secondary voltage monitor error of VCC, E0VCC, ISOVDD and AWOVDD. AVSEGSECVMONERR is initialized to 0000 0000_H at reset.

Access: This register is a read-only register that can be read in 32-bit units.
Address: <AVSEG_base> + 40_H
Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	AWOVDD_ULME	ISOVDD_ULME	EVCC_ULME	VCC_ULME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AWOVDD_LLME	ISOVDD_LLME	EVCC_LLME	VCC_LLME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 43.122 AVSEGSECVMONERR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19	AWOVDD_ULME	Upper limit voltage error AWOVDD 0: No error present 1: An error is present Setting condition Upper limit voltage error of AWOVDD is detected Clearing condition A value of 1 is written to AWOVDD_ULMEC in AVSEGSECVMONCLR
18	ISOVDD_ULME	Upper limit voltage error ISOVDD 0: No error present 1: An error is present Setting condition Upper limit voltage error of ISOVDD is detected Clearing condition A value of 1 is written to ISOVDD_ULMEC in AVSEGSECVMONCLR
17	EVCC_ULME	Upper limit voltage error E0VCC 0: No error present 1: An error is present Setting condition Upper limit voltage error of E0VCC is detected Clearing condition A value of 1 is written to EVCC_ULMEC in AVSEGSECVMONCLR

Table 43.122 AVSEGSECVMONERR Register Contents (2/2)

Bit Position	Bit Name	Function
16	VCC_ULME	Upper limit voltage error VCC 0: No error present 1: An error is present. Setting condition Upper limit voltage error of VCC is detected. Clearing condition A value of 1 is written to VCC_ULMEC in AVSEGSECVMONCLR
15 to 4	Reserved	When read, the value after reset is returned.
3	AWOVDD_LLME	Lower limit voltage error AWOVDD 0: No error present 1: An error is present. Setting condition Lower limit voltage error of AWOVDD is detected. Clearing condition A value of 1 is written to AWOVDD_LLMEC in AVSEGSECVMONCLR
2	ISOVDD_LLME	Lower limit voltage error ISOVDD 0: No error present 1: An error is present. Setting condition Lower limit voltage error of ISOVDD is detected. Clearing condition A value of 1 is written to ISOVDD_LLMEC in AVSEGSECVMONCLR
1	EVCC_LLME	Lower limit voltage error E0VCC 0: No error present 1: An error is present. Setting condition Lower limit voltage error of E0VCC is detected. Clearing condition A value of 1 is written to EVCC_LLMEC in AVSEGSECVMONCLR
0	VCC_LLME	Lower limit voltage error VCC 0: No error present 1: An error is present. Setting condition Lower limit voltage error of VCC is detected. Clearing condition A value of 1 is written to VCC_LLMEC in AVSEGSECVMONCLR

(10) AVSEGSECVMONCLR – Secondary voltage monitor error clear register

AVSEGSECVMONCLR is an 32-bit write-only register that control error clearing in AVSEGSECVMONERR register. AVSEGSECVMONCLR is always read as 0.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <AVSEG_base> + 44_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	AWOVDD_ULMEC	ISOVDD_ULMEC	EVCC_ULMEC	VCC_ULMEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AWOVDD_LLMEC	ISOVDD_LLMEC	EVCC_LLMEC	VCC_LLMEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 43.123 AVSEGSECVMONCLR Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When writing, write the value after reset.
19	AWOVDD_ULMEC	Upper limit voltage error AWOVDD clear Writing 0: Not clear Writing 1: Clear
18	ISOVDD_ULMEC	Upper limit voltage error ISOVDD clear Writing 0: Not clear Writing 1: Clear
17	EVCC_ULMEC	Upper limit voltage error E0VCC clear Writing 0: Not clear Writing 1: Clear
16	VCC_ULMEC	Upper limit voltage error VCC clear Writing 0: Not clear Writing 1: Clear
15 to 4	Reserved	When writing, write the value after reset.
3	AWOVDD_LLMEC	Lower limit voltage error AWOVDD clear Writing 0: Not clear Writing 1: Clear
2	ISOVDD_LLMEC	Lower limit voltage error ISOVDD clear Writing 0: Not clear Writing 1: Clear
1	EVCC_LLMEC	Lower limit voltage error E0VCC clear Writing 0: Not clear Writing 1: Clear
0	VCC_LLMEC	Lower limit voltage error VCC clear Writing 0: Not clear Writing 1: Clear

43.7.4 Operation

43.7.4.1 Noise Count Method

The AVSEG noise count methods are shown below. Noise is filtered and reduced by counting the number of times the ADC boundary values are exceeded using the input error pulses.

CAUTION

The input of both upper and lower pulses into the same module at the same time is prohibited.

Example for the upper error pulse AVSEG for VCC:

Method 1:

AVSEGVCCCNTCR.AVSEGVCCERMCNT = 1_H to 3_H and
AVSEGVCCCNTCR.AVSEGVCCNRMCNT = 1_H.

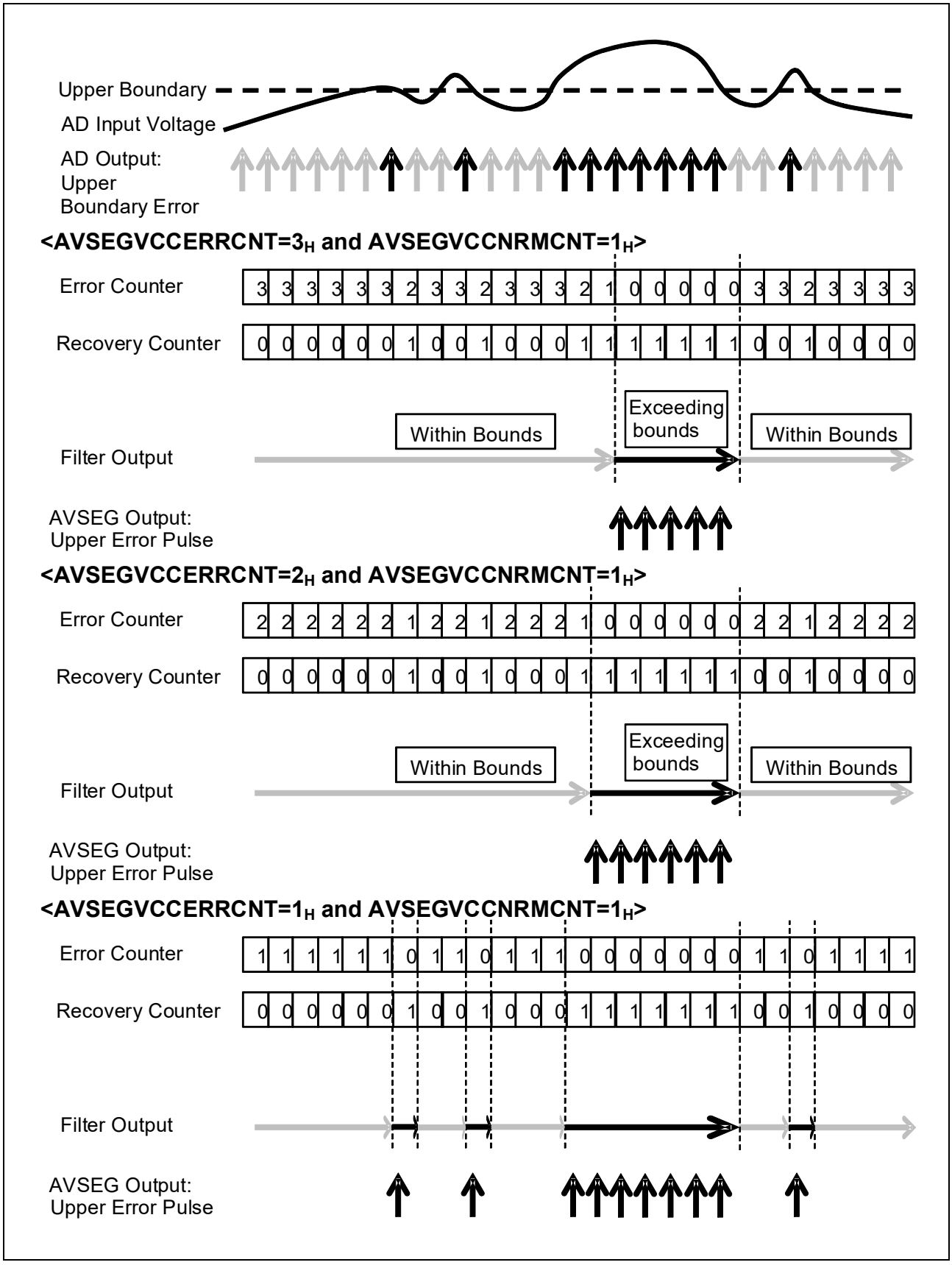


Figure 43.81 Noise Count (1)

Method 2:

AVSEGVCCCNTCR.AVSEGVCCERMCNT = 1_H to 3_H and
AVSEGVCCCNTCR.AVSEGVCCNRMCNT = 2_H.

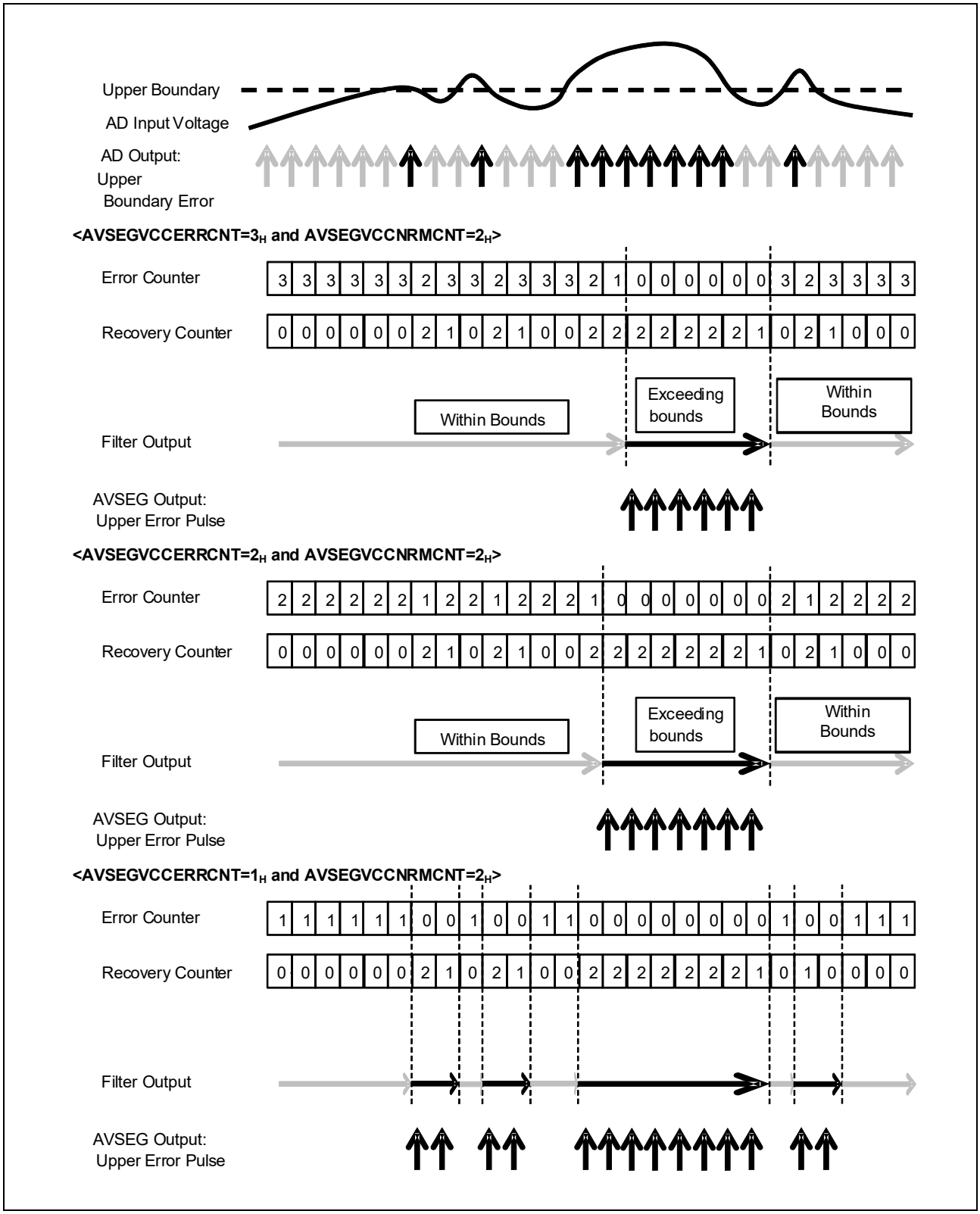


Figure 43.82 Noise Count (2)

Method 3:

AVSEGVCCCNTCR.AVSEGVCCERRCNT = 1_H to 3_H and
AVSEGVCCCNTCR.AVSEGVCCNRMCNT = 3_H.

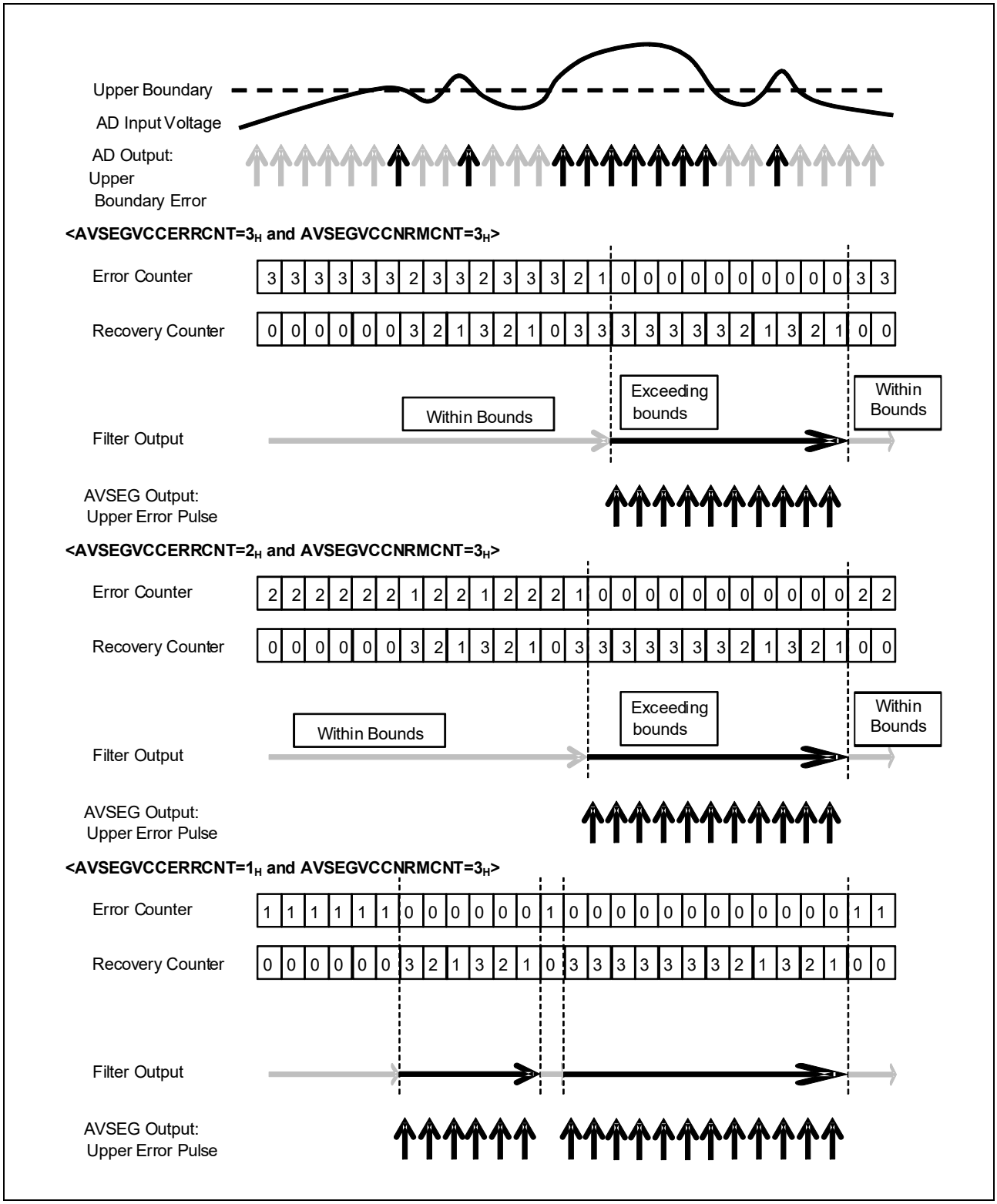


Figure 43.83 Noise Count (3)

43.7.4.2 Upper/Lower Error Pulse Output

The AVSEG upper/lower error pulse output signals are shown below. All output signals are generated by signals that have passed through the filter function to reduce noise.

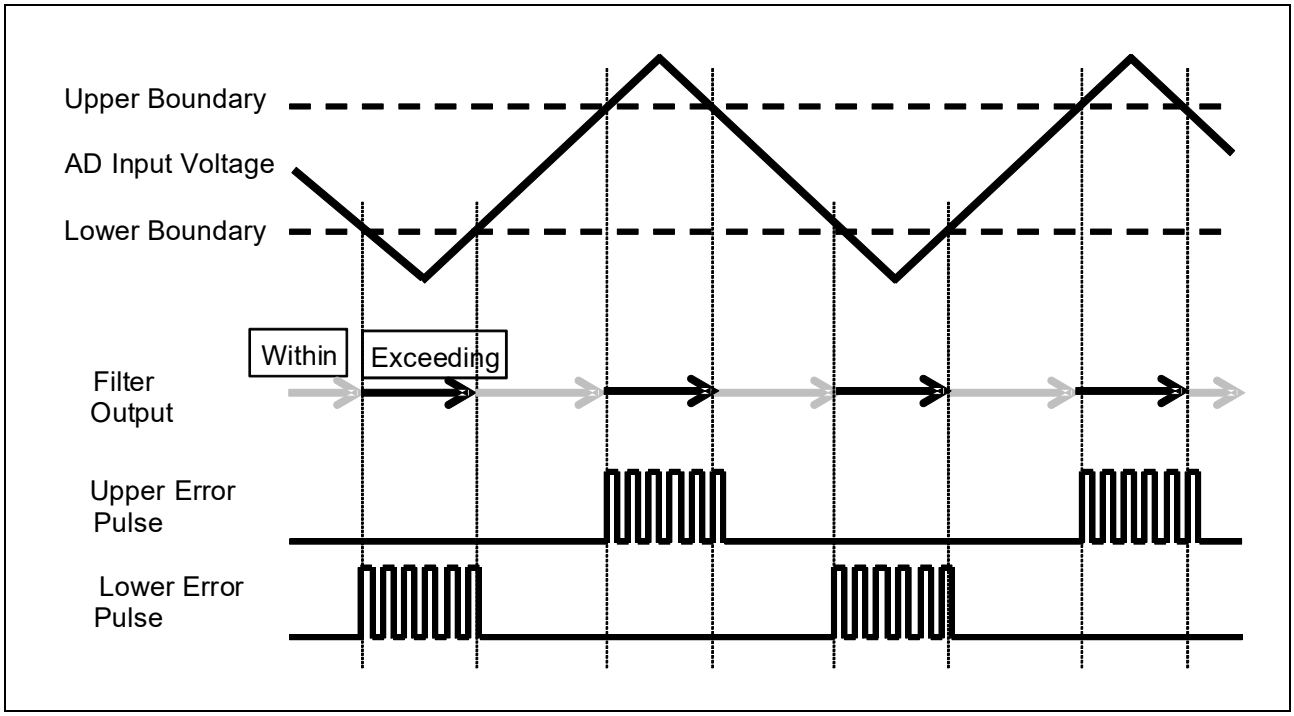


Figure 43.84 Upper/Lower Error Pulse Output

43.7.5 Use Method

(1) Startup Procedure Example

(a) ADC Settings (For details, refer to **Section 43, Analog to Digital Converter (ADCJ)**

Set the virtual channel registers and virtual channel upper/lower limit table registers so that comparisons are made between the limit registers and the input voltage during conversion.

(b) Interrupt setting

Refer to **Section 6, Interrupts**.

(c) Clear error status in AVSEGSECVMONERR by writing to AVSEGSECVMONCLR register.

(d) AVSEG Settings

Set the Input Channel Selector Register of respective voltage (AVSEGVCCCHSCR, AVSEGEVCCCHSCR, AVSEGISOVDDCHSCR and AVSEGAWOVDDCHSCR).

Set the Filter Counter Control Register of respective voltage (AVSEGVCCCNTCR, AVSEGEVCCCNTCR, AVSEGISOVDDCNTCR and AVSEGAWOVDDCNTCR).

(e) Begin A/D Conversion

Begin A/D conversion after setting up AVSEG. If the AVSEG Filter Counter Control Register is modified, the error counter and recovery counter will be set with the written values, and all outputs change will be masked.

(2) Stop Procedure Example

(a) End A/D Conversion

Stop the ADCJ0.

(b) AVSEG Setting

Set the filter counter enable bit, AVSEGVCCCNTCR. AVSEGVCCENB, AVSEGEVCCCNTCR. AVSEGEVCCENB, AVSEGISOVDDCNTCR. AVSEGISOVDDENB and AVSEGAWOVDDCNTCR. AVSEGAWOVDDENB to 0.

This will prevent any and all pulse signals from being output from AVSEG.

NOTE

The internal filter counters will be reset to the written values during the AVSEGVCCCNTCR, AVSEGEVCCCNTCR, AVSEGISOVDDCNTCR and AVSEGAWOVDDCNTCR write procedure.

(3) Usage Notes

(a) The input of both upper and lower error pulses into the same module at the same time is prohibited.

(b) Setting AVSEGVCCNRMCNT, AVSEGVCCERRCNT, AVSEGEVCCNRMCNT, AVSEGEVCCERRCNT, AVSEGISOVDDNRMCNT, AVSEGISOVDDERRCNT, AVSEGAWOVDDNRMCNT and AVSEGAWOVDDERRCNT to 0_H is prohibited.

Section 44 Functional Safety

44.1 Overview

This section describes the intended safety mechanisms provided to detect the MCU failures with a short detection time. Here, the failures include both the recoverable transient failures such as soft errors of a memory and the unrecoverable permanent failures.

This MCU was developed to meet the requirements for a Safety Element out of Context (SEooC) as described in the ISO26262. In a SEooC development, safety requirements of target device (component) are assumed from market requirements by component manufacturers, and then the device is developed according to these assumed requirements. As this application is targeted for chassis & safety and body applications, some special considerations are already included to minimize gaps with chassis & safety and body requirements, however this should not restrict use with other applications. If Renesas assumptions are not enough to achieve system safety requirements derived from safety goals, the system safety requirements need to be modified. Contact our sale office for details regarding the development process and safety organization.

The following lists the intended safety mechanisms provided by this product.

Note 1. The suitability of the safety mechanisms referenced in this chapter will finally be judged during the safety analysis and safety assessment of the MCU. Therefore, the safety documentation shall always be considered as additional requirements for the system or software derived from the results of the safety analysis.

ECC and EDC

Detect failures of memory and data transfer paths; correct failures.

Lockstep

Detects failures of the CPU and DTS with a short detection time.

Memory Protection

Detects erroneous access to the memory and peripheral circuits to protect the data in these elements against erroneous access.

BIST

Detects failures of the failure detection function itself.

Error Control Module (ECM)

Monitors various failure detection states in the MCU and defines the operation to be carried out when a failure occurs reaction.

Voltage Monitor

Monitors power supply voltage to detect over and under voltage.

Clock Monitor

Monitors the clock operation to detect abnormal operation.

Data CRC

Generates CRC to verify the data streams protected by CRC.

44.2 Reset Sources

Register reset conditions (defined in this section) are shown in **Table 44.1**.

Table 44.1 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ECC and EDC	All registers	√	√	√	√	√	√*1	—
Lockstep (CPU)	See **Section 3, CPU System**.							
Lockstep (DTS)	DTS_COMP_CNTRL	√	√	√	√	√	—	—
Memory Protection	All registers	√	√	√	√	√	—	—
BIST	See **Section 44.6.2.3, Reset of Registers**.							
Error Control Module (ECM)	See **Section 45, Error Control Module (ECM)**.							
Voltage Monitor	See **Section 11, Power Supply Voltage Monitor**.							
Clock Monitor	See **Section 14, Clock Monitor (CLMA)**.							
Data CRC	See **Section 46, Data CRC Function K (KCRC)**.							

Note 1. ECC for peripheral RAM is initialized by Module Reset. For the target modules of Module Reset, see **Section 9, Reset Controller**. For details of ECC for peripheral RAM, see **Section 44.3.10, ECC for Peripheral RAM**.

44.3 ECC and EDC

44.3.1 Overview

44.3.1.1 ECC

This product incorporates ECC for the following memories. The ECC enables detection and correction of errors of the data retained in memory. The ECC also enables detection and correction of errors produced between the ECC encoder and memory; and memory and ECC decoder.

Table 44.2 ECC Overview

Applicable Memory	Applicable Data Width [bits]	Operation upon Error Detection				Failure Insertion
		Detection/ Correction	Notice to ECM	Error Status	Address Capture	
Code flash	256	SEC-DED	Possible*1	Possible*1	Possible*1	Possible
Data flash	32	SEC-DED	Possible	Possible	Possible	Possible
Local RAM (all CPU cores)	32	SEC-DED	Possible	Possible	Possible	Possible
Cluster RAM	32	SEC-DED	Possible	Possible	Possible	Possible
Instruction cache (data)	64	SED-DED	Possible	Possible	Possible	Possible
Instruction cache (tag)	32	SED-DED	Possible	Possible	Possible	Possible
RAM for DTS	32	SEC-DED	Possible	Possible	Possible	Possible
RAM for sDMAC (Descriptor)	32	SEC-DED	Possible	Possible	Possible	Possible
RAM for sDMAC (Data Transfer)	64	SEC-DED	Possible	Possible	Possible	Possible
Peripheral RAM (32 bits)	32	SEC-DED	Possible	Possible	Possible	Possible
Data transfer path (data)	32/128	SEC-DED	Possible	Possible	N/A	Possible
Data transfer path (address)	32	SED-DED	Possible	Possible	N/A	Possible

Note 1. When an error except ECC 1-bit error correction occurs in instruction fetch at the time of cache miss or cache invalid, ECM error sources may not be notified and error status / error address registers may not be updated.
For details, see **Section 3, CPU System**.

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors*1 can only be detected.

SED-DED: 1-bit errors and 2-bit errors*1 can only be detected.

Note 1. All 0s and 1s of data bits and ECC bits are not a legal combination and detected as 2-bit errors.

Notice to ECM

A detected error can be reported to the ECM.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

An ECC error can be intentionally generated to enable software test of the ECC decoder operation.

44.3.1.2 Address Parity

This product incorporates address parity for the following types of memory. The address parity enables detection of errors during address decoding. It also enables detection of errors produced at addresses between the parity encoder and memory.

Table 44.3 Address Parity Overview

Applicable Memory	Parity Bit	Notice to ECM	Error Status	Address Capture	Failure Insertion
Code flash	1 bit	Possible*1	Possible*1	Possible*1	Possible

Note 1. When an error occurs in instruction fetch at the time of cache miss or cache invalid, ECM error sources may not be notified and error status / error address registers may not be updated.
For details, see **Section 3, CPU System**.

44.3.1.3 Address Feedback

This product incorporates an address feedback function for the following types of memory. The address feedback function enables detection of address errors inside memory module itself.

Table 44.4 Address Feedback Overview

Applicable Memory	Notice to ECM	Error Status	Address Capture	Failure Insertion
Local RAM (all CPU cores)	Possible	Possible	Possible	Possible
Cluster RAM	Possible	Possible	Possible	Possible
Instruction cache (data/tag)	Possible	Possible	Possible	Possible
RAM for DTS	Possible	Possible	Possible	Possible

44.3.2 Error Notifications

44.3.2.1 Error Notifications for Code Flash ECC and Address Parity

The error notifications of Code Flash ECC and Address Parity are listed in the following table.

Table 44.5 Error Notifications for Code Flash

Error Notification	Description	ECM Error Number	Error Response to bus master
Code Flash - Address parity error	Code Flash address parity error	80	√
Code Flash - Data ECC 2-bit error	Code Flash data ECC 2-bit error	81	√
Code Flash - Data ECC 1-bit error	Code Flash data ECC 1-bit error	82	√/— *1
Code Flash - ECC overflow error	Code Flash ECC error address buffer overflow error	83	—

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

44.3.2.2 Error Notifications for Data Flash

The error notifications of Data Flash ECC are listed in the following table.

Table 44.6 Error Notifications for Data Flash

Error Notification	Description	ECM Error Number	Error Response to bus master
Data Flash - Data ECC 2-bit error	Data Flash data ECC 2-bit error	88	√
Data Flash - Data ECC 1-bit error	Data Flash data ECC 1-bit error	89	√/—*1
Data Flash - ECC overflow error	Data Flash ECC error address buffer overflow error	90	—

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

44.3.2.3 Error Notifications for Local RAM ECC and Address Feedback

The error notifications of Local RAM ECC and Address Feedback are listed in the following table.

Table 44.7 Error Notifications for Local RAM (1/2)

Error Notification	Description	ECM Error Number	Error Response to bus master
Local RAM - Data ECC 2-bit error	LRAM data ECC 2-bit error (for read data loaded from the other PE's Local RAM speculatively)	96	√
Local RAM - Data ECC 1-bit error	LRAM data ECC 1-bit error (for read data loaded from the other PE's Local RAM speculatively)	97	√/—*1
Local RAM - ECC overflow error	LRAM data ECC error address buffer overflow error (for read data loaded from the other PE's Local RAM speculatively)	98	—
Local RAM (PE0) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error LRAM data ECC 2-bit error	232	√
Local RAM (PE0) - Data ECC 1-bit error	LRAM data ECC 1-bit error	233	√/—*1
Local RAM (PE0) - ECC overflow error	LRAM ECC error address buffer overflow error	234	—
Local RAM (PE1) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error LRAM data ECC 2-bit error	264	√
Local RAM (PE1) - Data ECC 1-bit error	LRAM data ECC 1-bit error	265	√/—*1
Local RAM (PE1) - ECC overflow error	LRAM ECC error address buffer overflow error	266	—

Table 44.7 Error Notifications for Local RAM (2/2)

Error Notification	Description	ECM Error Number	Error Response to bus master
Local RAM (PE2) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error	296	√
	LRAM data ECC 2-bit error		
Local RAM (PE2) - Data ECC 1-bit error	LRAM data ECC 1-bit error	297	√/—*1
Local RAM (PE2) - ECC overflow error	LRAM ECC error address buffer overflow error	298	—
Local RAM (PE3) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error	328	√
	LRAM data ECC 2-bit error		
Local RAM (PE3) - Data ECC 1-bit error	LRAM data ECC 1-bit error	329	√/— *1
Local RAM (PE3) - ECC overflow error	LRAM ECC error address buffer overflow error	330	—

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

44.3.2.4 Error Notifications for Cluster RAM ECC and Address Feedback

The error notifications of Cluster RAM ECC and Address Feedback are listed in the following table.

Table 44.8 Error Notifications for Cluster RAM ECC and Address Feedback

Error Notification	Description	ECM Error Number	Error Response to bus master
Cluster RAM - Address feedback compare error - Data ECC 2-bit error	CRAM address feedback compare error	104	√
	CRAM data ECC 2-bit error		
Cluster RAM - Data ECC 1-bit error	CRAM data ECC 1-bit error	105	√/—*1
Cluster RAM - ECC overflow error	CRAM data ECC error address buffer overflow error	106	—

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

44.3.2.5 Error Notifications for Instruction Cache EDC and Address Feedback

The error notifications of Instruction Cache EDC and Address Feedback are listed in the following table.

Table 44.9 Error Notifications for Instruction Cache

Error Notification	Description	ECM Error Number	Error Response to bus master
Instruction cache RAM (PE0) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	236	√/—*1
	Instruction cache RAM (data) address feedback compare error		
	Instruction cache RAM (tag) data EDC 2-bit error		
	Instruction cache RAM (tag) data EDC 1-bit error		
	Instruction cache RAM (data) data EDC 2-bit error		
	Instruction cache RAM (data) data EDC 1-bit error		
Instruction cache RAM (PE1) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	268	√/—*1
	Instruction cache RAM (data) address feedback compare error		
	Instruction cache RAM (tag) data EDC 2-bit error		
	Instruction cache RAM (tag) data EDC 1-bit error		
	Instruction cache RAM (data) data EDC 2-bit error		
	Instruction cache RAM (data) data EDC 1-bit error		
Instruction cache RAM (PE2) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	300	√/—*1
	Instruction cache RAM (data) address feedback compare error		
	Instruction cache RAM (tag) data EDC 2-bit error		
	Instruction cache RAM (tag) data EDC 1-bit error		
	Instruction cache RAM (data) data EDC 2-bit error		
	Instruction cache RAM (data) data EDC 1-bit error		
Instruction cache RAM (PE3) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	332	√/—*1
	Instruction cache RAM (data) address feedback compare error		
	Instruction cache RAM (tag) data EDC 2-bit error		
	Instruction cache RAM (tag) data EDC 1-bit error		
	Instruction cache RAM (data) data EDC 2-bit error		
	Instruction cache RAM (data) data EDC 1-bit error		

Note 1. It depends on setting of Instruction cache control (ICCTRL) and Instruction cache error (ICERR). For more details of these registers, refer to **Section 3, CPU System**.

44.3.2.6 Error Notifications for sDMAC/DTSRAM ECC and Address Feedback

The error notifications of sDMAC/DTSRAM ECC and Address Feedback are listed in the following table.

Table 44.10 Error Notifications for sDMAC/DTSRAM

Error Notification	Description	ECM Error Number	Error Response to bus master
DTSRAM - Address feedback compare error - Data ECC 2-bit error	DTSRAM address feedback compare error	112	√
	DTSRAM data ECC 2-bit error		
DTSRAM - Data ECC 1-bit error	DTSRAM data ECC 1-bit error	113	√/— *1
DTSRAM - ECC overflow error	DTSRAM ECC error address buffer overflow error	114	—
sDMAC0 RAM - Data ECC 2-bit error	sDMAC0 DPRAM data ECC 2-bit error sDMAC0 DATARAM data ECC 2-bit error	120	√
sDMAC0 RAM - Data ECC 1-bit error	sDMAC0 DPRAM data ECC 1-bit error sDMAC0 DATARAM data ECC 1-bit error	121	√/— *1
sDMAC1 RAM - Data ECC 2-bit error	sDMAC1 DPRAM data ECC 2-bit error sDMAC1 DATARAM data ECC 2-bit error	122	√
sDMAC1 RAM - Data ECC 1-bit error	sDMAC1 DPRAM data ECC 1-bit error sDMAC1 DATARAM data ECC 1-bit error	123	√/— *1
sDMACn RAM (n = 0, 1) - ECC overflow error	sDMAC0 DPRAM data ECC error address buffer overflow error	159	—
	sDMAC0 DATARAM data ECC error address buffer overflow error		
	sDMAC1 DPRAM data ECC error address buffer overflow error		
	sDMAC1 DATARAM data ECC error address buffer overflow error		

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

44.3.2.7 Error Notifications for Peripheral RAM

The error notifications of Peripheral RAM are listed in the following table.

Table 44.11 Error Notifications for Peripheral RAM (1/4)

Error Notification	Description	ECM Error Number	Error Response to bus master
FlexRay RAM - Data ECC 2-bit error	FLXA0 MRAM data ECC 2-bit error	128	—
	FLXA0 TBFRAM A data ECC 2-bit error		
	FLXA0 TBFRAM B data ECC 2-bit error		
	FLXA1 MRAM data ECC 2-bit error		
	FLXA1 TBFRAM A data ECC 2-bit error		
	FLXA1 TBFRAM B data ECC 2-bit error		
FlexRay RAM - Data ECC 1-bit error	FLXA0 MRAM data ECC 1-bit error	129	—
	FLXA0 TBFRAM A data ECC 1-bit error		
	FLXA0 TBFRAM B data ECC 1-bit error		
	FLXA1 MRAM data ECC 1-bit error		
	FLXA1 TBFRAM A data ECC 1-bit error		
	FLXA1 TBFRAM B data ECC 1-bit error		
RS-CANFD RAM - Data ECC 2-bit error	RSCFD0 AFLRAM 0 data ECC 2-bit error	130	—
	RSCFD0 AFLRAM 1 data ECC 2-bit error		
	RSCFD0 MRAM data ECC 2-bit error		
	RSCFD1 AFLRAM 0 data ECC 2-bit error		
	RSCFD1 AFLRAM 1 data ECC 2-bit error		
	RSCFD1 MRAM data ECC 2-bit error		
RS-CANFD RAM - Data ECC 1-bit error	RSCFD0 AFLRAM 0 data ECC 1-bit error	131	—
	RSCFD0 AFLRAM 1 data ECC 1-bit error		
	RSCFD0 MRAM data ECC 1-bit error		
	RSCFD1 AFLRAM 0 data ECC 1-bit error		
	RSCFD1 AFLRAM 1 data ECC 1-bit error		
	RSCFD1 MRAM data ECC 1-bit error		
MSPI RAM -Data ECC 2-bit error	MSPI0 RAM data ECC 2-bit error	132	—
	MSPI1 RAM data ECC 2-bit error		
	MSPI2 RAM data ECC 2-bit error		
	MSPI3 RAM data ECC 2-bit error		
	MSPI4 RAM data ECC 2-bit error		
	MSPI5 RAM data ECC 2-bit error		
	MSPI6 RAM data ECC 2-bit error		
	MSPI7 RAM data ECC 2-bit error		
	MSPI8 RAM data ECC 2-bit error		
	MSPI9 RAM data ECC 2-bit error		

Table 44.11 Error Notifications for Peripheral RAM (2/4)

Error Notification	Description	ECM Error Number	Error Response to bus master
MSPI RAM - Data ECC 1-bit error	MSPI0 RAM data ECC 1-bit error	133	—
	MSPI1 RAM data ECC 1-bit error		
	MSPI2 RAM data ECC 1-bit error		
	MSPI3 RAM data ECC 1-bit error		
	MSPI4 RAM data ECC 1-bit error		
	MSPI5 RAM data ECC 1-bit error		
	MSPI6 RAM data ECC 1-bit error		
	MSPI7 RAM data ECC 1-bit error		
	MSPI8 RAM data ECC 1-bit error		
	MSPI9 RAM data ECC 1-bit error		
GTM RAM - Data ECC 2-bit error	GTM MCS0 RAM 0 data ECC 2-bit error	134	—
	GTM MCS0 RAM 1 data ECC 2-bit error		
	GTM MCS1 RAM 0 data ECC 2-bit error		
	GTM MCS1 RAM 1 data ECC 2-bit error		
	GTM MCS2 RAM 0 data ECC 2-bit error		
	GTM MCS2 RAM 1 data ECC 2-bit error		
	GTM MCS3 RAM 0 data ECC 2-bit error		
	GTM MCS3 RAM 1 data ECC 2-bit error		
GTM RAM - Data ECC 1-bit error	GTM MCS0 RAM 0 data ECC 1-bit error	135	—
	GTM MCS0 RAM 1 data ECC 1-bit error		
	GTM MCS1 RAM 0 data ECC 1-bit error		
	GTM MCS1 RAM 1 data ECC 1-bit error		
	GTM MCS2 RAM 0 data ECC 1-bit error		
	GTM MCS2 RAM 1 data ECC 1-bit error		
	GTM MCS3 RAM 0 data ECC 1-bit error		
	GTM MCS3 RAM 1 data ECC 1-bit error		
Fast Ethernet RAM - Data ECC 2-bit error	ETNB0 Transmit FIFO data ECC 2-bit error	136	—
	ETNB0 Receive FIFO data ECC 2-bit error		
Fast Ethernet RAM - Data ECC 1-bit error	ETNB0 Transmit FIFO data ECC 1-bit error	137	—
	ETNB0 Receive FIFO data ECC 1-bit error		
Gigabit Ethernet RAM - Data ECC 2-bit error	ETNB1 Transmit FIFO data ECC 2-bit error	138	—
	ETNB1 Receive FIFO data ECC 2-bit error		
Gigabit Ethernet RAM - Data ECC 1-bit error	ETNB1 Transmit FIFO data ECC 1-bit error	139	—
	ETNB1 Receive FIFO data ECC 1-bit error		
MMCA RAM - Data ECC 2-bit error	MMCA0 RAM A data ECC 2-bit error	140	—
	MMCA0 RAM B data ECC 2-bit error		
MMCA RAM - Data ECC 1-bit error	MMCA0 RAM A data ECC 1-bit error	141	—
	MMCA0 RAM B data ECC 1-bit error		

Table 44.11 Error Notifications for Peripheral RAM (3/4)

Error Notification	Description	ECM Error Number	Error Response to bus master
Peripheral RAM - ECC overflow error	FLXA0 MRAM ECC error address buffer overflow error	159	—
	FLXA0 TBFRAM A data ECC error address buffer overflow error		
	FLXA0 TBFRAM B data ECC error address buffer overflow error		
	FLXA1 MRAM ECC error address buffer overflow error		
	FLXA1 TBFRAM A data ECC error address buffer overflow error		
	FLXA1 TBFRAM B data ECC error address buffer overflow error		
	RSCFD0 AFLRAM 0 data ECC error address buffer overflow error		
	RSCFD0 AFLRAM 1 data ECC error address buffer overflow error		
	RSCFD0 MRAM data ECC error address buffer overflow error		
	RSCFD1 AFLRAM 0 data ECC error address buffer overflow error		
	RSCFD1 AFLRAM 1 data ECC error address buffer overflow error		
	RSCFD1 MRAM data ECC error address buffer overflow error		
	MSPI0 RAM data ECC error address buffer overflow error		
	MSPI1 RAM data ECC error address buffer overflow error		
	MSPI2 RAM data ECC error address buffer overflow error		
	MSPI3 RAM data ECC error address buffer overflow error		
	MSPI4 RAM data ECC error address buffer overflow error		
	MSPI5 RAM data ECC error address buffer overflow error		
	MSPI6 RAM data ECC error address buffer overflow error		
	MSPI7 RAM data ECC error address buffer overflow error		
	MSPI8 RAM data ECC error address buffer overflow error		
	MSPI9 RAM data ECC error address buffer overflow error		
	GTM MCS0 RAM 0 data ECC error address buffer overflow error		
	GTM MCS0 RAM 1 data ECC error address buffer overflow error		
	GTM MCS1 RAM 0 data ECC error address buffer overflow error		
	GTM MCS1 RAM 1 data ECC error address buffer overflow error		

Table 44.11 Error Notifications for Peripheral RAM (4/4)

Error Notification	Description	ECM Error Number	Error Response to bus master
Peripheral RAM - ECC overflow error	GTM MCS2 RAM 0 data ECC error address buffer overflow error	159	—
	GTM MCS2 RAM 1 data ECC error address buffer overflow error		
	GTM MCS3 RAM 0 data ECC error address buffer overflow error		
	GTM MCS3 RAM 1 data ECC error address buffer overflow error		
	MMCA0 RAM A data ECC error address overflow error		
	MMCA0 RAM B data ECC error address overflow error		
	sDMAC0 DPRAM data ECC error address buffer overflow error		
	sDMAC0 DATARAM data ECC error address buffer overflow error		
	sDMAC1 DPRAM data ECC error address buffer overflow error		
	sDMAC1 DATARAM data ECC error address buffer overflow error		

44.3.2.8 Error Notifications for Data Transfer Path

The error notifications of Data Transfer Path are listed in the following table.

Table 44.12 Error Notifications for Data Transfer Path (1/2)

Error Notification	Description	ECM Error Number	Error Response to bus master
Data transfer path - Address EDC error	Data transfer path in Code Flash address EDC 2-bit error (CCIB, bridge from System Bus to Global FLASH Bus)	160	√/— *2
	Data transfer path in CRAM address EDC 2-bit error (Bridge from System Bus to CRAM Bus, CRAMC)		
	Data transfer path in LRAM address EDC 2-bit error (Bridge from System Bus to Inter-processor element Bus, VCIS I/F in PE)		
	Data transfer path in I-Bus address EDC 2-bit error (Slave modules on I-Bus)		
	Data transfer path in AXI2PVCI bridge address EDC 2-bit error (Bridge from System Bus to P-Bus)		
	Data transfer path in P-Bus address EDC 2-bit error (Slave modules on P-Bus)		
	Data transfer path in remap module address EDC 2-bit error (GCFU for Local FLASH Bus during remap, bridge from Global FLASH Bus to System Bus, GCFU for Global FLASH Bus during remap)		
	Data transfer path in H-Bus address EDC 2-bit error (Slave modules on H-Bus)		
	Data transfer path in Code Flash address EDC 1-bit error (CCIB, bridge from System Bus to Global FLASH Bus)		
	Data transfer path in CRAM address EDC 1-bit error (Bridge from System Bus to CRAM Bus, CRAMC)		
	Data transfer path in LRAM address EDC 1-bit error (Bridge from System Bus to Inter-processor element Bus, VCIS I/F in PE)		
	Data transfer path in I-Bus address EDC 1-bit error (Slave modules on I-Bus)		
	Data transfer path in AXI2PVCI bridge address EDC 1-bit error (Bridge from System Bus to P-Bus)		
	Data transfer path in P-Bus address EDC 1-bit error (Slave modules on P-Bus)		
	Data transfer path in remap module address EDC 1-bit error (GCFU for Local FLASH Bus during remap, bridge from Global FLASH Bus to System Bus, GCFU for Global FLASH Bus during remap)		
	Data transfer path in H-Bus address EDC 1-bit error (Slave modules on H-Bus)		

Table 44.12 Error Notifications for Data Transfer Path (2/2)

Error Notification	Description	ECM Error Number	Error Response to bus master
Data transfer path - data ECC 2-bit error	Data transfer path in CRAM data ECC 2-bit error (RMW-write in CRAMC, RMW-cdata in CRAMC)	161	√/—*2
	Data transfer path in DMAC/DTS data ECC 2-bit error (sDMAC masters, DTS master)		
	Data transfer path in LRAM data ECC 2-bit error (VCIS I/F in PE)		
	Data transfer path in I-Bus data ECC 2-bit error (slave modules on I-Bus)		
	Data transfer path in PE for read data from peripherals data ECC 2-bit error (VCIM I/F in PE)		
	Data transfer path in VCI2AXI bridge for write/read data data ECC 2-bit error (RMW-read in bridge from Inter-processor element Bus to System Bus, RMW-write bridge from Inter-processor element Bus to System Bus)		
	Data transfer path in P-Bus data ECC 2-bit error (Slave modules on P-Bus)		
	Data transfer path in remap module data ECC 2-bit error (Bridge from Global FLASH Bus to System Bus)		
	Data transfer path in H-Bus data ECC 2-bit error (Master/slave modules on H-Bus)		
Data transfer path - data ECC 1-bit error	Data transfer path in CRAM data ECC 1-bit error (RMW-write in CRAMC, RMW-cdata in CRAMC)	162	√/— *1, *2
	Data transfer path in DMAC/DTS data ECC 1-bit error (sDMAC masters, DTS master)		
	Data transfer path in LRAM data ECC 1-bit error (VCIS I/F in PE)		
	Data transfer path in I-Bus data ECC 1-bit error (Slave modules on I-Bus)		
	Data transfer path in PE for read data from peripherals data ECC 1-bit error (VCIM I/F in PE)		
	Data transfer path in VCI2AXI bridge for write/read data data ECC 1-bit error (RMW-read in bridge from Inter-processor element Bus to System Bus, RMW-write bridge from Inter-processor element Bus to System Bus)		
	Data transfer path in P-Bus data ECC 1-bit error (Slave modules on P-Bus)		
	Data transfer path in remap module data ECC 1-bit error (Bridge from Global FLASH Bus to System Bus)		
	Data transfer path in H-Bus data ECC 1-bit error (Master/slave modules on H-Bus)		

Note 1. It depends on setting of ECC 1-bit error correction disable bit (SECDIS). When ECC 1-bit error correction is disabled (SECDIS=1), it causes Error Response. When ECC 1-bit error correction is enabled (SECDIS=0), it doesn't cause Error Response.

Note 2. ECC modules on H-Bus don't make Error response.

44.3.3 Key Code Protection for ECC Control Registers

44.3.3.1 Overview

The product incorporates key code register to protect ECC control registers from unintended access.

44.3.3.2 List of Registers

Table 44.13 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCKCPROT (P-Bus Group 0)	ECC Control Key Code Protection register	ECCKCPROT	FFFB 2800$_H$	32	—	—

44.3.3.3 ECCKCPROT — ECC Control Key Code Protection Register

This register is used for protection against writing to the ECC control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FFFB 2800$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.14 ECCKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit*2 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.
Write A5A5A501$_H$ to this register to enable writing protected registers.

Note 2. Unlocking this protection is one of necessary conditions to enable writing the protected registers. It is also necessary to unlock security protection driven by the ICUMHA when ICUMHA function is active. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

44.3.4 Code Flash ECC and Address Parity

44.3.4.1 Overview

The code flash ECC is summarized in the table below.

Table 44.15 Code Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled; 1-bit error detection and correction and 2-bit error detection are carried out.
Address parity	Address parity check can be either enabled or disabled. When enabled, Address parity is checked during address decoding. When disabled, no address parity check is carried out. In the initial state, this function is enabled.
Error notification	The occurrence of an ECC error or a parity error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. **Address Parity Error:** • Error notification can be either enabled or disabled upon detection of an address parity error. In the initial state, error notification is enabled upon detection of an address parity error. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. An ECC 2-bit error, an ECC 1-bit error, an address parity error, and an overflow error are handled as individual sources in ECM. An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address parity error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address parity error is detected. The error status serves as the enable bit of the capture address. Address buffers are updated if the error status is cleared. Multi-stage address buffers are provided for an ECC 1-bit error. ECC 1-bit error: Four stages ECC 2-bit error, address parity error shared: One stage
Self-diagnosis	The ROM data and the ECC and address parity bits in Code Flash can be read directly. Self-diagnosis of ECC and address parity is possible by reading data from ECC test area.

44.3.4.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 44.16 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_CFP_PE0CL0_base>	FFC4 8000_H	P-Bus Group 0
<ECCCNT_CFP_PE1CL0_base>	FFC4 8080_H	P-Bus Group 0
<ECCCNT_CFP_PE2CL1_base>*1	FFC4 8100_H	P-Bus Group 0
<ECCCNT_CFP_PE3CL1_base>*1	FFC4 8180_H	P-Bus Group 0
<ECCCNT_CFCCL0_base>	FFC4 8800_H	P-Bus Group 0
<ECCCNT_CFCCL1_base>*1	FFC4 8880_H	P-Bus Group 0
<ECCCNT_CFS_base>	FFC4 $8A00_H$	P-Bus Group 0
<MECCCAP_CFL_base>	FFC5 2400_H	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.4.3 List of Registers

Table 44.17 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_CFP_PEnCL0 (n = 0, 1)	ECC control register	CFPECCCTL	<ECCCNT_CFP_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_CFP_PEnCL1 (n = 2, 3)*3	ECC control register	CFPECCCTL	<ECCCNT_CFP_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_CFCCL0	ECC control register	CFCECCCTL	<ECCCNT_CFCCL0_base> + 00_H	8, 16, 32	PBG00#3	ECCKCPROT
ECCCNT_CFCCL1*3	ECC control register	CFCECCCTL	<ECCCNT_CFCCL1_base> + 00_H	8, 16, 32	PBG00#4	ECCKCPROT
ECCCNT_CFS	ECC control register	CFSECCCTL	<ECCCNT_CFS_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
MECCCAP_CFL	Error notification control register	CF_ERRINT	<MECCCAP_CFL_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	CF_SSTCLR	<MECCCAP_CFL_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	CF_DSTCLR	<MECCCAP_CFL_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	CF_OVFCLR	<MECCCAP_CFL_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	CF_SERSTR	<MECCCAP_CFL_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	CF_DERSTR	<MECCCAP_CFL_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	CF_OVFSTR	<MECCCAP_CFL_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	CF_SERINF	<MECCCAP_CFL_base> + 30_H	32	PBG00#2	—

Table 44.17 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MECCCAP_CFL	1st 1-bit error address register	CF_00SEADR	<MECCCAP_CFL_base> + 70_H	32	PBG00#2	—
	2nd 1-bit error address register	CF_01SEADR	<MECCCAP_CFL_base> + 74_H	32	PBG00#2	—
	3rd 1-bit error address register	CF_02SEADR	<MECCCAP_CFL_base> + 78_H	32	PBG00#2	—
	4th 1-bit error address register	CF_03SEADR	<MECCCAP_CFL_base> + $7C_H$	32	PBG00#2	—
	1st fatal error address register	CF_00DEADR	<MECCCAP_CFL_base> + $F0_H$	32	PBG00#2	—

Note 1. n=0: PBG01#0
n=1: PBG01#1

Note 2. n=2: PBG01#2
n=3: PBG01#3

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.18 The relation between the module storing the error and the error source for Code Flash

Protection target	Function	Controlled by	Captured to	Applicable for
Data from CodeFlash at Code Flash I/F in PE0	Data ECC decode	ECCCNT_CFP_PE0CL0	MECCCAP_CFL	Instruction from CodeFlash area
				Load data from CodeFlash area
Data from CodeFlash at Code Flash I/F in PE1		ECCCNT_CFP_PE1CL0		Instruction from CodeFlash area
				Load data from CodeFlash area
Data from CodeFlash at Code Flash I/F in PE2*1		ECCCNT_CFP_PE2CL1		Instruction from CodeFlash area
				Load data from CodeFlash area
Data from CodeFlash at Code Flash I/F in PE3*1		ECCCNT_CFP_PE3CL1		Instruction from CodeFlash area
				Load data from CodeFlash area
Data from CodeFlash at CCIB (Cluster0)		ECCCNT_CFCCL0		Read data from CodeFlash area bank A/B in cluster 0
Data from CodeFlash at CCIB (Cluster1)*1		ECCCNT_CFCCL1		Read data from CodeFlash area bank C/D in cluster 1
Data from CodeFlash at SAXI2FAXI (bridge from System Bus to Global Flash Bus)		ECCCNT_CFS		Read data from CodeFlash area
Address to CodeFlash at CCIB (Cluster0)	Address parity check	ECCCNT_CFCCL0		Request address to CodeFlash area
Address to CodeFlash at CCIB (Cluster1)*1		ECCCNT_CFCCL1		Request address to CodeFlash area

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.4.4 CFPECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from CodeFlash to PE.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CFP_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_CFP_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.19 CFPECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction disable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correction is enabled when 1-bit error is detected 1: Correction is disabled when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled.

44.3.4.5 CFCECCCTL — ECC Control Register

This register controls address parity error detection, ECC error detection/correction and 1-bit error correction for read data from CodeFlash.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CFCCL0_base> + 00_H
<ECCCNT_CFCCL1_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	BLANKMASKENABLE	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	APEDIS	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 44.20 CFCECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	BLANKMASKENABLE	Blank read error mask bit Enable/disable ECC error and address parity error detection when reading blank data from CodeFlash 0: Blank read errors are checked 1: Blank read errors are ignored
23 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	APEDIS	Address parity error disable bit Setting address parity error detection to enable/disable. 0: Address parity error detection is enabled 1: Address parity error detection is disabled
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.4.6 CFSECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data to SAXI2FAXI bridge.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CFS_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.21 CFSECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.4.7 CF_ERRINT — Error Notification Control Register

This register controls whether error information is reported to the ECM when an address parity error, ECC error and/or overflow error has occurred.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 00_H

Value after reset: $0000\ 0087_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	—	APEIE	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W	R/W

Table 44.22 CF_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	APEIE	Controls error reports when address parity error is detected. 0: Address parity error report disabled 1: Address parity error report enabled
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.4.8 CF_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in CF_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 44.23 CF_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in CF_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in CF_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in CF_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in CF_SERSTR.

44.3.4.9 CF_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in CF_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.24 CF_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears APEF00 and DEDF00 in CF_DERSTR.

44.3.4.10 CF_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in CF_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.25 CF_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in CF_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in CF_OVFSTR.

44.3.4.11 CF_SERSTR — 1-bit Error Status Register

This register indicates whether an ECC 1-bit error has occurred. The location and address of the error that is detected are stored in CF_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in CF_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDF 03	SEDF 02	SEDF 01	SEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.26 CF_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in CF_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in CF_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in CF_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in CF_00SEADR register.

44.3.4.12 CF_DERSTR — Fatal Error Status Register

This register indicates whether an address parity error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in CF_00DEADR register when APEF00 and/or DEDF00 flags are set. The APEF00 and/or DEDF00 flag is set only when an address parity error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in CF_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 24_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	APEF00	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.27 CF_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	APEF00	Indicates that an address parity error was detected.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.4.13 CF_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether an overflow of the address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in CF_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CFL_base> + 28_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.28 CF_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that ECC 1-bit error s over working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error is detected when all ECC 1-bit error flags in CF_SERSTR are set.

44.3.4.14 CF_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in CF_SERSTR register were detected.

This register is updated whenever CF_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CFL_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDLINF07	SEDLINF06	SEDLINF05	SEDLINF04	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.29 CF_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned.
16	SEDLINF16	Indicates that an ECC 1-bit error was detected in read data from CodeFlash to SAXI2FAXI.
15 to 8	Reserved	When read, the value after reset is returned.
7	SEDLINF07	Indicates that an ECC 1-bit error was detected in read data loaded from CodeFlash to PE3.*1
6	SEDLINF06	Indicates that an ECC 1-bit error was detected in read data fetched from CodeFlash to PE3.*1
5	SEDLINF05	Indicates that an ECC 1-bit error was detected in read data loaded from CodeFlash to PE2.*1
4	SEDLINF04	Indicates that an ECC 1-bit error was detected in read data fetched from CodeFlash to PE2.*1
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in read data loaded from CodeFlash to PE1.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in read data fetched from CodeFlash to PE1.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in read data loaded from CodeFlash to PE0.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data fetched from CodeFlash to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.4.15 CF_nSEADR — n-th 1-bit Error Address Register (n = 00 to 03)

This register is used to hold the address and the location of the error when the corresponding SEDFn flag is set.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CFL_base> + n × 4_H + 70_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	SEADR0[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.30 CF_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26	Reserved	When read, the value after reset is returned.
25 to 2	SEADR0[25:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.31 SEDL information

SEDL[4:0]	Indicated information
31 to 17	Reserved
16	Indicates an ECC 1-bit error is detected in loading data from CodeFlash by any master except PEs.
15 to 8	Reserved
7	Indicates an ECC 1-bit error is detected in loading data from CodeFlash by PE3*1 master.
6	Indicates an ECC 1-bit error is detected in fetching instructions from CodeFlash by PE3*1 master.
5	Indicates an ECC 1-bit error is detected in loading data from CodeFlash by PE2*1 master.
4	Indicates an ECC 1-bit error is detected in fetching instructions from CodeFlash by PE2*1 master.
3	Indicates an ECC 1-bit error is detected in loading data from CodeFlash by PE1 master.
2	Indicates an ECC 1-bit error is detected in fetching instructions from CodeFlash by PE1 master.
1	Indicates an ECC 1-bit error is detected in loading data from CodeFlash by PE0 master.
0	Indicates an ECC 1-bit error is detected in fetching instructions from CodeFlash by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.4.16 CF_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither APEF00 nor DEDF00 flag is set and an address parity error and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CFL_base> + $F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	DEADR0[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.32 CF_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26	Reserved	When read, the value after reset is returned.
25 to 2	DEADR0[25:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.33 DEDL Information

DEDL[4:0]	Indicated information
31 to 17	Reserved
16	Indicates a fatal error is detected in loading data from CodeFlash by any master except PEs.
15 to 8	Reserved
7	Indicates a fatal error is detected in loading data from CodeFlash by PE3*1 master.
6	Indicates a fatal error is detected in fetching instructions from CodeFlash by PE3*1 master.
5	Indicates a fatal error is detected in loading data from CodeFlash by PE2*1 master.
4	Indicates a fatal error is detected in fetching instructions from CodeFlash by PE2*1 master.
3	Indicates a fatal error is detected in loading data from CodeFlash by PE1 master.
2	Indicates a fatal error is detected in fetching instructions from CodeFlash by PE1 master.
1	Indicates a fatal error is detected in loading data from CodeFlash by PE0 master.
0	Indicates a fatal error is detected in fetching instructions from CodeFlash by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.4.17 Test Function

(1) Reading Code Flash

(a) Set ECCDIS bit in CFPECCCTL register to 1 to disable ECC error detection and correction.

(b) When ECCDIS=1, neither error detection nor correction proceeds when the code flash is read; the data output from the code flash is read unchanged.

How to exit this test mode:

(a) Set ECCDIS bit in CFPECCCTL register to 0 to enable ECC error detection and correction.

(2) Reading the ECC and address parity bits

(a) When the Blank check area of Code Flash is read, the ECC and address parity bits of corresponding address are read. The bit assign is as follows:

Table 44.34 Bit assign of read data when the Blank check area of Code Flash is read

Bit Position	Function
31 to 16	Always 0
15	Blank Flag
14 to 11	Always 0
10	Address parity bit
9 to 0	ECC bits for Code Flash 256bit data

(3) Self-diagnosis of ECC and address parity check function

Self-diagnosis of the ECC decoder and address parity checker for the access ports is possible by reading data from ECC test area.

For details about the ECC test area, refer to *RH850/U2Ax Safety Application Note*.

For the address location of the ECC test area, refer to **Section 4, Address Space**.

44.3.5 Data Flash ECC

44.3.5.1 Overview

The data flash ECC is summarized in the table below.

Table 44.35 Data Flash ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of a 2-bit ECC error. • Error notification can be either enabled or disabled upon detection of a 1-bit ECC error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources in ECM.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address. If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured.
Self-diagnosis	The ROM data and the ECC bits in Data Flash can be read directly. Desired values can be written to the data and the ECC bits in Data Flash.

44.3.5.2 List of Registers

Table 44.36 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCDF (P-Bus Group 1)	Data Flash ECC control register	DFECCCTL	FFC6 2C00$_H$	16, 32	PBG10#2	DFKCPROT
	Data Flash error status register	DFERSTR	FFC6 2C04$_H$	32	PBG10#2	—
	Data Flash error status clear register	DFERSTC	FFC6 2C08$_H$	8, 16, 32	PBG10#2	—
	Data Flash error overflow status register	DFOVFSTR	FFC6 2C0C$_H$	32	PBG10#2	—
	Data Flash error overflow status clear register	DFOVFSTC	FFC6 2C10$_H$	8, 16, 32	PBG10#2	—
	Data Flash error notification control register	DFERRINT	FFC6 2C14$_H$	8, 16, 32	PBG10#2	DFKCPROT
	Data Flash 1st error address register	DFEADR	FFC6 2C18$_H$	32	PBG10#2	—
	Data flash test control register	DFTSTCTL	FFC6 2C1C$_H$	16, 32	PBG10#2	DFKCPROT
	Data Flash ECC key code protection register	DFKCPROT	FFC6 2C20$_H$	32	PBG10#2	—

Table 44.37 The relation between the module storing the error and the error source for Data Flash

Protection target	Function	Controlled by	Captured to	Applicable for
Data Flash	Data ECC decode	ECCDF	ECCDF	Read data from Data Flash area

44.3.5.3 DFECCCTL — Data Flash ECC Control Register

DFECCCTL enables or disables ECC error detection and 1-bit error correction for read access.

DFECCCTL register is protected by DFKCPROT register.

Access: This register can be read or write in 32-bit or 16-bit units.

Address: FFC6 2C00$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.38 DFECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correction is enabled when 1-bit error is detected. 1: Correction is disabled when 1-bit error is detected.
0	ECCDIS	ECC Disable Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

44.3.5.4 DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

SEDF bit is set if an ECC 1-bit error is detected while ECC error detection/correction is enabled (ECCDIS = 0), and the DEDF bit is set if an ECC 2-bit error is detected.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 2C04$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.39 DFERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: An ECC 2-bit error is not generated. 1: An ECC 2-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 2-bit error is generated.
0	SEDF	ECC 1-bit error Monitor Flag 0: An ECC 1-bit error is not generated. 1: An ECC 1-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 1-bit error is generated with both SEDF and DEDF being 0.

44.3.5.5 DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the Data Flash error status register. DFERSTC is a write-only register and is always read as 0.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: FFC6 2C08$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR CLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.40 DFERSTC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

44.3.5.6 DFOVFSTR — Data Flash Error Overflow Status Register

DFOVFSTR monitors occurrence of Data Flash error overflow.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 2C0C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.41 DFOVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag ERROVF is set if the following occur: • An ECC 1-bit error occurs when DFERSTR.SEDF = 1 and access address is different from DFEADR. • An ECC 1-bit error occurs when DFERSTR.DEDF = 1 • An ECC 2-bit error occurs when DFERSTR.SEDF = 1 • An ECC 2-bit error occurs when DFERSTR.DEDF = 1 and access address is different from DFEADR. 0: Did not occur 1: Occurred Clearing condition: Set the ERROVFCLR bit in DFOVFSTC to 1.

44.3.5.7 DFOVFSTC — Data Flash Error Overflow Status Clear Register

DFOVFSTC clears the Data Flash error overflow flag.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: FFC6 2C10$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF CLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.42 DFOVFSTC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear 0: No effect (Setting the ERROVFCLR bit to 0 does not affect the flag in DFOVFSTR.) 1: The ERROVF flag in the DFOVFSTR register is cleared.

44.3.5.8 DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal to ECM upon detection of an ECC 2-bit error or an ECC 1-bit error.

DFERRINT register is protected by DFKCPROT register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: FFC6 2C14$_H$

Value after reset: 0000 0007$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	EOVFIE	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 44.43 DFERRINT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	EOVFIE	ECC Error Address Overflow Notification Control Enables or disables generation of the error notification signal upon detection of an address buffer overflow error. 0: Error address overflow report disabled 1: Error address overflow report enabled
1	DEDIE	ECC 2-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection/correction is enabled (ECCDIS = 0). 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection/correction is enabled (ECCDIS = 0). 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

44.3.5.9 DFEADR — Data Flash 1st Error Address Register

DFEADR holds the address if the following occur:

- An ECC 1-bit error occurs when DFERSTR.SEDF = 0 and DFERSTR.DEDF = 0
- An ECC 2-bit error occurs when DFERSTR.DEDF = 0

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 2C18$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	DFEADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DFEADR[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.44 DFEADR Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 2	DFEADR[20:2]	ECC Error Address DFEADR is a read-only field to monitor the address at which an ECC error has occurred. This register holds an internal address. Convert it to the actual address by adding the data flash base address described in **Section 4, Address Space**.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

44.3.5.10 DFTSTCTL — Data Flash Test Control Register

DFTSTCTL is used for ECC testing.

The data of the ECC bit can be read after setting the ECC test mode (ECCTST = 1).

DFTSTCTL register is protected by DFKCPROT register.

Access: This register can be read or written in 32-bit or 16-bit units.

Address: FFC6 2C1C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.45 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test By setting ECC test mode bit to “1” (ECCTST = 1), CPU can read ECC bit.

44.3.5.11 DFKCPROT — Data Flash ECC Key Code Protection Register

DFKCPROT is used for access protection of other Data Flash ECC registers.

Protected registers:

DFECCCTL

DFERRINT

DFTSTCTL

Access: This register can be read or written in 32-bit units.

Address: FFC6 2C20$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.46 DFKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.
Write A5A5A501$_H$ to this register to enable writing protected registers.

44.3.5.12 Test Function

Data in the ROM and the ECC bits can be read by setting the data flash test control register (DFTSTCTL).

(1) Reading the ROM data

(a) Set ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.

(b) When ECCDIS = 1, neither error detection nor correction proceeds when the data flash is read; the data output from the data flash is read unchanged.

How to exit this test mode:

(a) Set ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.

(2) Reading the ECC bits

(a) Set ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.

(b) Set ECCTST bit in the data flash test control register to 1 to set test mode.

(c) When the data flash is read, the 7 lower-order bits of read data are read as ECC data.

How to exit this test mode:

(a) Set ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.

(b) Set ECCTST bit in the data flash test control register to 0 to set normal mode.

(3) Self-diagnosis of ECC check function

Self-diagnosis of the ECC decoder is possible by writing incorrect data to the data flash memory beforehand (fault injection) and then reading this data. A 1- or 2-bit ECC error fault can be injected by generating correct ECC bits once and inverting only the appropriate bits. For details on programming of the data flash, refer to the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

44.3.6 Local RAM (CPUs) ECC and Address Feedback

44.3.6.1 Overview

Local RAM ECC of CPUs is summarized in the table below.

Table 44.47 Local RAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected: • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Address feedback	Address feedback check can be either enabled or disabled. When enabled, Address feedback error detection is carried out. When disabled, no error detection is carried out. In the initial state, this function is enabled.
Error notification	The occurrence of an ECC error or an address feedback error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. **Address Feedback Error:** • Error notification can be either enabled or disabled upon detection of an address feedback error. In the initial state, error notification is enabled upon detection of an address feedback error. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. The error notification signal for both an address feedback error and an ECC 2-bit error is handled as one source in ECM. An ECC 1-bit error and an overflow error are handled as individual sources in ECM. An ECC 1-bit error signal is only issued to the ECM, if the ECC 1-bit error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the address that is captured. Address buffers are updated if the error status is cleared. Multi-stage address buffers are provided for an ECC 1-bit error. ECC 1-bit error: Eight stages (for each CPU) ECC 2-bit error and address feedback error: One stage (for each CPU)
Self-diagnosis	Desired values can be written to the RAM data and the ECC. The RAM data can be read directly and the ECC can be read via Read Buffer Register in ECC Test Mode. Moreover, the error injection to an address feedback checker is possible by setting AFINV.

ECC of Local RAM is provided for each 32 bits of data and each 32-bit data is called bank 0 to 3. The relationship between addresses and bank numbers are as follows.

Table 44.48 Relationship between addresses and bank numbers

4 lower-order bits of address (Hexadecimal Notation)	F_H to C_H	B_H to 8_H	7_H to 4_H	3_H to 0_H
Bank number	Bank 3	Bank 2	Bank 1	Bank 0

44.3.6.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.49 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_LR_PE0CL0_base>	FFC4 9800_H	P-Bus Group 0
<ECCCNT_LR_PE1CL0_base>	FFC4 9880_H	P-Bus Group 0
<ECCCNT_LR_PE2CL1_base>*1	FFC4 9900_H	P-Bus Group 0
<ECCCNT_LR_PE3CL1_base>*1	FFC4 9980_H	P-Bus Group 0
<MECCCAP_LR_PE0CL0_base>	FFC5 0000_H	P-Bus Group 0
<MECCCAP_LR_PE1CL0_base>	FFC5 0100_H	P-Bus Group 0
<MECCCAP_LR_PE2CL1_base>*1	FFC5 0200_H	P-Bus Group 0
<MECCCAP_LR_PE3CL1_base>*1	FFC5 0300_H	P-Bus Group 0
<MECCCAP_LRA_base>	FFC5 0800_H	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.6.3 List of Registers

Table 44.50 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_LR_PEnCL0 (n = 0, 1)	ECC control register	LRECCCTL	<ECCCNT_LR_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	ECC test control register	LRECCTSTCTL	<ECCCNT_LR_PEnCL0_base> + 10_H	8, 16, 32	*1	ECCKCPROT
	Address feedback test control register	LRAFINV	<ECCCNT_LR_PEnCL0_base> + 20_H	8, 16, 32	*1	ECCKCPROT
	ECC read buffer register	LRTDATBFECCF	<ECCCNT_LR_PEnCL0_base> + 60_H	8, 16, 32	*1	—
ECCCNT_LR_PEnCL1 (n = 2, 3)*3	ECC control register	LRECCCTL	<ECCCNT_LR_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	ECC test control register	LRECCTSTCTL	<ECCCNT_LR_PEnCL1_base> + 10_H	8, 16, 32	*2	ECCKCPROT
	Address feedback test control register	LRAFINV	<ECCCNT_LR_PEnCL1_base> + 20_H	8, 16, 32	*2	ECCKCPROT
	ECC read buffer register	LRTDATBFECCF	<ECCCNT_LR_PEnCL1_base> + 60_H	8, 16, 32	*2	—
MECCCAP_LR_PEnCL0 (n = 0, 1)	Error notification control register	LR0_ERRINT	<MECCCAP_LR_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	1-bit error status clear register	LR0_SSTCLR	<MECCCAP_LR_PEnCL0_base> + 10_H	8, 16, 32	*1	—
	Fatal error status clear register	LR0_DSTCLR	<MECCCAP_LR_PEnCL0_base> + 14_H	8, 16, 32	*1	—
	1-bit error overflow status clear register	LR0_OVFCLR	<MECCCAP_LR_PEnCL0_base> + 18_H	8, 16, 32	*1	—
	1-bit error status register	LR0_SERSTR	<MECCCAP_LR_PEnCL0_base> + 20_H	8, 16, 32	*1	—
	Fatal error status register	LR0_DERSTR	<MECCCAP_LR_PEnCL0_base> + 24_H	8, 16, 32	*1	—
	1-bit error overflow status register	LR0_OVFSTR	<MECCCAP_LR_PEnCL0_base> + 28_H	8, 16, 32	*1	—
	1-bit error location information register	LR0_SERINF	<MECCCAP_LR_PEnCL0_base> + 30_H	32	*1	—

Table 44.50 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MECCCAP_LR_PEnCL0 (n = 0, 1)	1st 1-bit error address register	LR0_00SEADR	<MECCCAP_LR_PEnCL0_base> + 70_H	32	*1	—
	2nd 1-bit error address register	LR0_01SEADR	<MECCCAP_LR_PEnCL0_base> + 74_H	32	*1	—
	3rd 1-bit error address register	LR0_02SEADR	<MECCCAP_LR_PEnCL0_base> + 78_H	32	*1	—
	4th 1-bit error address register	LR0_03SEADR	<MECCCAP_LR_PEnCL0_base> + $7C_H$	32	*1	—
	5th 1-bit error address register	LR0_04SEADR	<MECCCAP_LR_PEnCL0_base> + 80_H	32	*1	—
	6th 1-bit error address register	LR0_05SEADR	<MECCCAP_LR_PEnCL0_base> + 84_H	32	*1	—
	7th 1-bit error address register	LR0_06SEADR	<MECCCAP_LR_PEnCL0_base> + 88_H	32	*1	—
	8th 1-bit error address register	LR0_07SEADR	<MECCCAP_LR_PEnCL0_base> + $8C_H$	32	*1	—
	1st fatal error address register	LR0_00DEADR	<MECCCAP_LR_PEnCL0_base> + $F0_H$	32	*1	—
MECCCAP_LR_PEnCL1 (n = 2, 3) *3	Error notification control register	LR0_ERRINT	<MECCCAP_LR_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	1-bit error status clear register	LR0_SSTCLR	<MECCCAP_LR_PEnCL1_base> + 10_H	8, 16, 32	*2	—
	Fatal error status clear register	LR0_DSTCLR	<MECCCAP_LR_PEnCL1_base> + 14_H	8, 16, 32	*2	—
	1-bit error overflow status clear register	LR0_OVFCLR	<MECCCAP_LR_PEnCL1_base> + 18_H	8, 16, 32	*2	—
	1-bit error status register	LR0_SERSTR	<MECCCAP_LR_PEnCL1_base> + 20_H	8, 16, 32	*2	—
	Fatal error status register	LR0_DERSTR	<MECCCAP_LR_PEnCL1_base> + 24_H	8, 16, 32	*2	—
	1-bit error overflow status register	LR0_OVFSTR	<MECCCAP_LR_PEnCL1_base> + 28_H	8, 16, 32	*2	—
	1-bit error location information register	LR0_SERINF	<MECCCAP_LR_PEnCL1_base> + 30_H	32	*2	—
	1st 1-bit error address register	LR0_00SEADR	<MECCCAP_LR_PEnCL1_base> + 70_H	32	*2	—
	2nd 1-bit error address register	LR0_01SEADR	<MECCCAP_LR_PEnCL1_base> + 74_H	32	*2	—
	3rd 1-bit error address register	LR0_02SEADR	<MECCCAP_LR_PEnCL1_base> + 78_H	32	*2	—
	4th 1-bit error address register	LR0_03SEADR	<MECCCAP_LR_PEnCL1_base> + $7C_H$	32	*2	—
	5th 1-bit error address register	LR0_04SEADR	<MECCCAP_LR_PEnCL1_base> + 80_H	32	*2	—
	6th 1-bit error address register	LR0_05SEADR	<MECCCAP_LR_PEnCL1_base> + 84_H	32	*2	—
	7th 1-bit error address register	LR0_06SEADR	<MECCCAP_LR_PEnCL1_base> + 88_H	32	*2	—
	8th 1-bit error address register	LR0_07SEADR	<MECCCAP_LR_PEnCL1_base> + $8C_H$	32	*2	—
	1st fatal error address register	LR0_00DEADR	<MECCCAP_LR_PEnCL1_base> + $F0_H$	32	*2	—

Table 44.50 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MECCCAP_LRA	Error notification control register	LR1_ERRINT	<MECCCAP_LRA_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	LR1_SSTCLR	<MECCCAP_LRA_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	LR1_DSTCLR	<MECCCAP_LRA_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	LR1_OVFCLR	<MECCCAP_LRA_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	LR1_SERSTR	<MECCCAP_LRA_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	LR1_DERSTR	<MECCCAP_LRA_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	LR1_OVFSTR	<MECCCAP_LRA_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	LR1_SERINF	<MECCCAP_LRA_base> + 30_H	32	PBG00#2	—
	1st 1-bit error address register	LR1_00SEADR	<MECCCAP_LRA_base> + 70_H	32	PBG00#2	—
	2nd 1-bit error address register	LR1_01SEADR	<MECCCAP_LRA_base> + 74_H	32	PBG00#2	—
	3rd 1-bit error address register	LR1_02SEADR	<MECCCAP_LRA_base> + 78_H	32	PBG00#2	—
	4th 1-bit error address register	LR1_03SEADR	<MECCCAP_LRA_base> + $7C_H$	32	PBG00#2	—
	5th 1-bit error address register	LR1_04SEADR	<MECCCAP_LRA_base> + 80_H	32	PBG00#2	—
	6th 1-bit error address register	LR1_05SEADR	<MECCCAP_LRA_base> + 84_H	32	PBG00#2	—
	7th 1-bit error address register	LR1_06SEADR	<MECCCAP_LRA_base> + 88_H	32	PBG00#2	—
	8th 1-bit error address register	LR1_07SEADR	<MECCCAP_LRA_base> + $8C_H$	32	PBG00#2	—
	1st fatal error address register	LR1_00DEADR	<MECCCAP_LRA_base> + $F0_H$	32	PBG00#2	—

Note 1. n=0: PBG01#0
n=1: PBG01#1

Note 2. n=2: PBG01#2
n=3: PBG01#3

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.51 The relation between the module storing the error and the error source for Local RAM

Protection target	Function	Controlled by	Captured to	Applicable for
Data from LRAM at LRAM I/F in PE0	Data ECC decode	ECCCNT_LR_PE0CL0	MECCCAP_LR_PE0CL0	Read data from LRAM area of PE0 requested by PE0 master
			MECCCAP_LR_PE0CL0	Read data from LRAM area of PE0 requested by other than PEs
			MECCCAP_LR_PE0CL0	Read data from LRAM area of PE0 requested by other PEs in non-speculative access*1
			MECCCAP_LRA	Read data from LRAM area of PE0 requested by other PEs in speculative access*1
Data from LRAM at LRAM I/F in PE1	Data ECC decode	ECCCNT_LR_PE1CL0	MECCCAP_LR_PE1CL0	Read data from LRAM area of PE1 requested by PE1 master
			MECCCAP_LR_PE1CL0	Read data from LRAM area of PE1 requested by other than PEs
			MECCCAP_LR_PE1CL0	Read data from LRAM area of PE1 requested by other PEs in non-speculative access*1
			MECCCAP_LRA	Read data from LRAM area of PE1 requested by other PEs in speculative access*1
Data from LRAM at LRAM I/F in PE2*2	Data ECC decode	ECCCNT_LR_PE2CL1	MECCCAP_LR_PE2CL1	Read data from LRAM area of PE2 requested by PE2 master
			MECCCAP_LR_PE2CL1	Read data from LRAM area of PE2 requested by other than PEs
			MECCCAP_LR_PE2CL1	Read data from LRAM area of PE2 requested by other PEs in non-speculative access*1
			MECCCAP_LRA	Read data from LRAM area of PE2 requested by other PEs in speculative access*1
Data from LRAM at LRAM I/F in PE3*2	Data ECC decode	ECCCNT_LR_PE3CL1	MECCCAP_LR_PE3CL1	Read data from LRAM area of PE3 requested by PE3 master
			MECCCAP_LR_PE3CL1	Read data from LRAM area of PE3 requested by other than PEs
			MECCCAP_LR_PE3CL1	Read data from LRAM area of PE3 requested by other PEs in non-speculative access*1
			MECCCAP_LRA	Read data from LRAM area of PE3 requested by other PEs in speculative access*1
Address to LRAM at LRAM I/F in PE0	Address feedback check	ECCCNT_LR_PE0CL0	MECCCAP_LR_PE0CL0	Request address to LRAM area of PE0
Address to LRAM at LRAM I/F in PE1		ECCCNT_LR_PE1CL0	MECCCAP_LR_PE1CL0	Request address to LRAM area of PE1
Address to LRAM at LRAM I/F in PE2*2		ECCCNT_LR_PE2CL1	MECCCAP_LR_PE2CL1	Request address to LRAM area of PE2
Address to LRAM at LRAM I/F in PE3*2		ECCCNT_LR_PE3CL1	MECCCAP_LR_PE3CL1	Request address to LRAM area of PE3

Note 1. For details about speculative access, refer to **Section 3, CPU System**.

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.6.4 LRECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from Local RAM and address feedback error detection for request address to Local RAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_LR_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_LR_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 44.52 LRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.6.5 LRECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), either data field or ECC field of Local RAM can be written separately.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_LR_PEnCL0_base> + 10_H (n = 0, 1)
<ECCCNT_LR_PEnCL1_base> + 10_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.53 LRECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Sets the write mode to Local RAM. 0: Normal mode 1: Test mode Either data field or ECC field can be written according to DATSEL setting in test mode.
0	DATSEL	Sets the field to write data when ECCTST = 1. 0: Data field only 1: ECC field only

44.3.6.6 LRAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from Local RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_LR_PEnCL0_base> + 20_H (n = 0, 1)
<ECCCNT_LR_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	AFINV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFINV[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 44.54 LRAFINV Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 4	AFINV[20:4]	Specifies bit pattern to inject errors into feedback address.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.6.7 LRTDATBFECCF — ECC Read Buffer Register

This register is used to hold ECC field read from Local RAM for self-diagnosis. ECC field in requested address can be stored this register while reading from Local RAM where ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_LR_PEnCL0_base> + 60_H (n = 0, 1)
<ECCCNT_LR_PEnCL1_base> + 60_H (n = 2, 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BFECC3[6:0]							—	BFECC2[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BFECC1[6:0]							—	BFECC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.55 LRTDATBFECCF Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 24	BFECC3[6:0]	ECC field read from the address that is requested in Local RAM bank 3
23	Reserved	When read, the value after reset is returned.
22 to 16	BFECC2[6:0]	ECC field read from the address that is requested in Local RAM bank 2
15	Reserved	When read, the value after reset is returned.
14 to 8	BFECC1[6:0]	ECC field read from the address that is requested in Local RAM bank 1
7	Reserved	When read, the value after reset is returned.
6 to 0	BFECC0[6:0]	ECC field read from the address that is requested in Local RAM bank 0

44.3.6.8 LR0_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error has occurred.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 00_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 008B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 44.56 LR0_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.6.9 LR0_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in LR0_SERSTR. This is a write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 10_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 10_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 44.57 LR0_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in LR0_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in LR0_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in LR0_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in LR0_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in LR0_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in LR0_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in LR0_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in LR0_SERSTR.

44.3.6.10 LR0_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in LR0_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 14_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 14_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.58 LR0_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in LR0_DERSTR.

44.3.6.11 LR0_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in LR0_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 18_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 18_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.59 LR0_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in LR0_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in LR0_OVFSTR.

44.3.6.12 LR0_SERSTR — 1-bit Error Status Register

This register indicates whether an ECC 1-bit error has occurred. The location and address of the error that is detected are stored in LR0_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in LR0_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 20_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF 07	SEDF 06	SEDF 05	SEDF 04	SEDF 03	SEDF 02	SEDF 01	SEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.60 LR0_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in LR0_00SEADR register.

44.3.6.13 LR0_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in LR0_00DEADR register when AFEF00 and/or DEDF00 flag are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in LR0_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 24_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 24_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.61 LR0_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	AFEF00	Indicates that an address feedback error was detected.
2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.6.14 LR0_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address, is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in LR0_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 28_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 28_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.62 LR0_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error is detected when all ECC 1-bit error flags in LR0_SERSTR are set.

44.3.6.15 LR0_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in LR0_SERSTR register were detected.

This register is updated whenever LR0_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + 30_H (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + 30_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SEDLINF04	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.63 LR0_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4	SEDLINF04	Indicates that an ECC 1-bit error was detected in bank 3 of Local RAM.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in bank 2 of Local RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in bank 1 of Local RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in bank 0 of Local RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data loaded from its own Local RAM speculatively.

44.3.6.16 LR0_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location of the error when the corresponding SEDFn flag is set.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LR_PEmCL0_base> + n × 4_H + 70_H (m = 0, 1)
<MECCCAP_LR_PEmCL1_base> + n × 4_H + 70_H (m = 2, 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.64 LR0_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 16	Reserved	When read, the value after reset is returned.
15 to 2	SEADR0[15:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.65 SEDL information

SEDL[4:0]	Indicated information
31 to 5	Reserved
4	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 3 of Local RAM.
3	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 2 of Local RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 1 of Local RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or fetching instructions from bank 0 of Local RAM.
0	Indicates an ECC 1-bit error is detected in loading data from its own Local RAM.

44.3.6.17 LR0_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LR_PEnCL0_base> + $F0_H$ (n = 0, 1)
<MECCCAP_LR_PEnCL1_base> + $F0_H$ (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.66 LR0_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 16	Reserved	When read, the value after reset is returned.
15 to 2	DEADR0[15:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.67 DEDL Information

DEDL[4:0]	Indicated information
31 to 5	Reserved
4	Indicates a fatal error is detected in storing data to or fetching instructions from bank 3 of Local RAM.
3	Indicates a fatal error is detected in storing data to or fetching instructions from bank 2 of Local RAM.
2	Indicates a fatal error is detected in storing data to or fetching instructions from bank 1 of Local RAM.
1	Indicates a fatal error is detected in storing data to or fetching instructions from bank 0 of Local RAM.
0	Indicates a fatal error is detected in loading data from its own Local RAM.

44.3.6.18 LR1_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM when an ECC error and/or overflow error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 00_H

Value after reset: $0000\ 0083_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 44.68 LR1_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.6.19 LR1_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in LR1_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 44.69 LR1_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in LR1_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in LR1_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in LR1_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in LR1_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in LR1_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in LR1_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in LR1_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in LR1_SERSTR.

44.3.6.20 LR1_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in LR1_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.70 LR1_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in LR1_DERSTR.

44.3.6.21 LR1_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in LR1_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.71 LR1_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in LR1_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in LR1_OVFSTR.

44.3.6.22 LR1_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in LR1_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in LR1_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF 07	SEDF 06	SEDF 05	SEDF 04	SEDF 03	SEDF 02	SEDF 01	SEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.72 LR1_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in LR1_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error r was detected and error address is stored in LR1_00SEADR register.

44.3.6.23 LR1_DERSTR — Fatal Error Status Register

This register indicates whether ECC 2-bit error has occurred. The location and address of the error that is detected are stored in LR1_00DEADR register when DEDF00 flags are set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in LR1_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 24_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.73 LR1_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.6.24 LR1_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address, is detected when the address buffer is fully used. If the newly detected error has the same address with errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in LR1_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_LRA_base> + 28_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.74 LR1_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error is detected when all ECC 1-bit error flags in LR1_SERSTR are set.

44.3.6.25 LR1_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in LR1_SERSTR register were detected.

This register is updated whenever LR1_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LRA_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.75 LR1_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in read data to PE3*1 loaded from the other PE's Local RAM speculatively.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in read data to PE2*1 loaded from the other PE's Local RAM speculatively.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in read data to PE1 loaded from the other PE's Local RAM speculatively.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in read data to PE0 loaded from the other PE's Local RAM speculatively.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.6.26 LR1_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location when the corresponding SEDFn flag is set.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LRA_base> + n × 4_H + 70_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	SEADR0[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.76 LR1_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 24	Reserved	When read, the value after reset is returned.
23 to 2	SEADR0[23:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.77 SEDL information

SEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE3*1 master.
2	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE2*1 master.
1	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE1 master.
0	Indicates an ECC 1-bit error is detected in loading data from the other PE's Local RAM by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.6.27 LR1_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_LRA_base> + $F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	DEADR0[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.78 LR1_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 24	Reserved	When read, the value after reset is returned.
23 to 2	DEADR0[23:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.79 DEDL Information

DEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE3*1 master.
2	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE2*1 master.
1	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE1 master.
0	Indicates a fatal error is detected in loading data from the other PE's Local RAM by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.6.28 Test Function

Through appropriate register settings, desired values can be written as RAM data and to the ECC bits. Also, data in the RAM and the ECC can all be read.

(1) Writing RAM data

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) Set the corresponding DATSEL bit in the ECC Test Control Register to 0 to select RAM data for access when writing.

(c) When data is written to the Local RAM, only the RAM data can be modified without updating the ECC bits.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(2) Reading RAM data

(a) Set ECCDIS bit in the ECC Control Register to 1 to disable ECC error detection and correction.

(b) Read the Local RAM. Since neither error detection nor correction proceeds when the Local RAM is read, the RAM data is read unchanged.

How to exit this test mode:

(a) Set ECCDIS bit in the ECC Control Register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) Set the corresponding DATSEL bit in the ECC Test Control Register to 1 to select the ECC bits for access when writing.

(c) When data is written to the Local RAM, only the ECC bits can be modified without updating the RAM data. At that time, bits[6:0] are written to the ECC bits.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(4) Reading the ECC bits

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) When the Local RAM is read, the ECC bits are stored in the word corresponding to local RAM ECC read buffer register.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(5) Self-diagnosis of the ECC check function

Desired values can be written to the RAM data, ECC bits by the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the Local RAM in normal mode and checking the result of error correction or detection.

(6) Self-diagnosis of the address feedback check function

Self-diagnosis is enabled by following procedure below.

(a) Set the desired bit(s) of AFINV[20:4] in the Address Feedback Test Control Register to 1 to set test mode.

(b) When the Local RAM is read or written, address feedback error occurs because feedback address XOR-ed with AFINV can be input to address feedback checker.

How to exit this test mode:

(a) Set AFINV[20:4] bits in the Address Feedback Test Control Register all to 0 to disable test mode (normal mode).

44.3.7 Cluster RAM ECC and Address Feedback

44.3.7.1 Overview

The Cluster RAM ECC is summarized in the table below.

The ECC protection described below takes into consideration all possible Cluster RAM accesses from PE0, PE1, PE2*1, PE3*1, H-Bus, System Bus and read access for read-modify-write processing unit of 8-bit and 16-bit data.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.80 Cluster RAM ECC (1/2)

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Address feedback	Address feedback check can be either enabled or disabled. When enabled, Address feedback error detection is carried out. When disabled, no error detection is carried out. In the initial state, this function is enabled.
Error notification	The occurrence of an ECC error or an address feedback error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is enabled upon detection of an ECC 1-bit error. **Address Feedback Error:** • Error notification can be either enabled or disabled upon detection of an address feedback error. In the initial state, error notification is enabled upon detection of an address feedback error. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. The error notification signal for both an address feedback error and an ECC 2-bit error is handled as one source in ECM. An ECC 1-bit error and an overflow error are handled as individual sources in ECM. An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.

Table 44.80 Cluster RAM ECC (2/2)

Item	Description
Address capture	The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the address that is captured. Address buffers are updated if the error status is cleared. Multi-stage address buffers are provided for a 1-bit data error. • ECC 1-bit error: Eight stages • ECC 2-bit error and address feedback error shared: One stage
Self-diagnosis	Desired values can be written to the RAM data and the ECC bits. The RAM data can be read directly and the ECC can be read via Read Buffer Register in ECC Test Mode. Moreover, the error injection to an address feedback checker is possible by setting AFINV.

ECC of Cluster RAM is provided for each 32 bits of data and each 32-bit data is called bank 0 to 3. The relationship between addresses and bank numbers are as follows.

Table 44.81 Relationship between addresses and bank numbers

4 lower-order bits of address (Hexadecimal Notation)	F_H to C_H	B_H to 8_H	7_H to 4_H	3_H to 0_H
Bank number	Bank 3	Bank 2	Bank 1	Bank 0

44.3.7.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.82 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_CRCCL0_base>	FFC4 $9C00_H$	P-Bus Group 0
<ECCCNT_CRCCL1_base>*1	FFC4 $9C80_H$	P-Bus Group 0
<ECCCNT_CRCCL2_base>*2	FFC4 $9D00_H$	P-Bus Group 0
<ECCCNT_CRCCL3_base>	FFC4 $9D80_H$	P-Bus Group 0
<ECCCNT_CRA_base>	FFC4 $9E00_H$	P-Bus Group 0
<MECCCAP_CRAM_base>	FFC5 2000_H	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.7.3 List of Registers

Table 44.83 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_CRCCLn (n = 0-3) *2	Address feedback control register	CRCAFCTL	<ECCCNT_CRCCLn_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	ECC test control register	CRCECCTSTCTL	<ECCCNT_CRCCLn_base> + 10_H	8, 16, 32	*1	ECCKCPROT
	Address feedback test control register	CRCAFINV	<ECCCNT_CRCCLn_base> + 20_H	8, 16, 32	*1	ECCKCPROT
	ECC read buffer register	CRCTDATBFECCF	<ECCCNT_CRCCLn_base> + 60_H	8, 16, 32	*1	—
ECCCNT_CRA	ECC control register	CRAECCCTL	<ECCCNT_CRA_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
MECCCAP_CRAM	Error notification control register	CR_ERRINT	<MECCCAP_CRAM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	CR_SSTCLR	<MECCCAP_CRAM_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	CR_DSTCLR	<MECCCAP_CRAM_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	CR_OVFCLR	<MECCCAP_CRAM_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	CR_SERSTR	<MECCCAP_CRAM_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	CR_DERSTR	<MECCCAP_CRAM_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	CR_OVFSTR	<MECCCAP_CRAM_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	CR_SERINF	<MECCCAP_CRAM_base> + 30_H	32	PBG00#2	—
	1st 1-bit error address register	CR_00SEADR	<MECCCAP_CRAM_base> + 70_H	32	PBG00#2	—
	2nd 1-bit error address register	CR_01SEADR	<MECCCAP_CRAM_base> + 74_H	32	PBG00#2	—
	3rd 1-bit error address register	CR_02SEADR	<MECCCAP_CRAM_base> + 78_H	32	PBG00#2	—
	4th 1-bit error address register	CR_03SEADR	<MECCCAP_CRAM_base> + $7C_H$	32	PBG00#2	—
	5th 1-bit error address register	CR_04SEADR	<MECCCAP_CRAM_base> + 80_H	32	PBG00#2	—
	6th 1-bit error address register	CR_05SEADR	<MECCCAP_CRAM_base> + 84_H	32	PBG00#2	—
	7th 1-bit error address register	CR_06SEADR	<MECCCAP_CRAM_base> + 88_H	32	PBG00#2	—
	8th 1-bit error address register	CR_07SEADR	<MECCCAP_CRAM_base> + $8C_H$	32	PBG00#2	—
	1st fatal error address register	CR_00DEADR	<MECCCAP_CRAM_base> + $F0_H$	32	PBG00#2	—

Note 1. n=0: PBG00#3
n=1: PBG00#4
n=2: PBG00#5
n=3: PBG00#6

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins) with index n = 1 and RH850/U2A6 with index n = 1, 2.

Table 44.84 The relation between the module storing the error and the error source for Cluster RAM

Protection target	Function	Controlled by	Captured to	Applicable for
Data from CRAM at CRAM I/F in PE0	Data ECC decode	ECCCNT_CRA	MECCCAP_CRAM	Load data from CRAM area requested by PE0 master
Data from CRAM at CRAM I/F in PE1				Load data from CRAM area requested by PE1 master
Data from CRAM at CRAM I/F in PE2*2				Load data from CRAM area requested by PE2 master
Data from CRAM at CRAM I/F in PE3*2				Load data from CRAM area requested by PE3 master
Data during RMW at CRAM controller 0				Read data from CRAM0 area in RMW*1
Data during RMW at CRAM controller 1*2				Read data from CRAM1 area in RMW*1
Data during RMW at CRAM controller 2*3				Read data from CRAM2 area in RMW*1
Data during RMW at CRAM controller 3				Read data from CRAM3 area in RMW*1
Data at SAXI2MBI0 (bridge from System Bus to CRAM Bus)				Read data from CRAM0 area requested by other than PEs
Data at SAXI2MBI0 (bridge from System Bus to CRAM Bus)				Instruction from CRAM0 area
Data at SAXI2MBI1 (bridge from System Bus to CRAM Bus)*2				Read data from CRAM1 area requested by other than PEs
Data at SAXI2MBI1 (bridge from System Bus to CRAM Bus)*2				Instruction from CRAM1 area
Data at SAXI2MBI2 (bridge from System Bus to CRAM Bus)*3				Read data from CRAM2 area requested by other than PEs
Data at SAXI2MBI2 (bridge from System Bus to CRAM Bus)*3				Instruction from CRAM2 area
Data at SAXI2MBI3 (bridge from System Bus to CRAM Bus)				Read data from CRAM3 area requested by other than PEs
Data at SAXI2MBI3 (bridge from System Bus to CRAM Bus)				Instruction from CRAM3 area
Address to CRAM at CRAM controller 0	Address feedback check	ECCCNT_CRCCL0		Request address to Cluster RAM0 area
Address to CRAM at CRAM controller 1*2		ECCCNT_CRCCL1		Request address to Cluster RAM1 area
Address to CRAM at CRAM controller 2*3		ECCCNT_CRCCL2		Request address to Cluster RAM2 area
Address to CRAM at CRAM controller 3		ECCCNT_CRCCL3		Request address to Cluster RAM3 area

Note 1. Read-modify-write includes SET1, NOT1, CLR1, CAXI and write access less than 4 bytes. For example, 2-byte write access is included.

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 3. This function is not implemented in RH850/U2A6.

44.3.7.4 CRCAFCTL — Address Feedback Control Register

This register controls address feedback error detection for the address that is requested to Cluster RAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CRCCLn_base> + 00_H (n = 0-3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEDIS	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R

Table 44.85 CRCAFCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.7.5 CRCECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), either data field or ECC field of Cluster RAM can be written separately.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CRCCLn_base> + 10_H (n = 0-3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.86 CRCECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects the write mode to Cluster RAM. 0: Normal mode 1: Test mode Either data field or ECC field can be written according to DATSEL setting in test mode.
0	DATSEL	Selects field to write data when ECCTST = 1. 0: Data field only 1: ECC field only

44.3.7.6 CRCAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from Cluster RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CRCCLn_base> + 20_H (n = 0-3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	AFINV[20:19]		AFINV [18]	AFINV [17]	AFINV [16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	*1	*1	*2	*2	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AFINV[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 44.87 CRCAFINV Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 19	AFINV[20:19]*1	Specifies bit pattern to inject errors into feedback address.
18	AFINV[18]*2	Specifies bit pattern to inject errors into feedback address.
17	AFINV[17]*2	Specifies bit pattern to inject errors into feedback address.
16 to 4	AFINV[16:4]	Specifies bit pattern to inject errors into feedback address.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. AFINV[20:19] is implemented only in ECCCNT_CRCCL2. AFINV[20:19] is reserved bit and read only for ECCCNT_CRCCL0 and ECCCNT_CRCCL1 and ECCCNT_CRCCL3 in RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8.
AFINV[20:19] is reserved bit in RH850/U2A6.

Note 2. AFINV[18] is implemented only in ECCCNT_CRCCL0 and ECCCNT_CRCCL1 and ECCCNT_CRCCL2, AFINV[18] is reserved bit and read only for ECCCNT_CRCCL3 in RH850/U2A-EVA and RH850/U2A16.
AFINV[18:17] is implemented only in ECCCNT_CRCCL0 and ECCCNT_CRCCL2, AFINV[18:17] is reserved bit and read only for ECCCNT_CRCCL3 in RH850/U2A8.
AFINV[18:17] is implemented only in ECCCNT_CRCCL0, AFINV[18:17] is reserved bit and read only for ECCCNT_CRCCL3 in RH850/U2A6.

44.3.7.7 CRCTDATBFECCF — ECC Read Buffer Register

This register is used to hold ECC field read from Cluster RAM for self-diagnosis. ECC field in the address that is requested can be stored in this register while reading from Cluster RAM where ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CRCCLn_base> + 60_H (n = 0-3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	BFECC3[6:0]							—	BFECC2[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BFECC1[6:0]							—	BFECC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.88 CRCTDATBFECCF Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 24	BFECC3[6:0]	ECC field read from requested address in Cluster RAM bank 3
23	Reserved	When read, the value after reset is returned.
22 to 16	BFECC2[6:0]	ECC field read from requested address in Cluster RAM bank 2
15	Reserved	When read, the value after reset is returned.
14 to 8	BFECC1[6:0]	ECC field read from requested address in Cluster RAM bank 1
7	Reserved	When read, the value after reset is returned.
6 to 0	BFECC0[6:0]	ECC field read from requested address in Cluster RAM bank 0

44.3.7.8 CRAECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from Cluster RAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_CRA_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.89 CRAECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.7.9 CR_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 00_H

Value after reset: $0000\ 008B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 44.90 CR_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error has occurred. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.7.10 CR_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in CR_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SST CLR07	SST CLR06	SST CLR05	SST CLR04	SST CLR03	SST CLR02	SST CLR01	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 44.91 CR_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset
7	SSTCLR07	Writing 1 to this bit clears SEDF07 in CR_SERSTR.
6	SSTCLR06	Writing 1 to this bit clears SEDF06 in CR_SERSTR.
5	SSTCLR05	Writing 1 to this bit clears SEDF05 in CR_SERSTR.
4	SSTCLR04	Writing 1 to this bit clears SEDF04 in CR_SERSTR.
3	SSTCLR03	Writing 1 to this bit clears SEDF03 in CR_SERSTR.
2	SSTCLR02	Writing 1 to this bit clears SEDF02 in CR_SERSTR.
1	SSTCLR01	Writing 1 to this bit clears SEDF01 in CR_SERSTR.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in CR_SERSTR.

44.3.7.11 CR_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in CR_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.92 CR_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in CR_DERSTR.

44.3.7.12 CR_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in CR_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.93 CR_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in CR_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in CR_OVFSTR.

44.3.7.13 CR_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in CR_nSEADR register when SEDFn flag is set. The SEDFn flag is set only when a unique ECC 1-bit error, which is different from previous errors stored in 1-bit error address registers, is detected while the SEDFn is “0”. If the newly detected error has the same address with errors already stored, this flag is not set.

This register can be cleared by SSTCLRn in CR_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEDF 07	SEDF 06	SEDF 05	SEDF 04	SEDF 03	SEDF 02	SEDF 01	SEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.94 CR_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	SEDF07	Indicates that an ECC 1-bit error was detected and error address is stored in CR_07SEADR register.
6	SEDF06	Indicates that an ECC 1-bit error was detected and error address is stored in CR_06SEADR register.
5	SEDF05	Indicates that an ECC 1-bit error was detected and error address is stored in CR_05SEADR register.
4	SEDF04	Indicates that an ECC 1-bit error was detected and error address is stored in CR_04SEADR register.
3	SEDF03	Indicates that an ECC 1-bit error was detected and error address is stored in CR_03SEADR register.
2	SEDF02	Indicates that an ECC 1-bit error was detected and error address is stored in CR_02SEADR register.
1	SEDF01	Indicates that an ECC 1-bit error was detected and error address is stored in CR_01SEADR register.
0	SEDF00	Indicates that an ECC 1-bit error was detected and error address is stored in CR_00SEADR register.

44.3.7.14 CR_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in CR_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in CR_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 24_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.95 CR_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	AFEF00	Indicates that an address feedback error was detected.
2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.7.15 CR_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in CR_OVFCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_CRAM_base> + 28_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.96 CR_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error is detected when all ECC 1-bit error flags in CR_SERSTR are set.

44.3.7.16 CR_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in CR_SERSTR register were detected.

This register is updated whenever CR_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CRAM_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SEDLINF23	SEDLINF22	SEDLINF21	SEDLINF20	SEDLINF19	SEDLINF18	SEDLINF17	SEDLINF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEDLINF15	SEDLINF14	SEDLINF13	SEDLINF12	SEDLINF11	SEDLINF10	SEDLINF09	SEDLINF08	—	—	—	—	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.97 CR_SERINF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23	SEDLINF23	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 3 during the RMW operation.
22	SEDLINF22	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 3 during the RMW operation.
21	SEDLINF21	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 2*2 during the RMW operation.
20	SEDLINF20	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 2*2 during the RMW operation.
19	SEDLINF19	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 1*1 during the RMW operation.
18	SEDLINF18	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 1*1 during the RMW operation.
17	SEDLINF17	Indicates that an ECC 1-bit error was detected in upper address of Cluster RAM for cluster 0 during the RMW operation.
16	SEDLINF16	Indicates that an ECC 1-bit error was detected in lower address of Cluster RAM for cluster 0 during the RMW operation.
15	SEDLINF15	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 3.
14	SEDLINF14	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 3.
13	SEDLINF13	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 2*2.
12	SEDLINF12	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 2*2.
11	SEDLINF11	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 1*1.
10	SEDLINF10	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 1*1.

Table 44.97 CR_SERINF Register Contents (2/2)

Bit Position	Bit Name	Function
9	SEDLINF09	Indicates that an ECC 1-bit error was detected in the data read from upper address of Cluster RAM to SAXI2MBI for cluster 0.
8	SEDLINF08	Indicates that an ECC 1-bit error was detected in the data read from lower address of Cluster RAM to SAXI2MBI for cluster 0.
7 to 4	Reserved	When read, the value after reset is returned.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE3*1.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE2*1.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE1.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in the data read from Cluster RAM to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.7.17 CR_nSEADR — n-th 1-bit Error Address Register (n = 00 to 07)

This register is used to hold the address and the location of the error when the corresponding SEDFn flag is set.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CRAM_base> + n × 4_H + 70_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	SEADR0[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.98 CR_nSEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 24	Reserved	When read, the value after reset is returned.
23 to 2	SEADR0[23:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.99 SEDL information (1/2)

SEDL[4:0]	Indicated information
31 to 24	Reserved
23	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 3: Storing sub-word data by any master. CAXI operation by any PE master.
22	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 3: Storing sub-word data by any master. CAXI operation by any PE master.
21	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 2*2: Storing sub-word data by any master. CAXI operation by any PE master.

Table 44.99 SEDL information (2/2)

SEDL[4:0]	Indicated information
20	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 2*2: Storing sub-word data by any master. CAXI operation by any PE master.
19	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1*1: Storing sub-word data by any master. CAXI operation by any PE master.
18	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1*1: Storing sub-word data by any master. CAXI operation by any PE master.
17	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
16	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
15	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 3: Loading data by any master except PEs. Fetching instructions by any PE master.
14	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 3: Loading data by any master except PEs. Fetching instructions by any PE master.
13	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 2*2: Loading data by any master except PEs. Fetching instructions by any PE master.
12	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 2*2: Loading data by any master except PEs. Fetching instructions by any PE master.
11	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1*1: Loading data by any master except PEs. Fetching instructions by any PE master.
10	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1*1: Loading data by any master except PEs. Fetching instructions by any PE master.
9	Indicates an ECC 1-bit error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
8	Indicates an ECC 1-bit error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
7 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE3*1 master.
2	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE2*1 master.
1	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE1 master.
0	Indicates an ECC 1-bit error is detected in loading data from Cluster RAMs by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A6.

44.3.7.18 CR_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error, and/or ECC 2-bit error was detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_CRAM_base> + $F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	DEADR0[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DEADR0[15:2]														—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.100 CR_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 24	Reserved	When read, the value after reset is returned.
23 to 2	DEADR0[23:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.101 DEDL Information (1/2)

DEDL[4:0]	Indicated information
31 to 24	Reserved
23	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 3: Storing sub-word data by any master. CAXI operation by any PE master.
22	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 3: Storing sub-word data by any master. CAXI operation by any PE master.
21	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 2*2: Storing sub-word data by any master. CAXI operation by any PE master.
20	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 2*2: Storing sub-word data by any master. CAXI operation by any PE master.

Table 44.101 DEDL Information (2/2)

DEDL[4:0]	Indicated information
19	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1*1: Storing sub-word data by any master. CAXI operation by any PE master.
18	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1*1: Storing sub-word data by any master. CAXI operation by any PE master.
17	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
16	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Storing sub-word data by any master. CAXI operation by any PE master.
15	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 3: Loading data by any master except PEs. Fetching instructions by any PE master.
14	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 3: Loading data by any master except PEs. Fetching instructions by any PE master.
13	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 2*2: Loading data by any master except PEs. Fetching instructions by any PE master.
12	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 2*2: Loading data by any master except PEs. Fetching instructions by any PE master.
11	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 1*1: Loading data by any master except PEs. Fetching instructions by any PE master.
10	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 1*1: Loading data by any master except PEs. Fetching instructions by any PE master.
9	Indicates a fatal error is detected in the following operation to bank 1 or 3 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
8	Indicates a fatal error is detected in the following operation to bank 0 or 2 of Cluster RAM in cluster 0: Loading data by any master except PEs. Fetching instructions by any PE master.
7 to 4	Reserved
3	Indicates a fatal error is detected in loading data from Cluster RAMs by PE3*1 master.
2	Indicates a fatal error is detected in loading data from Cluster RAMs by PE2*1 master.
1	Indicates a fatal error is detected in loading data from Cluster RAMs by PE1 master.
0	Indicates a fatal error is detected in loading data from Cluster RAMs by PE0 master.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.7.19 Test Function

(1) Writing RAM data

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) Set the corresponding DATSEL bit in the ECC Test Control Register to 0 to select RAM data for access when writing.

(c) When data is written to the cluster RAM, the RAM data is modified without updating the ECC bits.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(2) Reading RAM data

(a) Set ECCDIS bit in the ECC Control Register to 1 to disable ECC error detection and correction.

(b) Read the Cluster RAM. Since neither error detection nor correction proceeds when the Cluster RAM is read, the RAM data is read unchanged.

How to exit this test mode:

(a) Set ECCDIS bit in the ECC Control Register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC bits

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) Set the corresponding DATSEL bit in the ECC Test Control Register to 1 to select the ECC bits for access when writing.

(c) When data is written to the Cluster RAM, the ECC bits are modified without updating the RAM data. At that time, bits[6:0] are respectively written to the ECC bits.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(4) Reading the ECC bits

(a) Set ECCTST bit in the ECC Test Control Register to 1 to set test mode.

(b) When the Cluster RAM is read, the ECC bits are stored in the word corresponding to cluster RAM ECC read buffer register.

How to exit this test mode:

(a) Set ECCTST bit in the ECC Test Control Register to 0 to disable test mode (normal mode).

(5) Self-diagnosis of the ECC check function

Desired values can be written to the RAM data, ECC bits by using the procedure described in (1) or (3) above. Therefore, a fault can be injected by, for example, inverting appropriate bits of the RAM data and/or ECC bits. After that, self-diagnosis of the ECC decoder is possible by reading the Cluster RAM in normal mode and checking the result of error correction or detection.

(6) Self-diagnosis of the address feedback check function

Self-diagnosis is enabled by following procedure below.

(a) Set the desired bit(s) of AFINV[20:4] in the Address Feedback Test Control Register to 1 to set test mode.

(b) When the Cluster RAM is read or written, address feedback error occurs because feedback address XOR-ed with AFINV can be input to address feedback checker.

How to exit this test mode:

(a) Set AFINV[20:4] bits in the Address Feedback Test Control Register all to 0 to disable test mode (normal mode).

Note: AFINV[20:19] is implemented only in ECCCNT_CRCCL2. AFINV[20:19] is reserved bit and read only for ECCCNT_CRCCL0 and ECCCNT_CRCCL1 and ECCCNT_CRCCL3.

Note: AFINV[18] is implemented only in ECCCNT_CRCCL0 and ECCCNT_CRCCL1 and ECCCNT_CRCCL2. AFINV[18] is reserved bit and read only for ECCCNT_CRCCL3

44.3.8 Instruction Cache EDC and Address Feedback

44.3.8.1 Overview

The instruction cache EDC is summarized in the table below.

Table 44.102 Instruction Cache EDC

Item	Description
ECC error detection	ECC error detection can be either enabled or disabled. When enabled, the following settings can be selected: • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, error detection is not carried out. In the initial state, the ECC function is enabled; 1-bit error detection, and 2-bit error detection are carried out.
Address feedback	Address feedback check can be either enabled or disabled. When enabled, Address feedback error detection is carried out. When disabled, no error detection is carried out. In the initial state, this function is enabled.
Error notification	The occurrence of an ECC error or an address feedback error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of both an ECC 2-bit error and an ECC 1-bit error is disabled. **Address Feedback Error:** • Error notification can be either enabled or disabled upon detection of an address feedback error. In the initial state, error notification is disabled upon detection of an address feedback error. The error notification signal is output with an ECC 2-bit error and an ECC 1-bit error and an address feedback error handled as one source in ECM. An ECC error signal is only issued to the ECM if the ECC error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error, an ECC 1-bit error or an address feedback error is detected. The error status serves as the enable bit of the captured address. If an ECC error occurs while no error status is set, the address at which the associated error occurred is captured.
Self-diagnosis	A cache instruction is used to write the desired values as RAM data and to the ECC bits. Similarly, data in the RAM and the ECC bits can be read directly. Since cache instructions go through the same encoding or decoding path as a normal cache fill or instruction fetch, errors can be both inserted and confirmed simply by using a cache instruction. Errors can also be injected into an address feedback checker by setting AFINV.

44.3.8.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 44.103 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_IT_PE0CL0_base>	FFC4 9000_H	P-Bus Group 0
<ECCCNT_IT_PE1CL0_base>	FFC4 9080_H	P-Bus Group 0
<ECCCNT_IT_PE2CL1_base>*1	FFC4 9100_H	P-Bus Group 0
<ECCCNT_IT_PE3CL1_base>*1	FFC4 9180_H	P-Bus Group 0
<ECCCNT_ID_PE0CL0_base>	FFC4 9400_H	P-Bus Group 0
<ECCCNT_ID_PE1CL0_base>	FFC4 9480_H	P-Bus Group 0
<ECCCNT_ID_PE2CL1_base>*1	FFC4 9500_H	P-Bus Group 0
<ECCCNT_ID_PE3CL1_base>*1	FFC4 9580_H	P-Bus Group 0
<MECCCAP_IT_PE0CL0_base>	FFC5 1000_H	P-Bus Group 0
<MECCCAP_IT_PE1CL0_base>	FFC5 1100_H	P-Bus Group 0
<MECCCAP_IT_PE2CL1_base>*1	FFC5 1200_H	P-Bus Group 0
<MECCCAP_IT_PE3CL1_base>*1	FFC5 1300_H	P-Bus Group 0
<MECCCAP_ID_PE0CL0_base>	FFC5 1800_H	P-Bus Group 0
<MECCCAP_ID_PE1CL0_base>	FFC5 1900_H	P-Bus Group 0
<MECCCAP_ID_PE2CL1_base>*1	FFC5 $1A00_H$	P-Bus Group 0
<MECCCAP_ID_PE3CL1_base>*1	FFC5 $1B00_H$	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.8.3 List of Registers

Table 44.104 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_IT_PEnCL0 (n = 0, 1)	ECC control register	ITECCCTL	<ECCCNT_IT_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	Address feedback test control register	ITAFINV	<ECCCNT_IT_PEnCL0_base> + 20_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_IT_PEnCL1 (n = 2, 3)*3	ECC control register	ITECCCTL	<ECCCNT_IT_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	Address feedback test control register	ITAFINV	<ECCCNT_IT_PEnCL1_base> + 20_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_ID_PEnCL0 (n = 0, 1)	ECC control register	IDECCCTL	<ECCCNT_ID_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	Address feedback test control register	IDAFINV	<ECCCNT_ID_PEnCL0_base> + 20_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_ID_PEnCL1 (n = 2, 3)*3	ECC control register	IDECCCTL	<ECCCNT_ID_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	Address feedback test control register	IDAFINV	<ECCCNT_ID_PEnCL1_base> + 20_H	8, 16, 32	*2	ECCKCPROT
MECCCAP_IT_PEnCL0 (n = 0, 1)	Error notification control register	IT_ERRINT	<MECCCAP_IT_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	1-bit error status clear register	IT_SSTCLR	<MECCCAP_IT_PEnCL0_base> + 10_H	8, 16, 32	*1	—
	Fatal error status clear register	IT_DSTCLR	<MECCCAP_IT_PEnCL0_base> + 14_H	8, 16, 32	*1	—
	1-bit error overflow status clear register	IT_OVFCLR	<MECCCAP_IT_PEnCL0_base> + 18_H	8, 16, 32	*1	—
	1-bit error status register	IT_SERSTR	<MECCCAP_IT_PEnCL0_base> + 20_H	8, 16, 32	*1	—
	Fatal error status register	IT_DERSTR	<MECCCAP_IT_PEnCL0_base> + 24_H	8, 16, 32	*1	—
	1-bit error overflow status register	IT_OVFSTR	<MECCCAP_IT_PEnCL0_base> + 28_H	8, 16, 32	*1	—
	1-bit error location information register	IT_SERINF	<MECCCAP_IT_PEnCL0_base> + 30_H	32	*1	—
	1st 1-bit error address register	IT_00SEADR	<MECCCAP_IT_PEnCL0_base> + 70_H	32	*1	—
	1st fatal error address register	IT_00DEADR	<MECCCAP_IT_PEnCL0_base> + $F0_H$	32	*1	—
MECCCAP_IT_PEnCL1 (n = 2, 3)*3	Error notification control register	IT_ERRINT	<MECCCAP_IT_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	1-bit error status clear register	IT_SSTCLR	<MECCCAP_IT_PEnCL1_base> + 10_H	8, 16, 32	*2	—
	Fatal error status clear register	IT_DSTCLR	<MECCCAP_IT_PEnCL1_base> + 14_H	8, 16, 32	*2	—
	1-bit error overflow status clear register	IT_OVFCLR	<MECCCAP_IT_PEnCL1_base> + 18_H	8, 16, 32	*2	—
	1-bit error status register	IT_SERSTR	<MECCCAP_IT_PEnCL1_base> + 20_H	8, 16, 32	*2	—
	Fatal error status register	IT_DERSTR	<MECCCAP_IT_PEnCL1_base> + 24_H	8, 16, 32	*2	—
	1-bit error overflow status register	IT_OVFSTR	<MECCCAP_IT_PEnCL1_base> + 28_H	8, 16, 32	*2	—
	1-bit error location information register	IT_SERINF	<MECCCAP_IT_PEnCL1_base> + 30_H	32	*2	—
	1st 1-bit error address register	IT_00SEADR	<MECCCAP_IT_PEnCL1_base> + 70_H	32	*2	—
	1st fatal error address register	IT_00DEADR	<MECCCAP_IT_PEnCL1_base> + $F0_H$	32	*2	—

Table 44.104 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MECCCAP_ID_PEnCL0 (n = 0, 1)	Error notification control register	ID_ERRINT	<MECCCAP_ID_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	1-bit error status clear register	ID_SSTCLR	<MECCCAP_ID_PEnCL0_base> + 10_H	8, 16, 32	*1	—
	Fatal error status clear register	ID_DSTCLR	<MECCCAP_ID_PEnCL0_base> + 14_H	8, 16, 32	*1	—
	1-bit error overflow status clear register	ID_OVFCLR	<MECCCAP_ID_PEnCL0_base> + 18_H	8, 16, 32	*1	—
	1-bit error status register	ID_SERSTR	<MECCCAP_ID_PEnCL0_base> + 20_H	8, 16, 32	*1	—
	Fatal error status register	ID_DERSTR	<MECCCAP_ID_PEnCL0_base> + 24_H	8, 16, 32	*1	—
	1-bit error overflow status register	ID_OVFSTR	<MECCCAP_ID_PEnCL0_base> + 28_H	8, 16, 32	*1	—
	1-bit error location information register	ID_SERINF	<MECCCAP_ID_PEnCL0_base> + 30_H	32	*1	—
	1st 1-bit error address register	ID_00SEADR	<MECCCAP_ID_PEnCL0_base> + 70_H	32	*1	—
	1st fatal error address register	ID_00DEADR	<MECCCAP_ID_PEnCL0_base> + $F0_H$	32	*1	—
MECCCAP_ID_PEnCL1 (n = 2, 3) *3	Error notification control register	ID_ERRINT	<MECCCAP_ID_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	1-bit error status clear register	ID_SSTCLR	<MECCCAP_ID_PEnCL1_base> + 10_H	8, 16, 32	*2	—
	Fatal error status clear register	ID_DSTCLR	<MECCCAP_ID_PEnCL1_base> + 14_H	8, 16, 32	*2	—
	1-bit error overflow status clear register	ID_OVFCLR	<MECCCAP_ID_PEnCL1_base> + 18_H	8, 16, 32	*2	—
	1-bit error status register	ID_SERSTR	<MECCCAP_ID_PEnCL1_base> + 20_H	8, 16, 32	*2	—
	Fatal error status register	ID_DERSTR	<MECCCAP_ID_PEnCL1_base> + 24_H	8, 16, 32	*2	—
	1-bit error overflow status register	ID_OVFSTR	<MECCCAP_ID_PEnCL1_base> + 28_H	8, 16, 32	*2	—
	1-bit error location information register	ID_SERINF	<MECCCAP_ID_PEnCL1_base> + 30_H	32	*2	—
	1st 1-bit error address register	ID_00SEADR	<MECCCAP_ID_PEnCL1_base> + 70_H	32	*2	—
	1st fatal error address register	ID_00DEADR	<MECCCAP_ID_PEnCL1_base> + $F0_H$	32	*2	—

Note 1. n=0: PBG01#0
n=1: PBG01#1

Note 2. n=2: PBG01#2
n=3: PBG01#3

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.105 The relation between the module storing the error and the error source for Instruction Cache

Protection target	Function	Controlled by	Captured to	Applicable for
Data from Cache tag RAM at Instruction Cache I/F in PE0	Data EDC decode	ECCCNT_IT_PE0CL0	MECCCAP_IT_PE0CL0	Read data from Instruction cache tag RAM of PE0
Data from Cache tag RAM at Instruction Cache I/F in PE1		ECCCNT_IT_PE1CL0	MECCCAP_IT_PE1CL0	Read data from Instruction cache tag RAM of PE1
Data from Cache tag RAM at Instruction Cache I/F in PE2*1		ECCCNT_IT_PE2CL1	MECCCAP_IT_PE2CL1	Read data from Instruction cache tag RAM of PE2
Data from Cache tag RAM at Instruction Cache I/F in PE3*1		ECCCNT_IT_PE3CL1	MECCCAP_IT_PE3CL1	Read data from Instruction cache tag RAM of PE3
Address to Cache tag RAM at Instruction Cache I/F in PE0	Address feedback check	ECCCNT_IT_PE0CL0	MECCCAP_IT_PE0CL0	Request address to Instruction cache tag RAM of PE0
Address to Cache tag RAM at Instruction Cache I/F in PE1		ECCCNT_IT_PE1CL0	MECCCAP_IT_PE1CL0	Request address to Instruction cache tag RAM of PE1
Address to Cache tag RAM at Instruction Cache I/F in PE2*1		ECCCNT_IT_PE2CL1	MECCCAP_IT_PE2CL1	Request address to Instruction cache tag RAM of PE2
Address to Cache tag RAM at Instruction Cache I/F in PE3*1		ECCCNT_IT_PE3CL1	MECCCAP_IT_PE3CL1	Request address to Instruction cache tag RAM of PE3
Data from Cache data RAM at Instruction Cache I/F in PE0	Data EDC decode	ECCCNT_ID_PE0CL0	MECCCAP_ID_PE0CL0	Read data from Instruction cache data RAM of PE0
Data from Cache data RAM at Instruction Cache I/F in PE1		ECCCNT_ID_PE1CL0	MECCCAP_ID_PE1CL0	Read data from Instruction cache data RAM of PE1
Data from Cache data RAM at Instruction Cache I/F in PE2*1		ECCCNT_ID_PE2CL1	MECCCAP_ID_PE2CL1	Read data from Instruction cache data RAM of PE2
Data from Cache data RAM at Instruction Cache I/F in PE3*1		ECCCNT_ID_PE3CL1	MECCCAP_ID_PE3CL1	Read data from Instruction cache data RAM of PE3
Address to Cache data RAM at Instruction Cache I/F in PE0	Address feedback check	ECCCNT_ID_PE0CL0	MECCCAP_ID_PE0CL0	Request address to Instruction cache data RAM of PE0
Address to Cache data RAM at Instruction Cache I/F in PE1		ECCCNT_ID_PE1CL0	MECCCAP_ID_PE1CL0	Request address to Instruction cache data RAM of PE1
Address to Cache data RAM at Instruction Cache I/F in PE2*1		ECCCNT_ID_PE2CL1	MECCCAP_ID_PE2CL1	Request address to Instruction cache data RAM of PE2
Address to Cache data RAM at Instruction Cache I/F in PE3*1		ECCCNT_ID_PE3CL1	MECCCAP_ID_PE3CL1	Request address to Instruction cache data RAM of PE3

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.8.4 ITECCCTL — ECC Control Register

This register controls ECC error detection for data read from cache tag RAM and address feedback error detection for the address that is requested to cache tag RAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_IT_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_IT_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 44.106 ITECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC disable bit Sets the ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

44.3.8.5 ITAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from cache tag RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_IT_PEnCL0_base> + 20_H (n = 0, 1)
<ECCCNT_IT_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	AFINV[12:5]								—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

Table 44.107 ITAFINV Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 5	AFINV[12:5]	Specifies bit pattern to inject errors into feedback address.
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.8.6 IDECCCTL — ECC Control Register

This register controls ECC error detection for data read from cache data RAM and address feedback error detection for the address that is requested to the cache data RAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_ID_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_ID_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W

Table 44.108 IDECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC disable bit Sets the ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

44.3.8.7 IDAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from the cache data RAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_ID_PEnCL0_base> + 20_H (n = 0, 1)
<ECCCNT_ID_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	AFINV[12:3]										—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.109 IDAFINV Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 3	AFINV[12:3]	Specifies bit pattern to inject errors into feedback address.
2 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.8.8 IT_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM when address feedback error and/or ECC error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 00_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 44.110 IT_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.8.9 IT_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in IT_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 10_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 10_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.111 IT_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in IT_SERSTR.

44.3.8.10 IT_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in IT_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 14_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 14_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.112 IT_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in IT_DERSTR.

44.3.8.11 IT_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in IT_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 18_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 18_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.113 IT_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in IT_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in IT_OVFSTR.

44.3.8.12 IT_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in IT_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in IT_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 20_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.114 IT_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDF00	Indicates that an ECC 1-bit error was detected.

44.3.8.13 IT_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in IT_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in IT_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 24_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 24_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.115 IT_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	AFEF00	Indicates that an address feedback error was detected.
2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.8.14 IT_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in IT_OVFCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 28_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 28_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.116 IT_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

44.3.8.15 IT_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in IT_SERSTR register were detected.

This is up-to-date whenever IT_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 30_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 30_H (n = 2, 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.117 IT_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in way 3 of cache tag RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in way 2 of cache tag RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in way 1 of cache tag RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in way 0 of cache tag RAM.

44.3.8.16 IT_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + 70_H (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + 70_H (n = 2, 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SEADR0[12:5]								—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.118 IT_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 13	Reserved	When read, the value after reset is returned.
12 to 5	SEADR0[12:5]	Indicates at which address this error was detected.
4 to 0	Reserved	When read, the value after reset is returned.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.119 SEDL information

SEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 3 of cache tag RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 2 of cache tag RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 1 of cache tag RAM.
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 0 of cache tag RAM.

44.3.8.17 IT_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_IT_PEnCL0_base> + $F0_H$ (n = 0, 1)
<MECCCAP_IT_PEnCL1_base> + $F0_H$ (n = 2, 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DEADR0[12:5]								—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.120 IT_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 13	Reserved	When read, the value after reset is returned.
12 to 5	DEADR0[12:5]	Indicates at which address this error was detected.
4 to 0	Reserved	When read, the value after reset is returned.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.121 DEDL Information

DEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates a fatal error is detected in storing data to or loading data from way 3 of cache tag RAM.
2	Indicates a fatal error is detected in storing data to or loading data from way 2 of cache tag RAM.
1	Indicates a fatal error is detected in storing data to or loading data from way 1 of cache tag RAM.
0	Indicates a fatal error is detected in storing data to or loading data from way 0 of cache tag RAM.

44.3.8.18 ID_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM when address feedback error and/or ECC error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 00_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 00_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 44.122 ID_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.8.19 ID_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in ID_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 10_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 10_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.123 ID_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in ID_SERSTR.

44.3.8.20 ID_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in ID_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 14_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 14_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.124 ID_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in ID_DERSTR.

44.3.8.21 ID_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in ID_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 18_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 18_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.125 ID_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in ID_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in ID_OVFSTR.

44.3.8.22 ID_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in ID_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in ID_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 20_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 20_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.126 ID_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDF00	Indicates an ECC 1-bit error is detected.

44.3.8.23 ID_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in ID_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in ID_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 24_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 24_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.127 ID_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	AFEF00	Indicates that address feedback error was detected.
2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that an ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.8.24 ID_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in ID_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 28_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 28_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.128 ID_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

44.3.8.25 ID_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in ID_SERSTR register were detected.

This is up-to-date whenever ID_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 30_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 30_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF03	SEDLINF02	SEDLINF01	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.129 ID_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	SEDLINF03	Indicates that an ECC 1-bit error was detected in way 3 of the cache data RAM.
2	SEDLINF02	Indicates that an ECC 1-bit error was detected in way 2 of the cache data RAM.
1	SEDLINF01	Indicates that an ECC 1-bit error was detected in way 1 of the cache data RAM.
0	SEDLINF00	Indicates that an ECC 1-bit error was detected in way 0 of the cache data RAM.

44.3.8.26 ID_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + 70_H (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + 70_H (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SEADR0[12:3]										—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.130 ID_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 13	Reserved	When read, the value after reset is returned.
12 to 3	SEADR0[12:3]	Indicates at which address this error was detected.
2 to 0	Reserved	When read, the value after reset is returned.

Refer to the following table to confirm the detailed information which each SEDL indicates.

Table 44.131 SEDL information

SEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 3 of cache data RAM.
2	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 2 of cache data RAM.
1	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 1 of cache data RAM.
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from way 0 of cache data RAM.

44.3.8.27 ID_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_ID_PEnCL0_base> + $F0_H$ (n = 0, 1)
<MECCCAP_ID_PEnCL1_base> + $F0_H$ (n = 2, 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	DEADR0[12:3]										—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.132 ID_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 13	Reserved	When read, the value after reset is returned.
12 to 3	DEADR0[12:3]	Indicates at which address this error was detected.
2 to 0	Reserved	When read, the value after reset is returned.

Refer to the following table to confirm the detailed information which each DEDL indicates.

Table 44.133 DEDL Information

DEDL[4:0]	Indicated information
31 to 4	Reserved
3	Indicates a fatal error is detected in storing data to or loading data from way 3 of cache data RAM.
2	Indicates a fatal error is detected in storing data to or loading data from way 2 of cache data RAM.
1	Indicates a fatal error is detected in storing data to or loading data from way 1 of cache data RAM.
0	Indicates a fatal error is detected in storing data to or loading data from way 0 of cache data RAM.

44.3.8.28 Test Function

A cache instruction is used to write the desired values as RAM data and to the ECC bits, and read data in the RAM and the ECC bits directly.

Since cache instructions go through the same encoding or decoding path as a normal cache fill or instruction fetch, errors can be both inserted and confirmed simply by using a cache instruction.

For details, refer to the *RH850G4MH User's Manual: Software*.

44.3.9 sDMAC/DTSRAM ECC and Address Feedback

44.3.9.1 Overview

sDMAC RAM ECC is summarized in the table below.

Table 44.134 sDMAC RAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected: • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Error notification	The occurrence of an ECC error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of both a 2-bit error and a 1-bit error is enabled. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. An ECC 2-bit error, an ECC 1-bit error and an overflow error are handled as individual sources. An ECC 1-bit error signal is only issued to the ECM if the ECC 1-bit error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the captured address. ECC 1-bit error: One stage ECC 2-bit error: One stage
Self-diagnosis	Desired values can be written to the RAM data and the ECC bits in sDMAC RAM in ECC Test Mode. The RAM data and the ECC bits in sDMAC RAM can be also read via Read Buffer Registers in this mode.

DTSRAM ECC is summarized in the table below.

Table 44.135 DTSRAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection, 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Address feedback	Address feedback check can be either enabled or disabled. When enabled, Address feedback error detection is carried out. When disabled, no error detection is carried out. In the initial state, this function is enabled.
Error notification	The occurrence of an ECC error or an address feedback error is reported to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of both a 2-bit error and a 1-bit error is enabled. **Address Feedback Error:** • Error notification can be either enabled or disabled upon detection of an address feedback error. In the initial state, error notification is enabled upon detection of an address feedback error. **Overflow Error:** • Error notification can be either enabled or disabled upon detection of an address buffer overflow error for ECC 1-bit error. In the initial state, error notification is enabled upon detection of an address buffer overflow error. The error notification signal is output with an ECC 2-bit error and an address feedback error handled as one source, and an ECC 1-bit error and an overflow error are handled as individual sources. An ECC 1-bit error signal is only issued to the ECM, if the ECC 1-bit error address is not yet stored in the error address buffer.
Error status	A status register is provided that indicates the statuses of ECC 2-bit error detection, ECC 1-bit error detection, and address feedback error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error, an ECC 1-bit error, or an address feedback error is detected. The error status serves as the enable bit of the capture address. ECC 1-bit error: One stage ECC 2-bit error and address feedback error: One stage
Self-diagnosis	Desired values can be written to the RAM data and the ECC bits in DTSRAM in ECC Test Mode. The RAM data and the ECC bits in DTSRAM can be also read via Read Buffer Registers in this mode. Moreover, the error injection to an address feedback checker is possible by setting AFINV.

44.3.9.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offset from the base addresses.

Table 44.136 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECCCNT_DTS_base>	FFC4 A000$_H$	P-Bus Group 0
<ECCCNT_DMDE0_base>	FFC4 A400$_H$	P-Bus Group 0
<ECCCNT_DMDA0_base>	FFC4 A480$_H$	P-Bus Group 0
<ECCCNT_DMDE1_base>	FFC4 A500$_H$	P-Bus Group 0
<ECCCNT_DMDA1_base>	FFC4 A580$_H$	P-Bus Group 0
<MECCCAP_DTS_base>	FFC5 2600$_H$	P-Bus Group 0
<MECCCAP_DMDE0_base>	FFC5 2800$_H$	P-Bus Group 0
<MECCCAP_DMDA0_base>	FFC5 2900$_H$	P-Bus Group 0
<MECCCAP_DMDE1_base>	FFC5 2A00$_H$	P-Bus Group 0
<MECCCAP_DMDA1_base>	FFC5 2B00$_H$	P-Bus Group 0

44.3.9.3 List of Registers

Table 44.137 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_DTS	ECC control register	DRECCCTL	<ECCCNT_DTS_base> + 00$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	DRECCTSTCTL	<ECCCNT_DTS_base> + 10$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Address feedback test control register	DRAFINV	<ECCCNT_DTS_base> + 20$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	DRECCTSTDIN0	<ECCCNT_DTS_base> + 30$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	DRECCTSTEIN	<ECCCNT_DTS_base> + 40$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Data read buffer register	DRTDATBFDATAF0	<ECCCNT_DTS_base> + 50$_H$	32	PBG00#2	—
	ECC read buffer register	DRTDATBFECCF	<ECCCNT_DTS_base> + 60$_H$	8, 16, 32	PBG00#2	—
ECCCNT_DMDEn (n = 0, 1)	ECC control register	DEECCCTL	<ECCCNT_DMDEn_base> + 00$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	DEECCTSTCTL	<ECCCNT_DMDEn_base> + 10$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	DEECCTSTDIN0	<ECCCNT_DMDEn_base> + 30$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	DEECCTSTEIN	<ECCCNT_DMDEn_base> + 40$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Data read buffer register	DETDATBFDATAF0	<ECCCNT_DMDEn_base> + 50$_H$	32	PBG00#2	—
	ECC read buffer register	DETDATBFECCF	<ECCCNT_DMDEn_base> + 60$_H$	8, 16, 32	PBG00#2	—
ECCCNT_DMDAn (n = 0, 1)	ECC control register	DAECCCTL	<ECCCNT_DMDAn_base> + 00$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	DAECCTSTCTL	<ECCCNT_DMDAn_base> + 10$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test lower data input register	DAECCTSTDIN0	<ECCCNT_DMDAn_base> + 30$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test upper data input register	DAECCTSTDIN1	<ECCCNT_DMDAn_base> + 34$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	DAECCTSTEIN	<ECCCNT_DMDAn_base> + 40$_H$	8, 16, 32	PBG00#2	ECCKCPROT
	Lower data read buffer register	DATDATBFDATAF0	<ECCCNT_DMDAn_base> + 50$_H$	32	PBG00#2	—
	Upper data read buffer register	DATDATBFDATAF1	<ECCCNT_DMDAn_base> + 54$_H$	32	PBG00#2	—
	ECC read buffer register	DATDATBFECCF	<ECCCNT_DMDAn_base> + 60$_H$	8, 16, 32	PBG00#2	—

Table 44.137 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
MECCCAP_DTS	Error notification control register	DR_ERRINT	<MECCCAP_DTS_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	DR_SSTCLR	<MECCCAP_DTS_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	DR_DSTCLR	<MECCCAP_DTS_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	DR_OVFCLR	<MECCCAP_DTS_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	DR_SERSTR	<MECCCAP_DTS_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	DR_DERSTR	<MECCCAP_DTS_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	DR_OVFSTR	<MECCCAP_DTS_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	DR_SERINF	<MECCCAP_DTS_base> + 30_H	32	PBG00#2	—
	1st 1-bit error address register	DR_00SEADR	<MECCCAP_DTS_base> + 70_H	32	PBG00#2	—
	1st fatal error address register	DR_00DEADR	<MECCCAP_DTS_base> + $F0_H$	32	PBG00#2	—
MECCCAP_DMDEn (n = 0, 1)	Error notification control register	DE_ERRINT	<MECCCAP_DMDEn_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	DE_SSTCLR	<MECCCAP_DMDEn_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	DE_DSTCLR	<MECCCAP_DMDEn_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	DE_OVFCLR	<MECCCAP_DMDEn_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	DE_SERSTR	<MECCCAP_DMDEn_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	DE_DERSTR	<MECCCAP_DMDEn_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	DE_OVFSTR	<MECCCAP_DMDEn_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	DE_SERINF	<MECCCAP_DMDEn_base> + 30_H	32	PBG00#2	—
	1st 1-bit error address register	DE_00SEADR	<MECCCAP_DMDEn_base> + 70_H	32	PBG00#2	—
	1st fatal error address register	DE_00DEADR	<MECCCAP_DMDEn_base> + $F0_H$	32	PBG00#2	—
MECCCAP_DMDAn (n = 0, 1)	Error notification control register	DA_ERRINT	<MECCCAP_DMDAn_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	1-bit error status clear register	DA_SSTCLR	<MECCCAP_DMDAn_base> + 10_H	8, 16, 32	PBG00#2	—
	Fatal error status clear register	DA_DSTCLR	<MECCCAP_DMDAn_base> + 14_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status clear register	DA_OVFCLR	<MECCCAP_DMDAn_base> + 18_H	8, 16, 32	PBG00#2	—
	1-bit error status register	DA_SERSTR	<MECCCAP_DMDAn_base> + 20_H	8, 16, 32	PBG00#2	—
	Fatal error status register	DA_DERSTR	<MECCCAP_DMDAn_base> + 24_H	8, 16, 32	PBG00#2	—
	1-bit error overflow status register	DA_OVFSTR	<MECCCAP_DMDAn_base> + 28_H	8, 16, 32	PBG00#2	—
	1-bit error location information register	DA_SERINF	<MECCCAP_DMDAn_base> + 30_H	32	PBG00#2	—
	1st 1-bit error address register	DA_00SEADR	<MECCCAP_DMDAn_base> + 70_H	32	PBG00#2	—
	1st fatal error address register	DA_00DEADR	<MECCCAP_DMDAn_base> + $F0_H$	32	PBG00#2	—

Note: The initial characters of DTS/sDMAC register symbols indicates the followings.
DR: DTSRAM
DE: Descriptor RAM of sDMAC
DA: Data RAM of sDMAC

Table 44.138 The relation between the module storing the error and the error source for sDMAC/DTSRAM

Protection target	Function	Controlled by	Captured to	Applicable for
Data from DTSRAM at DTS	Data ECC decode	ECCCNT_DTS	MECCCAP_DTS	Read data from DTSRAM
Address to DTSRAM at DTS	Address feedback check			Request address to DTSRAM
Data from Descriptor RAM at sDMAC0	Data ECC decode	ECCCNT_DMDE0	MECCCAP_DMDE0	Read data from Descriptor RAM of sDMAC0
Data from Descriptor RAM at sDMAC1		ECCCNT_DMDE1	MECCCAP_DMDE1	Read data from Descriptor RAM of sDMAC1
Data from Data RAM at sDMAC0		ECCCNT_DMDA0	MECCCAP_DMDA0	Read data from Data RAM of sDMAC0
Data from Data RAM at sDMAC1		ECCCNT_DMDA1	MECCCAP_DMDA1	Read data from Data RAM of sDMAC1

44.3.9.4 DRECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from DTSRAM and address feedback error detection for the address that is requested to DTSRAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFE DIS	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 44.139 DRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEDIS	Address feedback error disable bit Sets the address feedback error detection to enable/disable. 0: Address feedback error detection is enabled 1: Address feedback error detection is disabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.9.5 DRECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field of DTSRAM.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 44.140 DRECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for DTSRAM. 0: Normal mode 1: Test mode The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to data and ECC fields of DTSRAM when writing in test mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.9.6 DRAFINV — Address Feedback Test Control Register

This register is used to inject errors into the feedback address from DTSRAM for self-diagnosis. Feedback address XOR-ed with AFINV can be input to address feedback checker.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	AFINV[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 44.141 DRAFINV Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 2	AFINV[11:2]	Specifies bit pattern to inject errors into feedback address.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.9.7 DRECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of DTSRAM for self-diagnosis. The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to the requested address when write operation to DTSRAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 30_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.142 DRECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for DTSRAM. The data is provided to data field of DTSRAM when ECCTST = 1.

44.3.9.8 DRECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of DTSRAM for self-diagnosis. The values of DRECCTSTDIN0 and DRECCTSTEIN registers are written to the requested address when write operation to DTSRAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.143 DRECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for DTSRAM. The data is provided to ECC field of DTSRAM when ECCTST = 1.

44.3.9.9 DRTDATBFDATAF0 — Data Read Buffer Register

This register is used to hold data field read from DTSRAM for self-diagnosis. The data field in the requested address can be stored this register while reading from DTSRAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ECCCNT_DTS_base> + 50_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.144 DRTDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Data field read from requested address in DTSRAM

44.3.9.10 DRTDATBFECCF — ECC Read Buffer Register

This register is used to hold ECC field read from DTSRAM for self-diagnosis. ECC field in the requested address can be stored in this register while reading from DTSRAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DTS_base> + 60_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFECC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.145 DRTDATBFECCF Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	BFECC0[6:0]	ECC field read from the requested address in DTSRAM

44.3.9.11 DEECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read from Descriptor RAM of sDMAC.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDEn_base> + 00_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.146 DEECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets the 1-bit error correction to enable/disable. 0: Correct when 1-bit error r is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled.

44.3.9.12 DEECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field for Descriptor RAM of sDMAC.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDEn_base> + 10_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 44.147 DEECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for Descriptor RAM of sDMAC. 0: Normal mode 1: Test mode The values of DEECCTSTDIN0 and DEECCTSTEIN registers are written to data and ECC field for Descriptor RAM of sDMAC when writing is in test mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.9.13 DEECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of Descriptor RAM of sDMAC for self-diagnosis. The values of DEECCTSTDIN0 and DEECCTSTEIN registers are written to the requested address when write operation to Descriptor RAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDEn_base> + 30_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.148 DEECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for Descriptor RAM of sDMAC. The data is provided to data field of Descriptor RAM when ECCTST = 1.

44.3.9.14 DEECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of Descriptor RAM of sDMAC for self-diagnosis. The values of DEECCTSTDIN0 and DEECCTSTEIN registers are written to the requested address when write operation to Descriptor RAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDEn_base> + 40_H (n = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.149 DEECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for Descriptor RAM of sDMAC. The data is provided to ECC field of Descriptor RAM when ECCTST = 1.

44.3.9.15 DETDATBFDATAF0 — Data Read Buffer Register

This register is used to hold data field read from Descriptor RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Descriptor RAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ECCCNT_DMDEn_base> + 50_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.150 DETDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Data field read from the requested address in Descriptor RAM

44.3.9.16 DETDATBFECCF — ECC Read Buffer Register

This register is used to hold ECC field read from Descriptor RAM of sDMAC for self-diagnosis. ECC field in the address that is requested can be stored in this register while reading from Descriptor RAM where ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDEn_base> + 60_H (n = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BFECC0[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.151 DETDATBFECCF Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	BFECC0[6:0]	ECC field read from the address that is requested in Descriptor RAM

44.3.9.17 DAECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for read data from Data RAM of sDMAC.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 00_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.152 DAECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), set the 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets the ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.9.18 DAECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be written to data field and ECC field for Data RAM of sDMAC.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 10_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 44.153 DAECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for Data RAM of sDMAC. 0: Normal mode 1: Test mode The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to data and ECC field for Data RAM of sDMAC when buffering transfer data in test mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.9.19 DAECCTSTDIN0 — Test Lower Data Input Register

This register is used to inject errors into lower data field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 30_H (n = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.154 DAECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for Data RAM of sDMAC. The data is provided to lower data field of Data RAM when ECCTST = 1.

44.3.9.20 DAECCTSTDIN1 — Test Upper Data Input Register

This register is used to inject errors into upper data field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 34_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.155 DAECCTSTDIN1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for Data RAM of sDMAC. The data is provided to upper data field of Data RAM when ECCTST = 1.

44.3.9.21 DAECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of Data RAM of sDMAC for self-diagnosis. The values of DAECCTSTDIN0, DAECCTSTDIN1 and DAECCTSTEIN registers are written to the requested address when write operation to Data RAM occurs in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 40_H (n = 0, 1)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.156 DAECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ECC[7:0]	Specify the test data for Data RAM of sDMAC. The data is provided to ECC field of Data RAM when ECCTST = 1.

44.3.9.22 DATDATBFDATAF0 — Lower Data Read Buffer Register

This register is used to hold data field read from Data RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ECCCNT_DMDAn_base> + 50_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.157 DATDATBFDATAF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Lower data field read from the requested address in Data RAM

44.3.9.23 DATDATBFDATAF1 — Upper Data Read Buffer Register

This register is used to hold data field read from Data RAM of sDMAC for self-diagnosis. The data field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ECCCNT_DMDAn_base> + 54_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFDATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BFDATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.158 DATDATBFDATAF1 Register Contents

Bit Position	Bit Name	Function
31 to 0	BFDATA[31:0]	Upper data field read from the requested address in Data RAM

44.3.9.24 DATDATBFECCF — ECC Read Buffer Register

This register is used to hold ECC field read from Data RAM of sDMAC for self-diagnosis. ECC field in the requested address can be stored in this register while reading from Data RAM when ECCTST = 1.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_DMDAn_base> + 60_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BFECC0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.159 DATDATBFECCF Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	BFECC0[7:0]	ECC field read from the requested address in Data RAM

44.3.9.25 DR_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address feedback error, ECC error and/or overflow error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 00_H

Value after reset: $0000\ 008B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	AFEIE	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W

Table 44.160 DR_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	AFEIE	Controls error reports when address feedback error is detected. 0: Address feedback error report disabled 1: Address feedback error report enabled
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.9.26 DR_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DR_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.161 DR_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DR_SERSTR.

44.3.9.27 DR_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DR_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.162 DR_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears AFEF00 and DEDF00 in DR_DERSTR.

44.3.9.28 DR_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DR_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.163 DR_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DR_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DR_OVFSTR.

44.3.9.29 DR_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DR_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DR_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.164 DR_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDF00	Indicates that ECC 1-bit error was detected.

44.3.9.30 DR_DERSTR — Fatal Error Status Register

This register indicates whether an address feedback error and/or an ECC 2-bit error has occurred. The location and address of the error that is detected are stored in DR_00DEADR register when AFEF00 and/or DEDF00 flags are set. The AFEF00 and/or DEDF00 flag is set only when an address feedback error and/or an ECC 2-bit error is detected while all the flags are “0”. Multiple flags are set only when multiple errors occur due to one error cause.

This register can be cleared by DSTCLR00 in DR_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 24_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	AFEF00	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.165 DR_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	AFEF00	Indicates that address feedback error was detected.
2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.9.31 DR_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flag can be cleared by SERROVFCLR0 and SERROVFCLR1 in DR_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DTS_base> + 28_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.166 DR_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

44.3.9.32 DR_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DR_SERSTR register were detected.

This register is updated whenever DR_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DTS_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.167 DR_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDLINF00	Indicates that ECC 1-bit error was detected in DTSRAM.

44.3.9.33 DR_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DTS_base> + 70_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.168 DR_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	Reserved	When read, the value after reset is returned.
11 to 2	SEADR0[11:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Table 44.169 SEDL Information

SEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from DTSRAM.

44.3.9.34 DR_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when neither AFEF00 nor DEDF00 flag is set and an address feedback error and/or ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DTS_base> + $F0_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.170 DR_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	Reserved	When read, the value after reset is returned.
11 to 2	DEADR0[11:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Table 44.171 DEDL Information

DEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates a fatal error is detected in storing data to or loading data from DTSRAM.

44.3.9.35 DE_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when ECC error and/or overflow error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 00_H (n = 0, 1)

Value after reset: $0000\ 0083_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 44.172 DE_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.9.36 DE_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DE_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 10_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.173 DE_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DE_SERSTR.

44.3.9.37 DE_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DE_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 14_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.174 DE_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in DE_DERSTR.

44.3.9.38 DE_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DE_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 18_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.175 DE_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DE_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DE_OVFSTR.

44.3.9.39 DE_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DE_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DE_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 20_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.176 DE_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDF00	Indicates that ECC 1-bit error was detected.

44.3.9.40 DE_DERSTR — Fatal Error Status Register

This register indicates whether ECC 2-bit error has occurred. The location and the address of the error that is detected are stored in DE_00DEADR register when DEDF00 flag is set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in DE_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 24_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.177 DE_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.9.41 DE_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flags can be cleared by SERROVFCLR0 and SERROVFCLR1 in DE_OVFCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDEn_base> + 28_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.178 DE_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

44.3.9.42 DE_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DE_SERSTR register were detected.

This register is updated whenever DE_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDEn_base> + 30_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.179 DE_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDLINF00	Indicates that ECC 1-bit error was detected in Descriptor RAM of sDMAC.

44.3.9.43 DE_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDEn_base> + 70_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.180 DE_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 12	Reserved	When read, the value after reset is returned.
11 to 2	SEADR0[11:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Table 44.181 SEDL Information

SEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from Descriptor RAM of sDMAC.

44.3.9.44 DE_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and an ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDEn_base> + $F0_H$ (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEADR0[11:2]										—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.182 DE_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 12	Reserved	When read, the value after reset is returned.
11 to 2	DEADR0[11:2]	Indicates at which address this error was detected.
1, 0	Reserved	When read, the value after reset is returned.

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

Table 44.183 DEDL Information

DEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates a fatal error is detected in storing data to or loading data from Descriptor RAM of sDMAC.

44.3.9.45 DA_ERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when ECC error and/or overflow error occurs.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 00_H (n = 0, 1)

Value after reset: $0000\ 0083_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SEOVFIE	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 44.184 DA_ERRINT Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	SEOVFIE	Controls error reports when overflow error occurs. 0: Overflow error report disabled 1: Overflow error report enabled
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when ECC 2-bit error is detected. 0: ECC 2-bit error report disabled 1: ECC 2-bit error report enabled
0	SEDIE	Controls error reports when ECC 1-bit error is detected. 0: ECC 1-bit error report disabled 1: ECC 1-bit error report enabled

44.3.9.46 DA_SSTCLR — 1-bit Error Status Clear Register

This register is used to clear error flags in DA_SERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 10_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.185 DA_SSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SSTCLR00	Writing 1 to this bit clears SEDF00 in DA_SERSTR.

44.3.9.47 DA_DSTCLR — Fatal Error Status Clear Register

This register is used to clear error flags in DA_DERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 14_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.186 DA_DSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSTCLR00	Writing 1 to this bit clears DEDF00 in DA_DERSTR.

44.3.9.48 DA_OVFCLR — 1-bit Error Overflow Status Clear Register

This register is used to clear error overflow flags in DA_OVFSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 18_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF CLR1	SERR OVF CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 44.187 DA_OVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	SERROVFCLR1	Writing 1 to this bit clears SERROVF1 in DA_OVFSTR.
0	SERROVFCLR0	Writing 1 to this bit clears SERROVF0 in DA_OVFSTR.

44.3.9.49 DA_SERSTR — 1-bit Error Status Register

This register indicates whether ECC 1-bit error has occurred. The location and address of the error that is detected are stored in DA_00SEADR register when SEDF00 flag is set. The SEDF00 flag is set only when an ECC 1-bit error is detected while the SEDF00 is “0”.

This register can be cleared by SSTCLR00 in DA_SSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 20_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.188 DA_SERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDF00	Indicates that ECC 1-bit error was detected.

44.3.9.50 DA_DERSTR — Fatal Error Status Register

This register indicates whether ECC 2-bit error has occurred. The location and address of the error that is detected are stored in DA_00DEADR register when DEDF00 flag is set. The DEDF00 flag is set only when an ECC 2-bit error is detected while the DEDF00 is “0”.

This register can be cleared by DSTCLR00 in DA_DSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 24_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF00	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.189 DA_DERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF00	Indicates that ECC 2-bit error was detected.
0	Reserved	When read, the value after reset is returned.

44.3.9.51 DA_OVFSTR — 1-bit Error Overflow Status Register

This register indicates whether overflow of address buffer or working memory has occurred.

SERROVF0 flag is set when a unique ECC 1-bit error that differs from previous errors in error address is detected when the address buffer is fully used. If the newly detected error has the same address as errors already stored, this flag is not set.

On the other hand, SERROVF1 flag is set when the number of times an ECC 1-bit error occurs at the same time regardless of the same occurrence address exceed the size of internal buffer.

SERROVF0 and SERROVF1 flag can be cleared by SERROVFCLR0 and SERROVFCLR1 in DA_OVFCLR register respectively.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <MECCCAP_DMDAn_base> + 28_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SERR OVF1	SERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.190 DA_OVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	SERROVF1	Indicates that more ECC 1-bit errors than the working memory were detected simultaneously.
0	SERROVF0	Indicates that a unique ECC 1-bit error was detected when SEDF00 was already set.

44.3.9.52 DA_SERINF — 1-bit Error Location Information Register

This register is used to indicate summary of locations where ECC 1-bit errors stored in DA_SERSTR register were detected.

This register is updated whenever DA_SERSTR register is updated.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDAn_base> + 30_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDLINF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.191 DA_SERINF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	SEDLINF00	Indicates that ECC 1-bit error was detected in Data RAM of sDMAC.

44.3.9.53 DA_00SEADR — 1st 1-bit Error Address Register

This register is used to hold the address and the location of the error when SEDF00 flag is not set and an ECC 1-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDAn_base> + 70_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SEADR0[10:3]								—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.192 DA_00SEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	SEDL[4:0]	Indicates where this error was detected.
26 to 11	Reserved	When read, the value after reset is returned.
10 to 3	SEADR0[10:3]	Indicates at which address this error was detected.
2 to 0	Reserved	When read, the value after reset is returned.

Table 44.193 SEDL Information

SEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates an ECC 1-bit error is detected in storing data to or loading data from Data RAM of sDMAC.

44.3.9.54 DA_00DEADR — 1st Fatal Error Address Register

This register is used to hold the address and the location of the error when DEDF00 flag is not set and an ECC 2-bit error is detected.

This register can only be cleared by reset.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <MECCCAP_DMDAn_base> + $F0_H$ (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEDL[4:0]					—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DEADR0[10:3]								—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.194 DA_00DEADR Register Contents

Bit Position	Bit Name	Function
31 to 27	DEDL[4:0]	Indicates where this error was detected.
26 to 11	Reserved	When read, the value after reset is returned.
10 to 3	DEADR0[10:3]	Indicates at which address this error was detected.
2 to 0	Reserved	When read, the value after reset is returned.

Table 44.195 DEDL Information

DEDL[4:0]	Indicated information
31 to 1	These values are reserved
0	Indicates a fatal error is detected in storing data to or loading data from Data RAM of sDMAC.

44.3.9.55 Test Function

(1) DTSRAM ECC

The ECC of the DTSRAM can be tested by using the following procedure:

1. Make sure that the RAM address to be tested is not modified by any ongoing DTS transfer. It is recommended to disable DTS entirely.
2. Write data to any DTSRAM address.
3. Write 02_H to the DTSRAM ECC Test Control Register (DRECCTSTCTL). Enable ECC Test Mode and encode ECC from write data.
4. Read data from the same DTSRAM address.
5. Check Data and ECC field by using the Data Read Buffer Register (DRTDATBFDATAF0) and ECC Read Buffer Register (DRTDATBFECCF).
6. Restore DTSRAM data and ECC with 1-bit or 2-bit errors by using the ECC Test Data Input Register (DRECCTSTDIN0) or the ECC Test ECC Field Input Register (DRECCTSTEIN).
7. Write data to the same DTSRAM address.
8. Write 00_H to the DTSRAM ECC Test Control Register (DRECCTSTCTL). Then, Read data from the same DTSRAM address and check if ECC error occurs.

(2) sDMAC Descriptor RAM ECC

The ECC of the sDMAC Descriptor RAM can be tested by using the following procedure:

1. Make sure that the RAM address to be tested is not modified by any ongoing sDMAC transfer. It is recommended to disable sDMAC entirely.
2. Write data to any sDMAC Descriptor RAM address.
3. Write 02_H to the sDMAC Descriptor RAM ECC Test Control Register (DEECCTSTCTL). Enable ECC Test Mode and encode ECC from write data.
4. Read data from the same sDMAC Descriptor RAM address.
5. Check Data and ECC field by using the Data Read Buffer Register (DETDATBFDATAF0) and ECC Read Buffer Register (DETDATBFECCF).
6. Restore sDMAC Descriptor RAM data and ECC with 1-bit or 2-bit errors by using the ECC Test Data Input Register (DEECCTSTDIN0) or the ECC Test ECC Field Input Register (DEECCTSTEIN).
7. Write data to the same sDMAC Descriptor RAM address.
8. Write 00_H to the sDMAC Descriptor RAM ECC Test Control Register (DEECCTSTCTL). Then, Read data from the same sDMAC Descriptor RAM address and check if ECC error occurs.

(3) sDMAC Data RAM ECC

The ECC of the sDMAC Data RAM can be tested by using ECC calculation of ERRGEN. For details, see **Section 44.3.12.8, Usage of ERRGEN**.

44.3.10 ECC for Peripheral RAM

44.3.10.1 Overview

This is an ECC module for the RAM of the following peripheral modules:

RS-CANFD, FlexRay, Ethernet, GTM, MSPI and MMCA

Error Detection and Correction

Seven-bit ECC data is appended to the 32-bit RAM data.

This ECC module provides 2-bit ECC error detection and 1-bit ECC error detection and correction.

Table 44.196 RS-CANFD, FlexRay, Ethernet, GTM, MSPI and MMCA RAM ECC

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection, and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled; 2-bit error detection, and 1-bit error detection and correction are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. **Overflow Error:** (only for RS-CANFD, FlexRay, GTM and MSPI and MMCA RAM ECC, except for Ethernet) • Error notification is issued upon detection of an address buffer overflow error for ECC error. An ECC 2-bit error and an ECC 1-bit error are handled as individual source in ECM. Overflow errors of each peripheral module (except Ethernet) are notified to ECM as one factor. An ECC error signal is only issued to the ECM, if the ECC error address is not yet stored in the error address buffer.
Error status	A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. A one-stage buffer is provided for 2-bit error and 1-bit error. The error status can be cleared using the clear register.
Address capture	The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address. Address buffer is updated if the error status is cleared. One-stage address buffer is provided for an ECC 2-bit error and an ECC 1-bit error.
Self-diagnosis	• By setting the test mode, register values can be used as the data to be output to the RAM. When a peripheral module writes to the RAM, the E710TED.ECEDB[31:0] bit value can be written to the RAM data section, and the E710TRC.ECERDB[6:0] bit value can be written to the ECC redundant bit section. • By setting the test mode, the ECC redundant bit section can be latched when RAM data is read, and the value can be confirmed. • By setting the test mode, the ECC redundant bit (encoding circuit) and syndrome code (decoding circuit), which are generated from the input data, can be confirmed.

44.3.10.2 List of Registers

(1) List of ECC Modules

The RAMs of the multiple peripheral functions are provided with the ECC modules. The following table shows the peripheral functions provided with the ECC modules, the corresponding ECC module names, and base addresses of the ECC modules.

Table 44.197 List of ECC modules

Peripheral Functions		ECC Module Names and Register Base Addresses	
		Module Name	Base Address <base_addr>
RS-CANFD0 (P-Bus Group 8)	Message buffer RAM (MB RAM)	E7RC01	FFF2 B000$_H$
	Acceptance filter list RAM (AFL0 RAM)	E7RC02	FFF2 B200$_H$
	Acceptance filter list RAM (AFL1 RAM)	E7RC03	FFF2 B400$_H$
RS-CANFD1 (P-Bus Group 3)	Message buffer RAM (MB RAM)	E7RC11	FFD2 2000$_H$
	Acceptance filter list RAM (AFL0 RAM)	E7RC12	FFD2 2040$_H$
	Acceptance filter list RAM (AFL1 RAM)	E7RC13	FFD2 2080$_H$
FlexRay0 (P-Bus Group 9)	Message RAM (MRAM)	E7FR00	FF05 0000$_H$
	Temporary buffer (TBF A)	E7FR01	FF05 0200$_H$
	Temporary buffer (TBF B)	E7FR02	FF05 0400$_H$
FlexRay1*3 (P-Bus Group 9)	Message RAM (MRAM)	E7FR10	FF05 0600$_H$
	Temporary buffer (TBF A)	E7FR11	FF05 0800$_H$
	Temporary buffer (TBF B)	E7FR12	FF05 0A00$_H$
GTM (P-Bus Group 6H)	MCS0_RAM0	E7GT00	FF70 2000$_H$
	MCS0_RAM1	E7GT01	FF70 2200$_H$
	MCS1_RAM0	E7GT10	FF70 2400$_H$
	MCS1_RAM1	E7GT11	FF70 2600$_H$
	MCS2_RAM0	E7GT20	FF70 2800$_H$
	MCS2_RAM1	E7GT21	FF70 2A00$_H$
	MCS3_RAM0	E7GT30	FF70 2C00$_H$
	MCS3_RAM1	E7GT31	FF70 2E00$_H$
MSPIn (n=0,2,4*4,6*1,8*1) (P-Bus Group 4) (n=1,3,5*4,7*1,9*2) (P-Bus Group 5)	MSPI0 RAM	E7MS00	FFC7 5700$_H$
	MSPI1 RAM	E7MS01	FFC7 C800$_H$
	MSPI2 RAM	E7MS02	FFC7 5800$_H$
	MSPI3 RAM	E7MS03	FFC7 CA00$_H$
	MSPI4 RAM*4	E7MS04*4	FFC7 5900$_H$
	MSPI5 RAM*4	E7MS05*4	FFC7 CC00$_H$
	MSPI6 RAM*1	E7MS06*1	FFC7 5A00$_H$
	MSPI7 RAM*1	E7MS07*1	FFC7 CE00$_H$
	MSPI8 RAM*1	E7MS08*1	FFC7 5B00$_H$
	MSPI9 RAM*2	E7MS09*2	FFC7 D000$_H$
MMCA*5 (P-Bus Group 3)	MMCA0 RAM A	E7MM00	FFD5 5200$_H$
	MMCA0 RAM B	E7MM01	FFD5 5300$_H$
ETNB0*6 (P-Bus Group 9)	Fast Ethernet TX RAM	E7ME00	FF0A 3000$_H$
	Fast Ethernet RX RAM	E7ME01	FF0A 3200$_H$
ETNB1*3 (P-Bus Group 9)	Gb Ethernet TX RAM	E7GE00	FF0A 5400$_H$
	Gb Ethernet RX RAM	E7GE01	FF0A 5600$_H$

Note 1. This function is not implemented in RH850/U2A16 (292 pins), RH850/U2A8 (292 pins) and RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A16 (373/292 pins), RH850/U2A8 (373/292 pins) and RH850/U2A6.
Note 3. This function is not implemented in RH850/U2A6.
Note 4. This function is not implemented in RH850/U2A6 (144 pins).
Note 5. This function is not implemented in RH850/U2A6 (156/144 pins).
Note 6. This function is not implemented in RH850/U2A6 (156 pins).

(2) List of Registers

Table 44.198 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
E7RCnm(n=0,1, m=1-3) E7FRnm(n=0,1, m=0-2)*3 E7GTnm(n=0-3, m=0,1) E7MS0m(m=0-9)*2 E7MM0m(m=0,1)*4 E7ME0m(m=0,1)*5 E7GE0m(m=0,1)*3	ECC control register	E710CTL	<base_addr> + 00_H	8, 16	*1	—
	ECC test mode control register	E710TMC	<base_addr> + 04_H	8, 16	*1	—
	ECC redundant bit data control test register	E710TRC	<base_addr> + 08_H	32	*1	—
	ECC encoder and decoder data test register	E710TED	<base_addr> + $0C_H$	32	*1	—
	ECC error address register	E710EAD	<base_addr> + 10_H	32	*1	—

Note 1. E7RC01,E7RC02,E7RC03: PBG80#1
E7RC11,E7RC12,E7RC13: PBG30#1
E7FR00,E7FR01,E7FR02: PBG90#1
E7FR10,E7FR11,E7FR12: PBG90#1
E7GT00,E7GT01,E7GT10,E7GT11,E7GT20,E7GT21,E7G30,E7GT31: PBG6H0#1
E7MS00, E7MS02, E7MS04, E7MS06, E7MS08: PBG40#1
E7MS01, E7MS03, E7MS05, E7MS07, E7MS09: PBG50#1
E7MM00, E7MM01: PBG30#1
E7ME00, E7ME01: PBG90#1
E7GE00, E7GE0:1 PBG90#1
Note 2. E7MS0m (m = 6, 7, 8) is not implemented in RH850/U2A16 (292 pins), RH850/U2A8 (292 pins) and RH850/U2A6. And E7MS0m (m = 9) is not implemented in RH850/U2A16 (373/292 pins), RH850/U2A8 (373/292 pins) and RH850/U2A6. E7MS0m (m = 4, 5) is not implemented in RH850/U2A6 (144pins).
Note 3. E7FRnm (n = 1), E7GE0m (m = 0, 1) are not implemented in RH850/U2A6.
Note 4. E7MM0m (m = 0, 1) is not implemented in RH850/U2A6 (156/144 pins).
Note 5. E7ME0m (m = 0, 1) is not implemented in RH850/U2A6 (156 pins).

Table 44.199 The relation between the module storing the error and the error source for Peripheral RAM (1/2)

Protection target	Function	Controlled by	Captured to	Applicable for
RS-CANFD0 Message buffer RAM (MB RAM)	Data ECC decode	E7RC01	E7RC01	Read data from peripheral RAM (RS-CANFD0 Message buffer RAM (MB RAM))
RS-CANFD0 Acceptance filter list RAM (AFL0 RAM)		E7RC02	E7RC02	Read data from peripheral RAM (RS-CANFD0 Acceptance filter list RAM (AFL0 RAM))
RS-CANFD0 Acceptance filter list RAM (AFL1 RAM)		E7RC03	E7RC03	Read data from peripheral RAM (RS-CANFD0 Acceptance filter list RAM (AFL1 RAM))
RS-CANFD1 Message buffer RAM (MB RAM)	Data ECC decode	E7RC11	E7RC11	Read data from peripheral RAM (RS-CANFD1 Message buffer RAM (MB RAM))
RS-CANFD1 Acceptance filter list RAM (AFL0 RAM)		E7RC12	E7RC12	Read data from peripheral RAM (RS-CANFD1 Acceptance filter list RAM (AFL0 RAM))
RS-CANFD1 Acceptance filter list RAM (AFL1 RAM)		E7RC13	E7RC13	Read data from peripheral RAM (RS-CANFD1 Acceptance filter list RAM (AFL1 RAM))
FlexRay0 Message RAM (MRAM)	Data ECC decode	E7FR00	E7FR00	Read data from peripheral RAM (FlexRay0 Message RAM (MRAM))
FlexRay0 Temporary buffer (TBF A)		E7FR01	E7FR01	Read data from peripheral RAM (FlexRay0 Temporary buffer (TBF A))
FlexRay0 Temporary buffer (TBF B)		E7FR02	E7FR02	Read data from peripheral RAM (FlexRay0 Temporary buffer (TBF B))
FlexRay1 Message RAM (MRAM)*3	Data ECC decode	E7FR10	E7FR10	Read data from peripheral RAM (FlexRay1 Message RAM (MRAM))
FlexRay1 Temporary buffer (TBF A)*3		E7FR11	E7FR11	Read data from peripheral RAM (FlexRay1 Temporary buffer (TBF A))
FlexRay1 Temporary buffer (TBF B)*3		E7FR12	E7FR12	Read data from peripheral RAM (FlexRay1 Temporary buffer (TBF B))
GTM MCS0_RAM0	Data ECC decode	E7GT00	E7GT00	Read data from peripheral RAM (GTM MCS0_RAM0)
GTM MCS0_RAM1		E7GT01	E7GT01	Read data from peripheral RAM (GTM MCS0_RAM1)
GTM MCS1_RAM0		E7GT10	E7GT10	Read data from peripheral RAM (GTM MCS1_RAM0)
GTM MCS1_RAM1		E7GT11	E7GT11	Read data from peripheral RAM (GTM MCS1_RAM1)
GTM MCS2_RAM0		E7GT20	E7GT20	Read data from peripheral RAM (GTM MCS2_RAM0)
GTM MCS2_RAM1		E7GT21	E7GT21	Read data from peripheral RAM (GTM MCS2_RAM1)
GTM MCS3_RAM0		E7GT30	E7GT30	Read data from peripheral RAM (GTM MCS3_RAM0)
GTM MCS3_RAM1		E7GT31	E7GT31	Read data from peripheral RAM (GTM MCS3_RAM1)
MSPI0 RAM	Data ECC decode	E7MS00	E7MS00	Read data from peripheral RAM (MSPI0 RAM)
MSPI1 RAM		E7MS01	E7MS01	Read data from peripheral RAM (MSPI1 RAM)
MSPI2 RAM		E7MS02	E7MS02	Read data from peripheral RAM (MSPI2 RAM)
MSPI3 RAM		E7MS03	E7MS03	Read data from peripheral RAM (MSPI3 RAM)
MSPI4 RAM*4		E7MS04*4	E7MS04	Read data from peripheral RAM (MSPI4 RAM)
MSPI5 RAM*4		E7MS05*4	E7MS05	Read data from peripheral RAM (MSPI5 RAM)
MSPI6 RAM*1		E7MS06*1	E7MS06	Read data from peripheral RAM (MSPI6 RAM)
MSPI7 RAM*1		E7MS07*1	E7MS07	Read data from peripheral RAM (MSPI7 RAM)
MSPI8 RAM*1		E7MS08*1	E7MS08	Read data from peripheral RAM (MSPI8 RAM)
MSPI9 RAM*2		E7MS09*2	E7MS09	Read data from peripheral RAM (MSPI9 RAM)
MMCA0 RAM A*5	Data ECC decode	E7MM00	E7MM00	Read data from peripheral RAM (MMCA0 RAM A)
MMCA0 RAM B*5		E7MM01	E7MM01	Read data from peripheral RAM (MMCA0 RAM B)

Table 44.199 The relation between the module storing the error and the error source for Peripheral RAM (2/2)

Protection target	Function	Controlled by	Captured to	Applicable for
ETNB0 Fast Ethernet TX RAM*6	Data ECC decode	E7ME00	E7ME00	Read data from peripheral RAM (ETNB0 Fast Ethernet TX RAM)
ETNB0 Fast Ethernet RX RAM*6		E7ME01	E7ME01	Read data from peripheral RAM (ETNB0 Fast Ethernet RX RAM)
ETNB1 Gb Ethernet TX RAM*3	Data ECC decode	E7GE00	E7GE00	Read data from peripheral RAM (ETNB1 Gb Ethernet TX RAM)
ETNB1 Gb Ethernet RX RAM*3		E7GE01	E7GE01	Read data from peripheral RAM (ETNB1 Gb Ethernet RX RAM)

Note 1. This function is not implemented in RH850/U2A16 (292 pins), RH850/U2A8 (292 pins) and RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A16 (373/292 pins), RH850/U2A8 (373/292 pins) and RH850/U2A6.
Note 3. This function is not implemented in RH850/U2A6.
Note 4. This function is not implemented in RH850/U2A6 (144 pins).
Note 5. This function is not implemented in RH850/U2A6 (156/144 pins).
Note 6. This function is not implemented in RH850/U2A6(156 pins).

(3) Register Map

Table 44.200 Register Map

Symbol	31 24	23 16	15 8	7 0	Address
E710CTL	— (00_H)	— (00_H)	ECCTL[15:8]	ECCTL[7:0]	00_H
E710TMC	— (00_H)	— (00_H)	ECTMC[15:8]	ECTMC[7:0]	04_H
E710TRC	ECSYND[7:0]	ECHORD[7:0]	ECECRD[7:0]	ECERDB[7:0]	08_H
E710TED	ECEDB[31:24]	ECEDB[23:16]	ECEDB[15:8]	ECEDB[7:0]	$0C_H$
E710EAD	ECEAD[31:24]	ECEAD[23:16]	ECEAD[15:8]	ECEAD[7:0]	10_H

44.3.10.3 E710CTL — ECC Control Register

E710CTL controls the status and modes of the ECC modules.

E710CTL can be read and written to using the 16-bit or 8-bit manipulation instruction.

Access: This register can be read or written in 16-bit or 8-bit units.

Address: <base_addr> + 00_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R/W	R	R	R

Note 1. This bit is always read as 0.

Table 44.201 E710CTL Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	EMCA[1:0]	Sets Access Control 1 and 0 to ECC Mode Select Bit These bits enable or disable writing to bit 7. The read value is always 0.
13, 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ECOVFF	Error Overflow Detection Flag This flag is set when an error of any source other than the same error address occurs while the error flag (ECER2F or ECER1F) is set. One of the following clears this flag: • An internal or external reset • When ECC function is disabled (ECTHM = 1) • Writing 1 to ECER2C and ECER1C
10	ECER2C	2-Bit ECC Error Detection Flag Clear Clears the bit 2 (ECER2F) status flag. The read value is always 0, and writing 0 to ECER2C does not change the state. If writing 1 to ECER2C conflicts with the condition for setting bit 2, the former takes priority. Writing 1 to ECER2C while ECER2F is set clears ECER2F.
9	ECER1C	1-Bit ECC Error Detection Flag Clear Clears the bit 1 (ECER1F) status flag. The read value is always 0, and writing 0 to ECER1C does not change the internal state. If writing 1 to ECER1C conflicts with the condition for setting bit 1, the former takes priority. Writing 1 to ECER1C while ECER1F is set clears ECER1F.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function Disable Select Selects the ECC decoding operation. Write access to ECTHM is enabled when the value of bits 15 and 14 is 01_B. Therefore, only the 16-bit manipulation instruction is valid. Setting ECTHM to 1 disables error detection and bit correction. Here, if the data to be output to the peripheral module contains an error, the data is output without bit correction. Setting ECTHM to 1 has no effect on the encoder side. 0: Enable ECC detection and correction. 1: Disable ECC detection and correction.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit ECC Error Correction Enable 0: Enable 1-bit error correction upon error detection. 1: Disable 1-bit error correction upon error detection.

Table 44.201 E710CTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	EC2EDIC	2-Bit ECC Error Detection Notification Enable 0: When a 2-bit error is detected, the ECM is not notified of the error. 1: When a 2-bit error is detected, the ECM is notified of the error.
3	EC1EDIC	1-Bit ECC Error Detection Notification Enable 0: When a 1-bit error is detected, the ECM is not notified of the error. 1: When a 1-bit error is detected, the ECM is notified of the error.
2	ECER2F	2-Bit ECC Error Detection Flag Indicates that errors have been detected at two bits in bits 0 to 38 of the data read from the RAM during RAM read access while error detection is enabled. This bit is a read only flag. 0: A 2-bit error has not occurred since this bit was cleared. 1: A 2-bit error has occurred. Clearing conditions (1) Reset is applied. (2) 1 is written to ECER2C. (3) ECC function is disabled (ECTHM = 1)
1	ECER1F	1-Bit ECC Error Detection And Correction Flag Indicates that an error has been detected at one bit in bits 0 to 38 of the data read from the RAM during RAM read access while error detection is enabled. Setting this bit generates no interrupt signal. This bit is a read only flag. 0: A 1-bit error has not occurred. 1: A 1-bit error has occurred. Clearing conditions (1) Reset is applied. (2) 1 is written to ECER1C. (3) ECC function is disabled (ECTHM = 1)
0	ECEMF	ECC Error Message Flag Indicates that the current read data contains an error. ECEMF is updated every time RAM data is read. Since the RAM value after reset is undefined, if this bit is read before the RAM is initialized, this bit might be set, indicating an error. Therefore, the ECEMF value after reset is undefined. 0: The current RAM data contains no bit errors. 1: The current RAM data contains bit errors. Clearing conditions (1) ECC function is disabled (ECTHM = 1) (2) Decoding circuit input data contains no 1-bit errors. ECEMF remains set as long as RAM data containing a bit error is output with error detection being enabled.

44.3.10.4 E710TMC — ECC Test Mode Control Register

E710TMC is a 16-bit register used to switch the mode to test mode and control the mode.

E710TMC can be read and written to using the 16-bit or 8-bit manipulation instruction.

Access: This register can be read or written in 16-bit or 8-bit units.

Address: <base_addr> + 04_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. This bit is always read as 0.

Table 44.202 E710TMC Register Contents (1/2)

Bit Position	Bit Name	Function
15, 14	ETMA[1:0]	Access Control 1 and 0 to ECC Test Mode Control Enable Bit These bits enable or disable writing to bit 7. The read value is always 0.
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Control Enable ECTMCE enables or disables access to the test registers and test control bits. Write access to ETCMCE is enabled when the value of bits 15 and 14 is 10_B. 0: Disable access to the test registers and test control bits. 1: Enable access to the test registers and test control bits.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Control Enable ECTRRS enables the read status of RAM to be generated by reading the E710TED register, and also allows the RAM output data to be read out when the E710TRC:ECERDB[7:0] bits and the E710TED register are read. Write access to ECTRRS is enabled only when ECTMCE is 1 (can be set simultaneously). ECTRRS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Disable generation of RAM read status for testing when E710TED is read. 1: Enable generation of RAM read status for testing when E710TED is read. When E710TRC:ECERDB[7:0] and E710TED are read, the values of the RAM output data pin are read out.
3	ECREOS	ECC Redundant Bit Output Data Select ECREOS selects either the ECC encoder output data or the ECERDB register value to be output as the ECC redundant bit output. Write access to ECREOS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREOS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow encoding result to be output as the ECC redundant bit output. 1: Allow the E710TRC:ECERDB[6:0] value to be output as the ECC redundant bit output.

Table 44.202 E710TMC Register Contents (2/2)

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Select ECENS selects either the data value from the peripheral module or the internal test register value (E710TED:ECEDB[31:0]) as the input signal to be encoded. Write access to ECENS is enabled only when ECTMCE is 1 (can be set simultaneously). ECENS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow the RAM write data from the peripheral module to be input as the ECC encoder input data. 1: Allow the E710TED:ECEDB[31:0] value to be input as the ECC encoder input data.
1	ECDCS	ECC Decoder Input Select ECDCS selects either the lower 32-bit data value from the RAM or the internal test register value (E710TED:ECEDB[31:0]) as the lower 32-bit data of the input signal to be decoded. Write access to ECDCS is enabled only when ECTMCE is 1 (can be set simultaneously). ECDCS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow the lower 32-bit RAM output data to be input to the data area (32 lower-order bits) to the decoder. 1: Allow the E710TED:ECEDB[31:0] value to be input to the data area to the decoder.
0	ECREIS	ECC Redundant Bit Input Data Select ECREIS selects either the upper 7-bit data value from the RAM (redundant bit area) or the internal test register value (E710TRC:ECERDB[6:0]) as the upper 7-bit data of the input signal to be decoded. Write access to ECREIS is enabled only when ECTMCE is 1 (can be set simultaneously). ECREIS is cleared by writing 0 to ECTMCE (can be cleared synchronously). 0: Allow the upper 7-bit RAM output data to be input to the ECC redundant bit area to the decoder. 1: Allow the E710TRC:ECERDB[6:0] value to be input to the ECC redundant bit area to the decoder.

44.3.10.5 E710TED — ECC Encoder and Decoder Data Test Register

E710TED is a 32-bit data test register for ECC encoding and decoding.

When ECTMCE = 1, E710TED can be read and written to using the 32-bit manipulation instruction.

When ECTMCE = 0, E710TED is always read as 0.

In test mode, the E710TED value can be used as the data input to the encoding circuit and decoding circuit.

Access: This register can be read or written in 32-bit units.

Address: <base_addr> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

Changing ECTMCE from 1 to 0 resets E710TED synchronously.

When E710TMC:ECENS = 1, the ECEDB value is input to the encoding circuit and supplied to the RAM.

When E710TMC:ECDCS = 1, the ECEDB value is input as bits 31 to 0 of the data input to the decoding circuit.

When E710TMC:ECTRRS = 1, reading ECEDB returns the RAM output data instead of the data written to ECEDB.

44.3.10.6 E710TRC — ECC Redundant Bit Data Control Test Register

E710TRC is a 32-bit test register consisting of four fields (ECSYND, ECHORD, ECECRD, and ECERDB) corresponding to the ECC redundant bit area. Each field can be accessed as the 8-bit register with the same name. For details of each field, refer to the descriptions of these four registers.

When ECTMCE = 0, E710TRC is always read as 0.

When ECTMCE = 1, E710TRC can be read using the 32-bit manipulation instruction.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <base_addr> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECSYND								ECHORD							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECECRD								ECERDB							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NOTE

Changing ECTMCE from 1 to 0 resets E710TRC synchronously.

44.3.10.7 ECSYND — ECC Decoder Syndrome Data Register

ECSYND is a read-only register used to confirm the syndrome code generated by the decoding circuit in test mode (ECTMCE = 1).

Write access to ECSYND is ignored.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <base_addr> + $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

When read, ECSYND bits return the value of the syndrome code (SYND[6:0]) generated based on the data input to the decoding circuit.

ECSYND bits are not holding circuits; the register value changes as the input signal changes.

ECSYND is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned.

44.3.10.8 ECHORD — ECC 7-Bit Redundant Data Holding Test Register

ECHORD holds the 7-bit ECC redundant area (upper 7-bit RAM data) that cannot be confirmed by the peripheral module when the peripheral module accesses the RAM for reading in test mode (ECTMCE = 1).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <base_addr> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ECHORD bits are loaded with the upper 7-bit RAM output data at the next rising edge of the operating clock signal after the peripheral module accesses the RAM for reading data in test mode (ECTMCE = 1).

ECHORD bits are also loaded with the data on the input pins EC7TERI38 to EC7TERI32 at the next operating clock pulse when the ECEDB[15:0] register is read while E710TMC:ECTRRS = 1.

ECHORD is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned.

44.3.10.9 ECECRD — ECC Encoder Test Register

ECECRD is a read-only register used to read the 7-bit redundant section generated by the encoding circuit in test mode (ECTMCE = 1).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <base_addr> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

ECECRD is used to confirm the redundant bits generated by the input data from the peripheral module. Here, the data that is read is the result of encoding (ECC[6:0]), not the output value.

ECECRD is valid only when ECTMCE = 1, and is always read as 00_H when ECTMCE = 0.

Bit 7 is reserved. When read, the value after reset is returned.

44.3.10.10 ECERDB — ECC Redundant Bit Input and Output Substitution Buffer Register

ECERDB is a buffer register for the data that substitutes for the input and output data for the 7-bit ECC redundant data area in test mode (ECTMCE = 1).

ECERDB can be read and written to in ECC test mode (ECTMCE = 1).

Access: This register can be read or written in 8-bit units.

Address: <base_addr> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When ECREOS = 1, ECERDB value, instead of the seven redundant bits generated by the encoding circuit, is output to the pin and supplied to the RAM.

When ECREIS = 1, ECERDB value, instead of the upper seven data bits to be input to the decoding circuit, is handled by the decoding circuit.

When ECTRRS = 1, reading ECERDB returns the signal value supplied to RAM instead of the data written to ECERDB.

Bit 7 is reserved. When read, the value after reset is returned. When writing, write the value after reset.

44.3.10.11 E710EAD — ECC Error Address Register

E710EAD is a read-only register used to hold the address at which an ECC error has occurred.

If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in E710EAD as the address at which the ECC error occurred.

The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. The address can be calculated by adding the base address.

Only one address can be held in E710EAD.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <base_addr> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

NOTE

For the formula of the ECC error address calculation, refer to **Section 44.3.13, ECC error address calculation**.

44.3.10.12 Notification to ECM

Detection of errors in two bits can be set for this module and the ECM is notified of detected errors.

- 1-bit Error Notification

  While EC1EDIC is set to $1_{B,}$ when an error is detected in one bit from among bits 0 to 38 of data read from RAM, the ECM is notified of the 1-bit error. When ECER1F or ECER2F is already set, however, the ECM will not be notified of the error.

- 2-bit Error Notification

  While EC2EDIC is set to 1_B, when an error is detected in two bits from among bits 0 to 38 of data read from RAM, the ECM is notified of the 2-bit error. When ECER2F is already set, however, the ECM will not be notified of the error.

44.3.10.13 Test Function

(1) Writing RAM Data

Write data to the peripheral RAM. However, the ECC corresponding to the written data will be written to the ECC bits simultaneously. In order to write a specified value to the ECC bits, use the ECC test mode described in **(3)** below.

(2) Reading RAM Data

(a) Set ECTHM bit in E710CTL register to 1 to disable ECC error detection and correction.

(b) Read the peripheral RAM. Since neither error detection nor correction proceeds when the peripheral RAM is read, the RAM data is read unchanged.

How to exit this test mode:

(a) Set ECTHM bit in E710CTL register to 0 to enable ECC error detection and correction.

(3) Writing to the ECC Bits

(a) Set ECTMCE bit in E710TMC register to 1 to set ECC test mode.

(b) Write the value to be written to the ECC bits to E710TRC.ECERDB[6:0].

(c) Set ECREOS bit in E710TMC register to 1 to select writing of the value of E710TRC.ECERDB[6:0] bits to the ECC bits.

(d) When data is written to the peripheral RAM, the value in E710TRC.ECERDB[6:0] bits will be written to the ECC bits.

How to exit this test mode:

(a) Set ECTMCE bit in E710TMC register to 0 to set normal mode.

(4) Reading the ECC Bits

(a) Set ECTMCE bit in E710TMC register to 1 to set ECC test mode.

(b) When data in the peripheral RAM is read, the ECC bits are stored in E710TRC.ECHORD[6:0] bits.

How to exit this test mode:

(a) Set ECTMCE bit in E710TMC register to 0 to set normal mode.

44.3.11 Safety Mechanism on Data Transfer Path

44.3.11.1 Overview

This product provides the function which protects the transaction data and the address on each bus. The scope of this function covers Cluster RAM Bus, Local FLASH Bus, Global FLASH Bus, Inter-processor element Bus, Inter-cluster Bus, System Bus, Peripheral Bus and High speed Bus.

ECC protection on bus is summarized in the table below.

Table 44.203 ECC protection on Bus

Item	Description
Data ECC error detection and correction	ECC error detection and correction on data bus between master and slave can be either enabled or disabled. When enabled, either of the following settings can be selected: • ECC error detection and correction are carried out (2-bit error detection, 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection and correction, and 2-bit error detection are carried out.
Address ECC error detection	ECC error detection on address bus between master and slave can be either enabled or disabled. In the initial state, 1-bit error detection and 2-bit error detection are carried out.
Error notification	The occurrence of a Data ECC error or an Address EDC error is reported to the Error Control Module. **Data ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit and ECC 1-bit error. **Address EDC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit and ECC 1-bit error. The error notification signal is output with an Address ECC 2-bit error, an Address ECC 1-bit error handled as one source, and a Data ECC 2-bit error and a Data ECC 1-bit error are handled as individual sources.
Error status	A status register is provided that indicates the status of Data ECC 2-bit error detection, Data ECC 1-bit error detection, Address ECC 2-bit error detection, and Address ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Self-diagnosis	ERRGEN is available for self-diagnosis of ECC decoder of data transfer path.

44.3.11.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 44.204 Register Base Addresses (1/3)

Base Address Name	Base Address	Bus Group
<ECCCNT_A_CCIB0CL0_base>	FFC4 $B000_H$	P-Bus Group 0
<ECCCNT_A_CCIB1CL0_base>	FFC4 $B080_H$	P-Bus Group 0
<ECCCNT_A_CCIB0CL1_base>*1	FFC4 $B200_H$	P-Bus Group 0
<ECCCNT_A_CCIB1CL1_base>*1	FFC4 $B280_H$	P-Bus Group 0
<ECCCNT_SA_PE0CL0_base>	FFC4 $C000_H$	P-Bus Group 0
<ECCCNT_A_GCFU0ICL0_base>	FFC4 $C080_H$	P-Bus Group 0
<ECCCNT_A_GCFU0DCL0_base>	FFC4 $C100_H$	P-Bus Group 0
<ECCCNT_SA_PE1CL0_base>	FFC4 $C200_H$	P-Bus Group 0
<ECCCNT_A_GCFU1ICL0_base>	FFC4 $C280_H$	P-Bus Group 0
<ECCCNT_A_GCFU1DCL0_base>	FFC4 $C300_H$	P-Bus Group 0
<ECCCNT_SA_PE2CL1_base>*1	FFC4 $C400_H$	P-Bus Group 0
<ECCCNT_A_GCFU2ICL1_base>*1	FFC4 $C480_H$	P-Bus Group 0
<ECCCNT_A_GCFU2DCL1_base>*1	FFC4 $C500_H$	P-Bus Group 0
<ECCCNT_SA_PE3CL1_base>*1	FFC4 $C600_H$	P-Bus Group 0
<ECCCNT_A_GCFU3ICL1_base>*1	FFC4 $C680_H$	P-Bus Group 0
<ECCCNT_A_GCFU3DCL1_base>*1	FFC4 $C700_H$	P-Bus Group 0
<ECCCNT_A_X2VCL0_base>	FFC4 $D000_H$	P-Bus Group 0
<ECCCNT_A_X2VCL1_base>*1	FFC4 $D080_H$	P-Bus Group 0
<ECCCNT_A_BARR_base>	FFC4 $D200_H$	P-Bus Group 0
<ECCCNT_A_IPIR_base>	FFC4 $D280_H$	P-Bus Group 0
<ECCCNT_A_TPTM_base>	FFC4 $D380_H$	P-Bus Group 0
<ECCCNT_A_CRAM_base>	FFC4 $D400_H$	P-Bus Group 0
<ECCCNT_A_SG0_base>	FFC4 $D480_H$	P-Bus Group 0
<ECCCNT_A_SX2PV_base>	FFC4 $D500_H$	P-Bus Group 0
<ECCCNT_A_SX2FX_base>	FFC4 $D580_H$	P-Bus Group 0
<ECCCNT_A_FX2SX_base>*2	FFC4 $D600_H$	P-Bus Group 0
<ECCCNT_A_GCFUF_base>	FFC4 $D680_H$	P-Bus Group 0
<ECCCNT_A_SX2MB_base>	FFC4 $D700_H$	P-Bus Group 0
<ECCCNT_A_V2A1_base>	FFC6 3800_H	P-Bus Group 1
<ECCCNT_A_V2A2_base>	FFDE 0000_H	P-Bus Group 2L
<ECCCNT_A_V2A3_base>	FFC7 2000_H	P-Bus Group 3
<ECCCNT_A_V2A4_base>	FFC7 4000_H	P-Bus Group 4
<ECCCNT_A_V2A5_base>	FFC7 9000_H	P-Bus Group 5
<ECCCNT_A_V2A6_base>	FF75 0000_H	P-Bus Group 6H
<ECCCNT_A_V2A7_base>	FFF4 8000_H	P-Bus Group 7
<ECCCNT_A_V2A8_base>	FFF2 8000_H	P-Bus Group 8

Table 44.204 Register Base Addresses (2/3)

Base Address Name	Base Address	Bus Group
<ECCCNT_A_V2A9_base>	FF0A 0000$_H$	P-Bus Group 9
<ECCCNT_D_V2XWCL0_base>	FFC4 E000$_H$	P-Bus Group 0
<ECCCNT_D_V2XRCL0_base>	FFC4 E080$_H$	P-Bus Group 0
<ECCCNT_D_V2XWCL1_base>*1	FFC4 E100$_H$	P-Bus Group 0
<ECCCNT_D_V2XRCL1_base>*1	FFC4 E180$_H$	P-Bus Group 0
<ECCCNT_D_PV2APBW_base>	FFC4 E400$_H$	P-Bus Group 0
<ECCCNT_D_CRAM_base>	FFC4 E500$_H$	P-Bus Group 0
<ECCCNT_D_EMU_base>*2	FFC4 E580$_H$	P-Bus Group 0
<ECCCNT_D_DMDE0_base>	FFC4 E600$_H$	P-Bus Group 0
<ECCCNT_D_DMDE1_base>	FFC4 E680$_H$	P-Bus Group 0
<ECCCNT_SD_PE0CL0_base>	FFC4 F000$_H$	P-Bus Group 0
<ECCCNT_MD_PE0CL0_base>	FFC4 F080$_H$	P-Bus Group 0
<ECCCNT_SD_PE1CL0_base>	FFC4 F100$_H$	P-Bus Group 0
<ECCCNT_MD_PE1CL0_base>	FFC4 F180$_H$	P-Bus Group 0
<ECCCNT_SD_PE2CL1_base>*1	FFC4 F200$_H$	P-Bus Group 0
<ECCCNT_MD_PE2CL1_base>*1	FFC4 F280$_H$	P-Bus Group 0
<ECCCNT_SD_PE3CL1_base>*1	FFC4 F300$_H$	P-Bus Group 0
<ECCCNT_MD_PE3CL1_base>*1	FFC4 F380$_H$	P-Bus Group 0
<ECCCNT_D_BARR_base>	FFC4 F800$_H$	P-Bus Group 0
<ECCCNT_D_IPIR_base>	FFC4 F880$_H$	P-Bus Group 0
<ECCCNT_D_TPTM_base>	FFC4 F980$_H$	P-Bus Group 0
<ECCCNT_D_DTS_base>	FFC4 FA00$_H$	P-Bus Group 0
<ECCCNT_D_DMA0_base>	FFC4 FB00$_H$	P-Bus Group 0
<ECCCNT_D_DMA1_base>	FFC4 FB80$_H$	P-Bus Group 0
<ECCCNT_D_V2A1W_base>	FFC6 3900$_H$	P-Bus Group 1
<ECCCNT_D_V2A2W_base>	FFDE 0100$_H$	P-Bus Group 2L
<ECCCNT_D_V2A3W_base>	FFC7 2100$_H$	P-Bus Group 3
<ECCCNT_D_V2A4W_base>	FFC7 4100$_H$	P-Bus Group 4
<ECCCNT_D_V2A5W_base>	FFC7 9100$_H$	P-Bus Group 5
<ECCCNT_D_V2A6W_base>	FF75 0200$_H$	P-Bus Group 6H
<ECCCNT_D_V2A7W_base>	FFF4 8400$_H$	P-Bus Group 7
<ECCCNT_D_V2A8W_base>	FFF2 8400$_H$	P-Bus Group 8
<ECCCNT_D_V2A9W_base>	FF0A 0200$_H$	P-Bus Group 9
<BECCCAP_LRAM_base>	FFC5 4000$_H$	P-Bus Group 0
<BECCCAP_CRAM_base>	FFC5 4100$_H$	P-Bus Group 0
<BECCCAP_CFL_base>	FFC5 4300$_H$	P-Bus Group 0
<BECCCAP_PERI_base>	FFC5 4400$_H$	P-Bus Group 0
<BECCCAP_DMDT_base>	FFC5 4500$_H$	P-Bus Group 0
<BECCCAP_EMU_base>	FFC5 4800$_H$	P-Bus Group 0
<PB1ECC_base>	FFC6 2F60$_H$	P-Bus Group 1
<PB2ECC_base>	FFDE 0900$_H$	P-Bus Group 2L
<PB3ECC_base>	FFC7 2900$_H$	P-Bus Group 3
<PB4ECC_base>	FFC7 5100$_H$	P-Bus Group 4
<PB5ECC_base>	FFC7 A100$_H$	P-Bus Group 5

Table 44.204 Register Base Addresses (3/3)

Base Address Name	Base Address	Bus Group
<PB6ECC_base>	FF75 0900$_H$	P-Bus Group 6H
<PB7ECC_base>	FFF4 9100$_H$	P-Bus Group 7
<PB8ECC_base>	FFF2 9100$_H$	P-Bus Group 8
<PB9ECC_base>	FF0A 1100$_H$	P-Bus Group 9
<HB91MECC_base>*2	FF0C 1800$_H$	P-Bus Group 9 (RHSIF0 Master)
<HB91SECC_base>*2	FF0C 1C00$_H$	P-Bus Group 9 (RHSIF0 Slave)
<HB92MECC_base>	FF0C 0000$_H$	P-Bus Group 9 (FlexRay0 Master)
<HB92SECC_base>	FF0C 0400$_H$	P-Bus Group 9 (FlexRay0 Slave)
<HB93MECC_base>*2	FF0C 0800$_H$	P-Bus Group 9 (FlexRay1 Master)
<HB93SECC_base>*2	FF0C 0C00$_H$	P-Bus Group 9 (FlexRay1 Slave)
<HB94MECC_base>*2	FF0C 1400$_H$	P-Bus Group 9 (ETNB1 Gb Ethernet Master)
<HB95MECC_base>*3	FF0C 1000$_H$	P-Bus Group 9 (ETNB0 Fast Ethernet Master)
<HB96SECC_base>*4	FF0C 2000$_H$	P-Bus Group 9 (SFMA)
<BECCCAP_V2A1_base>	FFC6 3A00$_H$	P-Bus Group 1
<BECCCAP_V2A2_base>	FFDE 0200$_H$	P-Bus Group 2L
<BECCCAP_V2A3_base>	FFC7 2200$_H$	P-Bus Group 3
<BECCCAP_V2A4_base>	FFC7 4200$_H$	P-Bus Group 4
<BECCCAP_V2A5_base>	FFC7 9200$_H$	P-Bus Group 5
<BECCCAP_V2A6_base>	FF75 0400$_H$	P-Bus Group 6H
<BECCCAP_V2A7_base>	FFF4 8800$_H$	P-Bus Group 7
<BECCCAP_V2A8_base>	FFF2 8800$_H$	P-Bus Group 8
<BECCCAP_V2A9_base>	FF0A 0400$_H$	P-Bus Group 9

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A6.
Note 3. This function is not implemented in RH850/U2A6 (156 pins).
Note 4. This function is not implemented in RH850/U2A6 (156/144 pins).

44.3.11.3 List of Registers

Table 44.205 List of Registers (1/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_A_CCIBnCL0 (n = 0, 1)	ECC control register	BUSAECCCTL	<ECCCNT_A_CCIBnCL0_base> + 00_H	8, 16, 32	PBG00#3	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_CCIBnCL0_base> + 10_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_CCIBnCL0_base> + 30_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_CCIBnCL0_base> + 40_H	8, 16, 32	PBG00#3	ECCKCPROT
ECCCNT_A_CCIBnCL1 (n = 0, 1)*5	ECC control register	BUSAECCCTL	<ECCCNT_A_CCIBnCL1_base> + 00_H	8, 16, 32	PBG00#4	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_CCIBnCL1_base> + 10_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_CCIBnCL1_base> + 30_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_CCIBnCL1_base> + 40_H	8, 16, 32	PBG00#4	ECCKCPROT
ECCCNT_SA_PEnCL0 (n = 0, 1)	ECC control register	BUSAECCCTL	<ECCCNT_SA_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_SA_PEnCL0_base> + 10_H	8, 16, 32	*1	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_SA_PEnCL0_base> + 30_H	8, 16, 32	*1	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_SA_PEnCL0_base> + 40_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_A_GCFUnICL0 (n = 0, 1)	ECC control register	BUSAECCCTL	<ECCCNT_A_GCFUnICL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_GCFUnICL0_base> + 10_H	8, 16, 32	*1	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_GCFUnICL0_base> + 30_H	8, 16, 32	*1	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_GCFUnICL0_base> + 40_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_A_GCFUnDCL0 (n = 0, 1)	ECC control register	BUSAECCCTL	<ECCCNT_A_GCFUnDCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_GCFUnDCL0_base> + 10_H	8, 16, 32	*1	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_GCFUnDCL0_base> + 30_H	8, 16, 32	*1	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_GCFUnDCL0_base> + 40_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_SA_PEnCL1 (n = 2, 3)*5	ECC control register	BUSAECCCTL	<ECCCNT_SA_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_SA_PEnCL1_base> + 10_H	8, 16, 32	*2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_SA_PEnCL1_base> + 30_H	8, 16, 32	*2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_SA_PEnCL1_base> + 40_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_A_GCFUnICL1 (n = 2, 3)*5	ECC control register	BUSAECCCTL	<ECCCNT_A_GCFUnICL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_GCFUnICL1_base> + 10_H	8, 16, 32	*2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_GCFUnICL1_base> + 30_H	8, 16, 32	*2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_GCFUnICL1_base> + 40_H	8, 16, 32	*2	ECCKCPROT

Table 44.205 List of Registers (2/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_A_GCFUnDCL1 (n = 2, 3)*5	ECC control register	BUSAECCCTL	<ECCCNT_A_GCFUnDCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_GCFUnDCL1_base> + 10_H	8, 16, 32	*2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_GCFUnDCL1_base> + 30_H	8, 16, 32	*2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_GCFUnDCL1_base> + 40_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_A_X2VCL0	ECC control register	BUSAECCCTL	<ECCCNT_A_X2VCL0_base> + 00_H	8, 16, 32	PBG00#3	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_X2VCL0_base> + 10_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_X2VCL0_base> + 30_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_X2VCL0_base> + 40_H	8, 16, 32	PBG00#3	ECCKCPROT
ECCCNT_A_X2VCL1*5	ECC control register	BUSAECCCTL	<ECCCNT_A_X2VCL1_base> + 00_H	8, 16, 32	PBG00#4	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_X2VCL1_base> + 10_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_X2VCL1_base> + 30_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_X2VCL1_base> + 40_H	8, 16, 32	PBG00#4	ECCKCPROT
ECCCNT_A_BARR	ECC control register	BUSAECCCTL	<ECCCNT_A_BARR_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_BARR_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_BARR_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_BARR_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_IPIR	ECC control register	BUSAECCCTL	<ECCCNT_A_IPIR_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_IPIR_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_IPIR_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_IPIR_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_TPTM	ECC control register	BUSAECCCTL	<ECCCNT_A_TPTM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_TPTM_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_TPTM_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_TPTM_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_CRAM	ECC control register	BUSAECCCTL	<ECCCNT_A_CRAM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_CRAM_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_CRAM_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_CRAM_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_SG0	ECC control register	BUSAECCCTL	<ECCCNT_A_SG0_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_SX2PV	ECC control register	BUSAECCCTL	<ECCCNT_A_SX2PV_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_SX2PV_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_SX2PV_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_SX2PV_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_SX2FX	ECC control register	BUSAECCCTL	<ECCCNT_A_SX2FX_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_SX2FX_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_SX2FX_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_SX2FX_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT

Table 44.205 List of Registers (3/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_A_FX2SX[*6]	ECC control register	BUSAECCCTL	<ECCCNT_A_FX2SX_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_FX2SX_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_FX2SX_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_FX2SX_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_GCFUF	ECC control register	BUSAECCCTL	<ECCCNT_A_GCFUF_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_GCFUF_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_GCFUF_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_GCFUF_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_SX2MB	ECC control register	BUSAECCCTL	<ECCCNT_A_SX2MB_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSAECCTSTCTL	<ECCCNT_A_SX2MB_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSAECCTSTDIN0	<ECCCNT_A_SX2MB_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSAECCTSTEIN	<ECCCNT_A_SX2MB_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_A_V2An (n = 1 to 9)	ECC control register	BUSAECCCTL	<ECCCNT_A_V2An_base> + 00_H	8, 16, 32	[*3]	APECKCPROT[*4]
ECCCNT_D_V2XWCL0	ECC control register	BUSDVCECCCTL	<ECCCNT_D_V2XWCL0_base> + 00_H	8, 16, 32	PBG00#3	ECCKCPROT
	ECC test control register	BUSDVCECCTSTCTL	<ECCCNT_D_V2XWCL0_base> + 10_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test data input register	BUSDVCECCTSTDIN0	<ECCCNT_D_V2XWCL0_base> + 30_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test ECC input register	BUSDVCECCTSTEIN	<ECCCNT_D_V2XWCL0_base> + 40_H	8, 16, 32	PBG00#3	ECCKCPROT
ECCCNT_D_V2XRCL0	ECC control register	BUSDVCECCCTL	<ECCCNT_D_V2XRCL0_base> + 00_H	8, 16, 32	PBG00#3	ECCKCPROT
	ECC test control register	BUSDVCECCTSTCTL	<ECCCNT_D_V2XRCL0_base> + 10_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test data input register	BUSDVCECCTSTDIN0	<ECCCNT_D_V2XRCL0_base> + 30_H	8, 16, 32	PBG00#3	ECCKCPROT
	Test ECC input register	BUSDVCECCTSTEIN	<ECCCNT_D_V2XRCL0_base> + 40_H	8, 16, 32	PBG00#3	ECCKCPROT
ECCCNT_D_V2XWCL1[*5]	ECC control register	BUSDVCECCCTL	<ECCCNT_D_V2XWCL1_base> + 00_H	8, 16, 32	PBG00#4	ECCKCPROT
	ECC test control register	BUSDVCECCTSTCTL	<ECCCNT_D_V2XWCL1_base> + 10_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test data input register	BUSDVCECCTSTDIN0	<ECCCNT_D_V2XWCL1_base> + 30_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test ECC input register	BUSDVCECCTSTEIN	<ECCCNT_D_V2XWCL1_base> + 40_H	8, 16, 32	PBG00#4	ECCKCPROT
ECCCNT_D_V2XRCL1[*5]	ECC control register	BUSDVCECCCTL	<ECCCNT_D_V2XRCL1_base> + 00_H	8, 16, 32	PBG00#4	ECCKCPROT
	ECC test control register	BUSDVCECCTSTCTL	<ECCCNT_D_V2XRCL1_base> + 10_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test data input register	BUSDVCECCTSTDIN0	<ECCCNT_D_V2XRCL1_base> + 30_H	8, 16, 32	PBG00#4	ECCKCPROT
	Test ECC input register	BUSDVCECCTSTEIN	<ECCCNT_D_V2XRCL1_base> + 40_H	8, 16, 32	PBG00#4	ECCKCPROT
ECCCNT_D_PV2APBW	ECC control register	BUSDECCCTL	<ECCCNT_D_PV2APBW_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT

Table 44.205 List of Registers (4/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECCCNT_D_CRAM	ECC control register	BUSDVCECCCTL	<ECCCNT_D_CRAM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSDVCECCTSTCTL	<ECCCNT_D_CRAM_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSDVCECCTSTDIN0	<ECCCNT_D_CRAM_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSDVCECCTSTEIN	<ECCCNT_D_CRAM_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_EMU*6	ECC control register	BUSDEMECCCTL	<ECCCNT_D_EMU_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSDEMECCTSTCTL	<ECCCNT_D_EMU_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSDEMECCTSTDIN0	<ECCCNT_D_EMU_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSDEMECCTSTEIN	<ECCCNT_D_EMU_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_DMDEn (n = 0, 1)	ECC control register	BUSDDEECCCTL	<ECCCNT_D_DMDEn_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	ECC test control register	BUSDDEECCTSTCTL	<ECCCNT_D_DMDEn_base> + 10_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSDDEECCTSTDIN0	<ECCCNT_D_DMDEn_base> + 30_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test data input register	BUSDDEECCTSTDIN1	<ECCCNT_D_DMDEn_base> + 34_H	8, 16, 32	PBG00#2	ECCKCPROT
	Test ECC input register	BUSDDEECCTSTEIN	<ECCCNT_D_DMDEn_base> + 40_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_SD_PEnCL0 (n = 0, 1)	ECC control register	BUSDECCCTL	<ECCCNT_SD_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_MD_PEnCL0 (n = 0, 1)	ECC control register	BUSDECCCTL	<ECCCNT_MD_PEnCL0_base> + 00_H	8, 16, 32	*1	ECCKCPROT
ECCCNT_SD_PEnCL1 (n = 2, 3)*5	ECC control register	BUSDECCCTL	<ECCCNT_SD_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_MD_PEnCL1 (n = 2, 3)*5	ECC control register	BUSDECCCTL	<ECCCNT_MD_PEnCL1_base> + 00_H	8, 16, 32	*2	ECCKCPROT
ECCCNT_D_BARR	ECC control register	BUSDECCCTL	<ECCCNT_D_BARR_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_IPIR	ECC control register	BUSDECCCTL	<ECCCNT_D_IPIR_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_TPTM	ECC control register	BUSDECCCTL	<ECCCNT_D_TPTM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_DTS	ECC control register	BUSDECCCTL	<ECCCNT_D_DTS_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_DMAn (n = 0, 1)	ECC control register	BUSDECCCTL	<ECCCNT_D_DMAn_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
ECCCNT_D_V2AnW (n = 1 to 9)	ECC control register	BUSDVCECCCTL	<ECCCNT_D_V2AnW_base> + 00_H	8, 16, 32	*3	APECKCPROT*4
BECCCAP_LRAM	Error notification control register	LR_BUSERRINT	<BECCCAP_LRAM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Address 1-bit error status clear register	LR_BUSASSTCLR	<BECCCAP_LRAM_base> + 10_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status clear register	LR_BUSADSTCLR	<BECCCAP_LRAM_base> + 20_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status clear register	LR_BUSDSSTCLR	<BECCCAP_LRAM_base> + 30_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status clear register	LR_BUSDDSTCLR	<BECCCAP_LRAM_base> + 40_H	8, 16, 32	PBG00#2	—
	Address 1-bit error status register	LR_BUSASERSTR	<BECCCAP_LRAM_base> + 50_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status register	LR_BUSADERSTR	<BECCCAP_LRAM_base> + 60_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status register	LR_BUSDSERSTR	<BECCCAP_LRAM_base> + 70_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status register	LR_BUSDDERSTR	<BECCCAP_LRAM_base> + 80_H	8, 16, 32	PBG00#2	—

Table 44.205 List of Registers (5/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
BECCCAP_CRAM	Error notification control register	CR_BUSERRINT	<BECCCAP_CRAM_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Address 1-bit error status clear register	CR_BUSASSTCLR	<BECCCAP_CRAM_base> + 10_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status clear register	CR_BUSADSTCLR	<BECCCAP_CRAM_base> + 20_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status clear register	CR_BUSDSSTCLR	<BECCCAP_CRAM_base> + 30_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status clear register	CR_BUSDDSTCLR	<BECCCAP_CRAM_base> + 40_H	8, 16, 32	PBG00#2	—
	Address 1-bit error status register	CR_BUSASERSTR	<BECCCAP_CRAM_base> + 50_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status register	CR_BUSADERSTR	<BECCCAP_CRAM_base> + 60_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status register	CR_BUSDSERSTR	<BECCCAP_CRAM_base> + 70_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status register	CR_BUSDDERSTR	<BECCCAP_CRAM_base> + 80_H	8, 16, 32	PBG00#2	—
BECCCAP_CFL	Error notification control register	CF_BUSERRINT	<BECCCAP_CFL_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Address 1-bit error status clear register	CF_BUSASSTCLR	<BECCCAP_CFL_base> + 10_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status clear register	CF_BUSADSTCLR	<BECCCAP_CFL_base> + 20_H	8, 16, 32	PBG00#2	—
	Address 1-bit error status register	CF_BUSASERSTR	<BECCCAP_CFL_base> + 50_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status register	CF_BUSADERSTR	<BECCCAP_CFL_base> + 60_H	8, 16, 32	PBG00#2	—
BECCCAP_PERI	Error notification control register	PH_BUSERRINT	<BECCCAP_PERI_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Address 1-bit error status clear register	PH_BUSASSTCLR	<BECCCAP_PERI_base> + 10_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status clear register	PH_BUSADSTCLR	<BECCCAP_PERI_base> + 20_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status clear register	PH_BUSDSSTCLR	<BECCCAP_PERI_base> + 30_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status clear register	PH_BUSDDSTCLR	<BECCCAP_PERI_base> + 40_H	8, 16, 32	PBG00#2	—
	Address 1-bit error status register	PH_BUSASERSTR	<BECCCAP_PERI_base> + 50_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status register	PH_BUSADERSTR	<BECCCAP_PERI_base> + 60_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status register	PH_BUSDSERSTR	<BECCCAP_PERI_base> + 70_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status register	PH_BUSDDERSTR	<BECCCAP_PERI_base> + 80_H	8, 16, 32	PBG00#2	—
BECCCAP_DMDT	Error notification control register	DM_BUSERRINT	<BECCCAP_DMDT_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Data 1-bit error status clear register	DM_BUSDSSTCLR	<BECCCAP_DMDT_base> + 30_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status clear register	DM_BUSDDSTCLR	<BECCCAP_DMDT_base> + 40_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status register	DM_BUSDSERSTR	<BECCCAP_DMDT_base> + 70_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status register	DM_BUSDDERSTR	<BECCCAP_DMDT_base> + 80_H	8, 16, 32	PBG00#2	—
BECCCAP_EMU	Error notification control register	EM_BUSERRINT	<BECCCAP_EMU_base> + 00_H	8, 16, 32	PBG00#2	ECCKCPROT
	Address 1-bit error status clear register	EM_BUSASSTCLR	<BECCCAP_EMU_base> + 10_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status clear register	EM_BUSADSTCLR	<BECCCAP_EMU_base> + 20_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status clear register	EM_BUSDSSTCLR	<BECCCAP_EMU_base> + 30_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status clear register	EM_BUSDDSTCLR	<BECCCAP_EMU_base> + 40_H	8, 16, 32	PBG00#2	—
	Address 1-bit error status register	EM_BUSASERSTR	<BECCCAP_EMU_base> + 50_H	8, 16, 32	PBG00#2	—
	Address 2-bit error status register	EM_BUSADERSTR	<BECCCAP_EMU_base> + 60_H	8, 16, 32	PBG00#2	—
	Data 1-bit error status register	EM_BUSDSERSTR	<BECCCAP_EMU_base> + 70_H	8, 16, 32	PBG00#2	—
	Data 2-bit error status register	EM_BUSDDERSTR	<BECCCAP_EMU_base> + 80_H	8, 16, 32	PBG00#2	—
PBnECC (n = 1 to 9)	Key code protection register	APECKCPROT	<PBnECC_base> + 00_H	32	*3	—

Table 44.205 List of Registers (6/6)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
HBnECC*7 (n = 91M/S, 92M/S, 93M/S, 94M, 95M, 96S)	ECC control register	HBECECCCTL	<HBnECC_base> + 00_H	16, 32	PBG90#1	HBECKCPROT
	Error notification control register	HBECERRINT	<HBnECC_base> + 04_H	8, 16, 32	PBG90#1	HBECKCPROT
	Error status clear register	HBECSTCLR	<HBnECC_base> + 08_H	8, 16, 32	PBG90#1	—
	Error status register	HBEC1STERSTR	<HBnECC_base> + 10_H	8, 16, 32	PBG90#1	—
	ECC test control register	HBECTSTCTL	<HBnECC_base> + 150_H	8, 16, 32	PBG90#1	HBECKCPROT
	Test data input register	HBECTSTDIN0	<HBnECC_base> + 154_H	8, 16, 32	PBG90#1	HBECKCPROT
	Test ECC input register	HBECTSTEIN	<HBnECC_base> + 158_H	8, 16, 32	PBG90#1	HBECKCPROT
	Key code protection register	HBECKCPROT	<HBnECC_base> + $15C_H$	32	PBG90#1	—
BECCCAP_V2An (n = 1 to 9)	Error notification control register	PHC_BUSERRINT	<BECCCAP_V2An_base> + 00_H	8, 16, 32	*3	APECKCPROT*4
	Address 1-bit error status clear register	PHC_BUSASSTCLR	<BECCCAP_V2An_base> + 10_H	8, 16, 32	*3	—
	Address 2-bit error status clear register	PHC_BUSADSTCLR	<BECCCAP_V2An_base> + 20_H	8, 16, 32	*3	—
	Data 1-bit error status clear register	PHC_BUSDSSTCLR	<BECCCAP_V2An_base> + 30_H	8, 16, 32	*3	—
	Data 2-bit error status clear register	PHC_BUSDDSTCLR	<BECCCAP_V2An_base> + 40_H	8, 16, 32	*3	—
	Address 1-bit error status register	PHC_BUSASERSTR	<BECCCAP_V2An_base> + 50_H	8, 16, 32	*3	—
	Address 2-bit error status register	PHC_BUSADERSTR	<BECCCAP_V2An_base> + 60_H	8, 16, 32	*3	—
	Data 1-bit error status register	PHC_BUSDSERSTR	<BECCCAP_V2An_base> + 70_H	8, 16, 32	*3	—
	Data 2-bit error status register	PHC_BUSDDERSTR	<BECCCAP_V2An_base> + 80_H	8, 16, 32	*3	—

Note 1. n=0: PBG01#0
n=1: PBG01#1

Note 2. n=2: PBG01#2
n=3: PBG01#3

Note 3. n=1: PBG10#1
n=2: PBG20#1
n=3: PBG30#1
n=4: PBG40#1
n=5: PBG50#1
n=6: PBG6H0#1
n=7: PBG70#1
n=8: PBG80#1
n=9: PBG90#1

Note 4. For details of which P-Bus group each module belongs to, see **Table 44.204, Register Base Addresses**.

Note 5. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 6. This function is not implemented in RH850/U2A6.

Note 7. This function is not implemented in RH850/U2A6 with index n =91, 93, 94, in RH850/U2A6 (156 pins) with index n = 95 and RH850/U2A6 (156/144 pins) with index n = 96.

Table 44.206 The relation between the module storing the error and the error source for Data Transfer Path (1/5)

Protection target	Function	Controlled by	Captured to	Applicable for
Address to CodeFlash at CCIB(bankA)	Address EDC decode	ECCCNT_A_CCIB0CL0	BECCCAP_CFL	Request address to CodeFlash area
Address to CodeFlash at CCIB(bankB)		ECCCNT_A_CCIB1CL0		
Address to CodeFlash at CCIB(bankC)*3		ECCCNT_A_CCIB0CL1		
Address to CodeFlash at CCIB(bankD)*3		ECCCNT_A_CCIB1CL1		
Address to SAXI2FAXI (bridge from System Bus to Global Flash Bus)		ECCCNT_A_SX2FX		
Address to VCI slave I/F in PE0	Address EDC decode	ECCCNT_SA_PE0CL0	BECCCAP_LRAM	Request address to PE0 area
Address to VCI slave I/F in PE1		ECCCNT_SA_PE1CL0		Request address to PE1 area
Address to VCI slave I/F in PE2*3		ECCCNT_SA_PE2CL1		Request address to PE2 area
Address to VCI slave I/F in PE3*3		ECCCNT_SA_PE3CL1		Request address to PE3 area
Address to AXI2VCI0 (bridge from System Bus to Inter-processor element Bus)		ECCCNT_A_X2VCL0		Request address to PE0/PE1 area and I-Bus area
Address to AXI2VCI1 (bridge from System Bus to Inter-processor element Bus)*3		ECCCNT_A_X2VCL1		Request address to PE2/PE3 area
Data at VCI slave I/F in PE0	Data ECC decode	ECCCNT_SD_PE0CL0		Write data to PE0 area
Data at VCI slave I/F in PE1		ECCCNT_SD_PE1CL0		Write data to PE1 area
Data at VCI slave I/F in PE2*3		ECCCNT_SD_PE2CL1		Write data to PE2 area
Data at VCI slave I/F in PE3*3		ECCCNT_SD_PE3CL1		Write data to PE3 area

Table 44.206 The relation between the module storing the error and the error source for Data Transfer Path (2/5)

Protection target	Function	Controlled by	Captured to	Applicable for
Address to CRAM at CRAM controller 0	Address EDC decode	ECCCNT_A_CRAM	BECCCAP_CRAM	Request address to Cluster RAM0 area
Address to CRAM at CRAM controller 1*3		ECCCNT_A_CRAM		Request address to Cluster RAM1 area
Address to CRAM at CRAM controller 2*4		ECCCNT_A_CRAM		Request address to Cluster RAM2 area
Address to CRAM at CRAM controller 3		ECCCNT_A_CRAM		Request address to Cluster RAM3 area
Address to SAXI2MBI0 (bridge from System Bus to CRAM Bus)		ECCCNT_A_SX2MB		Request address to CRAM0 area
Address to SAXI2MBI1 (bridge from System Bus to CRAM Bus)*3		ECCCNT_A_SX2MB		Request address to CRAM1 area
Address to SAXI2MBI2 (bridge from System Bus to CRAM Bus)*4		ECCCNT_A_SX2MB		Request address to CRAM2 area
Address to SAXI2MBI3 (bridge from System Bus to CRAM Bus)		ECCCNT_A_SX2MB		Request address to CRAM3 area
Data in read-modify-write at CRAM controller 0	Data ECC decode	ECCCNT_D_CRAM		Write data to CRAM0 area in read-modify-write*1
Data in read-modify-write at CRAM controller 1*3		ECCCNT_D_CRAM		Write data to CRAM1 area in read-modify-write*1
Data in read-modify-write at CRAM controller 2*4		ECCCNT_D_CRAM		Write data to CRAM2 area in read-modify-write*1
Data in read-modify-write at CRAM controller 3		ECCCNT_D_CRAM		Write data to CRAM3 area in read-modify-write*1
Data from DataRAM at sDMAC0	Data ECC decode	ECCCNT_D_DMDE0	BECCCAP_DMDT	Read data from write buffer of sDMAC0
Data from DataRAM at sDMAC1		ECCCNT_D_DMDE1		Read data from write buffer of sDMAC1
Data at DTS		ECCCNT_D_DTS		Read data from anywhere during DTS transfer
Data at sDMAC0		ECCCNT_D_DMA0		Read data from anywhere during sDMAC0 transfer
Data at sDMAC1		ECCCNT_D_DMA1		Read data from anywhere during sDMAC1 transfer

Table 44.206 The relation between the module storing the error and the error source for Data Transfer Path (3/5)

Protection target	Function	Controlled by	Captured to	Applicable for
Address to IPIR	Address EDC decode	ECCCNT_A_IPIR	BECCCAP_PERI	Request address to IPIR area
Address to BarrierSync		ECCCNT_A_BARR		Request address to BarrierSync area
Address to TPTM		ECCCNT_A_TPTM		Request address to TPTM area
Address to VCI2APB0 (bridge to P-Bus group 0)		ECCCNT_A_SG0		Request address to P-Bus Group 0 area
Address to AXI2PVCI (bridge from System Bus to P-Bus)		ECCCNT_A_SX2PV		Request address to P-Bus area
Data at VCI master I/F in PE0	Data ECC decode	ECCCNT_MD_PE0CL0		Read data via Inter-processor element Bus requested by PE0 master
Data at VCI master I/F in PE1		ECCCNT_MD_PE1CL0		Read data via Inter-processor element Bus requested by PE1 master
Data at VCI master I/F in PE2*3		ECCCNT_MD_PE2CL1		Read data via Inter-processor element Bus requested by PE2 master
Data at VCI master I/F in PE3*3		ECCCNT_MD_PE3CL1		Read data via Inter-processor element Bus requested by PE3 master
Data at VCI2AXI0 (bridge from inter-processor element Bus to System Bus)	Data ECC decode	ECCCNT_D_V2XWCL0		Write data to SAXI (System Bus)s area requested by PE0 and PE1 in specific instructions*2
Data at VCI2AXI0 (bridge from inter-processor element Bus to System Bus)		ECCCNT_D_V2XRCL0		Read data from SAXI (System Bus) area requested by PE0 and PE1 in specific instructions*2
Data at VCI2AXI1 (bridge from inter-processor element Bus to System Bus)*3		ECCCNT_D_V2XWCL1		Write data to SAXI (System Bus) area requested by PE2 and PE3 in specific instructions*2
Data at VCI2AXI1 (bridge from inter-processor element Bus to System Bus)*3		ECCCNT_D_V2XRCL1		Read data from SAXI (System Bus) area requested by PE2 and PE3 in specific instructions*2
Data to VCI2APB0 (bridge to P-Bus group 0)	Data ECC decode	ECCCNT_D_PV2APBW		Write data to P-Bus Group 0 area
Data to IPIR		ECCCNT_D_IPIR		Write data to IPIR area
Data to BarrierSync		ECCCNT_D_BARR		Write data to BarrierSync area
Data to TPTM		ECCCNT_D_TPTM		Write data to TPTM area

Table 44.206 The relation between the module storing the error and the error source for Data Transfer Path (4/5)

Protection target	Function	Controlled by	Captured to	Applicable for
Address to CodeFlash at FAXI2SAXI (bridge from Global Flash Bus to System Bus)*4	Address EDC decode	ECCCNT_A_FX2SX	BECCCAP_EMU	Request address to FAXI2SAXI bridge
Address to GCFU for Global Flash Bus		ECCCNT_A_GCFUF		Request address to remap module from System Bus
Data at FAXI2SAXI (bridge from Global Flash Bus to System Bus)*4	Data ECC decode	ECCCNT_D_EMU		Read data to FAXI2SAXI bridge
Address to CodeFlash at GCFU(PE0/inst) for Local Flash Bus	Address EDC decode	ECCCNT_A_GCFU0ICL0		Request address to remap module for instruction from PE0
Address to CodeFlash at GCFU(PE0/data) for Local Flash Bus		ECCCNT_A_GCFU0DCL0		Request address to remap module for data read from PE0
Address to CodeFlash at GCFU(PE1/inst) for Local Flash Bus		ECCCNT_A_GCFU1ICL0		Request address to remap module for instruction from PE1
Address to CodeFlash at GCFU(PE1/data) for Local Flash Bus		ECCCNT_A_GCFU1DCL0		Request address to remap module for data read from PE1
Address to CodeFlash at GCFU(PE2/inst) for Local Flash Bus*3		ECCCNT_A_GCFU2ICL1		Request address to remap module for instruction from PE2
Address to CodeFlash at GCFU(PE2/data) for Local Flash Bus*3		ECCCNT_A_GCFU2DCL1		Request address to remap module for data read from PE2
Address to CodeFlash at GCFU(PE3/inst) for Local Flash Bus*3		ECCCNT_A_GCFU3ICL1		Request address to remap module for instruction from PE3
Address to CodeFlash at GCFU(PE3/data) for Local Flash Bus*3		ECCCNT_A_GCFU3DCL1		Request address to remap module for data read from PE3
Address to VCI2APB1 (bridge to P-Bus group 1)	Address EDC decode	ECCCNT_A_V2A1	BECCCAP_V2A1	Request address to P-Bus Group 1 area
Address to VCI2APB2 (bridge to P-Bus group 2)		ECCCNT_A_V2A2	BECCCAP_V2A2	Request address to P-Bus Group 2 area
Address to VCI2APB3 (bridge to P-Bus group 3)		ECCCNT_A_V2A3	BECCCAP_V2A3	Request address to P-Bus Group 3 area
Address to VCI2APB4 (bridge to P-Bus group 4)		ECCCNT_A_V2A4	BECCCAP_V2A4	Request address to P-Bus Group 4 area
Address to VCI2APB5 (bridge to P-Bus group 5)		ECCCNT_A_V2A5	BECCCAP_V2A5	Request address to P-Bus Group 5 area
Address to VCI2APB6 (bridge to P-Bus group 6)		ECCCNT_A_V2A6	BECCCAP_V2A6	Request address to P-Bus Group 6 area
Address to VCI2APB7 (bridge to P-Bus group 7)		ECCCNT_A_V2A7	BECCCAP_V2A7	Request address to P-Bus Group 7 area
Address to VCI2APB8 (bridge to P-Bus group 8)		ECCCNT_A_V2A8	BECCCAP_V2A8	Request address to P-Bus Group 8 area
Address to VCI2APB9 (bridge to P-Bus group 9)		ECCCNT_A_V2A9	BECCCAP_V2A9	Request address to P-Bus Group 9 area

Table 44.206 The relation between the module storing the error and the error source for Data Transfer Path (5/5)

Protection target	Function	Controlled by	Captured to	Applicable for
Data to VCI2APB1 (bridge to P-Bus group 1)	Data ECC decode	ECCCNT_D_V2A1W	BECCCAP_V2A1	Write data to P-Bus Group 1
Data to VCI2APB2 (bridge to P-Bus group 2)		ECCCNT_D_V2A2W	BECCCAP_V2A2	Write data to P-Bus Group 2
Data to VCI2APB3 (bridge to P-Bus group 3)		ECCCNT_D_V2A3W	BECCCAP_V2A3	Write data to P-Bus Group 3
Data to VCI2APB4 (bridge to P-Bus group 4)		ECCCNT_D_V2A4W	BECCCAP_V2A4	Write data to P-Bus Group 4
Data to VCI2APB5 (bridge to P-Bus group 5)		ECCCNT_D_V2A5W	BECCCAP_V2A5	Write data to P-Bus Group 5
Data to VCI2APB6 (bridge to P-Bus group 6)		ECCCNT_D_V2A6W	BECCCAP_V2A6	Write data to P-Bus Group 6
Data to VCI2APB7 (bridge to P-Bus group 7)		ECCCNT_D_V2A7W	BECCCAP_V2A7	Write data to P-Bus Group 7
Data to VCI2APB8 (bridge to P-Bus group 8)		ECCCNT_D_V2A8W	BECCCAP_V2A8	Write data to P-Bus Group 8
Data to VCI2APB9 (bridge to P-Bus group 9)		ECCCNT_D_V2A9W	BECCCAP_V2A9	Write data to P-Bus Group 9
Address to RHSIF0*4	Address EDC decode	HB91SECC	HB91SECC	Address to RHSIF0
Write Data to RHSIF0*4	Data ECC decode			Write data to RHSIF0
Read data to RHSIF0*4	Data ECC decode	HB91MECC	HB91MECC	Read data requested by RHSIF0
Address to FlexRay0	Address EDC decode	HB92SECC	HB92SECC	Address to FlexRay0
Write Data to FlexRay0	Data ECC decode			Write data to FlexRay0
Read data to FlexRay0	Data ECC decode	HB92MECC	HB92MECC	Read data requested by FlexRay0
Address to FlexRay1*4	Address EDC decode	HB93SECC	HB93SECC	Address to FlexRay1
Write Data to FlexRay1*4	Data ECC decode			Write data to FlexRay1
Read data to FlexRay1*4	Data ECC decode	HB93MECC	HB93MECC	Read data requested by FlexRay1
Address to SFMA*5	Address EDC decode	HB96SECC	HB96SECC	Address to SFMA
Write Data to SFMA*5	Data ECC decode			Write data to SFMA
Read data to Ethernet*4	Data ECC decode	HB94MECC	HB94MECC	Read data requested by Gb Ethernet
Read data to Fast Ethernet*6	Data ECC decode	HB95MECC	HB95MECC	Read data requested by Fast Ethernet

Note 1. Read-modify-write includes SET1, NOT1, CLR1, CAXI and write access less than 4 bytes. For example, 2-byte write access is included.

Note 2. An access by specific instructions(CLR1, NOT1, SET1 and CAXI) is targeted.

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 4. This function is not implemented in RH850/U2A6.

Note 5. This function is not implemented in RH850/U2A6 (156/144 pins).

Note 6. This function is not implemented in RH850/U2A6 (156 pins).

44.3.11.4 BUSAECCCTL — ECC Control Register

This register controls ECC error detection for requested address.

This register is protected by the Key Code Protection Register. For details whether this register is protected by either ECCKCPROT or APECKCPROT, see **Table 44.205, List of Registers**. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address:
<ECCCNT_A_CCIBnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_A_CCIBnCL1_base> + 00_H (n = 0, 1)
<ECCCNT_SA_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_SA_PEnCL1_base> + 00_H (n = 2, 3)
<ECCCNT_A_GCFUnICL0_base> + 00_H (n = 0, 1)
<ECCCNT_A_GCFUnICL1_base> + 00_H (n = 2, 3)
<ECCCNT_A_GCFUnDCL0_base> + 00_H (n = 0, 1)
<ECCCNT_A_GCFUnDCL1_base> + 00_H (n = 2, 3)
<ECCCNT_A_X2VCLn_base> + 00_H (n = 0, 1)
<ECCCNT_A_BARR_base> + 00_H
<ECCCNT_A_IPIR_base> + 00_H
<ECCCNT_A_TPTM_base> + 00_H
<ECCCNT_A_CRAM_base> + 00_H
<ECCCNT_A_SG0_base> + 00_H
<ECCCNT_A_SX2PV_base> + 00_H
<ECCCNT_A_SX2FX_base> + 00_H
<ECCCNT_A_FX2SX_base> + 00_H
<ECCCNT_A_GCFUF_base> + 00_H
<ECCCNT_A_SX2MB_base> + 00_H
<ECCCNT_A_V2An_base> + 00_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.207 BUSAECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	Address ECC disable bit Sets ECC error detection to enable/disable. 0: ECC error detection is enabled 1: ECC error detection is disabled

44.3.11.5 BUSAECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address:
<ECCCNT_A_CCIBnCL0_base> + 10_H (n = 0, 1)
<ECCCNT_A_CCIBnCL1_base> + 10_H (n = 0, 1)
<ECCCNT_SA_PEnCL0_base> + 10_H (n = 0, 1)
<ECCCNT_SA_PEnCL1_base> + 10_H (n = 2, 3)
<ECCCNT_A_GCFUnICL0_base> + 10_H (n = 0, 1)
<ECCCNT_A_GCFUnICL1_base> + 10_H (n = 2, 3)
<ECCCNT_A_GCFUnDCL0_base> + 10_H (n = 0, 1)
<ECCCNT_A_GCFUnDCL1_base> + 10_H (n = 2, 3)
<ECCCNT_A_X2VCLn_base> + 10_H (n = 0, 1)
<ECCCNT_A_BARR_base> + 10_H
<ECCCNT_A_IPIR_base> + 10_H
<ECCCNT_A_TPTM_base> + 10_H
<ECCCNT_A_CRAM_base> + 10_H
<ECCCNT_A_SX2PV_base> + 10_H
<ECCCNT_A_SX2FX_base> + 10_H
<ECCCNT_A_FX2SX_base> + 10_H
<ECCCNT_A_GCFUF_base> + 10_H
<ECCCNT_A_SX2MB_base> + 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 44.208 BUSAECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the address ECC decoder. 0: Normal mode 1: Test mode The values of BUSAECCTSTDIN0 and BUSAECCTSTEIN registers are provided to data and ECC fields of the ECC decoder in test mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.11.6 BUSAECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSAECCTSTDIN0 and BUSAECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address:
<ECCCNT_A_CCIBnCL0_base> + 30_H (n = 0, 1)
<ECCCNT_A_CCIBnCL1_base> + 30_H (n = 0, 1)
<ECCCNT_SA_PEnCL0_base> + 30_H (n = 0, 1)
<ECCCNT_SA_PEnCL1_base> + 30_H (n = 2, 3)
<ECCCNT_A_GCFUnICL0_base> + 30_H (n = 0, 1)
<ECCCNT_A_GCFUnICL1_base> + 30_H (n = 2, 3)
<ECCCNT_A_GCFUnDCL0_base> + 30_H (n = 0, 1)
<ECCCNT_A_GCFUnDCL1_base> + 30_H (n = 2, 3)
<ECCCNT_A_X2VCLn_base> + 30_H (n = 0,1)
<ECCCNT_A_BARR_base> + 30_H
<ECCCNT_A_IPIR_base> + 30_H
<ECCCNT_A_TPTM_base> + 30_H
<ECCCNT_A_CRAM_base> + 30_H
<ECCCNT_A_SX2PV_base> + 30_H
<ECCCNT_A_SX2FX_base> + 30_H
<ECCCNT_A_FX2SX_base> + 30_H
<ECCCNT_A_GCFUF_base> + 30_H
<ECCCNT_A_SX2MB_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.209 BUSAECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

44.3.11.7 BUSAECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSAECCTSTDIN0 and BUSAECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address:
<ECCCNT_A_CCIBnCL0_base> + 40_H (n = 0, 1)
<ECCCNT_A_CCIBnCL1_base> + 40_H (n = 0, 1)
<ECCCNT_SA_PEnCL0_base> + 40_H (n = 0, 1)
<ECCCNT_SA_PEnCL1_base> + 40_H (n = 2, 3)
<ECCCNT_A_GCFUnICL0_base> + 40_H (n = 0, 1)
<ECCCNT_A_GCFUnICL1_base> + 40_H (n = 2, 3)
<ECCCNT_A_GCFUnDCL0_base> + 40_H (n = 0,1)
<ECCCNT_A_GCFUnDCL1_base> + 40_H (n =2, 3)
<ECCCNT_A_X2VCLn_base> + 40_H (n=0,1)
<ECCCNT_A_BARR_base> + 40_H
<ECCCNT_A_IPIR_base> + 40_H
<ECCCNT_A_TPTM_base> + 40_H
<ECCCNT_A_CRAM_base> + 40_H
<ECCCNT_A_SX2PV_base> + 40_H
<ECCCNT_A_SX2FX_base> + 40_H
<ECCCNT_A_FX2SX_base> + 40_H
<ECCCNT_A_GCFUF_base> + 40_H
<ECCCNT_A_SX2MB_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.210 BUSAECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

44.3.11.8 BUSDVCECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction.

This register is protected by Key Code Protection Register. For details whether this register is protected by either ECCKCPROT or APECKCPROT, see **Table 44.205, List of Registers**. When KCE = 0, it cannot be written. It can be written only when KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_V2XWCLn_base> + 00_H (n = 0, 1)
<ECCCNT_D_V2XRCLn_base> + 00_H (n = 0, 1)
<ECCCNT_D_CRAM_base> + 00_H
<ECCCNT_D_V2AnW_base> + 00_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.211 BUSDVCECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when a 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.11.9 BUSDVCECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_V2XWCLn_base> + 10_H (n = 0, 1)
<ECCCNT_D_V2XRCLn_base> + 10_H (n = 0, 1)
<ECCCNT_D_CRAM_base> + 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.212 BUSDVCECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are provided to data and ECC fields of one of the ECC decoders along with DATSEL setting in test mode.
0	DATSEL	Selects ECC decoder to replace input data when ECCTST = 1. 0: ECC decoder for lower data 1: ECC decoder for upper data

44.3.11.10 BUSDVCECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_V2XWCLn_base> + 30_H (n = 0, 1)
<ECCCNT_D_V2XRCLn_base> + 30_H (n = 0, 1)
<ECCCNT_D_CRAM_base> + 30_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.213 BUSDVCECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

44.3.11.11 BUSDVCECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDVCECCTSTDIN0 and BUSDVCECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_V2XWCLn_base> + 40_H (n = 0, 1)
<ECCCNT_D_V2XRCLn_base> + 40_H (n = 0, 1)
<ECCCNT_D_CRAM_base> + 40_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.214 BUSDVCECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

44.3.11.12 BUSDEMECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for data read to FAXI2SAXI bridge.

This register is protected by the ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_EMU_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.215 BUSDEMECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.11.13 BUSDEMECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After ECC test mode (ECCTST = 1) setting, arbitrary data value can be input to ECC decoder.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_EMU_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DAT SEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.216 BUSDEMECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are provided to data and ECC field of one of the ECC decoders along with DATSEL setting in test mode.
0	DATSEL	Selects ECC decoder to replace input data under ECCTST = 1. 0: ECC decoder for lower data 1: ECC decoder for upper data

44.3.11.14 BUSDEMECCTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_EMU_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.217 BUSDEMECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to data field of the ECC decoder when ECCTST = 1.

44.3.11.15 BUSDEMECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDEMECCTSTDIN0 and BUSDEMECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_EMU_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.218 BUSDEMECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

44.3.11.16 BUSDDEECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction for internal buffer of sDMAC.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_DMDEn_base> + 00_H (n =0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.219 BUSDDEECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.11.17 BUSDDEECCTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After setting the ECC test mode (ECCTST = 1), arbitrary data value can be input to ECC decoder.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_DMDEn_base> + 10_H (n =0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 44.220 BUSDDEECCTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	Selects input data for the data ECC decoder. 0: Normal mode 1: Test mode The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are provided to data and ECC fields of the ECC decoder in test mode.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.11.18 BUSDDEECCTSTDIN0 — Test Lower Data Input Register

This register is used to inject errors into lower data field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_DMDEn_base> + 30_H (n =0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.221 BUSDDEECCTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to lower data field of the ECC decoder when ECCTST = 1.

44.3.11.19 BUSDDEECCTSTDIN1 — Test Upper Data Input Register

This register is used to inject errors into upper data field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_DMDEn_base> + 34_H (n =0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.222 BUSDDEECCTSTDIN1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the data ECC decoder. The data is provided to upper data field of the ECC decoder when ECCTST = 1.

44.3.11.20 BUSDDEECCTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of BUSDDEECCTSTDIN0, BUSDDEECCTSTDIN1 and BUSDDEECCTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_DMDEn_base> + 40_H (n =0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.223 BUSDDEECCTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	ECC[7:0]	Specify the test data for ECC decoder. The data is provided to ECC field of the ECC decoder when ECCTST = 1.

44.3.11.21 BUSDECCCTL — ECC Control Register

This register controls ECC error detection/correction and 1-bit error correction.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <ECCCNT_D_PV2APBW_base> + 00_H
<ECCCNT_SD_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_SD_PEnCL1_base> + 00_H (n = 2, 3)
<ECCCNT_MD_PEnCL0_base> + 00_H (n = 0, 1)
<ECCCNT_MD_PEnCL1_base> + 00_H (n = 2, 3)
<ECCCNT_D_BARR_base> + 00_H
<ECCCNT_D_IPIR_base> + 00_H
<ECCCNT_D_TPTM_base> + 00_H
<ECCCNT_D_DTS_base> + 00_H
<ECCCNT_D_DMAn_base> + 00_H (n = 0, 1)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.224 BUSDECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	Data ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	Data ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.11.22 LR_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 00_H

Value after reset: $0000\ 000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.225 LR_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.23 LR_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR17	ASSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR03	ASSTCLR02	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 44.226 LR_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	ASSTCLR17*1	Writing 1 to this bit clears ASEDF17 in LR_BUSASERSTR.
16	ASSTCLR16	Writing 1 to this bit clears ASEDF16 in LR_BUSASERSTR.
15 to 4	Reserved	When writing, write the value after reset.
3	ASSTCLR03*1	Writing 1 to this bit clears ASEDF03 in LR_BUSASERSTR.
2	ASSTCLR02*1	Writing 1 to this bit clears ASEDF02 in LR_BUSASERSTR.
1	ASSTCLR01	Writing 1 to this bit clears ASEDF01 in LR_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASEDF00 in LR_BUSASERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.24 LR_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTCLR17	ADSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADSTCLR03	ADSTCLR02	ADSTCLR01	ADSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 44.227 LR_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	ADSTCLR17*1	Writing 1 to this bit clears ADEDF17 in LR_BUSADERSTR.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in LR_BUSADERSTR.
15 to 4	Reserved	When writing, write the value after reset.
3	ADSTCLR03*1	Writing 1 to this bit clears ADEDF03 in LR_BUSADERSTR.
2	ADSTCLR02*1	Writing 1 to this bit clears ADEDF02 in LR_BUSADERSTR.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in LR_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in LR_BUSADERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.25 LR_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DSSTCLR03	DSSTCLR02	DSSTCLR01	DSSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 44.228 LR_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	DSSTCLR03*1	Writing 1 to this bit clears DSEDF03 in LR_BUSDSERSTR.
2	DSSTCLR02*1	Writing 1 to this bit clears DSEDF02 in LR_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in LR_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in LR_BUSDSERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.26 LR_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in LR_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DDSTCLR03	DDSTCLR02	DDSTCLR01	DDSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 44.229 LR_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	DDSTCLR03*1	Writing 1 to this bit clears DDEDF03 in LR_BUSDDERSTR.
2	DDSTCLR02*1	Writing 1 to this bit clears DDEDF02 in LR_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDEDF01 in LR_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in LR_BUSDDERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.27 LR_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, flags of the detected errors are all set.

This register can be cleared by LR_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 50_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF17	ASEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF03	ASEDF02	ASEDF01	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.230 LR_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	ASEDF17*1	Indicates that an address ECC 1-bit error was detected in request address to AXI2VCI bridge in cluster 1.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to AXI2VCI bridge in cluster 0.
15 to 4	Reserved	When read, the value after reset is returned.
3	ASEDF03*1	Indicates that an address ECC 1-bit error was detected in request address to PE3.
2	ASEDF02*1	Indicates that an address ECC 1-bit error was detected in request address to PE2.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to PE1.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.28 LR_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 60_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF17	ADEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF03	ADEDF02	ADEDF01	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.231 LR_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	ADEDF17*1	Indicates that an address ECC 2-bit error was detected in request address to AXI2VCI bridge in cluster 1.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to AXI2VCI bridge in cluster 0.
15 to 4	Reserved	When read, the value after reset is returned.
3	ADEDF03*1	Indicates that an address ECC 2-bit error was detected in request address to PE3.
2	ADEDF02*1	Indicates that an address ECC 2-bit error was detected in request address to PE2.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to PE1.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.29 LR_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 70_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF03	DSEDF02	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.232 LR_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	DSEDF03*1	Indicates that a data ECC 1-bit error was detected in write data to PE3.
2	DSEDF02*1	Indicates that a data ECC 1-bit error was detected in write data to PE2.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in write data to PE1.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.30 LR_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to Local RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by LR_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_LRAM_base> + 80_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DDEDF03	DDEDF02	DDEDF01	DDEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.233 LR_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	DDEDF03*1	Indicates that a data ECC 2-bit error has been detected in write data to PE3.
2	DDEDF02*1	Indicates that a data ECC 2-bit error has been detected in write data to PE2.
1	DDEDF01	Indicates that a data ECC 2-bit error has been detected in write data to PE1.
0	DDEDF00	Indicates that a data ECC 2-bit error has been detected in write data to PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.31 CR_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 00_H

Value after reset: $0000\ 000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.234 CR_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.32 CR_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ASSTCLR27	ASSTCLR26	ASSTCLR25	ASSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ASSTCLR14	—	—	—	ASSTCLR10	—	—	—	ASSTCLR06	ASSTCLR05	ASSTCLR04	—	ASSTCLR02	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	R	R	R	W	R	R	R	W	W	W	R	W	W	W

Table 44.235 CR_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When writing, write the value after reset.
27	ASSTCLR27	Writing 1 to this bit clears ASEDF27 in CR_BUSASERSTR.
26	ASSTCLR26*2	Writing 1 to this bit clears ASEDF26 in CR_BUSASERSTR.
25	ASSTCLR25*1	Writing 1 to this bit clears ASEDF25 in CR_BUSASERSTR.
24	ASSTCLR24	Writing 1 to this bit clears ASEDF24 in CR_BUSASERSTR.
23 to 15	Reserved	When writing, write the value after reset.
14	ASSTCLR14	Writing 1 to this bit clears ASEDF14 in CR_BUSASERSTR.
13 to 11	Reserved	When writing, write the value after reset.
10	ASSTCLR10*2	Writing 1 to this bit clears ASEDF10 in CR_BUSASERSTR.
9 to 7	Reserved	When writing, write the value after reset.
6	ASSTCLR06*1	Writing 1 to this bit clears ASEDF06 in CR_BUSASERSTR.
5	ASSTCLR05*1	Writing 1 to this bit clears ASEDF05 in CR_BUSASERSTR.
4	ASSTCLR04*1	Writing 1 to this bit clears ASEDF04 in CR_BUSASERSTR.
3	Reserved	When writing, write the value after reset.
2	ASSTCLR02	Writing 1 to this bit clears ASEDF02 in CR_BUSASERSTR.
1	ASSTCLR01	Writing 1 to this bit clears ASEDF01 in CR_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASEDF00 in CR_BUSASERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.33 CR_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ADSTCLR27	ADSTCLR26	ADSTCLR25	ADSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADSTCLR14	—	—	—	ADSTCLR10	—	—	—	ADSTCLR06	ADSTCLR05	ADSTCLR04	—	ADSTCLR02	ADSTCLR01	ADSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	R	R	R	W	R	R	R	W	W	W	R	W	W	W

Table 44.236 CR_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When writing, write the value after reset.
27	ADSTCLR27	Writing 1 to this bit clears ADEDF27 in CR_BUSADERSTR.
26	ADSTCLR26*2	Writing 1 to this bit clears ADEDF26 in CR_BUSADERSTR.
25	ADSTCLR25*1	Writing 1 to this bit clears ADEDF25 in CR_BUSADERSTR.
24	ADSTCLR24	Writing 1 to this bit clears ADEDF24 in CR_BUSADERSTR.
23 to 15	Reserved	When writing, write the value after reset.
14	ADSTCLR14	Writing 1 to this bit clears ADEDF14 in CR_BUSADERSTR.
13 to 11	Reserved	When writing, write the value after reset.
10	ADSTCLR10*2	Writing 1 to this bit clears ADEDF10 in CR_BUSADERSTR.
9 to 7	Reserved	When writing, write the value after reset.
6	ADSTCLR06*1	Writing 1 to this bit clears ADEDF06 in CR_BUSADERSTR.
5	ADSTCLR05*1	Writing 1 to this bit clears ADEDF05 in CR_BUSADERSTR.
4	ADSTCLR04*1	Writing 1 to this bit clears ADEDF04 in CR_BUSADERSTR.
3	Reserved	When writing, write the value after reset.
2	ADSTCLR02	Writing 1 to this bit clears ADEDF02 in CR_BUSADERSTR.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in CR_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in CR_BUSADERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.34 CR_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DSSTCLR14	—	—	—	DSSTCLR10	—	—	—	DSSTCLR06	DSSTCLR05	DSSTCLR04	—	DSSTCLR02	DSSTCLR01	DSSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	R	R	R	W	R	R	R	W	W	W	R	W	W	W

Table 44.237 CR_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write the value after reset.
14	DSSTCLR14	Writing 1 to this bit clears DSEDF14 in CR_BUSDSERSTR.
13 to 11	Reserved	When writing, write the value after reset.
10	DSSTCLR10*2	Writing 1 to this bit clears DSEDF10 in CR_BUSDSERSTR.
9 to 7	Reserved	When writing, write the value after reset.
6	DSSTCLR06*1	Writing 1 to this bit clears DSEDF06 in CR_BUSDSERSTR.
5	DSSTCLR05*1	Writing 1 to this bit clears DSEDF05 in CR_BUSDSERSTR.
4	DSSTCLR04*1	Writing 1 to this bit clears DSEDF04 in CR_BUSDSERSTR.
3	Reserved	When writing, write the value after reset.
2	DSSTCLR02	Writing 1 to this bit clears DSEDF02 in CR_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in CR_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in CR_BUSDSERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.35 CR_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in CR_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DDSTCLR14	—	—	—	DDSTCLR10	—	—	—	DDSTCLR06	DDSTCLR05	DDSTCLR04	—	DDSTCLR02	DDSTCLR01	DDSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	R	R	R	W	R	R	R	W	W	W	R	W	W	W

Table 44.238 CR_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When writing, write the value after reset.
14	DDSTCLR14	Writing 1 to this bit clears DDEDF14 in CR_BUSDDERSTR.
13 to 11	Reserved	When writing, write the value after reset.
10	DDSTCLR10*2	Writing 1 to this bit clears DDEDF10 in CR_BUSDDERSTR.
9 to 7	Reserved	When writing, write the value after reset.
6	DDSTCLR06*1	Writing 1 to this bit clears DDEDF06 in CR_BUSDDERSTR.
5	DDSTCLR05*1	Writing 1 to this bit clears DDEDF05 in CR_BUSDDERSTR.
4	DDSTCLR04*1	Writing 1 to this bit clears DDEDF04 in CR_BUSDDERSTR.
3	Reserved	When writing, write the value after reset.
2	DDSTCLR02	Writing 1 to this bit clears DDEDF02 in CR_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDEDF01 in CR_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in CR_BUSDDERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.36 CR_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 50_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ASEDF 27	ASEDF 26	ASEDF 25	ASEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ASEDF 14	—	—	—	ASEDF 10	—	—	—	ASEDF 06	ASEDF 05	ASEDF 04	—	ASEDF 02	ASEDF 01	ASEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.239 CR_BUSASERSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned.
27	ASEDF27	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 3.
26	ASEDF26*2	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 2.
25	ASEDF25*1	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 1.
24	ASEDF24	Indicates that an address ECC 1-bit error was detected in request address to SAXI2MBI bridge for cluster 0.
23 to 15	Reserved	When read, the value after reset is returned.
14	ASEDF14	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 3 from outside.
13 to 11	Reserved	When read, the value after reset is returned.
10	ASEDF10*2	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 2 from outside.
9 to 7	Reserved	When read, the value after reset is returned.
6	ASEDF06*1	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 1 from outside.
5	ASEDF05*1	Indicates that an address ECC 1-bit error was detected in request address from PE3 to Cluster RAM in cluster 1.
4	ASEDF04*1	Indicates that an address ECC 1-bit error was detected in request address from PE2 to Cluster RAM in cluster 1.
3	Reserved	When read, the value after reset is returned.

Table 44.239 CR_BUSASERSTR Register Contents (2/2)

Bit Position	Bit Name	Function
2	ASEDF02	Indicates that an address ECC 1-bit error was detected in request address to Cluster RAM in cluster 0 from outside.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address from PE1 to Cluster RAM in cluster 0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address from PE0 to Cluster RAM in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.37 CR_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 60_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ADEDF 27	ADEDF 26	ADEDF 25	ADEDF 24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADEDF 14	—	—	—	ADEDF 10	—	—	—	ADEDF 06	ADEDF 05	ADEDF 04	—	ADEDF 02	ADEDF 01	ADEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.240 CR_BUSADERSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
27	ADEDF27	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 3.
26	ADEDF26*2	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 2.
25	ADEDF25*1	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 1.
24	ADEDF24	Indicates that an address ECC 2-bit error was detected in request address to SAXI2MBI bridge for cluster 0.
23 to 15	Reserved	When read, the value after reset is returned.
14	ADEDF14	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 3 from outside.
13 to 11	Reserved	When read, the value after reset is returned.
10	ADEDF10*2	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 2 from outside.
9 to 7	Reserved	When read, the value after reset is returned.
6	ADEDF06*1	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 1 from outside.
5	ADEDF05*1	Indicates that an address ECC 2-bit error was detected in request address from PE3 to Cluster RAM in cluster 1.
4	ADEDF04*1	Indicates that an address ECC 2-bit error was detected in request address from PE2 to Cluster RAM in cluster 1.
3	Reserved	When read, the value after reset is returned.

Table 44.240 CR_BUSADERSTR Register Contents (2/2)

Bit Position	Bit Name	Function
2	ADEDF02	Indicates that an address ECC 2-bit error was detected in request address to Cluster RAM in cluster 0 from outside.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address from PE1 to Cluster RAM in cluster 0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address from PE0 to Cluster RAM in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.38 CR_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 70_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DSEDF 14	—	—	—	DSEDF 10	—	—	—	DSEDF 06	DSEDF 05	DSEDF 04	—	DSEDF 02	DSEDF 01	DSEDF 00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.241 CR_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	DSEDF14	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 3 from outside.
13 to 11	Reserved	When read, the value after reset is returned.
10	DSEDF10*2	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 2 from outside.
9 to 7	Reserved	When read, the value after reset is returned.
6	DSEDF06*1	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 1 from outside.
5	DSEDF05*1	Indicates that a data ECC 1-bit error was detected in write data from PE3 to Cluster RAM in cluster 1.
4	DSEDF04*1	Indicates that a data ECC 1-bit error was detected in write data from PE2 to Cluster RAM in cluster 1.
3	Reserved	When read, the value after reset is returned.
2	DSEDF02	Indicates that a data ECC 1-bit error was detected in write data to Cluster RAM in cluster 0 from outside.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in write data from PE1 to Cluster RAM in cluster 0.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data from PE0 to Cluster RAM in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.39 CR_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to Cluster RAMs. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected errors are all set.

This register can be cleared by CR_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CRAM_base> + 80_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DDEDF14	—	—	—	DDEDF10	—	—	—	DDEDF06	DDEDF05	DDEDF04	—	DDEDF02	DDEDF01	DDEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.242 CR_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	DDEDF14	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 3 from outside.
13 to 11	Reserved	When read, the value after reset is returned.
10	DDEDF10*2	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 2 from outside.
9 to 7	Reserved	When read, the value after reset is returned.
6	DDEDF06*1	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 1 from outside.
5	DDEDF05*1	Indicates that a data ECC 2-bit error was detected in write data from PE3 to Cluster RAM in cluster 1.
4	DDEDF04*1	Indicates that a data ECC 2-bit error was detected in write data from PE2 to Cluster RAM in cluster 1.
3	Reserved	When read, the value after reset is returned.
2	DDEDF02	Indicates that a data ECC 2-bit error was detected in write data to Cluster RAM in cluster 0 from outside.
1	DDEDF01	Indicates that a data ECC 2-bit error was detected in write data from PE1 to Cluster RAM in cluster 0.
0	DDEDF00	Indicates that a data ECC 2-bit error was detected in write data from PE0 to Cluster RAM in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.40 CF_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address ECC 2-bit error and address ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CFL_base> + 00_H

Value after reset: $0000\ 000C_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R

Table 44.243 CF_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.3.11.41 CF_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in CF_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CFL_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ASSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ASSTCLR05	ASSTCLR04	—	—	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	W	W

Table 44.244 CF_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When writing, write the value after reset.
24	ASSTCLR24	Writing 1 to this bit clears ASEDF24 in CF_BUSASERSTR.
23 to 6	Reserved	When writing, write the value after reset.
5	ASSTCLR05*1	Writing 1 to this bit clears ASEDF05 in CF_BUSASERSTR.
4	ASSTCLR04*1	Writing 1 to this bit clears ASEDF04 in CF_BUSASERSTR.
3, 2	Reserved	When writing, write the value after reset.
1	ASSTCLR01	Writing 1 to this bit clears ASEDF01 in CF_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASEDF00 in CF_BUSASERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.42 CF_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in CF_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CFL_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ADSTCLR24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ADSTCLR05	ADSTCLR04	—	—	ADSTCLR01	ADSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	R	R	W	W

Table 44.245 CF_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When writing, write the value after reset.
24	ADSTCLR24	Writing 1 to this bit clears ADEDF24 in CF_BUSADERSTR.
23 to 6	Reserved	When writing, write the value after reset.
5	ADSTCLR05*1	Writing 1 to this bit clears ADEDF05 in CF_BUSADERSTR.
4	ADSTCLR04*1	Writing 1 to this bit clears ADEDF04 in CF_BUSADERSTR.
3, 2	Reserved	When writing, write the value after reset.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in CF_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in CF_BUSADERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.43 CF_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to CodeFlash. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the flags of the detected error are all set.

This register can be cleared by CF_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CFL_base> + 50_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ASEDF24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ASEDF05	ASEDF04	—	—	ASEDF01	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.246 CF_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24	ASEDF24	Indicates that an address ECC 1-bit error was detected in request address to SAXI2FAXI bridge.
23 to 6	Reserved	When read, the value after reset is returned.
5	ASEDF05*1	Indicates that an address ECC 1-bit error was detected in request address to CodeFlash bank D in cluster 1.
4	ASEDF04*1	Indicates that an address ECC 1-bit error was detected in request address to CodeFlash bank C in cluster 1.
3, 2	Reserved	When read, the value after reset is returned.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to CodeFlash bank B in cluster 0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to CodeFlash bank A in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.44 CF_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to CodeFlash. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by CF_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_CFL_base> + 60_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ADEDF24	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ADEDF05	ADEDF04	—	—	ADEDF01	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.247 CF_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24	ADEDF24	Indicates that an address ECC 2-bit error was detected in request address to SAXI2FAXI bridge.
23 to 6	Reserved	When read, the value after reset is returned.
5	ADEDF05*1	Indicates that an address ECC 2-bit error was detected in request address to CodeFlash bank D in cluster 1.
4	ADEDF04*1	Indicates that an address ECC 2-bit error was detected in request address to CodeFlash bank C in cluster 1.
3, 2	Reserved	When read, the value after reset is returned.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to CodeFlash bank B in cluster 0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to CodeFlash bank A in cluster 0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.45 PH_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 00_H

Value after reset: $0000\ 000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.248 PH_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error was detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error was detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.46 PH_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASSTCLR25	—	—	—	—	ASSTCLR20	—	ASSTCLR18	ASSTCLR17	ASSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	W	R	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.249 PH_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When writing, write the value after reset.
25	ASSTCLR25	Writing 1 to this bit clears ASEDF25 in PH_BUSASERSTR.
24 to 21	Reserved	When writing, write the value after reset.
20	ASSTCLR20	Writing 1 to this bit clears ASEDF20 in PH_BUSASERSTR.
19	Reserved	When writing, write the value after reset.
18	ASSTCLR18	Writing 1 to this bit clears ASEDF18 in PH_BUSASERSTR.
17	ASSTCLR17	Writing 1 to this bit clears ASEDF17 in PH_BUSASERSTR.
16	ASSTCLR16	Writing 1 to this bit clears ASEDF16 in PH_BUSASERSTR.
15 to 0	Reserved	When writing, write the value after reset.

44.3.11.47 PH_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADSTCLR25	—	—	—	—	ADSTCLR20	—	ADSTCLR18	ADSTCLR17	ADSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	W	R	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.250 PH_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When writing, write the value after reset.
25	ADSTCLR25	Writing 1 to this bit clears ADEDF25 in PH_BUSADERSTR.
24 to 21	Reserved	When writing, write the value after reset.
20	ADSTCLR20	Writing 1 to this bit clears ADEDF20 in PH_BUSADERSTR.
19	Reserved	When writing, write the value after reset.
18	ADSTCLR18	Writing 1 to this bit clears ADEDF18 in PH_BUSADERSTR.
17	ADSTCLR17	Writing 1 to this bit clears ADEDF17 in PH_BUSADERSTR.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in PH_BUSADERSTR.
15 to 0	Reserved	When writing, write the value after reset.

44.3.11.48 PH_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DSSTCLR25	—	—	—	—	—	—	DSSTCLR18	DSSTCLR17	DSSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	R	R	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DSSTCLR11	DSSTCLR10	DSSTCLR09	DSSTCLR08	—	—	—	—	DSSTCLR03	DSSTCLR02	DSSTCLR01	DSSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	W	W	W	W

Table 44.251 PH_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When writing, write the value after reset.
25	DSSTCLR25	Writing 1 to this bit clears DSEDF25 in PH_BUSDSERSTR.
24 to 19	Reserved	When writing, write the value after reset.
18	DSSTCLR18	Writing 1 to this bit clears DSEDF18 in PH_BUSDSERSTR.
17	DSSTCLR17	Writing 1 to this bit clears DSEDF17 in PH_BUSDSERSTR.
16	DSSTCLR16	Writing 1 to this bit clears DSEDF16 in PH_BUSDSERSTR.
15 to 12	Reserved	When writing, write the value after reset.
11	DSSTCLR11*1	Writing 1 to this bit clears DSEDF11 in PH_BUSDSERSTR.
10	DSSTCLR10*1	Writing 1 to this bit clears DSEDF10 in PH_BUSDSERSTR.
9	DSSTCLR09	Writing 1 to this bit clears DSEDF09 in PH_BUSDSERSTR.
8	DSSTCLR08	Writing 1 to this bit clears DSEDF08 in PH_BUSDSERSTR.
7 to 4	Reserved	When writing, write the value after reset.
3	DSSTCLR03*1	Writing 1 to this bit clears DSEDF03 in PH_BUSDSERSTR.
2	DSSTCLR02*1	Writing 1 to this bit clears DSEDF02 in PH_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in PH_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in PH_BUSDSERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.49 PH_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in PH_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DDSTCLR25	—	—	—	—	—	—	DDSTCLR18	DDSTCLR17	DDSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R	R	R	R	R	R	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DDSTCLR11	DDSTCLR10	DDSTCLR09	DDSTCLR08	—	—	—	—	DDSTCLR03	DDSTCLR02	DDSTCLR01	DDSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	W	W	W	W

Table 44.252 PH_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When writing, write the value after reset.
25	DDSTCLR25	Writing 1 to this bit clears DDEDF25 in PH_BUSDDERSTR.
24 to 19	Reserved	When writing, write the value after reset.
18	DDSTCLR18	Writing 1 to this bit clears DDEDF18 in PH_BUSDDERSTR.
17	DDSTCLR17	Writing 1 to this bit clears DDEDF17 in PH_BUSDDERSTR.
16	DDSTCLR16	Writing 1 to this bit clears DDEDF16 in PH_BUSDDERSTR.
15 to 12	Reserved	When writing, write the value after reset.
11	DDSTCLR11*1	Writing 1 to this bit clears DDEDF11 in PH_BUSDDERSTR.
10	DDSTCLR10*1	Writing 1 to this bit clears DDEDF10 in PH_BUSDDERSTR.
9	DDSTCLR09	Writing 1 to this bit clears DDEDF09 in PH_BUSDDERSTR.
8	DDSTCLR08	Writing 1 to this bit clears DDEDF08 in PH_BUSDDERSTR.
7 to 4	Reserved	When writing, write the value after reset.
3	DDSTCLR03*1	Writing 1 to this bit clears DDEDF03 in PH_BUSDDERSTR.
2	DDSTCLR02*1	Writing 1 to this bit clears DDEDF02 in PH_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDEDF01 in PH_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in PH_BUSDDERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.50 PH_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 50_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ASEDF25	—	—	—	—	ASEDF20	—	ASEDF18	ASEDF17	ASEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.253 PH_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	ASEDF25	Indicates that an address ECC 1-bit error was detected in request address to VCI2APB bridge for P-Bus Group 0.
24 to 21	Reserved	When read, the value after reset is returned.
20	ASEDF20	Indicates that an address ECC 1-bit error was detected in request address to AXI2PVCI bridge.
19	Reserved	When read, the value after reset is returned.
18	ASEDF18	Indicates that an address ECC 1-bit error was detected in request address to IPIR.
17	ASEDF17	Indicates that an address ECC 1-bit error was detected in request address to BarrierSync.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to TPTM.
15 to 0	Reserved	When read, the value after reset is returned.

44.3.11.51 PH_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 60_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADEDF25	—	—	—	—	ADEDF20	—	ADEDF18	ADEDF17	ADEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.254 PH_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	ADEDF25	Indicates that an address ECC 2-bit error was detected in request address to VCI2APB bridge for P-Bus Group 0.
24 to 21	Reserved	When read, the value after reset is returned.
20	ADEDF20	Indicates that an address ECC 2-bit error was detected in request address to AXI2PVCI bridge.
19	Reserved	When read, the value after reset is returned.
18	ADEDF18	Indicates that an address ECC 2-bit error was detected in request address to IPIR.
17	ADEDF17	Indicates that an address ECC 2-bit error was detected in request address to BarrierSync.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to TPTM.
15 to 0	Reserved	When read, the value after reset is returned.

44.3.11.52 PH_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 70_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DSEDF25	—	—	—	—	—	—	DSEDF18	DSEDF17	DSEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DSEDF11	DSEDF10	DSEDF09	DSEDF08	—	—	—	—	DSEDF03	DSEDF02	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.255 PH_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	DSEDF25	Indicates that a data ECC 1-bit error was detected in write data to VCI2APB bridge for P-Bus Group 0.
24 to 19	Reserved	When read, the value after reset is returned.
18	DSEDF18	Indicates that a data ECC 1-bit error was detected in write data to IPIR.
17	DSEDF17	Indicates that a data ECC 1-bit error was detected in write data to BarrierSync.
16	DSEDF16	Indicates that a data ECC 1-bit error was detected in write data to TPTM.
15 to 12	Reserved	When read, the value after reset is returned.
11	DSEDF11*1	Indicates that a data ECC 1-bit error was detected in read data to VCI2AXI bridge in cluster 1.
10	DSEDF10*1	Indicates that a data ECC 1-bit error was detected in write data to VCI2AXI bridge in cluster 1.
9	DSEDF09	Indicates that a data ECC 1-bit error was detected in read data to VCI2AXI bridge in cluster 0.
8	DSEDF08	Indicates that a data ECC 1-bit error was detected in write data to VCI2AXI bridge in cluster 0.
7 to 4	Reserved	When read, the value after reset is returned.
3	DSEDF03*1	Indicates that a data ECC 1-bit error was detected in read data to PE3 from peripherals.
2	DSEDF02*1	Indicates that a data ECC 1-bit error was detected in read data to PE2 from peripherals.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in read data to PE1 from peripherals.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in read data to PE0 from peripherals.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.53 PH_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PH_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_PERI_base> + 80_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DDEDF25	—	—	—	—	—	—	DDEDF18	DDEDF17	DDEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DDEDF11	DDEDF10	DDEDF09	DDEDF08	—	—	—	—	DDEDF03	DDEDF02	DDEDF01	DDEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.256 PH_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	DDEDF25	Indicates that a data ECC 2-bit error was detected in write data to VCI2APB bridge for P-Bus Group 0.
24 to 19	Reserved	When read, the value after reset is returned.
18	DDEDF18	Indicates that a data ECC 2-bit error was detected in write data to IPIR.
17	DDEDF17	Indicates that a data ECC 2-bit error was detected in write data to BarrierSync.
16	DDEDF16	Indicates that a data ECC 2-bit error was detected in write data to TPTM.
15 to 12	Reserved	When read, the value after reset is returned.
11	DDEDF11*1	Indicates that a data ECC 2-bit error was detected in read data to VCI2AXI bridge in cluster 1.
10	DDEDF10*1	Indicates that a data ECC 2-bit error was detected in write data to VCI2AXI bridge in cluster 1.
9	DDEDF09	Indicates that a data ECC 2-bit error was detected in read data to VCI2AXI bridge in cluster 0.
8	DDEDF08	Indicates that a data ECC 2-bit error was detected in write data to VCI2AXI bridge in cluster 0.
7 to 4	Reserved	When read, the value after reset is returned.
3	DDEDF03*1	Indicates that a data ECC 2-bit error was detected in read data to PE3 from peripherals.
2	DDEDF02*1	Indicates that a data ECC 2-bit error was detected in read data to PE2 from peripherals.
1	DDEDF01	Indicates that a data ECC 2-bit error was detected in read data to PE1 from peripherals.
0	DDEDF00	Indicates that a data ECC 2-bit error was detected in read data to PE0 from peripherals.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.3.11.54 DM_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when data ECC 2-bit error and data ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_DMDT_base> + 00_H

Value after reset: $0000\ 0003_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.257 DM_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.55 DM_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in DM_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_DMDT_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DSSTCLR08	—	—	—	—	DSSTCLR03	DSSTCLR02	DSSTCLR01	DSSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W

Table 44.258 DM_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When writing, write the value after reset.
8	DSSTCLR08	Writing 1 to this bit clears DSEDF08 in DM_BUSDSERSTR.
7 to 4	Reserved	When writing, write the value after reset.
3	DSSTCLR03	Writing 1 to this bit clears DSEDF03 in DM_BUSDSERSTR.
2	DSSTCLR02	Writing 1 to this bit clears DSEDF02 in DM_BUSDSERSTR.
1	DSSTCLR01	Writing 1 to this bit clears DSEDF01 in DM_BUSDSERSTR.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in DM_BUSDSERSTR.

44.3.11.56 DM_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in DM_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_DMDT_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DDSTCLR08	—	—	—	—	DDSTCLR03	DDSTCLR02	DDSTCLR01	DDSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W	R	R	R	R	W	W	W	W

Table 44.259 DM_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When writing, write the value after reset.
8	DDSTCLR08	Writing 1 to this bit clears DDEDF08 in DM_BUSDDERSTR.
7 to 4	Reserved	When writing, write the value after reset.
3	DDSTCLR03	Writing 1 to this bit clears DDEDF03 in DM_BUSDDERSTR.
2	DDSTCLR02	Writing 1 to this bit clears DDEDF02 in DM_BUSDDERSTR.
1	DDSTCLR01	Writing 1 to this bit clears DDEDF01 in DM_BUSDDERSTR.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in DM_BUSDDERSTR.

44.3.11.57 DM_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to DMA/DTS. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by DM_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_DMDT_base> + 70_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DSEDF08	—	—	—	—	DSEDF03	DSEDF02	DSEDF01	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.260 DM_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8	DSEDF08	Indicates that a data ECC 1-bit error was detected in read data to DTS.
7 to 4	Reserved	When read, the value after reset is returned.
3	DSEDF03	Indicates that a data ECC 1-bit error was detected in internal buffer of sDMAC 1.
2	DSEDF02	Indicates that a data ECC 1-bit error was detected in read data to sDMAC 1.
1	DSEDF01	Indicates that a data ECC 1-bit error was detected in internal buffer of sDMAC 0.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in read data to sDMAC 0.

44.3.11.58 DM_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to DMA/DTS. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by DM_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_DMDT_base> + 80_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	DDEDF08	—	—	—	—	DDEDF03	DDEDF02	DDEDF01	DDEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.261 DM_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8	DDEDF08	Indicates that a data ECC 2-bit error was detected in read data to DTS.
7 to 4	Reserved	When read, the value after reset is returned.
3	DDEDF03	Indicates that a data ECC 2-bit error was detected in internal buffer of sDMAC 1.
2	DDEDF02	Indicates that a data ECC 2-bit error was detected in read data to sDMAC 1.
1	DDEDF01	Indicates that a data ECC 2-bit error was detected in internal buffer of sDMAC 0.
0	DDEDF00	Indicates that a data ECC 2-bit error was detected in read data to sDMAC 0.

44.3.11.59 EM_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by ECCKCPROT register. When ECCKCPROT.KCE = 0, it cannot be written. It can be written only when ECCKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 00_H

Value after reset: $0000\ 000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.262 EM_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error was detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error was detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error was detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error was detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.60 EM_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR17	ASSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ASSTCLR07	ASSTCLR06	ASSTCLR05	ASSTCLR04	ASSTCLR03	ASSTCLR02	ASSTCLR01	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 44.263 EM_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	ASSTCLR17*2	Writing 1 to this bit clears ASEDF17 in EM_BUSASERSTR.
16	ASSTCLR16	Writing 1 to this bit clears ASEDF16 in EM_BUSASERSTR.
15 to 8	Reserved	When writing, write the value after reset.
7	ASSTCLR07*1	Writing 1 to this bit clears ASEDF07 in EM_BUSASERSTR.
6	ASSTCLR06*1	Writing 1 to this bit clears ASEDF06 in EM_BUSASERSTR.
5	ASSTCLR05*1	Writing 1 to this bit clears ASEDF05 in EM_BUSASERSTR.
4	ASSTCLR04*1	Writing 1 to this bit clears ASEDF04 in EM_BUSASERSTR.
3	ASSTCLR03	Writing 1 to this bit clears ASEDF03 in EM_BUSASERSTR.
2	ASSTCLR02	Writing 1 to this bit clears ASEDF02 in EM_BUSASERSTR.
1	ASSTCLR01	Writing 1 to this bit clears ASEDF01 in EM_BUSASERSTR.
0	ASSTCLR00	Writing 1 to this bit clears ASEDF00 in EM_BUSASERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.61 EM_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADSTCLR17	ADSTCLR16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ADSTCLR07	ADSTCLR06	ADSTCLR05	ADSTCLR04	ADSTCLR03	ADSTCLR02	ADSTCLR01	ADSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 44.264 EM_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	ADSTCLR17*2	Writing 1 to this bit clears ADEDF17 in EM_BUSADERSTR.
16	ADSTCLR16	Writing 1 to this bit clears ADEDF16 in EM_BUSADERSTR.
15 to 8	Reserved	When writing, write the value after reset.
7	ADSTCLR07*1	Writing 1 to this bit clears ADEDF07 in EM_BUSADERSTR.
6	ADSTCLR06*1	Writing 1 to this bit clears ADEDF06 in EM_BUSADERSTR.
5	ADSTCLR05*1	Writing 1 to this bit clears ADEDF05 in EM_BUSADERSTR.
4	ADSTCLR04*1	Writing 1 to this bit clears ADEDF04 in EM_BUSADERSTR.
3	ADSTCLR03	Writing 1 to this bit clears ADEDF03 in EM_BUSADERSTR.
2	ADSTCLR02	Writing 1 to this bit clears ADEDF02 in EM_BUSADERSTR.
1	ADSTCLR01	Writing 1 to this bit clears ADEDF01 in EM_BUSADERSTR.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in EM_BUSADERSTR.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.62 EM_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 30_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSSTCLR17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.265 EM_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	DSSTCLR17*1	Writing 1 to this bit clears DSEDF17 in EM_BUSDSERSTR.
16 to 0	Reserved	When writing, write the value after reset.

Note 1. This function is not implemented in RH850/U2A6.

44.3.11.63 EM_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in EM_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 40_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDSTCLR17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.266 EM_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When writing, write the value after reset.
17	DDSTCLR17*1	Writing 1 to this bit clears DDEDF17 in EM_BUSDDERSTR.
16 to 0	Reserved	When writing, write the value after reset.

Note 1. This function is not implemented in RH850/U2A6.

44.3.11.64 EM_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 50_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF17	ASEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ASEDF07	ASEDF06	ASEDF05	ASEDF04	ASEDF03	ASEDF02	ASEDF01	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.267 EM_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	ASEDF17*2	Indicates that an address ECC 1-bit error was detected in request address to FAXI2SAXI bridge.
16	ASEDF16	Indicates that an address ECC 1-bit error was detected in request address to the remap module from a SAXI master.
15 to 8	Reserved	When read, the value after reset is returned.
7	ASEDF07*1	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE3.
6	ASEDF06*1	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE3.
5	ASEDF05*1	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE2.
4	ASEDF04*1	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE2.
3	ASEDF03	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE1.
2	ASEDF02	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE1.
1	ASEDF01	Indicates that an address ECC 1-bit error was detected in request address to the remap module for data read from PE0.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to the remap module for instruction fetch from PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.65 EM_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 60_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF17	ADEDF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ADEDF07	ADEDF06	ADEDF05	ADEDF04	ADEDF03	ADEDF02	ADEDF01	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.268 EM_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	ADEDF17*2	Indicates that an address ECC 2-bit error was detected in request address to FAXI2SAXI bridge.
16	ADEDF16	Indicates that an address ECC 2-bit error was detected in request address to the remap module from a SAXI master.
15 to 8	Reserved	When read, the value after reset is returned.
7	ADEDF07*1	Indicates that an address ECC 2-bit error was detected in request address to the remap module for data read from PE3.
6	ADEDF06*1	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE3.
5	ADEDF05*1	Indicates that an address ECC 2-bit error was detected in request address to remap module for data read from PE2.
4	ADEDF04*1	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE2.
3	ADEDF03	Indicates that an address ECC 2-bit error was detected in request address to the remap module for data read from PE1.
2	ADEDF02	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE1.
1	ADEDF01	Indicates that an address ECC 2-bit error was detected in request address to remap module for data read from PE0.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to remap module for instruction fetch from PE0.

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.3.11.66 EM_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 70_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.269 EM_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	DSEDF17*1	Indicates that a data ECC 1-bit error was detected in read data to FAXI2SAXI bridge.
16 to 0	Reserved	When read, the value after reset is returned.

Note 1. This function is not implemented in RH850/U2A6.

44.3.11.67 EM_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to emulation. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by EM_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_EMU_base> + 80_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDEDF17	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.270 EM_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	DDEDF17*1	Indicates that a data ECC 2-bit error was detected in read data to FAXI2SAXI bridge.
16 to 0	Reserved	When read, the value after reset is returned.

Note 1. This function is not implemented in RH850/U2A6.

44.3.11.68 APECKCPROT — Key Code Protection Register

This register is used for protection against writing to ECC control registers and Error notification control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <PBnECC_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.271 APECKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit*2 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

Note 2. Unlocking this protection is one of necessary conditions to enable writing the protected registers. It is also necessary to unlock security protection driven by the ICUMHA when ICUMHA function is active. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware.*
ECCCNT_A_V2An (n = 1, 2, 6, 9)
ECCCNT_D_V2AnW (n = 1, 2, 6, 9)
BECCCAP_V2An (n = 1, 2, 6, 9)

44.3.11.69 PHC_BUSERRINT — Error Notification Control Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error and address/data ECC 1-bit error are detected.

This register is protected by PBnECC.APECKCPROT register. When PBnECC.APECKCPROT.KCE = 0, it cannot be written. It can be written only when PBnECC.APECKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 00_H (n = 1-9)

Value after reset: 0000 $000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	DDEDIE	DSEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.272 PHC_BUSERRINT Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ADEDIE	Controls error reports when address ECC 2-bit error is detected. 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled
2	ASEDIE	Controls error reports when address ECC 1-bit error is detected. 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled
1	DDEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	DSEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.70 PHC_BUSASSTCLR — Address 1-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSASERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 10_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASSTCLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.273 PHC_BUSASSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ASSTCLR00	Writing 1 to this bit clears ASEDF00 in PHC_BUSASERSTR.

44.3.11.71 PHC_BUSADSTCLR — Address 2-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSADERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 20_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.274 PHC_BUSADSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ADSTCLR00	Writing 1 to this bit clears ADEDF00 in PHC_BUSADERSTR.

44.3.11.72 PHC_BUSDSSTCLR — Data 1-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSDSERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 30_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.275 PHC_BUSDSSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DSSTCLR00	Writing 1 to this bit clears DSEDF00 in PHC_BUSDSERSTR.

44.3.11.73 PHC_BUSDDSTCLR — Data 2-bit Error Status Clear Register

This register is used to clear error flags in PHC_BUSDDERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 40_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDST CLR00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.276 PHC_BUSDDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	DDSTCLR00	Writing 1 to this bit clears DDEDF00 in PHC_BUSDDERSTR.

44.3.11.74 PHC_BUSASERSTR — Address 1-bit Error Status Register

This register indicates that an address ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSASSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 50_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.277 PHC_BUSASERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ASEDF00	Indicates that an address ECC 1-bit error was detected in request address to VCI2APB bridge for P-Bus Group n.

44.3.11.75 PHC_BUSADERSTR — Address 2-bit Error Status Register

This register indicates that an address ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSADSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 60_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.278 PHC_BUSADERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ADEDF00	Indicates that an address ECC 2-bit error was detected in request address to VCI2APB bridge for P-Bus Group n.

44.3.11.76 PHC_BUSDSERSTR — Data 1-bit Error Status Register

This register indicates that a data ECC 1-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSDSSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 70_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.279 PHC_BUSDSERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DSEDF00	Indicates that a data ECC 1-bit error was detected in write data to VCI2APB bridge for P-Bus Group n.

44.3.11.77 PHC_BUSDDERSTR — Data 2-bit Error Status Register

This register indicates that a data ECC 2-bit error has occurred in a path related to peripherals. When a new error occurs, status flag corresponding to the error is set if the flag has been cleared. If multiple errors are detected at the same time, the errors that are detected are all set.

This register can be cleared by PHC_BUSDDSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <BECCCAP_V2An_base> + 80_H (n = 1-9)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDEDF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.280 PHC_BUSDDERSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DDEDF00	Indicates that a data ECC 2-bit error was detected in write data to VCI2APB bridge for P-Bus Group n.

44.3.11.78 HBECECCCTL — ECC Control Register

This register controls the ECC error detection/correction and 1-bit error correction.

This register is protected by HBnECC.HBECKCPROT register. When HBnECC.HBECKCPROT.KCE = 0, it cannot be written. It can be written only when HBnECC.HBECKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit or 16-bit units.

Address: <HBnECC_base> + 00_H (n=91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0008_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AECC DIS	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 44.281 HBECECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	AECCDIS	Address ECC disable bit (for H-Bus slave) Sets address ECC error detection to enable/disable. 0: Address ECC error detection is enabled 1: Address ECC error detection is disabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
1	SECDIS	ECC 1-bit error correction enable bit When using ECC error detection/correction (ECCDIS = 0), this bit sets 1-bit error correction to enable/disable. 0: Correct when 1-bit error is detected 1: Do not correct when 1-bit error is detected
0	ECCDIS	ECC disable bit Sets ECC error detection/correction to enable/disable. 0: ECC error detection/correction is enabled 1: ECC error detection/correction is disabled

44.3.11.79 HBECERRINT — ECC Error Information Register

This register controls whether error information is reported to ECM, when address/data ECC 2-bit error, address/data ECC 1-bit error are detected.

This register is protected by HBnECC.HBECKCPROT register. When HBnECC.HBECKCPROT.KCE = 0, it cannot be written. It can be written only when HBnECC.HBECKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 04_H(n=91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: 0000 0033_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ADEDIE	ASEDIE	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 44.282 HBECERRINT Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	ADEDIE	Controls error reports when address ECC 2-bit error is detected (for H-Bus slave). 0: Address ECC 2-bit error report disabled 1: Address ECC 2-bit error report enabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
4	ASEDIE	Controls error reports when address ECC 1-bit error is detected (for H-Bus slave). 0: Address ECC 1-bit error report disabled 1: Address ECC 1-bit error report enabled For H-Bus master, when writing, write the value after reset because this bit is reserved.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	Controls error reports when data ECC 2-bit error is detected. 0: Data ECC 2-bit error report disabled 1: Data ECC 2-bit error report enabled
0	SEDIE	Controls error reports when data ECC 1-bit error is detected. 0: Data ECC 1-bit error report disabled 1: Data ECC 1-bit error report enabled

44.3.11.80 HBECSTCLR — Error Status Clear Register

This register is used to clear error flag in HBEC1STERSTR. This is write only register and read value is always “0”.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 08_H (n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ST CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 44.283 HBECSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Writing 1 to this bit clears AECDEDF0/AECSEDF0/DEDF0/SEDF0 in HBEC1STERSTR.

44.3.11.81 HBEC1STERSTR — Error Status Register

This register indicates whether error has occurred. When all error flags are “0” and a new error occurs, error status flag is set.

If address ECC 1-bit error or data ECC 1-bit error are set and the new error is address ECC 2-bit error or data ECC 2-bit error, the new error is set (does not clear the previous error flag).

If multiple causes of errors are detected, the errors that are detected are all set (e.g. if data ECC 2-bit error and address ECC 2-bit error are detected at the same access, DEDF0 and AECDEDF0 are both set.).

This register can be cleared by STCLR0 in HBECSTCLR register.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 10_H (n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	AEC DEDF0	AEC SEDF0	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.284 HBEC1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	AECDEDF0	Indicates address ECC 2-bit error was detected (for H-Bus slave). This is set when address ECC 2-bit error is detected when AECDEDF0 and DEDF0 are all “0”. For H-Bus master, when read, the value after reset is returned because this bit is reserved.
6	AECSEDF0	Indicates address ECC 1-bit error was detected (for H-Bus slave). This is set when address ECC 1-bit error is detected when all error flags are “0”. For H-Bus master, when read, the value after reset is returned because this bit is reserved.
5 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	Indicates data ECC 2-bit error was detected. This is set when data ECC 2-bit error is detected when AECDEDF0 and DEDF0 are all “0”.
0	SEDF0	Indicates data ECC 1-bit error was detected. This is set when data ECC 1-bit error is detected when all error flags are “0”.

44.3.11.82 HBECTSTCTL — ECC Test Control Register

ECC test (self-diagnostics) register. After ECC test mode (ECCTST = 1) setting, arbitrary data value can be input to ECC decoder of H-Bus slave. For H-Bus master, keep the value after reset because this register is reserved.

This register is protected by the HBnECC.HBECKCPROT register. When HBnECC.HBECKCPROT.KCE = 0, it cannot be written. It can be written only when HBnECC.HBECKCPROT.KCE = 1.

When accessing any registers in the same H-Bus Group, ECCTST is cleared to 0.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 150_H(n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.285 HBECTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	ECCTST	Selects input data for the address ECC decoder. 0:Normal mode 1:Test mode The values of HBECTSTDIN0 and HBECTSTEIN registers are provided to data and ECC field of the ECC decoder respectively when accessing any registers in the same H-Bus Group in test mode. This bit is always read as 0.
0	Reserved	When writing, write the value after reset.

44.3.11.83 HBECTSTDIN0 — Test Data Input Register

This register is used to inject errors into data field of ECC decoder for self-diagnosis. The values of HBECTSTDIN0 and HBECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by the key-code protection HBnECC.HBECKCPROT register. When HBnECC.HBECKCPROT.KCE = 0, it cannot be written. It can be written only when HBnECC.HBECKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 154_H(n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.286 HBECTSTDIN0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA[31:0]	Specify the test data for the address ECC decoder. The data is provided to data field of the ECC decoder of H-Bus slave when accessing any registers in the same H-Bus Group when ECCTST = 1. For H-Bus master, keep the value after reset because this register is reserved.

NOTE

An actual access address for H-bus modules is determined according to the value of HBECTSTDIN0.

44.3.11.84 HBECTSTEIN — Test ECC Input Register

This register is used to inject errors into ECC field of ECC decoder for self-diagnosis. The values of HBECTSTDIN0 and HBECTSTEIN registers are input to data and ECC field of the target ECC decoder in test mode.

This register is protected by HBnECC.HBECKCPROT register. When HBnECC.HBECKCPROT.KCE = 0, it cannot be written. It can be written only when HBnECC.HBECKCPROT.KCE = 1.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <HBnECC_base> + 158_H (n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.287 HBECTSTEIN Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECC[6:0]	Specify the test data for the address ECC decoder. The data is provided to ECC field of the ECC decoder of H-Bus slave when accessing any registers in the same H-Bus Group when ECCTST = 1. For H-Bus master, keep the value after reset because this register is reserved.

44.3.11.85 HBECKCPROT — Key Code Protection Register

This register is used for protection against writing to the HBnECC control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <HBnECC_base> + $15C_H$(n = 91M, 91S, 92M, 92S, 93M, 93S, 94M, 95M, 96S)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.288 HBECKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.3.11.86 Test Function

ECC on the write data, the read data and the address are checked by the each ECC decoder. Each ECC decoder has a self-diagnosis function that uses ERRGEN module.

For details, refer to **Section 44.3.12.8, Usage of ERRGEN**.

44.3.12 Error Injection for ECC Function

44.3.12.1 Overview

ERRGEN is a utility module for self-diagnosis tests of ECC decoders. ERRGEN can be used to generate a data pairs and ECC including errors in any bit. ERRGEN has the following features.

Table 44.289 Error Injection for ECC Function

Item	Description
ECC calculation	Provides the correct 7-bit and 8-bit ECC value of arbitrary data.
Error injection	Injects various errors into read data include ECC field. • Error injection is only into the 32-bit data and 7-bit ECC fields. • No function to inject errors into 64-bit data and 8-bit ECC fields directly but ECC calculation register is available.
Register set	A set of registers in odd and even words respectively used to inject errors to 64-bit bus consisting of 2 sets of 32-bit data and 7-bit ECC are implemented, and also a register for users to calculate 8-bit ECC field.

44.3.12.2 List of Registers

Table 44.290 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ERRGEN (P-Bus Group 0)	Error data register	EGDATORGn (n = 0, 1)	FFC5 8400_H + n × 04_H	8, 16, 32	PBG00#2	—
	7-bit ECC calculation register	EGECCCALn (n = 0, 1)	FFC5 8408_H + n × 04_H	8, 16, 32	PBG00#2	—
	ECC Data Error injection register	EGDATINVn (n = 0, 1)	FFC5 8410_H + n × 04_H	8, 16, 32	PBG00#2	—
	ECC Field Error injection register	EGECCINVn (n = 0, 1)	FFC5 8418_H + n × 04_H	8, 16, 32	PBG00#2	—
	8-bit ECC calculation register	EGECC8CAL	FFC5 8420_H	8, 16, 32	PBG00#2	—

44.3.12.3 EGDATORGn — Error Data Register

This register is used to set original data before the error injection and to obtain the modified data containing injected errors according to the settings of EGDATINVn and EGECCINVn. The read data is the value of this register XOR-ed with EGDATINVn and the read value of ECC is EGECCCALn XOR-ed with EGECCINVn when reading this register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: FFC5 8400_H + n × 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATORG[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATORG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.291 EGDATORGn Register Contents

Bit Position	Bit Name	Function
31 to 0	DATORG[31:0]	Write data before injecting errors and read data after errors are injected

44.3.12.4 EGECCCALn — 7-bit ECC Calculation Register

This register is used to obtain a 7-bit ECC value for the arbitrary value written to EGDATORGn. This ECC value is then injected into an ECC decoder on the address channel. ECC value calculated from EGDATORGn can be read when reading this register. The value of this register is updated simultaneously when EGDATORGn is updated.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: FFC5 8408_H + n × 04_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECCCAL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.292 EGECCCALn Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	ECCCAL[6:0]	Read ECC values calculated from EGDATORGn

44.3.12.5 EGDATINVn — ECC Data Error Injection Register

This register is used to inject errors into the read data signal when reading EGDATORGn. The read data signal is driven by EGDATORGn XOR-ed with this register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: FFC5 8410_H + n × 04_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATINV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATINV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.293 EGDATINVn Register Contents

Bit Position	Bit Name	Function
31 to 0	DATINV[31:0]	Bit pattern to invert EGDATORGn value when reading EGDATORGn

44.3.12.6 EGECCINVn — ECC Field Error Injection Register

This register is used to inject errors into the ECC signal when reading EGDATORGn. The ECC signal is driven by EGECCCALn XOR-ed with this register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: FFC5 8418_H + n × 04_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	ECCINV[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.294 EGECCINVn Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECCINV[6:0]	Bit pattern to invert ECC value of EGDATORGn when reading EGDATORGn

44.3.12.7 EGECC8CAL — 8-bit ECC Calculation Register

This register is used to obtain an ECC values for the arbitrary values written to EGDATORG0 and EGDATORG1. These ECC values are then injected into ECC decoders on Data RAM of sDMAC. ECC values calculated from {EGDATORG1, EGDATORG0} can be read when reading this register. The value of this register is updated simultaneously when either EGDATORG0 or EGDATORG1 is updated.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: FFC5 8420_H

Value after reset: 0000 0003_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECC8CAL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.295 EGECC8CAL Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7 to 0	ECC8CAL[7:0]	Read ECC values calculated from {EGDATORG1,EGDATORG0}

44.3.12.8 Usage of ERRGEN

(1) Error Injection into Read Data Channel

ERRGEN can be used to run self-diagnosis tests for ECC decoders of some bus masters on read data channel. Self-diagnosis tests for an ECC decoder on read data channel are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Set any 1 or 2 bits among EGDATINV0, EGDATINV1, EGECCINV0 and EGECCINV1 to 1.
3. Read EGDATORG0 and EGDATORG1 at the same time in double word access from a target bus master.
4. Check ECC error notification asserted from the bus master.

(2) Error Injection into Write Data Channel

ERRGEN can be used to run self-diagnosis tests for ECC decoders of some bus slaves on write data channel. Self-diagnosis tests for an ECC decoder on write data channel are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Set any 1 or 2 bits among EGDATINV0, EGDATINV1, EGECCINV0 and EGECCINV1 to 1.
3. Make DTS transfer data from EGDATORG0 and EGDATORG1 to a target bus slave in 64-bit mode.
4. Check ECC error notification asserted from the bus slave.

(3) Error Injection into Address Channel

This case uses a different feature of ERRGEN, which can calculate ECC value of data. It is necessary for users to know correct ECC value for any data to inject various errors into ECC decoders of bus slaves on address channel. Self-diagnosis tests for an ECC decoder on address channel are made in the following way.

1. Write arbitrary data to EGDATORGn.
2. Read EGECCCALn to obtain correct ECC value of the data.
3. Invert any 1 or 2 bits among the data and the ECC value.
4. Set the modified data and ECC value to related registers*1 of the Address EDC decoder and enable its test function.
5. Read data from a target bus slave including the ECC decoder.
6. Check ECC error notification asserted from the ECC decoder.

(4) Error Injection into Data RAM of sDMAC

ERRGEN can also calculate ECC value of 64-bit data. It is necessary for users to learn correct ECC value for any data to inject various errors into ECC decoders on Data RAM of sDMAC. Self-diagnosis tests for an ECC decoder on Data RAM of sDMAC are made in the following way.

1. Write arbitrary data to EGDATORG0 and EGDATORG1.
2. Read EGECC8CAL to obtain correct ECC value of the data.
3. Invert any 1 or 2 bits among the data and the ECC value.
4. Set the modified data and ECC value to related registers*2 of the sDMAC Data RAM ECC module and enable its test function.

5. Make sDMAC including the ECC decoder transfer data from anywhere to anywhere.
6. Check ECC error notification asserted from the ECC decoder.

Note 1. For details, refer to **Section 44.3.11, Safety Mechanism on Data Transfer Path**.
Note 2. For details, refer to **Section 44.3.9, sDMAC/DTSRAM ECC and Address Feedback**.

44.3.13 ECC error address calculation

The ECC error address register does not show the actual address where the error has occured.

The actual address where the error was detected needs to be calculated by the related formula described in **Table 44.296**.

Table 44.296 ECC error address calculation table (1/2)

ROM/RAM (or error capture reigster name)	ECC error address register	Formula for ECC error address calculation
Code Flash	CF_nSEADR (n = 00 to 03)	0000 0000_H + (SEADR0[25:2] << 4)
	CF_00DEADR	0000 0000_H + (DEADR0[25:2] << 4)
Data Flash	DFEADR	FF20 0000_H + (DFEADR[20:2] << 2)
Local RAM (PE0)	LR0_nSEADR (n = 00 to 07)	FDC0 0000_H + (SEADR0[15:2] << 2)
		FDE0 0000_H + (SEADR0[15:2] << 2)
	LR0_00DEADR	FDC0 0000_H + (DEADR0[15:2] << 2)
		FDE0 0000_H + (DEADR0[15:2] << 2)
Local RAM (PE1)	LR0_nSEADR (n = 00 to 07)	FDA0 0000_H + (SEADR0[15:2] << 2)
		FDE0 0000_H + (SEADR0[15:2] << 2)
	LR0_00DEADR	FDA0 0000_H + (DEADR0[15:2] << 2)
		FDE0 0000_H + (DEADR0[15:2] << 2)
Local RAM (PE2)*1	LR0_nSEADR (n = 00 to 07)	FD80 0000_H + (SEADR0[15:2] << 2)
		FDE0 0000_H + (SEADR0[15:2] << 2)
	LR0_00DEADR	FD80 0000_H + (DEADR0[15:2] << 2)
		FDE0 0000_H + (DEADR0[15:2] << 2)
Local RAM (PE3)*1	LR0_nSEADR (n = 00 to 07)	FD60 0000_H + (SEADR0[15:2] << 2)
		FDE0 0000_H + (SEADR0[15:2] << 2)
	LR0_00DEADR	FD60 0000_H + (DEADR0[15:2] << 2)
		FDE0 0000_H + (DEADR0[15:2] << 2)
Local RAM (requested by other PEs in speculative access)	LR1_nSEADR (n = 00 to 07)	FD00 0000_H + (SEADR0[23:2] << 2)
	LR1_00DEADR	FD00 0000_H + (DEADR0[23:2] << 2)
Cluster RAM	CR_nSEADR (n = 00 to 07)	FE00 0000_H + (SEADR0[23:2] << 2)
	CR_00DEADR	FE00 0000_H + (DEADR0[23:2] << 2)
DTS RAM*5	DR_00SEADR	FFF8 9000_H + (SEADR0[11:2] << 2) + (SEADR0[11:5] << 5)
	DR_00DEADR	FFF8 9000_H + (DEADR0[11:2] << 2) + (DEADR0[11:5] << 5)
sDMAC0 Descriptor RAM	DE_00SEADR	FFF9 4000_H + (SEADR0[11:2] << 2)
	DE_00DEADR	FFF9 4000_H + (DEADR0[11:2] << 2)
sDMAC1 Descriptor RAM	DE_00SEADR	FFF9 $C000_H$ + (SEADR0[11:2] << 2)
	DE_00DEADR	FFF9 $C000_H$ + (DEADR0[11:2] << 2)
RS-CANFD0 Message buffer RAM (MB RAM)*2	E7RC01.E710EAD	RSCFD0CFDGTSTCFG.RTMPS[9:0] = E7RC01.E710EDA[15:6] + 060_H
		FFF5 8400_H + (E710EAD[5:0] << 2)
RS-CANFD0 Acceptance filter list RAM (AFL0 RAM)*2	E7RC02.E710EAD	RSCFD0CFDGTSTCFG.RTMPS[9:0] = 000_B + E7RC02.E710EAD[11:5]
		FFF5 8400_H + (E710EAD[4:0] << 3) + 000_B

Table 44.296 ECC error address calculation table (2/2)

ROM/RAM (or error capture reigster name)	ECC error address register	Formula for ECC error address calculation
RS-CANFD0 Acceptance filter list RAM (AFL1 RAM)*2	E7RC03.E710EAD	RSCFD0CFDGTSTCFG.RTMPS[9:0] = 000_B + E7RC03.E710EAD[11:5]
		FFF5 8400_H + (E710EAD[4:0] << 3) + 100_B
RS-CANFD1 Message buffer RAM (MB RAM)*2	E7RC11.E710EAD	RSCFD1CFDGTSTCFG.RTMPS[9:0] = E7RC11.E710EDA[15:6] + 060_H
		FFD0 8400_H + (E710EAD[5:0] << 2)
RS-CANFD1 Acceptance filter list RAM (AFL0 RAM)*2	E7RC12.E710EAD	RSCFD1CFDGTSTCFG.RTMPS[9:0] = 000_B + E7RC12.E710EAD[11:5]
		FFD0 8400_H + (E710EAD[4:0] << 3) + 000_B
RS-CANFD1 Acceptance filter list RAM (AFL1 RAM)*2	E7RC13.E710EAD	RSCFD1CFDGTSTCFG.RTMPS[9:0] = 000_B + E7RC13.E710EAD[11:5]
		FFD0 8400_H + (E710EAD[4:0] << 3) + 100_B
GTM MCS0_RAM0*3	E7GT00.E710EAD	FF63 8000_H + (ECEAD[10:0] << 2)
GTM MCS0_RAM1*3	E7GT01.E710EAD	FF63 $A000_H$ + (ECEAD[9:0] << 2)
GTM MCS1_RAM0*3	E7GT10.E710EAD	FF64 0000_H + (ECEAD[10:0] << 2)
GTM MCS1_RAM1*3	E7GT11.E710EAD	FF64 2000_H + (ECEAD[9:0] << 2)
GTM MCS2_RAM0*3	E7GT20.E710EAD	FF64 8000_H + (ECEAD[10:0] << 2)
GTM MCS2_RAM1*3	E7GT21.E710EAD	FF64 $A000_H$ + (ECEAD[9:0] << 2)
GTM MCS3_RAM0*3	E7GT30.E710EAD	FF65 0000_H + (ECEAD[10:0] << 2)
GTM MCS3_RAM1*3	E7GT31.E710EAD	FF65 2000_H + (ECEAD[9:0] << 2)
MSPI0 RAM [For U2A-EVA/U2A16/U2A8] *4	E7MS00.E710EAD	FFC7 7000_H + ECEAD[6:0]
MSPI1 RAM [For U2A-EVA/U2A16/U2A8]*4	E7MS01.E710EAD	FFC7 $F000_H$ + ECEAD[6:0]
MSPI2 RAM [For U2A-EVA/U2A16/U2A8]*4	E7MS02.E710EAD	FFD8 1000_H + ECEAD[6:0]
MSPI3 RAM [For U2A-EVA/U2A16/U2A8]*4	E7MS03.E710EAD	FFCC 1000_H + ECEAD[6:0]
MSPI4 RAM [For U2A-EVA/U2A16/U2A8]*4	E7MS04.E710EAD	FFD8 3000_H + ECEAD[6:0]
MSPI5 RAM [For U2A-EVA/U2A16/U2A8]*4	E7MS05.E710EAD	FFCC 5000_H + ECEAD[6:0]
MSPI6 RAM [For U2A-EVA/U2A16(516/373 pins)/ U2A8(373 pins)]*4	E7MS06.E710EAD	FFD8 5000_H + ECEAD[6:0]
MSPI7 RAM [For U2A-EVA/U2A16(516/373 pins)/ U2A8(373 pins)]*4	E7MS07.E710EAD	FFCC 9000_H + ECEAD[6:0]
MSPI8 RAM [For U2A-EVA/U2A16(516/373 pins)/ U2A8(373 pins)]*4	E7MS08.E710EAD	FFD8 7000_H + ECEAD[6:0]
MSPI9 RAM [For U2A-EVA/U2A16(516 pins)]*4	E7MS09.E710EAD	FFCC $D000_H$ + ECEAD[6:0]
MSPI0 RAM [For U2A6]*4	E7MS00.E710EAD	FFC7 7000_H + (ECEAD[8:2] << 2)
MSPI1 RAM [For U2A6]*4	E7MS01.E710EAD	FFC7 $F000_H$ + (ECEAD[8:2] << 2)
MSPI2 RAM [For U2A6]*4	E7MS02.E710EAD	FFD8 1000_H + (ECEAD[8:2] << 2)
MSPI3 RAM [For U2A6]*4	E7MS03.E710EAD	FFCC 1000_H + (ECEAD[8:2] << 2)
MSPI4 RAM [For U2A6(292/176/156 pins)]*4	E7MS04.E710EAD	FFD8 3000_H + (ECEAD[8:2] << 2)
MSPI5 RAM [For U2A6(292/176/156 pins)]*4	E7MS05.E710EAD	FFCC 5000_H + (ECEAD[8:2] << 2)

Note 1. This function is not implemented in RH850/U2A8 and RH850/U2A6.

Note 2. To access to RS-CANFD RAM, RAM test mode must be enabled. For more detail of RAM test mode, refer to **Section 23, CANFD Interface (RS-CANFD)**.
For the RS-CANFD Message Buffer RAM, ECEAD indicates the address right-shifted by 2 bits rather than the actual address, so it is necessary for users to multiply it by 4 and add the base address of RS-CANFD MB RAM in test mode to get address at which the error was detected. For the page number setting of the

RS-CANFD Message Buffer RAM, it can be calculated by E710EAD[15:6] + 60_H.
For the RS-CANFD Acceptance Filter List RAM, ECEAD indicates the address right-shifted by 3 bits rather than the actual address, so it is necessary for users to multiply it by 8 and add the base address of the RS-CANFD Acceptance Filter List RAM in test mode to get address at which the error was detected. For the page number setting of the RS-CANFD Acceptance Filter List RAM, it can be calculated by 0000_H + E710EAD[11:5].
For details of the RAM test mode, refer to **Section 23, CANFD Interface (RS-CANFD)**.

Note 3. GTM ECC error address indication
ECEAD indicates the address right-shifted by 2 bits rather than the actual address, so it is necessary for users to multiply it by 4 and add the base address of corresponding GTM RAM to get address at which the error was detected. It means the error address in the 32-bit address space is
GTM MCS RAM0 (E7GT00, E7GT10, E7GT20, E7GT30): <the address of corresponding GTM RAM0> + (ECEAD[10:0] << 2),
GTM MCS RAM1 (E7GT01, E7GT11, E7GT21, E7GT31): <the base address of corresponding GTM RAM1> + (ECEAD[9:0] << 2).
For the base address and configuration of GTM RAM, refer to **Section 38, Generic Timer Module (GTM)**.

Note 4. MSPI ECC error address indication
In RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8, ECEAD indicates the offset address that has to be added to the base address of the RAM to get the address at which the error was detected.
In RH850/U2A6, ECEAD indicates the address right-shifted by 2 bits rather than the actual address, so it is necessary for users to multiply it by 4 and add the base address of corresponding MSPIn RAM to get address at which the error was detected.
For the base address of MSPIn RAM, refer to **Section 19, Multichannel Serial Peripheral Interface (MSPI)**.

Note 5. The address for DTSnnnCM can be recognized from corresponding DTTCCnnn after calculation. For more detail, refer to **Section 8, DTS Controller**.

44.4 Hardware Redundancy

44.4.1 Overview

This product implements the CPU0/1/2/3 and DTS with the lockstep function to quickly detect failures without software interaction. The CPU0/1/2/3 and DTS execute the function using two different cores; that is, a master core and checker core, and constantly compare the execution results of the two cores. When the results do not match, an error is reported to the ECM. The comparison targets of CPU's bus outputs are own Local RAM, Cluster RAM, Inter-processor element Bus, Local FLASH Bus, and the tag RAM and the data RAM of Instruction cache. The lockstep function of the CPU1/2/3 can be disabled by the configuration of flash option byte 9. For details of the configuration, see **Section 51, Flash Memory**.

The lockstep function of the CPU0/1/2/3 and DTS features of failure insertion, through which errors can be intentionally triggered and thus self-diagnosis of the lockstep operation is possible. For details of CPU0/1/2/3's failure insertion, refer to **Section 3, CPU System**.

Following table shows redundant structure implemented in this product.

Table 44.297 Redundant Blocks

Block Name	Features	Comparator Error Injection
CPU0, CPU1, CPU2*1, CPU3*1	Dual lock step with 2-cycle delay	Yes
DTS	Dual lock step with 2-cycle delay	Yes
Bus bridge of sDMAC	Duplication and comparator	No
Cluster RAM Controller	Duplication and comparator	No
SAXI2FAXI bridge (from System Bus to Global FLASH Bus)	Duplication and comparator	No
SAXI2MBI bridge (from System Bus to CRAM Bus)	Duplication and comparator	No
VCI2APB bridge (from P-Bus to each P-Bus Group)	Duplication and comparator	No
VCI2AXI bridge (from Inter-processor element Bus to System Bus)	Duplication and comparator	No
AXI2VCI bridge (from System Bus to P-Bus)	Duplication and comparator	No
AXI2VCI bridge (from System Bus to Inter-processor element Bus)	Duplication and comparator	No
GCFU	Duplication and comparator	No
CCIB	Comparator of input and output signals	No

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.4.2 Error Notifications

The error notifications of this module are listed in the following table.

Table 44.298 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
DTS compare error	DTS redundant lock step compare error	36	—
BUS Bridge compare error of sDMAC	Duplication comparator error detected in sDMAC0	37	—
	Duplication comparator error detected in sDMAC1		
Bus Bridge compare error	Duplication comparator error detected in CRAM controller	38	—
	Duplication comparator error detected in SAXI2FAXI bridge (from System Bus to Global FLASH Bus)		
	Duplication comparator error detected in SAXI2MBI bridge (from System Bus to CRAM Bus)		
	Duplication comparator error detected in VCI2AXI bridge (from Inter-processor element Bus to System Bus)		
	Duplication comparator error detected in AXI2VCI bridge (from System Bus to P-Bus)		
	Duplication comparator error detected in AXI2VCI bridge (from System Bus to Inter-processor element Bus)		
	Duplication comparator error detected in VCI2APB bridge (from P-Bus to each P-Bus Group)		
	Duplication comparator error detected in GCFU block for SAXI masters		
	Duplication comparator error detected in GCFU block for instruction fetch or data access		
	Comparator error detected in CCIB block		
DCLS compare error (PE0)	Redundant lock step compare error	224	—
DCLS compare error (PE1)	Redundant lock step compare error	256	—
DCLS compare error (PE2)	Redundant lock step compare error	288	—
DCLS compare error (PE3)	Redundant lock step compare error	320	—

44.4.3 List of Registers

Table 44.299 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
DTS_COMPC	Error injection control register	DTS_COMP_CNTRL	FFC5 8200$_H$	32	PBG00#2	DTS_COMP_CNTRL

44.4.4 DTS_COMP_CNTRL — Error Injection Control Register

This register can control the output signals on the checker side of the DTS. A DTS comparison error can be generated by setting this register.

Writing to this register should be executed with PROT[31:30] = 01_B.

Access: This register can be read or written in 32-bit units.

Address: FFC5 8200_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROT[31:30]		DTSCMPERR[29:16]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTSCMPERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.300 DTS_COMP_CNTRL Register Contents

Bit Position	Bit Name	Function
31, 30	PROT[31:30]	Protection Bit 01_B: Writing is enabled Other: Writing is disabled These bits are always read as 0.
29 to 0	DTSCMPERR[29:0]	A DMA comparison error can be generated by writing 111111_11111011_11111111_11111111 $_B$ to DTSCMPERR together with the PROT bit. Clear all these bits to 0 if there is no need to generate this error.

44.4.5 Usage Note

Reading a register with a value that is undefined after a reset without initializing the register may lead to a CPU comparison error. Accordingly, such registers must be initialized with the desired settings. Even if the branch instruction and the subsequent instruction is issued in parallel, the CPU comparison error might be occurred by undefined register after the reset. It should be applied as specified below until the register which refer by subsequent instruction is initialized in case of branching in the preceding instruction.

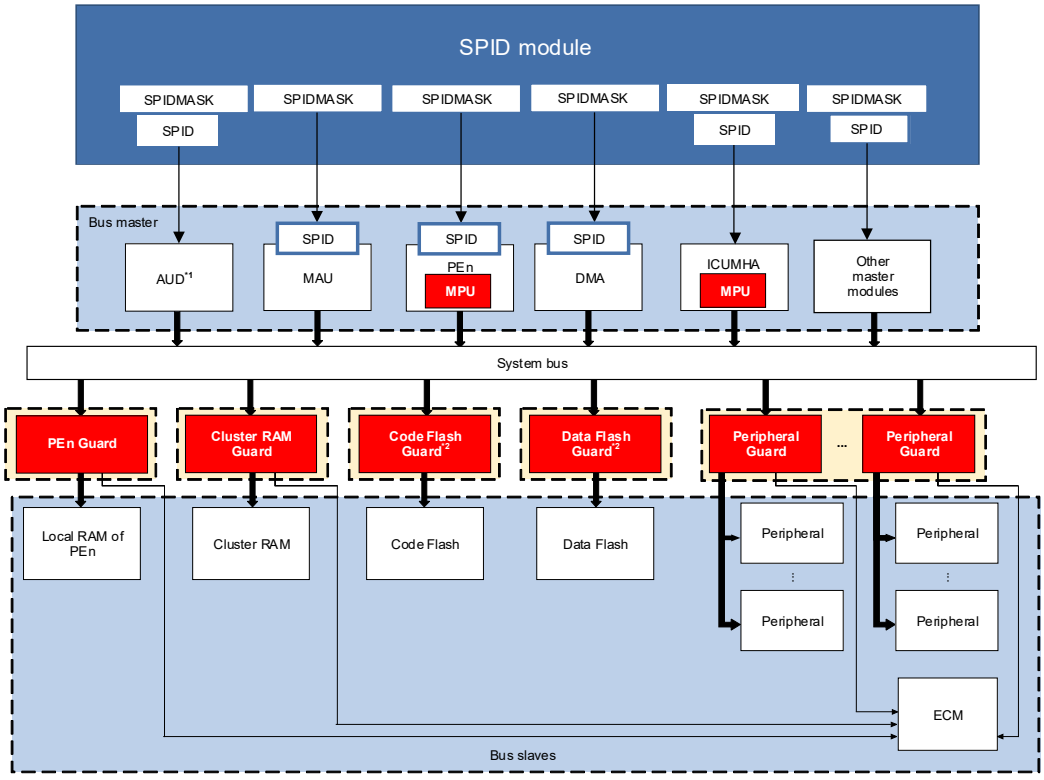
– Insert the SYNCI instruction or the RIE instruction following the branch instruction. (It has to be added by assembler language. When C language is used, it could be optimized.)

– Applicable branch instructions:
  - Bcond except BR
  - JARL, JMP

44.5 Memory Protection

44.5.1 Overview

This product incorporates the memory protection/guard functions to prevent erroneous access to data in the memory and control registers of the peripheral modules. **Figure 44.1** shows the overall memory protection architecture. Each programmable core (bus master) has a Memory Protection Unit (MPU) that defines the software access protection. In addition, each resource (bus slave) has a guard that controls the access by any bus master, including ones that do not have an MPU such as the DMA.



Note 1. AUD function is only supported in RH850/U2A-EVA.

Note 2. For details of Code Flash Guard and Data Flash Guard, See the *RH850/U2A-EVA Group Security User's Manual: Hardware.*

Figure 44.1 Memory Protection Architecture

- **MPU**

  All CPU cores and ICUMHA protect the memory against unintended access from the CPU cores and ICUMHA themselves. Requests to access addresses that are prohibited by the MPU are never issued by the CPU cores or ICUMHA. For details, see **Section 3, CPU System** and the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

- **Slave Guard**

  A specific memory is protected against unintended access from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.

  – PEG

    Local RAM and INTC1 are protected against unintended access except from the CPU incorporating the Local RAM and INTC1 themselves. For example, access from CPU0 to Local RAM in CPU0 is not rejected by the PEG.
    For details, refer to **Section 44.5.4, PEG**.

  – CRG

    Cluster RAM is protected against unintended access.

  – Peripheral Guard

    The control registers in the peripheral circuits and memory are protected against unintended access.
    Peripheral Guard includes INTC2 Guard, DTS Guard, sDMAC Guard, IBG, HBG and PBG.
    For details, refer to **Section 44.5.6, INTC2 Guard** to **Section 44.5.11, HBG**.

For this purpose, each guard has following common guard attributes for memory protection.

In addition to access address, each bus access contains the following access attributes that identifies the bus master for error information capture.

Table 44.301 Common guard attributes for each slave guard

Priority	Guard Attribute	Contents
High	SEC*1	Any non-secure masters are prohibited to access to the target slave if the attribute is enabled.
	DBG	Any debug masters can access to the target slave if this attribute is enabled.
↓	UM	Any masters in user mode are prohibited to access to the target slave if this attribute is disabled.
	SPID[m]	Any masters whose SPIDs are equivalent to m can access to the target slave if this attribute is enabled.
	WG	Any masters can write to the target slave if this attribute is enabled.
Low	RG	Any masters can read from the target slave if this attribute is enabled.

Note 1. This attribute is available for only PBG6L1.

Table 44.302 Access attributes of each bus access

Access attribute	Function
UM	Indicates the operating mode of bus master when the all bus masters access to an address. 0: Supervisor mode 1: User mode When CPU or ICUMHA*1 accesses an address, this access attribute indicates the value set in system register of CPU core. When DTS or sDMAC or MAU*2 accesses an address, this access attribute indicates the value set in each dedicated register. When the ICUM_AES0*1 accesses an address, the value of this access attribute is always 1. When the other masters accesses an address, the value of this access attribute is always 0.
SEC	Indicates the access mode of the ICUMHA or ICUM_AES0 when ICUMHA or ICUM_AES0 accesses an address. 0: Non-secure 1: Secure When the other master accesses an address, the value of this access attribute is always 0.
DBG	Indicates the access mode of the debug modules when the debug module accesses an address. 0: Normal mode 1: Debug mode When the other master accesses an address, the value of this access attribute is always 0.
SPID	Indicates the system protection identifier SPID that is assigned by themselves when all bus masters access to an address. When CPU accesses an address, this access attribute indicates the value set in the system register of CPU core. When DTS or sDMAC or MAU accesses an address, this access attribute indicates the value set in each dedicated register. When the other master accesses an address, this access attribute indicates the value set in SPID module. See **Section 44.5.3, SPID** for details.
WRITE	Indicates the access type of bus master. 0: Read access 1: Write access

Note 1. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Note 2. MAU is one of the debug functions to read and write the memory and registers.

- **Default MPU and Guard Configuration**

  The MPU of each PE and the ICUMHA is disabled after reset.
  The PBG and HBG are enabled after reset and access to the protected area is limited by the initial settings of the configuration registers to protect against unintended access. For details, see the reset values of the configuration registers of PBG and HBG in each section.
  All other guards require a configuration setting first to define the protected address range or modules. Therefore, their regions are disabled by default. See the reset values of the registers in the related sections for more information.

44.5.2 Error Notifications

The error notifications of these modules are listed in the following table. MPU can generate exceptions. For details, see **Section 3, CPU System** and the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Table 44.303 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
CRAM Guard error	CRG error	168	√
I-Bus Guard error	IBG error	169	√
P-Bus Guard error	DTS guard error sDMAC guard error INTC2 guard error PBG error	170	√
H-Bus Guard error	HBG error	171	√
PE Guard error (PE0)	PEG error by access to PE0 slave	240	√
PE Guard error by PE0 read access (PE0)	PEG error detected in a read request from PE0 to the other LRAM	241	√
PE Guard error (PE1)	PEG error by access to PE1 slave	272	√
PE Guard error by PE1 read access (PE1)	PEG error detected in a read request from PE1 to the other LRAM	273	√
PE Guard error (PE2)	PEG error by access to PE2 slave	304	√
PE Guard error by PE2 read access (PE2)	PEG error detected in a read request from PE2 to the other LRAM	305	√
PE Guard error (PE3)	PEG error by access to PE3 slave	336	√
PE Guard error by PE3 read access (PE3)	PEG error detected in a read request from PE3 to the other LRAM	337	√

44.5.3 SPID

44.5.3.1 Overview

SPID (System Protection identifier) indicates an identifier of the software task. The value of SPID for CPU cores, DTS/sDMAC and MAU[*2] can be set individually in each configuration register. For the other masters, it can be set in SPID module.

SPID module supervises all SPIDs except for some bus masters which have own configuration register. These SPID registers present in SPID module are configurable by software. In addition, the SPID module also supervises SPIDs which all bus masters use through SPID mask registers.

SPID mask registers can be locked by SPID mask lock registers for security purposes. Once SPID mask lock register locks a SPID mask register, it can be cleared by Power Up Reset, System Reset1, System Reset2, Application Reset and DeepSTOP Reset.

Table 44.304 shows the initial values of SPID for each bus master.

Table 44.304 Initial Value and Register Location of SPID

Initial value of SPID[*4]	Bus master	Register location
0	CPU0	CPU0
1	CPU1	CPU1
2	CPU2[*3]	CPU2
3	CPU3[*3]	CPU3
4 to 16	Reserved	-
17	Gb Ether[*6]	SPID module
18	Reserved	-
19	RHSIF0[*6]	SPID module
20	Reserved	-
21	Reserved	-
22	FlexRay1[*6]	SPID module
23	FlexRay0	SPID module
24	Fast Ether[*7]	SPID module
25	ICUM_AES0	SPID module
26	ICUMHA	SPID module
27	sDMAC1	sDMAC1
28	sDMAC0	sDMAC0
29	DTS[*1]	DTS
30	AUD[*5]	SPID module
31	MAU	MAU

Note 1. Initial value of SPID for DTS is stored in DTSRAM. Therefore, it is undefined after reset and it becomes 0 after RAM initialization. Refer to **Section 9.5.6, RAM Initialization**.

Note 2. MAU is one of the debug functions to read and write the memory and registers.

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 4. The initial value shown in this table is the same as the 'Bus master index n' in the register description in this section.

Note 5. AUD function is only supported in RH850/U2A-EVA.

Note 6. This function is not implemented in RH850/U2A6.

Note 7. This function is not implemented in RH850/U2A6 (156 pins).

Features of SPID:

- SPID of each bus master is configurable by software with high flexibility according to user's system.
- It is possible to assign one SPID to one bus master or assign one SPID to several bus masters.
- It is possible to limit SPID that each bus master uses by SPIDMASK register. Moreover this configuration can be locked.
- The lock function above prevents unauthorized use of SPID for security purpose.

Table 44.305 shows a case which eight SPIDs are allocated only to CPU0 and other SPIDs are allocated to each remained bus master. **Table 44.306** shows a case which some SPIDs are allocated to each CPU and some peripheral bus masters use one same SPID.

The system integrator should define the SPID assignment according to the system.

Table 44.305 Example of SPID setting (Several SPIDs are allocated only to CPU0)

SPID value	Bus master	Lock/Unlock
0	CPU0	Unlocked
1	CPU1	Locked
2	CPU2*1	Locked
3	CPU3*1	Locked
4 to 10	CPU0	Unlocked
11 to 16	Reserved	-
17	Gb Ether*3	Locked
18	Reserved	-
19	RHSIF0*3	Locked
20	Reserved	-
21	Reserved	-
22	FlexRay1*3	Locked
23	FlexRay0	Locked
24	Fast Ether*4	Locked
25	ICUM_AES0	Locked
26	ICUMHA	Locked
27	sDMAC1	Locked
28	sDMAC0	Locked
29	DTS	Locked
30	AUD*2	Locked
31	MAU	Locked

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.
Note 2. AUD function is only supported in RH850/U2A-EVA.
Note 3. This function is not implemented in RH850/U2A6.
Note 4. This function is not implemented in RH850/U2A6 (156 pins).

Table 44.306 Example of SPID setting (Several SPIDs are allocated to each CPU)

SPID value	Bus master	Lock/Unlock
0	CPU0	Unlocked
1	CPU1	Unlocked
2	CPU2*1	Unlocked
3	CPU3*1	Unlocked
4 to 12	CPU0	Unlocked
13 to 21	CPU1	Unlocked
22	CPU2*1	Unlocked
23	CPU3*1	Unlocked
24	RHSIF0*3/FlexRay0/FlexRay1*3/Gb Ether*3/Fast Ether*4	Locked
25	ICUM_AES0	Locked
26	ICUMHA	Locked
27	sDMAC1	Locked
28	sDMAC0	Locked
29	DTS	Locked
30	AUD*2	Locked
31	MAU	Locked

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.
Note 2. AUD function is only supported in RH850/U2A-EVA.
Note 3. This function is not implemented in RH850/U2A6.
Note 4. This function is not implemented in RH850/U2A6 (156 pins).

44.5.3.2 List of Registers

Table 44.307 List of SPID Registers (1/2)

Module Name	Register Name*1	Symbol	Address	Access Size	Access Protection	
					PBG	Other
SPIDCTL (P-Bus Group 9)	BusMaster 17 SPID Register	BM17SPID*4	FF0A 8044$_H$	8, 16, 32	PBG90#3	—
	BusMaster 19 SPID Register	BM19SPID*4	FF0A 804C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 22 SPID Register	BM22SPID*4	FF0A 8058$_H$	8, 16, 32	PBG90#3	—
	BusMaster 23 SPID Register	BM23SPID	FF0A 805C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 24 SPID Register	BM24SPID*5	FF0A 8060$_H$	8, 16, 32	PBG90#3	—
	BusMaster 25 SPID Register	BM25SPID	FF0A 8064$_H$	8, 16, 32	PBG90#3	—
	BusMaster 26 SPID Register	BM26SPID	FF0A 8068$_H$	8, 16, 32	PBG90#3	—
	BusMaster 30 SPID Register	BM30SPID*3	FF0A 8078$_H$	8, 16, 32	PBG90#3	—
	BusMaster 00 SPID mask Register	BM00SPIDMSK	FF0A 8100$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 01 SPID mask Register	BM01SPIDMSK	FF0A 8104$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 02 SPID mask Register	BM02SPIDMSK*2	FF0A 8108$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 03 SPID mask Register	BM03SPIDMSK*2	FF0A 810C$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 17 SPID mask Register	BM17SPIDMSK*4	FF0A 8144$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 19 SPID mask Register	BM19SPIDMSK*4	FF0A 814C$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 22 SPID mask Register	BM22SPIDMSK*4	FF0A 8158$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 23 SPID mask Register	BM23SPIDMSK	FF0A 815C$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 24 SPID mask Register	BM24SPIDMSK*5	FF0A 8160$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 25 SPID mask Register	BM25SPIDMSK	FF0A 8164$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 26 SPID mask Register	BM26SPIDMSK	FF0A 8168$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 27 SPID mask Register	BM27SPIDMSK	FF0A 816C$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 28 SPID mask Register	BM28SPIDMSK	FF0A 8170$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 29 SPID mask Register	BM29SPIDMSK	FF0A 8174$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 30 SPID mask Register	BM30SPIDMSK*3	FF0A 8178$_H$	8, 16, 32	PBG90#3	SPIDKCPROT
	BusMaster 31 SPID mask Register	BM31SPIDMSK	FF0A 817C$_H$	8, 16, 32	PBG90#3	SPIDKCPROT

Table 44.307 List of SPID Registers (2/2)

Module Name	Register Name*1	Symbol	Address	Access Size	Access Protection	
					PBG	Other
SPIDCTL (P-Bus Group 9)	BusMaster 00 SPID mask Lock Register	BM00SPIDMSKLOCK	FF0A 8200$_H$	8, 16, 32	PBG90#3	—
	BusMaster 01 SPID mask Lock Register	BM01SPIDMSKLOCK	FF0A 8204$_H$	8, 16, 32	PBG90#3	—
	BusMaster 02 SPID mask Lock Register	BM02SPIDMSKLOCK*2	FF0A 8208$_H$	8, 16, 32	PBG90#3	—
	BusMaster 03 SPID mask Lock Register	BM03SPIDMSKLOCK*2	FF0A 820C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 17 SPID mask Lock Register	BM17SPIDMSKLOCK*4	FF0A 8244$_H$	8, 16, 32	PBG90#3	—
	BusMaster 19 SPID mask Lock Register	BM19SPIDMSKLOCK*4	FF0A 824C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 22 SPID mask Lock Register	BM22SPIDMSKLOCK*4	FF0A 8258$_H$	8, 16, 32	PBG90#3	—
	BusMaster 23 SPID mask Lock Register	BM23SPIDMSKLOCK	FF0A 825C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 24 SPID mask Lock Register	BM24SPIDMSKLOCK*5	FF0A 8260$_H$	8, 16, 32	PBG90#3	—
	BusMaster 25 SPID mask Lock Register	BM25SPIDMSKLOCK	FF0A 8264$_H$	8, 16, 32	PBG90#3	—
	BusMaster 26 SPID mask Lock Register	BM26SPIDMSKLOCK	FF0A 8268$_H$	8, 16, 32	PBG90#3	—
	BusMaster 27 SPID mask Lock Register	BM27SPIDMSKLOCK	FF0A 826C$_H$	8, 16, 32	PBG90#3	—
	BusMaster 28 SPID mask Lock Register	BM28SPIDMSKLOCK	FF0A 8270$_H$	8, 16, 32	PBG90#3	—
	BusMaster 29 SPID mask Lock Register	BM29SPIDMSKLOCK	FF0A 8274$_H$	8, 16, 32	PBG90#3	—
	BusMaster 30 SPID mask Lock Register	BM30SPIDMSKLOCK*3	FF0A 8278$_H$	8, 16, 32	PBG90#3	—
	BusMaster 31 SPID mask Lock Register	BM31SPIDMSKLOCK	FF0A 827C$_H$	8, 16, 32	PBG90#3	—
	SPID Key code Protection Register	SPIDKCPROT	FF0A 8300$_H$	32	PBG90#3	—

Note 1. For "Bus master index n" in register name of below sections, refer **Table 44.304**.

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 3. These registers are only implemented in RH850/U2A-EVA device because AUD (Advanced User Debugger) function is just supported by RH850/U2A-EVA.

Note 4. This function is not implemented in RH850/U2A6.

Note 5. This function is not implemented in RH850/U2A6 (156 pins).

44.5.3.3 BMnSPID — BusMaster n SPID Register

This register is used to set the SPID of each bus master.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: BM17SPID:FF0A 8044$_H$
BM19SPID:FF0A 804C$_H$
BM22SPID:FF0A 8058$_H$
BM23SPID:FF0A 805C$_H$
BM24SPID:FF0A 8060$_H$
BM25SPID:FF0A 8064$_H$
BM26SPID:FF0A 8068$_H$
BM30SPID:FF0A 8078$_H$

Value after reset: 0000 00XX$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	(defined for each bus master)*1				
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W				

Table 44.308 BMnSPID Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID for each bus master.

Note 1. See **Section 44.5.3.1, Overview** for the initial value of each SPID.

44.5.3.4 BMnSPIDMSK — BusMaster n SPID Mask Register

This register is used to set the SPID that each bus-master can use.

This register is protected by key-code register.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: BM00SPIDMSK:FF0A 8100$_H$*1
BM01SPIDMSK:FF0A 8104$_H$*1
BM02SPIDMSK:FF0A 8108$_H$*1
BM03SPIDMSK:FF0A 810C$_H$*1
BM17SPIDMSK:FF0A 8144$_H$
BM19SPIDMSK:FF0A 814C$_H$
BM22SPIDMSK:FF0A 8158$_H$
BM23SPIDMSK:FF0A 815C$_H$
BM24SPIDMSK:FF0A 8160$_H$
BM25SPIDMSK:FF0A 8164$_H$
BM26SPIDMSK:FF0A 8168$_H$
BM27SPIDMSK:FF0A 816C$_H$
BM28SPIDMSK:FF0A 8170$_H$
BM29SPIDMSK:FF0A 8174$_H$
BM30SPIDMSK:FF0A 8178$_H$
BM31SPIDMSK:FF0A 817C$_H$

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDMSK[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDMSK[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.309 BMnSPIDMSK Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDMSK[31:0]	Set inhibition of SPID value which that each bus master can use. Each bit corresponds to a single SPID value. 0: The bus-master cannot use this SPID. Setting of this SPID is inhibited. 1: The bus-master can use this SPID. Setting of this SPID is not inhibited.

Note 1. The setting value of this register is reflected to SPIDLIST register in CPU.
For details, see **Section 3, CPU System**.

44.5.3.5 BMnSPIDMSKLOCK — BusMaster n SPID Mask Lock Register

This register is used to set write-mask to each BMnSPIDMSK.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: BM00SPIDMSKLOCK:FF0A 8200$_H$
BM01SPIDMSKLOCK:FF0A 8204$_H$
BM02SPIDMSKLOCK:FF0A 8208$_H$
BM03SPIDMSKLOCK:FF0A 820C$_H$
BM17SPIDMSKLOCK:FF0A 8244$_H$
BM19SPIDMSKLOCK:FF0A 824C$_H$
BM22SPIDMSKLOCK:FF0A 8258$_H$
BM23SPIDMSKLOCK:FF0A 825C$_H$
BM24SPIDMSKLOCK:FF0A 8260$_H$
BM25SPIDMSKLOCK:FF0A 8264$_H$
BM26SPIDMSKLOCK:FF0A 8268$_H$
BM27SPIDMSKLOCK:FF0A 826C$_H$
BM28SPIDMSKLOCK:FF0A 8270$_H$
BM29SPIDMSKLOCK:FF0A 8274$_H$
BM30SPIDMSKLOCK:FF0A 8278$_H$
BM31SPIDMSKLOCK:FF0A 827C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.310 BMnSPIDMSKLOCK Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LOCK	Lock of register 0: Register (BMnSPIDMSK) can be re-written 1: Any further write to the register (BMnSPIDMSK) is ignored. LOCK can be cleared by Power On Reset, System Reset 1, System Reset 2, and Application Reset and DeepSTOP Reset.

44.5.3.6 SPIDKCPROT — SPID Key Code Protection Register

This register is used to set the key-code lock or unlock of BMnSPIDMSK.

Access: This register can be read or written in 32-bit units.

Address: FF0A 8300$_H$

Value after reset: 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.311 SPIDKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.
Write A5A5A501$_H$ to this register to enable writing protected registers.

44.5.3.7 Usage

SPID is used for one of the identifiers of the memory protection/guard functions. SPID setting procedure is as follows.

1. Set the SPID key code protection register (SPIDKCPROT) to enable writing to SPID registers.
2. Set the SPID mask register (BMnSPIDMSK) to define the SPID values that the bus master can set according to system FFI.
3. Set the SPID mask lock register (BMnSPIDMSKLOCK) to prohibit changing the SPID mask register setting for security purposes.
4. Set the value allowed by the SPID mask register (BMnSPIDMSK) to the SPID register (BMnSPID) for the next time the bus master is accessed.
5. Set the SPID key code protection register (SPIDKCPROT) to disable writing to SPID registers.
6. Configure the memory protection/guard settings to prevent erroneous access identified by SPID and other Identifiers.

44.5.4 PEG

44.5.4.1 Overview

Each PE has a guard that controls the access to the Local RAM and INTC1 from other bus masters. Access by bus masters can be controlled via their bus context for eight regions through this PE Guard (PEG). Bus masters can only access the Local RAM area and INTC1 register area if permission is granted. PEG can prevent read*1 and write accesses against which the Local RAM area and INTC1 register area should be protected. If PEG detects illegal access, guard error is reported to ECM.

If the guard enable bit (GEN) of all PEGPROTm registers is disabled, all masters can access to both the Local RAM area and INTC1 registers.

If the guard enable bit (GEN) of either PEGPROTm registers is enabled, an access to Local RAM area and INTC1 registers is limited according to that setting.

In case of overlap within PEG channels, the access is granted to the bus masters if at least one guard permits the access.

Note 1. Including instruction fetch access.

44.5.4.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as an offset from the base addresses.

Table 44.312 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_PE0CL0_base>	FFC6 $C000_H$	P-Bus Group 0
<GUARD_PE1CL0_base>	FFC6 $C100_H$	P-Bus Group 0
<GUARD_PE2CL1_base>*1	FFC6 $C200_H$	P-Bus Group 0
<GUARD_PE3CL1_base>*1	FFC6 $C300_H$	P-Bus Group 0
<PEGCAP_M_PE0CL0_base>	FFC6 $C800_H$	P-Bus Group 0
<PEGCAP_S_PE0CL0_base>	FFC6 $C820_H$	P-Bus Group 0
<PEGCAP_M_PE1CL0_base>	FFC6 $C840_H$	P-Bus Group 0
<PEGCAP_S_PE1CL0_base>	FFC6 $C860_H$	P-Bus Group 0
<PEGCAP_M_PE2CL1_base>*1	FFC6 $C880_H$	P-Bus Group 0
<PEGCAP_S_PE2CL1_base>*1	FFC6 $C8A0_H$	P-Bus Group 0
<PEGCAP_M_PE3CL1_base>*1	FFC6 $C8C0_H$	P-Bus Group 0
<PEGCAP_S_PE3CL1_base>*1	FFC6 $C8E0_H$	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.5.4.3 List of Registers

Table 44.313 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_PEnCL0 (n = 0, 1)	Key code protection register	PEGKCPROT	<GUARD_PEnCL0_base> + 00_H	32	*1	—
	Channel protection control register	PEGPROTm (m = 0-7)	<GUARD_PEnCL0_base> + 40_H + m × 10_H	8, 16, 32	*1	PEGKCPROT
	Channel SPID setting register	PEGSPIDm (m = 0-7)	<GUARD_PEnCL0_base> + 44_H + m × 10_H	8, 16, 32	*1	PEGKCPROT
	Channel base address setting register	PEGBADm (m = 0-7)	<GUARD_PEnCL0_base> + 48_H + m × 10_H	32	*1	PEGKCPROT
	Channel valid bit setting register	PEGADVm (m = 0-7)	<GUARD_PEnCL0_base> + $4C_H$ + m × 10_H	32	*1	PEGKCPROT
GUARD_PEnCL1 (n = 2, 3)*2	Key code protection register	PEGKCPROT	<GUARD_PEnCL1_base> + 00_H	32	*1	—
	Channel protection control register	PEGPROTm (m = 0-7)	<GUARD_PEnCL1_base> + 40_H + m × 10_H	8, 16, 32	*1	PEGKCPROT
	Channel SPID setting register	PEGSPIDm (m = 0-7)	<GUARD_PEnCL1_base> + 44_H + m × 10_H	8, 16, 32	*1	PEGKCPROT
	Channel base address setting register	PEGBADm (m = 0-7)	<GUARD_PEnCL1_base> + 48_H + m × 10_H	32	*1	PEGKCPROT
	Channel valid bit setting register	PEGADVm (m = 0-7)	<GUARD_PEnCL1_base> + $4C_H$ + m × 10_H	32	*1	PEGKCPROT
PEGCAP_M_PEnCL0 (n = 0, 1)	Guard error overflow clear register	PEGOVFCLR	<PEGCAP_M_PEnCL0_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	PEGOVFSTAT	<PEGCAP_M_PEnCL0_base> + 04_H	8, 32	*1	—
	Guard error address register	PEGERRADDR	<PEGCAP_M_PEnCL0_base> + 08_H	32	*1	—
	Guard error access information register	PEGERRTYPE	<PEGCAP_M_PEnCL0_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	PEGSPIDERRCLR	<PEGCAP_M_PEnCL0_base> + 10_H	32	*1	—
	Guard SPID error status register	PEGSPIDERRSTAT	<PEGCAP_M_PEnCL0_base> + 14_H	32	*1	—
PEGCAP_S_PEnCL0 (n = 0, 1)	Guard error overflow clear register	PEGOVFCLR	<PEGCAP_S_PEnCL0_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	PEGOVFSTAT	<PEGCAP_S_PEnCL0_base> + 04_H	8, 32	*1	—
	Guard error address register	PEGERRADDR	<PEGCAP_S_PEnCL0_base> + 08_H	32	*1	—
	Guard error access information register	PEGERRTYPE	<PEGCAP_S_PEnCL0_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	PEGSPIDERRCLR	<PEGCAP_S_PEnCL0_base> + 10_H	32	*1	—
	Guard SPID error status register	PEGSPIDERRSTAT	<PEGCAP_S_PEnCL0_base> + 14_H	32	*1	—
PEGCAP_M_PEnCL1 (n = 2, 3)*2	Guard error overflow clear register	PEGOVFCLR	<PEGCAP_M_PEnCL1_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	PEGOVFSTAT	<PEGCAP_M_PEnCL1_base> + 04_H	8, 32	*1	—
	Guard error address register	PEGERRADDR	<PEGCAP_M_PEnCL1_base> + 08_H	32	*1	—
	Guard error access information register	PEGERRTYPE	<PEGCAP_M_PEnCL1_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	PEGSPIDERRCLR	<PEGCAP_M_PEnCL1_base> + 10_H	32	*1	—
	Guard SPID error status register	PEGSPIDERRSTAT	<PEGCAP_M_PEnCL1_base> + 14_H	32	*1	—
PEGCAP_S_PEnCL1 (n = 2, 3)*2	Guard error overflow clear register	PEGOVFCLR	<PEGCAP_S_PEnCL1_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	PEGOVFSTAT	<PEGCAP_S_PEnCL1_base> + 04_H	8, 32	*1	—
	Guard error address register	PEGERRADDR	<PEGCAP_S_PEnCL1_base> + 08_H	32	*1	—
	Guard error access information register	PEGERRTYPE	<PEGCAP_S_PEnCL1_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	PEGSPIDERRCLR	<PEGCAP_S_PEnCL1_base> + 10_H	32	*1	—
	Guard SPID error status register	PEGSPIDERRSTAT	<PEGCAP_S_PEnCL1_base> + 14_H	32	*1	—

Note 1. n=0: PBG01#0
n=1: PBG01#1
n=2: PBG01#2
n=3: PBG01#3

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.314 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
PE0 INTC1 registers	GUARD_PE0CL0	PEGCAP_S_PE0CL0	Access request from other than PE0
PE0 LRAM			Store request from other than PE0
			Fetch request from all PEs *1
			Access request from other than PEs
		PEGCAP_M_PE1CL0	Load request from PE1
		PEGCAP_M_PE2CL1	Load request from PE2
		PEGCAP_M_PE3CL1	Load request from PE3
PE1 INTC1 registers	GUARD_PE1CL0	PEGCAP_S_PE1CL0	Access request from other than PE1
PE1 LRAM			Store request from other than PE1
			Fetch request from all PEs *1
			Access request from other than PEs
		PEGCAP_M_PE0CL0	Load request from PE0
		PEGCAP_M_PE2CL1	Load request from PE2
		PEGCAP_M_PE3CL1	Load request from PE3
PE2 INTC1 registers*2	GUARD_PE2CL1	PEGCAP_S_PE2CL1	Access request from other than PE2
PE2 LRAM*2			Store request from other than PE2
			Fetch request from all PEs *1
			Access request from other than PEs
		PEGCAP_M_PE0CL0	Load request from PE0
		PEGCAP_M_PE1CL0	Load request from PE1
		PEGCAP_M_PE3CL1	Load request from PE3
PE3 INTC1 registers*2	GUARD_PE3CL1	PEGCAP_S_PE3CL1	Access request from other than PE3
PE3 LRAM*2			Store request from other than PE3
			Fetch request from all PEs *1
			Access request from other than PEs
		PEGCAP_M_PE0CL0	Load request from PE0
		PEGCAP_M_PE1CL0	Load request from PE1
		PEGCAP_M_PE2CL1	Load request from PE2

Note 1. It is necessary to give read permission even when fetching instructions from its own LRAM.

Note 2. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

44.5.4.4 PEGKCPROT — Key Code Protection Register

This register is used to unlock/lock key protection of the PE guard register.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_PEnCL0_base> + 00_H
<GUARD_PEnCL1_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.315 PEGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.4.5 PEGPROTm — Channel Protection Control Register

Specifies the PE resource access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_PEnCL0_base> + 40_H + m × 10_H
<GUARD_PEnCL1_base> + 40_H + m × 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.316 PEGPROTm Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, PEGSPIDm is used as a judgment condition. 1: During write, PEGSPIDm is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, PEGSPIDm is used as the judgment condition. 1: During read, PEGSPIDm is not used as a judgment condition.

44.5.4.6 PEGSPIDm — Channel SPID Setting Register

Specifies the PE resource access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_PEnCL0_base> + 44_H + m * 10_H
<GUARD_PEnCL1_base> + 44_H + m * 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.317 PEGSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID PEGSPIDm is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting PEGSPIDm to 0101_B enables access to areas SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = x

44.5.4.7 PEGBADm — Channel Base Address Setting Register

This register is used to define area n when the setting of PEGPROTm and PEGSPIDm is valid. PEGBADm specifies base address of area n and PEGADVm specifies valid bits of PEGBADm.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_PEnCL0_base> + 48_H + m * 10_H
<GUARD_PEnCL1_base> + 48_H + m * 10_H

Value after reset: FC00 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	BAD[25:16]									
Value after reset	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAD[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.318 PEGBADm Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 12	BAD[25:12]	Base address
11 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

44.5.4.8 PEGADVm — Channel Valid Bit Setting Register

This register is used to define area n when the setting of PEGPROTm and PEGSPIDm is valid. PEGBADm specifies base address of area n and PEGADVm specifies valid bits of PEGBADm.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_PEnCL0_base> + $4C_H$ + m * 10_H
<GUARD_PEnCL1_base> + $4C_H$ + m * 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADV[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.319 PEGADVm Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 12	ADV[25:12]	Valid bits of PEGBADm When an ADV bit is 1, the address of each access will be compared with the corresponding BAD bit of PEGBADm. When all ADV bits are set to 1, 4K bytes (the minimum unit) are targeted for protection based on the address specified by PEGBADm. When all ADV bits are set to 0, the target of protection is the whole of the Local RAM and INTC1 register area.
11 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Example: When PEGBADm[25:12] is set to $1C00_H$ and PEGADVm[25:12] is set to $3FFE_H$, PE guard protection area n is FDC0 0000_H to FDC0 $0FFF_H$ and FDC0 1000_H to FDC0 $1FFF_H$.

Commentary: When PEGBADm[25:12] is set to $1C00_H$, the base address is FDC0 0000_H and the configurable range of bits is enclosed in [ ] shown below.

1111 11[01 1100 0000 0000] 0000 0000 0000

F D C 0 0 0 0 0

When PEGADVm[25:12] is $3FFE_H$, bits set to 0 in that range and the 12 lower-order bits are ignored, so we have

1111 11[01 1100 0000 000X] XXXX XXXX XXXX

which means that the 4K byte ranges (a total of 8K byte) from

F D C 0 0 0 0 0 to

F D C 0 0 F F F,

and

F D C 0 1 0 0 0 to

F D C 0 1 F F F

will be protected.

44.5.4.9 PEGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear PEGOVFSTAT register. PEGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. 0 is always returned when reading this register.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <PEGCAP_M_PEnCL0_base> + 00_H
<PEGCAP_S_PEnCL0_base> + 00_H
<PEGCAP_M_PEnCL1_base> + 00_H
<PEGCAP_S_PEnCL1_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.320 PEGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears PEGOVFSTAT.OVF bit by writing 1
0	Reserved	When writing, write the value after reset.

44.5.4.10 PEGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. The OVF bit is set when a guard error is reported again under the condition any bit of PEGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing PEGOVFCLR register. See **Section 44.5.4.9, PEGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <PEGCAP_M_PEnCL0_base> + 04_H
<PEGCAP_S_PEnCL0_base> + 04_H
<PEGCAP_M_PEnCL1_base> + 04_H
<PEGCAP_S_PEnCL1_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.321 PEGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.4.11 PEGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when PEGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PEGCAP_M_PEnCL0_base> + 08_H
<PEGCAP_S_PEnCL0_base> + 08_H
<PEGCAP_M_PEnCL1_base> + 08_H
<PEGCAP_S_PEnCL1_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.322 PEGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.4.12 PEGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when PEGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <PEGCAP_M_PEnCL0_base> + $0C_H$
<PEGCAP_S_PEnCL0_base> + $0C_H$
<PEGCAP_M_PEnCL1_base> + $0C_H$
<PEGCAP_S_PEnCL1_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.323 PEGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.4.13 PEGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear PEGSPIDERRSTAT register. PEGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <PEGCAP_M_PEnCL0_base> + 10_H
<PEGCAP_S_PEnCL0_base> + 10_H
<PEGCAP_M_PEnCL1_base> + 10_H
<PEGCAP_S_PEnCL1_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.324 PEGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear PEGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.4.14 PEGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to PEGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Note 1. Error depends on Master and slave.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PEGCAP_M_PEnCL0_base> + 14_H
<PEGCAP_S_PEnCL0_base> + 14_H
<PEGCAP_M_PEnCL1_base> + 14_H
<PEGCAP_S_PEnCL1_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.325 PEGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.5 CRG

44.5.5.1 Overview

Cluster RAM guard (CRG) has two functions. One is function for Inter-VM Communication (IVC). For more detail of IVC see **Section 3, CPU System**. The other is function for protection of Cluster RAM as a memory guard. Bus masters can only access the Cluster RAM area if permission is granted. CRG can prevent read*1 and write accesses against which the Cluster RAM area should be protected. If CRG detects illegal access, guard error notification is signaled to ECM.

If the guard enable bit (GEN) of all CRGPROTm/CSGPROTm registers is disabled, all masters can access to the Cluster RAM area.

If the guard enable bit (GEN) of either CRGPROTm/CSGPROTm registers is enabled, an access to Cluster RAM area is limited according to that setting.

CRG as a memory guard consists of GUARD_CRAMCRGn and GUARD_CRAMCSGn. Access to the Cluster RAM can be controlled via the bus context for eight regions by GUARD_CRAMCRGn. Access to the Cluster RAM can be controlled via the bus context for four regions by GUARD_CRAMCSGn.

In case regions have an overlapping address range, the resulting access permission is different between the GUARD_CRAMCRGn and the GUARD_CRAMCSGn:

- In case of overlap within GUARD_CRAMCRGn, the access is granted to the bus masters if at least one guard permits the access.
- In case of overlap within GUARD_CRAMCSGn, the access is not granted to the bus masters if at least one guard prohibits the access.
- In case of overlap between GUARD_CRAMCRGn and GUARD_CRAMCSGn, the access permission depends on the GUARD_CRAMCSGn setting.

Note 1. Including instruction fetch access.

44.5.5.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.326 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_CRAMCRG0_base>	FFC6 D000$_H$	P-Bus Group 0
<GUARD_CRAMCRG1_base>*1	FFC6 D200$_H$	P-Bus Group 0
<GUARD_CRAMCRG2_base>*2	FFC6 D400$_H$	P-Bus Group 0
<GUARD_CRAMCRG3_base>	FFC6 D600$_H$	P-Bus Group 0
<CRGCAP_PE0CL0_base>	FFC6 D800$_H$	P-Bus Group 0
<CRGCAP_PE1CL0_base>	FFC6 D820$_H$	P-Bus Group 0
<CRGCAP_PE2CL1_base>*1	FFC6 D840$_H$	P-Bus Group 0
<CRGCAP_PE3CL1_base>*1	FFC6 D860$_H$	P-Bus Group 0
<CRGCAP_CRAMHCL0_base>	FFC6 DA00$_H$	P-Bus Group 0
<CRGCAP_CRAMLCL0_base>	FFC6 DA20$_H$	P-Bus Group 0
<CRGCAP_SX2MBHCL0_base>	FFC6 DA40$_H$	P-Bus Group 0
<CRGCAP_SX2MBLCL0_base>	FFC6 DA60$_H$	P-Bus Group 0
<CRGCAP_CRAMHCL1_base>*1	FFC6 DA80$_H$	P-Bus Group 0
<CRGCAP_CRAMLCL1_base>*1	FFC6 DAA0$_H$	P-Bus Group 0
<CRGCAP_SX2MBHCL1_base>*1	FFC6 DAC0$_H$	P-Bus Group 0
<CRGCAP_SX2MBLCL1_base>*1	FFC6 DAE0$_H$	P-Bus Group 0
<CRGCAP_CRAMHCL2_base>*2	FFC6 DB00$_H$	P-Bus Group 0
<CRGCAP_CRAMLCL2_base>*2	FFC6 DB20$_H$	P-Bus Group 0
<CRGCAP_SX2MBHCL2_base>*2	FFC6 DB40$_H$	P-Bus Group 0
<CRGCAP_SX2MBLCL2_base>*2	FFC6 DB60$_H$	P-Bus Group 0
<CRGCAP_CRAMHCL3_base>	FFC6 DB80$_H$	P-Bus Group 0
<CRGCAP_CRAMLCL3_base>	FFC6 DBA0$_H$	P-Bus Group 0
<CRGCAP_SX2MBHCL3_base>	FFC6 DBC0$_H$	P-Bus Group 0
<CRGCAP_SX2MBLCL3_base>	FFC6 DBE0$_H$	P-Bus Group 0
<GUARD_CRAMCSG0_base>	FFC6 E000$_H$	P-Bus Group 0
<GUARD_CRAMCSG1_base>*1	FFC6 E200$_H$	P-Bus Group 0
<GUARD_CRAMCSG2_base> *2	FFC6 E400$_H$	P-Bus Group 0
<GUARD_CRAMCSG3_base>	FFC6 E600$_H$	P-Bus Group 0

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

44.5.5.3 List of Registers

Table 44.327 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_CRAMCRGn (n = 0-3)*3	Key code protection register	CRGKCPROT	<GUARD_CRAMCRGn_base> + 00_H	32	*1	—
	Channel protection control register	CRGPROTm (m = 0-7)	<GUARD_CRAMCRGn_base> + 10_H + m × 20_H	8, 16, 32	*1	CRGKCPROT
	Channel SPID setting register	CRGSPIDm (m = 0-7)	<GUARD_CRAMCRGn_base> + 14_H + m × 20_H	8, 16, 32	*1	CRGKCPROT
	Channel base address setting register	CRGBADm (m = 0-7)	<GUARD_CRAMCRGn_base> + 18_H + m × 20_H	8, 16, 32	*1	CRGKCPROT
	Channel address valid bit setting register	CRGADVm (m = 0-7)	<GUARD_CRAMCRGn_base> + $1C_H$ + m × 20_H	8, 16, 32	*1	CRGKCPROT
GUARD_CRAMCSGn (n = 0-3)*3	Channel protection control register	CSGPROTm (m = 0-3)	<GUARD_CRAMCSGn_base> + 00_H + m × 20_H	8, 16, 32	*1	CRGKCPROT
	Channel SPID setting register	CSGSPIDm (m = 0-3)	<GUARD_CRAMCSGn_base> + 04_H + m * 20_H	8, 16, 32	*1	CRGKCPROT
	Channel base address setting register	CSGBADm (m = 0-3)	<GUARD_CRAMCSGn_base> + 08_H + m * 20_H	8, 16, 32	*1	CRGKCPROT
	Channel address valid bit setting	CSGADVm (m = 0-3)	<GUARD_CRAMCSGn_base> + $0C_H$ + m * 20_H	8, 16, 32	*1	CRGKCPROT
CRGCAP_PEnCL0 (n = 0, 1)	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_PEnCL0_base> + 00_H	8, 16, 32	*2	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_PEnCL0_base> + 04_H	8, 32	*2	—
	Guard error address register	CRGERRADDR	<CRGCAP_PEnCL0_base> + 08_H	32	*2	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_PEnCL0_base> + $0C_H$	16, 32	*2	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_PEnCL0_base> + 10_H	32	*2	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_PEnCL0_base> + 14_H	32	*2	—
CRGCAP_PEnCL1 (n = 2, 3)*4	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_PEnCL1_base> + 00_H	8, 16, 32	*2	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_PEnCL1_base> + 04_H	8, 32	*2	—
	Guard error address register	CRGERRADDR	<CRGCAP_PEnCL1_base> + 08_H	32	*2	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_PEnCL1_base> + $0C_H$	16, 32	*2	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_PEnCL1_base> + 10_H	32	*2	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_PEnCL1_base> + 14_H	32	*2	—
CRGCAP_CRAMHCLn (n = 0-3)*3	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_CRAMHCLn_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_CRAMHCLn_base> + 04_H	8, 32	*1	—
	Guard error address register	CRGERRADDR	<CRGCAP_CRAMHCLn_base> + 08_H	32	*1	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_CRAMHCLn_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_CRAMHCLn_base> + 10_H	32	*1	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_CRAMHCLn_base> + 14_H	32	*1	—
CRGCAP_CRAMLCLn (n = 0-3)*3	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_CRAMLCLn_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_CRAMLCLn_base> + 04_H	8, 32	*1	—
	Guard error address register	CRGERRADDR	<CRGCAP_CRAMLCLn_base> + 08_H	32	*1	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_CRAMLCLn_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_CRAMLCLn_base> + 10_H	32	*1	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_CRAMLCLn_base> + 14_H	32	*1	—

Table 44.327 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
CRGCAP_SX2MBHCLn (n = 0-3)*3	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_SX2MBHCLn_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_SX2MBHCLn_base> + 04_H	8, 32	*1	—
	Guard error address register	CRGERRADDR	<CRGCAP_SX2MBHCLn_base> + 08_H	32	*1	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_SX2MBHCLn_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_SX2MBHCLn_base> + 10_H	32	*1	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_SX2MBHCLn_base> + 14_H	32	*1	—
CRGCAP_SX2MBLCLn (n = 0-3)*3	Guard error overflow clear register	CRGOVFCLR	<CRGCAP_SX2MBLCLn_base> + 00_H	8, 16, 32	*1	—
	Guard error overflow status register	CRGOVFSTAT	<CRGCAP_SX2MBLCLn_base> + 04_H	8, 32	*1	—
	Guard error address register	CRGERRADDR	<CRGCAP_SX2MBLCLn_base> + 08_H	32	*1	—
	Guard error access information register	CRGERRTYPE	<CRGCAP_SX2MBLCLn_base> + $0C_H$	16, 32	*1	—
	Guard SPID error clear register	CRGSPIDERRCLR	<CRGCAP_SX2MBLCLn_base> + 10_H	32	*1	—
	Guard SPID error status register	CRGSPIDERRSTAT	<CRGCAP_SX2MBLCLn_base> + 14_H	32	*1	—

Note 1. n=0: PBG00#3
n=1: PBG00#4
n=2: PBG00#5
n=3: PBG00#6

Note 2. n=0: PBG01#0
n=1: PBG01#1
n=2: PBG01#2
n=3: PBG01#3

Note 3. This function is not implemented in RH850/U2A8 (373/292 pins) with index n = 1.
This function is not implemented in RH850/U2A6 with index n = 1, 2.

Note 4. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Table 44.328 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
Cluster0 RAM	GUARD_CRAMCRG0 GUARD_CRAMCSG0	CRGCAP_PE0CL0	Load request from PE0
		CRGCAP_PE1CL0	Load request from PE1
		CRGCAP_PE2CL1*1	Load request from PE2
		CRGCAP_PE3CL1*1	Load request from PE3
		CRGCAP_CRAMHCL0*3	Store request from all masters for odd words
		CRGCAP_CRAMLCL0*3	Store request from all masters for even words
		CRGCAP_SX2MBHCL0*3	Load request from other than PEs for odd words
			Fetch request from all PEs for odd words
		CRGCAP_SX2MBLCL0*3	Load request from other than PEs for even words
			Fetch request from all PEs for even words
Cluster1 RAM*1	GUARD_CRAMCRG1 GUARD_CRAMCSG1	CRGCAP_PE0CL0	Load request from PE0
		CRGCAP_PE1CL0	Load request from PE1
		CRGCAP_PE2CL1	Load request from PE2
		CRGCAP_PE3CL1	Load request from PE3
		CRGCAP_CRAMHCL1*3	Store request from all masters for odd words
		CRGCAP_CRAMLCL1*3	Store request from all masters for even words
		CRGCAP_SX2MBHCL1*3	Load request from other than PEs for odd words
			Fetch request from all PEs for odd words
		CRGCAP_SX2MBLCL1*3	Load request from other than PEs for even words
			Fetch request from all PEs for even words
Cluster2 RAM*2	GUARD_CRAMCRG2 GUARD_CRAMCSG2	CRGCAP_PE0CL0	Load request from PE0
		CRGCAP_PE1CL0	Load request from PE1
		CRGCAP_PE2CL1*1	Load request from PE2
		CRGCAP_PE3CL1*1	Load request from PE3
		CRGCAP_CRAMHCL2*3	Store request from all masters for odd words
		CRGCAP_CRAMLCL2*3	Store request from all masters for even words
		CRGCAP_SX2MBHCL2*3	Load request from other than PEs for odd words
			Fetch request from all PEs for odd words
		CRGCAP_SX2MBLCL2*3	Load request from other than PEs for even words
			Fetch request from all PEs for even words
Cluster3 RAM	GUARD_CRAMCRG3 GUARD_CRAMCSG3	CRGCAP_PE0CL0	Load request from PE0
		CRGCAP_PE1CL0	Load request from PE1
		CRGCAP_PE2CL1*1	Load request from PE2
		CRGCAP_PE3CL1*1	Load request from PE3
		CRGCAP_CRAMHCL3*3	Store request from all masters for odd words
		CRGCAP_CRAMLCL3*3	Store request from all masters for even words
		CRGCAP_SX2MBHCL3*3	Load request from other than PEs for odd words
			Fetch request from all PEs for odd words
		CRGCAP_SX2MBLCL3*3	Load request from other than PEs for even words
			Fetch request from all PEs for even words

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins), RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6.

Note 3. "even words" means bank0 and bank2, and "odd words" means bank1 and bank3. For more detail of bank information, refer to **Section 44.3.7.1, Overview**.

44.5.5.4 CRGKCPROT — Key Code Protection Register

This register is used to unlock/lock key protection of the Cluster RAM guard register.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_CRAMCRGn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.329 CRGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.5.5 CRGPROTm — Channel Protection Control Register

Specifies the Cluster RAM access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base> + 10_H + m * 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OW
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.330 CRGPROTm Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	OW	Enables/disables Inter-VM communication function. For more detail see **Section 3, CPU System**.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, CRGSPIDm is used as a judgment condition. 1: During write, CRGSPIDm is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, CRGSPIDm is used as the judgment condition. 1: During read, CRGSPIDm is not used as a judgment condition.

44.5.5.6 CRGSPIDm — Channel SPID Setting Register

Specifies the Cluster RAM access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base> + 14_H + m * 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.331 CRGSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID CRGSPIDm is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting CRGSPIDm to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = x

44.5.5.7 CRGBADm — Channel Base Address Setting Register

This register is used to define area n when the setting of CRGPROTm and CRGSPIDm is valid. CRGBADm specifies base address of area n and CRGADVm specifies valid bits of CRGBADm.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base> + 18_H + m * 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAD[20:19]		BAD[18]	BAD[17]	BAD[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	*1	*1	*2	*2	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAD[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 44.332 CRGBADm Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20, 19*1	BAD[20:19]	Base address
18*2	BAD[18]	Base address
17*2	BAD[17]	Base address
16 to 8	BAD[16:8]	Base address
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. BAD[20:19] is implemented for GUARD_CRAMCRG2. BAD[20:19] is reserved bit and read only for GUARD_CRAMCRG0 and GUARD_CRAMCRG1 and GUARD_CRAMCRG3 in RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8.
BAD[20:19] is reserved bit in RH850/U2A6.

Note 2. BAD[18] is implemented only in GUARD_CRAMCRG0 and GUARD_CRAMCRG1 and GUARD_CRAMCRG2, BAD[18] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A-EVA and RH850/U2A16.
BAD[18:17] is implemented only in GUARD_CRAMCRG0 and GUARD_CRAMCRG2, BAD[18:17] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A8.
BAD[18:17] is implemented only in GUARD_CRAMCRG0, BAD[18:17] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A6.

44.5.5.8 CRGADVm — Channel Valid Bit Setting Register

This register is used to define area n when the setting of CRGPROTm and CRGSPIDm is valid. CRGBADm specifies base address of area n and CRGADVm specifies valid bits of CRGBADm.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCRGn_base> + $1C_H$ + m * 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADV[20:19]		ADV[18]	ADV[17]	ADV[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	*1	*1	*2	*2	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 44.333 CRGADVm Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 19*1	ADV[20:19]	Valid bits of CRGBADm When an ADV bit is 1, the address of each access will be compared with the corresponding BAD bit of CRGBADm. When all ADV bits are set to 1, 256 bytes (the minimum unit) are targeted for protection based on the address specified by CRGBADm. When all ADV bits are set to 0, the target of protection is the whole of the Cluster RAM.
18*2	ADV[18]	Valid bits of CRGBADm
17*2	ADV[17]	Valid bits of CRGBADm
16 to 8	ADV[16:8]	Valid bits of CRGBADm
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. ADV[20:19] is implemented only in GUARD_CRAMCRG2. ADV[20:19] is reserved bit and read only for GUARD_CRAMCRG0 and GUARD_CRAMCRG1 and GUARD_CRAMCRG3 in RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8.
ADV[20:19] is reserved bit in RH850/U2A6.

Note 2. ADV[18] is implemented only in GUARD_CRAMCRG0 and GUARD_CRAMCRG1 and GUARD_CRAMCRG2, ADV[18] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A-EVA and RH850/U2A16.
ADV[18:17] is implemented only in GUARD_CRAMCRG0 and GUARD_CRAMCRG2, ADV[18:17] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A8.
ADV[18:17] is implemented only in GUARD_CRAMCRG0, ADV[18:17] is reserved bit and read only for GUARD_CRAMCRG3 in RH850/U2A6.

This is example for cluster #0. For valid bit of each clusters, refer to Note1 and Note2 described in CRGBADm/CRGADVm Register Contents.

Example: When CRGBADm[18:8] is set to 000_H and CRGADVm[18:8] is set to $7EF_H$, Cluster RAM guard protection area n is FE00 0000_H to FE00 $00FF_H$ and FE00 1000_H to FE00 $10FF_H$.

Commentary: When CRGBADm[18:8] is set to 000_H, the base address is FE00 0000_H and the configurable range of bits is enclosed in [ ] shown below.

1111 1110 0000 0[000 0000 0000] 0000 0000

F E 0 0 0 0 0 0

When CRGADVm[18:8] is $7EF_H$, bits set to 0 in that range and the 8 lower-order bits are ignored, so we have

1111 1110 0000 0[000 000X 0000] XXXX XXXX

which means that the 256-byte ranges (a total of 512-byte) from

F E 0 0 0 0 0 0 to

F E 0 0 0 0 F F,

and

F E 0 0 1 0 0 0 to

F E 0 0 1 0 F F

will be protected.

44.5.5.9 CSGPROTm — Channel Protection Control Register

Specifies the access to be protected for Cluster RAM. Access prohibited by any of the identifiers is protected as unauthorized access. This register is used to specify the settings for those other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCSGn_base> + 00_H + m * 20_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOCK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.334 CSGPROTm Register Contents

Bit Position	Bit Name	Function
31	LOCK	Register (CSGPROTm, CSGSPIDm, CSGBADm, CSGADVm) access lock function 0: CSGPROTm, CSGSPIDm, CSGBADm and CSGADVm register can be written. 1: CSGPROTm, CSGSPIDm, CSGBADm and CSGADVm register cannot be written. Write access to these registers is ignored. Lock function can only be cleared by reset.
30 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, CSGSPIDm is used as a judgment condition. 1: During write, CSGSPIDm is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, CSGSPIDm is used as the judgment condition. 1: During read, CSGSPIDm is not used as a judgment condition.

44.5.5.10 CSGSPIDm — Channel SPID Setting Register

Specifies the access to be protected for Cluster RAM. Access prohibited by any of the identifiers is protected as unauthorized access. This register is used to specify the SPID setting.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCSGn_base> + 04_H + m * 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.335 CSGSPIDm Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting by SPID CSGSPIDm is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting CSGSPIDm to 0101_B enables access with SPID = 0 and SPID = 2. 0: Read/Write with SPID = x depends on the RG and WG bit setting 1: Enables Read/Write with SPID = x

44.5.5.11 CSGBADm — Channel Base Address Setting Register

This register is used to define area n where the setting of CSGPROTm and CSGSPIDm is valid. CSGBADm specifies base address of area n and CSGADVm specifies valid bits of CSGBADm.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCSGn_base> + 08_H + m * 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BAD[20:19]		BAD[18]	BAD[17]	BAD[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	*1	*1	*2	*2	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAD[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 44.336 CSGBADm Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20, 19*1	BAD[20:19]	Base address
18*2	BAD[18]	Base address
17*2	BAD[17]	Base address
16 to 8	BAD[16:8]	Base address
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. BAD[20:19] is implemented for GUARD_CRAMCSG2. BAD[20:19] is reserved bit and read only for GUARD_CRAMCSG0 and GUARD_CRAMCSG1 and GUARD_CRAMCSG3 in RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8.
BAD[20:19] is reserved bit in RH850/U2A6.

Note 2. BAD[18] is implemented only in GUARD_CRAMCSG0 and GUARD_CRAMCSG1 and GUARD_CRAMCSG2, BAD[18] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A-EVA and RH850/U2A16.
BAD[18:17] is implemented only in GUARD_CRAMCSG0 and GUARD_CRAMCSG2, BAD[18:17] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A8.
BAD[18:17] is implemented only in GUARD_CRAMCSG0, BAD[18:17] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A6.

44.5.5.12 CSGADVm — Channel Valid Bit Setting Register

This register is used to define area n when the setting of CSGPROTm and CSGSPIDm is valid. CSGBADm specifies base address of area n and CSGADVm specifies valid bits of CSGBADm.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_CRAMCSGn_base> + $0C_H$ + m * 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADV[20:19]		ADV[18]	ADV[17]	ADV[16]
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	*1	*1	*2	*2	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:8]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 44.337 CSGADVm Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 19*1	ADV[20:19]	Valid bits of CSGBADm When an ADV bit is 1, the address of each access will be compared with the corresponding BAD bit of CSGBADm. When all ADV bits are set to 1, 256 bytes (the minimum unit) are targeted for protection based on the address specified by CSGBADm. When all ADV bits are set to 0, the target of protection is the whole of the Cluster RAM.
18*2	ADV[18]	Valid bits of CSGBADm
17*2	ADV[17]	Valid bits of CSGBADm
16 to 8	ADV[16:8]	Valid bits of CSGBADm
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. ADV[20:19] is implemented only in GUARD_CRAMCSG2. ADV[20:19] is reserved bit and read only for GUARD_CRAMCSG0 and GUARD_CRAMCSG1 and GUARD_CRAMCSG3 in RH850/U2A-EVA, RH850/U2A16 and RH850/U2A8.
ADV[20:19] is reserved bit in RH850/U2A6.

Note 2. ADV[18] is implemented only in GUARD_CRAMCSG0 and GUARD_CRAMCSG1 and GUARD_CRAMCSG2, ADV[18] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A-EVA and RH850/U2A16.
ADV[18:17] is implemented only in GUARD_CRAMCSG0 and GUARD_CRAMCSG2, ADV[18:17] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A8.
ADV[18:17] is implemented only in GUARD_CRAMCSG0, ADV[18:17] is reserved bit and read only for GUARD_CRAMCSG3 in RH850/U2A6.

44.5.5.13 CRGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear CRGOVFSTAT register. CRGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <CRGCAP_PEnCL0_base> + 00_H
<CRGCAP_PEnCL1_base> + 00_H
<CRGCAP_CRAMHCLn_base> + 00_H
<CRGCAP_CRAMLCLn_base> + 00_H
<CRGCAP_SX2MBHCLn_base> + 00_H
<CRGCAP_SX2MBLCLn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.338 CRGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears CRGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.5.14 CRGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard errors overflow was occurred. OVF bit is set when a guard error is reported again under the condition any bit of CRGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to CRGOVFCLR register. See **Section 44.5.5.13, CRGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <CRGCAP_PEnCL0_base> + 04_H
<CRGCAP_PEnCL1_base> + 04_H
<CRGCAP_CRAMHCLn_base> + 04_H
<CRGCAP_CRAMLCLn_base> + 04_H
<CRGCAP_SX2MBHCLn_base> + 04_H
<CRGCAP_SX2MBLCLn_base> + 04_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.339 CRGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.5.15 CRGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the CRGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CRGCAP_PEnCL0_base> + 08_H
<CRGCAP_PEnCL1_base> + 08_H
<CRGCAP_CRAMHCLn_base> + 08_H
<CRGCAP_CRAMLCLn_base> + 08_H
<CRGCAP_SX2MBHCLn_base> + 08_H
<CRGCAP_SX2MBLCLn_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.340 CRGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.5.16 CRGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the CRGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <CRGCAP_PEnCL0_base> + $0C_H$
<CRGCAP_PEnCL1_base> + $0C_H$
<CRGCAP_CRAMHCLn_base> + $0C_H$
<CRGCAP_CRAMLCLn_base> + $0C_H$
<CRGCAP_SX2MBHCLn_base> + $0C_H$
<CRGCAP_SX2MBLCLn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.341 CRGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.5.17 CRGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the CRGSPIDERRSTAT register. The CRGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <CRGCAP_PEnCL0_base> + 10_H
<CRGCAP_PEnCL1_base> + 10_H
<CRGCAP_CRAMHCLn_base> + 10_H
<CRGCAP_CRAMLCLn_base> + 10_H
<CRGCAP_SX2MBHCLn_base> + 10_H
<CRGCAP_SX2MBLCLn_base> + 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.342 CRGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear CRGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.5.18 CRGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the CRGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a Read-only register that can be read in 32-bit units.

Address: <CRGCAP_PEnCL0_base> + 14_H
<CRGCAP_PEnCL1_base> + 14_H
<CRGCAP_CRAMHCLn_base> + 14_H
<CRGCAP_CRAMLCLn_base> + 14_H
<CRGCAP_SX2MBHCLn_base> + 14_H
<CRGCAP_SX2MBLCLn_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.343 CRGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.6 INTC2 Guard

44.5.6.1 Overview

INTC2 guard is one of the slave guards used to control the access to INTC2 registers. INTC2 guard can protect respectively the read and write access against INTC2 registers of each channel. If INTC2 guard detects illegal access, guard error notification is signaled to ECM.

44.5.6.2 List of Registers

Table 44.344 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_INTC2 (P-Bus Group 0)	Guard error overflow clear register	INTC2GOVFCLR	FFC6 4000_H	8, 32	PBG00#0	—
	Guard error overflow status register	INTC2GOVFSTAT	FFC6 4004_H	8, 32	PBG00#0	—
	Guard error address register	INTC2GERRADDR	FFC6 4008_H	32	PBG00#0	—
	Guard error access information register	INTC2GERRTYPE	FFC6 $400C_H$	16, 32	PBG00#0	—
	Guard SPID error clear register	INTC2GSPIDERRCLR	FFC6 4010_H	32	PBG00#0	—
	Guard SPID error status register	INTC2GSPIDERRSTAT	FFC6 4014_H	32	PBG00#0	—
	Key code protection register	INTC2GKCPROT	FFC6 4018_H	32	PBG00#0	—
	SPID setting register	INTC2GMPIDm (m = 0-7)	FFC6 4040_H + m × 04_H	8, 32	PBG00#0	INTC2GKCPROT
	INTC2 protection control register	INTC2GPROT_GR	FFC6 $40F0_H$	32	PBG00#0	INTC2GKCPROT
	INTC2 IMR protection control register	INTC2GPROT_IMR	FFC6 40F4	32	PBG00#0	INTC2GKCPROT
	Channel protection control register	INTC2GPROT_n (n = 032-767)	FFC6 4100_H + n × 04_H	32	PBG00#0	INTC2GKCPROT

Table 44.345 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
EICn (n = 32 to 767), EEICn (n = 32 to 767)	INTC2GPROT_n (n = 32 to 767)	GUARD_INTC2	Access request from all masters
EIBDn (n = 32 to 767), I2EIBGn (n = 0 to 3)	INTC2GPROT_GR		
IMRn (n = 1 to 23)	INTC2GPROT_IMR		

44.5.6.3 INTC2GOVFCLR — Guard Error Overflow Clear Register

This register is used to clear INTC2GOVFSTAT register. INTC2GOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit or 8-bit units.

Address: FFC6 4000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.346 INTC2GOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears INTC2GOVFSTAT.OVF bit by writing 1
0	Reserved	When writing, write the value after reset.

44.5.6.4 INTC2GOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is reported again under the condition any bit of INTC2GSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to the INTC2GOVFCLR register. See **Section 44.5.6.3, INTC2GOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: FFC6 4004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.347 INTC2GOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.6.5 INTC2GERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the INTC2GSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 4008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.348 INTC2GERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.6.6 INTC2GERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the INTC2GSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: FFC6 400C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.349 INTC2GERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.6.7 INTC2GSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the INTC2GSPIDERRSTAT register.
The INTC2GSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.
Address: FFC6 4010$_H$
Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.350 INTC2GSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear INTC2GSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.6.8 INTC2GSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the INTC2GSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 4014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.351 INTC2GSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.6.9 INTC2GKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/INTC2/INTC2 IMR protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FFC6 4018$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.352 INTC2GKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.
Write A5A5A501$_H$ to this register to enable writing protected registers.

44.5.6.10 INTC2GMPIDm — SPID Setting Register

INTC2GMPIDm are used to specify a set of SPIDs, which are referred by INTC2GPROT_n/GR/IMR registers. Access protection can be controlled by combination of INTC2GMPIDm registers and INTC2GPROT_n/GR/IMR register. For example, masters with SPID=1 or SPID=7 can access to global registers when setting INTC2GPROT_GR.MPID=00000101_B, INTC2GMPID0.SPID=01_H and INTC2GMPID2.SPID=07_H.

Access: This register can be read or written in 32-bit or 8-bit units.

Address: FFC6 4040_H + m * 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 44.353 INTC2GMPIDm Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

44.5.6.11 INTC2GPROT_n/GR/IMR — Channel/INTC2/INTC2 IMR Protection Control Register

Specifies the INTC2 slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Access: This register can be read or written in 32-bit units.

Address: INTC2GPROT_n: FFC6 4100_H + n * 04_H
INTC2GPROT_GR: FFC6 $40F0_H$
INTC2GPROT_IMR: FFC6 $40F4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.354 INTC2GPROT_n/GR/IMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101_B enables access to areas MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = x
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

44.5.7 DTS Guard

44.5.7.1 Overview

DTS guard is one of the slave guards used to control the access to the registers for DTS transfer. DTS guard can prevent read and write access to the DTS registers of each channel. If DTS guard detects illegal access, guard error notification is signaled to ECM.

44.5.7.2 List of Registers

Table 44.355 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_DTS (P-Bus Group0)	Guard error overflow clear register	DTSGOVFCLR	FFC6 8000_H	8, 32	PBG00#0	—
	Guard error overflow status register	DTSGOVFSTAT	FFC6 8004_H	8, 32	PBG00#0	—
	Guard error address register	DTSGERRADDR	FFC6 8008_H	32	PBG00#0	—
	Guard error access information register	DTSGERRTYPE	FFC6 $800C_H$	16, 32	PBG00#0	—
	Guard SPID error clear register	DTSGSPIDERRCLR	FFC6 8010_H	32	PBG00#0	—
	Guard SPID error status register	DTSGSPIDERRSTAT	FFC6 8014_H	32	PBG00#0	—
	Key code protection register	DTSGKCPROT	FFC6 8018_H	32	PBG00#0	—
	SPID setting register	DTSGMPIDm (m = 0-7)	FFC6 8040_H + m × 04_H	8, 32	PBG00#0	DTSGKCPROT
	DTS protection control register	DTSGPROT_GR	FFC6 $80F0_H$	32	PBG00#0	DTSGKCPROT
	Channel protection control register	DTSGPROT_nnn (nnn = 000-127)	FFC6 8100_H + nnn × 04_H	32	PBG00#0	DTSGKCPROT

Table 44.356 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
DTS registers (1)	DTSGPROT_GR	GUARD_DTS	Access request from all masters
DTS registers (2)	DTSGPROT_nnn (nnn = 000-127)		

Table 44.357 Detail target register of protection target

Protection target	Detail target register
DTS registers (1)	DTSCTL1, DTSCTL2, DTSSTS, DTSER, DTSPR0, DTSPR1, DTSPR2, DTSPR3, DTSPR4, DTSPR5, DTSPR6, DTSPR7, DTSnnnCM (nnn = 000 to 127)
DTS registers (2)	DTSAnnn, DTDAnnn, DTTCnnn, DTTCTnnn, DTRSAnnn, DTRDAnnn, DTRTCnnn, DTTCCnnn, DTFSLnnn, DTFSTnnn, DTFSSnnn, DTFSCnnn (nnn = 000 to 127)

44.5.7.3 DTSGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear DTSGOVFSTAT register. DTSGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit or 8-bit units.

Address: FFC6 8000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.358 DTSGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears DTSGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.7.4 DTSGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is reported again under the condition any bit of DTSGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to the DTSGOVFCLR register. See **Section 44.5.7.3, DTSGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: FFC6 8004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.359 DTSGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.7.5 DTSGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the DTSGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 8008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.360 DTSGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.7.6 DTSGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the DTSGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: FFC6 800C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.361 DTSGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.7.7 DTSGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the DTSGSPIDERRSTAT register. The DTSGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC6 8010_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.362 DTSGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear DTSGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.7.8 DTSGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the DTSGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC6 8014_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.363 DTSGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.7.9 DTSGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/DTS protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: FFC6 8018$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.364 DTSGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write A5A5A500$_H$ to this register to disable writing protected registers.
Write A5A5A501$_H$ to this register to enable writing protected registers.

44.5.7.10 DTSGMPIDm — SPID Setting Register

DTSGMPIDm are used to specify a set of SPIDs, which are referred by DTSGPROT_nnn/GR registers. Access protection can be controlled by combination of DTSGMPIDm registers and DTSGPROT_nnn/GR register. For example, masters with SPID=1 or SPID=7 can access to global registers when setting DTSGPROT_GR.MPID=00000101_B, DTSGMPID0.SPID=01_H and DTSGMPID2.SPID=07_H.

Access: This register can be read or written in 32-bit or 8-bit units.

Address: FFC6 8040_H + m * 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 44.365 DTSGMPIDm Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

44.5.7.11 DTSGPROT_nnn/GR — Channel/DTS Protection Control Register

Specifies the DTS slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Access: This register can be read or written in 32-bit units.

Address: DTSGPROT_nnn: FFC6 8100_H + nnn * 04_H
DTSGPROT_GR: FFC6 $80F0_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.366 DTSGPROT_nnn/GR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101_B enables access to areas with MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = x
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

NOTE

Use DTS channels except channel 4 and channel 68 in case that it is necessary to handle the setting of the DTSGPROT_nnn register after a DTS channel nnn operation starts.

44.5.8 sDMAC Guard

44.5.8.1 Overview

sDMAC guard is one of the slave guards used to control the access to the registers for sDMAC transfer. sDMAC guard can prevent read and write access to the sDMAC registers of each channel. If sDMAC guard detects illegal access, guard error notification is signaled to ECM.

44.5.8.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.367 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_DMAC0_base>	FFC6 9000_H	P-Bus Group 0
<GUARD_DMAC1_base>	FFC6 9400_H	P-Bus Group 0

44.5.8.3 List of Registers

Table 44.368 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_DMACj (j = 0, 1)	Guard error overflow clear register	DMAGOVFCLR	<GUARD_DMACj_base> + 00_H	8, 16, 32	PBG00 #0	—
	Guard error overflow status register	DMAGOVFSTAT	<GUARD_DMACj_base> + 04_H	8, 32	PBG00 #0	—
	Guard error address register	DMAGERRADDR	<GUARD_DMACj_base> + 08_H	32	PBG00 #0	—
	Guard error access information register	DMAGERRTYPE	<GUARD_DMACj_base> + $0C_H$	16, 32	PBG00 #0	—
	Guard SPID error clear register	DMAGSPIDERRCLR	<GUARD_DMACj_base> + 10_H	32	PBG00 #0	—
	Guard SPID error status register	DMAGSPIDERRSTAT	<GUARD_DMACj_base> + 14_H	32	PBG00 #0	—
	Key code protection register	DMAGKCPROT	<GUARD_DMACj_base> + 18_H	32	PBG00 #0	—
	SPID setting register	DMAGMPIDm (m = 0-7)	<GUARD_DMACj_base> + 40_H + m × 04_H	8, 16, 32	PBG00 #0	DMAGK CPROT
	DMA protection control register	DMAGPROT_GR	<GUARD_DMACj_base> + $F0_H$	8, 16, 32	PBG00 #0	DMAGK CPROT
	DMA descriptor protection control register	DMAGPROT_DP	<GUARD_DMACj_base> + $F4_H$	8, 16, 32	PBG00 #0	DMAGK CPROT
	Channel protection control register	DMAGPROT_n (n = 000-015)	<GUARD_DMACj_base> + 100_H + n × 04_H	8, 16, 32	PBG00 #0	DMAGK CPROT

Table 44.369 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
sDMAC0 Global registers	DMAGPROT_GR (GUARD_DMAC0)	GUARD_DMAC0	Access request from all masters
sDMAC0 Descriptor RAM	DMAGPROT_DP (GUARD_DMAC0)		
sDMAC0 Channel registers	DMAGPROT_n (n = 000 to 015) (GUARD_DMAC0)		
sDMAC1 Global registers	DMAGPROT_GR (GUARD_DMAC1)	GUARD_DMAC1	
sDMAC1 Descriptor RAM	DMAGPROT_DP (GUARD_DMAC1)		
sDMAC1 Channel registers	DMAGPROT_n (n = 000 to 015) (GUARD_DMAC1)		

44.5.8.4 DMAGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear DMAGOVFSTAT register. The DMAGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_DMACj_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.370 DMAGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears DMAGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.8.5 DMAGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is reported again under the condition any bit of DMAGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to DMAGOVFCLR register. See **Section 44.5.8.4, DMAGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <GUARD_DMACj_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.371 DMAGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.8.6 DMAGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the DMAGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_DMACj_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.372 DMAGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.8.7 DMAGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the DMAGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <GUARD_DMACj_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.373 DMAGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.8.8 DMAGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the DMAGSPIDERRSTAT register. The DMAGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <GUARD_DMACj_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.374 DMAGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear DMAGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.8.9 DMAGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the DMAGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_DMACj_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.375 DMAGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.8.10 DMAGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/DMA/DMA descriptor protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_DMACj_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.376 DMAGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.8.11 DMAGMPIDm — Guard SPID Setting Register

DMAGMPIDm are used to specify a set of SPIDs, which are referred by DMAGPROT_n/GR/DP registers. Access protection can be controlled by combination of DMAGMPIDm registers and DMAGPROT_n/GR/DP register. For example, masters with SPID = 1 or SPID = 7 can access to global registers when setting DMAGPROT_GR.MPID = 00000101_B, DMAGMPID0.SPID = 01_H and DMAGMPID2.SPID = 07_H.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_DMACj_base> + 40_H + m * 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 44.377 DMAGMPIDm Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	SPID[4:0]	SPID setting

44.5.8.12 DMAGPROT_n/GR/DP — Channel/DMA/DMA Descriptor Protection Control Register

Specifies the sDMAC slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: DMAGPROT_n: <GUARD_DMACj_base> + 100_H + n * 04_H
DMAGPROT_GR: <GUARD_DMACj_base> + $F0_H$
DMAGPROT_DP: <GUARD_DMACj_base> + $F4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	MPID[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.378 DMAGPROT_n/GR/DP Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	MPID[7:0]	R/W enable setting based on MPID MPID is a list of bits each representing one MPID value. Multiple MPID values are enabled simultaneously by setting multiple bits. For example, setting MPID to 0101_B enables access to areas with MPID = 0 and MPID = 2. 0: Reading/writing the area with MPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with MPID = x
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, MPID is used as a judgment condition. 1: During write, MPID is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, MPID is used as the judgment condition. 1: During read, MPID is not used as a judgment condition.

44.5.9 IBG

44.5.9.1 Overview

IPIR, BarrierSync and TPTM connect to Inter-cluster Bus (I-Bus). I-Bus Guard (IBG) is one of the slave guards used to control the access to these I-Bus modules. IBG can prevent read and write access to the registers of each module. If IBG detects illegal access, guard error notification is signaled to ECM.

44.5.9.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.379 Register Base Addresses

Base Address Name	Base Address	Bus Group
<GUARD_IPIR_base>	FFC6 A200$_H$	P-Bus Group 0
<GUARD_BARR_base>	FFC6 A000$_H$	P-Bus Group 0
<GUARD_TPTM_base>	FFC6 A600$_H$	P-Bus Group 0

44.5.9.3 List of Registers

Table 44.380 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_IPIR	Guard error overflow clear register	IPIGOVFCLR	<GUARD_IPIR_base> + 00$_H$	8, 32	PBG00#0	—
	Guard error overflow status register	IPIGOVFSTAT	<GUARD_IPIR_base> + 04$_H$	8, 32	PBG00#0	—
	Guard error address register	IPIGERRADDR	<GUARD_IPIR_base> + 08$_H$	32	PBG00#0	—
	Guard error access information register	IPIGERRTYPE	<GUARD_IPIR_base> + 0C$_H$	16, 32	PBG00#0	—
	Guard SPID error clear register	IPIGSPIDERRCLR	<GUARD_IPIR_base> + 10$_H$	32	PBG00#0	—
	Guard SPID error status register	IPIGSPIDERRSTAT	<GUARD_IPIR_base> + 14$_H$	32	PBG00#0	—
	Key code protection register	IPIGKCPROT	<GUARD_IPIR_base> + 18$_H$	32	PBG00#0	—
	Channel protection control register (Receive side)	IPIGPROT0_Rn (n = 0-3)	<GUARD_IPIR_base> + 80$_H$ + n × 10$_H$	8, 16, 32	PBG00#0	IPIGKCPROT
	Channel SPID setting register (Receive side)	IPIGPROT1_Rn (n = 0-3)	<GUARD_IPIR_base> + 84$_H$ + n × 10$_H$	8, 16, 32	PBG00#0	IPIGKCPROT
	Channel protection control register (Transmit side)	IPIGPROT0_Tn (n = 0-3)	<GUARD_IPIR_base> + 88$_H$ + n × 10$_H$	8, 16, 32	PBG00#0	IPIGKCPROT
	Channel SPID setting register (Transmit side)	IPIGPROT1_Tn (n = 0-3)	<GUARD_IPIR_base> + 8C$_H$ + n × 10$_H$	8, 16, 32	PBG00#0	IPIGKCPROT
	Common protection control register	IPIGPROT0_4	<GUARD_IPIR_base> + C0$_H$	8, 16, 32	PBG00#0	IPIGKCPROT
	Common SPID setting register	IPIGPROT1_4	<GUARD_IPIR_base> + C4$_H$	8, 16, 32	PBG00#0	IPIGKCPROT

Table 44.380 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GUARD_BARR	Guard error overflow clear register	BRGOVFCLR	<GUARD_BARR_base> + 00_H	8, 32	PBG00#0	—
	Guard error overflow status register	BRGOVFSTAT	<GUARD_BARR_base> + 04_H	8, 32	PBG00#0	—
	Guard error address register	BRGERRADDR	<GUARD_BARR_base> + 08_H	32	PBG00#0	—
	Guard error access information register	BRGERRTYPE	<GUARD_BARR_base> + $0C_H$	16, 32	PBG00#0	—
	Guard SPID error clear register	BRGSPIDERRCLR	<GUARD_BARR_base> + 10_H	32	PBG00#0	—
	Guard SPID error status register	BRGSPIDERRSTAT	<GUARD_BARR_base> + 14_H	32	PBG00#0	—
	Key code protection register	BRGKCPROT	<GUARD_BARR_base> + 18_H	32	PBG00#0	—
	Channel protection control register	BRGPROT0_n (n = 0-15)	<GUARD_BARR_base> + 80_H + n × 08_H	8, 16, 32	PBG00#0	BRGKCPROT
	Channel SPID setting register	BRGPROT1_n (n = 0-15)	<GUARD_BARR_base> + 84_H + n × 08_H	8, 16, 32	PBG00#0	BRGKCPROT
	Common protection control register	BRGPROT0_16	<GUARD_BARR_base> + 100_H	8, 16, 32	PBG00#0	BRGKCPROT
	Common SPID setting register	BRGPROT1_16	<GUARD_BARR_base> + 104_H	8, 16, 32	PBG00#0	BRGKCPROT
GUARD_TPTM	Guard error overflow clear register	TPTGOVFCLR	<GUARD_TPTM_base> + 00_H	8, 32	PBG00#0	—
	Guard error overflow status register	TPTGOVFSTAT	<GUARD_TPTM_base> + 04_H	8, 32	PBG00#0	—
	Guard error address register	TPTGERRADDR	<GUARD_TPTM_base> + 08_H	32	PBG00#0	—
	Guard error access information register	TPTGERRTYPE	<GUARD_TPTM_base> + $0C_H$	16, 32	PBG00#0	—
	Guard SPID error clear register	TPTGSPIDERRCLR	<GUARD_TPTM_base> + 10_H	32	PBG00#0	—
	Guard SPID error status register	TPTGSPIDERRSTAT	<GUARD_TPTM_base> + 14_H	32	PBG00#0	—
	Key code protection register	TPTGKCPROT	<GUARD_TPTM_base> + 18_H	32	PBG00#0	—
	Channel protection control register	TPTGPROT0_n (n = 0 to 3, 8 to 9)*1	<GUARD_TPTM_base> + 80_H + n × 08_H	8, 16, 32	PBG00#0	TPTGKCPROT
	Channel SPID setting register	TPTGPROT1_n (n = 0 to 3, 8 to 9)*1	<GUARD_TPTM_base> + 84_H + n × 08_H	8, 16, 32	PBG00#0	TPTGKCPROT

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins) and RH850/U2A6 with index n = 2, 3.

Table 44.381 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
I-Bus Group0 registers (1) (IPIR)	IPIGPROT0_Rn (n = 0-3) (GUARD_IPIR)	GUARD_IPIR	Access request from all masters
I-Bus Group0 registers (2) (IPIR)	IPIGPROT0_Tn (n = 0-3) (GUARD_IPIR)		
I-Bus Group0 registers (3) (IPIR)	IPIGPROT0_4 (GUARD_IPIR)		
I-Bus Group1 registers (1) (BARR)	BRGPROT0_n (n = 0-15) (GUARD_BARR)	GUARD_BARR	
I-Bus Group1 registers (2) (BARR)	BRGPROT0_16 (GUARD_BARR)		
I-Bus Group2 registers (1) (TPTM)	TPTGPROT0_n (n = 0 to 3)*1 (GUARD_TPTM)	GUARD_TPTM	
I-Bus Group2 registers (2) (TPTM)	TPTGPROT0_8 (GUARD_TPTM)		
I-Bus Group2 registers (3) (TPTM)	TPTGPROT0_9 (GUARD_TPTM)		

Note 1. This function is not implemented in RH850/U2A8 (373/292 pins) and RH850/U2A6 with index n = 2, 3.

Table 44.382 Detail target register of protection target

Protection target	Detail target register
I-Bus Group0 registers (1) (IPIR)	IPInENS, IPInFLGS, IPInFCLRS[*1]
I-Bus Group0 registers (2) (IPIR)	IPInREQS, IPInRCLRS[*1]
I-Bus Group0 registers (3) (IPIR)	IPInENm, IPInFLGm, IPInFCLRm, IPInREQm, IPInRCLRm[*1]
I-Bus Group1 registers (1) (BARR)	BRnINIT, BRnEN, BRnCHKS, BRnSYNCS[*2]
I-Bus Group1 registers (2) (BARR)	BRnCHKm, BRnSYNCm[*2]
I-Bus Group2 registers (1) (TPTM)	TPTMnIRUN, TPTMnIRRUN, TPTMnISTP, TPTMnISTR, TPTMnIIEN, TPTMnIUSTR, TPTMnIDIV, TPTMnFRUN, TPTMnFRRUN, TPTMnFSTP, TPTMnFSTR, TPTMnFDIV, TPTMnICNT0, TPTMnILD0, TPTMnICNT1, TPTMnILD1, TPTMnFCNT, TPTMnURUN, TPTMnURRUN, TPTMnUSTP, TPTMnUSTR, TPTMnUIEN, TPTMnUDIV, TPTMmUTRG, TPTMnUCNT0, TPTMnUCNT1, TPTMnUCMP00, TPTMnUCMP01, TPTMnUCMP02, TPTMnUCMP03, TPTMnUCMP10, TPTMnUCMP11, TPTMnUCMP12, TPTMnUCMP13[*3 *4]
I-Bus Group2 registers (2) (TPTM)	TPTMG0URUN,TPTMG0URRUN, TPTMG0USTP
I-Bus Group2 registers (3) (TPTM)	TPTMG1URUN,TPTMG1URRUN, TPTMG1USTP

Note 1. n = 0 to 3: n is the IPIR channel number. m = 0 to 3: m is the CPU number. m = 2, 3 is not implemented in RH850/U2A8 (373/292 pins) and RH850/U2A6.

Note 2. n = 0 to 15: n is the Barrier-Synchronization channel number. m = 0 to 3: m is the CPU number. m = 2, 3 is not implemented in RH850/U2A8 (373/292 pins) and RH850/U2A6.

Note 3. n = 0 to 3: n is the CPU number. n = 2, 3 is not implemented in RH850/U2A8 (373/292 pins) and RH850/U2A6.

Note 4. TPTM Self region registers are protected according to TPTGPROT0_n setting.

44.5.9.4 IPIGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear IPIGOVFSTAT register. IPIGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit or 8-bit units.

Address: <GUARD_IPIR_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.383 IPIGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears IPIGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.9.5 IPIGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error overflow is reported again under the condition any bit of IPIGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to IPIGOVFCLR register. See **Section 44.5.9.4, IPIGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <GUARD_IPIR_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.384 IPIGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.9.6 IPIGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the IPIGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_IPIR_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.385 IPIGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.9.7 IPIGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the IPIGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <GUARD_IPIR_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.386 IPIGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.9.8 IPIGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the IPIGSPIDERRSTAT register. The IPIGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <GUARD_IPIR_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.387 IPIGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear IPIGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.9.9 IPIGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the IPIGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_IPIR_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.388 IPIGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.9.10 IPIGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/common protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_IPIR_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.389 IPIGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.9.11 IPIGPROT0_Rn/Tn, IPIGPROT0_4 — Channel, Common Protection Control Register

Specifies the IPIR slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: IPIGPROT0_Rn: <GUARD_IPIR_base> + 80_H + n * 10_H
IPIGPROT0_Tn: <GUARD_IPIR_base> + 88_H + n * 10_H
IPIGPROT0_4: <GUARD_IPIR_base> + $C0_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.390 IPIGPROT0_Rn/Tn, IPIGPROT0_4 Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, IPIGPROT1_Rn/Tn, IPIGPROT1_4 is used as a judgment condition. 1: During write, IPIGPROT1_Rn/Tn, IPIGPROT1_4 is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, IPIGPROT1_Rn/Tn, IPIGPROT1_4 is used as the judgment condition. 1: During read, IPIGPROT1_Rn/Tn, IPIGPROT1_4 is not used as a judgment condition.

44.5.9.12 IPIGPROT1_Rn/Tn, IPIGPROT1_4 — Channel, Common SPID Setting Register

Specifies the IPIR slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: IPIGPROT1_Rn: <GUARD_IPIR_base> + 84_H + n * 10_H
IPIGPROT1_Tn: <GUARD_IPIR_base> + $8C_H$ + n * 10_H
IPIGPROT1_4: <GUARD_IPIR_base> + $C4_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.391 IPIGPROT1_Rn/Tn, IPIGPROT1_4 Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID IPIGPROT1_Rn/Tn/4 is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting IPIGPROT1_Rn/Tn/4 to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = m

44.5.9.13 BRGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear BRGOVFSTAT register. BRGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit or 8-bit units.

Address: <GUARD_BARR_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.392 BRGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears BRGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.9.14 BRGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is reported again under the condition any bit of when SPIDERR bit of BRGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to BRGOVFCLR register. See **Section 44.5.9.13, BRGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <GUARD_BARR_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.393 BRGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.9.15 BRGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the BRGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_BARR_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.394 BRGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.9.16 BRGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the BRGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <GUARD_BARR_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.395 BRGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.9.17 BRGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the BRGSPIDERRSTAT register. The BRGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <GUARD_BARR_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.396 BRGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear BRGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.9.18 BRGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the BRGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_BARR_base> + 14_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.397 BRGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.9.19 BRGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel/common protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_BARR_base> + 18_H

Value after reset: $0000\ 0000_H$

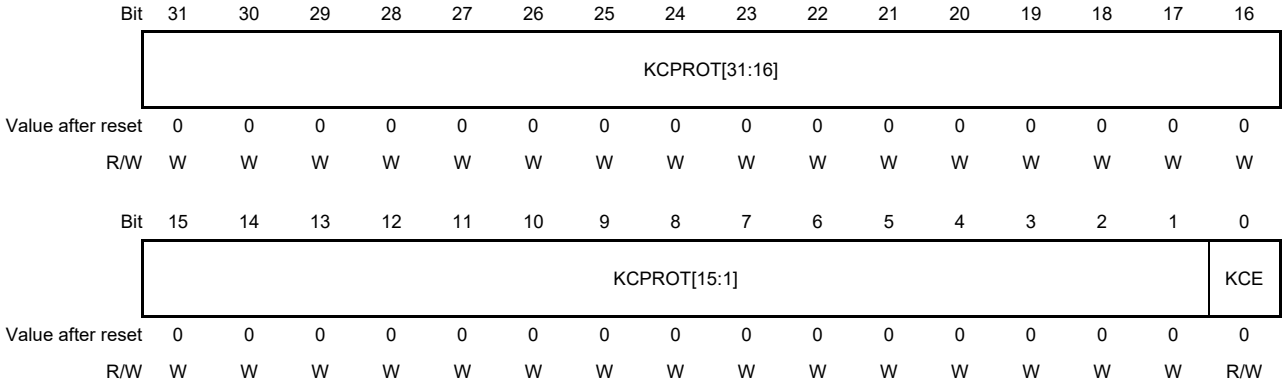


Table 44.398 BRGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.9.20 BRGPROT0_n/16 — Channel/Common Protection Control Register

Specifies the BarrierSync slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: BRGPROT0_n: <GUARD_BARR_base> + 80_H + n * 08_H
BRGPROT0_16: <GUARD_BARR_base> + 100_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.399 BRGPROT0_n Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, BRGPROT1_n/16 is used as a judgment condition. 1: During write, BRGPROT1_n/16 is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, BRGPROT1_n/16 is used as the judgment condition. 1: During read, BRGPROT1_n/16 is not used as a judgment condition.

44.5.9.21 BRGPROT1_n/16 — Channel/Common SPID Setting Register

Specifies the BarrierSync slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: BRGPROT1_n: <GUARD_BARR_base> + 84_H + n * 08_H
BRGPROT1_16: <GUARD_BARR_base> + 104_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.400 BRGPROT1_n Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID BRGPROT1_n/16 is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting BRGPROT1_n/16 to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = m

44.5.9.22 TPTGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear TPTGOVFSTAT register. TPTGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit or 8-bit units.

Address: <GUARD_TPTM_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.401 TPTGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	CLRO	Clears TPTGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.9.23 TPTGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is notified again under condition any bit of TPTGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to TPTGOVFCLR register. See **Section 44.5.9.22, TPTGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <GUARD_TPTM_base> + 04_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.402 TPTGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.9.24 TPTGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the TPTGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_TPTM_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.403 TPTGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.9.25 TPTGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the TPTGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <GUARD_TPTM_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.404 TPTGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.9.26 TPTGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the TPTGSPIDERRSTAT register. The TPTGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <GUARD_TPTM_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.405 TPTGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear TPTGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.9.27 TPTGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the TPTGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <GUARD_TPTM_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.406 TPTGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.9.28 TPTGKCPROT — Key Code Protection Register

This register is used for protection against writing to the channel protection control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <GUARD_TPTM_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.407 TPTGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.9.29 TPTGPROT0_n — Channel Protection Control Register

Specifies the TPTM slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_TPTM_base> + 80_H + n * 08_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.408 TPTGPROT0_n Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, TPTGPROT1_n is used as a judgment condition. 1: During write, TPTGPROT1_n is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, TPTGPROT1_n is used as the judgment condition. 1: During read, TPTGPROT1_n is not used as a judgment condition.

44.5.9.30 TPTGPROT1_n — Channel SPID Setting Register

Specifies the TPTM slave access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID settings.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <GUARD_TPTM_base> + 84_H + n * 08_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.409 TPTGPROT1_n Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID TPTGPROT1_n is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting TPTGPROT1_n to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = m

44.5.10 PBG

44.5.10.1 Overview

The PBG module is divided into several PBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG channel can designate the access against which a single peripheral circuit should be protected. Each PBG group can hold the information of the access that has been rejected.

The following table lists the peripheral circuits to be protected, the corresponding PBG group names, and the PBG channel numbers. For details of PORT categories for PBG see **Table 44.411**.

Note: Even if protected module or channel is not available by product variants, guard registers for that module are available (except PBG01#2,3).
For details of protection target modules by product variants, refer to corresponding each section.

Table 44.410 List of Peripheral Circuit Modules to be Protected (1/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG00	0	P-Bus guard control and error register (PBG00, PBG01, PBGERRSLV00) INTC2 guard register, DTS guard register, sDMAC guard register, I-Bus guard register
	1	GCFU, CFU
	2	BOOTCTRL register, DTS_COMP_CNTRL register, ERRGEN, ECC/Address feedback/Address parity controller without suffix PE0, PE1, PE2, PE3 and Cluster0, Cluster1, Cluster2 and Cluster3
	3	ECC/Address feedback/Address parity controller for Cluster0 Cluster RAM guard for Cluster0 without suffix PE0, PE1, PE2 and PE3 CCIBH0
	4	ECC/Address feedback/Address parity controller for Cluster1 Cluster RAM guard for Cluster1 without suffix PE0, PE1, PE2 and PE3
	5	ECC/Address feedback/Address parity controller for Cluster2 Cluster RAM guard for Cluster2 without suffix PE0, PE1, PE2 and PE3
	6	ECC/Address feedback/Address parity controller for Cluster3 Cluster RAM guard for Cluster3 without suffix PE0, PE1, PE2 and PE3
PBG01	0	ECC/Address feedback controller for PE0 PEG register for PE0 CRG register with suffix PE0
	1	ECC/Address feedback controller for PE1 PEG register for PE1 CRG register with suffix PE1
	2*4	ECC/Address feedback controller for PE2 PEG register for PE2 CRG register with suffix PE2
	3*4	ECC/Address feedback controller for PE3 PEG register for PE3 CRG register with suffix PE3
PBG10	0	P-Bus guard control and error register (PBG10, PBGERRSLV10)
	1	ECC on Bus for P-Bus Group 1
	2	ECC controller for Data Flash

Table 44.410 List of Peripheral Circuit Modules to be Protected (2/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG20	0	P-Bus guard control and error register (PBG20, PBG21, PBG22, PBGERRSLV20)
	1	ECC on Bus for P-Bus Group 2H/2L
	2	System Control – Category 1 Registers of the following modules: • Clock Controller • Power Supply Voltage Monitor And registers of the following modules that are NOT related to ICUMHA: • Reset Controller (including BIST skip control registers) • Standby Controller • Clock Monitor • Operating Mode Assumption of Use: a group which is assigned to the safety core.
	3	System Control – Category 2 Registers SWSRESA_ICUM and SWARESA_ICUM*1. Assumption of Use: a group which is assigned to the ICUMHA or the safety core.
	4	BIST
	5	TAUJ2
	6	TAUJ3
	7	LPS0
	8	FENC, FEINC (for PE0, PE1, PE2, PE3)
	9	ADCJ2
	10	RTCA0
	11	WDTBA
	12	Port configuration register – Category A Assumption of Use: a group which is configured at start-up
	13	Port configuration register – Category B Assumption of Use: a group which is configured at start-up
	14	Port configuration register – Category C Assumption of Use: a group which is configured at start-up
	15	JPort configuration register – Category C Assumption of Use: a group which is configured at start-up

Table 44.410 List of Peripheral Circuit Modules to be Protected (3/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG21	0	Port configuration register – Category D Assumption of Use: a group which is configured at start-up
	1	JPort configuration register – Category D Assumption of Use: a group which is configured at start-up
	2	Port configuration register – Category E Assumption of Use: a group which is configured at start-up
	3	JPort configuration register – Category F Assumption of Use: a group which is configured at start-up
	4	Port configuration register for Port group 0 – Category F Assumption of Use: a group which is configured dynamically
	5	Port configuration register for Port group 1 – Category F Assumption of Use: a group which is configured dynamically
	6	Port configuration register for Port group 2 – Category F Assumption of Use: a group which is configured dynamically
	7	Port configuration register for Port group 3 – Category F Assumption of Use: a group which is configured dynamically
	8	Port configuration register for Port group 4 – Category F Assumption of Use: a group which is configured dynamically
	9	Port configuration register for Port group 5 – Category F Assumption of Use: a group which is configured dynamically
	10	Port configuration register for Port group 6 – Category F Assumption of Use: a group which is configured dynamically
	11	Port configuration register for Port group 8 – Category F Assumption of Use: a group which is configured dynamically
	12	Port configuration register for Port group 9 – Category F Assumption of Use: a group which is configured dynamically
	13	Port configuration register for Port group 10 – Category F Assumption of Use: a group which is configured dynamically
	14	Port configuration register for Port group 11 – Category F Assumption of Use: a group which is configured dynamically
	15	Port configuration register for Port group 12 – Category F Assumption of Use: a group which is configured dynamically

Table 44.410 List of Peripheral Circuit Modules to be Protected (4/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG22	0	Port configuration register for Port group 17 – Category F Assumption of Use: a group which is configured dynamically
	1	Port configuration register for Port group 18 – Category F Assumption of Use: a group which is configured dynamically
	2	Port configuration register for Port group 19 – Category F Assumption of Use: a group which is configured dynamically
	3	Port configuration register for Port group 20 – Category F Assumption of Use: a group which is configured dynamically
	4	Port configuration register for Port group 21 – Category F Assumption of Use: a group which is configured dynamically
	5	Port configuration register for Port group 22 – Category F Assumption of Use: a group which is configured dynamically
	6	Port configuration register for Port group 23 – Category F Assumption of Use: a group which is configured dynamically
	7	Port configuration register for Port group 24 – Category F Assumption of Use: a group which is configured dynamically
	8	Port configuration register for Port group AP0 – Category F Assumption of Use: a group which is configured dynamically
	9	Port configuration register for Port group AP1 – Category F Assumption of Use: a group which is configured dynamically
	10	Port configuration register for Port group AP2 – Category F Assumption of Use: a group which is configured dynamically
	11	Port configuration register for Port group AP3 – Category F Assumption of Use: a group which is configured dynamically
	12	Port configuration register for Port group AP4 – Category F Assumption of Use: a group which is configured dynamically
	13	Port configuration register for Port group AP5 – Category F Assumption of Use: a group which is configured dynamically
PBG30	0	P-Bus guard control and error register (PBG30, PBG31, PBG32, PBG33, PBGERRSLV30)
	1	ECC on Bus for P-Bus Group 3 ECC controller for RSCFD1 related RAM ECC controller for MMCA0 related RAM
	2	PSI50
	3	PSI51
	4	PSI52
	5	PSI53
	6	PSI5TSSEL register
	7	PSI5S0
	8	PSI5S1
	9	SCI30
	10	SCI31
	11	SCI32
	12	ECM
	13	KCRC0
	14	KCRC2
	15	KCRC4

Table 44.410 List of Peripheral Circuit Modules to be Protected (5/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG31	0	KCRC6
	1	CXPI0
	2	CXPI1
	3	CXPI2
	4	CXPI3
	5	RSENT0
	6	RSENT1
	7	RSENT2
	8	RSENT3
	9	RSENT4
	10	RSENT5
	11	RSENT6
	12	RSENT7
	13	RSENTTSSEL register
PBG32	0	MMCA0
	1	RLIN30
	2	RLIN32
	3	RLIN34
	4	RLIN36
	5	RLIN38
	6	RLIN310
	7	RLIN312
	8	RLIN314
	9	RLIN316
	10	RLIN318
	11	RLIN320
	12	RLIN322
	13	RSCFD1 CAN8
	14	RSCFD1 CAN9
	15	RSCFD1 CAN10
PBG33	0	RSCFD1 CAN11
	1	RSCFD1 CAN12
	2	RSCFD1 CAN13
	3	RSCFD1 CAN14
	4	RSCFD1 CAN15
	5	RSCFD1 common registers

Table 44.410 List of Peripheral Circuit Modules to be Protected (6/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG40	0	P-Bus guard control and error register (PBG40, PBGERRSLV40)
	1	ECC on Bus for P-Bus Group 4 ECC controller for MSPI0, MSPI2, MSPI4, MSPI6, MSPI8 related RAM
	2	DMA/DTSTRIGGEN (for MSPI)
	3	MSPI0
	4	MSPI2
	5	MSPI4
	6	MSPI6
	7	MSPI8
PBG50	0	P-Bus guard control and error register (PBG50, PBG51, PBG52, PBG53, PBGERRSLV50)
	1	ECC on Bus for P-Bus Group 5 ECC controller for MSPI1, MSPI3, MSPI5, MSPI7, MSPI9 related RAM
	2	WDTB0, SWDT0
	3	WDTB1
	4	WDTB2
	5	WDTB3
	6	LTSC0
	7	OSTM0
	8	OSTM1
	9	OSTM2
	10	OSTM3
	11	OSTM4
	12	OSTM5
	13	OSTM6
	14	OSTM7
	15	OSTM8
PBG51	0	OSTM9
	1	TAUD0
	2	TAUD1
	3	TAUD2 (including DNFA registers)
	4	TAUJ0
	5	TAUJ1
	6	TSG30
	7	TSG31
	8	ENCA0
	9	ENCA1
	10	TAPA0
	11	TAPA1
	12	TAPA2
	13	TAPA3
	14	TPBA0
	15	TPBA1

Table 44.410 List of Peripheral Circuit Modules to be Protected (7/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG52	0	PIC1
	1	PIC20, PIC21, PIC22
	2	KCRC1
	3	KCRC3
	4	KCRC5
	5	KCRC7
	6	MSPI1
	7	MSPI3
	8	MSPI5
	9	MSPI7
	10	MSPI9
	11	RLIN31
	12	RLIN33
	13	RLIN35
	14	RLIN37
	15	RLIN39
PBG53	0	RLIN311
	1	RLIN313
	2	RLIN315
	3	RLIN317
	4	RLIN319
	5	RLIN321
	6	RLIN323
PBG6L0	0	P-Bus guard control and error register (PBG6L0, PBGERRSLV6L0)
	1	Flash protection (IDCTRL)
	2	OTS0
	3	EINT
	4	ADCJ0, AVSEG
	5	PWM-diag (PWBA, PWGC, PWSD)
	6	FLMD
PBG6L1*2	0	P-Bus guard control and error register (PBG6L1, PBGERRSLV6L1)
	1	FACI0 Register Area for CPU(PE) FACI0 Command-Issuing Area for CPU(PE)
	2	FACI1 Register Area for CPU(PE) FACI1 Command-Issuing Area for CPU(PE)
	3*3	FACI2 Register Area for CPU(PE) FACI2 Command-Issuing Area for CPU(PE)
PBG6H0	0	P-Bus guard control and error register (PBG6H0, PBGERRSLV6H0)
	1	ECC on Bus for P-Bus Group 6H/6L ECC controller for GTM related RAM
	2	GTM

Table 44.410 List of Peripheral Circuit Modules to be Protected (8/8)

PBG Group	PBG Channel Number	Module to be Protected
PBG70	0	P-Bus guard control and error register (PBG70, PBGERRSLV70)
	1	ECC on Bus for P-Bus Group 7
	2	ADCJ1
	3	RIIC0
	4	RIIC1
PBG80	0	P-Bus guard control and error register (PBG80, PBGERRSLV80)
	1	ECC on Bus for P-Bus Group 8 ECC controller for RSCFD0 related RAM
	2	RSCFD0 CAN0
	3	RSCFD0 CAN1
	4	RSCFD0 CAN2
	5	RSCFD0 CAN3
	6	RSCFD0 CAN4
	7	RSCFD0 CAN5
	8	RSCFD0 CAN6
	9	RSCFD0 CAN7
	10	RSCFD0 common registers
PBG90	0	P-Bus guard control and error register (PBG90, PBGERRSLV90) H-Bus guard control and error register (HBG91, HBGERRSLV91, HBG92, HBGERRSLV92, HBG93, HBGERRSLV93, HBG96, HBGERRSLV96)
	1	ECC on Bus for P-Bus Group 9 ECC on Bus for H-Bus Master modules/Slave modules ECC controller for FLXA0/FLXA1 related RAM ECC controller for ETNB0/ETNB1 related RAM
	2	RHSIF0
	3	SPIDCTL
	4	INTIF, DMATRGSEL
	5	ICUM_CMDREG*[1]
	6	ETNB0 (Fast Ether)
	7	ETNB1 (Gb Ether)

Note 1. For more detail of these registers refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Note 2. When ICUMHA is enable, PBG error is notified to ICUMHA. When ICUMHA is disable, PBG error is notified to ECM.

Note 3. This protected module is available only when ICUMHA is disable.

Note 4. The register of this channel is not available in RH850/U2A6.

Table 44.411 Relation between categories and target registers for PORT configuration registers

Category		Target register
Category A		PKCPROT, PWE, LVDSCTRLA, LVDSCTRLB
Category B		FCLACTLm_<name>, DNFACTL_<name>, DNFAEN_<name>, DNFAENH_<name>, DNFAENL_<name>*1
Category C		PCRn_m, APCRn_m, JPCRn_m
Category D		PIBCn, PBDCn, PIPCn, PUn,PDn, PODCn, PDSCn, PISn, PISAn, PUCCn, PODCEn, APIBCn, APBDCn, APODCn, APDSCn, APODCEn, JPIBC0, JPBDC0, JPU0, JPD0, JPODC0, JPDSC0, JPIS0, JPISA0, JPUCC0, JPODCE0
Category E		PSFCn, PSFTSn, PSFTSEn, APSFCn, APSFTSn, APSFTSEn
Category F	For Port Group	Pn, PSRn, PNOTn, PPRn, PMn, PMCn, PFCn, PFCEn, PMSRn, PMCSRn, PFCAEn, PINVn
	For APort Group	APn, APSRn, APNOTn, APPRn, APMn, APMSRn, APINVn
	For JTAG Port	JP0, JPSR0, JPNOT0, JPPR0, JPM0, JPMC0, JPFC0, JPFCE0, JPMSR0, JPMCSR0, JPINV0

Note 1. Except DNFA registers for TAUD2

44.5.10.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.412 Register Base Addresses

Base Address Name	Base Address	Bus Group
<PBG00_base>	FFC6 B080$_H$	P-Bus Group 0
<PBG01_base>	FFC6 B100$_H$	P-Bus Group 0
<PBGERRSLV00_base>	FFC6 B000$_H$	P-Bus Group 0
<PBG10_base>	FFC6 3000$_H$	P-Bus Group 1
<PBGERRSLV10_base>	FFC6 3100$_H$	P-Bus Group 1
<PBG20_base>	FFDE 0B00$_H$	P-Bus Group 2L
<PBG21_base>	FFDE 0C00$_H$	P-Bus Group 2L
<PBG22_base>	FFDE 0D00$_H$	P-Bus Group 2L
<PBGERRSLV20_base>	FFDE 1000$_H$	P-Bus Group 2L
<PBG30_base>	FFC7 2B00$_H$	P-Bus Group 3
<PBG31_base>	FFC7 2C00$_H$	P-Bus Group 3
<PBG32_base>	FFC7 2D00$_H$	P-Bus Group 3
<PBG33_base>	FFC7 2E00$_H$	P-Bus Group 3
<PBGERRSLV30_base>	FFC7 3200$_H$	P-Bus Group 3
<PBG40_base>	FFC7 5300$_H$	P-Bus Group 4
<PBGERRSLV40_base>	FFC7 5400$_H$	P-Bus Group 4
<PBG50_base>	FFC7 A300$_H$	P-Bus Group 5
<PBG51_base>	FFC7 A400$_H$	P-Bus Group 5
<PBG52_base>	FFC7 A500$_H$	P-Bus Group 5
<PBG53_base>	FFC7 A600$_H$	P-Bus Group 5
<PBGERRSLV50_base>	FFC7 B000$_H$	P-Bus Group 5
<PBG6L0_base>	FFC8 1000$_H$	P-Bus Group 6L
<PBG6L1_base>	FFC8 1200$_H$	P-Bus Group 6L
<PBGERRSLV6L0_base>	FFC8 3000$_H$	P-Bus Group 6L
<PBGERRSLV6L1_base>	FFC8 3040$_H$	P-Bus Group 6L
<PBG6H0_base>	FF75 1000$_H$	P-Bus Group 6H
<PBGERRSLV6H0_base>	FF75 2000$_H$	P-Bus Group 6H
<PBG70_base>	FFF4 9400$_H$	P-Bus Group 7
<PBGERRSLV70_base>	FFF4 A000$_H$	P-Bus Group 7
<PBG80_base>	FFF2 9300$_H$	P-Bus Group 8
<PBGERRSLV80_base>	FFF2 A000$_H$	P-Bus Group 8
<PBG90_base>	FF0A 1300$_H$	P-Bus Group 9
<PBGERRSLV90_base>	FF0A 1400$_H$	P-Bus Group 9

44.5.10.3 List of Registers

Table 44.413 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
PBGERRSLVn (n = 00, 10, 20, 30, 40, 50, 6L0, 6L1, 6H0, 70, 80, 90)	Guard error overflow clear register	PBGOVFCLR	<PBGERRSLVn_base> + 00_H	8, 16, 32	*2	—
	Guard error overflow status register	PBGOVFSTAT	<PBGERRSLVn_base> + 04_H	8, 32	*2	—
	Guard error address register	PBGERRADDR	<PBGERRSLVn_base> + 08_H	32	*2	—
	Guard error access information register	PBGERRTYPE	<PBGERRSLVn_base> + $0C_H$	16, 32	*2	—
	Guard SPID error clear register	PBGSPIDERRCLR	<PBGERRSLVn_base> + 10_H	32	*2	—
	Guard SPID error status register	PBGSPIDERRSTAT	<PBGERRSLVn_base> + 14_H	32	*2	—
	Key code protection register	PBGKCPROT	<PBGERRSLVn_base> + 18_H	32	*2	—
PBGn (n = 00, 01, 10, 20, 21, 22, 30, 31, 32, 33, 40, 50, 51, 52, 53, 6L0, 6L1, 6H0, 70, 80, 90)	Channel protection control register	PBGPROT0_m (m = 0-15)*1	<PBGn_base> + 00_H + m × 08_H	8, 16, 32	*2	PBGKCPROT
	Channel SPID setting register	PBGPROT1_m (m = 0-15)*1	<PBGn_base> + 04_H + m × 08_H	8, 16, 32	*2	PBGKCPROT

Note 1. “m” of PBGPROT0_m and PBGPROT1_m indicates the number of PBG channel. Each channel number is different from each other then refer to **Section 44.5.10.1, Overview**.

Note 2.
n = 00 and 01: PBG00#0
n = 10: PBG10#0
n = 20, 21, and 22: PBG20#0
n = 30, 31, 32, and 33: PBG30#0
n = 40: PBG40#0
n = 50, 51, 52, and 53: PBG50#0
n = 6L0: PBG6L0#0
n = 6L1: PBG6L1#0
n = 6H0: PBG6H0#0
n = 70: PBG70#0
n = 80: PBG80#0
n = 90: PBG90#0

Table 44.414 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
P-Bus Group 0 registers	PBG00	PBGERRSLV00	Access request from all masters
	PBG01		
P-Bus Group 1 registers	PBG10	PBGERRSLV10	
P-Bus Group 2H/2L registers	PBG20	PBGERRSLV20	
	PBG21		
	PBG22		
P-Bus Group 3 registers	PBG30	PBGERRSLV30	
	PBG31		
	PBG32		
	PBG33		
P-Bus Group 4 registers	PBG40	PBGERRSLV40	
P-Bus Group 5 registers	PBG50	PBGERRSLV50	
	PBG51		
	PBG52		
	PBG53		
P-Bus Group 6L registers	PBG6L0	PBGERRSLV6L0	
	PBG6L1	PBGERRSLV6L1	
P-Bus Group 6H registers	PBG6H0	PBGERRSLV6H0	
P-Bus Group 7 registers	PBG70	PBGERRSLV70	
P-Bus Group 8 registers	PBG80	PBGERRSLV80	
P-Bus Group 9 registers	PBG90	PBGERRSLV90	

44.5.10.4 PBGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear PBGOVFSTAT register. PBGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <PBGERRSLVn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.415 PBGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset
1	CLRO	Clears PBGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.10.5 PBGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard error overflow was occurred. OVF bit is set when a guard error is notified again under the condition any bit of PBGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to PBGOVFCLR register. See **Section 44.5.10.4, PBGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit or 8-bit units.

Address: <PBGERRSLVn_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.416 PBGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.10.6 PBGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the PBGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PBGERRSLVn_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.417 PBGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.10.7 PBGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the PBGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <PBGERRSLVn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.418 PBGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.10.8 PBGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the PBGSPIDERRSTAT register. The PBGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <PBGERRSLVn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.419 PBGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear PBGSPIDERRSTAT.SPIDERR bit by writing 1.

44.5.10.9 PBGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the PBGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <PBGERRSLVn_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.420 PBGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.10.10 PBGKCPROT — Key Code Protection Register

This register is used for protection against writing to the P-Bus guard control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <PBGERRSLVn_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.421 PBGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.10.11 PBGPROT0_m — Channel Protection Control Register

Specifies the P-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

PBG01 PBGPROT0_2, PBGPROT0_3 are not available in RH850/U2A6.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PBGn_base> + 00_H + m * 08_H

Value after reset: 00000141_H
PBG6L1 PBGPROT0_1, PBGPRTOT0_2: $0x000001C1_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOCK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	SEC	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	1	*1	1	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W

Table 44.422 PBGPROT0_m Register Contents

Bit Position	Bit Name	Function
31	LOCK	Allows blocking any further change of PBGPROT0_m and PBGPROT1_m. 0: Register can be re-written 1: Any further write to PBGPROT0_m and PBGPROT1_m is ignored This bit can only be cleared by reset.
30 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting
7*2 *3	SEC	R/W disable setting for non-secure master 0: R/W disabled 1: R/W depends on other enable/disable settings
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, PBGPROT1_m is used as a judgment condition. 1: During write, PBGPROT1_m is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, PBGPROT1_m is used as the judgment condition. 1: During read, PBGPROT1_m is not used as a judgment condition.

Note 1. Initial value of this bit is 1 for only PBGPROT0_1 and PBGPROT0_2 for PBG6L1.

Note 2. When ICUMHA is enable, this bit is available only PBG6L1. For other PBG, this bit is not available regardless of ICUMHA enable/disable.

Note 3. When ICUMHA is disable, this bit for PBG6L1 is no function although it can be written and read.

44.5.10.12 PBGPROT1_m — Channel SPID Setting Register

Specifies the P-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID setting.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <PBGn_base> + 04_H + m * 08_H

Value after reset: See **Table 44.424**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	*1	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.423 PBGPROT1_m Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID PBGPROT1_m is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting PBGPROT1_m to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = x depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = x

Note 1. This depends on each channel. See **Table 44.424**.

Table 44.424 PBGPROT1_m Register Initial Values (1/7)

Module Name	PROT1_m Register	Initial value
PBG00	PBGPROT1_0	0400 000F_H
	PBGPROT1_1	0400 000F_H
	PBGPROT1_2	0400 000F_H
	PBGPROT1_3	0400 000F_H
	PBGPROT1_4	0400 000F_H
	PBGPROT1_5	0400 000F_H
	PBGPROT1_6	0400 000F_H
PBG01	PBGPROT1_0	0000 000F_H
	PBGPROT1_1	0000 000F_H
	PBGPROT1_2*1	0000 000F_H
	PBGPROT1_3*1	0000 000F_H
PBG10	PBGPROT1_0	0400 000F_H
	PBGPROT1_1	0000 000F_H
	PBGPROT1_2	0000 000F_H

Table 44.424 PBGPROT1_m Register Initial Values (2/7)

Module Name	PROT1_m Register	Initial value
PBG20	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0400 000F$_H$
	PBGPROT1_3	0400 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$
PBG21	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$

Table 44.424 PBGPROT1_m Register Initial Values (3/7)

Module Name	PROT1_m Register	Initial value
PBG22	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
PBG30	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$
PBG31	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$

Table 44.424 PBGPROT1_m Register Initial Values (4/7)

Module Name	PROT1_m Register	Initial value
PBG32	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$
PBG33	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
PBG40	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$

Table 44.424 PBGPROT1_m Register Initial Values (5/7)

Module Name	PROT1_m Register	Initial value
PBG50	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$
PBG51	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$

Table 44.424 PBGPROT1_m Register Initial Values (6/7)

Module Name	PROT1_m Register	Initial value
PBG52	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
	PBGPROT1_11	0000 000F$_H$
	PBGPROT1_12	0000 000F$_H$
	PBGPROT1_13	0000 000F$_H$
	PBGPROT1_14	0000 000F$_H$
	PBGPROT1_15	0000 000F$_H$
PBG53	PBGPROT1_0	0000 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
PBG6L0	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0400 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
PBG6L1	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0400 000F$_H$
	PBGPROT1_2	0400 000F$_H$
	PBGPROT1_3	0400 000F$_H$
PBG6H0	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
PBG70	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$

Table 44.424 PBGPROT1_m Register Initial Values (7/7)

Module Name	PROT1_m Register	Initial value
PBG80	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0000 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0000 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$
	PBGPROT1_8	0000 000F$_H$
	PBGPROT1_9	0000 000F$_H$
	PBGPROT1_10	0000 000F$_H$
PBG90	PBGPROT1_0	0400 000F$_H$
	PBGPROT1_1	0000 000F$_H$
	PBGPROT1_2	0000 000F$_H$
	PBGPROT1_3	0400 000F$_H$
	PBGPROT1_4	0000 000F$_H$
	PBGPROT1_5	0400 000F$_H$
	PBGPROT1_6	0000 000F$_H$
	PBGPROT1_7	0000 000F$_H$

Note 1. The register of this channel is not available in RH850/U2A6.

44.5.11 HBG

44.5.11.1 Overview

HBG can prevent read and write access against which a peripheral circuit on H-Bus should be protected. If HBG detects illegal access, guard error notification is signaled to ECM.

The following table lists the peripheral circuits to be protected and the corresponding HBG names.

Table 44.425 List of Peripheral Circuit Module to be Protected

HBG	Module to be Protected
HBG91	RHSIF0*1
HBG92	FLXA0
HBG93	FLXA1*1
HBG96	SFMA0*2

Note 1. This function is not implemented in RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A6 (156/144 pins).

44.5.11.2 Register Base Address

Base addresses of safety modules are listed in the following table. Each register address is given as offsets from the base addresses.

Table 44.426 Register Base Addresses

Base Address Name	Base Address	Bus Group
<HBG91_base>*1	FF0D 0600$_H$	P-Bus Group 9
<HBGERRSLV91_base>*1	FF0D 1600$_H$	P-Bus Group 9
<HBG92_base>	FF0D 0000$_H$	P-Bus Group 9
<HBGERRSLV92_base>	FF0D 1000$_H$	P-Bus Group 9
<HBG93_base>*1	FF0D 0200$_H$	P-Bus Group 9
<HBGERRSLV93_base>*1	FF0D 1200$_H$	P-Bus Group 9
<HBG96_base>*2	FF0D 0400$_H$	P-Bus Group 9
<HBGERRSLV96_base>*2	FF0D 1400$_H$	P-Bus Group 9

Note 1. This function is not implemented in RH850/U2A6.
Note 2. This function is not implemented in RH850/U2A6 (156/144 pins).

44.5.11.3 List of Registers

Table 44.427 List of Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
HBGERRSLVn (n = 91, 92, 93, 96)*1	Guard error overflow clear register	HBGOVFCLR	<HBGERRSLVn_base> + 00_H	8, 16, 32	PBG90#0	—
	Guard error overflow status register	HBGOVFSTAT	<HBGERRSLVn_base> + 04_H	8, 16, 32	PBG90#0	—
	Guard error address register	HBGERRADDR	<HBGERRSLVn_base> + 08_H	32	PBG90#0	—
	Guard error access information register	HBGERRTYPE	<HBGERRSLVn_base> + $0C_H$	16, 32	PBG90#0	—
	Guard SPID error clear register	HBGSPIDERRCLR	<HBGERRSLVn_base> + 10_H	32	PBG90#0	—
	Guard SPID error status register	HBGSPIDERRSTAT	<HBGERRSLVn_base> + 14_H	32	PBG90#0	—
	Key code protection register	HBGKCPROT	<HBGERRSLVn_base> + 18_H	32	PBG90#0	—
HBGn (n = 91, 92,93, 96)*1	Channel protection control register	HBGPROT0	<HBGn_base> + 00_H	8, 16, 32	PBG90#0	HBGKCPROT
	Channel SPID setting register	HBGPROT1	<HBGn_base> + 04_H	8, 16, 32	PBG90#0	HBGKCPROT

Note 1. This function is not implemented in RH850/U2A6 with index n = 91, 93 and in RH850/U2A6 (156/144 pins) with index n = 96.

Table 44.428 Relation between guard controllers and capture modules for protection targets

Protection target	Controlled by	Capture to	Applicable for
H-Bus Group 0 registers (RHSIF0)*1	HBG91	HBGERRSLV91	Access request from all masters
H-Bus Group 1 registers (FLXA0)	HBG92	HBGERRSLV92	
H-Bus Group 2 registers (FLXA1)*1	HBG93	HBGERRSLV93	
H-Bus Group 3 registers (SFMA0)*2	HBG96	HBGERRSLV96	

Note 1. This function is not implemented in RH850/U2A6.

Note 2. This function is not implemented in RH850/U2A6 (156/144 pins).

44.5.11.4 HBGOVFCLR — Guard Error Overflow Clear Register

This register is used to clear HBGOVFSTAT register. HBGOVFSTAT.OVF bit is cleared immediately after writing 1 to this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit, 16-bit or 8-bit units.

Address: <HBGERRSLVn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R

Table 44.429 HBGOVFCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset
1	CLRO	Clears HBGOVFSTAT.OVF bit by writing 1.
0	Reserved	When writing, write the value after reset.

44.5.11.5 HBGOVFSTAT — Guard Error Overflow Status Register

This register is used to notify users whether guard errors have occurred. OVF bit is set when a guard error is reported again under the condition any bit of HBGSPIDERRSTAT register is already set. This register is not writable and is cleared when writing 1 to HBGOVFCLR register. See **Section 44.5.11.4, HBGOVFCLR — Guard Error Overflow Clear Register**.

Access: This register is a read-only register that can be read in 32-bit, 16-bit or 8-bit units.

Address: <HBGERRSLVn_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.430 HBGOVFSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Overflow status flag
0	Reserved	When read, the value after reset is returned.

44.5.11.6 HBGERRADDR — Guard Error Address Register

This register is used to get an access address when a guard error is occurred. Error address is stored in this register only when the HBGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <HBGERRSLVn_base> + 08_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.431 HBGERRADDR Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access address to the target slave when a guard error has occurred

44.5.11.7 HBGERRTYPE — Guard Error Access Information Register

This register is used to get an access attributes when a guard error is occurred. Error information is stored in this register only when the HBGSPIDERRSTAT register is all 0.

Access: This register is a read-only register that can be read in 32-bit or 16-bit units.

Address: <HBGERRSLVn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SEC	DBG	UM	SPID[4:0]					—	—	—	—	—	WRITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.432 HBGERRTYPE Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned.
13	SEC	Access attribute of SEC to the target slave when a guard error has occurred
12	DBG	Access attribute of DBG to the target slave when a guard error has occurred
11	UM	Access attribute of UM to the target slave when a guard error has occurred
10 to 6	SPID[4:0]	Access attribute of SPID to the target slave when a guard error has occurred
5 to 1	Reserved	When read, the value after reset is returned.
0	WRITE	Access type of read or write to the target slave when a guard error has occurred

44.5.11.8 HBGSPIDERRCLR — Guard SPID Error Clear Register

This register is used to clear the HBGSPIDERRSTAT register. The HBGSPIDERRSTAT.SPIDERR[x] bit is cleared immediately after writing 1 to the x-th bit in this register. Read data is always 0 when reading this register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <HBGERRSLVn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDCLR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDCLR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 44.433 HBGSPIDERRCLR Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDCLR[31:0]	Clear PBGSPIDERRSTAT.SPIDERR bit.

44.5.11.9 HBGSPIDERRSTAT — Guard SPID Error Status Register

This register is used for users to know whether guard errors were occurred or not. The SPIDERR[x] bit is set when a guard error caused by a master whose SPID is x is notified by an error detecting module. This register is not writable and is cleared when writing 1 to the HBGSPIDERRCLR register. Guard error is notified to ECM when SPIDERR[x] changes from 0 to 1.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <HBGERRSLVn_base> + 14_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPIDERR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPIDERR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.434 HBGSPIDERRSTAT Register Contents

Bit Position	Bit Name	Function
31 to 0	SPIDERR[31:0]	Guard error status flag based on SPID 0: No guard error. 1: Shows SPID that a guard error has occurred.

44.5.11.10 HBGKCPROT — Key Code Protection Register

This register is used for protection against writing to the H-Bus guard control registers due to program malfunction and the like.

Access: This register can be read or written in 32-bit units.

Address: <HBGERRSLVn_base> + 18_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 44.435 HBGKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

44.5.11.11 HBGPROT0 — Channel Protection Control Register

Specifies the H-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the settings other than SPID.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <HBGn_base> + 00_H

Value after reset: $0000\ 0141_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LOCK	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GEN	—	DBG	—	UM	—	—	WG	RG
Value after reset	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R	R/W	R/W

Table 44.436 HBGPROT0 Register Contents (1/2)

Bit Position	Bit Name	Function
31	LOCK	Allows to block any further change of HBGPROT0 and HBGPROT1. 0: Register can be re-written 1: Any further write to HBGPROT0 and HBGPROT1 is ignored This bit can only be cleared by reset.
30 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	GEN	Enables/disables guard setting 0: Disables the guard setting 1: Enables the guard setting **CAUTION** When this bit is set to 1, HBG causes incorrect write accesses when: 1. Two consecutive*1 write access to same bus slave. 2. Second write*2 access is guarded by HBG. Following things are occurred. 1. First write access cannot complete. 2. Second access's write data is written to first access's address. 3. H-Bus cannot accept new read/write*3. Note 1. Consecutive: Two sequential write access one after another without empty bus cycle in between (within 2 bus cycle). Note 2. Read operation is not affected. Note 3. H-Bus cannot respond the bus access completion to the bus master. And H-Bus cannot accept new read/write request.
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	DBG	R/W enable setting for debug master 0: Depends on other enable/disable settings 1: Enables R/W
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UM	R/W disable setting in user mode 0: R/W disabled 1: R/W depends on other enable/disable settings

Table 44.436 HBGPROT0 Register Contents (2/2)

Bit Position	Bit Name	Function
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	WG	Write Global Enable 0: During write, HBGPROT1 is used as a judgment condition. 1: During write, HBGPROT1 is not used as a judgment condition.
0	RG	Read Global Enable 0: During read, HBGPROT1 is used as the judgment condition. 1: During read, HBGPROT1 is not used as a judgment condition.

44.5.11.12 HBGPROT1 — Channel SPID Setting Register

Specifies the H-Bus peripheral access to be protected against. Access prohibited by any of the identifiers is blocked as unauthorized access. This register is used to specify the SPID setting.

Access: This register can be read or written in 32-bit, 16-bit or 8-bit units.

Address: <HBGn_base> + 04_H

Value after reset: $0000\ 000F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPID[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.437 HBGPROT1 Register Contents

Bit Position	Bit Name	Function
31 to 0	SPID[31:0]	R/W enable setting based on SPID HBGPROT1 is a list of bits each representing one SPID value. Multiple SPID values are enabled simultaneously by setting multiple bits. For example, setting HBGPROT1 to 0101_B enables access to areas with SPID = 0 and SPID = 2. 0: Reading/writing the area with SPID = m depends on the RG and WG bit setting 1: Enables reading/writing the area with SPID = m

44.6 BIST

44.6.1 Overview

This product incorporates Built-In-Self-Test (BIST) function to detect failures of the safety mechanism itself. The BIST consists of Logic BIST (LBIST) and Memory BIST (MBIST). LBIST is considered as the effective hardware safety mechanism to measure latent faults which can reduce software load for error injection tests.

Power-On BIST is executed for safety mechanism in both of ISO and AWO areas during the reset sequence of Power On Reset, System Reset 1 and System Reset 2 before the CPU starts operation (called FBIST0). In addition, by setting System Reset 2, BIST (FBIST0) can be performed during the reset sequence in accordance with the setting of the BSEQ0CTL register (hereafter called Shut-Down BIST).

Standby-Resume BIST is executed for safety mechanism in ISO area only during the reset sequence of DeepSTOP reset before the CPU starts operation. By entering DeepSTOP mode, BIST (FBIST1) can be performed during the reset sequence in accordance with the setting of the BSEQ1CTL register after DeepSTOP Reset is generated when MCU mode transits from DeepSTOP mode to RUN mode. Also when MCU mode transits from DeepSTOP mode to CyclicRUN mode, BIST (FBIST2) can be performed during the reset sequence in accordance with the setting of the BSEQ2CTL register after DeepSTOP Reset is generated.

BIST execution results can be identified by the BIST result register (BSEQ0ST). When running BIST, any of LBIST ONLY, MBIST ONLY or both can be selected either by using flash option byte 3 (at Power-On BIST using Power On Reset or System Reset 1, or at Shut-Down BIST using System Reset 2) or by using BSEQ0SEL register (at Shut-Down BIST using System Reset 2 or Standby-Resume BIST using DeepSTOP Reset).

44.6.2 Registers

44.6.2.1 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc. Write protected registers can be written by releasing the protection of RESKCPROT0. For details of RESKCPROT0, see **Section 9, Reset Controller**.

44.6.2.2 List of Registers

Table 44.438 List of Registers (1/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
BIST	Logic BIST reference value register 1	LBISTREF1	FF9A 6000$_H$	32	PBG20#4	—
BIST	Logic BIST reference value register 2	LBISTREF2	FF9A 6004$_H$	32	PBG20#4	—
BIST	Memory BIST reference value register 1	MBISTREF1	FF9A 6008$_H$	32	PBG20#4	—
BIST	Memory BIST reference value register 2	MBISTREF2	FF9A 600C$_H$	32	PBG20#4	—
BIST	Memory BIST reference value register 3	MBISTREF3	FF9A 6010$_H$	32	PBG20#4	—
BIST	Logic BIST signature value register 1	LBISTSIG1	FF9A 6014$_H$	32	PBG20#4	—

Table 44.438 List of Registers (2/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
BIST	Logic BIST signature value register 2	LBISTSIG2	FF9A 6018$_H$	32	PBG20#4	—
BIST	Memory BIST signature value register 1	MBISTSIG1	FF9A 601C$_H$	32	PBG20#4	—
BIST	Memory BIST signature value register 2	MBISTSIG2	FF9A 6020$_H$	32	PBG20#4	—
BIST	Memory BIST signature value register 3	MBISTSIG3	FF9A 6024$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 0	MBIST1FTAG0	FF9A 6028$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 1	MBIST1FTAG1	FF9A 602C$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 2	MBIST1FTAG2	FF9A 6030$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 3	MBIST1FTAG3	FF9A 6034$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 4	MBIST1FTAG4	FF9A 6038$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 5	MBIST1FTAG5	FF9A 603C$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 6	MBIST1FTAG6	FF9A 6040$_H$	32	PBG20#4	—
BIST	Memory BIST1 FTAG signature value register 7	MBIST1FTAG7	FF9A 6044$_H$	32	PBG20#4	—
BIST	Memory BIST2 FTAG signature value register 0	MBIST2FTAG0	FF9A 6048$_H$	32	PBG20#4	—
BIST	Memory BIST2 FTAG signature value register 1	MBIST2FTAG1	FF9A 604C$_H$	32	PBG20#4	—
BIST	Memory BIST2 FTAG signature value register 2	MBIST2FTAG2	FF9A 6050$_H$	32	PBG20#4	—
BIST	Memory BIST3 FTAG signature value register 0	MBIST3FTAG0	FF9A 6068$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 0	MBIST1ECC0	FF9A 6088$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 1	MBIST1ECC1	FF9A 608C$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 2	MBIST1ECC2	FF9A 6090$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 3	MBIST1ECC3	FF9A 6094$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 4	MBIST1ECC4	FF9A 6098$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 5	MBIST1ECC5	FF9A 609C$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 6	MBIST1ECC6	FF9A 60A0$_H$	32	PBG20#4	—
BIST	Memory BIST1 ECC result register 7	MBIST1ECC7	FF9A 60A4$_H$	32	PBG20#4	—
BIST	Memory BIST2 ECC result register 0	MBIST2ECC0	FF9A 60A8$_H$	32	PBG20#4	—
BIST	Memory BIST2 ECC result register 1	MBIST2ECC1	FF9A 60AC$_H$	32	PBG20#4	—

Table 44.438 List of Registers (3/3)

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
BIST	Memory BIST2 ECC result register 2	MBIST2ECC2	FF9A 60B0$_H$	32	PBG20#4	—
BIST	Memory BIST3 ECC result register 0	MBIST3ECC0	FF9A 60C8$_H$	32	PBG20#4	—
BIST	BIST Sequence status register	BSEQ0ST	FF9A 4400$_H$	32	PBG20#4	—
BIST	BIST Sequence inverted status register	BSEQ0STB	FF9A 4404$_H$	32	PBG20#4	—
BIST	BIST Result register	BISTST	FF9A 4408$_H$	32	PBG20#4	—
BIST	BIST scenario select register	BSEQ0SEL	FF9A 440C$_H$	32	PBG20#4	—
SYS0	BIST skip control register for FBIST0	BSEQ0CTL	FF98 8400$_H$	32	PBG20#2	RESKCPROT0*1
SYS0	BIST skip control register for FBIST1	BSEQ1CTL	FF98 8404$_H$	32	PBG20#2	RESKCPROT0*1
SYS0	BIST skip control register for FBIST2	BSEQ2CTL	FF98 8408$_H$	32	PBG20#2	RESKCPROT0*1

Note 1. See **Section 9, Reset Controller**.

44.6.2.3 Reset of Registers

Register reset conditions are shown in **Table 44.439, Reset Sources**.

Table 44.439 Reset Sources (1/2)

Register Name	Reset Source						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
LBISTREF1	√	√	√	—	√	—	—
LBISTREF2	√	√	√	—	√	—	—
MBISTREF1	√	√	√	—	√	—	—
MBISTREF2	√	√	√	—	√	—	—
MBISTREF3	√	√	√	—	√	—	—
LBISTSIG1	√	√	√	—	√	—	—
LBISTSIG2	√	√	√	—	√	—	—
MBISTSIG1	√	√	√	—	√	—	—
MBISTSIG2	√	√	√	—	√	—	—
MBISTSIG3	√	√	√	—	√	—	—
MBIST1FTAG0	√	√	√	—	√	—	—
MBIST1FTAG1	√	√	√	—	√	—	—
MBIST1FTAG2	√	√	√	—	√	—	—
MBIST1FTAG3	√	√	√	—	√	—	—
MBIST1FTAG4	√	√	√	—	√	—	—
MBIST1FTAG5	√	√	√	—	√	—	—
MBIST1FTAG6	√	√	√	—	√	—	—
MBIST1FTAG7	√	√	√	—	√	—	—
MBIST2FTAG0	√	√	√	—	√	—	—
MBIST2FTAG1	√	√	√	—	√	—	—
MBIST2FTAG2	√	√	√	—	√	—	—
MBIST3FTAG0	√	√	√	—	√	—	—
MBIST1ECC0	√	√	√	—	√	—	—
MBIST1ECC1	√	√	√	—	√	—	—
MBIST1ECC2	√	√	√	—	√	—	—
MBIST1ECC3	√	√	√	—	√	—	—
MBIST1ECC4	√	√	√	—	√	—	—
MBIST1ECC5	√	√	√	—	√	—	—
MBIST1ECC6	√	√	√	—	√	—	—
MBIST1ECC7	√	√	√	—	√	—	—
MBIST2ECC0	√	√	√	—	√	—	—
MBIST2ECC1	√	√	√	—	√	—	—
MBIST2ECC2	√	√	√	—	√	—	—
MBIST3ECC0	√	√	√	—	√	—	—
BSEQ0ST	√	√	√	—	√	—	—
BSEQ0STB	√	√	√	—	√	—	—
BISTST	√	√	√	—	√	—	—
BSEQ0SEL	√	√	—	—	—	—	—

Table 44.439 Reset Sources (2/2)

Register Name	Reset Source						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
BSEQ0CTL	√	√	—	—	—	—	—
BSEQ1CTL	√	√	—	—	—	—	—
BSEQ2CTL	√	√	—	—	—	—	—

44.6.2.4 LBISTREF1 — Logic BIST Reference Value Register 1

This register indicates the reference value of the Logic BIST. This register is automatically updated by the reference value after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6000$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTREF1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTREF1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.440 LBISTREF1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	LBISTREF1[19:0]	Reference signature value (expected signature value)*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	000A 5A5A$_H$
BIST is not executed without any of LBIST scenario:	Reference signature value (expected signature value)
BIST is not executed with any of LBIST scenario:	Reference signature value (expected signature value)
BIST is executed without any of LBIST scenario:	Reference signature value (expected signature value)
BIST is executed with any of LBIST scenario:	Reference signature value (expected signature value)

44.6.2.5 LBISTREF2 — Logic BIST Reference Value Register 2

This register indicates the reference value of the Logic BIST. This register is automatically updated by the reference value after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6004$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTREF2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTREF2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.441 LBISTREF2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	LBISTREF2[19:0]	Reference signature value (expected signature value)*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0005 A5A5$_H$
BIST is not executed without any of LBIST scenario:	Reference signature value (expected signature value)
BIST is not executed with any of LBIST scenario:	Reference signature value (expected signature value)
BIST is executed without any of LBIST scenario:	Reference signature value (expected signature value)
BIST is executed with any of LBIST scenario:	Reference signature value (expected signature value)

44.6.2.6 MBISTREF1 — Memory BIST Reference Value Register 1

This register indicates the reference value of the Memory BIST. This register is automatically updated by the reference value after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6008$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTREF1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTREF1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.442 MBISTREF1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTREF1[19:0]	Reference signature value (expected signature value)*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	000A A55A$_H$
BIST is not executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is not executed with any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed with any of MBIST scenario:	Reference signature value (expected signature value)

44.6.2.7 MBISTREF2 — Memory BIST Reference Value Register 2

This register indicates the reference value of the Memory BIST. This register is automatically updated by the reference value after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 600C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTREF2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTREF2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.443 MBISTREF2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTREF2[19:0]	Reference signature value (expected signature value)*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0005 5AA5$_H$
BIST is not executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is not executed with any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed with any of MBIST scenario:	Reference signature value (expected signature value)

44.6.2.8 MBISTREF3 — Memory BIST Reference Value Register 3

This register indicates the reference value of the Memory BIST. This register is automatically updated by the reference value after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6010$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTREF3[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTREF3[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.444 MBISTREF3 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTREF3[19:0]	Reference signature value (expected signature value)*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	000A AA55$_H$
BIST is not executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is not executed with any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed without any of MBIST scenario:	Reference signature value (expected signature value)
BIST is executed with any of MBIST scenario:	Reference signature value (expected signature value)

44.6.2.9 LBISTSIG1 — Logic BIST Signature Value Register 1

This register indicates the signature value of the Logic BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user compares the reference signature of LBISTREF1 against the resulting signature LBISTSIG1. The Logic BIST is passed if these are equal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6014$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTSIG1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTSIG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.445 LBISTSIG1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	LBISTSIG1[19:0]	Logic BIST1 result signature value*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0005 A5A5$_H$
BIST is not executed without any of LBIST scenario:	0005 A5A5$_H$
BIST is not executed with any of LBIST scenario:	Inverted value of LBISTREF1 register
BIST is executed without any of LBIST scenario:	0005 A5A5$_H$
BIST is executed with any of LBIST scenario:	Logic BIST1 result signature value

44.6.2.10 LBISTSIG2 — Logic BIST Signature Value Register 2

This register indicates the signature value of the Logic BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of LBISTREF2 against the resulting signature LBISTSIG2. The Logic BIST is passed if these are equal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6018$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	LBISTSIG2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBISTSIG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.446 LBISTSIG2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	LBISTSIG2[19:0]	Logic BIST2 result signature value*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	000A 5A5A$_H$
BIST is not executed without any of LBIST scenario:	000A 5A5A$_H$
BIST is not executed with any of LBIST scenario:	Inverted value of LBISTREF2 register
BIST is executed without any of LBIST scenario:	000A 5A5A$_H$
BIST is executed with any of LBIST scenario 9 or 11 to 20:	000A 5A5A$_H$
BIST is executed with any of LBIST scenario 1 to 3, 5 to 7 or 10:	Logic BIST2 result signature value

44.6.2.11 MBISTSIG1 — Memory BIST Signature Value Register 1

This register indicates the signature value of the Memory BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of MBISTREF1 against the resulting signature MBISTSIG1. The Memory BIST is passed if these are equal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 601C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTSIG1[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTSIG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.447 MBISTSIG1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTSIG1[19:0]	Memory BIST1 result signature value*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0005 5AA5$_H$
BIST is not executed without any of MBIST scenario:	0005 5AA5$_H$
BIST is not executed with any of MBIST scenario:	Inverted value of MBISTREF1 register
BIST is executed without any of MBIST scenario:	0005 5AA5$_H$
BIST is executed with any of MBIST scenario:	Memory BIST1 result signature value

44.6.2.12 MBISTSIG2 — Memory BIST Signature Value Register 2

This register indicates the signature value of the Memory BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of MBISTREF2 against the resulting signature MBISTSIG2. The Memory BIST is passed if these are equal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6020$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTSIG2[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTSIG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.448 MBISTSIG2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTSIG2[19:0]	Memory BIST2 result signature value*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	000A A55A$_H$
BIST is not executed without any of MBIST scenario:	000A A55A$_H$
BIST is not executed with any of MBIST scenario:	Inverted value of MBISTREF2 register
BIST is executed without any of MBIST scenario:	000A A55A$_H$
BIST is executed with either of MBIST scenario 3 or 4:	Inverted value of MBISTREF2 register
BIST is executed with either of MBIST scenario 1 or 2:	Memory BIST2 result signature value

44.6.2.13 MBISTSIG3 — Memory BIST Signature Value Register 3

This register indicates the signature value of the Memory BIST. This register is automatically updated by the BIST result signature after BIST execution.

The user shall compare the reference signature of MBISTREF3 against the resulting signature MBISTSIG3. The Memory BIST is passed if these are equal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6024$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	MBISTSIG3[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBISTSIG3[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.449 MBISTSIG3 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 0	MBISTSIG3[19:0]	Memory BIST3 result signature value*1

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0005 55AA$_H$
BIST is not executed without any of MBIST scenario 1:	0005 55AA$_H$
BIST is not executed with MBIST scenario 1:	Inverted value of MBISTREF3 register
BIST is executed without any of MBIST scenario 1:	0005 55AA$_H$
BIST is executed with MBIST scenario 1:	Memory BIST3 result signature value

44.6.2.14 MBIST1FTAG0 — Memory BIST1 FTAG signature value register 0

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6028$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.450 MBIST1FTAG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG0[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.451, Mapping of MBIST1FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped: FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario: FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario: FFFF FFFF$_H$
BIST is executed without any of MBIST scenario: FFFF FFFF$_H$
BIST is executed with any of MBIST scenario: The state of Memory BIST result for each RAM

Table 44.451 Mapping of MBIST1FTAG0 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG0[31]	ICUM Local RAM	√	√	√	√	FLXA0 MRAM	√	√	√	√
30	MBIST1FTAG0[30]	ICUM Local RAM	√	√	√	√	FLXA0 TBFRAM A/B	√	√	√	√
29	MBIST1FTAG0[29]	ICUM Local RAM	√	√	√	√	Reserved	—	—	—	—
28	MBIST1FTAG0[28]	ICUM Local RAM	√	√	√	√	Reserved	—	—	—	—
27	MBIST1FTAG0[27]	Reserved	—	—	—	—	FLXA1 MRAM	√	√	√	—
26	MBIST1FTAG0[26]	Reserved	—	—	—	—	FLXA1 TBFRAM A/B	√	√	√	—
25	MBIST1FTAG0[25]	Reserved	—	—	—	—	Reserved	—	—	—	—
24	MBIST1FTAG0[24]	Reserved	—	—	—	—	Reserved	—	—	—	—
23	MBIST1FTAG0[23]	ICUM Cache RAM (tag0)	√	√	√	√	ETNB0 Transmit FIFO	√	√	√	√

Table 44.451 Mapping of MBIST1FTAG0 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
22	MBIST1FTAG0[22]	ICUM Cache RAM (tag1)	√	√	√	√	ETNB0 Receive FIFO	√	√	√	√
21	MBIST1FTAG0[21]	ICUM Cache RAM (data0)	√	√	√	√	ETNB1 Transmit FIFO	√	√	√	—
20	MBIST1FTAG0[20]	ICUM Cache RAM (data0)	√	√	√	√	ETNB1 Receive FIFO	√	√	√	—
19	MBIST1FTAG0[19]	ICUM Cache RAM (data0)	√	√	√	√	Reserved	—	—	—	—
18	MBIST1FTAG0[18]	ICUM Cache RAM (data0)	√	√	√	√	Reserved	—	—	—	—
17	MBIST1FTAG0[17]	ICUM Cache RAM (data1)	√	√	√	√	MMCA0 RAM A/B	√	√	√	√
16	MBIST1FTAG0[16]	ICUM Cache RAM (data1)	√	√	√	√	Reserved	—	—	—	—
15	MBIST1FTAG0[15]	ICUM Cache RAM (data1)	√	√	√	√	MSPI0/1/2/3 RAM	√	√	√	√
14	MBIST1FTAG0[14]	ICUM Cache RAM (data1)	√	√	√	√	MSPI4/5/6/7 RAM	√	√	√	√*2
13	MBIST1FTAG0[13]	PKCC RAM	√	√	√	√	MSPI8/9 RAM	√	√	√*1	—
12	MBIST1FTAG0[12]	Reserved	—	—	—	—	Reserved	—	—	—	—
11	MBIST1FTAG0[11]	DTSRAM	√	√	√	√	Reserved	—	—	—	—
10	MBIST1FTAG0[10]	Reserved	—	—	—	—	Reserved	—	—	—	—
9	MBIST1FTAG0[9]	Reserved	—	—	—	—	Reserved	—	—	—	—
8	MBIST1FTAG0[8]	Reserved	—	—	—	—	Reserved	—	—	—	—
7	MBIST1FTAG0[7]	sDMAC0 DATARAM	√	√	√	√	GTM MCS0/1 RAM 0	√	√	√	√
6	MBIST1FTAG0[6]	sDMAC0 DPRAM	√	√	√	√	GTM MCS2/3 RAM 0	√	√	√	√
5	MBIST1FTAG0[5]	Reserved	—	—	—	—	GTM MCS0/1 RAM 1	√	√	√	√
4	MBIST1FTAG0[4]	Reserved	—	—	—	—	GTM MCS2/3 RAM 1	√	√	√	√
3	MBIST1FTAG0[3]	sDMAC1 DATARAM	√	√	√	√	Reserved	—	—	—	—
2	MBIST1FTAG0[2]	sDMAC1 DPRAM	√	√	√	√	Reserved	—	—	—	—
1	MBIST1FTAG0[1]	Reserved	—	—	—	—	Reserved	—	—	—	—
0	MBIST1FTAG0[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

Note 1. MSPI9 is not implemented in RH850/U2A8.

Note 2. MSPI6/7 are not implemented in RH850/U2A6.

44.6.2.15 MBIST1FTAG1 — Memory BIST1 FTAG Signature Value Register 1

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 602C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG1[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.452 MBIST1FTAG1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG1[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.453, Mapping of MBIST1FTAG1 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.453 Mapping of MBIST1FTAG1 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG1[31]	Local RAM (PE0) Address from FDC0 0000_H to FDC0 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD0 AFLRAM 0	√	√	√	√
30	MBIST1FTAG1[30]	Local RAM (PE0) Address from FDC0 0000_H to FDC0 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD0 AFLRAM 0	√	—	—	—
29	MBIST1FTAG1[29]	Local RAM (PE0) Address from FDC0 0000_H to FDC0 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD0 AFLRAM 1	√	√	√	√
28	MBIST1FTAG1[28]	Local RAM (PE0) Address from FDC0 0000_H to FDC0 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD0 AFLRAM 1	√	—	—	—
27	MBIST1FTAG1[27]	Local RAM (PE1) Address from FDA0 0000_H to FDA0 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD0 MRAM	√	√	√	√
26	MBIST1FTAG1[26]	Local RAM (PE1) Address from FDA0 0000_H to FDA0 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD0 MRAM	√	√	√	√
25	MBIST1FTAG1[25]	Local RAM (PE1) Address from FDA0 0000_H to FDA0 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD0 MRAM	√	√	√	√
24	MBIST1FTAG1[24]	Local RAM (PE1) Address from FDA0 0000_H to FDA0 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD0 MRAM	√	√	√	√
23	MBIST1FTAG1[23]	Instruction cache RAM (tag0/1/2/3) (PE0)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
22	MBIST1FTAG1[22]	Instruction cache RAM (data0/1) (PE0)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
21	MBIST1FTAG1[21]	Instruction cache RAM (data2/3) (PE0)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
20	MBIST1FTAG1[20]	Reserved	—	—	—	—	RS-CANFD0 MRAM	√	√	√	—
19	MBIST1FTAG1[19]	Instruction cache RAM (tag0/1/2/3) (PE1)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
18	MBIST1FTAG1[18]	Instruction cache RAM (data0/1) (PE1)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
17	MBIST1FTAG1[17]	Instruction cache RAM (data2/3) (PE1)	√	√	√	√	RS-CANFD0 MRAM	√	√	√	—
16	MBIST1FTAG1[16]	Reserved	—	—	—	—	RS-CANFD0 MRAM	√	√	√	—
15	MBIST1FTAG1[15]	Local RAM (PE2) Address from FD80 0000_H to FD80 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	RS-CANFD0 MRAM	√	√	√	—
14	MBIST1FTAG1[14]	Local RAM (PE2) Address from FD80 0000_H to FD80 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
13	MBIST1FTAG1[13]	Local RAM (PE2) Address from FD80 0000_H to FD80 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—

Table 44.453 Mapping of MBIST1FTAG1 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
12	MBIST1FTAG1[12]	Local RAM (PE2) Address from FD80 0000_H to FD80 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
11	MBIST1FTAG1[11]	Local RAM (PE3) Address from FD60 0000_H to FD60 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
10	MBIST1FTAG1[10]	Local RAM (PE3) Address from FD60 0000_H to FD60 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
9	MBIST1FTAG1[9]	Local RAM (PE3) Address from FD60 0000_H to FD60 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
8	MBIST1FTAG1[8]	Local RAM (PE3) Address from FD60 0000_H to FD60 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
7	MBIST1FTAG1[7]	Instruction cache RAM (tag0/1/2/3) (PE2)	√	√	—	—	Reserved	—	—	—	—
6	MBIST1FTAG1[6]	Instruction cache RAM (data0/1) (PE2)	√	√	—	—	Reserved	—	—	—	—
5	MBIST1FTAG1[5]	Instruction cache RAM (data2/3) (PE2)	√	√	—	—	Reserved	—	—	—	—
4	MBIST1FTAG1[4]	Reserved	—	—	—	—	Reserved	—	—	—	—
3	MBIST1FTAG1[3]	Instruction cache RAM (tag0/1/2/3) (PE3)	√	√	—	—	Reserved	—	—	—	—
2	MBIST1FTAG1[2]	Instruction cache RAM (data0/1) (PE3)	√	√	—	—	Reserved	—	—	—	—
1	MBIST1FTAG1[1]	Instruction cache RAM (data2/3) (PE3)	√	√	—	—	Reserved	—	—	—	—
0	MBIST1FTAG1[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

44.6.2.16 MBIST1FTAG2 — Memory BIST1 FTAG Signature Value Register 2

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6030$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG2[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.454 MBIST1FTAG2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG2[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.455, Mapping of MBIST1FTAG2 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.455 Mapping of MBIST1FTAG2 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG2[31]	Cluster RAM (Cluster 0) Address from FE00 0000_H to FE00 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD1 AFLRAM 0	√	√	√	√
30	MBIST1FTAG2[30]	Cluster RAM (Cluster 0) Address from FE00 0000_H to FE00 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD1 AFLRAM 0	√	—	—	—
29	MBIST1FTAG2[29]	Cluster RAM (Cluster 0) Address from FE00 0000_H to FE00 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD1 AFLRAM 1	√	√	√	√
28	MBIST1FTAG2[28]	Cluster RAM (Cluster 0) Address from FE00 0000_H to FE00 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD1 AFLRAM 1	√	—	—	—
27	MBIST1FTAG2[27]	Cluster RAM (Cluster 0) Address from FE01 0000_H to FE01 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	√
26	MBIST1FTAG2[26]	Cluster RAM (Cluster 0) Address from FE01 0000_H to FE01 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	√
25	MBIST1FTAG2[25]	Cluster RAM (Cluster 0) Address from FE01 0000_H to FE01 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
24	MBIST1FTAG2[24]	Cluster RAM (Cluster 0) Address from FE01 0000_H to FE01 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
23	MBIST1FTAG2[23]	Cluster RAM (Cluster 0) Address from FE02 0000_H to FE02 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
22	MBIST1FTAG2[22]	Cluster RAM (Cluster 0) Address from FE02 0000_H to FE02 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
21	MBIST1FTAG2[21]	Cluster RAM (Cluster 0) Address from FE02 0000_H to FE02 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
20	MBIST1FTAG2[20]	Cluster RAM (Cluster 0) Address from FE02 0000_H to FE02 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
19	MBIST1FTAG2[19]	Cluster RAM (Cluster 0) Address from FE03 0000_H to FE03 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
18	MBIST1FTAG2[18]	Cluster RAM (Cluster 0) Address from FE03 0000_H to FE03 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
17	MBIST1FTAG2[17]	Cluster RAM (Cluster 0) Address from FE03 0000_H to FE03 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
16	MBIST1FTAG2[16]	Cluster RAM (Cluster 0) Address from FE03 0000_H to FE03 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
15	MBIST1FTAG2[15]	Cluster RAM (Cluster 0) Address from FE04 0000_H to FE04 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	RS-CANFD1 MRAM	√	√	√	—
14	MBIST1FTAG2[14]	Cluster RAM (Cluster 0) Address from FE04 0000_H to FE04 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
13	MBIST1FTAG2[13]	Cluster RAM (Cluster 0) Address from FE04 0000_H to FE04 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
12	MBIST1FTAG2[12]	Cluster RAM (Cluster 0) Address from FE04 0000_H to FE04 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—

Table 44.455 Mapping of MBIST1FTAG2 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG2[11]	Cluster RAM (Cluster 0) Address from FE05 0000_H to FE05 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	Reserved	—	—	—	—
10	MBIST1FTAG2[10]	Cluster RAM (Cluster 0) Address from FE05 0000_H to FE05 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
9	MBIST1FTAG2[9]	Cluster RAM (Cluster 0) Address from FE05 0000_H to FE05 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
8	MBIST1FTAG2[8]	Cluster RAM (Cluster 0) Address from FE05 0000_H to FE05 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—
7	MBIST1FTAG2[7]	Cluster RAM (Cluster 0) Address from FE06 0000_H to FE06 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	Reserved	—	—	—	—
6	MBIST1FTAG2[6]	Cluster RAM (Cluster 0) Address from FE06 0000_H to FE06 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
5	MBIST1FTAG2[5]	Cluster RAM (Cluster 0) Address from FE06 0000_H to FE06 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
4	MBIST1FTAG2[4]	Cluster RAM (Cluster 0) Address from FE06 0000_H to FE06 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—
3	MBIST1FTAG2[3]	Cluster RAM (Cluster 0) Address from FE07 0000_H to FE07 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	Reserved	—	—	—	—
2	MBIST1FTAG2[2]	Cluster RAM (Cluster 0) Address from FE07 0000_H to FE07 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
1	MBIST1FTAG2[1]	Cluster RAM (Cluster 0) Address from FE07 0000_H to FE07 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
0	MBIST1FTAG2[0]	Cluster RAM (Cluster 0) Address from FE07 0000_H to FE07 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—

44.6.2.17 MBIST1FTAG3 — Memory BIST1 FTAG Signature Value Register 3

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6034$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG3[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG3[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.456 MBIST1FTAG3 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG3[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.457, Mapping of MBIST1FTAG3 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.457 Mapping of MBIST1FTAG3 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG3[31]	Cluster RAM (Cluster 1) Address from FE10 0000_H to FE10 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
30	MBIST1FTAG3[30]	Cluster RAM (Cluster 1) Address from FE10 0000_H to FE10 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
29	MBIST1FTAG3[29]	Cluster RAM (Cluster 1) Address from FE10 0000_H to FE10 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
28	MBIST1FTAG3[28]	Cluster RAM (Cluster 1) Address from FE10 0000_H to FE10 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
27	MBIST1FTAG3[27]	Cluster RAM (Cluster 1) Address from FE11 0000_H to FE11 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
26	MBIST1FTAG3[26]	Cluster RAM (Cluster 1) Address from FE11 0000_H to FE11 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
25	MBIST1FTAG3[25]	Cluster RAM (Cluster 1) Address from FE11 0000_H to FE11 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
24	MBIST1FTAG3[24]	Cluster RAM (Cluster 1) Address from FE11 0000_H to FE11 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
23	MBIST1FTAG3[23]	Cluster RAM (Cluster 1) Address from FE12 0000_H to FE12 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
22	MBIST1FTAG3[22]	Cluster RAM (Cluster 1) Address from FE12 0000_H to FE12 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
21	MBIST1FTAG3[21]	Cluster RAM (Cluster 1) Address from FE12 0000_H to FE12 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
20	MBIST1FTAG3[20]	Cluster RAM (Cluster 1) Address from FE12 0000_H to FE12 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
19	MBIST1FTAG3[19]	Cluster RAM (Cluster 1) Address from FE13 0000_H to FE13 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
18	MBIST1FTAG3[18]	Cluster RAM (Cluster 1) Address from FE13 0000_H to FE13 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
17	MBIST1FTAG3[17]	Cluster RAM (Cluster 1) Address from FE13 0000_H to FE13 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
16	MBIST1FTAG3[16]	Cluster RAM (Cluster 1) Address from FE13 0000_H to FE13 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
15	MBIST1FTAG3[15]	Cluster RAM (Cluster 1) Address from FE14 0000_H to FE14 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
14	MBIST1FTAG3[14]	Cluster RAM (Cluster 1) Address from FE14 0000_H to FE14 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
13	MBIST1FTAG3[13]	Cluster RAM (Cluster 1) Address from FE14 0000_H to FE14 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
12	MBIST1FTAG3[12]	Cluster RAM (Cluster 1) Address from FE14 0000_H to FE14 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—

Table 44.457 Mapping of MBIST1FTAG3 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG3[11]	Cluster RAM (Cluster 1) Address from FE15 0000_H to FE15 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
10	MBIST1FTAG3[10]	Cluster RAM (Cluster 1) Address from FE15 0000_H to FE15 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
9	MBIST1FTAG3[9]	Cluster RAM (Cluster 1) Address from FE15 0000_H to FE15 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
8	MBIST1FTAG3[8]	Cluster RAM (Cluster 1) Address from FE15 0000_H to FE15 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
7	MBIST1FTAG3[7]	Cluster RAM (Cluster 1) Address from FE16 0000_H to FE16 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
6	MBIST1FTAG3[6]	Cluster RAM (Cluster 1) Address from FE16 0000_H to FE16 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
5	MBIST1FTAG3[5]	Cluster RAM (Cluster 1) Address from FE16 0000_H to FE16 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
4	MBIST1FTAG3[4]	Cluster RAM (Cluster 1) Address from FE16 0000_H to FE16 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
3	MBIST1FTAG3[3]	Cluster RAM (Cluster 1) Address from FE17 0000_H to FE17 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
2	MBIST1FTAG3[2]	Cluster RAM (Cluster 1) Address from FE17 0000_H to FE17 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
1	MBIST1FTAG3[1]	Cluster RAM (Cluster 1) Address from FE17 0000_H to FE17 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
0	MBIST1FTAG3[0]	Cluster RAM (Cluster 1) Address from FE17 0000_H to FE17 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—

44.6.2.18 MBIST1FTAG4 — Memory BIST1 FTAG Signature Value Register 4

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6038$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG4[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG4[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.458 MBIST1FTAG4 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG4[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.459, Mapping of MBIST1FTAG4 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.459 Mapping of MBIST1FTAG4 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG4[31]	Cluster RAM (Cluster 2) Address from FE40 0000_H to FE40 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
30	MBIST1FTAG4[30]	Cluster RAM (Cluster 2) Address from FE40 0000_H to FE40 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
29	MBIST1FTAG4[29]	Cluster RAM (Cluster 2) Address from FE40 0000_H to FE40 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
28	MBIST1FTAG4[28]	Cluster RAM (Cluster 2) Address from FE40 0000_H to FE40 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
27	MBIST1FTAG4[27]	Cluster RAM (Cluster 2) Address from FE41 0000_H to FE41 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
26	MBIST1FTAG4[26]	Cluster RAM (Cluster 2) Address from FE41 0000_H to FE41 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
25	MBIST1FTAG4[25]	Cluster RAM (Cluster 2) Address from FE41 0000_H to FE41 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
24	MBIST1FTAG4[24]	Cluster RAM (Cluster 2) Address from FE41 0000_H to FE41 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
23	MBIST1FTAG4[23]	Cluster RAM (Cluster 2) Address from FE42 0000_H to FE42 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
22	MBIST1FTAG4[22]	Cluster RAM (Cluster 2) Address from FE42 0000_H to FE42 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
21	MBIST1FTAG4[21]	Cluster RAM (Cluster 2) Address from FE42 0000_H to FE42 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
20	MBIST1FTAG4[20]	Cluster RAM (Cluster 2) Address from FE42 0000_H to FE42 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
19	MBIST1FTAG4[19]	Cluster RAM (Cluster 2) Address from FE43 0000_H to FE43 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
18	MBIST1FTAG4[18]	Cluster RAM (Cluster 2) Address from FE43 0000_H to FE43 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
17	MBIST1FTAG4[17]	Cluster RAM (Cluster 2) Address from FE43 0000_H to FE43 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
16	MBIST1FTAG4[16]	Cluster RAM (Cluster 2) Address from FE43 0000_H to FE43 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
15	MBIST1FTAG4[15]	Cluster RAM (Cluster 2) Address from FE44 0000_H to FE44 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
14	MBIST1FTAG4[14]	Cluster RAM (Cluster 2) Address from FE44 0000_H to FE44 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
13	MBIST1FTAG4[13]	Cluster RAM (Cluster 2) Address from FE44 0000_H to FE44 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
12	MBIST1FTAG4[12]	Cluster RAM (Cluster 2) Address from FE44 0000_H to FE44 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—

Table 44.459 Mapping of MBIST1FTAG4 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG4[11]	Cluster RAM (Cluster 2) Address from FE45 0000_H to FE45 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
10	MBIST1FTAG4[10]	Cluster RAM (Cluster 2) Address from FE45 0000_H to FE45 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
9	MBIST1FTAG4[9]	Cluster RAM (Cluster 2) Address from FE45 0000_H to FE45 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
8	MBIST1FTAG4[8]	Cluster RAM (Cluster 2) Address from FE45 0000_H to FE45 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
7	MBIST1FTAG4[7]	Cluster RAM (Cluster 2) Address from FE46 0000_H to FE46 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
6	MBIST1FTAG4[6]	Cluster RAM (Cluster 2) Address from FE46 0000_H to FE46 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
5	MBIST1FTAG4[5]	Cluster RAM (Cluster 2) Address from FE46 0000_H to FE46 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
4	MBIST1FTAG4[4]	Cluster RAM (Cluster 2) Address from FE46 0000_H to FE46 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
3	MBIST1FTAG4[3]	Cluster RAM (Cluster 2) Address from FE47 0000_H to FE47 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
2	MBIST1FTAG4[2]	Cluster RAM (Cluster 2) Address from FE47 0000_H to FE47 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
1	MBIST1FTAG4[1]	Cluster RAM (Cluster 2) Address from FE47 0000_H to FE47 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
0	MBIST1FTAG4[0]	Cluster RAM (Cluster 2) Address from FE47 0000_H to FE47 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—

44.6.2.19 MBIST1FTAG5 — Memory BIST1 FTAG Signature Value Register 5

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 603C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG5[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG5[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.460 MBIST1FTAG5 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG5[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.461, Mapping of MBIST1FTAG5 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.461 Mapping of MBIST1FTAG5 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG5[31]	Cluster RAM (Cluster 2) Address from FE48 0000_H to FE48 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
30	MBIST1FTAG5[30]	Cluster RAM (Cluster 2) Address from FE48 0000_H to FE48 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
29	MBIST1FTAG5[29]	Cluster RAM (Cluster 2) Address from FE48 0000_H to FE48 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
28	MBIST1FTAG5[28]	Cluster RAM (Cluster 2) Address from FE48 0000_H to FE48 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
27	MBIST1FTAG5[27]	Cluster RAM (Cluster 2) Address from FE49 0000_H to FE49 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
26	MBIST1FTAG5[26]	Cluster RAM (Cluster 2) Address from FE49 0000_H to FE49 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
25	MBIST1FTAG5[25]	Cluster RAM (Cluster 2) Address from FE49 0000_H to FE49 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
24	MBIST1FTAG5[24]	Cluster RAM (Cluster 2) Address from FE49 0000_H to FE49 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
23	MBIST1FTAG5[23]	Cluster RAM (Cluster 2) Address from FE4A 0000_H to FE4A $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
22	MBIST1FTAG5[22]	Cluster RAM (Cluster 2) Address from FE4A 0000_H to FE4A $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
21	MBIST1FTAG5[21]	Cluster RAM (Cluster 2) Address from FE4A 0000_H to FE4A $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
20	MBIST1FTAG5[20]	Cluster RAM (Cluster 2) Address from FE4A 0000_H to FE4A $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
19	MBIST1FTAG5[19]	Cluster RAM (Cluster 2) Address from FE4B 0000_H to FE4B $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
18	MBIST1FTAG5[18]	Cluster RAM (Cluster 2) Address from FE4B 0000_H to FE4B $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
17	MBIST1FTAG5[17]	Cluster RAM (Cluster 2) Address from FE4B 0000_H to FE4B $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
16	MBIST1FTAG5[16]	Cluster RAM (Cluster 2) Address from FE4B 0000_H to FE4B $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
15	MBIST1FTAG5[15]	Cluster RAM (Cluster 2) Address from FE4C 0000_H to FE4C $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
14	MBIST1FTAG5[14]	Cluster RAM (Cluster 2) Address from FE4C 0000_H to FE4C $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
13	MBIST1FTAG5[13]	Cluster RAM (Cluster 2) Address from FE4C 0000_H to FE4C $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
12	MBIST1FTAG5[12]	Cluster RAM (Cluster 2) Address from FE4C 0000_H to FE4C $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—

Table 44.461 Mapping of MBIST1FTAG5 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG5[11]	Cluster RAM (Cluster 2) Address from FE4D 0000_H to FE4D $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
10	MBIST1FTAG5[10]	Cluster RAM (Cluster 2) Address from FE4D 0000_H to FE4D $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
9	MBIST1FTAG5[9]	Cluster RAM (Cluster 2) Address from FE4D 0000_H to FE4D $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
8	MBIST1FTAG5[8]	Cluster RAM (Cluster 2) Address from FE4D 0000_H to FE4D $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
7	MBIST1FTAG5[7]	Cluster RAM (Cluster 2) Address from FE4E 0000_H to FE4E $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
6	MBIST1FTAG5[6]	Cluster RAM (Cluster 2) Address from FE4E 0000_H to FE4E $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
5	MBIST1FTAG5[5]	Cluster RAM (Cluster 2) Address from FE4E 0000_H to FE4E $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
4	MBIST1FTAG5[4]	Cluster RAM (Cluster 2) Address from FE4E 0000_H to FE4E $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—
3	MBIST1FTAG5[3]	Cluster RAM (Cluster 2) Address from FE4F 0000_H to FE4F $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	—	Reserved	—	—	—	—
2	MBIST1FTAG5[2]	Cluster RAM (Cluster 2) Address from FE4F 0000_H to FE4F $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	—	Reserved	—	—	—	—
1	MBIST1FTAG5[1]	Cluster RAM (Cluster 2) Address from FE4F 0000_H to FE4F $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	—	Reserved	—	—	—	—
0	MBIST1FTAG5[0]	Cluster RAM (Cluster 2) Address from FE4F 0000_H to FE4F $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	—	Reserved	—	—	—	—

44.6.2.20 MBIST1FTAG6 — Memory BIST1 FTAG Signature Value Register 6

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6040$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG6[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG6[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.462 MBIST1FTAG6 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG6[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.463, Mapping of MBIST1FTAG6 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.463 Mapping of MBIST1FTAG6 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG6[31]	Cluster RAM (Cluster 2) Address from FE50 0000$_H$ to FE50 FFFC$_H$ and 4 bit of LSB is 0$_H$	√	√	—	—	Reserved	—	—	—	—
30	MBIST1FTAG6[30]	Cluster RAM (Cluster 2) Address from FE50 0000$_H$ to FE50 FFFC$_H$ and 4 bit of LSB is 4$_H$	√	√	—	—	Reserved	—	—	—	—
29	MBIST1FTAG6[29]	Cluster RAM (Cluster 2) Address from FE50 0000$_H$ to FE50 FFFC$_H$ and 4 bit of LSB is 8$_H$	√	√	—	—	Reserved	—	—	—	—
28	MBIST1FTAG6[28]	Cluster RAM (Cluster 2) Address from FE50 0000$_H$ to FE50 FFFC$_H$ and 4 bit of LSB is C$_H$	√	√	—	—	Reserved	—	—	—	—
27	MBIST1FTAG6[27]	Cluster RAM (Cluster 2) Address from FE51 0000$_H$ to FE51 FFFC$_H$ and 4 bit of LSB is 0$_H$	√	√	—	—	Reserved	—	—	—	—
26	MBIST1FTAG6[26]	Cluster RAM (Cluster 2) Address from FE51 0000$_H$ to FE51 FFFC$_H$ and 4 bit of LSB is 4$_H$	√	√	—	—	Reserved	—	—	—	—
25	MBIST1FTAG6[25]	Cluster RAM (Cluster 2) Address from FE51 0000$_H$ to FE51 FFFC$_H$ and 4 bit of LSB is 8$_H$	√	√	—	—	Reserved	—	—	—	—
24	MBIST1FTAG6[24]	Cluster RAM (Cluster 2) Address from FE51 0000$_H$ to FE51 FFFC$_H$ and 4 bit of LSB is C$_H$	√	√	—	—	Reserved	—	—	—	—
23	MBIST1FTAG6[23]	Cluster RAM (Cluster 2) Address from FE52 0000$_H$ to FE52 FFFC$_H$ and 4 bit of LSB is 0$_H$	√	√	—	—	Reserved	—	—	—	—
22	MBIST1FTAG6[22]	Cluster RAM (Cluster 2) Address from FE52 0000$_H$ to FE52 FFFC$_H$ and 4 bit of LSB is 4$_H$	√	√	—	—	Reserved	—	—	—	—
21	MBIST1FTAG6[21]	Cluster RAM (Cluster 2) Address from FE52 0000$_H$ to FE52 FFFC$_H$ and 4 bit of LSB is 8$_H$	√	√	—	—	Reserved	—	—	—	—
20	MBIST1FTAG6[20]	Cluster RAM (Cluster 2) Address from FE52 0000$_H$ to FE52 FFFC$_H$ and 4 bit of LSB is C$_H$	√	√	—	—	Reserved	—	—	—	—
19	MBIST1FTAG6[19]	Cluster RAM (Cluster 2) Address from FE53 0000$_H$ to FE53 FFFC$_H$ and 4 bit of LSB is 0$_H$	√	√	—	—	Reserved	—	—	—	—
18	MBIST1FTAG6[18]	Cluster RAM (Cluster 2) Address from FE53 0000$_H$ to FE53 FFFC$_H$ and 4 bit of LSB is 4$_H$	√	√	—	—	Reserved	—	—	—	—
17	MBIST1FTAG6[17]	Cluster RAM (Cluster 2) Address from FE53 0000$_H$ to FE53 FFFC$_H$ and 4 bit of LSB is 8$_H$	√	√	—	—	Reserved	—	—	—	—
16	MBIST1FTAG6[16]	Cluster RAM (Cluster 2) Address from FE53 0000$_H$ to FE53 FFFC$_H$ and 4 bit of LSB is C$_H$	√	√	—	—	Reserved	—	—	—	—
15	MBIST1FTAG6[15]	Cluster RAM (Cluster 2) Address from FE54 0000$_H$ to FE54 FFFC$_H$ and 4 bit of LSB is 0$_H$	√	√	—	—	Reserved	—	—	—	—
14	MBIST1FTAG6[14]	Cluster RAM (Cluster 2) Address from FE54 0000$_H$ to FE54 FFFC$_H$ and 4 bit of LSB is 4$_H$	√	√	—	—	Reserved	—	—	—	—
13	MBIST1FTAG6[13]	Cluster RAM (Cluster 2) Address from FE54 0000$_H$ to FE54 FFFC$_H$ and 4 bit of LSB is 8$_H$	√	√	—	—	Reserved	—	—	—	—
12	MBIST1FTAG6[12]	Cluster RAM (Cluster 2) Address from FE54 0000$_H$ to FE54 FFFC$_H$ and 4 bit of LSB is C$_H$	√	√	—	—	Reserved	—	—	—	—

Table 44.463 Mapping of MBIST1FTAG6 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG6[11]	Cluster RAM (Cluster 2) Address from FE55 0000_H to FE55 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
10	MBIST1FTAG6[10]	Cluster RAM (Cluster 2) Address from FE55 0000_H to FE55 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
9	MBIST1FTAG6[9]	Cluster RAM (Cluster 2) Address from FE55 0000_H to FE55 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
8	MBIST1FTAG6[8]	Cluster RAM (Cluster 2) Address from FE55 0000_H to FE55 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
7	MBIST1FTAG6[7]	Cluster RAM (Cluster 2) Address from FE56 0000_H to FE56 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
6	MBIST1FTAG6[6]	Cluster RAM (Cluster 2) Address from FE56 0000_H to FE56 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
5	MBIST1FTAG6[5]	Cluster RAM (Cluster 2) Address from FE56 0000_H to FE56 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
4	MBIST1FTAG6[4]	Cluster RAM (Cluster 2) Address from FE56 0000_H to FE56 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
3	MBIST1FTAG6[3]	Cluster RAM (Cluster 2) Address from FE57 0000_H to FE57 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
2	MBIST1FTAG6[2]	Cluster RAM (Cluster 2) Address from FE57 0000_H to FE57 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
1	MBIST1FTAG6[1]	Cluster RAM (Cluster 2) Address from FE57 0000_H to FE57 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
0	MBIST1FTAG6[0]	Cluster RAM (Cluster 2) Address from FE57 0000_H to FE57 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—

44.6.2.21 MBIST1FTAG7 — Memory BIST1 FTAG Signature Value Register 7

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6044$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1FTAG7[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1FTAG7[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.464 MBIST1FTAG7 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1FTAG7[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.465, Mapping of MBIST1FTAG7 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

Table 44.465 Mapping of MBIST1FTAG7 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST1FTAG7[31]	Cluster RAM (Cluster 2) Address from FE58 0000_H to FE58 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
30	MBIST1FTAG7[30]	Cluster RAM (Cluster 2) Address from FE58 0000_H to FE58 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
29	MBIST1FTAG7[29]	Cluster RAM (Cluster 2) Address from FE58 0000_H to FE58 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
28	MBIST1FTAG7[28]	Cluster RAM (Cluster 2) Address from FE58 0000_H to FE58 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
27	MBIST1FTAG7[27]	Cluster RAM (Cluster 2) Address from FE59 0000_H to FE59 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
26	MBIST1FTAG7[26]	Cluster RAM (Cluster 2) Address from FE59 0000_H to FE59 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
25	MBIST1FTAG7[25]	Cluster RAM (Cluster 2) Address from FE59 0000_H to FE59 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
24	MBIST1FTAG7[24]	Cluster RAM (Cluster 2) Address from FE59 0000_H to FE59 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
23	MBIST1FTAG7[23]	Cluster RAM (Cluster 2) Address from FE5A 0000_H to FE5A $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
22	MBIST1FTAG7[22]	Cluster RAM (Cluster 2) Address from FE5A 0000_H to FE5A $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
21	MBIST1FTAG7[21]	Cluster RAM (Cluster 2) Address from FE5A 0000_H to FE5A $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
20	MBIST1FTAG7[20]	Cluster RAM (Cluster 2) Address from FE5A 0000_H to FE5A $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
19	MBIST1FTAG7[19]	Cluster RAM (Cluster 2) Address from FE5B 0000_H to FE5B $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
18	MBIST1FTAG7[18]	Cluster RAM (Cluster 2) Address from FE5B 0000_H to FE5B $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
17	MBIST1FTAG7[17]	Cluster RAM (Cluster 2) Address from FE5B 0000_H to FE5B $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
16	MBIST1FTAG7[16]	Cluster RAM (Cluster 2) Address from FE5B 0000_H to FE5B $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
15	MBIST1FTAG7[15]	Cluster RAM (Cluster 2) Address from FE5C 0000_H to FE5C $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
14	MBIST1FTAG7[14]	Cluster RAM (Cluster 2) Address from FE5C 0000_H to FE5C $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
13	MBIST1FTAG7[13]	Cluster RAM (Cluster 2) Address from FE5C 0000_H to FE5C $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
12	MBIST1FTAG7[12]	Cluster RAM (Cluster 2) Address from FE5C 0000_H to FE5C $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—

Table 44.465 Mapping of MBIST1FTAG7 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 to 3	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
11	MBIST1FTAG7[11]	Cluster RAM (Cluster 2) Address from FE5D 0000_H to FE5D $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
10	MBIST1FTAG7[10]	Cluster RAM (Cluster 2) Address from FE5D 0000_H to FE5D $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
9	MBIST1FTAG7[9]	Cluster RAM (Cluster 2) Address from FE5D 0000_H to FE5D $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
8	MBIST1FTAG7[8]	Cluster RAM (Cluster 2) Address from FE5D 0000_H to FE5D $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
7	MBIST1FTAG7[7]	Cluster RAM (Cluster 2) Address from FE5E 0000_H to FE5E $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
6	MBIST1FTAG7[6]	Cluster RAM (Cluster 2) Address from FE5E 0000_H to FE5E $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
5	MBIST1FTAG7[5]	Cluster RAM (Cluster 2) Address from FE5E 0000_H to FE5E $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
4	MBIST1FTAG7[4]	Cluster RAM (Cluster 2) Address from FE5E 0000_H to FE5E $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
3	MBIST1FTAG7[3]	Cluster RAM (Cluster 2) Address from FE5F 0000_H to FE5F $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
2	MBIST1FTAG7[2]	Cluster RAM (Cluster 2) Address from FE5F 0000_H to FE5F $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
1	MBIST1FTAG7[1]	Cluster RAM (Cluster 2) Address from FE5F 0000_H to FE5F $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
0	MBIST1FTAG7[0]	Cluster RAM (Cluster 2) Address from FE5F 0000_H to FE5F $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—

44.6.2.22 MBIST2FTAG0 — Memory BIST2 FTAG Signature Value Register 0

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6048$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2FTAG0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2FTAG0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.466 MBIST2FTAG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2FTAG0[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.467, Mapping of MBIST2FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

Table 44.467 Mapping of MBIST2FTAG0 Register

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 and 2	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 3 and 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST2FTAG0[31]	FLXA0 MRAM	√	√	√	√	Reserved	—	—	—	—
30	MBIST2FTAG0[30]	FLXA0 TBFRAM A/B	√	√	√	√	Reserved	—	—	—	—
29	MBIST2FTAG0[29]	Reserved	—	—	—	—	Reserved	—	—	—	—
28	MBIST2FTAG0[28]	Reserved	—	—	—	—	Reserved	—	—	—	—
27	MBIST2FTAG0[27]	FLXA1 MRAM	√	√	√	—	Reserved	—	—	—	—
26	MBIST2FTAG0[26]	FLXA1 TBFRAM A/B	√	√	√	—	Reserved	—	—	—	—
25	MBIST2FTAG0[25]	Reserved	—	—	—	—	Reserved	—	—	—	—
24	MBIST2FTAG0[24]	Reserved	—	—	—	—	Reserved	—	—	—	—
23	MBIST2FTAG0[23]	ETNB0 Transmit FIFO	√	√	√	√	Reserved	—	—	—	—
22	MBIST2FTAG0[22]	ETNB0 Receive FIFO	√	√	√	√	Reserved	—	—	—	—
21	MBIST2FTAG0[21]	ETNB1 Transmit FIFO	√	√	√	—	Reserved	—	—	—	—
20	MBIST2FTAG0[20]	ETNB1 Receive FIFO	√	√	√	—	Reserved	—	—	—	—
19	MBIST2FTAG0[19]	Reserved	—	—	—	—	Reserved	—	—	—	—
18	MBIST2FTAG0[18]	Reserved	—	—	—	—	Reserved	—	—	—	—
17	MBIST2FTAG0[17]	MMCA0 RAM A/B	√	√	√	√	Reserved	—	—	—	—
16	MBIST2FTAG0[16]	Reserved	—	—	—	—	Reserved	—	—	—	—
15	MBIST2FTAG0[15]	MSPI0/1/2/3 RAM	√	√	√	√	Reserved	—	—	—	—
14	MBIST2FTAG0[14]	MSPI4/5/6/7 RAM	√	√	√	√*2	Reserved	—	—	—	—
13	MBIST2FTAG0[13]	MSPI8/9 RAM	√	√	√*1	—	Reserved	—	—	—	—
12	MBIST2FTAG0[12]	Reserved	—	—	—	—	Reserved	—	—	—	—
11	MBIST2FTAG0[11]	Reserved	—	—	—	—	Reserved	—	—	—	—
10	MBIST2FTAG0[10]	Reserved	—	—	—	—	Reserved	—	—	—	—
9	MBIST2FTAG0[9]	Reserved	—	—	—	—	Reserved	—	—	—	—
8	MBIST2FTAG0[8]	Reserved	—	—	—	—	Reserved	—	—	—	—
7	MBIST2FTAG0[7]	GTM MCS0/1 RAM 0	√	√	√	√	Reserved	—	—	—	—
6	MBIST2FTAG0[6]	GTM MCS2/3 RAM 0	√	√	√	√	Reserved	—	—	—	—
5	MBIST2FTAG0[5]	GTM MCS0/1 RAM 1	√	√	√	√	Reserved	—	—	—	—
4	MBIST2FTAG0[4]	GTM MCS2/3 RAM 1	√	√	√	√	Reserved	—	—	—	—
3	MBIST2FTAG0[3]	Reserved	—	—	—	—	Reserved	—	—	—	—
2	MBIST2FTAG0[2]	Reserved	—	—	—	—	Reserved	—	—	—	—
1	MBIST2FTAG0[1]	Reserved	—	—	—	—	Reserved	—	—	—	—
0	MBIST2FTAG0[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

Note 1. MSPI9 is not implemented in RH850/U2A8.

Note 2. MSPI6/7 are not implemented in RH850/U2A6.

44.6.2.23 MBIST2FTAG1 — Memory BIST2 FTAG Signature Value Register 1

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 604C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2FTAG1[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2FTAG1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.468 MBIST2FTAG1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2FTAG1[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.469, Mapping of MBIST2FTAG1 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

Table 44.469 Mapping of MBIST2FTAG1 Register

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 and 2	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 3 and 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST2FTAG1[31]	RS-CANFD0 AFLRAM 0	√	√	√	√	Reserved	—	—	—	—
30	MBIST2FTAG1[30]	RS-CANFD0 AFLRAM 0	√	—	—	—	Reserved	—	—	—	—
29	MBIST2FTAG1[29]	RS-CANFD0 AFLRAM 1	√	√	√	√	Reserved	—	—	—	—
28	MBIST2FTAG1[28]	RS-CANFD0 AFLRAM 1	√	—	—	—	Reserved	—	—	—	—
27	MBIST2FTAG1[27]	RS-CANFD0 MRAM	√	√	√	√	Reserved	—	—	—	—
26	MBIST2FTAG1[26]	RS-CANFD0 MRAM	√	√	√	√	Reserved	—	—	—	—
25	MBIST2FTAG1[25]	RS-CANFD0 MRAM	√	√	√	√	Reserved	—	—	—	—
24	MBIST2FTAG1[24]	RS-CANFD0 MRAM	√	√	√	√	Reserved	—	—	—	—
23	MBIST2FTAG1[23]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
22	MBIST2FTAG1[22]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
21	MBIST2FTAG1[21]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
20	MBIST2FTAG1[20]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
19	MBIST2FTAG1[19]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
18	MBIST2FTAG1[18]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
17	MBIST2FTAG1[17]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
16	MBIST2FTAG1[16]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
15	MBIST2FTAG1[15]	RS-CANFD0 MRAM	√	√	√	—	Reserved	—	—	—	—
14	MBIST2FTAG1[14]	Reserved	—	—	—	—	Reserved	—	—	—	—
13	MBIST2FTAG1[13]	Reserved	—	—	—	—	Reserved	—	—	—	—
12	MBIST2FTAG1[12]	Reserved	—	—	—	—	Reserved	—	—	—	—
11	MBIST2FTAG1[11]	Reserved	—	—	—	—	Reserved	—	—	—	—
10	MBIST2FTAG1[10]	Reserved	—	—	—	—	Reserved	—	—	—	—
9	MBIST2FTAG1[9]	Reserved	—	—	—	—	Reserved	—	—	—	—
8	MBIST2FTAG1[8]	Reserved	—	—	—	—	Reserved	—	—	—	—
7	MBIST2FTAG1[7]	Reserved	—	—	—	—	Reserved	—	—	—	—
6	MBIST2FTAG1[6]	Reserved	—	—	—	—	Reserved	—	—	—	—
5	MBIST2FTAG1[5]	Reserved	—	—	—	—	Reserved	—	—	—	—
4	MBIST2FTAG1[4]	Reserved	—	—	—	—	Reserved	—	—	—	—
3	MBIST2FTAG1[3]	Reserved	—	—	—	—	Reserved	—	—	—	—
2	MBIST2FTAG1[2]	Reserved	—	—	—	—	Reserved	—	—	—	—
1	MBIST2FTAG1[1]	Reserved	—	—	—	—	Reserved	—	—	—	—
0	MBIST2FTAG1[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

44.6.2.24 MBIST2FTAG2 — Memory BIST2 FTAG Signature Value Register 2

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6050$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2FTAG2[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2FTAG2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.470 MBIST2FTAG2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2FTAG2[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.471, Mapping of MBIST2FTAG2 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

Table 44.471 Mapping of MBIST2FTAG2 Register

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1 and 2	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 3 and 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST2FTAG2[31]	RS-CANFD1 AFLRAM 0	√	√	√	√	Reserved	—	—	—	—
30	MBIST2FTAG2[30]	RS-CANFD1 AFLRAM 0	√	—	—	—	Reserved	—	—	—	—
29	MBIST2FTAG2[29]	RS-CANFD1 AFLRAM 1	√	√	√	√	Reserved	—	—	—	—
28	MBIST2FTAG2[28]	RS-CANFD1 AFLRAM 1	√	—	—	—	Reserved	—	—	—	—
27	MBIST2FTAG2[27]	RS-CANFD1 MRAM	√	√	√	√	Reserved	—	—	—	—
26	MBIST2FTAG2[26]	RS-CANFD1 MRAM	√	√	√	√	Reserved	—	—	—	—
25	MBIST2FTAG2[25]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
24	MBIST2FTAG2[24]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
23	MBIST2FTAG2[23]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
22	MBIST2FTAG2[22]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
21	MBIST2FTAG2[21]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
20	MBIST2FTAG2[20]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
19	MBIST2FTAG2[19]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
18	MBIST2FTAG2[18]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
17	MBIST2FTAG2[17]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
16	MBIST2FTAG2[16]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
15	MBIST2FTAG2[15]	RS-CANFD1 MRAM	√	√	√	—	Reserved	—	—	—	—
14	MBIST2FTAG2[14]	Reserved	—	—	—	—	Reserved	—	—	—	—
13	MBIST2FTAG2[13]	Reserved	—	—	—	—	Reserved	—	—	—	—
12	MBIST2FTAG2[12]	Reserved	—	—	—	—	Reserved	—	—	—	—
11	MBIST2FTAG2[11]	Reserved	—	—	—	—	Reserved	—	—	—	—
10	MBIST2FTAG2[10]	Reserved	—	—	—	—	Reserved	—	—	—	—
9	MBIST2FTAG2[9]	Reserved	—	—	—	—	Reserved	—	—	—	—
8	MBIST2FTAG2[8]	Reserved	—	—	—	—	Reserved	—	—	—	—
7	MBIST2FTAG2[7]	Reserved	—	—	—	—	Reserved	—	—	—	—
6	MBIST2FTAG2[6]	Reserved	—	—	—	—	Reserved	—	—	—	—
5	MBIST2FTAG2[5]	Reserved	—	—	—	—	Reserved	—	—	—	—
4	MBIST2FTAG2[4]	Reserved	—	—	—	—	Reserved	—	—	—	—
3	MBIST2FTAG2[3]	Reserved	—	—	—	—	Reserved	—	—	—	—
2	MBIST2FTAG2[2]	Reserved	—	—	—	—	Reserved	—	—	—	—
1	MBIST2FTAG2[1]	Reserved	—	—	—	—	Reserved	—	—	—	—
0	MBIST2FTAG2[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

44.6.2.25 MBIST3FTAG0 — Memory BIST3 FTAG signature value register 0

This register indicates the Memory BIST status of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6068$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST3FTAG0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST3FTAG0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.472 MBIST3FTAG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST3FTAG0[31:0]	The state of self-diagnostic MBIST*1 0: MBIST status is PASS. 1: MBIST status is FAIL. Refer to **Table 44.473, Mapping of MBIST3FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without MBIST scenario 1:	FFFF FFFF$_H$
BIST is not executed with MBIST scenario 1:	FFFF FFFF$_H$
BIST is executed without MBIST scenario 1:	FFFF FFFF$_H$
BIST is executed with MBIST scenario 1:	The state of Memory BIST result for each RAM

Table 44.473 Mapping of MBIST3FTAG0 Register (1/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 2 to 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
31	MBIST3FTAG0[31]	Cluster RAM (Cluster 3) Address from FE80 0000_H to FE80 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	Reserved	—	—	—	—
30	MBIST3FTAG0[30]	Cluster RAM (Cluster 3) Address from FE80 0000_H to FE80 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
29	MBIST3FTAG0[29]	Cluster RAM (Cluster 3) Address from FE80 0000_H to FE80 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
28	MBIST3FTAG0[28]	Cluster RAM (Cluster 3) Address from FE80 0000_H to FE80 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—
27	MBIST3FTAG0[27]	Cluster RAM (Cluster 3) Address from FE81 0000_H to FE81 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	√	√	Reserved	—	—	—	—
26	MBIST3FTAG0[26]	Cluster RAM (Cluster 3) Address from FE81 0000_H to FE81 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	√	√	Reserved	—	—	—	—
25	MBIST3FTAG0[25]	Cluster RAM (Cluster 3) Address from FE81 0000_H to FE81 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	√	√	Reserved	—	—	—	—
24	MBIST3FTAG0[24]	Cluster RAM (Cluster 3) Address from FE81 0000_H to FE81 $FFFC_H$ and 4 bit of LSB is C_H	√	√	√	√	Reserved	—	—	—	—
23	MBIST3FTAG0[23]	Cluster RAM (Cluster 3) Address from FE82 0000_H to FE82 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
22	MBIST3FTAG0[22]	Cluster RAM (Cluster 3) Address from FE82 0000_H to FE82 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
21	MBIST3FTAG0[21]	Cluster RAM (Cluster 3) Address from FE82 0000_H to FE82 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
20	MBIST3FTAG0[20]	Cluster RAM (Cluster 3) Address from FE82 0000_H to FE82 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
19	MBIST3FTAG0[19]	Cluster RAM (Cluster 3) Address from FE83 0000_H to FE83 $FFFC_H$ and 4 bit of LSB is 0_H	√	√	—	—	Reserved	—	—	—	—
18	MBIST3FTAG0[18]	Cluster RAM (Cluster 3) Address from FE83 0000_H to FE83 $FFFC_H$ and 4 bit of LSB is 4_H	√	√	—	—	Reserved	—	—	—	—
17	MBIST3FTAG0[17]	Cluster RAM (Cluster 3) Address from FE83 0000_H to FE83 $FFFC_H$ and 4 bit of LSB is 8_H	√	√	—	—	Reserved	—	—	—	—
16	MBIST3FTAG0[16]	Cluster RAM (Cluster 3) Address from FE83 0000_H to FE83 $FFFC_H$ and 4 bit of LSB is C_H	√	√	—	—	Reserved	—	—	—	—
15	MBIST3FTAG0[15]	Reserved	—	—	—	—	Reserved	—	—	—	—
14	MBIST3FTAG0[14]	Reserved	—	—	—	—	Reserved	—	—	—	—
13	MBIST3FTAG0[13]	Reserved	—	—	—	—	Reserved	—	—	—	—
12	MBIST3FTAG0[12]	Reserved	—	—	—	—	Reserved	—	—	—	—
11	MBIST3FTAG0[11]	Reserved	—	—	—	—	Reserved	—	—	—	—
10	MBIST3FTAG0[10]	Reserved	—	—	—	—	Reserved	—	—	—	—
9	MBIST3FTAG0[9]	Reserved	—	—	—	—	Reserved	—	—	—	—
8	MBIST3FTAG0[8]	Reserved	—	—	—	—	Reserved	—	—	—	—

Table 44.473 Mapping of MBIST3FTAG0 Register (2/2)

Bit Position	Bit Name	Memory assignment for the result of MBIST scenario 1	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)	Memory assignment for the result of MBIST scenario 2 to 4	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516/373/292 pins)	RH850/U2A8 (373/292 pins)	RH850/U2A6 (292/176/156/144 pins)
7	MBIST3FTAG0[7]	Reserved	—	—	—	—	Reserved	—	—	—	—
6	MBIST3FTAG0[6]	Reserved	—	—	—	—	Reserved	—	—	—	—
5	MBIST3FTAG0[5]	Reserved	—	—	—	—	Reserved	—	—	—	—
4	MBIST3FTAG0[4]	Reserved	—	—	—	—	Reserved	—	—	—	—
3	MBIST3FTAG0[3]	Reserved	—	—	—	—	Reserved	—	—	—	—
2	MBIST3FTAG0[2]	Reserved	—	—	—	—	Reserved	—	—	—	—
1	MBIST3FTAG0[1]	Reserved	—	—	—	—	Reserved	—	—	—	—
0	MBIST3FTAG0[0]	Reserved	—	—	—	—	Reserved	—	—	—	—

44.6.2.26 MBIST1ECC0 — Memory BIST1 ECC result register 0

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6088$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.474 MBIST1ECC0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC0[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG0 shows "0_B", this register shows no error. When MBIST1FTAG0 shows "1_B", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG0 register. Refer to **Table 44.451, Mapping of MBIST1FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.27 MBIST1ECC1 — Memory BIST1 ECC result register 1

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 608C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC1[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.475 MBIST1ECC1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC1[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG1 shows "0$_B$", this register shows no error. When MBIST1FTAG1 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG1 register. Refer to **Table 44.453, Mapping of MBIST1FTAG1 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.28 MBIST1ECC2 — Memory BIST1 ECC result register 2

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6090$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC2[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.476 MBIST1ECC2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC2[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG2 shows "0_B", this register shows no error. When MBIST1FTAG2 shows "1_B", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG2 register. Refer to **Table 44.455, Mapping of MBIST1FTAG2 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.29 MBIST1ECC3 — Memory BIST1 ECC result register 3

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6094$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC3[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC3[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.477 MBIST1ECC3 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC3[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG3 shows "0_B", this register shows no error. When MBIST1FTAG3 shows "1_B", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG3 register. Refer to **Table 44.457, Mapping of MBIST1FTAG3 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.30 MBIST1ECC4 — Memory BIST1 ECC result register 4

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 6098$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC4[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC4[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.478 MBIST1ECC4 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC4[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG4 shows "0_B", this register shows no error. When MBIST1FTAG4 shows "1_B", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG4 register. Refer to **Table 44.459, Mapping of MBIST1FTAG4 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.31 MBIST1ECC5 — Memory BIST1 ECC result register 5

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 609C$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC5[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC5[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.479 MBIST1ECC5 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC5[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG5 shows "0$_B$", this register shows no error. When MBIST1FTAG5 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG5 register. Refer to **Table 44.461, Mapping of MBIST1FTAG5 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.32 MBIST1ECC6 — Memory BIST1 ECC result register 6

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60A0$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC6[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC6[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.480 MBIST1ECC6 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC6[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG6 shows "0$_B$", this register shows no error. When MBIST1FTAG6 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG6 register. Refer to **Table 44.463, Mapping of MBIST1FTAG6 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.33 MBIST1ECC7 — Memory BIST1 ECC result register 7

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60A4$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST1ECC7[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST1ECC7[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.481 MBIST1ECC7 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST1ECC7[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST1FTAG7 shows "0_B", this register shows no error. When MBIST1FTAG7 shows "1_B", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST1FTAG7 register. Refer to **Table 44.465, Mapping of MBIST1FTAG7 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with any of MBIST scenario:	The state of Memory BIST result for each RAM

44.6.2.34 MBIST2ECC0 — Memory BIST2 ECC result register 0

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60A8$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2ECC0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2ECC0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.482 MBIST2ECC0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2ECC0[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST2FTAG0 shows "0$_B$", this register shows no error. When MBIST2FTAG0 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST2FTAG0 register. Refer to **Table 44.467, Mapping of MBIST2FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

44.6.2.35 MBIST2ECC1 — Memory BIST2 ECC result register 1

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60AC$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2ECC1[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2ECC1[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.483 MBIST2ECC1 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2ECC1[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST2FTAG1 shows "0$_B$", this register shows no error. When MBIST2FTAG1 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST2FTAG1 register. Refer to **Table 44.469, Mapping of MBIST2FTAG1 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

44.6.2.36 MBIST2ECC2 — Memory BIST2 ECC result register 2

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60B0$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST2ECC2[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST2ECC2[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.484 MBIST2ECC2 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST2ECC2[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST2FTAG2 shows "0$_B$", this register shows no error. When MBIST2FTAG2 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST2FTAG2 register. Refer to **Table 44.471, Mapping of MBIST2FTAG2 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is not executed with any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed without any of MBIST scenario:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 3 or 4:	FFFF FFFF$_H$
BIST is executed with either of MBIST scenario 1 or 2:	The state of Memory BIST result for each RAM

44.6.2.37 MBIST3ECC0 — Memory BIST3 ECC result register 0

This register indicates the Memory BIST ECC check result of each RAM group (bridge). This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 60C8$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBIST3ECC0[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIST3ECC0[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.485 MBIST3ECC0 Register Contents

Bit Position	Bit Name	Function
31 to 0	MBIST3ECC0[31:0]	The state of self-diagnostic MBIST*1 0: No error or ECC is correctable. When MBIST3FTAG0 shows "0$_B$", this register shows no error. When MBIST3FTAG0 shows "1$_B$", this register shows correctable error. 1: ECC is uncorrectable. The mapping of this register for each RAM has same bit assign as MBIST3FTAG0 register. Refer to **Table 44.473, Mapping of MBIST3FTAG0 Register** for mapping of each RAM.

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	FFFF FFFF$_H$
BIST is not executed without MBIST scenario 1:	FFFF FFFF$_H$
BIST is not executed with MBIST scenario 1:	FFFF FFFF$_H$
BIST is executed without MBIST scenario 1:	FFFF FFFF$_H$
BIST is executed with MBIST scenario 1:	The state of Memory BIST result for each RAM

44.6.2.38 BSEQ0ST — BIST Sequencer Status Register

This register indicates the state of the BIST sequencer. This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 4400$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BISTEND	CMPERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.486 BSEQ0ST Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	BISTEND	0: The BIST sequence did not finish within the time limit. 1: The BIST sequence finished within the time limit.
0	CMPERR	0: BIST normal end 1: BIST abnormal end

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0000 0001$_H$
BIST is not executed:	0000 0002$_H$
BIST is executed:	The state of BIST sequencer for the BIST result

44.6.2.39 BSEQ0STB — BIST Sequencer Inverted Status Register

This register indicates the inverted state of the BIST sequencer. This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 4404$_H$

Value after reset: *1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BISTENDB	CMPERRB
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.487 BSEQ0STB Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	BISTENDB	0: The BIST sequence finished within the time limit. 1: The BIST sequence did not finish within the time limit.
0	CMPERRB	0: BIST abnormal end 1: BIST normal end

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.

BIST is skipped:	0000 0002$_H$
BIST is not executed:	0000 0001$_H$
BIST is executed:	The inverted state of BIST sequencer for the BIST result

44.6.2.40 BISTST — BIST Result Register

This register indicates the result of the BIST. This register is automatically updated by the BIST result after BIST execution.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FF9A 4408$_H$

Value after reset: *1*2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	MBIST3ST	MBIST2ST	MBIST1ST	LBIST2ST	LBIST1ST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	*2	*2	*2	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 44.488 BISTST Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4	MBIST3ST	0: MBIST3 passed successfully 1: MBIST3 has detected an error
3	MBIST2ST	0: MBIST2 passed successfully 1: MBIST2 has detected an error
2	MBIST1ST	0: MBIST1 passed successfully 1: MBIST1 has detected an error
1	LBIST2ST	0: LBIST2 passed successfully 1: LBIST2 has detected an error
0	LBIST1ST	0: LBIST1 passed successfully 1: LBIST1 has detected an error

Note 1. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.
BIST is skipped:
LBIST1ST and LBIST2ST show "11_B"
BIST is not executed with any of LBIST scenario:
LBIST1ST and LBIST2ST show "11_B"
BIST is executed with any of LBIST scenario 9 or 11 to 20:
The BIST result is stored in LBIST1ST (LBIST2ST shows "1_B")
BIST is executed with any of LBIST scenario 1 to 3, 5 to 7 or 10:
The BIST result is stored in LBIST1ST and LBIST2ST

Note 2. The value depends on the BIST execution condition. For details about the BIST execution condition, see **Section 44.6.3.1, BIST Execution Condition**.
BIST is skipped:
MBIST1ST, MBIST2ST and MBIST3ST show "111_B"
BIST is not executed with any of MBIST scenario:
MBIST1ST, MBIST2ST and MBIST3ST show "111_B"
BIST is executed with any of MBIST scenario 3 or 4:
The BIST result is stored in MBIST1ST (MBIST2ST and MBIST3ST show "11_B")
BIST is executed with any of MBIST scenario 2:
The BIST result is stored in MBIST1ST and MBIST2ST (MBIST3ST shows "1_B")
BIST is executed with any of MBIST scenario 1:
The BIST result is stored in MBIST1ST, MBIST2ST and MBIST3ST

44.6.2.41 BSEQ0SEL — BIST Scenario Select Register

This register is used to select the execution scenario of BIST at the next System Reset 2 or DeepSTOP Reset occurrence.

Access: This register can be read or written in 32-bit units.

Address: FF9A 440C$_H$

Value after reset: 0000 0030$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PARAMSEL	—	—	—	—	—	—	—	—	—	HWTESTSET[1:0]		HWLBISTSEL[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.489 BSEQ0SEL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PARAMSEL	Selects BIST parameters (HWLBISTSEL[1:0], HWTESTSET[1:0]) of BIST which will be executed during next System Reset 2 or DeepSTOP Reset. 0: Parameters specified by Flash option byte.*1 (default) 1: Parameters specified by BSEQ0SEL registers.*2
14 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 4	HWTESTSET[1:0]	BIST selection 00_B: Prohibited 01_B: LBIST ONLY 10_B: MBIST ONLY 11_B: LBIST AND MBIST
3 to 0	HWLBISTSEL[3:0]	Refer to **Table 44.490** and **Table 44.491**.

Note 1. For details, see **Section 51, Flash Memory**.

Note 2. This bit has to be set to "1_B" when Standby Resume BIST will be executed at next DeepSTOP Reset.

Table 44.490 BIST scenario selection at the next System Reset 2

HWLBISTSEL	LBIST scenario	MBIST scenario*2	Run time*1		
			LBIST only	MBIST only	LBIST and MBIST
0000_B	1	1	5.1ms	6.5ms	11.3ms
0001_B	2	1	10.2ms	6.5ms	16.4ms
0010_B	3	1	15.3ms	6.5ms	21.5ms
Other	Reserved	—	—	—	—

Note 1. Run time is calculated by the typical frequency of the CLK_HSIOSC. For details, see **Section 55, Electrical Characteristics**.

Note 2. For details of RAM group mapping for each executed MBIST scenario, see **Section 44.6.2.14, MBIST1FTAG0 — Memory BIST1 FTAG signature value register 0** to **Section 44.6.2.25, MBIST3FTAG0 — Memory BIST3 FTAG signature value register 0**.

Table 44.491 BIST scenario selection at the next DeepSTOP Reset

HWLBISTSEL	LBIST scenario	MBIST scenario*2	Run time*1		
			LBIST only	MBIST only	LBIST and MBIST
0000_B	5	2	5.1ms	6.5ms	11.3ms
0001_B	6	2	10.2ms	6.5ms	16.4ms
0010_B	7	2	15.3ms	6.5ms	21.5ms
0011_B	Reserved	—	—	—	—
0100_B	9	3	2.04ms	3.5ms	5.24ms
0101_B	10	4	2.04ms	3.5ms	5.24ms
0110_B	11	3	2.04ms	3.5ms	5.24ms
0111_B	12	4	2.04ms	3.5ms	5.24ms
1000_B	13	3	2.04ms	3.5ms	5.24ms
1001_B	14	4	2.04ms	3.5ms	5.24ms
1010_B	15	3	2.04ms	3.5ms	5.24ms
1011_B	16	4	2.04ms	3.5ms	5.24ms
1100_B	17	3	2.04ms	3.5ms	5.24ms
1101_B	18	4	2.04ms	3.5ms	5.24ms
1110_B	19	3	2.04ms	3.5ms	5.24ms
1111_B	20	4	2.04ms	3.5ms	5.24ms

Note 1. Run time is calculated by the typical frequency of the CLK_HSIOSC. For details, see **Section 55, Electrical Characteristics**.

Note 2. For details of RAM group mapping for each executed MBIST scenario, see **Section 44.6.2.14, MBIST1FTAG0 — Memory BIST1 FTAG signature value register 0** to **Section 44.6.2.25, MBIST3FTAG0 — Memory BIST3 FTAG signature value register 0**.

Table 44.492 The relationship of executed scenario and target of ASIL level*1

Executed scenario	Target ASIL level
LBIST scenario 1	ASIL-B
LBIST scenario 2	ASIL-C
LBIST scenario 3	ASIL-D
LBIST scenario 5	ASIL-B
LBIST scenario 6	ASIL-C
LBIST scenario 7	ASIL-D
LBIST scenario 9 to LBIST scenario 12*2	ASIL-B
LBIST scenario 9 to LBIST scenario 16*2	ASIL-C
LBIST scenario 9 to LBIST scenario 20*2	ASIL-D

Note 1. ISO26262 compliance for LFM (Latent Fault Metric) by LBIST.

Note 2. All scenarios for target ASIL level must be executed and the result must be confirmed within the fault detection interval for latent fault.

44.6.2.42 BSEQ0CTL — BIST Skip Control Register for FBIST0

This register is used to control the BIST execution at the next System Reset 2 occurrence.

Access: This register can be read or written in 32-bit units.

Address: FF98 8400$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HWBISTSEL	HWBISTEXE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 44.493 BSEQ0CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	HWBISTSEL	Selects flash option byte value or register value for the BIST execution setting. 0: BIST is executed according to Flash Option Byte*1 value. 1: BIST is executed according to HWBISTEXE value.
0	HWBISTEXE	This bit selects if BIST is skipped or not at the next system reset 2 0: BIST is skipped at the next system reset 2 1: BIST is executed at the next system reset 2

Note 1. For details, see **Section 51, Flash Memory**

44.6.2.43 BSEQ1CTL — BIST Skip Control Register for FBIST1

This register is used to control the BIST execution at the next DeepSTOP Reset occurrence for FBIST1.

Access: This register can be read or written in 32-bit units.

Address: FF98 8404$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HWBISTEXE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.494 BSEQ1CTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	HWBISTEXE	This bit selects if BIST is skipped or not at the next DeepSTOP Reset 0: BIST is skipped at the next DeepSTOP Reset 1: BIST is executed at the next DeepSTOP Reset This bit is automatically set to 0 after FBIST1 has finished.

44.6.2.44 BSEQ2CTL — BIST Skip Control Register for FBIST2

This register is used to control the BIST execution at the next DeepSTOP Reset occurrence for FBIST2.

Access: This register can be read or written in 32-bit units.

Address: FF98 8408$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HWBISTEXE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 44.495 BSEQ2CTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	HWBISTEXE	This bit selects if BIST is skipped or not at the next DeepSTOP Reset 0: BIST is skipped at the next DeepSTOP Reset 1: BIST is executed at the next DeepSTOP Reset This bit is automatically set to 0 after FBIST2 has finished.

44.6.3 Functions

44.6.3.1 BIST Execution Condition

The BIST is executed during reset sequence of Power On Reset, System Reset 1, System Reset 2 and DeepSTOP Reset. The following tables explain the conditions of BIST execution.

Table 44.496 BIST Execution in Power On Reset, System Reset 1 for FBIST0

Conditions			
Reset Category	**$\overline{\text{TRST}}$ pin**	**HWBIST (Flash option*1)**	**BIST Execution during Reset Sequence**
Power On Reset System Reset 1	High	0	BIST is skipped. Reset is released without waiting for the selected BIST scenario time.
		1	BIST is not executed. Reset is released after the expiration of the selected BIST scenario time*1.
	Low	0	BIST is skipped. Reset is released without waiting for the selected BIST scenario time.
		1	BIST is executed with preset parameters*1. Reset is released after BIST is finished.

Note 1. For details, see **Section 51, Flash Memory**.

Table 44.497 BIST Execution in System Reset 2 for FBIST0

Reset Category	**$\overline{\text{TRST}}$ pin**	**BSEQ0CTL.HWBISTSEL**	**BIST Execution during Reset Sequence**
System Reset 2	High	0	HWBIST (Flash option*1) value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is not executed. Reset is released after the expiration of the selected BIST scenario time*1.
		1	BSEQ0CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is not executed. Reset is released after the expiration of the selected BIST scenario time.
	Low	0	HWBIST (Flash option*1) value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is executed with preset parameters*1. Reset is released after BIST is finished.
		1	BSEQ0CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is executed with preset parameters. Reset is released after BIST is finished.

Note 1. For details, see **Section 51, Flash Memory**.

Table 44.498 BIST Execution in DeepSTOP Reset for FBIST1

Reset Category	$\overline{\text{TRST}}$ pin	BIST Execution during Reset Sequence*1
DeepSTOP Reset	High	BSEQ1CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is not executed. Reset is released after the expiration of the selected BIST scenario time.
	Low	BSEQ1CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is executed with preset parameters. Reset is released after BIST is finished.

Note 1. MBIST3 related registers except MBISTREF3 register always show “BIST is skipped” because the RAM in AWO area is not target of Standby Resume BIST.

Table 44.499 BIST Execution in DeepSTOP Reset for FBIST2

Reset Category	$\overline{\text{TRST}}$ pin	BIST Execution during Reset Sequence*1
DeepSTOP Reset	High	BSEQ2CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is not executed. Reset is released after the expiration of the selected BIST scenario time.
	Low	BSEQ2CTL.HWBISTEXE value 0: BIST is skipped. Reset is released without waiting for the selected BIST scenario time. 1: BIST is executed with preset parameters. Reset is released after BIST is finished.

Note 1. MBIST3 related registers except MBISTREF3 register always show “BIST is skipped” because the RAM in AWO area is not target of Standby Resume BIST.

44.6.3.2 BIST Result Confirmation

After the BIST execution, read and compare the following register values to confirm whether the BIST execution completed normally without failure in case of executed both of LBIST and MBIST at Power On BIST.

- BSEQ0ST = $0000\ 0002_H$
- BSEQ0STB = $0000\ 0001_H$
- LBISTSIG1 = LBISTREF1
- LBISTSIG2 = LBISTREF2
- MBISTSIG1 = MBISTREF1
- MBISTSIG2 = MBISTREF2
- MBISTSIG3 = MBISTREF3
- BISTST = $0000\ 0000_H$

CAUTION

In RH850/U2A-EVA product, Logic BIST signature value register 1/2 and LBIST1/2ST bits of BIST Result Register do not show correct BIST result at Standby Resume BIST.

MBISTxECCy (x = 1 to 3, y = 0 to 7) registers are not supported in RH850/U2A-EVA product.

44.6.4 Operation

The target of LBIST and MBIST is difference between Power-On BIST and Standby-Resume BIST. When running BIST, any of LBIST ONLY, MBIST ONLY or both can be selected either by using flash option byte 3 (at Power-On BIST using Power On Reset or System Reset 1, or at Shut-Down BIST using System Reset 2) or by using BSEQ0SEL register (at Shut-Down BIST using System Reset 2 or Standby-Resume BIST using DeepSTOP Reset).

Table 44.500 BIST Scenario Select Condition

RESET	Source	BIST Scenario Select
Power On Reset	POC Reset	Flash option byte*1
System Reset 1	External Reset, Primary VMON Reset	Flash option byte*1
System Reset 2	Software System Reset, ECM Reset, WDTBA reset	Flash option byte*1 (BSEQ0SEL.PARAMSEL = 0) BSEQ0SEL register (BSEQ0SEL.PARAMSEL = 1)
DeepSTOP Reset	DeepSTOP Reset	BSEQ0SEL register (BSEQ0SEL.PARAMSEL = 1)

Note 1. For details, see **Section 51, Flash Memory**.

44.7 ECM

The ECM monitors various failure detection states in this microcontroller, and defines the operation to be carried out upon failure detection. For details of the ECM, see **Section 45, Error Control Module (ECM)**.

44.8 Voltage Monitor

The Voltage Monitor detects over and under voltage of the voltage domains, E0VCC, VCC, ISOVDD and AWOVDD. For details of the Voltage Monitor, see **Section 11, Power Supply Voltage Monitor**.

44.9 Clock Monitor

The Clock Monitor detects abnormal frequencies of internal clocks related to the safety of the device. For details of the Clock Monitor, see **Section 14, Clock Monitor (CLMA)**.

44.10 Data CRC Function

The Data CRC Function generates CRC to verify the data streams protected by CRC. For details of the Data CRC Function, see **Section 46, Data CRC Function K (KCRC)**.

Section 45 Error Control Module (ECM)

45.1 Features

45.1.1 Number of Units and Channels

This microcontroller has the following number of ECM unit.

Table 45.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1									
Name	ECM									

Table 45.2 Index

Index	RH850/U2A-EVA U2A16 mode (516 pins), RH850/U2A16 (516/373/292 pins)	RH850/U2A-EVA U2A8 mode (516 pins), RH850/U2A8 (373/292 pins), RH850/U2A6 (292/176/ 156/144 pins)	Meaning
i	i = 0 to 3	i = 0 to 1	Throughout this section, the core number is identified by the index “i”.
j	j = 0 to 10	j = 0 to 8	Throughout this section, the error number divided by 32 is identified by the index “j”.
k	k = 0 to 3	k = 0 to 1	Throughout this section, the external error input number is identified by the index “k”.
m	m = M, C		Throughout this section, the individual ECM Master and ECM Checker are identified by the index “m”.
n	n = 0 to 3	n = 0 to 1	Throughout this section, the Error trigger number is identified by the index “n”.

45.1.2 Register Base Addresses

ECM base addresses are listed in the following table.

ECM register addresses are given as offsets from the base addresses in general.

Table 45.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<ECMM_base>	FFD3 9000_H	P-Bus Group 3
<ECMC_base>	FFD3 $A000_H$	P-Bus Group 3
<ECM_base>	FFD3 8000_H	P-Bus Group 3

45.1.3 Clock Supply

Clock supply by and to ECM is listed in the following table.

Table 45.4 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
ECM	PCLK	CLK_HSB
	cntclk*1	CLK_ECMCNT

Note 1. cntclk is used for delay timer and clear mask timer logics.

45.1.4 Interrupt Requests and Error Notifications

ECM interrupt requests are listed in the following table. The interrupt request signal is driven to the high level with a pulse width of one PCLK cycle when the error source status flag of an error source for which interrupt generation is enabled is set.

Table 45.5 Interrupt Requests

Interrupt symbol name	Unit Interrupt Signal	Description	Interrupt Number	sDMA Trigger Number	DTS Trigger Number
INTECMMI	INTECMMI	ECM maskable interrupt (EI level)	EIINT8	—	—
INTECMDCLSMI	INTECMDCLSMI *1	DCLS error interrupt (EI level)	EIINT9	—	—
FEINT	INTECMNMI	ECM maskable interrupt (FE level)	FEINT	—	—

Note 1. When DCLS function is disabled by Flash option byte setting, ECM does not generate DCLS error interrupt (EI level).

The error notifications of this module are listed in the following table.

Table 45.6 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
ECMmESET monitor	The status of ECMmESET.ECMmEST writing	2	—
ECMmESET0 monitor	The status of ECMmESET0.ECMmEST writing	3	—
ECMmESET1 monitor	The status of ECMmESET1.ECMmEST writing	4	—
ECMmESET2 monitor	The status of ECMmESET2.ECMmEST writing	5	—
ECMmESET3 monitor	The status of ECMmESET3.ECMmEST writing	6	—
Delay timer overflow for Error output	The status of the delay timer overflow for $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins	11	—
ECM compare error	ECM redundant compare error	20	—
External Error Input 0	$\overline{\text{ERRORIN0}}$ through digital noise filter	184	—
External Error Input 1	$\overline{\text{ERRORIN1}}$ through digital noise filter	185	—
External Error Input 2	$\overline{\text{ERRORIN2}}$ through digital noise filter	186	—
External Error Input 3	$\overline{\text{ERRORIN3}}$ through digital noise filter	187	—
Software Alarm 0	Software error generation 0 using pseudo error trigger	192	—
Software Alarm 1	Software error generation 1 using pseudo error trigger	193	—
Software Alarm 2	Software error generation 2 using pseudo error trigger	194	—
Software Alarm 3	Software error generation 3 using pseudo error trigger	195	—

45.1.5 Reset Sources

ECMM, ECMC and ECM reset sources are listed in the following table. ECMM, ECMC and ECM are initialized by these reset sources.

Table 45.7 Reset Sources (1/2)

Unit Name	Register Name	Reset condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ECM Master Registers								
ECMM	ECM master error set trigger register	√	√	√	√	√	—	—
ECMM	ECM master error set trigger register n	√	√	√	√	√	—	—
ECMM	ECM master error clear trigger register	√	√	√	√	√	—	—
ECMM	ECM master error clear trigger register n	√	√	√	√	√	—	—
ECMM	ECM master error source status register j	√	—*1	—	—	√	—	—
ECM Checker Registers								
ECMC	ECM checker error set trigger register	√	√	√	√	√	—	—
ECMC	ECM checker error set trigger register n	√	√	√	√	√	—	—
ECMC	ECM checker error clear trigger register	√	√	√	√	√	—	—
ECMC	ECM checker error clear trigger register n	√	√	√	√	√	—	—
ECMC	ECM checker error source status register j	√	—*1	—	—	√	—	—
ECM Common Registers								
ECM	ECM error pulse configuration register	√	√	√	√	√	—	—
ECM	ECM interrupt type selection configuration register j	√	√	√	√	√	—	—
ECM	ECM interrupt notification configuration register i j	√	√	√	√	√	—	—
ECM	ECM internal reset configuration register j	√	√	√	√	√	—	—
ECM	ECM error mask register j	√	√	√	√	√	—	—
ECM	ECM error trigger n mask register j	√	√	√	√	√	—	—
ECM	ECM error source status clear register j	√	√	√	√	√	—	—
ECM	ECM key-code protection register	√	√	√	√	√	—	—
ECM	ECM pseudo error trigger register j	√	√	√	√	√	—	—
ECM	ECM delay timer control register	√	√	√	√	√	—	—

Table 45.7 Reset Sources (2/2)

Unit Name	Register Name	Reset condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
ECM	ECM delay timer register	√	√	√	√	√	—	—
ECM	ECM delay timer compare register	√	√	√	√	√	—	—
ECM	ECM delay timer configuration register j	√	√	√	√	√	—	—
ECM	ECM error output clear invalidation configuration register	√	√	√	—	√	—	—
ECM	ECM error trigger n clear invalidation configuration register	√	√	√	—	√	—	—
ECM	ECM pseudo error mask register	√	√	√	√	√	—	—

Note 1. When Primary VMON Reset occurred, the registers show undefined value.

45.1.6 External Input and Output Signals

External Input/output signals of ECM are listed below.

Table 45.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
$\overline{\text{ERROROUT\_M}}$	Error output master signal	$\overline{\text{ERROROUT\_M}}$
$\overline{\text{ERROROUT\_C}}$	Error output checker signal	$\overline{\text{ERROROUT\_C}}$
$\overline{\text{ERRORINk}}$	External error input signal k *1	$\overline{\text{ERRORINk}}$

Note 1. Digital noise filter must be configured before the function is used. For details of configuration, see **Section 2, Pin Functions**.

45.1.7 Internal Output Signals

Internal output signals of ECM are listed below.

Table 45.9 Internal Output Signals

Unit Signal Name	Description	Connected to
ERROROUTZ	Error trigger signal for Hi-Z control of PIC	PIC
ERROROUTnZ	Error trigger signal n for Port Safe State	Port
ECMRESZ	Internal reset generation signal	Reset Controller

45.2 Overview

45.2.1 Specification Overview

ECM (Error Control Module) collects error signals coming from different error sources and monitoring circuits. It also outputs error signals from the error pins ($\overline{\text{ERROROUT\_M}}$, $\overline{\text{ERROROUT\_C}}$) and generates interrupts and Error Control Module Reset signals. **Table 45.10** shows the specification overview of ECM.

Table 45.10 Specification Overview

Item	Description
Safety processing	ECM can handle the following processing in response to error signal inputs from individual modules. • Error flag set • Interrupt generation Either FE or EI level interrupt generation can be selected for individual errors. Notified CPU can be selected with one, multicast or broadcast for individual errors. If there is configuration that interrupt does not notify anywhere, it means interrupt generation is disable. • Internal reset generation System Reset 2 or Application Reset generation can be controlled (enabled/disabled) for individual errors. • Error pin output Pin output mask can be controlled (enabled/disabled) for individual errors. Output can be toggled in response to a timer input or made at a fixed level.
Error status	ECM incorporates error source status registers, which can be used to confirm the error status from the error flag. The error flags are only cleared by a Power On Reset, DeepSTOP Reset or writing to clear register. In case of System Reset 1, System Reset 2, Application Reset, Module Reset and JTAG Reset, the error flags are kept and the reset generation source can be confirmed by reading the status register after reset.
Debug, self-diagnosis	• Pseudo errors can be generated for debug and self-diagnosis. The operation during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for the masking of the error pin output, interrupt, or Error Control Module Reset apply in the same way. • ECM incorporates a loop-back function of the error output pin ($\overline{\text{ERROROUT\_M}}$) that is used to diagnose the path to the error output pin. The status of the error output pin is reflected to an internal register and can be confirmed by reading the register.
Timeout function	ECM incorporates a function that generates an error output or Error Control Module Reset when the count value of the delay timer matches with the delay timer compare register because the delay timer was not stopped during the interrupt processing after being started simultaneously with the occurrence of an interrupt request.
Port safe state	Error triggers (ERROROUTnZ) connect to port safe state and ECM can control the state of general purpose I/O to a condition (Hi-Z) according to user configuration. For details of function, see **Section 2, Pin Functions**.
Register protection	A write-protection with a key code is implemented to protect registers from illegal write access. For details of function, see **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**.
Error output and Error triggers clear masking	ECM incorporates a function that can mask software clearance of Error output and Error triggers until the counter which is started from error occurrence reaches the value specified in the configuration registers. If another error occurs during time counting, then the time count is reset and restarted from 0.
Hi-Z control signal trigger into PIC	ECM triggers Hi-Z control signal into PIC. Hi-Z control signal (ERROROUTZ) is triggered at the same condition when ECM activates Error output pin. Regarding of enabling of Hi-Z control, see **Section 41, Peripheral Interconnect (PIC)**.
Others	ECM is duplexed. ECM incorporates the error output pin. Error output and Error triggers for Port Safe State from the ECM master and checker are constantly compared. If they do not match, an ECM compare error occurs.

45.2.2 Block Diagram

ECM is redundantly implemented using ECM Master and ECM Checker. See **Figure 45.1, Connection among ECM Master, ECM Checker and peripherals**, and **Figure 45.2, Structure of binding components**.

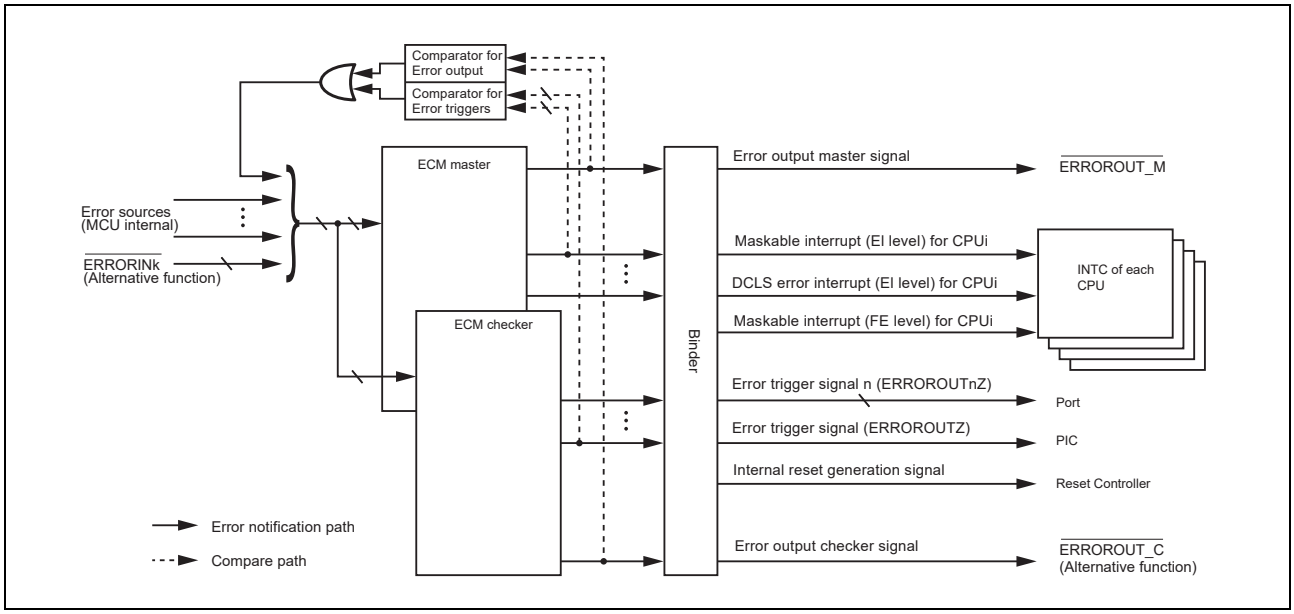


Figure 45.1 Connection among ECM Master, ECM Checker and peripherals

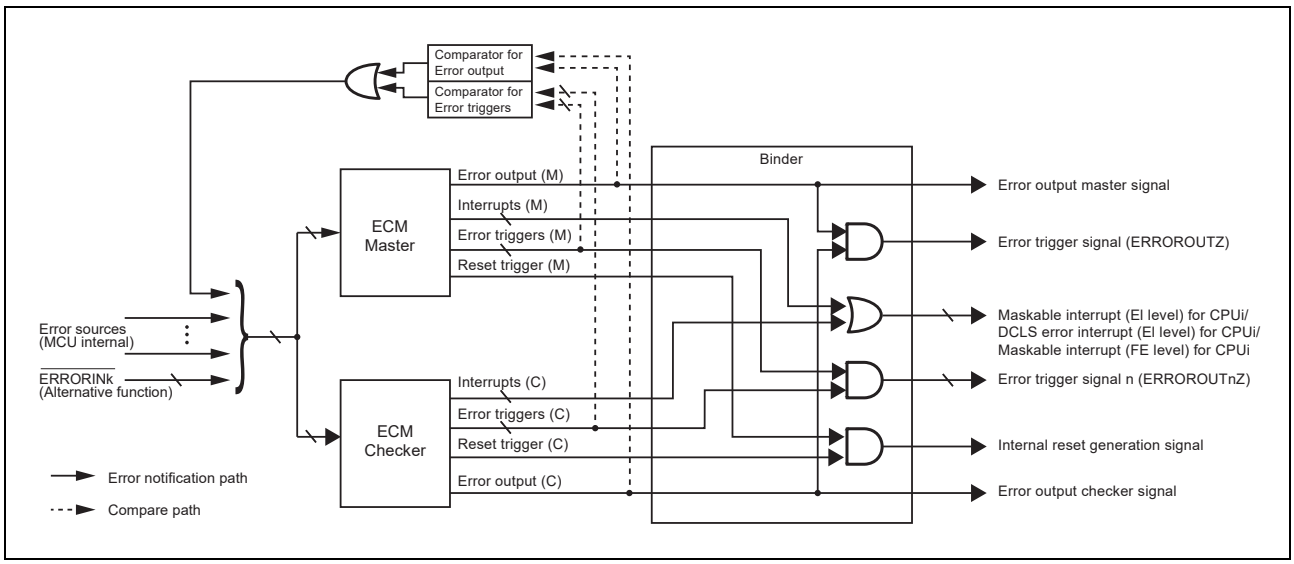


Figure 45.2 Structure of binding components

CAUTION

Pay attention to the difference in output voltage between the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins because different power supply systems are used for those pins.

$\overline{\text{ERROROUT\_M}}$ pin: E1VCC

$\overline{\text{ERROROUT\_C}}$ pin: GETH0RVCC for P3_8 [For U2AEVA, U2A16, U2A8 only]; E0VCC for P3_8 [For U2A6 (292 pins) only]; E0VCC for P4_13 and P5_3; E1VCC for P10_14

45.2.3 Error Input

Table 45.11 shows the error inputs to ECM.

"—": not covered
"√": covered
Gray hatching: its number is reserve bit.

Table 45.11 List of Error Inputs (1/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
0	Error output monitor	The status of ERROROUT_M pin	√	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√
1	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
2	ECMmESET monitor	The status of ECMmESET. ECMmEST writing	√	—	—	—	√	√*1	—	—	√	√	√	√	√	√	√	√	√
3	ECMmESET0 monitor	The status of ECMmESET0.ECMmEST writing	√	—	—	—	—	√*2	—	—	√	√	√	√	√	√	√	√	√
4	ECMmESET1 monitor	The status of ECMmESET1.ECMmEST writing	√	—	—	—	—	√*2	—	—	√	√	√	√	√	√	√	√	√
5	ECMmESET2 monitor	The status of ECMmESET2.ECMmEST writing	√	—	—	—	—	√*2	—	—	√	√	√	—	—	—	—	—	—
6	ECMmESET3 monitor	The status of ECMmESET3.ECMmEST writing	√	—	—	—	—	√*2	—	—	√	√	√	—	—	—	—	—	—
7	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
8	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
9	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
10	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
11	Delay timer overflow for Error output	The status of the delay timer overflow for ERROROUT_M and ERROROUT_C pins	√	—	—	√	√	√*1	—	√	√	√	√	√	√	√	√	√	√
12	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
13	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
15	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
16	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
17	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
18	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
19	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
20	ECM compare error	ECM redundant compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
21	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
22	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
23	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (2/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
24	Mode error during Normal Operation Mode/User Boot Mode	Unintended activation of Production Test Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
25	Mode error during User Boot Mode	Unintended activation of Normal Operation Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
26	Mode error during Normal Operation Mode	Unintended deactivation of Normal Operation Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
27	Mode error during Normal Operation Mode/User Boot Mode	Unintended activation of Serial Programming Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
28	Mode error during Normal Operation Mode	Unintended activation of User Boot Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
29	Mode error during User Boot Mode	Unintended deactivation of User Boot Mode	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
30	Mode error during any mode	Mode check error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
31	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
32	Flash access error	Flash access error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
33	FACI reset transfer error	FACI reset transfer error	√*3	—	—	—	√	√*1	—	√	√	√	√	√	√	√	√	√	√
	FACI reset transfer warning	FACI reset transfer warning																	
34	BIST parameter transfer error	BIST parameter transfer error	√*3	—	—	—	√	√*1	—	√	√	√	√	√	√	√	√	√	√
35	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
36	DTS compare error	DTS redundant lock step compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
37	BUS Bridge compare error of sDMAC	Comparator error of data bus in sDMAC	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

Table 45.11 List of Error Inputs (3/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
38	BUS Bridge compare error	Comparator error of CRAM controller	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Comparator error of SAXI2FAXI bridge (from System Bus to Global FLASH Bus)																	
		Comparator error of SAXI2MBI bridge (from System Bus to CRAM Bus)																	
		Comparator error of VCI2AXI bridge (from Inter-processor element Bus to System Bus)																	
		Comparator error of AXI2VCI bridge (from System Bus to Inter-processor element Bus)																	
		Comparator error of AXI2VCI bridge (from System Bus to P-Bus)																	
		Comparator error of VCI2APB bridge (from P-Bus to each P-Bus Group)																	
		Comparator error of GCFU block for SAXI masters																	
		Comparator error of GCFU block for instruction fetch or data access																	
		Comparator error of CCIB block																	
39	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
40	Inter-processor element Bus routing error	Routing error of CIVCI bus (Inter-processor element Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
41	Inter-cluster Bus (I-Bus) routing error	Routing error of GIVCI bus (I-Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
42	Peripheral Bus (P-Bus) routing error	Routing error of PVCI bus (P-Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
43	CRAM Bus routing error	Routing error GL2MBI bus (CRAM Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
44	System Bus routing error	Routing error of SAXI bus (System Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
45	Global Flash Bus routing error	Routing error of GFAXI bus (Global FLASH Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
46	Local Flash Bus routing error	Routing error of LFAXI bus (Local FLASH Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
47	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48	Clock monitor error for CLK_MOSC	CLMA0 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
49	Clock monitor error for CLK_WDT	CLMA1 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√

Table 45.11 List of Error Inputs (4/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
50	Clock monitor error for CLK_LSIOSC	CLMA2 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
51	Clock monitor error for CLK_LSB	CLMA3 over/under frequency error	√*4	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
		CLMA4 over/under frequency error																	
52	Clock monitor error for CLK_UHSB	CLMA5 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
53	Clock monitor error for CLK_HBUS	CLMAS over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
54	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
55	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
56	OS timer 1 interrupt	OSTM1 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
57	OS timer 2 interrupt	OSTM2 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
58	OS timer 3 interrupt	OSTM3 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
59	OS timer 4 interrupt	OSTM4 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
60	OS timer 5 interrupt	OSTM5 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
61	OS timer 6 interrupt	OSTM6 interrupt	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
62	OS timer 7 interrupt	OSTM7 interrupt	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
63	OS timer 8 interrupt	OSTM8 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
64	OS timer 9 interrupt	OSTM9 interrupt	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
65	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
66	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
67	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
68	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
69	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
70	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
71	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
72	A/D converter parity error	ADCJ0 parity error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		ADCJ1 parity error																	
		ADCJ2 parity error																—	
73	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
74	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
75	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
76	Temperature sensor error	Abnormal temperature error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
77	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
78	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
79	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
80	Code Flash - Address parity error	Code Flash address parity error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
81	Code Flash - Data ECC 2-bit error	Code Flash data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
82	Code Flash - Data ECC 1-bit error	Code Flash data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

Table 45.11 List of Error Inputs (5/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
83	Code Flash - ECC overflow error	Code Flash ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
84	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
85	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
86	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
87	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
88	Data Flash - Data ECC 2-bit error	Data Flash data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
89	Data Flash - Data ECC 1-bit error	Data Flash data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
90	Data Flash - ECC overflow error	Data Flash ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
91	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
92	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
93	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
94	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
95	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
96	Local RAM - Data ECC 2-bit error	LRAM data ECC 2-bit error (for read data loaded from the other PE's Local RAM speculatively)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
97	Local RAM - Data ECC 1-bit error	LRAM data ECC 1-bit error (for read data loaded from the other PE's Local RAM speculatively)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
98	Local RAM - ECC overflow error	LRAM data ECC error address buffer overflow error (for read data loaded from the other PE's Local RAM speculatively)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
99	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
100	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
101	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
102	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
103	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
104	Cluster RAM - Address feedback compare error - Data ECC 2-bit error	CRAM address feedback compare error CRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
105	Cluster RAM - Data ECC 1-bit error	CRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
106	Cluster RAM- ECC overflow error	CRAM data ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
107	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
108	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (6/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
109	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
110	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
111	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
112	DTSRAM - Address feedback compare error - Data ECC 2-bit error	DTSRAM address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		DTSRAM data ECC 2-bit error																	
113	DTSRAM - Data ECC 1-bit error	DTSRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
114	DTSRAM - ECC overflow error	DTSRAM ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
115	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
116	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
117	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
118	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
119	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
120	sDMAC0 RAM - Data ECC 2-bit error	sDMAC0 DPRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC0 DATARAM data ECC 2-bit error																	
121	sDMAC0 RAM - Data ECC 1-bit error	sDMAC0 DPRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC0 DATARAM data ECC 1-bit error																	
122	sDMAC1 RAM - Data ECC 2-bit error	sDMAC1 DPRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC1 DATARAM data ECC 2-bit error																	
123	sDMAC1 RAM - Data ECC 1-bit error	sDMAC1 DPRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC1 DATARAM data ECC 1-bit error																	
124	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
125	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
126	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
127	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (7/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
128	FlexRay RAM - Data ECC 2-bit error	FLXA0 MRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		FLXA0 TBFRAM A data ECC 2-bit error																	
		FLXA0 TBFRAM B data ECC 2-bit error																	
		FLXA1 MRAM data ECC 2-bit error															—	—	—
		FLXA1 TBFRAM A data ECC 2-bit error																	
		FLXA1 TBFRAM B data ECC 2-bit error																	
129	FlexRay RAM - Data ECC 1-bit error	FLXA0 MRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		FLXA0 TBFRAM A data ECC 1-bit error																	
		FLXA0 TBFRAM B data ECC 1-bit error																	
		FLXA1 MRAM data ECC 1-bit error															—	—	—
		FLXA1 TBFRAM A data ECC 1-bit error																	
		FLXA1 TBFRAM B data ECC 1-bit error																	
130	RS-CANFD RAM - Data ECC 2-bit error	RSCFD0 AFLRAM 0 data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		RSCFD0 AFLRAM 1 data ECC 2-bit error																	
		RSCFD0 MRAM data ECC 2-bit error																	
		RSCFD1 AFLRAM 0 data ECC 2-bit error																	
		RSCFD1 AFLRAM 1 data ECC 2-bit error																	
		RSCFD1 MRAM data ECC 2-bit error																	
131	RS-CANFD RAM - Data ECC 1-bit error	RSCFD0 AFLRAM 0 data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		RSCFD0 AFLRAM 1 data ECC 1-bit error																	
		RSCFD0 MRAM data ECC 1-bit error																	
		RSCFD1 AFLRAM 0 data ECC 1-bit error																	
		RSCFD1 AFLRAM 1 data ECC 1-bit error																	
		RSCFD1 MRAM data ECC 1-bit error																	

Table 45.11 List of Error Inputs (8/22)

Error sources			Functions								Products							
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
132	MSPI RAM - Data ECC 2-bit error	MSPI0 RAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		MSPI1 RAM data ECC 2-bit error																	—
		MSPI2 RAM data ECC 2-bit error																	
		MSPI3 RAM data ECC 2-bit error																	
		MSPI4 RAM data ECC 2-bit error																	
		MSPI5 RAM data ECC 2-bit error																	
		MSPI6 RAM data ECC 2-bit error											—	—		—	—	—	
		MSPI7 RAM data ECC 2-bit error																	
		MSPI8 RAM data ECC 2-bit error																	
		MSPI9 RAM data ECC 2-bit error										—			—				
133	MSPI RAM - Data ECC 1-bit error	MSPI0 RAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		MSPI1 RAM data ECC 1-bit error																	—
		MSPI2 RAM data ECC 1-bit error																	
		MSPI3 RAM data ECC 1-bit error																	
		MSPI4 RAM data ECC 1-bit error																	
		MSPI5 RAM data ECC 1-bit error																	
		MSPI6 RAM data ECC 1-bit error											—	—		—	—	—	
		MSPI7 RAM data ECC 1-bit error																	
		MSPI8 RAM data ECC 1-bit error																	
		MSPI9 RAM data ECC 1-bit error										—			—				

Table 45.11 List of Error Inputs (9/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
134	GTM RAM - Data ECC 2-bit error	GTM MCS0 RAM 0 data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		GTM MCS0 RAM 1 data ECC 2-bit error																	
		GTM MCS1 RAM 0 data ECC 2-bit error																	
		GTM MCS1 RAM 1 data ECC 2-bit error																	
		GTM MCS2 RAM 0 data ECC 2-bit error																	
		GTM MCS2 RAM 1 data ECC 2-bit error																	
		GTM MCS3 RAM 0 data ECC 2-bit error																	
		GTM MCS3 RAM 1 data ECC 2-bit error																	
135	GTM RAM - Data ECC 1-bit error	GTM MCS0 RAM 0 data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		GTM MCS0 RAM 1 data ECC 1-bit error																	
		GTM MCS1 RAM 0 data ECC 1-bit error																	
		GTM MCS1 RAM 1 data ECC 1-bit error																	
		GTM MCS2 RAM 0 data ECC 1-bit error																	
		GTM MCS2 RAM 1 data ECC 1-bit error																	
		GTM MCS3 RAM 0 data ECC 1-bit error																	
		GTM MCS3 RAM 1 data ECC 1-bit error																	
136	Fast Ethernet RAM - Data ECC 2-bit error	ETNB0 Transmit FIFO data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—	√
		ETNB0 Receive FIFO data ECC 2-bit error																	
137	Fast Ethernet RAM - Data ECC 1-bit error	ETNB0 Transmit FIFO data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—	√
		ETNB0 Receive FIFO data ECC 1-bit error																	
138	Gigabit Ethernet RAM - Data ECC 2-bit error	ETNB1 Transmit FIFO data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	—	√	√	—	—	—
		ETNB1 Receive FIFO data ECC 2-bit error																	
139	Gigabit Ethernet RAM - Data ECC 1-bit error	ETNB1 Transmit FIFO data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	—	√	√	—	—	—
		ETNB1 Receive FIFO data ECC 1-bit error																	

Table 45.11 List of Error Inputs (10/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
140	MMCA RAM - Data ECC 2-bit error	MMCA0 RAM A data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—	—
		MMCA0 RAM B data ECC 2-bit error																	
141	MMCA RAM - Data ECC 1-bit error	MMCA0 RAM A data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—	—
		MMCA0 RAM B data ECC 1-bit error																	
142	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
143	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
144	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
145	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
146	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
147	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
148	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
149	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
150	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
151	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
152	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
153	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
154	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
155	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
156	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
157	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
158	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
159	Peripheral RAM - ECC overflow error	sDMAC0 DPRAM data ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC0 DATARAM data ECC error address buffer overflow error																	
		sDMAC1 DPRAM data ECC error address buffer overflow error																	
		sDMAC1 DATARAM data ECC error address buffer overflow error																	
		FLXA0 MRAM data ECC error address buffer overflow error																	
		FLXA0 TBFRAM A data ECC error address buffer overflow error																	
		FLXA0 TBFRAM B data ECC error address buffer overflow error																	

Table 45.11 List of Error Inputs (11/22)

Error sources			Functions								Products							
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
159	Peripheral RAM - ECC overflow error	FLXA1 MRAM data ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	—	—	—
		FLXA1 TBFRAM A data ECC error address buffer overflow error																	
		FLXA1 TBFRAM B data ECC error address buffer overflow error																	
		RSCFD0 AFLRAM 0 data ECC error address buffer overflow error															√	√	√
		RSCFD0 AFLRAM 1 data ECC error address buffer overflow error																	
		RSCFD0 MRAM data ECC error address buffer overflow error																	
		RSCFD1 AFLRAM 0 data ECC error address buffer overflow error																	
		RSCFD1 AFLRAM 1 data ECC error address buffer overflow error																	
		RSCFD1 MRAM data ECC error address buffer overflow error																	
		MSPI0 RAM data ECC error address buffer overflow error																	
		MSPI1 RAM data ECC error address buffer overflow error																	
		MSPI2 RAM data ECC error address buffer overflow error																	
		MSPI3 RAM data ECC error address buffer overflow error																	
		MSPI4 RAM data ECC error address buffer overflow error																	—
		MSPI5 RAM data ECC error address buffer overflow error																	—
		MSPI6 RAM data ECC error address buffer overflow error											—	—		—	—	—	

Table 45.11 List of Error Inputs (12/22)

Error sources			Functions								Products							
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
159	Peripheral RAM - ECC overflow error	MSPI7 RAM data ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	—	—	√	—	—	—	—
		MSPI8 RAM data ECC error address buffer overflow error																	
		MSPI9 RAM data ECC error address buffer overflow error										—			—				
		GTM MCS0 RAM 0 data ECC error address buffer overflow error										√	√	√	√	√	√	√	√
		GTM MCS0 RAM 1 data ECC error address buffer overflow error																	
		GTM MCS1 RAM 0 data ECC error address buffer overflow error																	
		GTM MCS1 RAM 1 data ECC error address buffer overflow error																	
		GTM MCS2 RAM 0 data ECC error address buffer overflow error																	
		GTM MCS2 RAM 1 data ECC error address buffer overflow error																	
		GTM MCS3 RAM 0 data ECC error address buffer overflow error																	
		GTM MCS3 RAM 1 data ECC error address buffer overflow error																	
		MMCA0 RAM A data ECC error address overflow error																—	—
		MMCA0 RAM B data ECC error address overflow error																	
160	Data transfer path - Address EDC error	Data transfer path in Code Flash address EDC 2-bit error (CCIB, bridge from System Bus to Global FLASH Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Data transfer path in CRAM address EDC 2-bit error (Bridge from System Bus to CRAM Bus, CRAMC)																	
		Data transfer path in LRAM address EDC 2-bit error (Bridge from System Bus to Inter-processor element Bus, VCIS I/F in PE)																	

Table 45.11 List of Error Inputs (13/22)

Error sources			Functions								Products							
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
160	Data transfer path - Address EDC error	Data transfer path in I-Bus address EDC 2-bit error (Slave modules on I-Bus)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Data transfer path in AXI2PVCI bridge address EDC 2-bit error (Bridge from System Bus to P-Bus)																	
		Data transfer path in P-Bus address EDC 2-bit error (Slave modules on P-Bus)																	
		Data transfer path in remap module address EDC 2-bit error (GCFU for Local FLASH Bus during remap, bridge from Global FLASH Bus to System Bus, GCFU for Global FLASH Bus during remap)																	
		Data transfer path in H-Bus address EDC 2-bit error (Slave modules on H-Bus)																	
		Data transfer path in Code Flash address EDC 1-bit error (CCIB, bridge from System Bus to Global FLASH Bus)																	
		Data transfer path in CRAM address EDC 1-bit error (Bridge from System Bus to CRAM Bus, CRAMC)																	
		Data transfer path in LRAM address EDC 1-bit error (Bridge from System Bus to Inter-processor element Bus, VCIS I/F in PE)																	
		Data transfer path in I-Bus address EDC 1-bit error (Slave modules on I-Bus)																	
		Data transfer path in AXI2PVCI bridge address EDC 1-bit error (Bridge from System Bus to P-Bus)																	
		Data transfer path in P-Bus address EDC 1-bit error (Slave modules on P-Bus)																	

Table 45.11 List of Error Inputs (14/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
160	Data transfer path - Address EDC error	Data transfer path in remap module address EDC 1-bit error (GCFU for Local FLASH Bus during remap, bridge from Global FLASH Bus to System Bus, GCFU for Global FLASH Bus during remap)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Data transfer path in H-Bus address EDC 1-bit error (Slave modules on H-Bus)																	
161	Data transfer path - Data ECC 2-bit error	Data transfer path in CRAM data ECC 2-bit error (RMW-write in CRAMC, RMW-cdata in CRAMC)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Data transfer path in DMAC/DTS data ECC 2-bit error (sDMAC masters, DTS master)																	
		Data transfer path in LRAM data ECC 2-bit error (VCIS I/F in PE)																	
		Data transfer path in I-Bus data ECC 2-bit error (Slave modules on I-Bus)																	
		Data transfer path in PE for read data from peripherals data ECC 2-bit error (VCIM I/F in PE)																	
		Data transfer path in VCI2AXI bridge for write/read data data ECC 2-bit error (RMW-read in bridge from Inter-processor element Bus to System Bus, RMW-write bridge from Inter-processor element Bus to System Bus)																	
		Data transfer path in P-Bus data ECC 2-bit error (Slave modules on P-Bus)																	
		Data transfer path in remap module data ECC 2-bit error (Bridge from Global FLASH Bus to System Bus)															—	—	—
		Data transfer path in H-Bus data ECC 2-bit error (Master/slave modules on H-Bus)															√	√	√

Table 45.11 List of Error Inputs (15/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
162	Data transfer path - Data ECC 1-bit error	Data transfer path in CRAM data ECC 1-bit error (RMW-write in CRAMC, RMW-cdata in CRAMC)	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Data transfer path in DMAC/DTS data ECC 1-bit error (sDMAC masters, DTS master)																	
		Data transfer path in LRAM data ECC 1-bit error (VCIS I/F in PE)																	
		Data transfer path in I-Bus data ECC 1-bit error (Slave modules on I-Bus)																	
		Data transfer path in PE for read data from peripherals data ECC 1-bit error (VCIM I/F in PE)																	
		Data transfer path in VCI2AXI bridge for write/read data ECC 1-bit error (RMW-read in bridge from Inter-processor element Bus to System Bus, RMW-write bridge from Inter-processor element Bus to System Bus)																	
		Data transfer path in P-Bus data ECC 1-bit error (Slave modules on P-Bus)																	
		Data transfer path in remap module data ECC 1-bit error (Bridge from Global FLASH Bus to System Bus)															—	—	—
		Data transfer path in H-Bus data ECC 1-bit error (Master/slave modules on H-Bus)															√	√	√
163	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
164	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
165	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
166	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
167	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
168	CRAM Guard error	CRG error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
169	I-Bus Guard error	IBG error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
170	P-Bus Guard error	DTS guard error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		sDMAC guard error																	
		INTC2 guard error																	
		PBG error																	
171	H-Bus Guard error	HBG error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
172	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (16/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
173	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
174	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
175	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
176	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
177	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
178	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
179	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
180	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
181	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
182	DTS transfer error/ DMA address error*7	DTS transfer error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		DMA address error																	
183	H-Bus master Transfer Error*7	H-Bus master transfer error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
184	External Error Input 0	$\overline{\text{ERRORIN0}}$ through digital noise filter	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
185	External Error Input 1	$\overline{\text{ERRORIN1}}$ through digital noise filter	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
186	External Error Input 2	$\overline{\text{ERRORIN2}}$ through digital noise filter	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
187	External Error Input 3	$\overline{\text{ERRORIN3}}$ through digital noise filter	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
188	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
189	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
190	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
191	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
192	Software Alarm 0	Software error generation 0 using pseudo error trigger	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
193	Software Alarm 1	Software error generation 1 using pseudo error trigger	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
194	Software Alarm 2	Software error generation 2 using pseudo error trigger	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
195	Software Alarm 3	Software error generation 3 using pseudo error trigger	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
196	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
197	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
198	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
199	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
200	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
201	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
202	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
203	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
204	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
205	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
206	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
207	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
208	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
209	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (17/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
210	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
211	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
212	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
213	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
214	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
215	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
216	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
217	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
218	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
219	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
220	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
221	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
222	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
223	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
224	DCLS compare error (PE0)	Redundant lock step compare error	√	√*5	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
225	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
226	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
227	Unintended Debug Enable detection (PE0)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
228	Watchdog timer error (PE0)	WDTB0 error	√	√	√	√*6	√	√	√	√	√	√	√	√	√	√	√	√	√
229	Clock monitor error for CLK_CPU (PE0)	CLMA6 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
230	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
231	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
232	Local RAM (PE0) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error LRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
233	Local RAM (PE0) - Data ECC 1-bit error	LRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
234	Local RAM (PE0) - ECC overflow error	LRAM ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
235	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (18/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
236	Instruction cache RAM (PE0) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Instruction cache RAM (data) address feedback compare error																	
		Instruction cache RAM (tag) data EDC 2-bit error																	
		Instruction cache RAM (tag) data EDC 1-bit error																	
		Instruction cache RAM (data) data EDC 2-bit error																	
		Instruction cache RAM (data) data EDC 1-bit error																	
237	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
238	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
239	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
240	PE guard error (PE0)	PEG error by access to PE0 slave	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
241	PE guard error by PE0 read access (PE0)	PEG error detected in a read request from PE0 to the other LRAM	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
242	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
243	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
244	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
245	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
246	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
247	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
248	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
249	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
250	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
251	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
252	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
253	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
254	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
255	Data Access Error (PE0)*7	PE0 data access error	√	√	√	√	—*8	—	√	√	√	√	√	√	√	√	√	√	√
256	DCLS compare error (PE1)	Redundant lock step compare error	√	√*5	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
257	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
258	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
259	Unintended Debug Enable detection (PE1)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
260	Watchdog timer error (PE1)	WDTB1 error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

Table 45.11 List of Error Inputs (19/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
261	Clock monitor error for CLK_CPU (PE1)	CLMA7 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	√	√	√	√	√	√
262	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
263	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
264	Local RAM (PE1) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		LRAM data ECC 2-bit error																	
265	Local RAM (PE1) - Data ECC 1-bit error	LRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
266	Local RAM (PE1) - ECC overflow error	LRAM ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
267	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
268	Instruction cache RAM (PE1) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
		Instruction cache RAM (data) address feedback compare error																	
		Instruction cache RAM (tag) data EDC 2-bit error																	
		Instruction cache RAM (tag) data EDC 1-bit error																	
		Instruction cache RAM (data) data EDC 2-bit error																	
		Instruction cache RAM (data) data EDC 1-bit error																	
269	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
270	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
271	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
272	PE guard error (PE1)	PEG error by access to PE1 slave	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
273	PE guard error by PE1 read access (PE1)	PEG error detected in a read request from PE1 to the other LRAM	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
274	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
275	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
276	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
277	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
278	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
279	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
280	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
281	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
282	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
283	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
284	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
285	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
286	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (20/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
287	Data Access Error (PE1)*7	PE1 data access error	√	√	√	√	—*8	—	√	√	√	√	√	√	√	√	√	√	√
288	DCLS compare error (PE2)	Redundant lock step compare error	√	√*5	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
289	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
290	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
291	Unintended Debug Enable detection (PE2)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
292	Watchdog timer error (PE2)	WDTB2 error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
293	Clock monitor error for CLK_CPU (PE2)	CLMA8 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	—	—	—	—	—	—
294	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
295	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
296	Local RAM (PE2) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
		LRAM data ECC 2-bit error																	
297	Local RAM (PE2) - Data ECC 1-bit error	LRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
298	Local RAM (PE2) - ECC overflow error	LRAM ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
299	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
300	Instruction cache RAM (PE2) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
		Instruction cache RAM (data) address feedback compare error																	
		Instruction cache RAM (tag) data EDC 2-bit error																	
		Instruction cache RAM (tag) data EDC 1-bit error																	
		Instruction cache RAM (data) data EDC 2-bit error																	
		Instruction cache RAM (data) data EDC 1-bit error																	
301	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
302	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
303	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
304	PE guard error (PE2)	PEG error by access to PE2 slave	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
305	PE guard error by PE2 read access (PE2)	PEG error detected in a read request from PE2 to the other LRAM	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
306	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
307	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (21/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
308	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
309	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
310	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
311	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
312	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
313	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
314	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
315	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
316	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
317	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
318	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
319	Data Access Error (PE2)*7	PE2 data access error	√	√	√	√	—*8	—	√	√	√	√	√	—	—	—	—	—	—
320	DCLS compare error (PE3)	Redundant lock step compare error	√	√*5	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
321	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
322	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
323	Unintended Debug Enable detection (PE3)	This error issued when CPU operating mode transits to debug mode without authentication from debugger	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
324	Watchdog timer error (PE3)	WDTB3 error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
325	Clock monitor error for CLK_CPU (PE3)	CLMA9 over/under frequency error	√	√	√	√	√	√	—	√	√	√	√	—	—	—	—	—	—
326	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
327	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
328	Local RAM (PE3) - Address feedback compare error - Data ECC 2-bit error	LRAM address feedback compare error LRAM data ECC 2-bit error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
329	Local RAM (PE3) - Data ECC 1-bit error	LRAM data ECC 1-bit error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
330	Local RAM (PE3) - ECC overflow error	LRAM ECC error address buffer overflow error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
331	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 45.11 List of Error Inputs (22/22)

Error sources			Functions								Products								
No.	Error name	Description	Error flag set	Maskable Interrupt (EI level)	Maskable Interrupt (FE level)	Internal reset generation	Error output	Error trigger	Delay timer start	Pseudo error generation	U2A-EVA U2A16 mode (516 pins), U2A16 (516 pins)	U2A16 (373 pins)	U2A16 (292 pins)	U2A-EVA U2A8 mode (516 pins)	U2A8 (373 pins)	U2A8 (292 pins)	U2A6 (292/176 pins)	U2A6 (156 pins)	U2A6 (144 pins)
332	Instruction cache RAM (PE3) - Address feedback compare error - Data EDC error	Instruction cache RAM (tag) address feedback compare error	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
		Instruction cache RAM (data) address feedback compare error																	
		Instruction cache RAM (tag) data EDC 2-bit error																	
		Instruction cache RAM (tag) data EDC 1-bit error																	
		Instruction cache RAM (data) data EDC 2-bit error																	
		Instruction cache RAM (data) data EDC 1-bit error																	
333	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
334	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
335	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
336	PE guard error (PE3)	PEG error by access to PE3 slave	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
337	PE guard error by PE3 read access (PE3)	PEG error detected in a read request from PE3 to the other LRAM	√	√	√	√	√	√	√	√	√	√	√	—	—	—	—	—	—
338	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
339	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
340	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
341	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
342	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
343	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
344	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
345	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
346	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
347	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
348	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
349	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
350	Reserved	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
351	Data Access Error (PE3)*7	PE3 data access error	√	√	√	√	—*8	—	√	√	√	√	√	—	—	—	—	—	—

Note 1. Error trigger for Hi-Z control of PIC only is supported.

Note 2. Error trigger for Port Safe State only is supported.

Note 3. FACI reset transfer error/warning and BIST parameter transfer error have a possible to be flagged at the same time, when System Reset 1 occurs during reset sequence. Then FACI Reset Transfer Status registers and BIST result registers must be checked. For details of FACI Reset Transfer Status registers, see **Section 51, Flash Memory** and the detail of BIST result registers, see **Section 44, Functional Safety**. If FACI Reset Transfer Status registers have not detected any error information, FACI reset transfer error/warning can be recognized as pseudo error. If BIST has been confirmed to be correctly executed by checking of BIST result registers, BIST parameter transfer error can be recognized as pseudo error.

Note 4. The error input of CLMA3 may be detected in the initial state. For details, see **Section 14, Clock Monitor (CLMA)**.

Note 5. This module generates specific interrupts for Redundant lock step compare error when maskable interrupt (EI level) is selected. For details, see **45.2.12, DCLS Error Interrupt (EI Level) and EI Level Interrupt**.

Note 6. The internal reset generation is enabled in the initial state.

Note 7. ECM functions for an error handling can be used for debug purpose only. In other use cases, ECM functions must be disabled by related configuration registers.

Note 8. Before $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins are cleared to high level, the error output of Data Access Error must be disabled by appropriate error output signal mask control bits which are described in **Section 45.3.11, ECMEMKj — ECM Error Mask Register j**.

45.2.4 Operations for Error Output

After reset release, the $\overline{\text{ERROROUT\_M}}$ pin outputs the low (error) level. Follow the procedure described in **Section 45.3.4, ECMmECLR — ECM Master/Checker Error Clear Trigger Register**, to clear the error before using ECM. As the $\overline{\text{ERROROUT\_C}}$ pin is multiplexed with a general-purpose port and other functions, select the $\overline{\text{ERROROUT\_C}}$ function before use. For settings, see **Section 2, Pin Functions**.

The error output can be configured for two different modes of operation, non-dynamic or dynamic.

The error output is in synchronization with the occurrence of error source conditions and the error level is output as the pin state regardless of the dynamic mode pulse cycle.

Error triggers connect to port safe state and ECM can control the state of general purpose I/O to safe state according to user configuration. For details of function, see **Section 2, Pin Functions**.

Table 45.12 Operation for Error Output

Error Status ECMmSSE	Operating Mode ECMEPCFG.ECMSL0 Bit	Error Output Operating Mode	Error Output Level*1	Error Trigger Level*1	Error Status
0	0	Non-dynamic	H	H	No error
	1	Dynamic	Toggles (according to timer input)	H	No error
1	0	Non-dynamic	L	L	Error
	1	Dynamic	L	L	Error

Note 1. The level of the error output ($\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$) can be masked by ECMEMKj registers and the level of the error trigger (ERROROUTnZ) can be masked by ECMETMKn_j registers.

45.2.5 $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ Behavior at Reset

Below table explains the behavior of the error output logic and the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins at reset. Also the level of the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ signals during and after reset is explained.

Table 45.13 $\overline{\text{ERROROUT\_M}}$ Behavior at Reset

Category	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
$\overline{\text{ERROROUT\_M}}$ pin level during reset	Hi-Z	Hi-Z*1	Low level	Level according to error status before reset*2	Hi-Z	Level according to error status before reset	Level according to error status before reset
$\overline{\text{ERROROUT\_M}}$ pin level after reset	Low level	Low level	Low level	Level according to error status before reset*2	Low level	Level according to error status before reset	Level according to error status before reset

Note 1. In case of Debugger Disconnection Reset, $\overline{\text{ERROROUT\_M}}$ pin level during reset shows low level.

Note 2. The level of the $\overline{\text{ERROROUT\_M}}$ can be changed by clearing ECMEMKj registers at reset. To keep the level, it is necessary to clear the ECMMESSTRj bit which corresponds to ECMEMKj bit of setting which masked error output before reset.

Table 45.14 $\overline{\text{ERROROUT\_C}}$ Behavior at Reset

Category	Reset Condition						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
$\overline{\text{ERROROUT\_C}}$ pin level during reset	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Level according to error status before reset*1	Level according to error status before reset	Level according to error status before reset
$\overline{\text{ERROROUT\_C}}$ pin level after reset	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Level according to error status before reset*1	Level according to error status before reset	Level according to error status before reset

Note 1. The pin level is kept with I/O hold state function. Port alternative function for $\overline{\text{ERROROUT\_C}}$ has to be reconfigured before releasing I/O hold state, if low level output needs to be kept.

45.2.5.1 Dynamic Mode Enable

1. Initialize the related timer OSTM0.
2. Set the error output to high level by setting ECMmECT bit in ECM master/checker error clear trigger register to 1.
3. Set ECMEPCFG.ECMSL0 bit to 1 for dynamic mode.
4. Start the timer OSTM0.

45.2.5.2 Dynamic Mode Disable

1. Set the error output to low level by setting ECMmEST bit in ECM master/checker error set trigger register to 1.
2. Stop the timer OSTM0.
3. Clear ECMSL0 bit in ECM error pulse configuration register to 0 to specify non-dynamic mode.

45.2.6 Loop-Back Function

ECM incorporates a loop-back function that is used to check the path to the error output pin ($\overline{\text{ERROROUT\_M}}$). The output level of the error output pin can be checked with ECMmSSE0_0 bit in ECM master/checker error source status register 0.

45.2.7 Pseudo Error Generation

ECM incorporates a function that can generate pseudo errors for test or debug purposes. The operation of the ECM during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for error masks, interrupt, internal reset, or delay timer apply in the same way.

45.2.8 Error Status

The error status is indicated by ECM master/checker error source status registers. The error status is only cleared by Power On reset, DeepSTOP reset or writing to ECMESSTCj. In case of reset except for Power On reset and DeepSTOP reset, the error status is kept and the error that triggered the reset can be confirmed by reading the ECM master/checker error source status registers after reset release.

45.2.9 Register Protection

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

By releasing the protection of ECMKCPROT, it can be written. For details, see **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**.

45.2.10 Timeout Function for Interrupt Processing

The delay timer incorporated in ECM can be started simultaneously with the occurrence of an interrupt request. ECM incorporates a function that generates an error output or Error Control Module Reset when the count value of the delay timer matches with the value of the delay timer compare register because the delay timer was not stopped during the interrupt processing. The timer counting is not stopped when a break occurs.

The counting of the delay timer always starts from 0. Configure the duration until an Error Control Module Reset or error output is generated with the settings of the delay timer compare register.

In response to the occurrence of a new error source condition with the setting to start the delay timer while the delay timer is operating, the counter value of the delay timer which is currently operating is not reset. Instead, counting by the timer continues.

45.2.11 Masking of error clear trigger registers

The active error output/error triggers status must be cleared by software via the Error clear trigger registers (ECMMECLR/ECMMECLRn/ECMCECLR/ECMCECLRn). A minimum activation time of the error output is achieved by the Error output clear invalidation counter. This counter is (re)started each time a new error event is triggered at the ECM. It counts up from 00 0000_H to FF $FFFF_H$. Error output clear by software is not possible unless this counter reaches the compare value configured in ECMEOCCFG/ECMETCCFGn register. If Error output/trigger clear invalidation counter is still running, Error output/trigger clear is masked and Error output/trigger clear request by software is ignored.

45.2.12 DCLS Error Interrupt (EI Level) and EI Level Interrupt

This module can generate specific interrupts of DCLS error for each CPU. When DCLS error for a certain PE is occurred in the state of setting Maskable (EI level) interrupt by ECMISCFGj register and interrupt generation enabled by ECMINCFGi_j register, DCLS error interrupt is generated to the PE and EI level interrupt is generated to other PEs. The following figure shows the generation logic of DCLS error interrupt and EI level interrupt from ECM. For each CPUi, EI level interrupt of CPUi is generated by EI level interrupt from ECM master, ECM checker and other DCLS error interrupt except self's DCLS error interrupt being OR-ed.

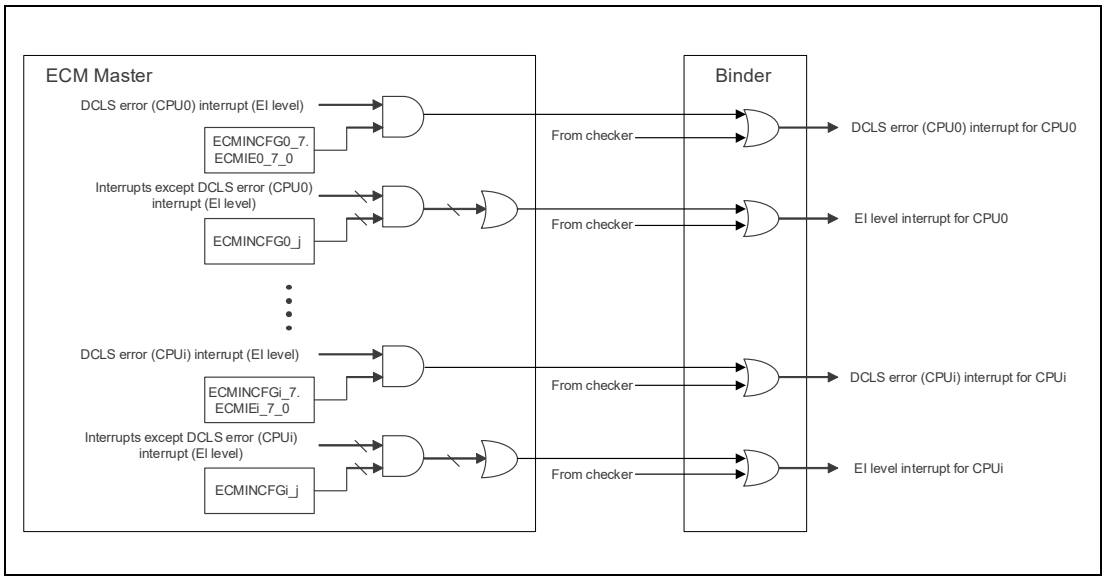


Figure 45.3 DCLS Error Interrupt (EI Level) and EI Level Interrupt

45.3 Register Specification

45.3.1 List of Registers

ECM consists of three address areas: ECM master, ECM checker, ECM common.

The following shows the register map of the ECM master and checker registers.

Table 45.15 Address List of ECM Master and Checker Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECM Master Registers <ECMM_base: FFD3 9000_H>						
ECMM	ECM master error set trigger register	ECMMESET	<ECMM_base> + 000_H	32	PBG30#12	ECMKCPROT
ECMM	ECM master error set trigger register n	ECMMESETn	<ECMM_base> + 008_H to 020_H	32	PBG30#12	ECMKCPROT
ECMM	ECM master error clear trigger register	ECMMECLR	<ECMM_base> + 004_H	32	PBG30#12	ECMKCPROT
ECMM	ECM master error clear trigger register n	ECMMECLRn	<ECMM_base> + $00C_H$ to 024_H	32	PBG30#12	ECMKCPROT
ECMM	ECM master error source status register j	ECMMESSTRj	<ECMM_base> + 048_H to 070_H	32	PBG30#12	—
ECM Checker Registers <ECMC_base: FFD3 $A000_H$>						
ECMC	ECM checker error set trigger register	ECMCESET	<ECMC_base> + 000_H	32	PBG30#12	ECMKCPROT
ECMC	ECM checker error set trigger register n	ECMCESETn	<ECMC_base> + 008_H to 020_H	32	PBG30#12	ECMKCPROT
ECMC	ECM checker error clear trigger register	ECMCECLR	<ECMC_base> + 004_H	32	PBG30#12	ECMKCPROT
ECMC	ECM checker error clear trigger register n	ECMCECLRn	<ECMC_base> + $00C_H$ to 024_H	32	PBG30#12	ECMKCPROT
ECMC	ECM checker error source status register j	ECMCESSTRj	<ECMC_base> + 048_H to 070_H	32	PBG30#12	—

The following shows the register map of the ECM common registers.

Table 45.16 Address List of ECM Common Registers

Module Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
ECM Common Registers <ECM_base: FFD3 8000_H>						
ECM	ECM error pulse configuration register	ECMEPCFG	<ECM_base> + 000_H	32	PBG30#12	ECMKCPROT
ECM	ECM interrupt type selection configuration register j	ECMISCFGj	<ECM_base> + 004_H to $02C_H$	32	PBG30#12	ECMKCPROT
ECM	ECM interrupt notification configuration register i_j	ECMINCFGi_j	<ECM_base> + 040_H to $011C_H$	32	PBG30#12	ECMKCPROT
ECM	ECM internal reset configuration register j	ECMIRCFGj	<ECM_base> + 220_H to 248_H	32	PBG30#12	ECMKCPROT
ECM	ECM error mask register j	ECMEMKj	<ECM_base> + $25C_H$ to 284_H	32	PBG30#12	ECMKCPROT
ECM	ECM error trigger n mask register j	ECMETMKn_j	<ECM_base> + 298_H to 374_H	32	PBG30#12	ECMKCPROT
ECM	ECM error source status clear trigger register j	ECMESSTCj	<ECM_base> + 478_H to $4A0_H$	32	PBG30#12	ECMKCPROT
ECM	ECM key code protection register	ECMKCPROT	<ECM_base> + $4B4_H$	32	PBG30#12	—
ECM	ECM pseudo error trigger register j	ECMPEj	<ECM_base> + $4B8_H$ to $4E0_H$	32	PBG30#12	ECMKCPROT
ECM	ECM delay timer control register	ECMDTMCTL	<ECM_base> + $4F4_H$	32	PBG30#12	ECMKCPROT
ECM	ECM delay timer register	ECMDTMR	<ECM_base> + $4F8_H$	32	PBG30#12	—
ECM	ECM delay timer compare register	ECMDTMCMP	<ECM_base> + $4FC_H$	32	PBG30#12	ECMKCPROT
ECM	ECM delay timer configuration register j	ECMDTMCFGj	<ECM_base> + 500_H to 528H	32	PBG30#12	ECMKCPROT
ECM	ECM error output clear invalidation configuration register	ECMEOCCFG	<ECM_base> + $71C_H$	32	PBG30#12	ECMKCPROT
ECM	ECM error trigger n clear invalidation configuration register	ECMETCCFGn	<ECM_base> + 720_H to $72C_H$	32	PBG30#12	ECMKCPROT
ECM	ECM pseudo error mask register	ECMPEM	<ECM_base> + 740_H	32	PBG30#12	—

The register areas of the ECM registers are common to the redundancy area to be implemented. Writes to the common register areas are conducted simultaneously. The common area for ECM master is read by reading access to the common area. The ECM master register and the ECM checker register represent the address areas which can be written separately.

CAUTION

The register bit related to reserved function and reserved error shown by "—" in Table 45.11, List of Error Inputs must not be configured and register access must follow "When reading, the value after reset is read. When writing, write the value after reset".

45.3.2 ECMmESET — ECM Master/Checker Error Set Trigger Register

The ECM master/checker error set trigger register is for setting the error signal from the error pin to the low level. When ECMmEST bit is set to 1, the error pin outputs the low level. The output cannot be masked. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register** for details of key code protection. This register is always read as $0000\ 0000_H$.

Access: This register is a write-only register that can be written in 32-bit units.

Address: ECMMESET: <ECMM_base> + 000_H
ECMCESET: <ECMC_base> + 000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECMmEST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 45.17 ECMmESET Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ECMmEST	Error set trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the active (low) level.

CAUTIONS

Setting the error output from the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins via ECMmESET register will set ECMmSSE0_20 bit of ECMmESSTR0 register (ECM compare error). Therefore, ECMmESET register has to be set following the sequence below.

1. Set MSKM bit and MSKC bit of ECMPEM register to “masked”.
2. Set ECMmEST bit in ECMmESET register.
3. Wait until $\overline{\text{ERROROUT\_M}}$ become low by checking that ECMmSSE0_0 bit of ECMmESSTR0 register is "0” and $\overline{\text{ERROROUT\_C}}$ become low by checking that PPR bit of the port register is "0”.
4. Set MSKM bit and MSKC bit of ECMPEM register to “not masked”.

45.3.3 ECMmESETn — ECM Master/Checker Error Set Trigger Register n

The ECM master/checker error set trigger register is for triggering the port safe state function. When ECMmEST bit is set to 1, the configured pins with port register enters the safe state. When error trigger is set, each port state will be changed to safe state according to user's setting. Regarding to port side setting, refer to **Section 2, Pin Functions**. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register** for details of key code protection. This register is always read as $0000\ 0000_H$.

Access: This register is a write-only register that can be written in 32-bit units.

Address: ECMMESETn: <ECMM_base> + n × 8_H + 008_H
ECMCESETn: <ECMC_base> + n × 8_H + 008_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECMmEST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 45.18 ECMmESETn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ECMmEST	Error set trigger bit 0: Writing 0 is invalid 1: Triggers to enter the port safe state.

CAUTIONS

Setting the error trigger via ECMmESETn register will set ECMmSSE0_20 bit of ECMmESSTR0 register (ECM compare error). Therefore, ECMmESETn register has to be set following the sequence below.

1. Set MSKM bit and MSKC bit of ECMPEM register to "masked".
2. Set ECMmEST bit in ECMmESETn register.
3. Wait until pin level become expected value by checking that PPR bit of the port register is "0" or "1".
4. Set MSKM bit and MSKC bit of ECMPEM register to "not masked".

45.3.4 ECMmECLR — ECM Master/Checker Error Clear Trigger Register

The ECM master/checker error clear trigger register is for setting the error signal from the error pin to the high level (toggle). When the ECMmECT bit is set to 1, the error pin outputs the high level (toggle) as long as there are no other sources that set the error pin to the low level. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register** for details of key code protection. This register is always read as $0000\ 0000_H$.

Access: This register is a write-only register that can be written in 32-bit units.

Address: ECMMECLR: <ECMM_base> + 004_H
ECMCECLR: <ECMC_base> + 004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECMm ECT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 45.19 ECMmECLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ECMmECT	Error clear trigger bit 0: Writing 0 is invalid 1: Sets the output level from the error pin to the inactive (high) level.

CAUTIONS

Clearing of the error output pin is only possible if all errors, not masked by ECMEMKj registers, are cleared beforehand.

Clearing the error output via ECMmECLR register will set ECMmSSE0_20 bit of ECMmESSTR0 register (ECM compare error). Therefore, ECMmECLR register has to be set following the sequence below.

1. Set MSKM bit and MSKC bit of ECMPEM register to “masked”.
2. Set ECMmECT bit in ECMmECLR register.
3. Wait until $\overline{\text{ERROROUT\_M}}$ become high by reading ECMmSSE0_0 bit of ECMmESSTR0 register 5 times and $\overline{\text{ERROROUT\_C}}$ become high by reading PPR bit of the port register 5 times. After that, check that ECMmSSE0_0 bit of ECMmESSTR0 register and PPR bit of the port register are “1”. If ECMmSSE0_0 bit of ECMmESSTR0 register and PPR bit of the port register are not “1”, a new error may have occurred.
4. Set MSKM bit and MSKC bit of ECMPEM register to “not masked”.

Note: This procedure is used when ECMEOCCFG register is not set. If the Error output invalidation counter is still running, the Error output clear function is masked and Error output clear requests are ignored. Error output clear function is executed, after the Error output invalidation counter has expired and the Error output clear register is written.

45.3.5 ECMmECLRn — ECM Master/Checker Error Clear Trigger Register n

The ECM master/checker error clear trigger register is for clearing the port safe state function. When ECMmECT bit is set to 1, the configured pins with port register release from port safe state as long as there are no other sources that set to enter the port safe state. For the configuration flow of port safe state mode setting, see **Section 2.6.5, Port Safe State Function Setting**. For the configuration flow of port safe state mode setting, see **Section 2, Pin Functions**. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register** for details of key code protection. This register is always read as 0000 0000_H.

Access: This register is a write-only register that can be written in 32-bit units.

Address: ECMMECLRn: <ECMM_base> + n × 8_H + $00C_H$
ECMCECLRn: <ECMC_base> + n × 8_H + $00C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECMmECT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 45.20 ECMmECLRn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	ECMmECT	Error clear trigger bit 0: Writing 0 is invalid 1: Releases from the port safe state.

CAUTIONS

Clearing of the error triggers is only possible if all errors, not masked by ECMETMKn_j registers, are cleared beforehand.

Clearing the error triggers via ECMmECLRn registers will set ECMmSSE0_20 bit of ECMmESSTR0 register (ECM compare error). Therefore, ECMmECLRn registers have to be set following the sequence below.

1. Set MSKM bit and MSKC bit of ECMPEM register to “masked”.
2. Set ECMmECT bit in ECMmECLRn register.
3. Wait until pin level become expected value by reading PPR bit of the port register. After that, check that PPR bit of the port register is "0" or "1". If PPR bit of the port register is not expected value, a new error may have occurred.
4. Set MSKM bit and MSKC bit of ECMPEM register to “not masked”.

Note: This procedure is used when ECMETCCFGn register is not set. If the Error trigger invalidation counter is still running, the Error trigger clear function is masked and Error trigger clear requests are ignored. Error trigger clear function is executed, after the Error trigger invalidation counter has expired and the Error trigger clear register is written.

45.3.6 ECMmESSTRj — ECM Master/Checker Error Source Status Register j

The ECM master/checker error source status register is read-only register.

This register represents the status of individual internal error sources, and is unrelated to the setting of the error mask.

ECMmSSEs are provided for the each status of individual internal error sources.

Access: This register is a read-only register that can be read in 32-bit units.

Address: ECMMESSTRj: <ECMM_base> + j × 4_H + 048_H
ECMCESSTRj: <ECMC_base> + j × 4_H + 048_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMmSSEj_31	ECMmSSEj_30	ECMmSSEj_29	ECMmSSEj_28	ECMmSSEj_27	ECMmSSEj_26	ECMmSSEj_25	ECMmSSEj_24	ECMmSSEj_23	ECMmSSEj_22	ECMmSSEj_21	ECMmSSEj_20	ECMmSSEj_19	ECMmSSEj_18	ECMmSSEj_17	ECMmSSEj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMmSSEj_15	ECMmSSEj_14	ECMmSSEj_13	ECMmSSEj_12	ECMmSSEj_11	ECMmSSEj_10	ECMmSSEj_9	ECMmSSEj_8	ECMmSSEj_7	ECMmSSEj_6	ECMmSSEj_5	ECMmSSEj_4	ECMmSSEj_3	ECMmSSEj_2	ECMmSSEj_1	ECMmSSEj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.21 ECMmESSTRj Register Contents*1

Bit Position	Bit Name	Function
31 to 0	ECMmSSEj_31 to ECMmSSEj_0	Error source status bit ECMmSSEj_31 to ECMmSSEj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Error not occurred 1: Error occurred

Note 1. ECMmESSTR0.ECMmSSE0_0 shows the status of the $\overline{\text{ERROROUT\_M}}$ pin.

45.3.7 ECMEPCFG — ECM Error Pulse Configuration Register

The ECM error pulse configuration register is a read/write register. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + $0000\ 0000_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECMSL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 45.22 ECMEPCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECMSL0	Error pin operation configuration bit Error output operation setting for the error pin 0: Non-dynamic mode 1: Dynamic mode*1

Note 1. In Dynamic mode the timer output OSTM0 determines the output wave of the $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pin in case of no error.

CAUTION

After setting the dynamic mode, it is recommended not to change to non-dynamic mode again, because this can cause a glitch in the error output.

45.3.8 ECMISCFGj — ECM Interrupt type Selection Configuration Register j

The ECM interrupt type selection configuration register is used to select the generation type of INTECMMI and INTECMDCLSMI interrupts (EI level interrupt), or INTECMNMI interrupt (FE level interrupt). Writing to these registers are protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + j × 4_H + 004_H

Value after reset: $0000\ 0000_H$

ECMISCFG0 to ECMISCFG6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMISj_31	ECMISj_30	ECMISj_29	ECMISj_28	ECMISj_27	ECMISj_26	ECMISj_25	ECMISj_24	ECMISj_23	ECMISj_22	ECMISj_21	ECMISj_20	ECMISj_19	ECMISj_18	ECMISj_17	ECMISj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMISj_15	ECMISj_14	ECMISj_13	ECMISj_12	ECMISj_11	ECMISj_10	ECMISj_9	ECMISj_8	ECMISj_7	ECMISj_6	ECMISj_5	ECMISj_4	ECMISj_3	ECMISj_2	ECMISj_1	ECMISj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.23 ECMISICFG0 to ECMISICFG6 Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMISj_31 to ECMISj_0	ECM maskable (EI level)/maskable (FE level) interrupt generation selection bit ECMISj_31 to ECMISj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Maskable (EI level) interrupt 1: Maskable (FE level) interrupt

ECMISCFG7 to ECMISCFG10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMISj_31	ECMISj_30	ECMISj_29	ECMISj_28	ECMISj_27	ECMISj_26	ECMISj_25	ECMISj_24	ECMISj_23	ECMISj_22	ECMISj_21	ECMISj_20	ECMISj_19	ECMISj_18	ECMISj_17	ECMISj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMISj_15	ECMISj_14	ECMISj_13	ECMISj_12	ECMISj_11	ECMISj_10	ECMISj_9	ECMISj_8	ECMISj_7	ECMISj_6	ECMISj_5	ECMISj_4	ECMISj_3	ECMISj_2	ECMISj_1	ECMISj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.24 ECMISICFG7 to ECMISICFG10 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 1	ECMISj_31 to ECMISj_1	ECM maskable (EI level)/maskable (FE level) interrupt generation selection bit ECMISj_31 to ECMISj_1 correspond to error sources [j × 32 + 31] to [j × 32 + 1]. 0: Maskable (EI level) interrupt 1: Maskable (FE level) interrupt

Table 45.24 ECMISICFG7 to ECMISICFG10 Register Contents (2/2)

Bit Position	Bit Name	Function
0	ECMISj_0	DCLS error interrupt (EI level) and ECM maskable (EI level)/maskable (FE level) interrupt generation selection bit ECMISj_0 correspond to error source [j × 32 + 0]. 0: DCLS error interrupt (EI level) interrupt for own core and Maskable (EI level) interrupt for other core 1: Maskable (FE level) interrupt

45.3.9 ECMINCFGi_j — ECM Interrupt Notification Configuration Register i j

The ECM interrupt notification configuration register is used to set the notification of interrupt. This register is prepared for each core and the interrupt generation enable/disable of each error can be configured. This register must be set after configure ECMISCFGj register, otherwise ECM will notify to each CPUs with different interrupt type in case of error is detected during ECM configuration for interrupt. Writing to these registers are protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + i × $3C_H$ + j × 4_H + 040_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMIEi_j_31	ECMIEi_j_30	ECMIEi_j_29	ECMIEi_j_28	ECMIEi_j_27	ECMIEi_j_26	ECMIEi_j_25	ECMIEi_j_24	ECMIEi_j_23	ECMIEi_j_22	ECMIEi_j_21	ECMIEi_j_20	ECMIEi_j_19	ECMIEi_j_18	ECMIEi_j_17	ECMIEi_j_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMIEi_j_15	ECMIEi_j_14	ECMIEi_j_13	ECMIEi_j_12	ECMIEi_j_11	ECMIEi_j_10	ECMIEi_j_9	ECMIEi_j_8	ECMIEi_j_7	ECMIEi_j_6	ECMIEi_j_5	ECMIEi_j_4	ECMIEi_j_3	ECMIEi_j_2	ECMIEi_j_1	ECMIEi_j_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.25 ECMINCFGi_j Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMIEi_j_31 to ECMIEi_j_0	ECM interrupt notification control bit ECMIEi_j_31 to ECMIEi_j_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Interrupt generation disabled 1: Interrupt generation enabled

45.3.10 ECMIRCFGj — ECM Internal Reset Configuration Register j

The ECM internal reset configuration register is used to set the generation of Error Control Module Reset in response to internal errors. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + j × 4_H + 220_H

Value after reset: 0000 0000_H (ECMIRCFG0 to ECMIRCFG6 and ECMIRCFG8 to ECMIRCFG10), 0000 0010_H (ECMIRCFG7)

ECMIRCFG0 to ECMIRCFG6 and ECMIRCFG8 to ECMIRCFG10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMIREj_31	ECMIREj_30	ECMIREj_29	ECMIREj_28	ECMIREj_27	ECMIREj_26	ECMIREj_25	ECMIREj_24	ECMIREj_23	ECMIREj_22	ECMIREj_21	ECMIREj_20	ECMIREj_19	ECMIREj_18	ECMIREj_17	ECMIREj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMIREj_15	ECMIREj_14	ECMIREj_13	ECMIREj_12	ECMIREj_11	ECMIREj_10	ECMIREj_9	ECMIREj_8	ECMIREj_7	ECMIREj_6	ECMIREj_5	ECMIREj_4	ECMIREj_3	ECMIREj_2	ECMIREj_1	ECMIREj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.26 ECMIRCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMIREj_31 to ECMIREj_0	ECM internal reset generation control bit ECMIREj_31 to ECMIREj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled

ECMIRCFG7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMIRE7_31	ECMIRE7_30	ECMIRE7_29	ECMIRE7_28	ECMIRE7_27	ECMIRE7_26	ECMIRE7_25	ECMIRE7_24	ECMIRE7_23	ECMIRE7_22	ECMIRE7_21	ECMIRE7_20	ECMIRE7_19	ECMIRE7_18	ECMIRE7_17	ECMIRE7_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMIRE7_15	ECMIRE7_14	ECMIRE7_13	ECMIRE7_12	ECMIRE7_11	ECMIRE7_10	ECMIRE7_9	ECMIRE7_8	ECMIRE7_7	ECMIRE7_6	ECMIRE7_5	ECMIRE7_4	ECMIRE7_3	ECMIRE7_2	ECMIRE7_1	ECMIRE7_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.27 ECMIRCFG7 Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMIRE7_31 to ECMIRE7_0	ECM internal reset generation control bit ECMIRE7_31 to ECMIRE7_0 correspond to error sources 255 to 224. 0: ECM internal reset generation disabled 1: ECM internal reset generation enabled

45.3.11 ECMEMKj — ECM Error Mask Register j

The ECM error mask register is used to mask the individual error sources of the error pin output. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + j × 4_H + $25C_H$

Value after reset: 0000 0000_H

ECMEMK0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMEMK0_31	ECMEMK0_30	ECMEMK0_29	ECMEMK0_28	ECMEMK0_27	ECMEMK0_26	ECMEMK0_25	ECMEMK0_24	ECMEMK0_23	ECMEMK0_22	ECMEMK0_21	ECMEMK0_20	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ECMEMK0_11	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 45.28 ECMEMK0 Register Contents

Bit Position	Bit Name	Function
31 to 20	ECMEMK0_31 to ECMEMK0_20	ECM error output signal mask control bit ECMEMK0_31 to ECMEMK0_20 correspond to error sources 31 to 20. 0: Error output is not masked 1: Error output is masked
19 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	ECMEMK0_11	ECM error output signal mask control bit ECMEMK0_11 correspond to delay timer overflow for error output. 0: Error output is not masked 1: Error output is masked
10 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

ECMEMK1 to ECMEMK10

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMEMKj_31	ECMEMKj_30	ECMEMKj_29	ECMEMKj_28	ECMEMKj_27	ECMEMKj_26	ECMEMKj_25	ECMEMKj_24	ECMEMKj_23	ECMEMKj_22	ECMEMKj_21	ECMEMKj_20	ECMEMKj_19	ECMEMKj_18	ECMEMKj_17	ECMEMKj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMEMKj_15	ECMEMKj_14	ECMEMKj_13	ECMEMKj_12	ECMEMKj_11	ECMEMKj_10	ECMEMKj_9	ECMEMKj_8	ECMEMKj_7	ECMEMKj_6	ECMEMKj_5	ECMEMKj_4	ECMEMKj_3	ECMEMKj_2	ECMEMKj_1	ECMEMKj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.29 ECMEMK1 to ECMEMK10 Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMEMKj_31 to ECMEMKj_0	ECM error output signal mask control bit ECMEMKj_31 to ECMEMKj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Error output is not masked 1: Error output is masked

45.3.12 ECMETMKn_j— ECM Error Trigger n Mask Register j

The ECM error trigger n mask register is used to mask the individual error sources of the error trigger n to enter the port safe state. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + n × $3C_H$ + j × 4_H + 298_H

Value after reset: FFFF $FFFF_H$

ECMETMKn_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM ETMK n_0_31	ECM ETMK n_0_30	ECM ETMK n_0_29	ECM ETMK n_0_28	ECM ETMK n_0_27	ECM ETMK n_0_26	ECM ETMK n_0_25	ECM ETMK n_0_24	ECM ETMK n_0_23	ECM ETMK n_0_22	ECM ETMK n_0_21	ECM ETMK n_0_20	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.30 ECMETMKn_0 Register Contents

Bit Position	Bit Name	Function
31 to 20	ECMETMKn_0_31 to ECMETMKn_0_20	ECM error trigger mask control bit ECMETMKn_0_31 to ECMETMKn_0_20 correspond to error sources 31 to 20. 0: Error trigger is not masked 1: Error trigger is masked
19 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

ECMETMKn_j except ECMETMKn_0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM ETMK n_j_31	ECM ETMK n_j_30	ECM ETMK n_j_29	ECM ETMK n_j_28	ECM ETMK n_j_27	ECM ETMK n_j_26	ECM ETMK n_j_25	ECM ETMK n_j_24	ECM ETMK n_j_23	ECM ETMK n_j_22	ECM ETMK n_j_21	ECM ETMK n_j_20	ECM ETMK n_j_19	ECM ETMK n_j_18	ECM ETMK n_j_17	ECM ETMK n_j_16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM ETMK n_j_15	ECM ETMK n_j_14	ECM ETMK n_j_13	ECM ETMK n_j_12	ECM ETMK n_j_11	ECM ETMK n_j_10	ECM ETMK n_j_09	ECM ETMK n_j_08	ECM ETMK n_j_07	ECM ETMK n_j_06	ECM ETMK n_j_05	ECM ETMK n_j_04	ECM ETMK n_j_03	ECM ETMK n_j_02	ECM ETMK n_j_01	ECM ETMK n_j_00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.31 ECMETMKn_j except ECMETMKn_0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMETMKn_j_ 31 to ECMETMKn_j_0	ECM error trigger mask control bit ECMETMKn_j_31 to ECMETMKn_j_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Error trigger is not masked 1: Error trigger is masked

45.3.13 ECMESSTCj — ECM Error Source Status Clear Trigger Register j

The ECM error source status clear trigger register is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <ECM_base> + j × 4_H + 478_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMCLSSEj_31	ECMCLSSEj_30	ECMCLSSEj_29	ECMCLSSEj_28	ECMCLSSEj_27	ECMCLSSEj_26	ECMCLSSEj_25	ECMCLSSEj_24	ECMCLSSEj_23	ECMCLSSEj_22	ECMCLSSEj_21	ECMCLSSEj_20	ECMCLSSEj_19	ECMCLSSEj_18	ECMCLSSEj_17	ECMCLSSEj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMCLSSEj_15	ECMCLSSEj_14	ECMCLSSEj_13	ECMCLSSEj_12	ECMCLSSEj_11	ECMCLSSEj_10	ECMCLSSEj_9	ECMCLSSEj_8	ECMCLSSEj_7	ECMCLSSEj_6	ECMCLSSEj_5	ECMCLSSEj_4	ECMCLSSEj_3	ECMCLSSEj_2	ECMCLSSEj_1	ECMCLSSEj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.32 ECMESSTCj Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMCLSSEj_31 to ECMCLSSEj_0	ECM error status clear bit ECMCLSSEj_31 to ECMCLSSEj_0 correspond to ECMmSSEj_31 to ECMmSSEj_0. 0: Corresponding error status unchanged 1: Corresponding error status cleared

45.3.14 ECMKCPROT — ECM Key Code Protection Register

The ECM key code protection register is used for protection against writing operation to the configuration registers.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + $4B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	KCPROT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KCPROT[15:1]															KCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	R/W

Table 45.33 ECMKCPROT Register Contents

Bit Position	Bit Name	Function
31 to 1	KCPROT[31:1]	Enable or disable modification of the KCE bit. The written value is not retained. These bits are always read as 0.*1
0	KCE	Key Code Enable bit 0: Disable write access to protected registers 1: Enable write access to protected registers

Note 1. Write $A5A5A500_H$ to this register to disable writing protected registers.
Write $A5A5A501_H$ to this register to enable writing protected registers.

45.3.15 ECMPEj — ECM Pseudo Error Trigger Register j

The ECM pseudo error trigger register is write-only register. This register is used to generate a error for Software alarm 0 to 3 (Error number 192 to 195) and a pseudo error for test purpose. The ECM operation in response to the generation of a pseudo error is identical to that of a real error source. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register is a write-only register that can be written in 32-bit units.

Address: <ECM_base> + j × 4_H + $4B8_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECM PEj_31	ECM PEj_30	ECM PEj_29	ECM PEj_28	ECM PEj_27	ECM PEj_26	ECM PEj_25	ECM PEj_24	ECM PEj_23	ECM PEj_22	ECM PEj_21	ECM PEj_20	ECM PEj_19	ECM PEj_18	ECM PEj_17	ECM PEj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECM PEj_15	ECM PEj_14	ECM PEj_13	ECM PEj_12	ECM PEj_11	ECM PEj_10	ECM PEj_9	ECM PEj_8	ECM PEj_7	ECM PEj_6	ECM PEj_5	ECM PEj_4	ECM PEj_3	ECM PEj_2	ECM PEj_1	ECM PEj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.34 ECMPEj Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMPEj_31 to ECMPEj_0	ECM pseudo error trigger bit ECMPEj_31 to ECMPEj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Pseudo error is not generated. 1: Pseudo error is generated.

45.3.16 ECMDTMCTL — ECM Delay Timer Control Register

The ECM delay timer control register is a read/write register. This register is used to control the delay timer. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + $4F4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DTMSTACNTCLK	—	—	DTMSTP	DTMSTA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	R/W

Table 45.35 ECMDTMCTL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	DTMSTACNTCLK	Delay timer start confirmation status. 0: Delay timer does not start 1: Delay timer starts.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DTMSTP	Delay timer stop bit By writing “1” to this bit, delay timer is stopped (writing 0 is ignored). Simultaneously, DTMSTA bit will be 0. 0: Delay timer is completed or stop request is not in progress. 1: Stop request for delay timer is in progress.
0	DTMSTA	Delay timer start bit Specifies the operation of the delay timer at the occurrence of an error event. 0: Delay timer does not start 1: Delay timer starts

CAUTIONS

1. **ECMDTMCTL register can be accessed via P-Bus but delay timer runs with not P-Bus clock but dedicated counter clock.**
   Therefore, time lag exists between writing of ECMDTMCTL and enabling of delay timer.
   DTMSTACNTCLK can be used to confirm whether delay timer is enabled or not.
   Reconfirm that DTMSTA has been updated by checking DTMSTACNTCLK after writing to DTMSTA.
2. **ECMDTMCTL register can be written only when (DTMSTA, DTMSTACNTCLK) = (0, 0) or (1, 1).**
   Confirm the values of DTMSTA and DTMSTACNTCLK before writing to ECMDTMCTL.
3. **The delay timer needs 3 clocks of cntclk until it stops an overflow certainly. Consider to set the sufficient value to ECMDTMCMP[23:0].**

45.3.17 ECMDTMR — ECM Delay Timer Register

The ECM delay timer register is a read-only register. The ECM delay timer register is initialized by setting the DTMSTA bit of the ECM delay timer control register from 1 (timer in operation) to 0 (timer stopped).

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ECM_base> + $4F8_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ECMDTMR[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.36 ECMDTMR Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	ECMDTMR[23:0]	Delay timer counter value.

45.3.18 ECMDTMCMP — ECM Delay Timer Compare Register

The ECM delay timer compare register is a read/write register. ECMmESSTR0.ECMmSSE0_11 bit is set according to configuration of ECMDTMCFGj register when the value of this registers matches with the value of the ECM delay timer register.

Writing data to this register has to be conducted while the delay timer is stopped. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + $4FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CMPW	ECMDTMCMP[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.37 ECMDTMCMP Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CMPW	Indicates whether a write to the counter clock region of ECMDTMCMP register setting is in progress 0: Not in progress 1: In progress. While the bit is set, further setting to ECMDTMCMP is prohibited.
23 to 0	ECMDTMCMP[23:0]	Delay timer compare value

CAUTIONS

1. **ECMDTMCMP register is set via P-bus, however actual delay timer exists in a different clock domain. When ECMDTMCMP is configured, the value is copied across clock domains to reflect to the delay timer, which takes a certain time. CMPW indicates whether reflecting the new ECMDTMCMP value across clock domains is in progress.**
2. **While CMPW is "1", further setting to ECMDTMCMP is ignored. Confirm CMPW = 0, before writing of ECMDTMCMP.**

45.3.19 ECMDTMCFGj — ECM Delay Timer Configuration Register j

The ECM delay timer configuration register is a read/write register. This register is used to set enable/disable of the delay timer start caused by either EI level interrupt or FE level interrupt configured to ECMINCFGi_j and ECMISCFGj registers in response to errors. When delay timer overflow occurs, delay timer overflow event (error factor #11) is raised according to this register configuration. Writing data to this register has to be conducted while the delay timer is stopped. Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + j × 4_H + 500_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMDTMEj_31	ECMDTMEj_30	ECMDTMEj_29	ECMDTMEj_28	ECMDTMEj_27	ECMDTMEj_26	ECMDTMEj_25	ECMDTMEj_24	ECMDTMEj_23	ECMDTMEj_22	ECMDTMEj_21	ECMDTMEj_20	ECMDTMEj_19	ECMDTMEj_18	ECMDTMEj_17	ECMDTMEj_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMEj_15	ECMDTMEj_14	ECMDTMEj_13	ECMDTMEj_12	ECMDTMEj_11	ECMDTMEj_10	ECMDTMEj_9	ECMDTMEj_8	ECMDTMEj_7	ECMDTMEj_6	ECMDTMEj_5	ECMDTMEj_4	ECMDTMEj_3	ECMDTMEj_2	ECMDTMEj_1	ECMDTMEj_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.38 ECMDTMCFGj Register Contents

Bit Position	Bit Name	Function
31 to 0	ECMDTMEj_31 to ECMDTMEj_0	ECM delay timer start control bit ECMDTMEj_31 to ECMDTMEj_0 correspond to error sources [j × 32 + 31] to [j × 32 + 0]. 0: Delay timer start disabled 1: Delay timer start enabled

45.3.20 ECMEOCCFG — ECM Error Output Clear Invalidation Configuration Register

This register is a read/write register and can be written in 32-bit units.

After counter for Error Output clear invalidation exceed the value which is configured to this register, it is possible to clear non-safe status of error output by SW.

Configure to this register only if error output status is safe.

Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + $71C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EOCIEN	—	—	—	—	—	—	CMPW	ECMEOUTCLRT[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMEOUTCLRT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.39 ECMEOCCFG Register Contents

Bit Position	Bit Name	Function
31	EOCIEN	Error Output Clear Invalidation Function Enabled 0: Disabled 1: Enabled
30 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CMPW	Indicates on execution of ECMEOUTCLRT[23:0] register setting to counter clock domain 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMEOUTCLRT is prohibited.
23 to 0	ECMEOUTCLRT [23:0]	The number of clock cycles after which it is possible to clear error output by SW.

CAUTIONS

1. **ECMEOCCFG register is set via P-bus, however actual "output clear invalidation counter" exists in a different clock domain. When ECMEOCCFG is configured, the value is copied across clock domains to reflect to the counter, which takes a certain time. CMPW indicates whether the reflection across clock domains is in progress.**
2. **While CMPW is "1", writing of ECMEOUTCLRT[23:0] is ignored. Confirm CMPW = 0 before writing of ECMEOUTCLRT[23:0].**

45.3.21 ECMETCCFGn — ECM Error Trigger n Clear Invalidation Configuration Register

This register is a read/write register and can be written in 32-bit units.

After counter for Error trigger n clear invalidation exceed the value which is configured to this register, it is possible to clear non-safe status of error trigger by SW.

Configure to this register only if error trigger status is safe.

Writing to this register is protected by ECMKCPROT. Refer to **Section 45.3.14, ECMKCPROT — ECM Key Code Protection Register**, for details of key code protection.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + n × 4_H + 720_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ETCIEN	—	—	—	—	—	—	CMPW	ECMETRGCLRT[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMETRGCLRT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.40 ECMEOCCFGn Register Contents

Bit Position	Bit Name	Function
31	ETCIEN	Error Trigger Invalidation Function Enabled 0: Disabled 1: Enabled
30 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CMPW	Indicates on execution of ECMETRGCLRT[23:0] register setting to counter clock domain 0: Not in progress 1: In progress. While the bit is set, further setting to the ECMETRGCLRT is prohibited.
23 to 0	ECMETRGCLRT [23:0]	The number of clock cycles after which it is possible to clear error trigger by SW.

CAUTIONS

1. **ECMETCCFGn register is set via P-bus, however actual "trigger clear invalidation counter" exists in a different clock domain. When ECMETCCFGn is configured, the value is copied across clock domains to reflect to the counter, which takes a certain time. CMPW indicates whether the reflection across clock domains is in progress.**
2. **While CMPW is "1", writing of ECMETRGCLRT[23:0] is ignored. Confirm CMPW = 0 before writing of ECMETRGCLRT[23:0].**

45.3.22 ECMPEM — ECM Pseudo Error Mask Register

This register can mask the pseudo error of "ECM compare error" to support self-diagnosis of the binding components for $\overline{\text{ERROROUT\_M}}$ and $\overline{\text{ERROROUT\_C}}$ pins and Error triggers.

Access: This register can be read or written in 32-bit units.

Address: <ECM_base> + 740_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSKM	MSKC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 45.41 ECMPEM Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MSKM	0: Pseudo error of "ECM compare error" for ECM master is NOT masked. 1: Pseudo error of "ECM compare error" for ECM master is masked.
0	MSKC	0: Pseudo error of "ECM compare error" for ECM checker is NOT masked. 1: Pseudo error of "ECM compare error" for ECM checker is masked.

Section 46 Data CRC Function K (KCRC)

This section contains a generic description of the Data CRC Function K (KCRC). The first part of this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the KCRC.

46.1 Feature

46.1.1 Number of Units and Channels

This microcontroller has the following number of KCRC units.

Table 46.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)	8 (n = 0 to 7)
Name	KCRCn									

Table 46.2 Index

Index	Description
n	Throughout this section, the individual KCRC units are identified by the index “n” (n = 0 to 7). For example, KCRCnCTL indicates the KCRCn control register.

46.1.2 Register Base Addresses

KCRCn base addresses are listed in the following table. KCRCn register addresses are given as offsets from the base addresses in general.

Table 46.3 Register Base Addresses

Base Address Name	Base Address	Bus Group
<KCRC0_base>	FFD3 4000_H	Peripheral Group 3
<KCRC1_base>	FFBF $F400_H$	Peripheral Group 5
<KCRC2_base>	FFD3 5000_H	Peripheral Group 3
<KCRC3_base>	FFBF $F700_H$	Peripheral Group 5
<KCRC4_base>	FFD3 6000_H	Peripheral Group 3
<KCRC5_base>	FFBF $FA00_H$	Peripheral Group 5
<KCRC6_base>	FFD3 7000_H	Peripheral Group 3
<KCRC7_base>	FFBF $FD00_H$	Peripheral Group 5

46.1.3 Clock Supply

Table 46.4 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name	Description
KCRCn	KCRC module clock	CLK_HSB	Operation clock
	Register access clock	CLK_HSB	Bus clock

For details of clock supply, see **Section 13, Clock Controller**.

46.1.4 Interrupt, DMA/DTS Requests and Error Notifications

This module has no interrupt and DMA/DTS requests.

This module has no error notifications.

46.1.5 Reset Sources

KCRCn reset sources are shown below. KCRCn is initialized by the following reset sources.

Table 46.5 Reset Sources

Unit Name	Register Name	Reset Condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
KCRCn	All registers	√	√	√	√	√	—	—

For details of reset source, see **Section 9, Reset Controller**.

46.1.6 External Input/Output Signals

This module has no external input/output signals.

46.2 Overview

46.2.1 Functional Overview

The Data CRC Function K can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- Supported CRC polynomials (For details, refer to **Table 46.16, Setting Table**)
  - 64-bit CRC64ECMA
    $42F0E1EB\ A9EA3693_H$: $X^{64}+ X^{62}+ X^{57}+ X^{55}+ X^{54}+ X^{53}+ X^{52}+ X^{47}+ X^{46}+ X^{45}+ X^{40}+ X^{39}+ X^{38}+ X^{37}+ X^{35}+ X^{33}+ X^{32}+ X^{31}+ X^{29}+ X^{27}+ X^{24}+ X^{23}+ X^{22}+ X^{21}+ X^{19}+ X^{17}+ X^{13}+ X^{12}+ X^{10}+ X^{9}+ X^{7}+ X^{4}+ X+ 1$
  - 32-bit CRC32P4
    $F4ACFB13_H$: $X^{32}+ X^{31}+ X^{30}+ X^{29}+ X^{28}+ X^{26}+ X^{23}+ X^{21}+ X^{19}+ X^{18}+ X^{15}+ X^{14}+ X^{13}+ X^{12}+ X^{11}+ X^{9}+ X^{8}+ X^{4}+ X+ 1$
  - 32-bit Ethernet CRC
    $04C1\ 1DB7_H$: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$
  - 32-bit CRC32 Reversed Polynomial
    $EDB88320_H$: $X^{32}+ X^{31}+ X^{30}+ X^{29}+ X^{27}+ X^{26}+ X^{24}+ X^{23}+ X^{21}+ X^{20}+ X^{19}+ X^{15}+ X^{9}+ X^{8}+ X^{5}+ 1$
  - 32-bit CRC32C (Castagnoli)
    $1EDC\ 6F41_H$: $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^{9} + X^{8} + X^{6} + 1$
  - 16-bit CCITT CRC
    1021_H: $X^{16} + X^{12} + X^{5} + 1$
  - 16-bit Baicheva00
    $90D9_H$: $X^{16} + X^{15} + X^{12} + X^{7} + X^{6} + X^{4} + X^{3} + 1$
  - 15-bit CRC15CAN
    4599_H: $X^{15}+ X^{14}+ X^{10}+ X^{8}+ X^{7}+ X^{4}+ X^{3}+ 1$
  - 16-bit ARC
    8005_H: $X^{16}+ X^{15}+ X^{2}+ 1$
  - 8-bit SAE J1850 CRC
    $1D_H$: $X^{8} + X^{4} + X^{3} + X^{2} + 1$
  - 8-bit 0x2F CRC
    $2F_H$: $X^{8} + X^{5} + X^{3} + X^{2} + X + 1$
- Selectable input data size. (8/16/32-bit)
- Supported Reflection of Input data.
- Supported Reflection of Output data.
- Supported Reflection of Polynomial.
- Supported XOR Mask of output data.
- CRC generation to an arbitrary data block length by register.
- After initialization of the KCRC data register, every write access to the KCRC input register generates a new CRC according to the chosen polynomial and the result is stored in the KCRC data register.
- Supported AUTOSAR 4.3.0.

46.2.2 Block Diagram

The following figure shows the block diagram of the Data CRC Function K.

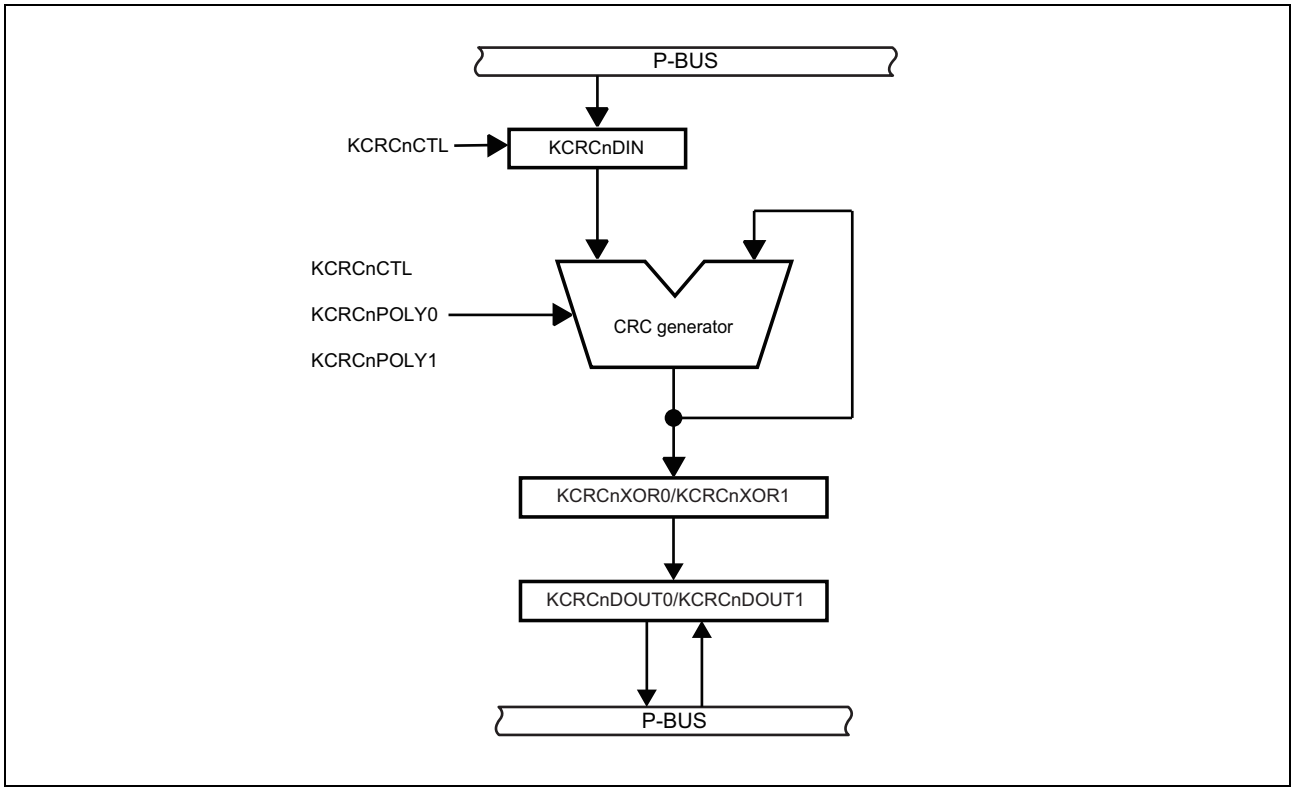


Figure 46.1 Block Diagram of Data CRC Function K

46.3 Registers

46.3.1 List of Registers

KCRC registers are listed in the table below.

For details about <KCRCn_base>, see **Section 46.1.2, Register Base Addresses**.

Table 46.6 List of Registers

Module Name	Register Name	Symbol	Address	Access Protection	
				PBG	Other
KCRCn	KCRCn data input register	KCRCnDIN	<KCRCn_base> + 00_H	*1	—
	KCRCn data output register (Lower)	KCRCnDOUT0	<KCRCn_base> + 80_H	*1	—
	KCRCn data output register (Upper)	KCRCnDOUT1	<KCRCn_base> + 84_H	*1	—
	KCRCn control register	KCRCnCTL	<KCRCn_base> + 90_H	*1	—
	KCRCn Polynomial register (Lower)	KCRCnPOLY0	<KCRCn_base> + $A0_H$	*1	—
	KCRCn Polynomial register (Upper)	KCRCnPOLY1	<KCRCn_base> + $A4_H$	*1	—
	KCRCn XOR mask register (Lower)	KCRCnXOR0	<KCRCn_base> + $B0_H$	*1	—
	KCRCn XOR mask register (Upper)	KCRCnXOR1	<KCRCn_base> + $B4_H$	*1	—

Note 1. n = 0: PBG30#13
n = 1: PBG52#2
n = 2: PBG30#14
n = 3: PBG52#3
n = 4: PBG30#15
n = 5: PBG52#4
n = 6: PBG31#0
n = 7: PBG52#5

46.3.2 KCRCnDIN — KCRCn Data Input Register

This register holds the input data for the CRC calculation. The effective bit width used for CRC calculation must be set by KCRCnCTL.DW[2:0].

When data is written to this register, the CRC code is generated. The CRC calculation is immediately started after the KCRCnDIN register is written. The KCRCnDOUT register must be initialized, with the initial value, before the first data of the data block is written to KCRCnDIN register.

Access: This register can be read or written in 32/16/8-bit units.

Address: <KCRCn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DIN[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DIN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.7 KCRCnDIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DIN[31:0]	Input data for CRC calculation. The valid bits are controlled by KCRCnCTL.DW[2:0]: For 32 bits, effective bit width: DIN[31:0] For 16 bits, effective bit width: DIN[15:0] For 8 bits, effective bit width: DIN[7:0]

46.3.3 KCRCnDOUT0 — KCRCn data output register (Lower)

This register stores the result of the CRC code generated by the CRC polynomial selected by KCRCnPOLY0.POLY[31:0].

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + 80_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DOUT[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DOUT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.8 KCRCnDOUT0 Register Contents

Bit Position	Bit Name	Function
31 to 0	DOUT[31:0]	Result of the CRC code generation. The valid bits are controlled by KCRCnCTL.PSIZE[5:0]: For 32-bit effective bit width: DOUT[31:0] For 16-bit effective bit width: DOUT[15:0] For 15-bit effective bit width: DOUT[14:0] For 8-bit effective bit width: DOUT[7:0] When reading from this register, the value is XORed by hardware with the value specified in **Table 46.16, Setting Table** for the chosen polynomial. Therefore, for the CRC32 polynomial (as default setting), 0000 0000_H is read from this register even in the initial state.

CAUTION

The read value after reset is 0000 0000_H, since the CRC32 polynomial is selected as the CRC generating function after reset and the value is XORed onto FFFF $FFFF_H$ by hardware.

This register must be initialized with the start value before the first data of the data block is written to KCRCnDIN register.

46.3.4 KCRCnDOUT1 — KCRCn data output register (Upper)

This register stores the result of the CRC code generated by CRC polynomial selected by KCRCnPOLY1.POLY[63:32].

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + 84_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DOUT[63:49]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DOUT[48:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.9 KCRCnDOUT1 Register Contents

Bit Position	Bit Name	Function
31 to 0	DOUT[63:32]	Result of the CRC code generation (Upper side of 64 bits width). The valid bits are controlled by KCRCnCTL.PSIZE[5:0]. For 64-bit polynomial size, CRC result upper side from bit 64 to bit 32 will be stored in KCRCnDOUT1 register after CRC calculation. For other polynomial sizes, the value of this register will be undefined after CRC calculation.

CAUTION

The read value after reset is 0000 0000_H, since the CRC32 polynomial is selected as the CRC generating function after reset and the value is XORed onto FFFF $FFFF_H$ by hardware.

This register must be initialized with the start value before the first data of the data block is written to KCRCnDIN register.

46.3.5 KCRCnCTL — KCRCn Control register

This register provide CRC calculate settings. All settings shall use as the **Table 46.16, Setting Table**.

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + 90_H

Value after reset: $001F\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	PSIZE[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CMD0	—	—	CMD1	CMD2	—	DW[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 46.10 KCRCnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	PSIZE[5:0]	Specifies the CRC polynomial size $0x07_H$: 8 bits $0x0E_H$: 15 bits $0x0F_H$: 16 bits $0x1F_H$: 32 bits $0x3F_H$: 64 bits Other: Setting prohibited For details of settings, see **Table 46.16**.
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CMD0	Specifies the CRC calculate mode 0 (for output) 0: Mode N (N = Normal) 1: Mode R (R = Reflect) For details of settings, see **Table 46.16**.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CMD1	Specifies the CRC calculate mode 1 (for input) 0: Mode N (N = Normal) 1: Mode R (R = Reflect) For details of settings, see **Table 46.16**.
4	CMD2	Specifies the CRC calculate mode 2 (for input/output and polynomials) 0: Mode M (M = The byte order of KCRCnDIN and KCRCnPOLY are MSB*1 first) 1: Mode L (L = The byte order of KCRCnDIN and KCRCnPOLY are LSB*2 first) Note 1. MSB = Most Significant Byte Note 2. LSB = Least Significant Byte For details of settings, see **Table 46.16**.

Table 46.10 KCRCnCTL Register Contents (2/2)

Bit Position	Bit Name	Function
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DW[2:0]	Specified the CRC input data size select These bits define valid data width of data input. 000: 32-bit fix mode 001: 16-bit fix mode 011: 8-bit fix mode Other: Setting prohibit

NOTE

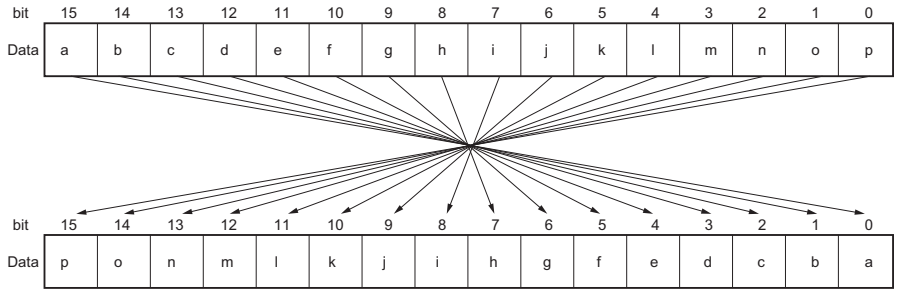
After changing the CRC calculate setting (KCRCnCTL), the KCRCnDOUT[0-1] registers must be initialized.

CAUTION

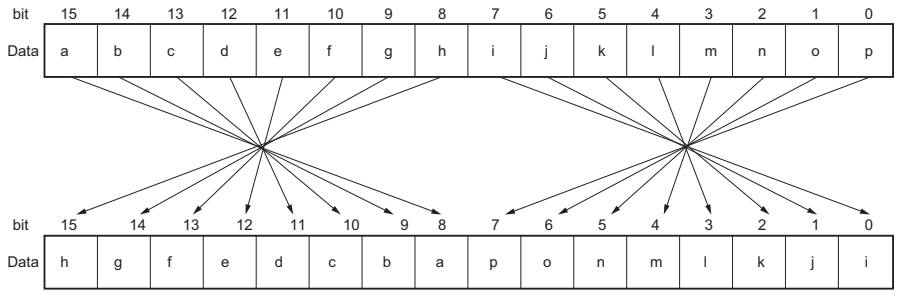
- **The CRC input bit width (KCRCnCTL.DW[2:0]) must be set according to the data block bit width. After the final CRC result is read from KCRCnDOUT[0-1] registers, the KCRCnDOUT[0-1] registers must be initialized with the initial value.**
- **Switching the CRC polynomial (KCRCnPOLY.POLY[0-1][31:0]) is also not allowed during processing of a data block.**

USAGE NOTES

- If CMD0 is setting 1 in PSIZE = 16-bit. Output data is reflected as show below.

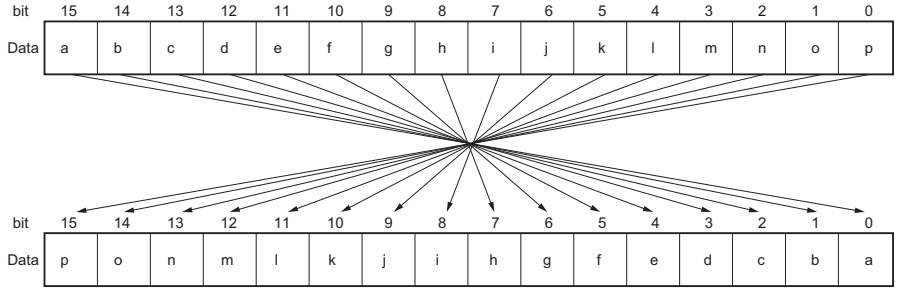


- If CMD1 is setting 1 in DW = 16-bit. Input data is reflected as show below.

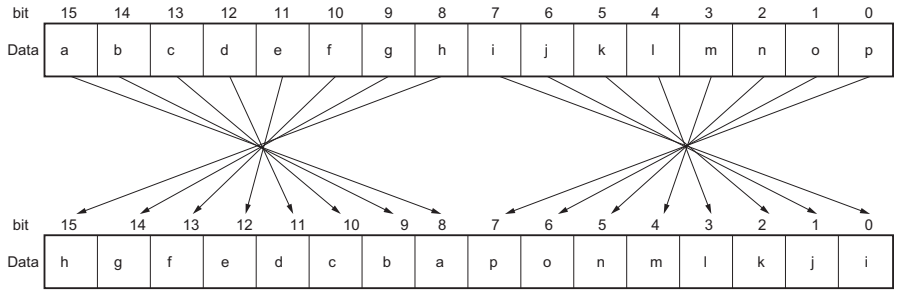


- If CMD2 is setting 1 in PSIZE = 16-bit and DW = 16-bit.

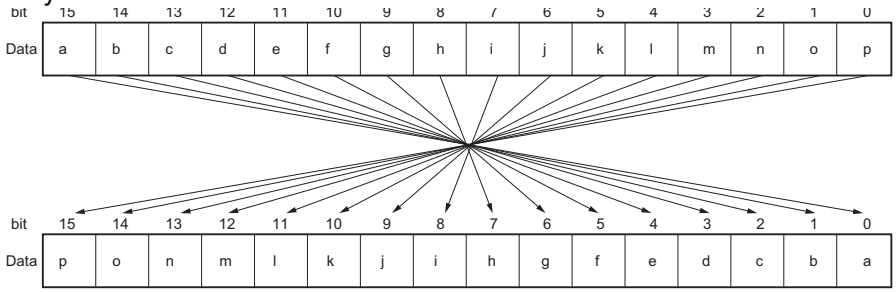
→ Output data is reflected as shown below.



→ Input data is reflected as shown below.



→ Polynomial is reflected as shown below.



46.3.6 KCRCnPOLY0 — KCRCn Polynomial register (Lower)

This register controls the CRC polynomial type for CRC calculation process.

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + $A0_H$

Value after reset: $04C1\ 1DB7_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POLY[31:16]															
Value after reset	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POLY[15:0]															
Value after reset	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.11 KCRCnPOLY0 Register Contents

Bit Position	Bit Name	Function
31 to 0	POLY[31:0]	Specifies the polynomial type for CRC calculation. For details of settings, see **Table 46.16**.

46.3.7 KCRCnPOLY1— KCRCn Polynomial register (Upper)

This register controls the CRC polynomial 64 bits type for CRC calculation process.

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + $A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POLY[63:49]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POLY[48:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.12 KCRCnPOLY1 Register Contents

Bit Position	Bit Name	Function
31 to 0	POLY[63:32]	Specifies the polynomial 64 bits type for CRC calculation. Setting must be able to set only **Table 46.16**.

46.3.8 KCRCnXOR0 — KCRCn XOR mask register (Lower)

This register stores the XOR value of the CRC code specified in the AUTOSAR standard for the chosen polynomial. This value will be XORed with initial value of KCRCnDOUT0 for CRC calculation process.

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + $B0_H$

Value after reset: $FFFF\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XOR[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XOR[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.13 KCRCnXOR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	XOR[31:0]	Specifies the XOR mask for CRC calculation. For details of settings, see **Table 46.16**.

46.3.9 KCRCnXOR1 — KCRCn XOR mask register (Upper)

This register stores the XOR value of the CRC code specified in the AUTOSAR standard for the chosen polynomial. This value will be XORed with initial value of KCRCnDOUT1 for CRC calculation process.

Access: This register can be read or written in 32-bit units.

Address: <KCRCn_base> + $B4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XOR[63:48]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XOR[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.14 KCRCnXOR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	XOR[63:32]	Specifies the XOR mask 64 bits type for Data output. Setting must be **Table 46.16**.

46.4 Operation

KCRC support several types of CRC as follows. All settings shall be used follow.

Table 46.16 Setting Table

Name	Polynomial	input data reflect	output data reflect	XOR	Initial	KCRCnCTL	CHECK*1
0x2F	00000000 0000002F$_H$	No	No	00000000 000000FF$_H$	00000000 000000FF$_H$	00070003$_H$	DF$_H$
SAE_J1850	00000000 0000001D$_H$	No	No	00000000 000000FF$_H$	00000000 000000FF$_H$	00070003$_H$	4B$_H$
CCITT16	00000000 00001021$_H$	No	No	00000000 00000000$_H$	00000000 0000FFFF$_H$	000F0003$_H$	29B1$_H$
CRC32	00000000 04C11DB7$_H$	Yes	Yes	00000000 FFFFFFFF$_H$	00000000 FFFFFFFF$_H$	001F0123$_H$	CBF4 3926$_H$
CRC32 reversed polynomial	00000000 EDB88320$_H$	Yes	Yes	00000000 FFFFFFFF$_H$	00000000 FFFFFFFF$_H$	001F0013$_H$	CBF4 3926$_H$
CRC32C	00000000 1EDC6F41$_H$	Yes	Yes	00000000 FFFFFFFF$_H$	00000000 FFFFFFFF$_H$	001F0123$_H$	E306 9283$_H$
Baicheva00	00000000 000090D9$_H$	No	No	00000000 00000000$_H$	00000000 00000000$_H$	000F0003$_H$	913A$_H$
CRC32P4	00000000 F4ACFB13$_H$	Yes	Yes	00000000 FFFFFFFF$_H$	00000000 FFFFFFFF$_H$	001F0123$_H$	1697 D06A$_H$
ARC	00000000 00008005$_H$	Yes	Yes	00000000 00000000$_H$	00000000 00000000$_H$	000F0123$_H$	BB3D$_H$
CRC15CAN	00000000 00004599$_H$	No	No	00000000 00000000$_H$	00000000 00000000$_H$	000E0003$_H$	059E$_H$
CRC64ECMA	42F0E1EB A9EA3693$_H$	Yes	Yes	FFFFFFFF FFFFFFFF$_H$	FFFFFFFF FFFFFFFF$_H$	003F0123$_H$	995DC9BB DF1939FA$_H$

Note: KCRCnCTL.DW can change by user software.

Note 1. Check value can provide check sequence. See **Section 46.4, (1) Check** .
The Polynomial shall set to KCRCnPOLY0 and KCRCnPOLY1 registers.
The XOR shall set to KCRCnXOR0 and KCRCnXOR1 registers.
The Initial value shall set to KCRCnDOUT0 and KCRCnDOUT1 registers.

(1) Check

This field is a check value that can be used as a weak validator of implementations of the algorithm. The field contains the checksum obtained when the ASCII values '1' '2' '3' '4' '5' '6' '7' '8' '9' corresponding to values 31_H, 32_H, 33_H, 34_H, 35_H, 36_H, 37_H, 38_H, 39_H is fed through the specified algorithm.

(2) Data flow

The Data CRC Function K generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the Data CRC Function in 8-, 16-, 32-bit or 64-bit units. The CRC polynomial can either be selected for 64-bit CRC64ECMA, 32-bit Ethernet, 32-bit CRC32C, 16-bit CCITT, 16-bit Baicheva00, 8-bit SAE J1850 or 8-bit 0x2F, etc, .. polynomial CRC, the initial starting value must be set at the KCRCnDOUT0 and KCRCnDOUT1 registers before the first write access to the KCRC input register (KCRBnDIN) is performed.

After the last write access to the KCRCnDIN register is performed, the result can be read-out from the KCRCnDOUT0 and KCRCnDOUT1 registers after one clock period.

The flow chart below shows the CRC generating procedure.

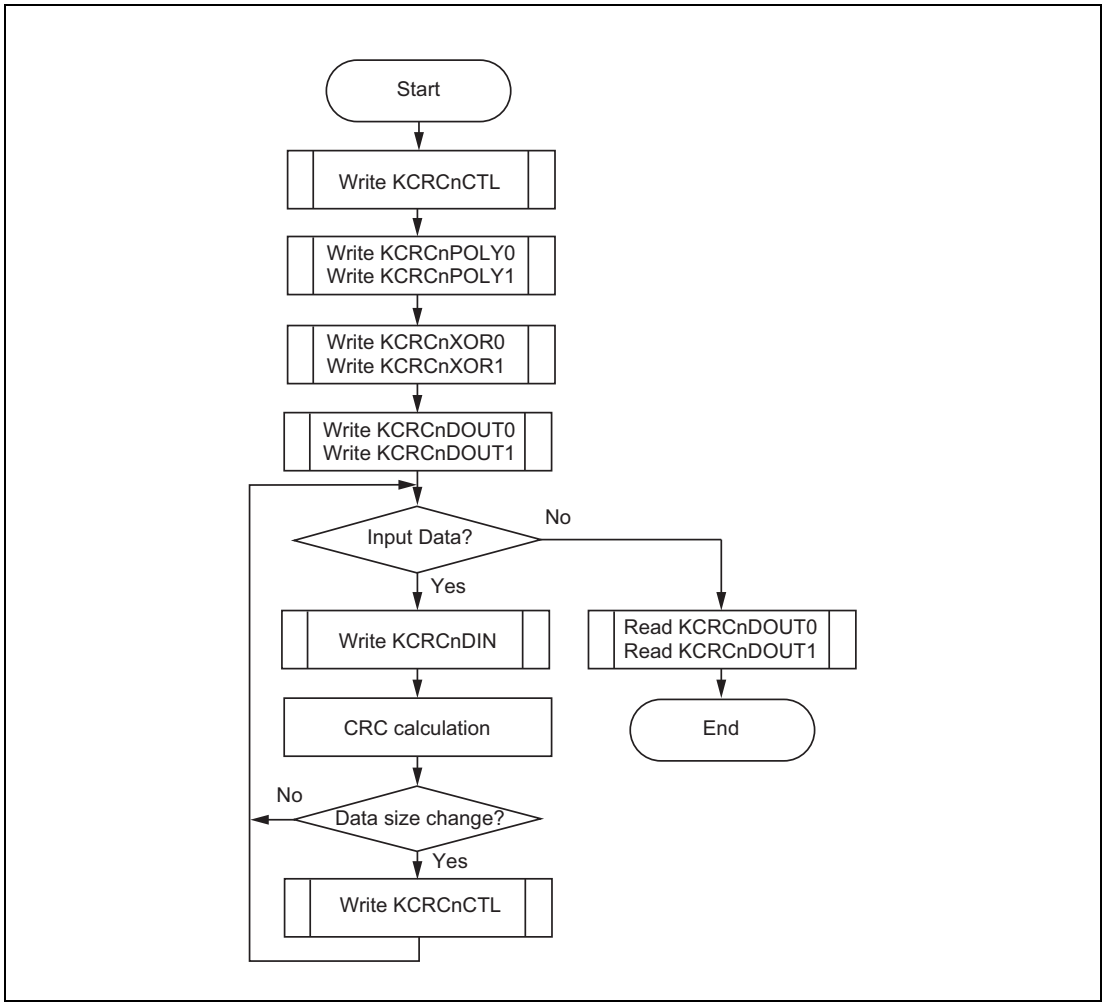


Figure 46.2 Data Flow

NOTES

- All settings shall be to set as **Table 46.16**
- The initial value that has to be written into KCRCnDOUT0 and KCRCnDOUT1 can be chosen by the user. To be compliant to the AUTOSAR standard the initial value has to be chosen according to the value specified for the chosen polynomial.
- When reading from register KCRCnDOUT0 and KCRCnDOUT1, the value is XORed by hardware with the value specified in the AUTOSAR standard for the chosen polynomial.
- Generation of CRC code and storage to KCRCnDOUT0 and KCRCnDOUT1 are done by hardware.
- All registers are accessible by CPU or by CPU-independent read/write access via DMA/DTS. Independent from access of the KCRCn via CPU or DMA/DTS the verification of the CRC signature (CRC-check result in KCRCnDOUT0 and KCRCnDOUT1) has to be performed by SW.

(3) Sample data generation sequence

(Example: Input data is as follows,

Data input1 = 34333231_H

Data input2 = 38373635_H

Data input3 = 39_H)

- Case: check sequence of CCITT 16

```
Write32 KCRCnCTL      = 000F0000H ;  // input data size 32-bit mode
Write32 KCRCnPOLY0    = 00001021H ;  // Polynomial setting
Write32 KCRCnPOLY1    = 00000000H ;  // Polynomial setting
Write32 KCRCnXOR0     = 00000000H ;  // xor mask setting
Write32 KCRCnXOR1     = 00000000H ;  // xor mask setting
Write32 KCRCnDOUT0    = 0000FFFFH ;  // initialize
Write32 KCRCnDOUT1    = 00000000H ;  // initialize
Write32 KCRCnDIN      = 34333231H ;  // Data input 1 (32-bit access )
Write32 KCRCnDIN      = 38373635H ;  // Data input 2 (32-bit access)
Write32 KCRCnCTL      = 000F0003H ;  // input data size change to 8-bit mode.
Write8  KCRCnDIN      = 39H ;        // Data input 3 (8-bit access) ONLY KCRCnCTL.DW is
                                       able to change.
Read32  KCRCnDOUT0;                  // Return check value (KCRCnCTL.DOUT0 show
                                       000029B1H)
```

Note: KCRCnCTL.DW[2:0] can change in processing for less than 4 bytes. Usually, KCRCnCTL.DW set to 4 byte mode for data transfer efficiency.

Section 47 Basic Hardware Protection (BHP)

To protect the code flash memory, data flash memory and Flash extra area, this product has several Flash protection functions.

The security function that is used by the ID authentication and setting of Flash Option Byte is described in this section. The ID code and the Flash Option Byte that is described in this section are mapped at Security Settings area and Block Protection Settings area.

In the initial state of the product at shipment, the ID code are set by all - 1. These ID should be set at the first programming to Flash Option Byte. For details of the programming to the flash memory, see **Section 51, Flash Memory**.

47.1 Features

47.1.1 Functional features

(1) Mode and connection entry protection

- System requires some password base authentication to enter other than Normal operation mode.
- Authentication ID and related option can be configured by Security settings area.
- The 256 bit SPID password authentication is needed to enter Serial programming mode.
- The 256 bit OCDID password authentication is needed when On-chip debug.
- The RHSIF ID authentication is needed to connect to RHSIF Link Partner [For U2A-EVA/ U2A16/U2A8 only].

(2) Flash Protection by ID authentication

- The system requires 256 bit Customer ID authentication for re-programming to User area by each block.
- The Customer ID authentication is needed to read the User/User boot area when debugger is connected.
- The system requires 256 bit Data Flash ID authentication for re-programming to Data area.
- The Data Flash ID authentication is needed to read the Data area when debugger is connected.

(3) Flash protection by OTP

- User area on code flash supports OTP (One Time Programmable) functionality for each block.
- Flash extra area protection by OTP
- Dedicated configuration area supports OTP functionality for each 4 Byte unit.

(4) ID code

- **Table 47.1** shows the ID code that can be set in this product.
  The system requires each ID authentication for programming and reading the ID.

Table 47.1 ID Code

ID Code	Function
OCD ID	Protection for debug connection
Serial Programmer ID	Protection to enter the serial programming mode
RHSIF ID	Protection for RHSIF Link Partner [For U2A-EVA/U2A16/U2A8 only]
Customer ID A, B, C	Protection for Code Flash and Flash extra area
Data Flash ID	Protection for Data Flash
C-TEST ID	Refer to the *RH850/U2A-EVA Group Security User's Manual: Hardware*. When this ID is not used, set all bits to 1.

- The setting and function that related each ID code can select the Flash Option Byte. The system requires each ID authentication to re-programming these Flash Option Byte.

(5) Flash protection by the security function of ICUMHA

- When security function settings by ICUMHA is enabled, programming/erase of the code flash area from masters other than the ICUMHA or Hardware property setting is prohibited. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

47.2 Security Functions in Serial Programming Mode

Two functions are provided as security functions in serial programming mode: ID authentication and prohibition of serial programmer connection.

1. Serial Programmer ID Authentication
   The ID codes are checked for authenticity to protect the code flash memory, data flash memory and Flash extra area. Programming, erasure, and reading of the code flash and data flash memory can proceed upon successful ID authentication.
2. Prohibition of Serial Programmer Connection
   All serial programmer commands is disabled by boot firmware.

47.3 Security Functions in Debug Interfaces

This product is capable of restricting connection of the debug interfaces to protect against unauthorized access via the debug interfaces. This feature has two security levels.

- Security level 1:
  Debug interface can be used. At this level, the on-chip debugging (OCD) function is protected by using OCD ID authentication. For OCD to be used, it must be unlocked by using OCD ID authentication.
- Security level 2:
  At this level, the debug interface cannot be used.
  These security levels can be changed by Flash Option Byte (JPDBGIF_EN) in Security Setting area of the flash memory.

When ICUMHA is enabled, The C&R authentication is added to security function.
For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware.*

47.3.1 Security Levels and State of Restricting the Connection of Debug Interfaces

Table 47.2 show each security level and the corresponding security states and **Figure 47.1** shows the conditions for transitions.

Table 47.2 Security Levels and State of Restricting Connection to the Debug Interfaces

State	Result of OCD ID Authentication	JPDBGIF_EN*1	Restriction on Debug Interface Connection
Security level 1	Unlocked	1_B	No restriction on access via the debug interfaces
	Locked	1_B	Restriction on access via the debug interfaces is in place.
Security level 2	—	0_B	Connection to the debug interfaces is prohibited.

Note 1. For details of the JPDBGIF_EN bits, see **Section 47.8.3, S_OPBT0 — OCD ID Related Option Byte.**

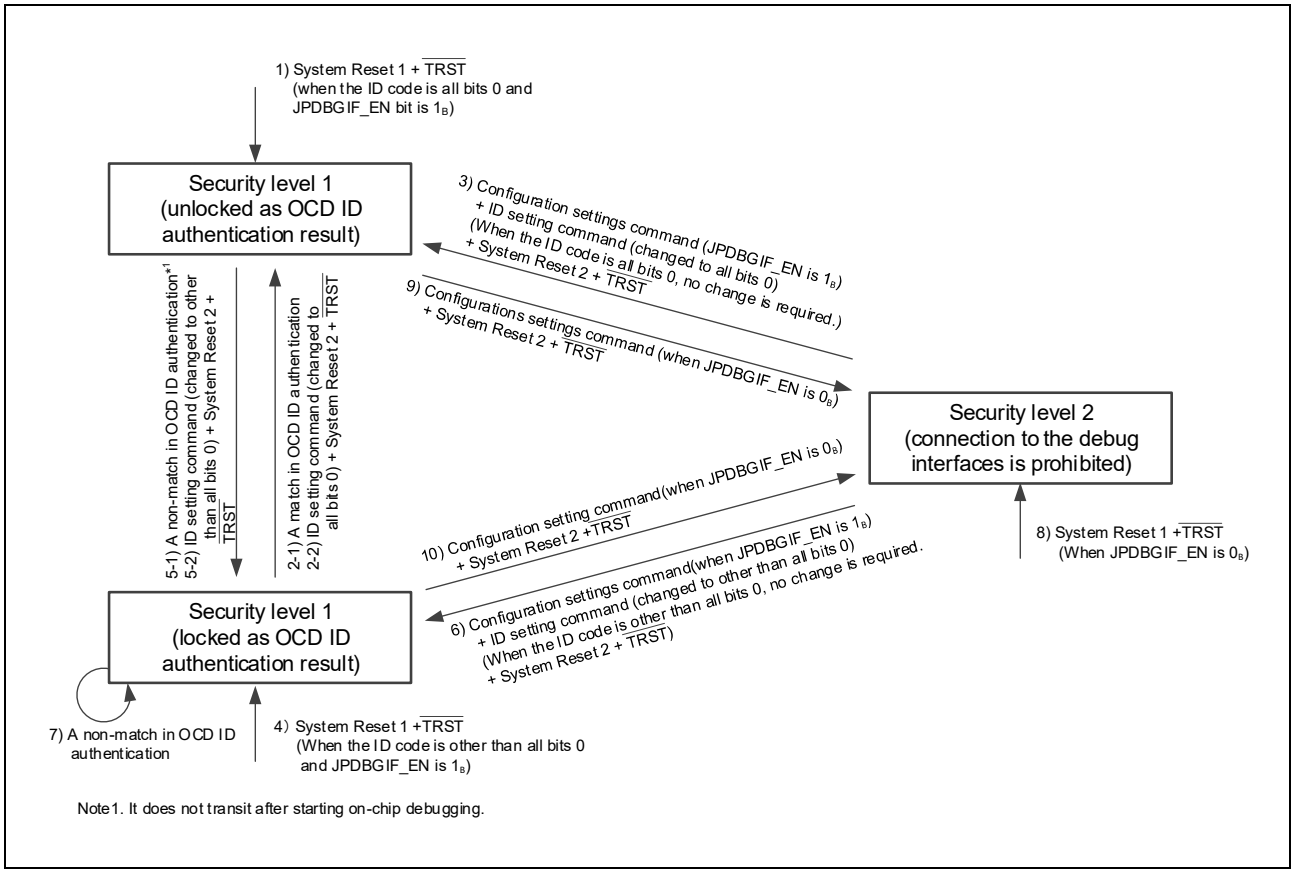


Figure 47.1 Transitions of Security Level

Conditions for transitions to each security level described in **Table 47.2**.

- Conditions for transitions to security level 1 (unlocked as OCD ID authentication result)

1. Startup in the state of security level 1 (unlocked as OCD ID authentication result)
   When System Reset 1 and a reset by $\overline{\text{TRST}}$ are applied while the ID code is all 0s and while the value of the JPDBGIF_EN is 1_B, startup proceeds with security level 1 (unlocked as OCD ID authentication result).
2. Transition from security level 1 (locked as OCD ID authentication result) to security level 1 (unlocked as OCD ID authentication result)

– Transition to security level 1 (unlocked as OCD ID authentication result) follows a match in OCD ID authentication.
– When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the ID code to “all bits 0”, startup proceeds at security level 1 (unlocked as OCD ID authentication result).

3. Transition from security level 2 to security level 1 (unlocked as OCD ID authentication result)
When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the ID code to “all bits 0” and the value of the JPDBGIF_EN to 1_B, startup proceeds at security level 1 (unlocked as OCD ID authentication result).

- Conditions for Transitions to Security Level 1 (locked as OCD ID Authentication result)

4. Startup in the state of security level 1 (locked as OCD ID authentication result)
When System Reset 1 and a reset by $\overline{\text{TRST}}$ is applied while the ID code is “other than all bits 0” and the value of the JPDBGIF_EN is 1_B, startup proceeds at security level 1 (locked OCD ID authentication result).
5. Transition from security level 1 (unlocked as OCD ID authentication result) to security level 1 (locked as OCD ID authentication result).

– A transition to security level 1 (locked as OCD ID authentication result) follows a non-match in OCD ID authentication. However, it does not transit after starting on-chip debugging.
– When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the ID code to “other than all bits 0”, startup proceeds at security level 1 (locked as OCD ID authentication result).

6. Transition from security level 2 to security level 1 (locked as OCD ID authentication result)
When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the ID code to “other than all bits 0” and the value of JPDBGIF_EN is 1_B, startup proceeds at security level 1 (locked as OCD ID authentication result).
7. Retaining security level 1 (locked as OCD ID authentication result)
In case of a non-match in OCD ID authentication, security level 1 (failure in OCD ID authentication) remains in place.

- Transition to security level 2

8. Startup at security level 2
When System Reset 1 and a reset by $\overline{\text{TRST}}$ is applied while the value of JPDBGIF_EN is 0_B, startup proceeds at security level 2.
9. Transition from security level 1 (unlocked as OCD ID authentication result) to security level 2
When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the value of the JPDBGIF_EN to 0_B, startup proceeds at security level 2.
10. Transition from security level 1 (locked as OCD ID authentication result) to security level 2
When System Reset 2 and a reset by $\overline{\text{TRST}}$ is applied after changing the value of the JPDBGIF_EN to 0_B, startup proceeds at security level 2.

47.4 Security Functions in Connection of RHSIF Link Partner [For U2A-EVA/U2A16/U2A8 only]

Two functions are provided as security functions unique to RHSIF ID authentication.

- RHSIF ID Authentication
  ID codes are checked for authenticity to protect RHSIF Link Partner connection.
  When ICUMHA is enabled, the C&R authentication is added to security function.
  For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.
- Access Window setting
  User can set the access area from Link Partner that is configured the access window register. The access window register can not set from Link Partner.
  The details of this function, see **Section 29, Renesas High-Speed Serial I/F (RHSIF)**.

47.5 Code Flash/Data Area Protection

Protection based on section password operates for non-secure area and doesn't affect the access from ICUMHA. The section password is stored in Security settings area for each of code flash and Data Area (Customer ID and Data Flash ID). User can unlock protection by inputting correct section password by software or external debugging tool.

Inputting target registers for software, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Customer ID and Data Flash ID, see **Section 47.8.11, CUSTOMERIDn — Customer ID A (n = 0 to 7), Section 47.8.14, CUSTOMERIDBn — Customer ID B (n = 0 to 7), Section 47.8.15, CUSTOMERIDCn — Customer ID C (n = 0 to 7),** and **Section 47.8.12, DATAFLASHIDn — Data Flash ID (n = 0 to 7)**.

The ID authentication for read access can be skipped by setting read protection flag in Security settings area for code flash and Data Area respectively (CFRPF and DFRPF).

Table 47.3 shows access protection to User Area or Extended Data Area by Customer ID. ID authentication to read access in Normal Operation mode/On-chip debug is done via debugger I/F. ID authentication for reading access in serial programming mode is not needed.

Table 47.4 shows access protection to Data Area by Data Flash ID. ID authentication to access is same as User Area/Extended Data Area. But the ID to authenticate becomes Data Flash ID.

Table 47.3 Access Protection to User Area/Extended Data Area from PEs by Customer ID

Mode	Operation	Access from PEs		Access Via PEs Debug Master	
		Read	Erase/Write	Read	Erase/Write
Normal Operation Mode /User Boot Mode	Self-programming	Don't care	Customer ID A/ B/C authentication by SW	N/A	N/A
	On-chip debug	Customer ID A authentication by debugger I/F	Customer ID A authentication by debugger I/F*1	Customer ID A authentication by debugger I/F	Customer ID A authentication by debugger I/F*1
			Customer ID B/C authentication by SW		Customer ID B/C authentication by SW
Serial programming mode	—	Don't care	Don't care	N/A	N/A

Note: Don't care: there are no access restriction, N/A: not applicable

Note 1. If target ID has not been authenticated by debugger I/F, the ID certification by SW is necessary.

Table 47.4 Access Protection to Data Area from PEs by Data Flash ID

Mode	Operation	Access from PEs		Access Via PEs Debug Master	
		Read	Erase/Write	Read	Erase/Write
Normal Operation Mode/User Boot Mode	Self-programming	Don't care	Data Flash ID authentication by SW	N/A	N/A
	On-chip debug	Data Flash ID authentication by debugger I/F	Data Flash ID authentication by debugger I/F*1	Data Flash ID authentication by debugger I/F	Data Flash ID authentication by debugger I/F*1
Serial programming mode	—	Don't care	Don't care	N/A	N/A

Note: Don't care: there are no access restriction, N/A: not applicable

Note 1. If target ID has not been authenticated by debugger I/F, the ID certification by SW is necessary.

47.5.1 Block Protection Function

If FACI command is issued to a block that was protected by ID, the flash sequencer enters "Command Lock" state.

- User Area/Extended Data Area Erase/Write Protection
  User Area/Extended Data Area can be protected from Erase/Write by Customer ID authentication. This function can be set by 4 bit per Block of code flash. **Figure 47.2** shows an example of Block Protection structure. Once set, the OTP setting cannot be released.
- Data Area Erase/Write Protection
  Data Area can be protected from Erase/Write by Data Flash ID authentication. This function can be set by DPROT. (see **Section 47.8.5, S_OPBT4 — Data Flash ID Related Option Byte**.)

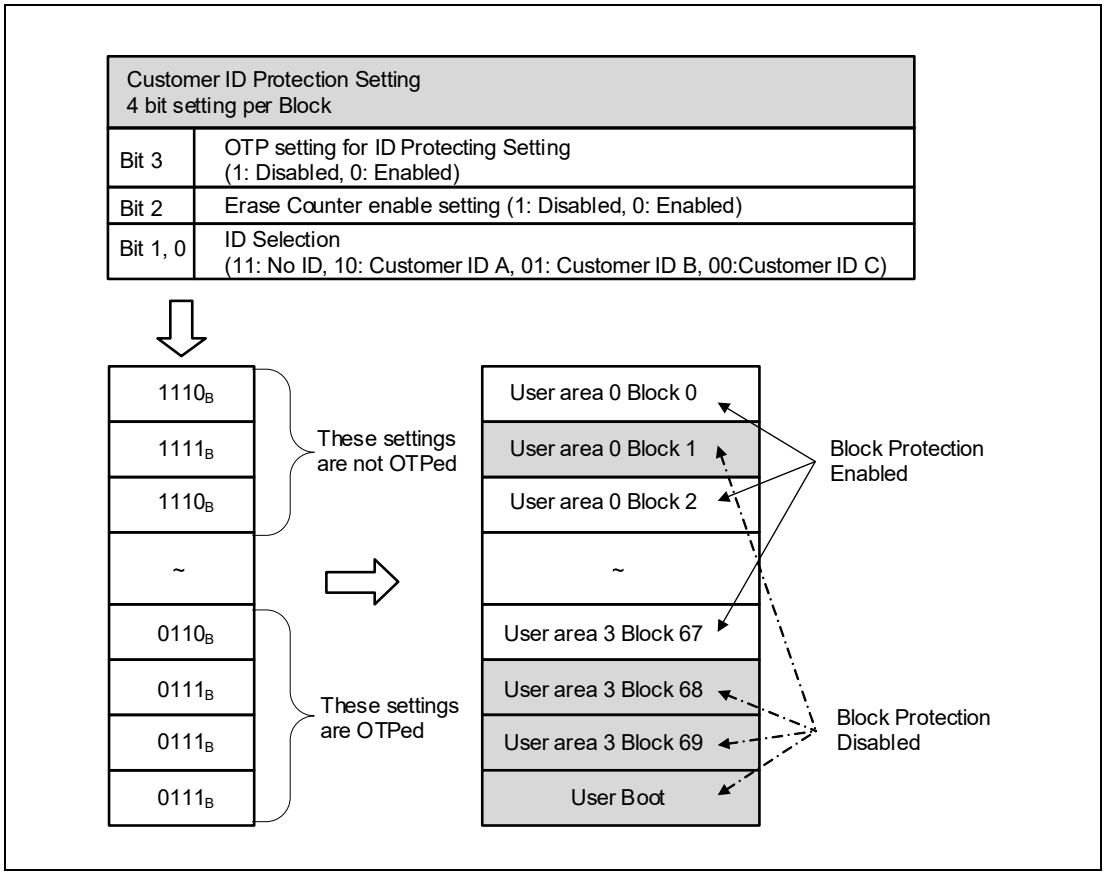


Figure 47.2 Block Protection Function Structure

47.5.2 OTP (One Time Programmable) Function

OTP can be individually set for each block of the user area and the user boot area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self programming is prohibited. Once set, the OTP setting cannot be released.

If a "Programming" or "Block Erasure" command is issued to a OTP protected block, flash sequencer enters "Command Lock" state.

For details of base address in the following table, see **Section 51.3.3, Mapping of Hardware Property Area in Data Flash Memory**.

Table 47.5, **Table 47.6**, and **Table 47.7** show Mapping of OTP setting for User Area, User Boot Area, and Extended Data Area.

Table 47.5 Mapping of OTP Setting (User Area m and User Boot Area m) (m[*1] = 0, 1) (k = f, b)

Address	Bit	Protection Area
<BPA0k_base>+ 0200_H+ 20_H x m	[0]	Block 0 of User Area m
	[1]	Block 1 of User Area m
	[2]	Block 2 of User Area m
	:	:
	[31]	Block 31 of User Area m
<BPA0k_base>+ 0204_H+ 20_H x m	[0]	Block 32 of User Area m
	[1]	Block 33 of User Area m
	:	:
	[31]	Block 63 of User Area m
<BPA0k_base>+ 0208_H+ 20_H x m	[0]	Block 64 of User Area m
	[1]	Block 65 of User Area m
	:	:
	[5]	Block 69 of User Area m
	[6]	Reserved
	:	:
	[31]	Reserved
<BPA0k_base>+ ($020C_H$ to 0218_H) + 20_H x m	[31:0]	Reserved[*2]
<BPA0k_base>+ $021C_H$+ 20_H x m	[0]	Reserved
	:	:
	[30]	Reserved
	[31]	User Boot Area m

Note 1. OTP setting area of m = 0 is shared with User Area 1, User Boot Area 1 when Double Map Mode.
OTP setting area of m = 1 is reserved when Double Map Mode.

Note 2. These reserved words need to be programmed when programming Block Protection Area for FPSYS0.

Table 47.6 Mapping of OTP Setting (Extended Data Area) (k = f, b)

Address	Bit	Protection Area
<BPA0k_base>+ $03E0_H$	[0]	Extended Data Area
	[1]	Reserved
	:	:
	[31]	Reserved
<BPA0k_base>+ $03E4_H$ to $03FC_H$	[31:0]	Reserved

Table 47.7 Mapping of OTP Setting (User Area m) (m[*1] = 2, 3) (k = f, b) [For U2A-EVA U2A16 mode/U2A16]

Address	Bit	Protection Area
<BPA1k_base>+ 0200_H+ 20_H x (m - 2)	[0]	Block 0 of User Area m
	[1]	Block 1 of User Area m
	[2]	Block 2 of User Area m
	:	:
	[31]	Block 31 of User Area m
<BPA1k_base>+ 0204_H+ 20_H x (m - 2)	[0]	Block 32 of User Area m
	[1]	Block 33 of User Area m
	:	:
	[31]	Block 63 of User Area m
<BPA1k_base>+ 0208_H+ 20_H x (m - 2)	[0]	Block 64 of User Area m
	[1]	Block 65 of User Area m
	:	:
	[5]	Block 69 of User Area m
	[6]	Reserved
	:	:
	[31]	Reserved
<BPA1k_base>+ ($020C_H$ to $021C_H$)+ 20_H x (m - 2)	[31:0]	Reserved[*2]

Note 1. OTP setting area of m = 2 is shared with User Area 3 when Double Map Mode.
OTP setting area of m = 3 is reserved when Double Map Mode.

Note 2. These reserved words need to be programmed when programming Block Protection Area for FPSYS1.

47.6 Hardware Property Area Protection

The access to Hardware Property Area (Configuration Setting, Security Setting, Block Protection, Switch, TAG Area) is protected by each protection ID. ID authentication for reading/writing access is done by SW. Customer ID A authentication by debugger I/F for read access can be skipped by setting read protection flag in Security Setting Area (CFRPF).

Table 47.8 shows access protection to Hardware Property Area (Configuration Setting, Security Setting, Block Protection, Switch, TAG Area). Protection ID of Configuration Setting Area, see **Section 47.7, Configuration Setting Area (Option Bytes, Reset Vector)**. Protection ID of Security Setting Area, see **Section 47.8, Security Setting Area**. Protection ID of Block Protection Area, see **Section 47.9, Block Protection Area for FPSYS0/FPSYS1**. Protection ID of Switch Area, see **Section 47.10, Switch Area**. Protection ID of TAG Area, see **Section 47.11, TAG Area**.

Table 47.8 Access Protection to Hardware Property Area (Configuration Setting, Security Setting, Block Protection, Switch, TAG Area) from PEs

Mode	Operation	Access from PEs		Access via PEs Debug Master	
		Read	Erase*2/Write	Read	Erase*2/Write
Normal Operation Mode / User Boot Mode	Self-programming	ID authentication by SW	ID authentication by SW	N/A	N/A
	On-chip debug	Customer ID A authentication by debugger I/F and ID authentication by SW*1	ID authentication by SW*1	Customer ID A authentication by debugger I/F and ID authentication by SW*1	ID authentication by SW*1
Serial programming mode	—	ID authentication by SW	ID authentication by SW	N/A	N/A

Note: N/A: not applicable

Note 1. If target ID has been authenticated by debugger I/F, the ID certification by SW is unnecessary.

Note 2. For Switch Area and TAG Area Only.

47.7 Configuration Setting Area (Option Bytes, Reset Vector)

Configuration Setting Area can be protected by the following functions.

(1) OTP
When OTP protection is effective, configuration setting which is protected by the OTP setting is not changed by "Property Programming" command (the data will be copied from valid area automatically regardless of the input programming data), and flash sequencer does not enter "Command Lock" state. Invalid area of Configuration setting area always can be erased by "Property Erasure" command regardless of the OTP setting.

(2) ID authentication
When ID authentication protection is effective, configuration setting which is protected by the ID authentication is not changed by "Property Programming" command (the data will be copied from valid area automatically regardless of the input programming data), and flash sequencer does not enter "Command Lock" state. Invalid area of Configuration setting area always can be erased by "Property Erasure" command regardless of ID authentication. Configuration Setting Area can be protected from programming by Customer ID A.
This protection doesn't affect the access from ICUMHA.

(3) Clear protection against "Property Programming" command
OTP setting bits cannot be changed from "0" to "1" by "Property Programming" command.

Table 47.9 shows Mapping of OTP setting for Configuration Setting Area.

For details of base address in the following table, see **Section 51.3.3, Mapping of Hardware Property Area in Data Flash Memory**.

Table 47.9 Mapping of OTP Setting (Configuration Setting Area) (k = f, b)

Address	Bit	Protection Area (32 bit)
<CSAk_base>+ 0200_H	[0]	<CSAb_base>+0300_H
	[1]	<CSAb_base>+0304_H
	[2]	<CSAb_base>+0308_H
	:	:
	[31]	<CSAb_base>+$037C_H$
<CSAk_base>+ 0204_H	[0]	<CSAb_base>+0380_H
	[1]	<CSAb_base>+0384_H
	[2]	<CSAb_base>+0388_H
	:	:
	[31]	<CSAb_base>+$03FC_H$
<CSAk_base>+ 0208_H	[0]	<CSAb_base>+0400_H
	[1]	<CSAb_base>+0404_H
	[2]	<CSAb_base>+0408_H
	:	:
	[31]	<CSAb_base>+$047C_H$

47.8 Security Setting Area

Security Setting Area consists of two areas (Area 0 and Area 1). Valid area is mapped in front side, Invalid area is mapped in back side. Valid area cannot be updated whereas Invalid area can be updated.

Area 0 is selected as valid area at the shipping from RENESAS.

Area 1 is selected as invalid area and erased at the shipping from RENESAS.

The flash memory has the area to store a data specified by the user for security settings. Changes in settings become effective after TAG Update and release from the reset state. OTP setting of Security Settings Area becomes effective without reset. Valid / Invalid area switching for read access also becomes effective without reset.

Security Setting Area can be protected by the following functions.

(1) OTP
When OTP protection is effective, Security setting which is protected by the OTP setting is not changed by "Property Programming" command (the data will be copied from valid area automatically regardless of the input programming data), and flash sequencer does not enter "Command Lock" state. Invalid area of Security setting area always can be erased by "Property Erasure" regardless of the OTP setting.

(2) ID authentication
When ID authentication protection is effective, Security setting which is protected by the ID authentication is not changed by "Property Programming" command (the data will be copied from valid area automatically regardless of the input programming data), and flash sequencer does not enter "Command Lock" state. Invalid area of Security setting area always can be erased by "Property Erasure" command regardless of ID authentication. Security setting area can be protected from programming by Customer ID A/B/C, OCD ID, Serial Programming ID, Data Flash ID, and RHSIF ID. See **Table 47.11** for details.
This protection doesn't affect the access from ICUMHA.

(3) Clear protection against "Property Programming" command
OTP setting bits cannot be changed from "0" to "1" by "Property Programming" command.

Table 47.10 shows Mapping of OTP setting for Security Setting Area.

For details of base address in the following table, see **Section 51.3.3, Mapping of Hardware Property Area in Data Flash Memory**.

Table 47.10 Mapping of OTP Setting (Security Setting Area) (k = f, b)

Address	Bit	Protection Area (32bit)
<SSAk_base>+ 0200_H	[0]	<SSAb_base>+0300_H
	[1]	<SSAb_base>+0304_H
	[2]	<SSAb_base>+0308_H
	:	:
	[31]	<SSAb_base>+$037C_H$
<SSAk_base>+ 0204_H	[0]	<SSAb_base>+0380_H
	[1]	<SSAb_base>+0384_H
	[2]	<SSAb_base>+0388_H
	:	:
	[31]	<SSAb_base>+$03FC_H$
<SSAk_base>+ 0208_H	[0]	<SSAb_base>+0400_H
	[1]	<SSAb_base>+0404_H
	[2]	<SSAb_base>+0408_H
	:	:
	[23]	<SSAb_base>+$045C_H$
	[24]	Reserved (need to program ‘1’)
	:	:
	[31]	Reserved (need to program ‘1’)
<SSAk_base>+ 0220_H	[0]	<SSAb_base>+0700_H
	[1]	<SSAb_base>+0704_H
	[2]	<SSAb_base>+0708_H
	:	:
	[7]	<SSAb_base>+$071C_H$
	[8]	Reserved (need to program ‘1’)
	:	:
	[31]	Reserved (need to program ‘1’)

Table 47.11 shows Security settings area. For setting and reading this area, see the *Renesas Flash Programmer Flash Programming Software User's Manual*, or the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 47.11 Security Settings Area (1/2)

Name	Address*11	State at the shipping*1	Write Protection ID*3	Read Protection ID*3, *7	SSAVOF/ SSAVOFC Number
Valid Option Byte Flag (SSAVOF0-1)*8, *10	<SSAk_base> + 0000_H to 0004_H	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved *9	<SSAk_base>+ 0008_H to 0018_H	Erased	Can not Write	—	—
Valid Option Byte Flag (SSAVOF7-18) *8, *10	<SSAk_base> + $001C_H$ to 0048_H)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*9	<SSAk_base> + ($004C_H$ to $009C_H$)	Erased	Can not Write	—	—
Valid Option Byte Flag (SSAVOF40) *8, *10	<SSAk_base> + $00A0_H$	5AA5 A55A$_H$	Can not Write	—	—
Reserved*9	<SSAk_base>+ ($00A4_H$ to $00FC_H$)	Erased	Can not Write	—	—
VOF Program completion flag (SSAVOFC0-1)*8, *10	<SSAk_base> + (0100_H to 0104_H)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*9	<SSAk_base>+ (0108_H to 0118_H)	Erased	Can not Write	—	—
VOF Program completion flag (SSAVOFC7-18)*8, *10	<SSAk_base> + ($011C_H$ to 0148_H)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*9	<SSAk_base> + ($014C_H$ to $019C_H$)	Erased	Can not Write	—	—
VOF Program completion flag (SSAVOFC40)*8, *10	<SSAk_base> + $01A0_H$	5AA5 A55A$_H$	Can not Write	—	—
Reserved*9	<SSAk_base>+ ($01A4_H$ to $01FC_H$)	Erased	Can not Write	—	—
OTP Setting of Security Settings Area*2	<SSAk_base>+ (0200_H to 0208_H)	All FFFF FFFF$_H$	—	—	0
Reserved (need to be programmed)*5	<SSAk_base>+ ($020C_H$ to $021C_H$)	All FFFF FFFF$_H$	—	—	0
OTP Setting of Security Settings Area*2	<SSAk_base>+ 0220_H	All FFFF FFFF$_H$	—	—	1
Reserved (need to be programmed)*5	<SSAk_base>+ (0224_H to $023C_H$)	All FFFF FFFF$_H$	—	—	1
Reserved*9	<SSAk_base>+ (0240_H to $02DC_H$)	Erased	Can not Write	—	—
Reserved (need to be programmed)*5*12	<SSAk_base>+ $02E0_H$ to $02FC_H$)	FFFF FFFF$_H$	—	—	7
OCD ID related option bytes (S_OPBT0)	<SSAk_base>+ 0300_H	FFFF FFFF$_H$	OCD ID	—	8
Reserved (need to be programmed)*12	<SSAk_base>+ 0304_H	FFFF FFFF$_H$	—	—	8
Reserved (need to be programmed)*12	<SSAk_base>+ 0308_H	FFFF FFFF$_H$	Customer ID A	—	8
Customer ID A related option bytes (S_OPBT3)	<SSAk_base>+ $030C_H$	FFFF FFFF$_H$	Customer ID A	—	8
DataFlash ID related option bytes (S_OPBT4)	<SSAk_base>+ 0310_H	FFFF FFFF$_H$	Data Flash ID	—	8

Table 47.11 Security Settings Area (2/2)

Name	Address*11	State at the shipping*1	Write Protection ID*3	Read Protection ID*3, *7	SSAVOF/ SSAVOFC Number
Serial Programmer ID related option bytes (S_OPBT5)	<SSAk_base>+ 0314_H	FFFF $FFFF_H$	Serial Programmer ID*6	—	8
RHSIF ID related option bytes (S_OPBT6)*13	<SSAk_base>+ 0318_H	FFFE $FFFF_H$	RHSIF ID	—	8 [For U2A-EVA/U2A16/ U2A8 only]
Debugging and Calibration related Option Byte (S_OPBT7)	<SSAk_base>+ $031C_H$	FDFF $FFFF_H$	OCD ID & Customer ID A*4	—	8
OCD ID (OCDIDn)	<SSAk_base>+ (0320_H to $033C_H$)	All FFFF $FFFF_H$	OCD ID	OCD ID	9
Serial Programmer ID (SPIDn)	<SSAk_base>+ (0340_H to $035C_H$)	All FFFF $FFFF_H$	Serial Programmer ID*6	Serial Programmer ID*6	10
Customer ID A (CUSTOMERIDn)	<SSAk_base>+ (0360_H to $037C_H$)	All FFFF $FFFF_H$	Customer ID A	Customer ID A	11
Data Flash ID (DATAFLASHIDn)	<SSAk_base>+ (0380_H to $039C_H$)	All FFFF $FFFF_H$	Data Flash ID	Data Flash ID	12
Reserved (need to be programmed)*12	<SSAk_base>+ ($03A0_H$ to $03BC_H$)	All FFFF $FFFF_H$	—	—	13
RHSIF ID (RHSIFIDn)*13	<SSAk_base>+ ($03C0_H$ to $03DC_H$)	All FFFF $FFFF_H$	RHSIF ID	RHSIF ID	14 [For U2A-EVA/U2A16/ U2A8 only]
Customer ID B (CUSTOMERIDBn)	<SSAk_base>+ ($03E0_H$ to $03FC_H$)	All FFFF $FFFF_H$	Customer ID B	Customer ID B	15
Customer ID C (CUSTOMERIDCn)	<SSAk_base>+ (0400_H to $041C_H$)	All FFFF $FFFF_H$	Customer ID C	Customer ID C	16
Reserved (need to be programmed)*5	<SSAk_base>+ (0420_H to $045C_H$)	All FFFF $FFFF_H$	—	—	17 to 18
Reserved*9	<SSAk_base>+ (0460_H to $06FC_H$)	Erased	Can not Write	—	—
Reserved (need to be programmed)*12	<SSAk_base>+ 0700_H	FFFF $FFFF_H$	Customer ID A	Customer ID A	40
Reserved (need to be programmed)*12	<SSAk_base>+ 0704_H	FFFF $FFFF_H$	Customer ID A	Customer ID A	40
Reserved (need to be programmed)*12	<SSAk_base>+ 0708_H	FFFF $FFFF_H$	Customer ID A	Customer ID A	40
Reserved (need to be programmed)*12	<SSAk_base>+ $070C_H$	FFFF $FFFF_H$	Customer ID A	Customer ID A	40
Reserved (need to be programmed)*12	<SSAk_base>+ 0710_H	FFFF $7FEF_H$	Customer ID A	Customer ID A	40
Reserved (need to be programmed)*12	<SSAk_base>+ (0714_H to $071C_H$)	All FFFF $FFFF_H$	Customer ID A	Customer ID A	40
Reserved*9	<SSAk_base>+ (0720_H to $073C_H$)	Erased	Can not Write	Customer ID A	—
Reserved*9	<SSAk_base>+ (0740_H to $07FC_H$)	Erased	Can not Write	—	—

Note 1. This is the value of valid area at the shipping. Invalid area is erased at the shipping.

Note 2. For details of this function, see **Section 47.8, Security Setting Area**.

Note 3. When all bits of Protection ID are “1” or “0”, the ID authentication is no needed to program/read. And when this protection is effective, Config area is not change/modify by "Property Program" (copied data from valid area is programed), and flash sequencer does not enter "Command Lock" state. Config area always can erase by "Property Erase" regardless of ID authentication.

Note 4. When CFRPF is set (CFRPF bit is 0).

Note 5. For this reserved area, program FFFF $FFFF_H$ when programming Security Setting Area.
Note 6. When the all bits are “1”, the ID authentication is needed to connect a flash memory programmer.
Note 7. When On-chip debug and S_OPBT3.CFRPF is "0", the Customer ID A authentication is needed to read.
Note 8. This area will be updated automatically when programming Security Setting Area.
Note 9. This area cannot be programmed. When this area is read, it behaves as if it is erased.
Note 10. For details of this function, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.
Note 11. <SSAk_base> (k = f, b)
Note 12. For this reserved area, program the value of valid area at the shipping when programming Security Setting Area. The details of this function, see the *RH850/U2A-EVA Group Security User’s Manual: Hardware*.
Note 13. S_OPBT6 is reserved in U2A6. For this reserved area, program FFFE $FFFF_H$ when programming Security Setting Area.
RHSIFIDn is reserved in U2A6. For this reserved area, program FFFF $FFFF_H$ when programming Security Setting Area.

47.8.1 SSAVOFn — Valid Option Byte Flag n for Security Setting Area (n = 0,1,7 to 18,40)

This flag indicates that the related option bytes are programmed or not.
The relation between each SSAVOFn and option bytes are shown in **Table 47.11**.
This flag is referred by flash sequencer in order to check whether the related option bytes are programmed or not.
This flag will be updated automatically when programming Security Setting Area.
The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + 0000$_H$ + 04$_H$ x n

Value at the shipping: 5AA5 A55A$_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSAVOFn[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSAVOFn[15:0]															

Table 47.12 SSAVOFn Contents

Bit Position	Bit Name	Function
31 to 0	SSAVOFn [31:0]	Program completion flag. 5AA5 A55A$_H$: The related option bytes are programmed. Other than above: The related option bytes are not programmed.

47.8.2 SSAVOFCn — VOF Program completion flag n for Security Setting Area (n = 0,1,7 to 18,40)

This flag indicates that SSAVOFn is programmed or not.
The relation between each SSAVOFCn and option bytes are shown in **Table 47.11**.
This flag is prepared to check whether the related option bytes are programmed or not by user program.
This flag will be updated automatically when programming Security Setting Area.
The updated value will be 5AA5 $A55A_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <SSAk_base> + 0100_H + 04_H x n

Value at the shipping: 5AA5 $A55A_H$ (valid area)
Erased (invalid area)

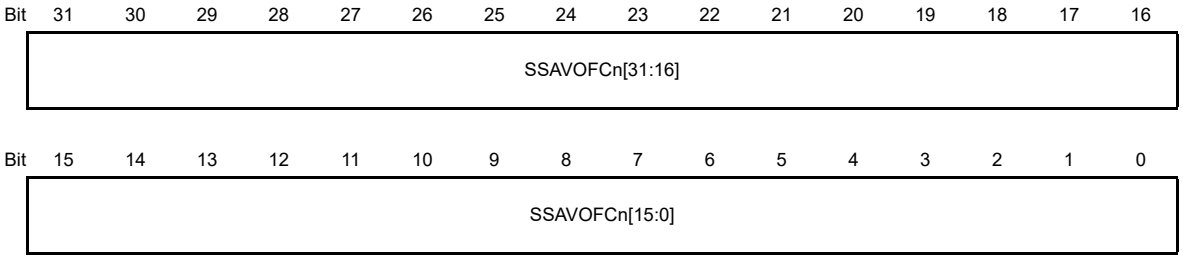


Table 47.13 SSAVOFCn Contents

Bit Position	Bit Name	Function
31 to 0	SSAVOFCn [31:0]	SSAVOFn Program completion flag. 5AA5 $A55A_H$: SSAVOFn is programmed. Other than above: SSAVOFn is not programmed.

47.8.3 S_OPBT0 — OCD ID Related Option Byte

OCD ID authentication is necessary to program these data.

When all bits of OCD ID are “1” or “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + 0300_H

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	JPDBGIF_EN	DRDY_EN	—	—	—	—	—	CPUBTMSK_EN	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIDDIS

Table 47.14 S_OPBT0 Contents

Bit Position	Bit Name	Function
31	JPDBGIF_EN	Security level 2 setting 1: Unset security level 2 (can use debug interface) 0: Set security level 2 (can not use debug interface)
30	DRDY_EN	Switching of the DRDY use for Nexus 1: Use DRDY 0: Not use DRDY(Use as GPIO)
29 to 25	Reserved	Set the value of valid area at the shipping.
24	CPUBTMSK_EN	PE0 boot mask enable (when the $\overline{TRST}$ = H mode only) 0: PE0 boot mask disable 1: PE0 boot mask enable
23 to 1	Reserved	Set the value of valid area at the shipping.
0	OIDDIS	OCD ID Authentication Disable 0: ID authentication is disabled (Can not entry Debug mode) 1: ID authentication is enabled

47.8.4 S_OPBT3 — Customer ID A Related Option Byte

Customer ID A authentication is necessary to program these data.

When all bits of Customer ID A are “1” or “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + $030C_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFRPF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 47.15 S_OPBT3 Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	Set the value of valid area at the shipping.
16	CFRPF	Code Flash Read Protection Flag Controlling section password authentication for code flash and Hardware Property Area read access at debug mode. 0: Customer ID A authentication for read access is needed. 1: Customer ID A authentication for read access is not needed.
15 to 0	Reserved	Set the value of valid area at the shipping.

47.8.5 S_OPBT4 — Data Flash ID Related Option Byte

Data Flash ID authentication is necessary to program these data.

When all bits of Data Flash ID are “1” or “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + 0310_H

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DFRPF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPROT

Table 47.16 S_OPBT4 Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	Set the value of valid area at the shipping.
16	DFRPF	Data Flash Read Protection Flag. Controlling section password authentication for Data Area read access at debug mode. 0: Data Flash ID authentication for read access is needed. 1: Data Flash ID authentication for read access is not needed.
15 to 1	Reserved	Set the value of valid area at the shipping.
0	DPROT	Data Flash Erase/Write protection Flag. Controlling section password authentication for Data Area Erase / Write access. 0: Data Flash ID authentication for Erase/Write access is needed. 1: Data Flash ID authentication for Erase/Write access is not needed.

47.8.6 S_OPBT5 — Serial Programmer ID Related Option Byte

Serial Programmer ID authentication is necessary to program these data.

When all bits of Serial Programmer ID are “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + 0314_H

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPD

Table 47.17 S_OPBT5 Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	Set the value of valid area at the shipping.
0	SPD	Serial Programmer Disable 0: All serial programmer commands is disabled by boot firmware. 1: Serial programmer commands is enabled.

47.8.7 S_OPBT6 — RHSIF ID Related Option Byte [For U2A-EVA/U2A16/U2A8 only]

RHSIF ID authentication is necessary to program these data.

When all bits of RHSIF ID are “1” or “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + 0318_H

Value at the shipping: FFFE $FFFF_H$ (valid area)
Erased (invalid area)

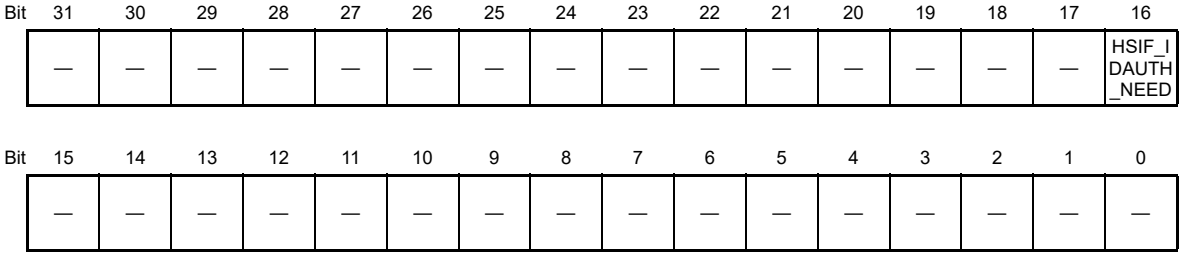


Table 47.18 S_OPBT6 Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	Set the value of valid area at the shipping.
16	HSIF_IDAUTH_NEED	RHSIF Link Partner Authentication Setting 0: RHSIF ID authentication for Link Partner access is not needed. 1: RHSIF ID authentication for Link Partner access is needed.
15 to 0	Reserved	Set the value of valid area at the shipping.

47.8.8 S_OPBT7 — Debugging and Calibration related Option Byte

OCD ID authentication and Customer ID A authentication (CFRPF bit of S_OPBT3 is 0) are necessary to program these data.

When all bits of Protection ID are “1”, the ID authentication are not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + $031C_H$

Value at the shipping: FDFF $FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	DDRSTDIS	—	—	—	—	—	—	—	OVERLAYEN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AUDREN*1

Note 1. This bit is available in U2AEVA only. For U2A16/U2A8/U2A6, this bit is reserved.

Table 47.19 S_OPBT7 Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	Set the value of valid area at the shipping.
24	DDRSTDIS	[For U2A-EVA] Reserved. Set the value of valid area at the shipping. [For U2A16/U2A8/U2A6] Debugger Disconnection Reset Control Flag 0: Debugger Disconnection Reset is enabled. 1: Debugger Disconnection Reset is disabled.
23 to 17	Reserved	Set the value of valid area at the shipping.
16	OVERLAYEN	Overlay function enable/disable 0: Overlay function of CFU, GCFU is disabled. 1: Overlay function of CFU, GCFU can be enabled.
15 to 1	Reserved	Set the value of valid area at the shipping.
0	AUDREN	[For U2A-EVA Only] AUDR monitor enable 0: AUDR is disabled. 1: AUDR is enabled.

47.8.9 OCDIDn — OCD ID (n = 0 to 7)

OCD ID authentication is necessary to read / program these data.

When all bits of OCD ID are “1” or “0”, the ID authentication is not necessary to read / program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + 0320_H + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

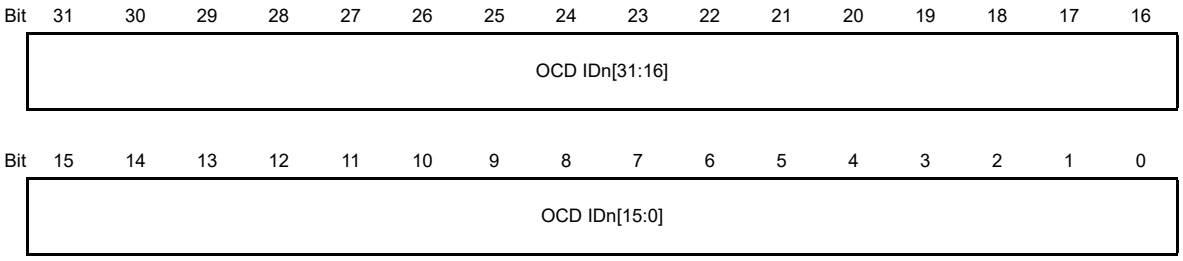


Table 47.20 OCDIDn Contents

Bit Position	Bit Name	Function
31 to 0	OCD ID	ID Code for On-chip debug connection authentication. Bit mapping table is as follows: OCD ID[31:0]: OCDID0[31:0] OCD ID[63:32]: OCDID1[31:0] … OCD ID[255:224]: OCDID7[31:0]

47.8.10 SPIDn — Serial Programmer ID (n = 0 to 7)

Serial Programmer ID authentication is needed for read/program these data.

When all bits of Serial Programmer ID are “0”, the ID authentication is not necessary to read/program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + 0340_H + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

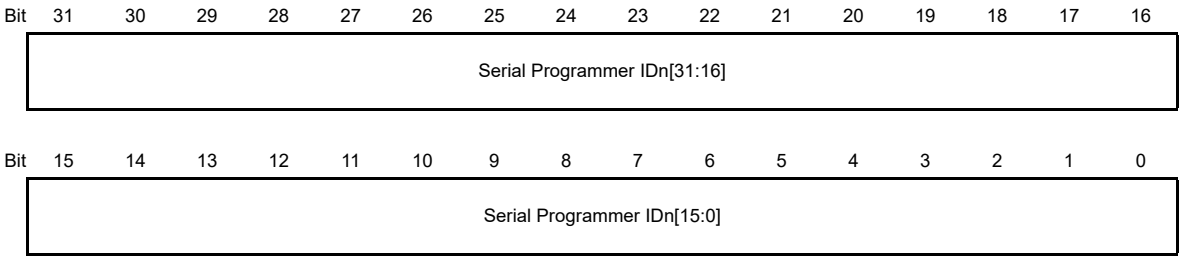


Table 47.21 SPIDn Contents

Bit Position	Bit Name	Function
31 to 0	Serial Programmer ID	ID Code for Serial Programmer authentication. Bit mapping table is as follows: Serial Programmer ID[31:0]: SPID0[31:0] Serial Programmer ID[63:32]: SPID1[31:0] … Serial Programmer ID[255:224]: SPID7[31:0]

47.8.11 CUSTOMERIDn — Customer ID A (n = 0 to 7)

Customer ID A authentication is needed for read / program these data.

When all bits of Customer ID A are “1” or “0”, the ID authentication is not necessary to read / program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + 0360_H + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

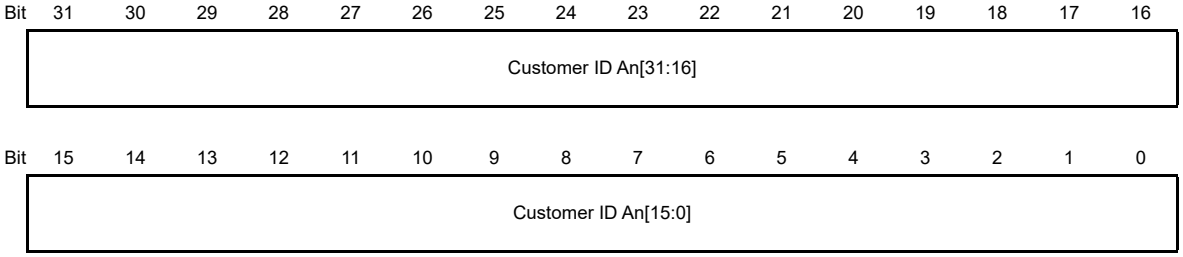


Table 47.22 CUSTOMERIDn Contents

Bit Position	Bit Name	Function
31 to 0	Customer ID A	ID Code for Code Flash Protection, Configuration setting and etc. Bit mapping table is as follows: Customer ID A[31:0]: CUSTOMERID0[31:0] Customer ID A[63:32]: CUSTOMERID1[31:0] … Customer ID A[255:224]: CUSTOMERID7[31:0]

47.8.12 DATAFLASHIDn — Data Flash ID (n = 0 to 7)

Data Flash ID authentication is needed for read / program these data.

When all bits of Data Flash ID are “1” or “0”, the ID authentication is not necessary to read / program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + 0380_H + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

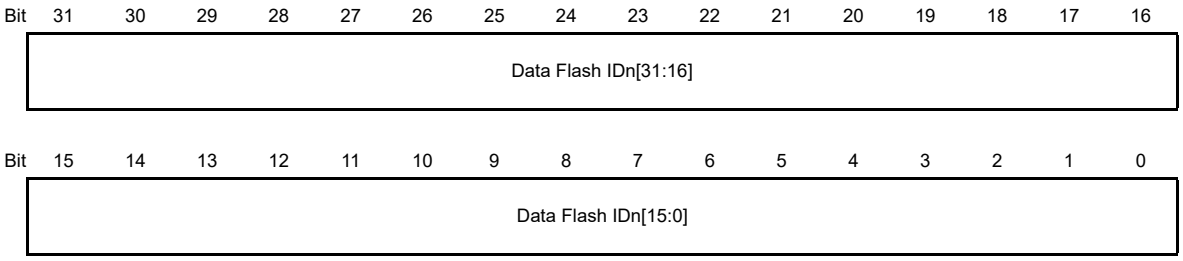


Table 47.23 DATAFLASHIDn Contents

Bit Position	Bit Name	Function
31 to 0	Data Flash ID	ID Code for Data Area Protection. Bit mapping table is as follows: Data Flash ID[31:0]: DATAFLASHID0[31:0] Data Flash ID[63:32]: DATAFLASHID1[31:0] … Data Flash ID[255:224]: DATAFLASHID7[31:0]

47.8.13 RHSIFIDn — RHSIF ID (n = 0 to 7) [For U2A-EVA/U2A16/U2A8 only]

RHSIF ID authentication is needed for read / program these data.

When all bits of RHSIF ID are “1” or “0”, the ID authentication is not necessary to read/program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <SSAk_base> + $03C0_H$ + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

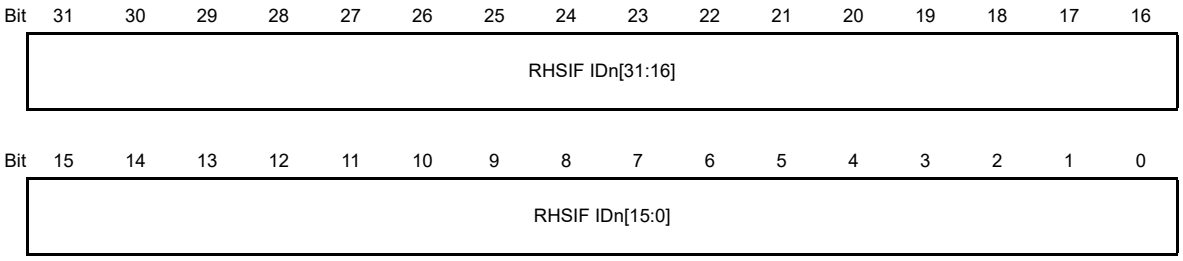


Table 47.24 RHSIFIDn Contents

Bit Position	Bit Name	Function
31 to 0	RHSIF ID	ID Code for RHSIF Link Partner connection authentication. Bit mapping table is as follows: RHSIF ID[31:0]: RHSIFID0[31:0] RHSIF ID[63:32]: RHSIFID1[31:0] … RHSIF ID[255:224]: RHSIFID7[31:0]

47.8.14 CUSTOMERIDBn — Customer ID B (n = 0 to 7)

Customer ID B authentication is needed for read/program these data.

When all bits of Customer ID B are “1” or “0”, the ID authentication is not necessary to read/program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + $03E0_H$ + 04_H x n

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

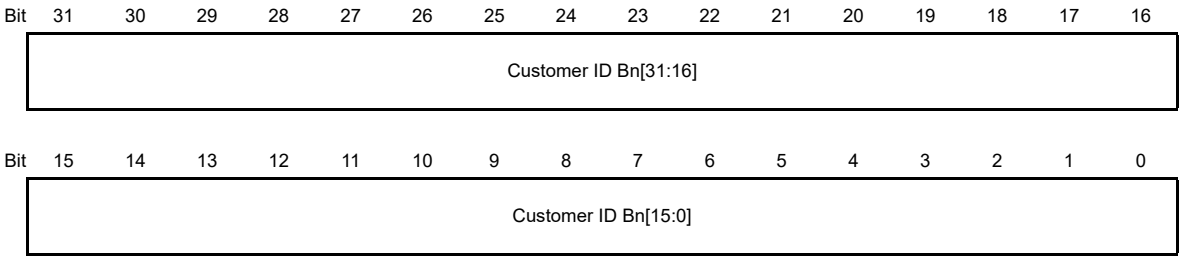


Table 47.25 CUSTOMERIDBn Contents

Bit Position	Bit Name	Function
31 to 0	Customer ID B	ID Code for Code Flash Protection, Bit mapping table is as follows: Customer ID B[31:0]: CUSTOMERIDB0[31:0] Customer ID B[63:32]: CUSTOMERIDB1[31:0] … Customer ID B[255:224]: CUSTOMERIDB7[31:0]

47.8.15 CUSTOMERIDCn — Customer ID C (n = 0 to 7)

Customer ID C authentication is needed for read/program these data.

When all bits of Customer ID C are “1” or “0”, the ID authentication is not necessary to read/program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SSAk_base> + 0400_H + 04_H x n

Value at the shipping: FFFF $FFFF_H$ (valid area)
Erased (invalid area)

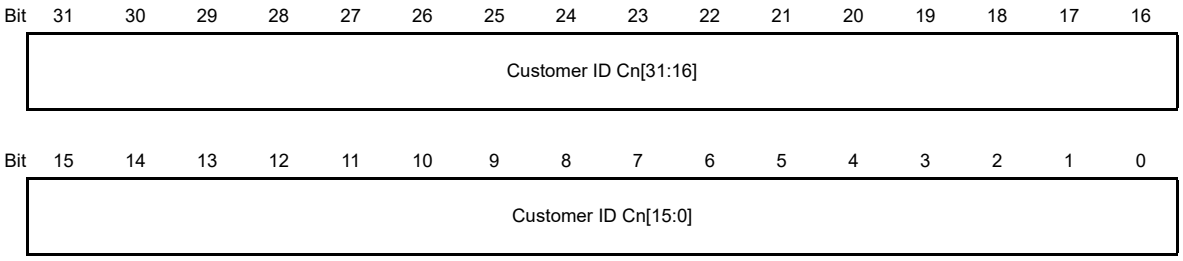


Table 47.26 CUSTOMERIDCn Contents

Bit Position	Bit Name	Function
31 to 0	Customer ID C	ID Code for Code Flash Protection Bit mapping table is as follows: Customer ID C[31:0]: CUSTOMERIDC0[31:0] Customer ID C[63:32]: CUSTOMERIDC1[31:0] … Customer ID C[255:224]: CUSTOMERIDC7[31:0]

47.9 Block Protection Area for FPSYS0/FPSYS1

Block Protection Area for each FPSYSn (n = 0,1) consists of two areas (Area 0 and Area 1). Valid area is mapped in front side, Invalid area is mapped in back side. Valid area cannot be updated whereas Invalid area can be updated.

Area 0 is selected as valid area at the shipping from RENESAS.

Area 1 is selected as invalid area and erased at the shipping from RENESAS.

The flash memory has the area to store a data specified by the user for Block Protection settings. Changes in settings become effective after TAG update without reset. Valid/Invalid area switching for read access also becomes effective without reset.

Do not issue reprogramming of user/data area or Extended data area without reset after changing the settings of Block Protection.

Block Protection Area can be protected by the following functions.

(1) ID authentication
When ID authentication protection is effective, Block protection which is protected by the ID authentication is not changed by "Property Programming" command (the data will be copied from valid area automatically regardless of the input programming data), and flash sequencer does not enter "Command Lock" state. Invalid area of Block protection area always can be erased by "Property Erasure" command regardless of ID authentication. Block protection area can be protected from programming by Customer ID A.
This protection doesn't affect the access from ICUMHA.

(2) Protection against "Property Programming" and "Property Erasure" command Block Protection Area for FPSYS1 cannot be erased/programmed while "TAG update/TAG erasure" command is being issued.

(3) Clear protection against "Property Programming" command
OTP setting bits cannot be changed from "0" to "1" by "Property Programming" command.

Table 47.27 and **Table 47.28** show Block Protection Area. For setting and reading these areas, see the *Renesas Flash Programmer Flash Programming Software User's Manual*, or the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 47.27 Block Protection Area for FPSYS0 (1/2)

Name	Address*10	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	BPA0VOF/ BPA0VOFC Number
Valid Option Byte Flag (BPA0VOF0-7) *7, *9	<BPA0k_base>+ (0000_H to $001C_H$)	5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA0k_base>+ (0020_H to 0038_H)	Erased	Can not Write	—	—
Valid Option Byte Flag (BPA0VOF15-31)*7, *9	<BPA0k_base>+ ($003C_H$ to $007C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA0k_base>+ (0080_H to 0098_H)	Erased	Can not Write	—	—
Valid Option Byte Flag (BPA0VOF39)*7, *9	<BPA0k_base>+ $009C_H$	5AA5 A55A$_H$	Can not Write	—	—

Table 47.27 Block Protection Area for FPSYS0 (2/2)

Name	Address*10	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	BPA0VOF/ BPA0VOFC Number
Reserved*8	<BPA0k_base>+ ($00A0_H$ to $00FC_H$)	Erased	Can not Write	—	—
VOF Program completion flag (BPA0VOFC0-7)*7, *9	<BPA0k_base>+ (0100_H to $011C_H$)	5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA0k_base>+ (0120_H to 0138_H)	Erased	Can not Write	—	—
VOF Program completion flag (BPA0VOFC15-31)*7, *9	<BPA0k_base>+ ($013C_H$ to $017C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA0k_base>+ (0180_H to 0198_H)	Erased	Can not Write	—	—
VOF Program completion flag (BPA0VOFC39)*7, *9	<BPA0k_base>+ $019C_H$	5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA0k_base>+ ($01A0_H$ to $01FC_H$)	Erased	Can not Write	—	—
OTP Setting of User area 0*4	<BPA0k_base>+ (0200_H to $021C_H$)	All FFFF FFFF$_H$	—	—	0
OTP Setting of User area 1*4	<BPA0k_base>+ (0220_H to $023C_H$)	All FFFF FFFF$_H$	—	—	1
Reserved(need to be programmed) *6	<BPA0k_base>+ (0240_H to $02FC_H$)	All FFFF FFFF$_H$	—	—	2 to 7
Reserved*8	<BPA0k_base>+ (0300_H to $03DC_H$)	Erased	Can not Write	—	—
OTP Setting of Extended Data Area.*4	<BPA0k_base>+ ($03E0_H$ to $03FC_H$)	All FFFF FFFF$_H$	—	—	15
Customer ID Protection & Erase Counter Setting for user area0 & user boot area0*5	<BPA0k_base>+ (0400_H to $043C_H$)	All FFFF FFFF$_H$	Customer ID A	—	16 to 17
Customer ID Protection & Erase Counter Setting for user area1 & user boot area1*5	<BPA0k_base>+ (0440_H to $047C_H$)	All FFFF FFFF$_H$	Customer ID A	—	18 to 19
Reserved(need to be programmed)*6	<BPA0k_base>+ (0480_H to $05FC_H$)	All FFFF FFFF$_H$	Customer ID A	—	20 to 31
Reserved*8	<BPA0k_base>+ (0600_H to $06DC_H$)	Erased	Can not Write	—	—
Customer ID Protection Setting for Extended data area	<BPA0k_base>+ ($06E0_H$ to $06FC_H$)	All FFFF FFFF$_H$	Customer ID A	—	39
Reserved*8	<BPA0k_base>+ (0700_H to $07FC_H$)	Erased	Can not Write	—*8	—

Table 47.28 Block Protection Area for FPSYS1

Name	Address*10	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	BPA1VOF/ BPA1VOFC Number
Valid Option Byte Flag (BPA1VOF0-7)*7, *9	<BPA1k_base>+ (0000_H to $001C_H$)	5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA1k_base>+ (0020_H to $003C_H$)	Erased	Can not Write*8	—*8	—
Valid Option Byte Flag (BPA1VOF16-31)*7, *9	<BPA1k_base>+ (0040_H to $007C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA1k_base>+ (0080_H to $00FC_H$)	Erased	Can not Write*8	—*8	—
VOF Program completion flag (BPA1VOFC0-7)*7, *9	<BPA1k_base>+ (0100_H to $011C_H$)	5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA1k_base>+ (0120_H to $013C_H$)	Erased	Can not Write*8	—*8	—
VOF Program completion flag (BPA1VOFC16-31)*7, *9	<BPA1k_base>+ (0140_H to $017C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*8	<BPA1k_base>+ (0180_H to $01FC_H$)	Erased	Can not Write*8	—*8	—
OTP Setting of User area 2*4	<BPA1k_base>+ (0200_H to $021C_H$)	All FFFF FFFF$_H$	—	—	0
OTP Setting of User area 3*4	<BPA1k_base>+ (0220_H to $023C_H$)	All FFFF FFFF$_H$	—	—	1
Reserved(need to be programmed)*6	<BPA1k_base>+ (0240_H to $02FC_H$)	All FFFF FFFF$_H$	—	—	2 to 7
Reserved*8	<BPA1k_base>+ (0300_H to $03FC_H$)	Erased	Can not Write*8	—*8	—
Customer ID Protection & Erase Counter Setting for user area2*5	<BPA1k_base>+ (0400_H to $043C_H$)	All FFFF FFFF$_H$	Customer ID A	—	16 to 17
Customer ID Protection & Erase Counter Setting for user area3*5	<BPA1k_base>+ (0440_H to $047C_H$)	All FFFF FFFF$_H$	Customer ID A	—	18 to 19
Reserved(need to be programmed)*6	<BPA1k_base>+ (0480_H to $05FC_H$)	All FFFF FFFF$_H$	Customer ID A	—	20 to 31
Reserved*8	<BPA1k_base>+ (0600_H to $07FC_H$)	Erased	Can not Write*8	—*8	—

Note 1. This is the value of valid area at the shipping. Invalid area is erased at the shipping.

Note 2. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary for programming the data.

Note 3. When On-chip debug and S_OPBT3.CFRPF is “0”, the Customer ID A authentication is needed to read.

Note 4. For details of this function, see **Section 47.5.2, OTP (One Time Programmable) Function**.

Note 5. For details of this function, see **Section 47.9.5, Block Protection/Erase Counter Enable setting for User Area, User Boot Area, Extended Data Area**.

Note 6. For this reserved area, program FFFF FFFF$_H$ when programming Block Protection Area for each FPSYS.

Note 7. This area will be updated automatically when programming Block Protection Area for each FPSYS.

Note 8. This area cannot be programmed. When this area is read, it behaves as if it is erased.

Note 9. For details of this function, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Note 10. <BPA0k_base> (k = f, b), <BPA1k_base> (k = f, b)

47.9.1 BPA0VOFn — Valid Option Byte Flag n for Block Protection Area for FPSYS0 (n = 0 to 7,15 to 31,39)

This flag indicates that the related option bytes are programmed or not.
The relation between each BPA0VOFn and option bytes are shown in **Table 47.27**.
This flag is referred by flash sequencer in order to check whether the related option bytes are programmed or not.
This flag will be updated automatically when programming Block Protection Area for FPSYS0.
The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <BPA0k_base> + 0000$_H$ + 04$_H$ x n

Value at the shipping: 5AA5 A55A$_H$ (valid area)
Erased (invalid area)

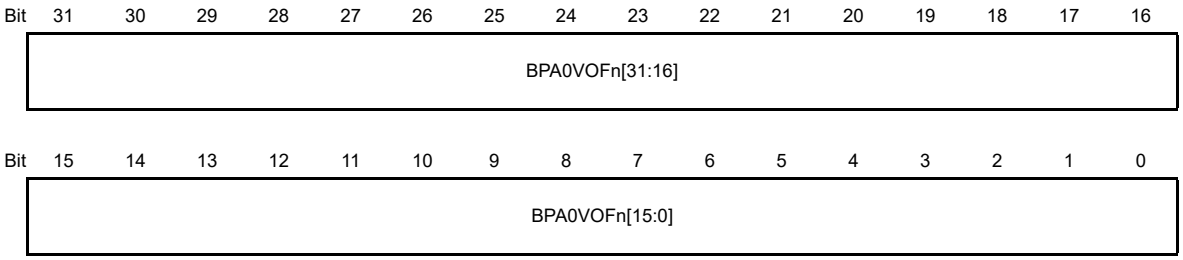


Table 47.29 BPA0VOFn Contents

Bit Position	Bit Name	Function
31 to 0	BPA0VOFn [31:0]	Program completion flag. 5AA5 A55A$_H$: The related option bytes are programmed. Other than above: The related option bytes are not programmed.

47.9.2 BPA0VOFCn — VOF Program completion flag n for Block Protection Setting Area for FPSYS0 (n = 0 to 7,15 to 31,39)

This flag indicates that BPA0VOFn is programmed or not.
The relation between each BPA0VOFCn and option bytes are shown in **Table 47.27**.
This flag is prepared to check whether the related option bytes are programmed or not by user program.
This flag will be updated automatically when programming Block Protection Area for FPSYS0.
The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <BPA0k_base> + 0100$_H$ + 04$_H$ x n

Value at the shipping: 5AA5 A55A$_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BPA0VOFCn[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BPA0VOFCn[15:0]															

Table 47.30 BPA0VOFCn Contents

Bit Position	Bit Name	Function
31 to 0	BPA0VOFCn [31:0]	BPA0VOFn Program completion flag. 5AA5 A55A$_H$: BPA0VOFn is programmed. Other than above: BPA0VOFn is not programmed.

47.9.3 BPA1VOFn — Valid Option Byte Flag n for Block Protection Area for FPSYS1 (n = 0 to 7,16 to 31)

This flag indicates that BPA1VOFn is programmed or not.
The relation between each BPA1VOFn and option bytes are shown in **Table 47.28**.
This flag is referred by flash sequencer in order to check whether the related option bytes are programmed or not.
This flag will be updated automatically when programming Block Protection Area for FPSYS1.
The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <BPA1k_base> + 0000$_H$ + 04$_H$ x n

Value at the shipping: 5AA5 A55A$_H$ (valid area)
Erased (invalid area)

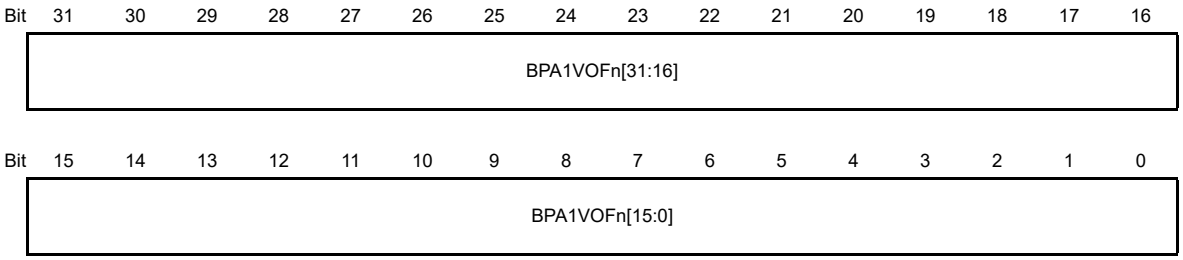


Table 47.31 BPA1VOFn Contents

Bit Position	Bit Name	Function
31 to 0	BPA1VOFn [31:0]	Program completion flag. 5AA5 A55A$_H$: The related option bytes are programmed. Other than above: The related option bytes are not programmed.

47.9.4 BPA1VOFCn — VOF Program completion flag n for Block Protection Area for FPSYS1 (n = 0 to 7,16 to 31)

This flag indicates that BPA1VOFn is programmed or not.
The relation between each BPA1VOFCn and option bytes are shown in **Table 47.28**.

This flag is prepared to check whether the related option bytes are programmed or not by user program.
This flag will be updated automatically when programming Block Protection Area for FPSYS1.
The updated value will be 5AA5 $A55A_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <BPA1k_base> + 0100_H + 04_H x n

Value at the shipping: 5AA5 $A55A_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BPA1VOFCn[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BPA1VOFCn[15:0]															

Table 47.32 BPA1VOFCn Contents

Bit Position	Bit Name	Function
31 to 0	BPA1VOFCn [31:0]	BPA1VOFn Program completion flag. 5AA5 $A55A_H$: BPA1VOFn is programmed. Other than above: BPA1VOFn is not programmed.

47.9.5 Block Protection/Erase Counter Enable setting for User Area, User Boot Area, Extended Data Area

For details of this function, see **Section 47.5, Code Flash/Data Area Protection** and **Section 51.15, Erase Counter Area**.

Non-secure Code flash can be protected from Programming/Erase by Customer ID A, B or C authentication which has a function called Block Protection. Erase Counter can be enabled/disabled for each erase block respectively. These functions can be set by 4 bits per Block of code flash.

Table 47.33 Customer ID and Erase Counter Setting 4 bit setting per block

Customer ID & Erase Counter Setting 4 bit setting per block	
Bit 3	OTP setting for ID Protection Setting*2
Bit 2	Erase Counter enable setting*1*2 0: Erase Counter of the block is enabled. 1: Erase Counter of the block is disabled.
Bit 1 to 0	ID selection 00_B: Customer ID C 01_B: Customer ID B 10_B: Customer ID A 11_B: No ID

Note 1. Erase counter is not supported for Extended Data Area.

Note 2. These bits are one time programmable (OTP) and user cannot return these bits to 1 once these bits are set to 0.

Table 47.34 Mapping of Block Protection/Erase Counter Enable Setting (User Area m and User Boot Area m) (m*1 = 0, 1) (k = f, b)

Address	**Bit**	**Protection Area**
<BPA0k_base>+ 0400_H+ 40_H x m	[3:0]	Block 0 of User Area m
	[7:4]	Block 1 of User Area m
	[11:8]	Block 2 of User Area m
	[15:12]	Block 3 of User Area m
	[19:16]	Block 4 of User Area m
	[23:20]	Block 5 of User Area m
	[27:24]	Block 6 of User Area m
	[31:28]	Block 7 of User Area m
:	:	:
<BPA0k_base>+ 0420_H+ 40_H x m	[3:0]	Block 64 of User Area m
	[7:4]	Block 65 of User Area m
	[11:8]	Block 66 of User Area m
	[15:12]	Block 67 of User Area m
	[19:16]	Block 68 of User Area m
	[23:20]	Block 69 of User Area m
	[31:24]	Reserved
<BPA0k_base>+ (0424_H to 0438_H) + 40_H x m	[31:0]	Reserved*2
<BPA0k_base>+ $043C_H$ + 40_H x m	[27:0]	Reserved
	[31:28]	User Boot Area m

Note 1. Block Protection/Erase Counter Enable setting area of m = 0 is shared with User Area 1, User Boot Area 1 when Double Map Mode. Block Protection/Erase Counter Enable setting area of m = 1 is reserved when Double Map Mode.

Note 2. These reserved words need to be programmed when programming Block Protection Area for FPSYS0.

Table 47.35 Mapping of Block Protection (Extended Data Area) (k = f, b)

Address	Bit	Protection Area
<BPA0k_base>+ $06E0_H$	[3:0]	Extended Data Area
	[31:4]	Reserved

Table 47.36 Mapping of Block Protection/Erase Counter Enable Setting (User Area m) (m[*1] = 2, 3) (k = f, b) [For U2A-EVA U2A16 mode/U2A16]

Address	Bit	Protection Area
<BPA1k_base>+ 0400_H+ 40_H x (m - 2)	[3:0]	Block 0 of User Area m
	[7:4]	Block 1 of User Area m
	[11:8]	Block 2 of User Area m
	[15:12]	Block 3 of User Area m
	[19:16]	Block 4 of User Area m
	[23:20]	Block 5 of User Area m
	[27:24]	Block 6 of User Area m
	[31:28]	Block 7 of User Area m
:	:	:
<BPA1k_base>+ 0420_H+ 40_H x (m - 2)	[3:0]	Block 64 of User Area m
	[7:4]	Block 65 of User Area m
	[11:8]	Block 66 of User Area m
	[15:12]	Block 67 of User Area m
	[19:16]	Block 68 of User Area m
	[23:20]	Block 69 of User Area m
	[31:24]	Reserved
<BPA1k_base>+ 0424_H to $043C_H$ + 40_H x (m - 2)	[31:0]	Reserved[*2]

Note 1. Block Protection/Erase Counter Enable setting area of m = 2 is shared with User Area 3 when Double Map Mode.
Block Protection/Erase Counter Enable setting area of m = 3 is reserved when Double Map Mode.

Note 2. These reserved words need to be programmed when programming Block Protection Area for FPSYS1.

47.10 Switch Area

Switch Area can be protected by the following functions.

(1) ID authentication
When Customer ID A authentication protection is effective, invalid area of Switch Area is not changed by "Switch Programming", "Switch Erasure" command, and flash sequencer enters "Command Lock" state.
This protection doesn't affect the access from ICUMHA.

47.11 TAG Area

TAG Area can be protected by the following functions.

(1) ID authentication
When Customer ID A authentication protection is effective, TAG Area is not changed by "TAG Update", "TAG Erasure" command, and flash sequencer enters "Command Lock" state.
This protection doesn't affect the access from ICUMHA.

47.12 Usage Note

(1) OTP setting for Double Map Mode
The setting for User Area 0 also affects User Area 1, the setting for User Area 2 also affects User Area 3, the setting for User Boot Area 0 also affects User Boot Area 1 in case of Double Map Mode.
Make sure that the contents of the block in both valid and invalid area is completely same before setting the OTP of the block. Otherwise, unintentional fallback may happen when switching the valid area and invalid area of User Area/User Boot Area.

(2) Counter measure for prohibiting unsuitable fallback
The settings of this product can be fallback by updating Switch/TAG area if the old settings still remain in the invalid area of Configuration setting/Security setting/Block protection area.
It means that the security function such as OTP/Block protection setting is also able to be fallback.
It is necessary to erase the old settings in invalid area or copy the new settings to invalid area after the old settings become unnecessary in order to prohibit unsuitable fallback.

(3) P6_15, P10_7, P10_8 can be used for security function
Do not enable pull-up resistor or output of P6_15 when the security function is used.
For details of the security function, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Section 48 Intelligent Cryptographic Unit/Master (ICUMHA)

The Intelligent Cryptographic Unit/Master (ICUMHA) is a hardware security module (HSM). ICUMHA can be activated by Option Byte setting.

This section describes about resources in which an accessibility is controlled by ICUMHA when ICUMHA is activated. For details of ICUMHA, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

48.1 Overview

48.1.1 Number of Units

This microcontroller has the following number of units.

Table 48.1 Number of Units

Product Name	RH850/ U2A-EVA (516 pins)	RH850/ U2A16 (516 pins)	RH850/ U2A16 (373 pins)	RH850/ U2A16 (292 pins)	RH850/ U2A8 (373 pins)	RH850/ U2A8 (292 pins)	RH850/ U2A6 (292 pins)	RH850/ U2A6 (176 pins)	RH850/ U2A6 (156 pins)	RH850/ U2A6 (144 pins)
Number of Units	1	1	1	1	1	1	1	1	1	1
Name	ICUMHAn (n = 0)									

48.1.2 Features

The Intelligent Cryptographic Unit/Master (ICUMHA) is a hardware security module (HSM).

The features of ICUMHA are described below.

- ICUMHA integrates an Intelligent Cryptographic Unit Processor (ICUP) that controls the ICUMHA sub-system and runs the user-defined security services.
- ICUMHA integrates an accelerator that supports a block cipher algorithm based on the AES (Advanced Encryption Standard, FIPS PUB 197).
- ICUMHA integrates several public key cryptographic algorithms like RSA, ECC.
- ICUMHA integrates an accelerator that supports HASH function based on SHA-1, SHA-224 and SHA-256 (Secure Hash Algorithm, FIPS-PUB 180).
- ICUMHA integrates a true random number generator.
- ICUMHA can be used as a slave peripheral to run security services upon request. Also the ICUMHA handles processing of other user-defined security services (for example, checking contents of memory for completeness and integrity) and operates as a master handling procedure to strengthen measures for security.
- ICUMHA has an exclusive read access and programming access to a specific area of non-volatile (flash) memory which contains confidential data (keys and certificates) and highly confidential code for operating security services.

48.2 ICUM_CMDREG

48.2.1 Registers

48.2.1.1 ICUM_ACTFLAG – ICUMHA Activating Flag Register

This register indicates active state of ICUMHA.

Access: This register can be read in 32-bit units.

Address: FF1F 0018$_H$

Value after reset: Specified by user

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ICUP R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
CPU (PE) R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICUMACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1	0/1*1
ICUP R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
CPU (PE) R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This value is dependent on the value in the flash memory which is specified by the user.

Table 48.2 ICUM_ACTFLAG Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICUMACT[1:0]	ICUMHA active status 00: ICUMHA is inactive 11: ICUMHA is active

48.3 Clock Monitor

For details of clock monitor for ICUMHA (CLMAS), see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

48.3.1 Interrupt Requests and Error Notifications

This module has no interrupt requests for CPU(PE).

The error notifications of this module are listed in the following table.

Table 48.3 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
Clock monitor error for CLK_HBUS	CLMAS over/under frequency error	53	—

48.4 Flash Memory

48.4.1 Programming Sequence while ICUMHA is Enabled

If the ICUMHA is enabled, arbitration between the ICUP and CPU(PE) by software is required to avoid flash access conflict. For details, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

48.5 Operation

48.5.1 Enabling ICUMHA

CAUTION

When ICUMHA is enabled, default start mode (WDTBA) must be disabled.

Always select software trigger start mode by setting OPBT0.OPWDRUNA = 0.
For OPBT0, see Section 51.12.7, OPBT0 — Option Byte 0.

Section 49 Secure Watchdog Timer (SWDT)

About SWDT, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

Section 50 Debugging and Calibration

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001™-2003 Class 1, a Nexus debug interface standard.

Production devices support Nexus Class 1 debug functionality. Emulation devices support Nexus Class 3 debug functionality.

CAUTION

The debug functions described in this section are supported by the microcontroller but whether these functions can actually be used depends on the debugger. For details on debugging, see the user's manual of the debugger.

50.1 Debug Interface

This microcontroller supports the following debug interfaces:

(1) Nexus JTAG Interface

(2) Low Pin Count Debug Interface (4-pin) (LPD4).

(3) In U2A-EVA emulation devices, a 4-lane Aurora trace port is provided.

(4) Trigger Input / Output pin ($\overline{\text{EVTI}}$/$\overline{\text{EVTO}}$)

On-chip debugging can be performed by using these debug interfaces.

Table 50.1 Debug Interface Modes

Serial programming mode*1	Normal operating mode/user boot mode*1	Debug ports		Option bytes*3		
2-wire UART/ CSI*4		$\overline{\text{TRST}}$	TDI (JP0_0)*2	DRDY_EN	JPDBGIF_EN	JP0 port usage
1	0	x	x	x	x	Serial programming *5
0	1	0	x	x	0	GPIO
0	1	0	x	x	1	GPIO*8
0	1	1	0	x	0	GPIO (security level 2 *6)*7
0	1	1	0	x	1	LPD4 *8
0	1	1	1	x	0	JTAG Boundary Scan(security level 2*6)
0	1	1	1	1	1	Nexus (with $\overline{\text{DRDY}}$) & JTAG Boundary Scan*8
0	1	1	1	0	1	Nexus (without $\overline{\text{DRDY}}$) & JTAG Boundary Scan*8

Note: x = don't care

Note 1. For details, see **Section 5, Operating Modes**.

Note 2. Latch output at rising edge of $\overline{\text{TRST}}$.

Note 3. For details, see **Section 47, Basic Hardware Protection (BHP)**.

Note 4. For details, see **Section 51, Flash Memory**.

Note 5. Boot firm changes the function of the terminal to 2-wire UART or CSI interface.

Note 6. For details, see **Section 47, Basic Hardware Protection (BHP)**.

Note 7. Operation at the time of reset assertion in the data output is not guaranteed. When use GPIO, $\overline{\text{TRST}}$ must be "0".

Note 8. The JP0 port register settings must be their initial value.

50.2 Run Control Functions

(1) Debug functions

These basic debug functions are implemented:

- Downloading a user-created program
- Reading and writing the memory and registers
- Running a user-created program starting at any address

(2) On-chip break functions

A maximum of 12 breakpoints can be specified at any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified for the data access address. Asynchronous Break function is supported.

The following break functions are supported:

- Relay break function: Each CPU can be configured to stop when another CPU hits a breakpoint.
- Individual break function: Each CPU can be configured to continue executing when another CPU hits a breakpoint.

CAUTION

Whether the individual break function can be used or not depends on the debugger.

(3) Software break function

Software breakpoints can be specified at any execution address.

(4) Forced break function

Execution of a user-created program can be interrupted forcibly by a debugger.

(5) Peripheral break control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

(6) Forced reset function

This device (microcontroller) can be forcibly reset by a debugger.

(7) Reset mask function

A reset factor (external reset, software reset, and ECM reset) can be masked.

(8) Event detection function

Events can be detected based on execution address, data access address, data value, data value range, and sequential execution.

(9) Trigger input interface

This microcontroller incorporates an event trigger input interface to acknowledge external events. It can acknowledge an external event in response to an input from the $\overline{\text{EVTI}}$ pin.

(10) Trigger output interface

An event trigger output interface is included in this microcontroller to notify an external debug device of event detection. Output from the $\overline{\text{EVTO}}$ pin reports detection of an event trigger to the outside. Debug trigger registers control trigger output to an external debug device via a request from the CPU. Up to eight debug trigger registers are supported. Each register supports 32 software triggers.

When $\overline{\text{TRST/LPDRST}}$ is low level, the system reset signal is output from $\overline{\text{EVTO}}$.

(11) Debug interrupt interface function

Execution of a user program can be forcibly suspended by asserting an input signal on the $\overline{\text{EVTI}}$ pin from the outside.

(12) Multi core debug function

The following multi-core debug functions are supported for each CPU: synchronization functions (including reset, execution, and break), synchronous setting, and simultaneous tracing for multiple cores.

50.3 Calibration Functions

This microcontroller includes emulation RAM as emulation memory for the on-chip flash memory.

(1) Real-time RAM monitoring (RRM)

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

(2) Dynamic memory modification (DMM)

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

(3) Emulation RAM (ERAM) [For U2A-EVA and U2A6 Only]

The Emulation RAM (CERAM and GERAM) is available in the Code Flash Emulation Function. The ROM data can be dynamically modified during execution of a user program via the Emulation RAM which has mapped to the Code Flash area.

(4) Calibration Function Unit (CFU) [For U2A-EVA and U2A6 Only]

The Calibration Function Unit (CFU) controls access to Cluster ERAM of Flash emulation memory.

(5) Global Calibration Function Unit (GCFU)

The Global Calibration Function Unit (GCFU) controls access to Global ERAM [For U2A-EVA only] and Cluster RAM of Flash emulation memory [For U2A16, U2A8].

50.4 Trace Control Functions

This microcontroller provides several trace functions including branch PC trace and data trace for each CPU, and DMA data trace.

(1) Trace RAM [For U2A-EVA and U2A6 Only]

This microcontroller has 64 KB of trace RAM [For U2A-EVA only], 32KB of trace RAM [For U2A6 only]. The trace information in the trace RAM is accessible via Nexus and LPD4 debug interfaces.

(2) Trace filter RAM [For U2A-EVA and U2A6 Only]

- Advanced data trace filtering allows selection of small 4 byte sections of data which can be individually enabled for data trace.
- Trace filter RAM does not have ECC.

(3) Software trace

This function enables the obtaining of user program execution histories, data changes, etc. The software trace information can be output to the trace RAM or via LPD4, Aurora debug interface.

(4) Trace over reset

Remaining trace data is output after a reset.

(5) Global timestamp information

The global timestamp information is the absolute time from (absolute time 0) when the trace buffer is cleared.

(6) The trace message format conforms to the Nexus standard

(7) Windowed instruction trace or data trace

50.5 Performance Measurement Function

This microcontroller provides a two-event interval time difference measurement function to measure the time difference between the points at which two events occur.

(1) Two-event interval time difference measurement function

- Provides a function to measure the time difference (maximum time, minimum time, accumulation time) between two events (measurement start event to measurement stop event).
- Provides a function to measure the number of measurements (the number of measurement stop events) for two-event intervals (measurement start event to measurement stop event).
- Provides a threshold value violation detection function for two-event intervals (measurement start event to measurement stop event).
- Measurement clock: Debugging clock or CPU clock
- Measurement counter: 32-bit counter × 4 channels
- Measurement items: Maximum value, minimum value, accumulation value, measurement count (measurement stop event count)
- Count conditions: Number of debugging clock cycles, measurement count (measurement stop event count); number of CPU clock cycles, instruction execution or other events.
- Measurement limitations: When the interval between the measurement start event and the measurement stop event (from measurement start to measurement stop) or between a measurement stop event and the next measurement start event (from measurement stop to measurement start) is short, it may not be possible to correctly detect the measurement start or measurement stop events due to synchronization processing (from the system clock to the debugging clock).
  Note that the required interval for detection depends on the system clock and the debugging clock frequencies.
- Measurement of the cache hit rate
- Measurement of the number of CPU cycles when a pipeline stall occurred

50.6 Debug Support Function

(1) Hot plug-in function

Debugging can be started in normal operating mode without external reset input.

NOTE

When the hot plug-in function is used in power save mode, the INTDCUTDI interrupt is required to return from power save mode as the wake-up process. About the INTDCUTDI interrupt, refer to Appendix file "**Interrupt_table.xlsx**".

(2) Security function

To prevent the contents of the flash memory from being read by an unauthorized person, a 256-bit ID code must be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code in the microcontroller, the flash memory cannot be accessed.

For details, see **Section 47, Basic Hardware Protection (BHP)**.

(3) Halt after reset

The debugger can control the start of CPU0 after internal reset release.

This must be set by using an option byte. For details, see **Section 47, Basic Hardware Protection (BHP)**.

(4) ICUMHA debug function

For details, see the separate document for security functions.

(5) GTM debug function

For details, see **Section 38, Generic Timer Module (GTM)**.

(6) Device ID function

Device ID (device identification) information can be acquired via the debug interface (Nexus).

(7) Reset output pin

The reset output pin ($\overline{\text{RESETOUT}}$) reflects the state of the internal reset condition.

For details, see **Section 2, Pin Functions**.

50.7 Peripheral Break Control

The peripheral break function generates a stop request for the peripheral modules as a break signal from each CPU if the user-created program is stopped, for instance upon a breakpoint hit.

During peripheral break, the peripheral modules operate as follows:

1. Modules that stop unconditionally regardless of the EPC.SVSTOP setting WDTBn (n = 0 to 3), SWDT0, and TPTMn (n = 0 to 3, Interval timer and Free-run timer)
2. Modules that stop when EPC.SVSTOP = 1 and EPC_SVSTOPn setting OSTMn (n = 0 to 9), ENCAn (n = 0, 1), GTM, ADCn (n = 0 to 2), PWM, RTCA, TSG3n (n = 0, 1), TAUJn (n = 0 to 3), TPBAn (n = 0 to 1), TAUDn (n = 0 to 2), TAPAn (n = 0 to 3), RLIN3n (n = 0 to 23), LTSC, TPTMn (n = 0 to 3, Up timer) and LPS modules
3. Module that stop when EPC_SVSTOP58 setting, but regardless of the EPC.SVSTOP setting WDTBA

50.8 Hot Plug-in in Each Mode

50.8.1 Run Mode

When Hot Plug-in has occurred in Run mode, it is necessary to maintain the current state.

50.8.2 STOP/DeepSTOP Mode

When Hot Plug-in has occurred in STOP/DeepSTOP mode, it is necessary to transfer in RUN mode.

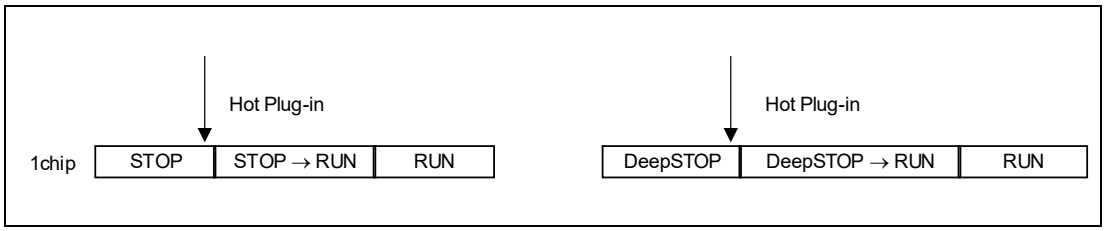


Figure 50.1 Hot Plug-in in STOP/DeepSTOP Mode

50.8.3 Cyclic RUN Mode

When Hot Plug-in has occurred in Cyclic Run mode, it is necessary to transfer in DeepSTOP mode and wakeup to RUN mode by the following reason.

All regulators in ISO are made operating state in OCD mode.

To do ID authentication, it's necessary to transmit the data of Flash extra.

Then, on the occurrence of interrupt of "DCUTDI Rising Edge Detection interrupt (INTDCUTDI)", it is necessary for a transfer order to be executed to DeepSTOP mode by user (software) if on-chip debugging is performed.

The sequence of the time when to enter to OCD mode in Cyclic RUN mode is mentioned below.

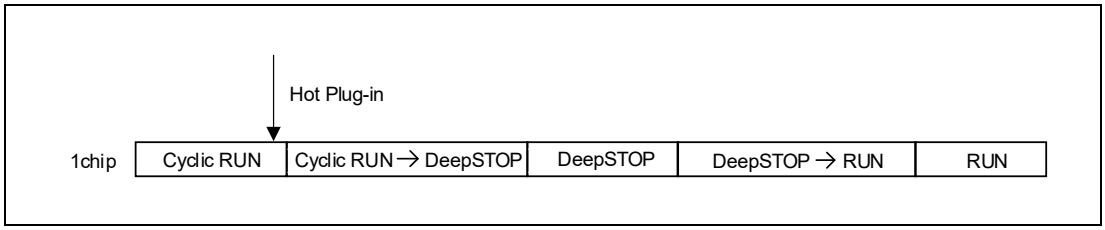


Figure 50.2 Hot Plug-in in Cyclic RUN Mode

50.8.4 Cyclic STOP Mode

When Hot Plug-in has occurred in Cyclic STOP mode, it is necessary to transfer in Cyclic RUN mode.

Then, on the occurrence of wakeup factor of INTDCUTDI, it is necessary for a transfer to be executed as the same case of Cyclic RUN mode.

The sequence of the time when to enter to OCD mode in Cyclic STOP mode is mentioned below.

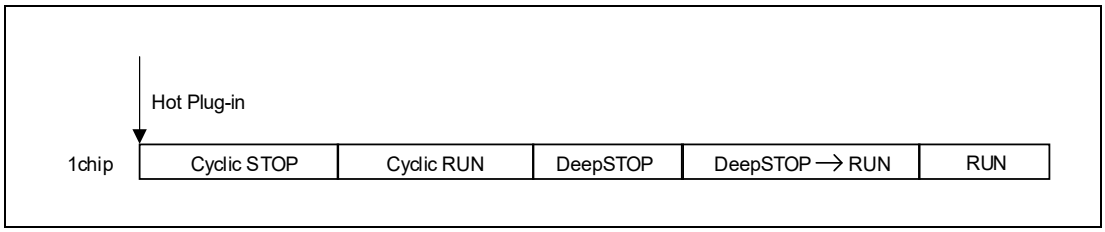


Figure 50.3 Hot Plug-in in Cyclic STOP Mode

50.9 Aurora Trace Interface

50.9.1 Overview

Aurora is lightweight link-layer protocol for communicating over high-speed serial lanes.

Nexus introduces this protocol in IEEE-ISTO 5001-2012 (Nexus standard).

This product supports a simplex TX-only interface. 4-lane is used.

Aurora for this product is fully compatible with Nexus standards and Xilinx's Aurora Protocol spec V2.2.

Table 50.2 shows the Aurora reference documents for this product.

Table 50.2 Reference Documents

Reference Document	Document Name	Reference
Aurora 8b / 10b protocol specification V2.2	aurora_8b10b_protocol_spec_sp002.pdf	http://www.xilinx.com/
The Nexus 5001 Forum™ Standard for a Global Embedded Processor Debug Interface V3.0.1	ieee_isto_5001_2012_301_Final.pdf	http://www.nexus5001.org.

50.10 Advanced User Debugger RAM Monitor (AUDR)

AUD (Advanced User Debugger) RAM monitor (AUDR) is implemented to help debug the user program. AUDR can read or write resources mapped memory space (ex the peripheral register or internal memory) during LSI operation. AUDR can't operate during System Reset.

50.11 Cautions on Using On-Chip Debugger

(1) Handling of devices used for debugging
Do not install a device that was used for debugging on a mass-produced product. This is because the flash memory will be rewritten during system debugging and thus the write / erase count of flash memory cannot be guaranteed.

(2) This product does not support hot plug out for power off (including removal of the connector) of the debug tool during debug mode. Do not turn off the power of the Nexus tool (including removal of the connector) in debug mode.

(3) When ending on-chip debugging, set the $\overline{\text{TRST}}$ pin and external reset pin to low level.

(4) When a debugger is used, the program written to the microcontroller before the OCD emulator is ready to communicate with the microcontroller is executed from the reset vector. To stop the operation of the program, release the $\overline{\text{TRST}}$ before $\overline{\text{RESET}}$.

50.12 Cautions on Using U2A-EVA to Emulate U2A8 Functions [For U2A8 Only]

Though U2A-EVA has U2A8 mode to emulate U2A8, there is a restriction as follows.

Hence, be cautious about the following difference in case of emulating U2A8 by using U2A-EVA.

- In case of accessing to below areas, the behavior of U2A8 is different from U2A-EVA.
  - Areas related to ETNB1 (ETNB1, E7GE00, E7GE01 and HB94MECC)
  - Areas related to MSPI6, MSPI7 and MSPI8 (MSPIn, E7MS0n and MSPInINTIF (n = 6 to 8))
  - Areas related to RLIN3n (n = 12 to 23)

In U2A8 mode of U2A-EVA, the error response is signaled, because these areas are prohibited to access.

(Hence, above functions can not be emulated on U2A8 mode of U2A-EVA.)

In U2A8, above functions including related functions are accessed without error response.

For details, see **Section 4, Address Space**.

50.13 Cautions on Using U2A-EVA to Emulate U2A6 Functions [For U2A6 Only]

U2A-EVA does not have U2A6 mode to emulate U2A6.

Hence, be cautious about the following differences in case of emulating U2A6 by using U2A-EVA.

- Memory size of Flash and RAM (for details, see **Section 1, Overview**)
- Number of IP units and channels (for details, see **Section 1, Overview**)
  ==> Unmapped area access error for unimplemented resources in U2A6 is not supported.

- MSPI RAM area of U2A6 is different from U2A-EVA.

  For details, see **Table 19.19, Areas of MSPI RAM**.

- Especially for RS-CANFD, below configurations of U2A6 are different from U2A-EVA.
  - Memory size of RAM
  - Number of TXMB, AFL entry and RXMB
  - Number of channels per unit

  For details, see **Section 23, CANFD Interface (RS-CANFD)**.

- In case of accessing to FA00 0000_H to FA01 $7FFF_H$, the behavior of U2A6 is different from U2A-EVA.
  In U2A-EVA, Instrumentation RAM is accessed without error response.
  In U2A6, the error response is signaled, because this area is prohibited to access.
  For details, see **Section 4, Address Space**.

Section 51 Flash Memory

This product incorporates Code Flash Memory and Data Flash Memory.

Product Name	Code Flash Memory		Data Flash Memory	
	User Area	User Boot Area	Data Area	Exclusively for ICUMHA
RH850/U2A-EVA (U2A16 mode)	16 Mbytes	2x 64 Kbytes	512 Kbytes	64 Kbytes
RH850/U2A-EVA (U2A8 mode)	8 Mbytes	2x 64 Kbytes	256 Kbytes	64 Kbytes
RH850/U2A16	16 Mbytes	2x 64 Kbytes	512 Kbytes	64 Kbytes
RH850/U2A8	8 Mbytes	2x 64 Kbytes	256 Kbytes	64 Kbytes
RH850/U2A6	6 Mbytes	2x 64 Kbytes	192 Kbytes	64 Kbytes

51.1 Features

51.1.1 Units

This product contains three Flash Programming Systems (FPSYS0, FPSYS1, FPSYS2).

Each FPSYS contains Flash Memory and flash sequencer.

Each flash sequencer contains Flash Application Command Interface (FACI).

51.1.2 Register Base Address

Register base addresses are listed in the following table.

Register addresses are given as offsets from the base addresses in general.

Table 51.1 Register Base Address

Base Address Name	Base Address	Bus Group
<CCIB0_base>	FFFB 0800_H	P-Bus Group 0
<SCDS_base>	FFCD 0000_H	P-Bus Group 6L
<FACI0_base>	FFA1 0000_H	P-Bus Group 6L
<FACI1_base>	FFA1 4000_H	P-Bus Group 6L
<FACI2_base>	FFA1 8000_H	P-Bus Group 6L

51.1.3 Clock Supply

Flash Programming System clock supply is shown in the following table.

Table 51.2 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
FACIn	Peripheral clock	CLK_LSB

51.1.4 Interrupt Requests and Error Notifications

Flash Programming System supports an interrupt to indicate completion of processing by the flash sequencer.

Table 51.3 Interrupt and DMA/DTS Requests

Unit Interrupt Name	Description	Interrupt Number	sDMA Trig-ger Number	DTS Trigger Number
INTFL0ENDNM	FPSYS0 Flash sequencer processing end interrupt	34	—	—
INTFL1ENDNM	FPSYS1 Flash sequencer processing end interrupt	36	—	—
INTFL2ENDNM	FPSYS2 Flash sequencer processing end interrupt	38	—	—
INTDMAFL0	DMA request (FPSYS0 DMA programming) interrupt	—	Group0-154	Group3-34
INTDMAFL1	DMA request (FPSYS1 DMA programming) interrupt	—	Group0-155	Group3-35
INTDMAFL2	DMA request (FPSYS2 DMA programming) interrupt	—	Group0-156	Group3-36

The error notifications related to Flash Memory are listed in the following table.

Table 51.4 Error Notifications

Error Notification	Description	ECM Error Number	Error Response to bus master
Flash access error*1	FPSYS0 Flash access error.	32	—
	FPSYS1 Flash access error.		—
	FPSYS2 Flash access error.		—
FACI reset transfer error	FPSYS0 FACI reset transfer error.	33	—
	FPSYS1 FACI reset transfer error.		—
	FPSYS2 FACI reset transfer error.		—
FACI reset transfer warning	FPSYS0 FACI reset transfer warning.		—
	FPSYS1 FACI reset transfer warning.*2		—
BIST parameter transfer error	BIST parameter transfer error.	34	—
Code Flash read BGO error	See **Section 51.8.2, Background Operation**.	—	√
Data Flash read BGO error	See **Section 51.8.2, Background Operation**.	—	√

Note 1. See the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.
Note 2. For U2A-EVA and U2A16.

51.2 Overview

51.2.1 Functional Overview

This product incorporates 16 Mbytes (U2A8: 8 Mbytes, U2A6: 6 Mbytes) code flash allowing high-speed accesses, which enables each CPU to access this flash memory.

This memory can be reprogrammed while it is placed on an application system. This can shorten the system development period and significantly improve the serviceability after the system is delivered.

This product also has 576 Kbytes (U2A8: 320 Kbytes, U2A6: 256 Kbytes) data flash that is available for storing EEPROM data.

Code Flash

- Capacity: Up to 16 Mbytes of User Area and 2x 64 Kbytes of User Boot Area.
- Multi banked configuration.
- Program unit: 512 bytes
- Erase unit: 16 Kbytes for 8 blocks and 64 Kbytes for remaining blocks in each User Area
- OTP (One Time Programmable) is supported for each block.
- Address map swapping between banks is supported.

Data Flash

- Data Area:
  - Capacity: Up to 512 Kbytes (Two Data Areas: 256 Kbytes + 256 Kbytes) and 64K bytes for ICUMHA exclusively
  - Program unit: 4, 8, 16, 32, 64, 128 bytes
    DMA can initiate 4-byte program in multiple time without software overhead. (within one Data Area)
  - Erase unit: Nx 4 Kbytes (N = 1, 2, 3 …) (within one Data Area)

- Hardware Property Area:

  The settings of this product can be configured in Hardware Property Area in Data Flash Memory.

  Hardware Property Area consists of Configuration Setting Area, Extended Data Area, Security Setting Area, Block Protection Area, Switch Area, TAG Area and Erase Counter Area.

  - Configuration Setting Area: To store the System Configuration Parameters. (Flash Option Byte, Reset Vector, Software Configuration Option Byte, etc.)
  - Security Setting Area: To store the Security Parameters. (ID Codes, Security Setting flag, etc.)
  - Block Protection Area: To store the Code Flash Protection Settings. (OTP flag, etc.)
  - Switch Area and TAG Area: To update Configuration Setting Area, Security Setting Area and Block Protection Area in an atomic and robust way.
  - Erase Counter Area: To store the Erase Counter.
  - Extended Data Area To store any data to use by user software.
    Capacity: 2 Kbytes

Program unit: 4, 8, 16, 32, 64, 128 bytes
Erase unit: 2 Kbytes

Other Functions

- Programming method
  - Serial programming: Programming of flash memory by external flash memory programmer through serial interface.
  - Self-programming: Programming of flash memory by a user program written on Code Flash already.
- Support for security functions to protect against illicit tampering and illicit reading data in flash memory
- Support for protection functions to protect against erroneous programming/erasure into the flash memory
- ECC support for error detection and correction for both Code Flash and Data Flash
  - Reading erased Code Flash results in an ECC error.
  - Reading erased Data Flash results in an ECC error.
- Background Operation (BGO) support
  The readability of flash memory during programming/erasure within one FPSYS is as follows.
  - Code Flash read is possible during Data Flash programming/erasure
  - Code Flash read from other bank is possible during programming/erasure of a bank of Code Flash
  - Data Flash read from other Data Area is possible during programming/erasure of a Data Area of Data Flash
  - Data Flash read is possible during Code Flash programming/erasure
- Multi FPSYS Operation support
  - The Flash memories that are belonging to the different FPSYS can be programmed/erased simultaneously.
- Suspend/Resume support for both Code Flash and Data Flash (Data Area and Extended Data Area)
  - Programming / erasure operation can be suspended. Suspended operation can be resumed. It can be used for interruption processing received during self-programming.
- Blank Check Area for code flash memory

  This area is used for checking that User Area or User Boot Area is blank or not. Which area (User or User Boot) to specify is selected by BLCHKSELR register. It can be accessed by $0C000000_H$ address offset of User Area or 04000000_H address offset of User Boot Area.

- Non-overlay Area for code flash memory

  This area is mirror of User Area without remapping. It can be accessed by 04000000_H address offset of User Area. In case that remap setting is enabled by CFU or GCFU, the data of corresponding non-secure User Area in Code Flash Memory cannot be read from bus masters via non-secure User Area. On the other hand, the data in Code Flash Memory can be read via this area without being influenced by CFU/GCFU setting.

- Blank Check Area for data flash memory

  This area is used for checking that Data Area or Hardware Property Area is blank or not. It can be accessed by 00200000_H address offset of Data Area or Hardware Property Area.

- OTA (Over-the-Air) update support

  Code Flash programming / erasure is possible while executing the user program that is stored in the User Area.

  Two types of Code Flash Memory Mapping Mode are supported (Single Map Mode and Double Map Mode).

  For details, see **Section 51.3.1, Mapping of Code Flash Memory**.

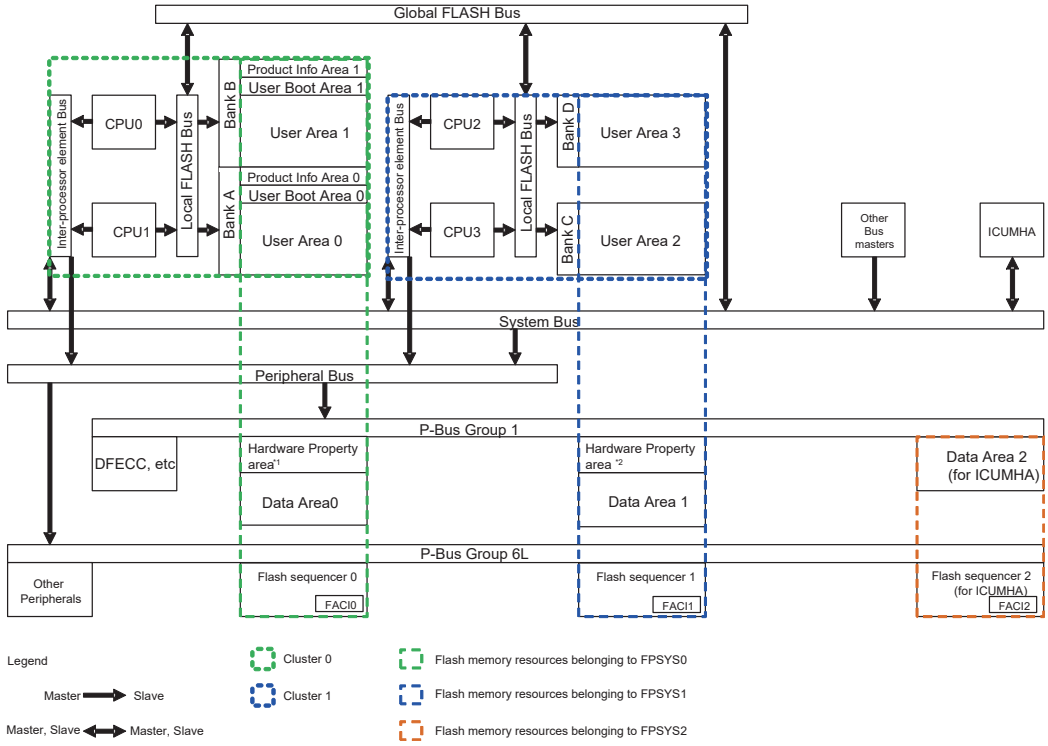
  Code Flash Memory Mapping Mode can be selected by Option Bytes setting.

  – Single Map Mode
  Hardware remapping of User Area read address is supported by GCFU (Overlay function).

  – Double Map Mode
  Hardware swapping of address map between Code Flash Memory Banks is supported.
  The Code Flash Memory Banks are mapped into two areas. One is valid area and the other is invalid area.
  The Banks in the valid area and invalid area can be swapped by Option Bytes setting.
  The user program can be executed from the banks in valid area while the banks of invalid area are being programmed / erased.

51.2.2 Block Diagram

Figure 51.1 shows block diagram of U2A-EVA/U2A16 Flash Programming System structure.

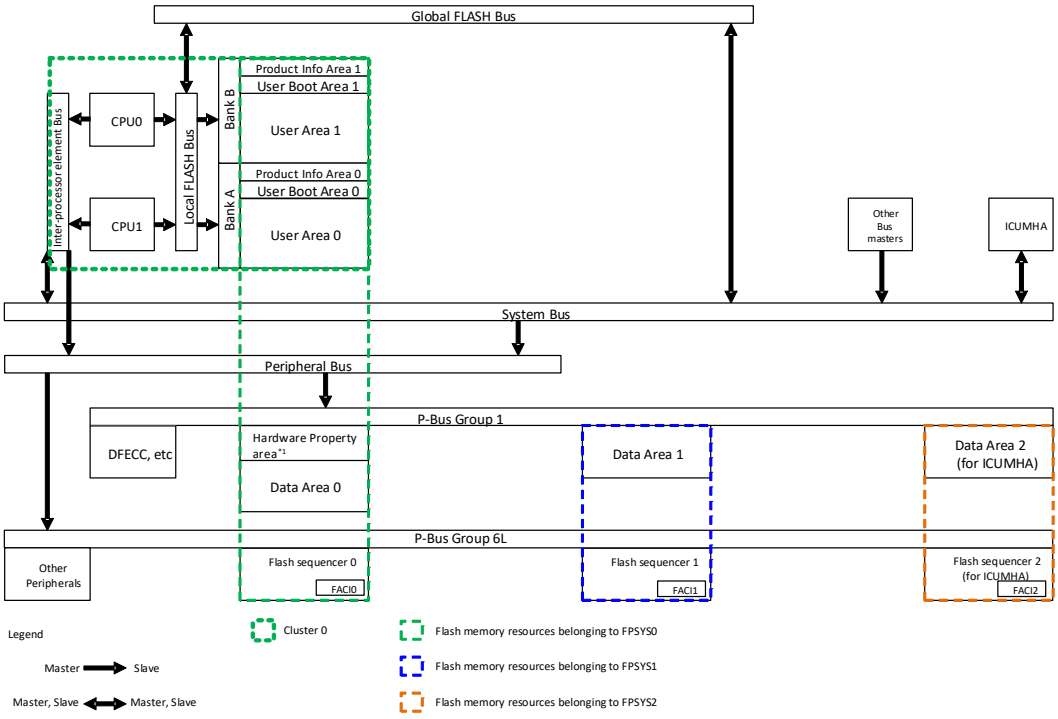


Note 1. Extended Data Area, Configuration Setting Area, Security Setting Area, Block Protection Area for FPSYS0, Switch Area, TAG Area

Note 2. Block Protection Area for FPSYS1.

Figure 51.1 U2A-EVA/U2A16 Flash Programming System structure

Figure 51.2 shows block diagram of U2A8/U2A6 Flash Programming System structure.



Note 1. Extended Data Area, Configuration Setting Area, Security Setting Area, Block Protection Area for FPSYS0, Switch Area, TAG Area

Figure 51.2 U2A8/U2A6 Flash Programming System structure

51.3 Structure of Memory

Table 51.5 gives information on all of these areas.

Table 51.5 Area on the Flash memory

Area	Address	Bus Group
User Area (code flash)	See **Section 51.3.1, Mapping of Code Flash Memory**	—
User Boot Area (code flash)		
Product Info Area (code flash)		
Data Area (data flash)	See **Section 51.3.2, Mapping of Data Area in Data Flash Memory**	P-Bus Group 1
Extended Data Area (data flash)	See **Section 51.3.3, Mapping of Hardware Property Area in Data Flash Memory**	P-Bus Group 1
Configuration Setting Area (data flash)		
Security Setting Area (data flash)		
Block Protection Area (data flash)		
Switch Area (data flash)		
TAG Area (data flash)		
Erase Counter Area (data flash)		

Table 51.6 shows the summary of supported area for each FPSYS.

Table 51.6 Summary of supported area for each FPSYS.

Area		Supported Area (√: Supported,×: Not Supported)					
		U2A-EVA (U2A16 mode) / U2A16			U2A-EVA (U2A8 mode) / U2A8 / U2A6		
		FPSYS0	FPSYS1	FPSYS2	FPSYS0	FPSYS1	FPSYS2
Code Flash	User Area 0 (Bank A), User Boot Area 0 (Bank A), Product Info Area 0 (Bank A)	√	×	×	√	×	×
	User Area 1 (Bank B), User Boot Area 1 (Bank B), Product Info Area 1 (Bank B)	√	×	×	√	×	×
	User Area 2 (Bank C)	×	√	×	×	×	×
	User Area 3 (Bank D)	×	√	×	×	×	×
Data Flash	Data Area 0	√	×	×	√	×	×
	Data Area 1	×	√	×	×	√	×
	Data Area 2	×	×	√	×	×	√
Data Flash (Hardware Property Area)	Extended Data Area	√	×	×	√	×	×
	Configuration Setting Area	√	×	×	√	×	×
	Security Setting Area	√	×	×	√	×	×
	Block Protection Area	√	√	×	√	×	×
	Switch Area	√	×	×	√	×	×
	TAG Area	√	×	×	√	×	×
	Erase Counter Area	√	√	×	√	×	×

51.3.1 Mapping of Code Flash Memory

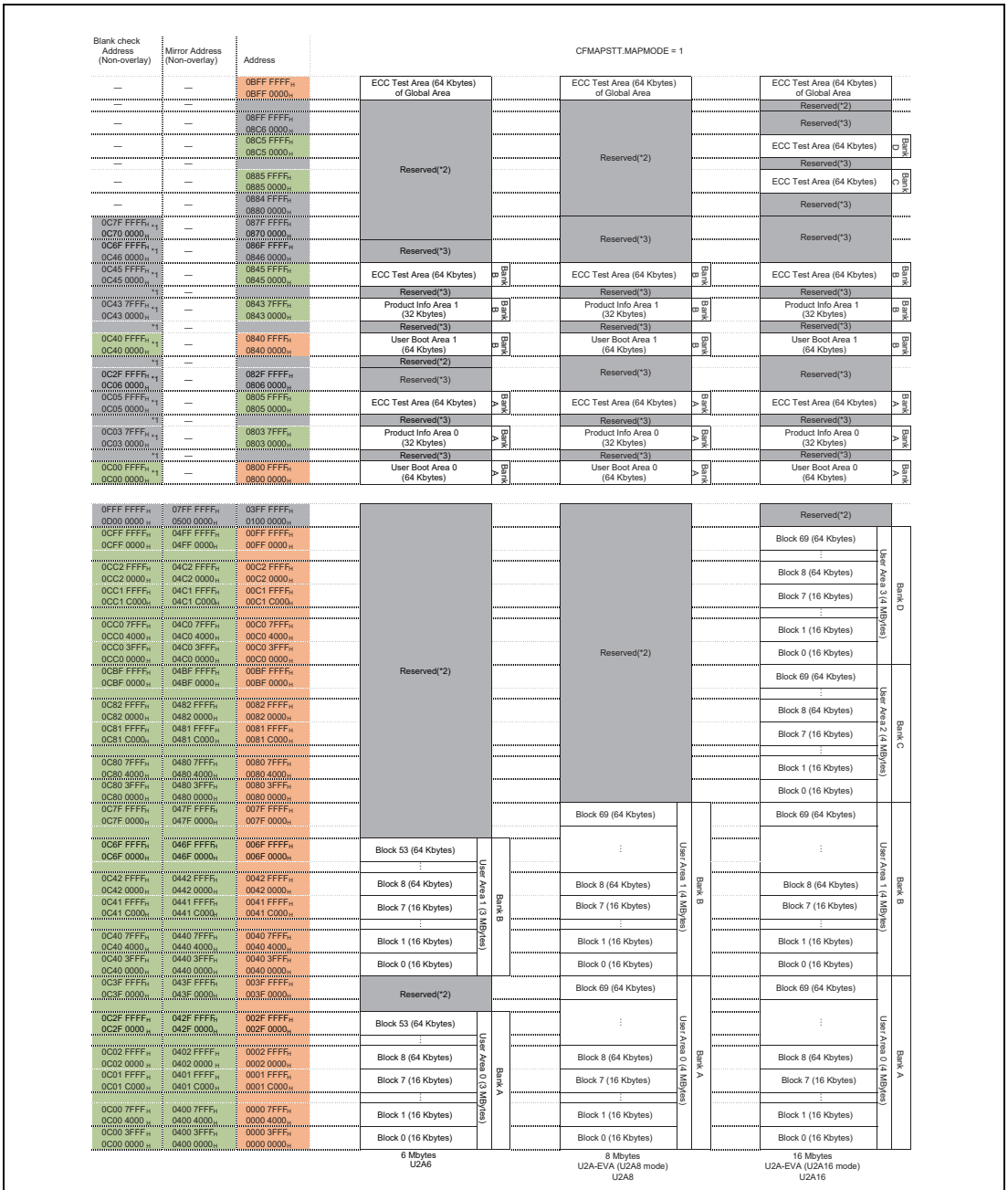
Figure 51.3 illustrates the mapping of the code flash memory for the 16-Mbyte (U2A-EVA (U2A16 mode), U2A16), 8-Mbyte (U2A-EVA (U2A8 mode), U2A8) and 6-Mbyte (U2A6) devices at the Single Map Mode.

Figure 51.4 and **Figure 51.5** illustrate the mapping of the code flash memory for the 16-Mbyte (U2A-EVA (U2A16 mode), U2A16) devices at the Double Map Mode.

Figure 51.6 illustrates the mapping of the code flash memory for the 8-Mbyte (U2A-EVA (U2A8 mode), U2A8) devices at the Double Map Mode.

Figure 51.7 illustrates the mapping of the code flash memory for the 6-Mbyte (U2A6) device at the Double Map Mode.

The User Area in the code flash memory of this product is divided into 16-Kbyte and 64-Kbyte blocks for each User Area, which can be erased individually. A single block of 64-Kbyte User Boot Area is also incorporated in Bank A and Bank B respectively. The User Area can be used to store the user program. The User Boot Area can be used to store a non-rewritable boot program during user program operation, such as a boot program for rewriting code flash memory through a selected user interface. The Product Info Area stores Product information (See **Section 51.11.1, Registers Related to Product Information**). This area is not rewritable. The BGO condition of Product Info Area is same as User Area /User Boot Area of same bank.

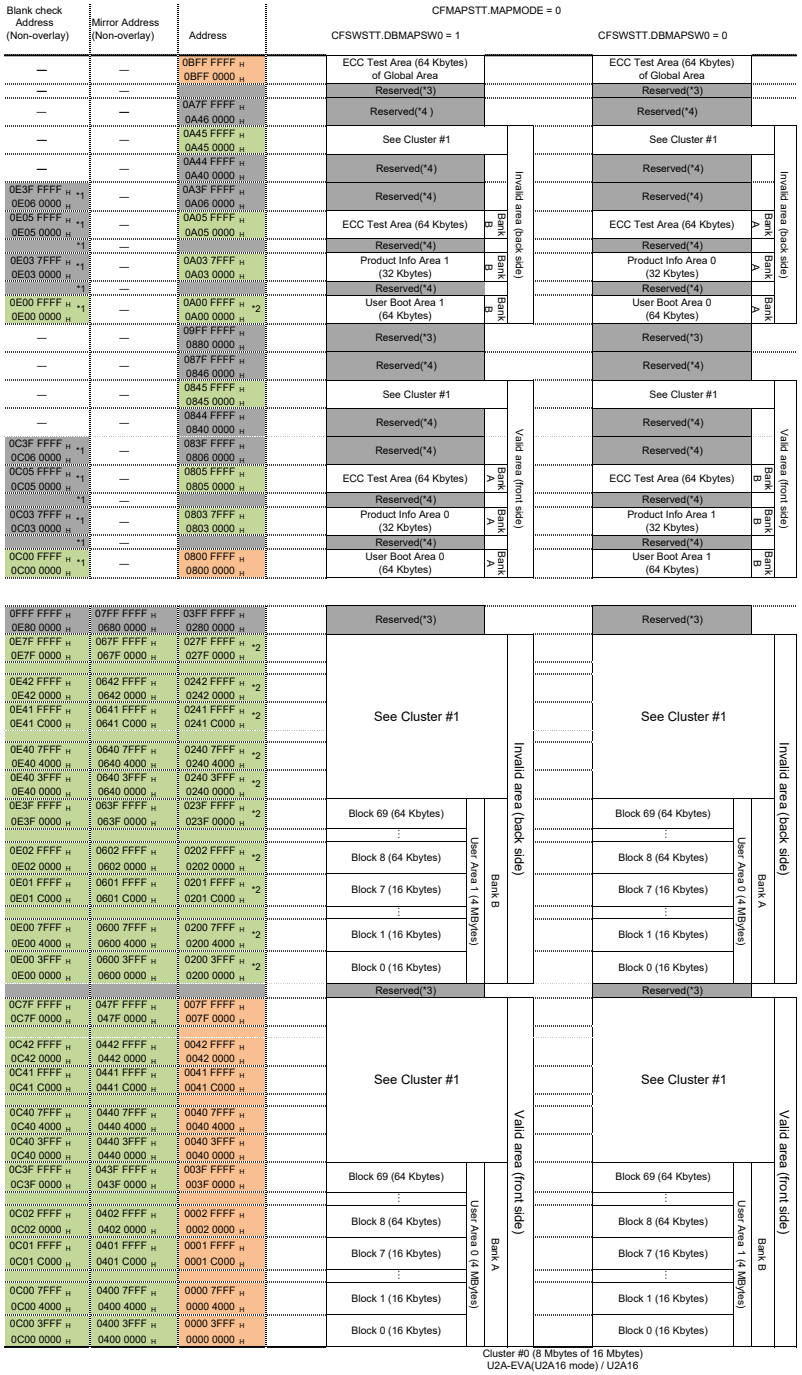


Note 1. When BLCHKSELR.BLNKSEL=1. For details, see **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.
Note 2. If this area is read, Error response occurs.
Note 3. If this area is read, Error response, ECC 2-bit error occur. Address parity error also occurs depending on the address.

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

Figure 51.3 Code Flash mapping of Single Map Mode

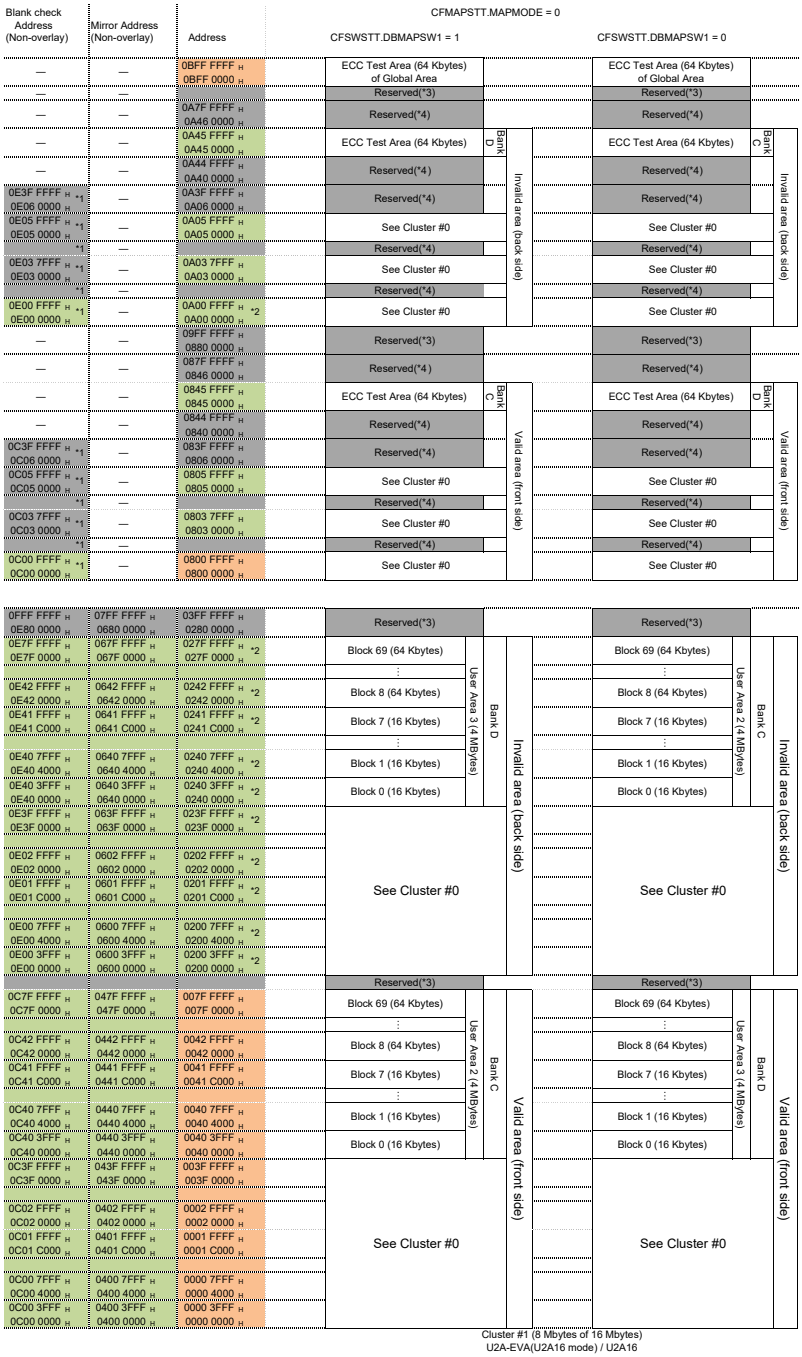


Note 1. When BLCHKSELR.BLNKSEL=1 for details refer to **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.
Note 2. Instruction fetch to invalid area should be prohibited by MPU setting.
Note 3. If this area is read, Error response occurs.
Note 4. If this area is read, Error response, ECC 2-bit error occur. Address parity error also occurs depending on the address.

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

Figure 51.4 Code Flash mapping of Double Map Mode U2A-EVA (U2A16 mode) / U2A16 Cluster #0

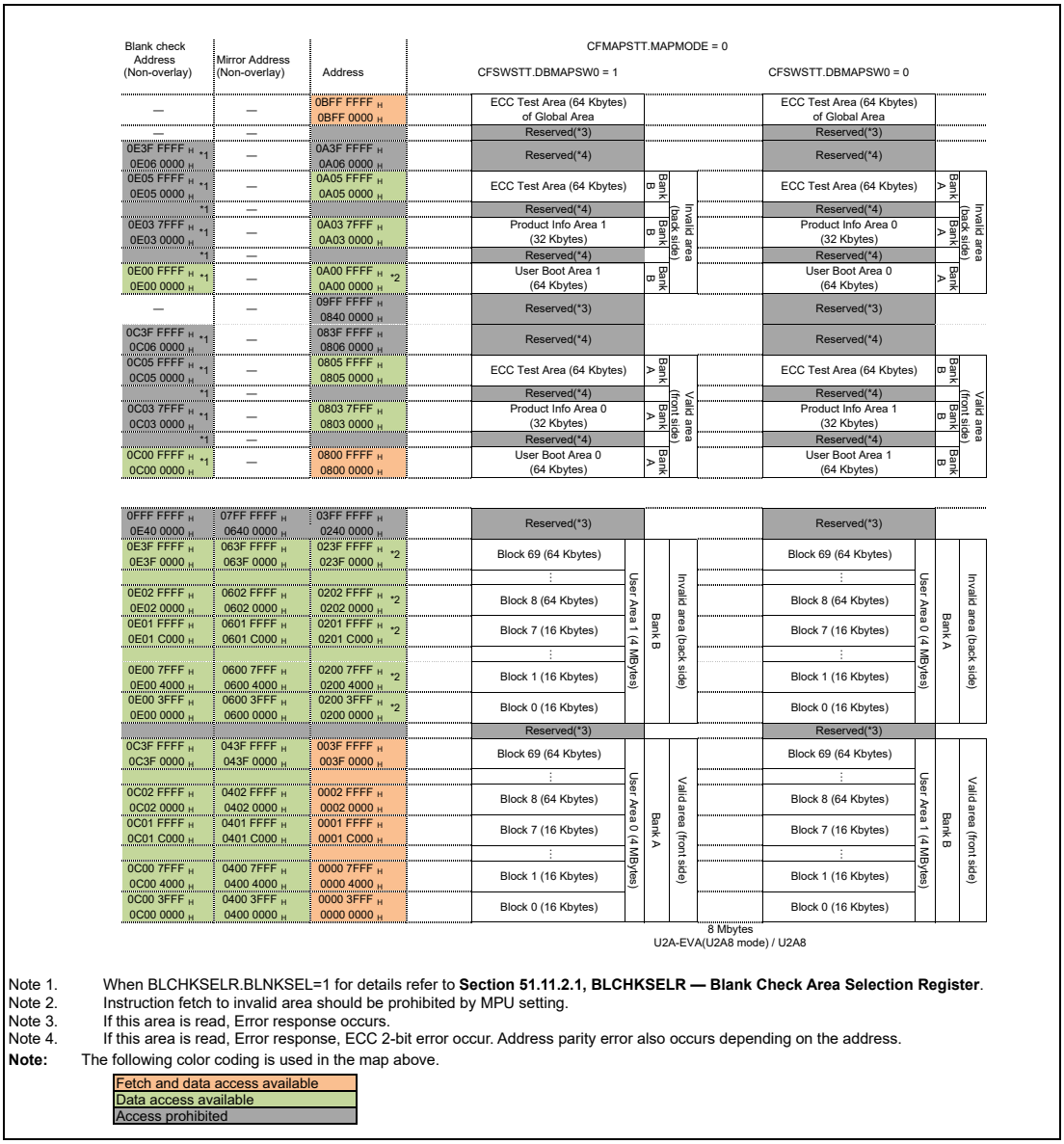


Note 1. When BLCHKSELR.BLNKSEL=1 for details refer to **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.
Note 2. Instruction fetch to invalid area should be prohibited by MPU setting.
Note 3. If this area is read, Error response occurs.
Note 4. If this area is read, Error response, ECC 2-bit error occur. Address parity error also occurs depending on the address.

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

Figure 51.5 Code Flash mapping of Double Map Mode U2A-EVA (U2A16 mode) / U2A16 Cluster #1

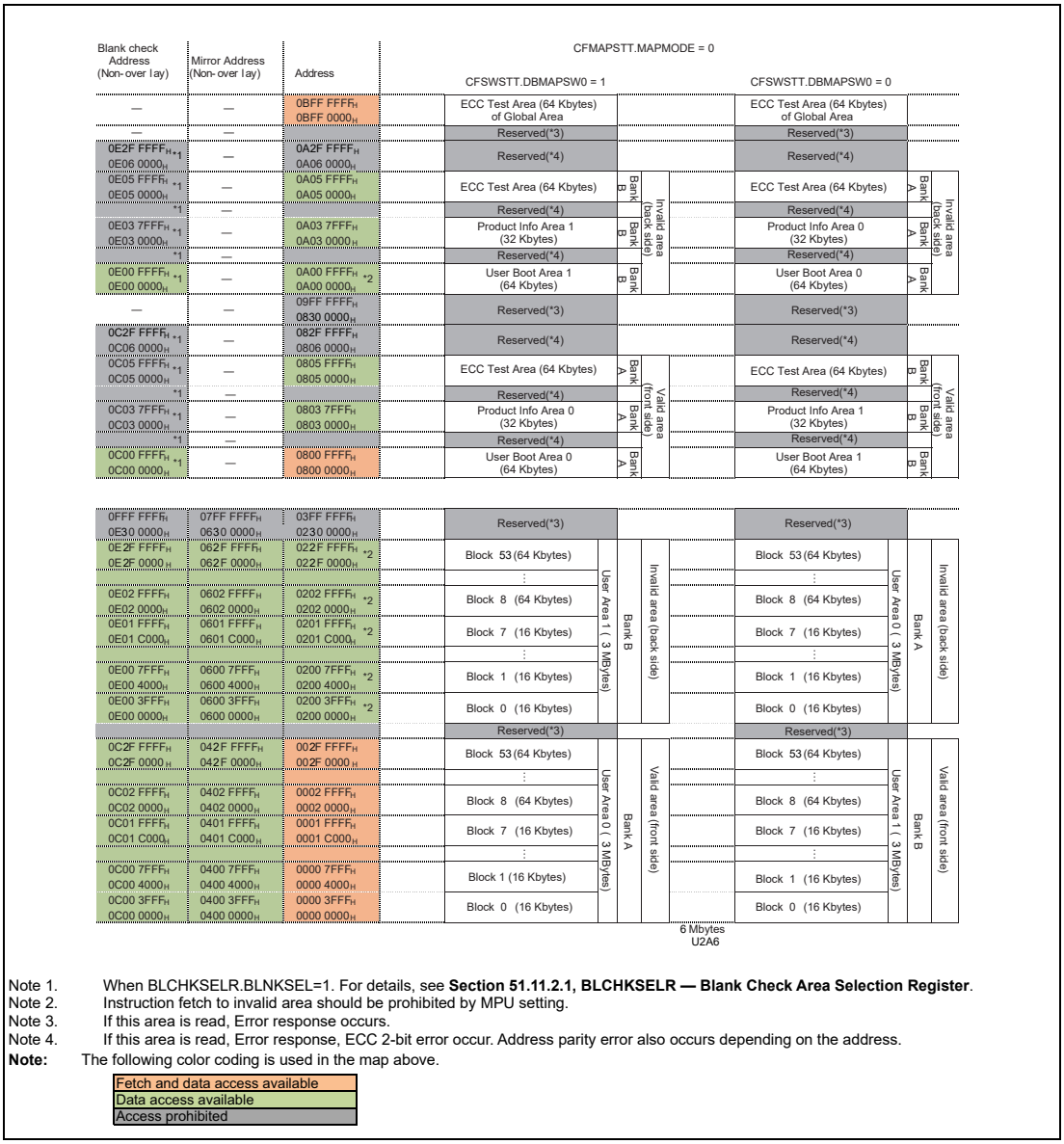


Note 1. When BLCHKSELR.BLNKSEL=1 for details refer to **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.
Note 2. Instruction fetch to invalid area should be prohibited by MPU setting.
Note 3. If this area is read, Error response occurs.
Note 4. If this area is read, Error response, ECC 2-bit error occur. Address parity error also occurs depending on the address.

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

Figure 51.6 Code Flash mapping of Double Map Mode U2A-EVA (U2A8 mode) / U2A8



Note 1. When BLCHKSELR.BLNKSEL=1. For details, see **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.
Note 2. Instruction fetch to invalid area should be prohibited by MPU setting.
Note 3. If this area is read, Error response occurs.
Note 4. If this area is read, Error response, ECC 2-bit error occur. Address parity error also occurs depending on the address.

Note: The following color coding is used in the map above.

Fetch and data access available
Data access available
Access prohibited

Figure 51.7 Code Flash mapping of Double Map Mode U2A6

Table 51.7, **Table 51.8** and **Table 51.9** show Base address of Product Info Area in each Code Flash Memory Map Mode.

Table 51.7 Code Flash Memory Map Single Map Mode (CFMAPSTT.MAPMODE=1)

Base Address Name <PRDINFk_base> (k = f, b)	Base Address	Bus Group
<PRDINFf_base>	$0803\ 0000_H$ (Product Info Area 0, Bank A)	—
<PRDINFb_base>	$0843\ 0000_H$ (Product Info Area 1, Bank B)	—

f … front side. b… back side.

Table 51.8 Code Flash Memory Map Double Map Mode, Bank A is valid (CFMAPSTT.MAPMODE=0, CFSWSTT.DBMAPSW0=1)

Base Address Name <PRDINFk_base> (k = f, b)	Base Address	Bus Group
<PRDINFf_base>	$0803\ 0000_H$ (Product Info Area 0, Bank A)	—
<PRDINFb_base>	$0A03\ 0000_H$ (Product Info Area 1, Bank B)	—

f … front side (valid). b… back side (invalid).

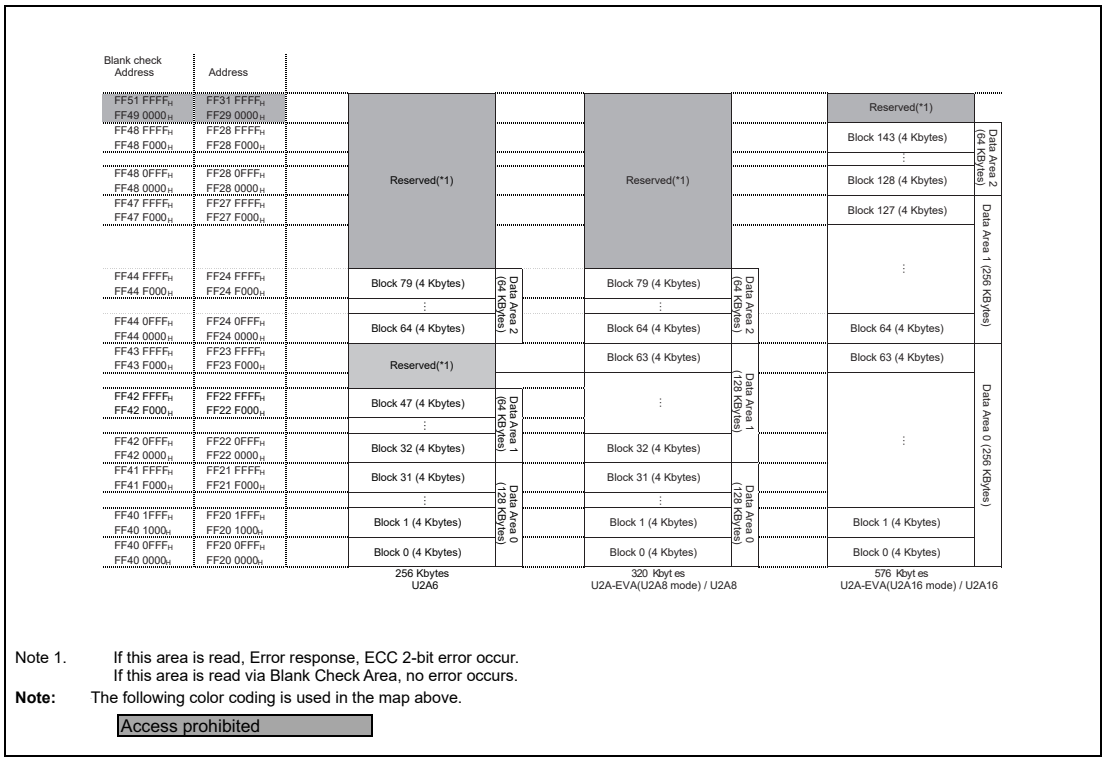
Table 51.9 Code Flash Memory Map Double Map Mode, Bank B is valid (CFMAPSTT.MAPMODE=0, CFSWSTT.DBMAPSW0=0)

Base Address Name <PRDINFk_base> (k = f, b)	Base Address	Bus Group
<PRDINFf_base>	$0803\ 0000_H$ (Product Info Area 1, Bank B)	—
<PRDINFb_base>	$0A03\ 0000_H$ (Product Info Area 0, Bank A)	—

f … front side (valid). b… back side (invalid)

51.3.2 Mapping of Data Area in Data Flash Memory

The Data Area of the data flash memory in this product is divided into 4-Kbyte blocks, with each being a unit for erasure. **Figure 51.8** shows data flash memory map.



Note 1. If this area is read, Error response, ECC 2-bit error occur.
If this area is read via Blank Check Area, no error occurs.

Note: The following color coding is used in the map above.

Access prohibited

Figure 51.8 Mapping of Data Area

51.3.3 Mapping of Hardware Property Area in Data Flash Memory

Hardware Property Area of this product consists of seven areas (Extended Data Area, Configuration Setting Area, Security Setting Area, Block Protection Area, Switch Area, TAG Area and Erase Counter Area).

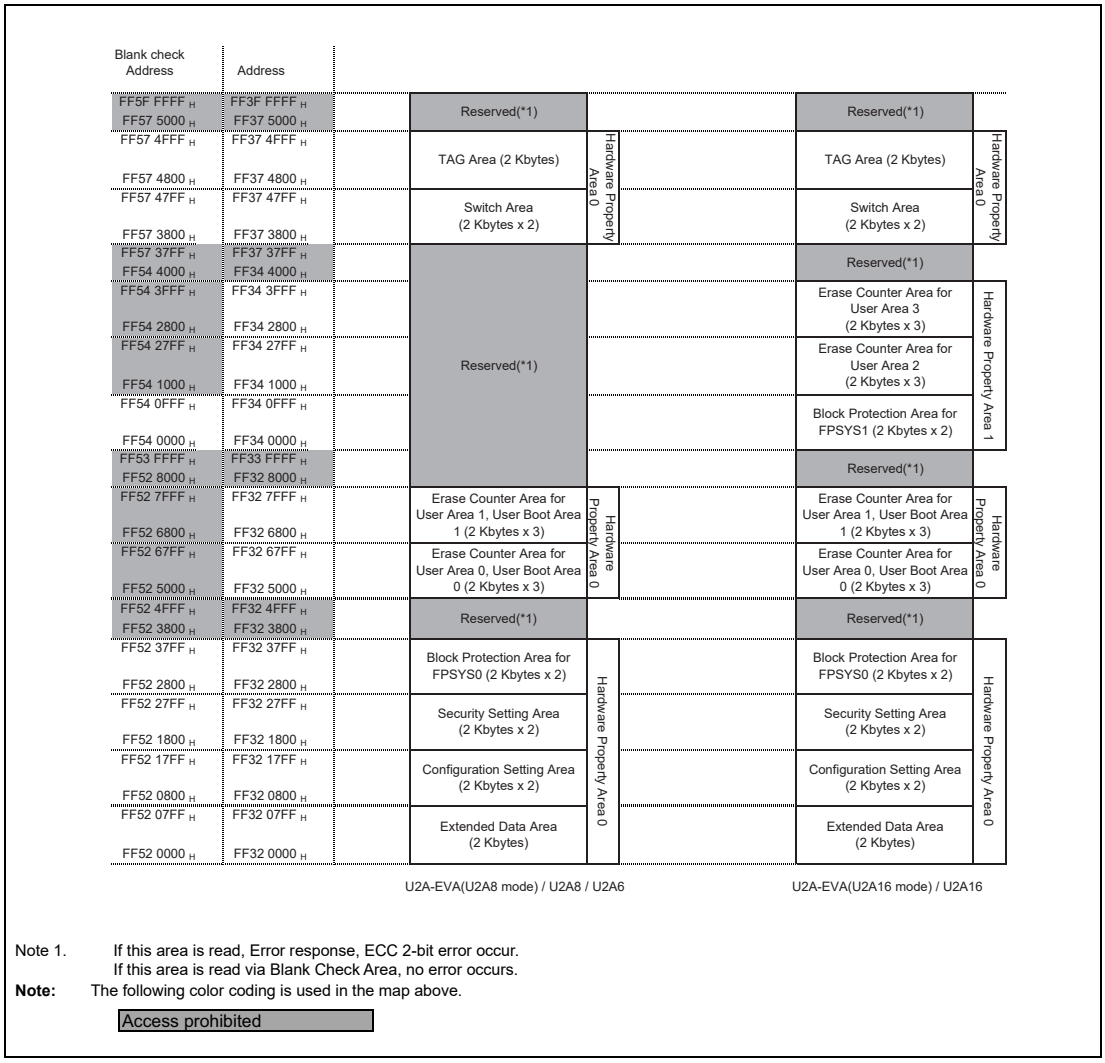
- For Security Setting Area, see **Section 47.8, Security Setting Area**.
- For Extended Data Area, see **Section 51.16, Extended Data Area**.
- For Configuration Setting Area, see **Section 51.12, Configuration Setting Area (Option Bytes, Reset Vector)**.
- For Block Protection Area, see **Section 47.9, Block Protection Area for FPSYS0/ FPSYS1**.
- For Switch Area, see **Section 51.13, Switch Area**.
- For TAG Area, see **Section 51.14, TAG Area**.
- For Erase Counter Area, see **Section 51.15, Erase Counter Area**.

Dedicated programming/erasure command is supported for Configuration Setting Area, Security Setting Area, Block Protection Area, Switch Area and TAG Area.

Configuration Setting Area and Security Setting Area can be set OTP (One Time Programming) by 4 bytes unit.

Extended Data Area can be set OTP (One Time Programming).

Table 51.8 shows Hardware Property Area memory map. This area is placed in the data flash memory. The condition of BGO, see **Table 51.27, Conditions under which Background Operation is Usable**.



Note 1. If this area is read, Error response, ECC 2-bit error occur.
If this area is read via Blank Check Area, no error occurs.

Note: The following color coding is used in the map above.

Access prohibited

Figure 51.9 Mapping of Hardware Property Area

Table 51.10 and **Table 51.11** show Base address of Configuration Setting Area.

Table 51.10 Base address of Configuration Setting Area in case of Area 0 is valid (FSWASTAT_0.CFGVA=0)

Base Address Name <CSAk_base> (k = f, b)	Base Address	Bus Group
<CSAf_base>	FF32 0800$_H$ (Configuration Setting Area 0)	P-Bus Group 1
<CSAb_base>	FF32 1000$_H$ (Configuration Setting Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.11 Base address of Configuration Setting Area in case of Area 1 is valid (FSWASTAT_0.CFGVA=1)

Base Address Name <CSAk_base> (k = f, b)	Base Address	Bus Group
<CSAf_base>	FF32 0800$_H$ (Configuration Setting Area 1)	P-Bus Group 1
<CSAb_base>	FF32 1000$_H$ (Configuration Setting Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.12 and **Table 51.13** show Base address of Security Setting Area.

Table 51.12 Base address of Security Setting Area in case of Area 0 is valid (FSWASTAT_0.SECVA=0)

Base Address Name <SSAk_base> (k = f, b)	Base Address	Bus Group
<SSAf_base>	FF32 1800$_H$ (Security Setting Area 0)	P-Bus Group 1
<SSAb_base>	FF32 2000$_H$ (Security Setting Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.13 Base address of Security Setting Area in case of Area1 is valid (FSWASTAT_0.SECVA=1)

Base Address Name <SSAk_base> (k = f, b)	Base Address	Bus Group
<SSAf_base>	FF32 1800$_H$ (Security Setting Area 1)	P-Bus Group 1
<SSAb_base>	FF32 2000$_H$ (Security Setting Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.14 and **Table 51.15** show Base address of Block Protection Area for FPSYS0.

Table 51.14 Base address of Block Protection Area for FPSYS0 (Area 0 is valid, FSWASTAT_0.BPVA0=0)

Base Address Name <BPA0k_base> (k = f, b)	Base Address	Bus Group
<BPA0f_base>	FF32 2800$_H$ (Block Protection Area for FPSYS0 Area 0)	P-Bus Group 1
<BPA0b_base>	FF32 3000$_H$ (Block Protection Area for FPSYS0 Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.15 Base address of Block Protection Area for FPSYS0 (Area 1 is valid, FSWASTAT_0.BPVA0=1)

Base Address Name <BPA0k_base> (k = f, b)	Base Address	Bus Group
<BPA0f_base>	FF32 2800_H (Block Protection Area for FPSYS0 Area 1)	P-Bus Group 1
<BPA0b_base>	FF32 3000_H (Block Protection Area for FPSYS0 Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.16 and **Table 51.17** show Base address of Block Protection Area for FPSYS1.

Table 51.16 Base address of Block Protection Area for FPSYS1 (Area 0 is valid, FSWASTAT_0.BPVA1=0)

Base Address Name <BPA1k_base> (k = f, b)	Base Address	Bus Group
<BPA1f_base>	FF34 0000_H (Block Protection area for FPSYS1 Area 0)	P-Bus Group 1
<BPA1b_base>	FF34 0800_H (Block Protection area for FPSYS1 Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.17 Base address of Block Protection Area for FPSYS1 (Area 1 is valid, FSWASTAT_0.BPVA1=1)

Base Address Name <BPA1k_base> (k = f, b)	Base Address	Bus Group
<BPA1f_base>	FF34 0000_H (Block Protection area for FPSYS1 Area 1)	P-Bus Group 1
<BPA1b_base>	FF34 0800_H (Block Protection area for FPSYS1 Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.18 and **Table 51.19** show Base address of Switch Area.

Table 51.18 Base address of Switch Area (Area 0 is valid, FSWASTAT_0.SWVA=0)

Base Address Name <SWAk_base> (k = f, b)	Base Address	Bus Group
<SWAf_base>	FF37 3800_H (Switch Area 0)	P-Bus Group 1
<SWAb_base>	FF37 4000_H (Switch Area 1)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.19 Base address of Switch Area (Area 1 is valid, FSWASTAT_0.SWVA=1)

Base Address Name <SWAk_base> (k = f, b)	Base Address	Bus Group
<SWAf_base>	FF37 3800_H (Switch Area 1)	P-Bus Group 1
<SWAb_base>	FF37 4000_H (Switch Area 0)	P-Bus Group 1

f … front side (valid). b… back side (invalid).

Table 51.20 shows Base address of TAG Area.

Table 51.20 Base address of TAG Area

Base Address Name	Base Address	Bus Group
<TAG_base>	FF37 4800_H	P-Bus Group 1

Table 51.21 shows Base address of Erase Counter Area.

Table 51.21 Base address of Erase Counter Area.

Base Address Name	Base Address	Bus Group
<EC0_base>	FF32 5000_H	P-Bus Group 1
<EC1_base>	FF32 6800_H	P-Bus Group 1
<EC2_base>	FF34 1000_H	P-Bus Group 1
<EC3_base>	FF34 2800_H	P-Bus Group 1

51.4 Operating Modes Associated with Flash Memory

Figure 51.10 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 5, Operating Modes**.

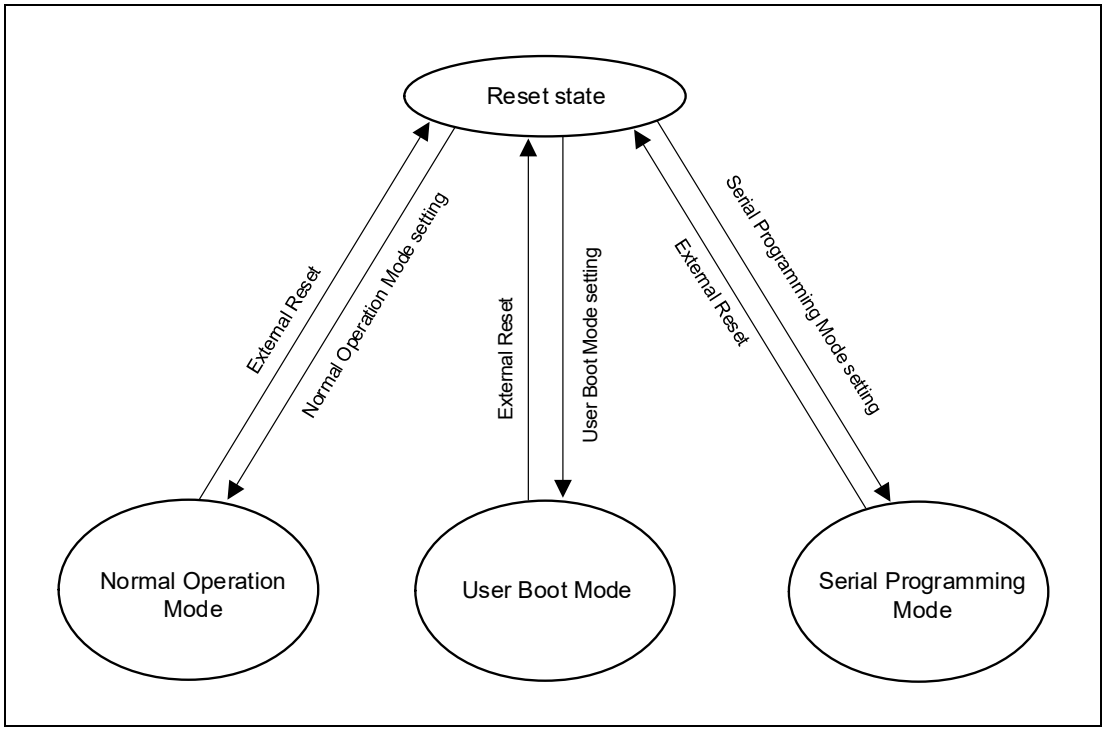


Figure 51.10 Mode Transition Associated with Flash Memory

Table 51.22 shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 51.22 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release

Item	Normal Operation Mode	User Boot Mode	Serial Programming Mode
Programmable and erasable area	• User Area • Data Area • Hardware Property Area	• User Area • Data Area • Hardware Property Area	• User Area • User Boot Area • Data Area • Hardware Property Area
Boot program after reset release	Program in User Area (Changeable by using the variable reset vector)	Program in User Boot Area (CPU0) Program in User Area (Changeable by using the variable reset vector) (CPU1-3)	Boot Firmware program for serial programming

51.5 Programming Overview

The flash memory of this device can be updated via a serial interface by a dedicated flash memory programmer (serial programming), before being mounted on the target system or on a flash adapter system.

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as external communications, etc., and this allows programming under various conditions.

Table 51.23 gives an overview of the methods of programming and the corresponding operating modes.

Table 51.23 Methods of Programming

Method of Programming	Overview of Functionality	Operating Mode
Serial programming	A dedicated flash memory programmer allows on-board programming the flash memory after the device is mounted on the target system.	Serial Programming Mode
	A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	The user program that is written to code flash memory in advance by serial programming executing also allows updating the flash memory. When data flash memory is being programmed with self-programming, the BGO function enables instruction fetch and data read from code flash memory. Thus, data flash memory can be programmed by executing a program on code flash memory prepared for flash programming. When code flash memory bank is being programmed with self-programming, instruction fetch and data read from the bank are prohibited. This programming needs to be carried out by executing a program prepared for flash programming that has been transferred to the Cluster RAM or the Local RAM in advance or a program on a different bank.	Normal Operation Mode, User Boot Mode

When executing self-programming, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 51.24 lists the functions of the flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the flash memory by interface operation of flash memory or the user program enables self-programming.

Table 51.24 Basic Functions at a Glance

Function	Description in Overview	Level of Support (√: Supported,×: Not Supported)	
		Serial Programming	Self-Programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded.	√	√
Erasure	This is for erasing the contents of specified block of memory.	√	√
Programming	This is for writing to a specified address.	√	√
Reading	Data that have been written to the flash memory are read out.	√	√
Setting of Security Settings (Setting an ID Codes)	Security Settings (ID codes, Debugger connection prohibited setting, etc.) are set and the security is made effective. An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the On-chip debug, and etc. See **Section 47.8, Security Setting Area** for details.	√	√
Setting of Block Protection	A specified block of code flash memory is set for Customer ID Protection Setting and OTP (one-time programming). (OTP can only be set, that is, it is not possible to release a block's OTP setting). See **Section 47.9, Block Protection Area for FPSYS0/FPSYS1** for details.	√	√
Setting of Configuration Settings	Configuration Settings (Option bytes, Reset Vector, etc.) are set to change them from the initial values for this product.	√	√

For details on serial programming, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

For details on self-programming, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

The flash memory supports various security functions.

Setting of OTP and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, and prohibiting connection of a dedicated flash memory programmer are available for use as security functions.

Table 51.25 lists security functions that are supported by the flash memory. Security Settings is established in the Security Setting Area. For Security Settings, see **Section 47.8, Security Setting Area**.

Table 51.25 Summary of Security Functions

Function	Description
OTP	OTP can be individually set for each block of the User Area and the User Boot Area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self-programming is prohibited. Once set, the OTP setting cannot be released.
ID authentication	The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of code flash memory and data flash memory writing by self-programming.
Prohibition of connection of a dedicated flash memory programmer*1	The connection of a dedicated flash memory programmer for serial programming is prohibited.

Note 1. The “Prohibition of connection of a dedicated flash memory programmer” function can be used in conjunction with “ID authentication”.

The flash memory supports various protection functions. **Table 51.26** lists protection functions that are supported by the flash memory.

Table 51.26 Summary of Protection Functions

Function	Description
User boot protection	Programming or erasure of the User Boot Area by self-programming is prohibited. Programming or erasure of the User Boot Area by serial programming is available.
Variable reset vector	The protection settings include control of the reset vector. As shown in **Figure 51.11**, after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the User Area.
Code Flash Valid area protection	Code Flash Valid area protection is function for front side(valid) of User Area in Double Map Mode. The front side(valid) of User Area can be protected from programming, block erasure, and program/erasure resumption command. For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.
Flag protection for Switch Area and TAG Area	Switch Area cannot be changed when VAF in TAG Area is not correct. TAG Area cannot be changed when CVA/SVA/BVA0/BVA1 in Switch Area or VOFs in Configuration Setting Area/Security Setting Area/Block Protection Area are not correct. For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

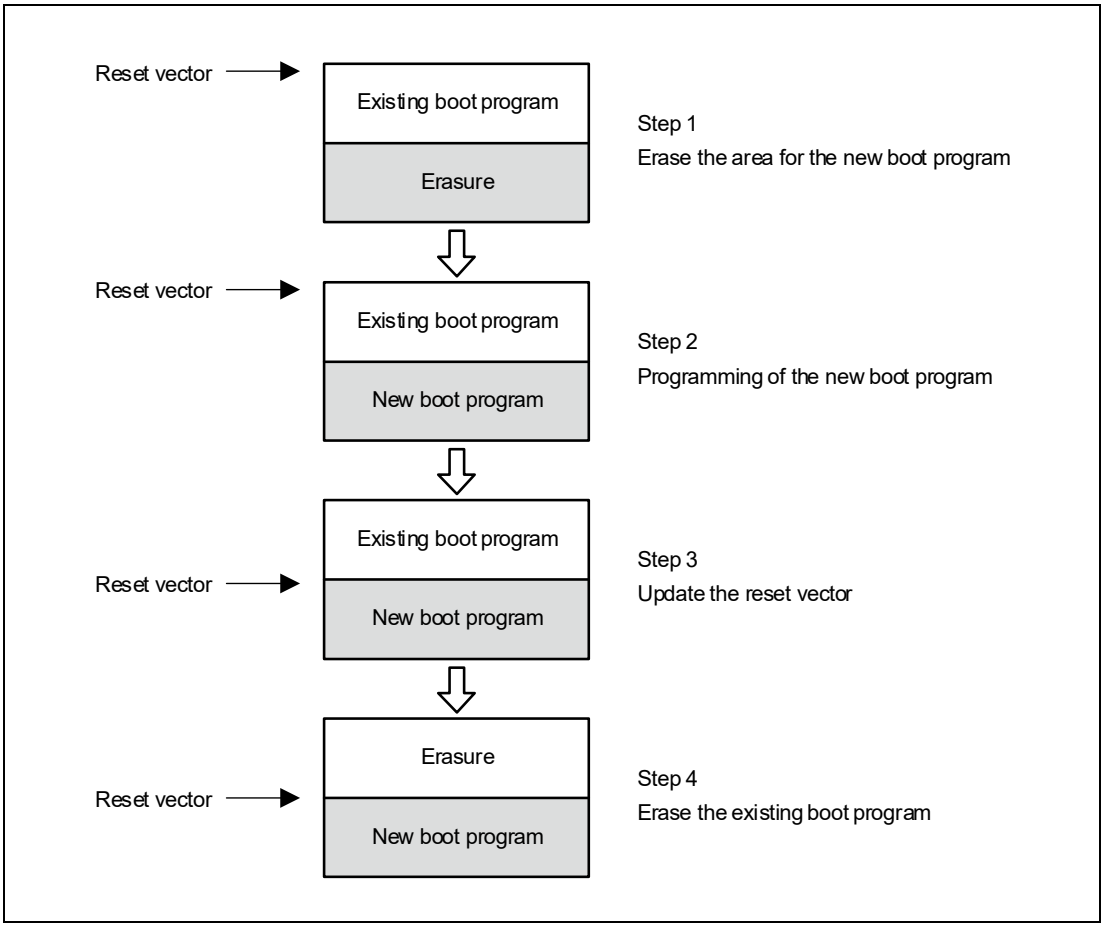


Figure 51.11 Utilizing the Variable Reset Vector Function to Update the Boot Program

51.6 Serial Programming

A dedicated flash memory programmer can be used to handle the flash memory in Serial Programming Mode.

Serial Programming

The microcontroller is mounted on the system board at the time of serial programming. Providing a connector to the board enables handling of the target microcontroller by the flash memory programmer to proceed.

NOTE

For details on Renesas Flash Programmer, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

51.7 Flash Programming Interface

51.7.1 Asynchronous Flash Programming Interface — 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output

51.7.2 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

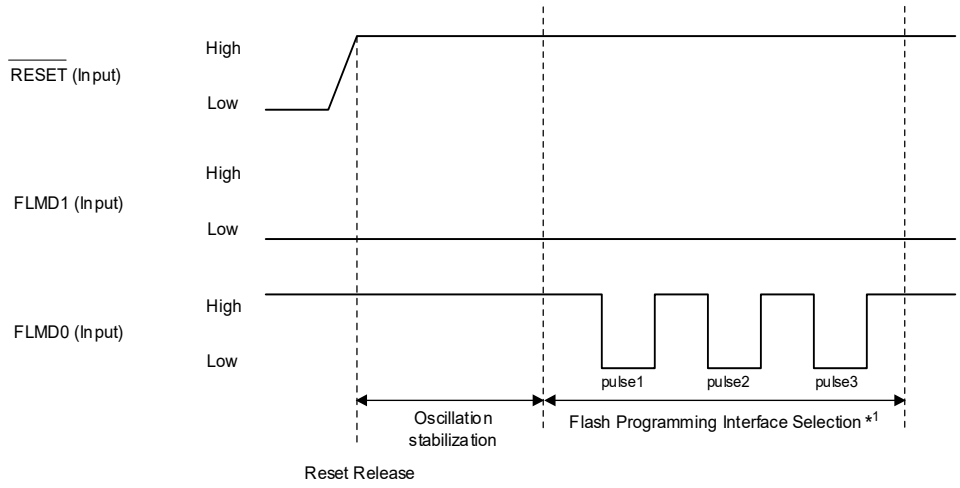
- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output
- FPCK(JP0_2): Serial clock input

The flash memory programmer outputs the serial data clock FPCK, and the microcontroller operates as a slave.

51.7.3 Selection of Flash Programming Interface

In this product, Flash Programming Interface can be selected by pulse input to the FLMD0 pin (up to 3 pulses) after transition to the Serial Programming Mode. The FLMD0 pulse is generated by a dedicated flash memory programmer.

Figure 51.12 shows the relation between the number of pulses and Flash Programming Interface.



Note 1. The number of FLMD0 Pulses to be inserted depends on Flash Programming Interface as in the following table.

FLMD0 Pulse	Flash Programming Interface
0	CSI
3	2-wire UART
Other than above	Setting prohibited

Figure 51.12 Selection of Flash Programming Interface

For details of AC Characteristics, see the **Section 55, Electrical Characteristics**.

51.8 Self-Programming

51.8.1 Outline

This product supports programming of the flash memory by user program itself. The code flash and data flash memory can be programmed by using the commands of Flash Application Command Interface (FACI) for flash memory programming in user's applications. Therefore, update of the user program and programming of constant data fields can be possible.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to the Local RAM or the Cluster RAM in advance and executed to program the data flash memory.

When the code flash memory is programmed, the background operation facility makes it possible to execute a programming program form the code flash memory of a bank to program the code flash memory of the other bank. Furthermore, the programming program can be copied to the Local RAM or the Cluster RAM in advance and executed to program the code flash memory.

For comprehensive information on flash self-programming, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

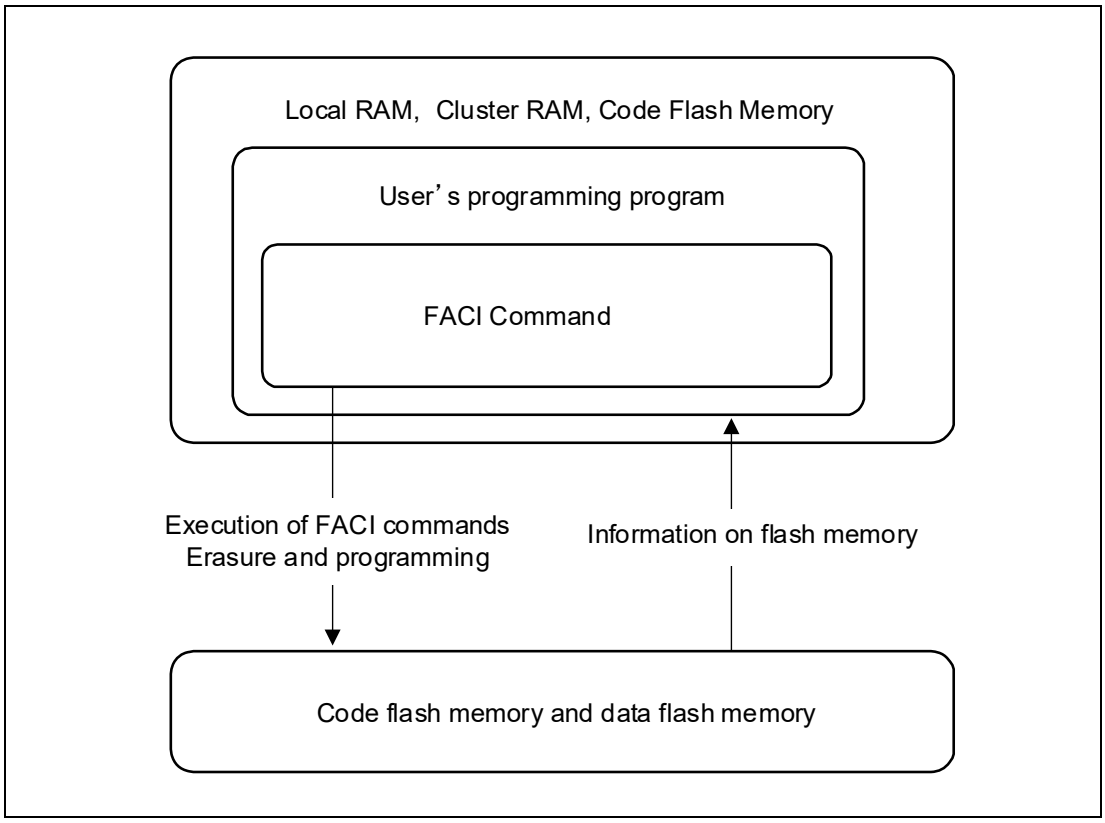


Figure 51.13 Concept of Self-Programming

51.8.2 Background Operation

Background operations can be used when the combination of the flash memory for programming/erasure and the flash memory for reading within one FPSYS is any of those listed below.

Error response occurs when the area could not be read due to the conditions under which Background Operation is not usable. On instruction fetch, SYSERR also occurs. On data flash read access, ECC 2-bit error also occurs.

Note that no error occurs when Blank Check Area for data flash memory could not be read due to the conditions under which Background Operation is not usable.

Table 51.27 Conditions under which Background Operation is Usable

			Code Flash				Data Flash			
			Bank N*1		Bank M*1		Data Area n*1		Hardware Property Area n*1	
			R	P/E	R	P/E	R	P/E	R	P/E
Code Flash	Bank N*1	R	—	—	Yes	Yes	Yes	Yes	Yes	Yes
		P/E	—	—	Yes	—	Yes	—	Yes*2	—
	Bank M*1	R	Yes	Yes	—	—	Yes	Yes	Yes	Yes
		P/E	Yes	—	—	—	Yes	—	Yes*2	—
Data Flash	Data Area n*1	R	Yes	Yes	Yes	Yes	—	—	Yes	Yes
		P/E	Yes	—	Yes	—	—	—	Yes	—
	Hardware Property Area n*1	R	Yes	Yes*2	Yes	Yes*2	Yes	Yes	—	—
		P/E	Yes	—	Yes	—	Yes	—	—	—

Note 1. Available Code Flash Bank/Data Area/Hardware Property Area is dependent on each FPSYS.
(N, M, n) = (A, B, 0) for FPSYS0.
(N, M, n) = (C, D, 1) for FPSYS1.
(N, M, n) = (N/A, N/A, 2) for FPSYS2. Hardware Property Area is not available for FPSYS2.

Note 2. Hardware Property Area is not readable while Code Flash block which is configured to enable Erase Counter is being erased.

Legend
R: Reading
P/E: Programming/Erasure

51.8.3 Multi FPSYS Operation

Multi operations can be used when the combination of the flash memory for programming/erasure and the flash memory for programming/erasure is any of those listed below.

Table 51.28 Conditions under which Multi Operation is Usable

		FPSYSn		FPSYSm (m != n)	
		R	P/E	R	P/E
FPSYSn	R	See **Section 51.8.2, Background Operation**		Yes	Yes
	P/E			Yes	Yes*1
FPSYSm (m != n)	R	Yes	Yes	See **Section 51.8.2, Background Operation**	
	P/E	Yes	Yes*1		

Note 1. Simultaneous Programming/Erasure of Hardware Property Area 0 and 1 is prohibited.
Simultaneous Programming/Erasure of multiple FPSYS should be operated so as not to deviate operational condition.

51.8.4 Switching of Hardware Property Area

Switch and TAG Area are used to switch Configuration Settings, Security Settings, and Block Protection data in an atomic and robust way.

To improve robustness against unintentional interruption during update of Hardware Property Area and make fallback procedure of Hardware Property Area easily, below functions are supported.

1. Configuration Setting Area, Security Setting Area, and Block Protection Area are duplicated. One is valid area, the other is invalid area.
2. Valid Area Flags which indicate valid area of the above duplicated area are stored in Switch and TAG Area.
3. The flags which indicate progress of programming / erasure of each data.

Figure 51.14 shows concept of Configuration Setting, Security Setting, Block Protection, Switch, and TAG Area structure. Note that U2A-EVA (U2A16 mode) and U2A16 have Block Protection Area for FPSYS0 and FPSYS1 respectively.

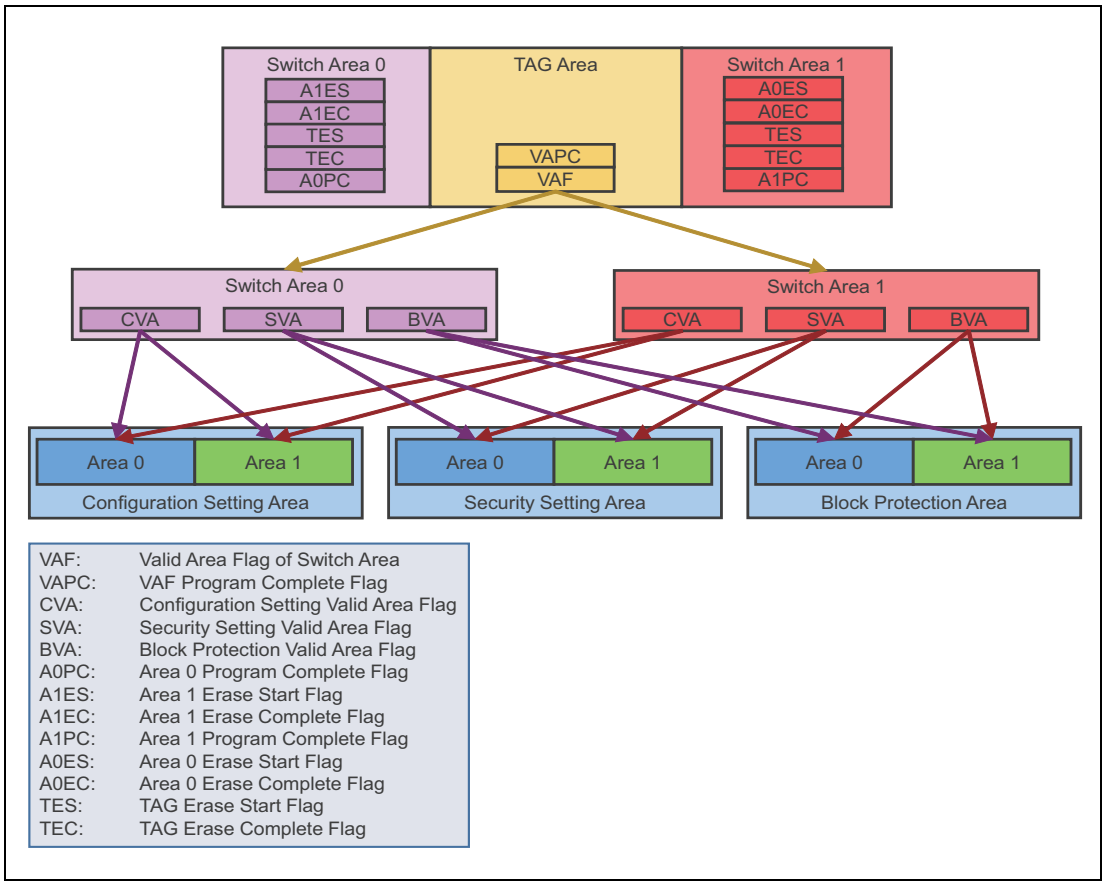


Figure 51.14 Concept of Configuration Setting, Security Setting, Block Protection, Switch, and TAG Area structure

Valid area of Switch Area is selected by VAF in TAG Area.

Valid area of Configuration Setting Area, Security Setting Area, and Block Protection Area are selected by CVA, SVA, BVA in valid area of Switch Area respectively.

The other flags are prepared to indicate progress of programming / erasure of each data in order to improve robustness against unintentional interruption during programming / erasure of each data (Note that the complete robustness is not guaranteed).

51.8.4.1 How to update and switch Configuration Setting Area, Security Setting Area, Block Protection Area

Update invalid area of Configuration Setting Area, Security Setting Area, or Block Protection Area beforehand if it is necessary. Then switch the valid area of those area.

VAPC in TAG Area may be undefined state if update of TAG Area is interrupted.

Unintentional fallback may be caused by FACI reset transfer if this state is left uncontrolled.

To avoid such unintentional fallback, run the flow of Recovering Switch Area Status.

For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

51.8.5 Example of User Program Update Procedure in the Field

The figure below is an example of user program update procedure in the field using hardware swapping in Double Map Mode.

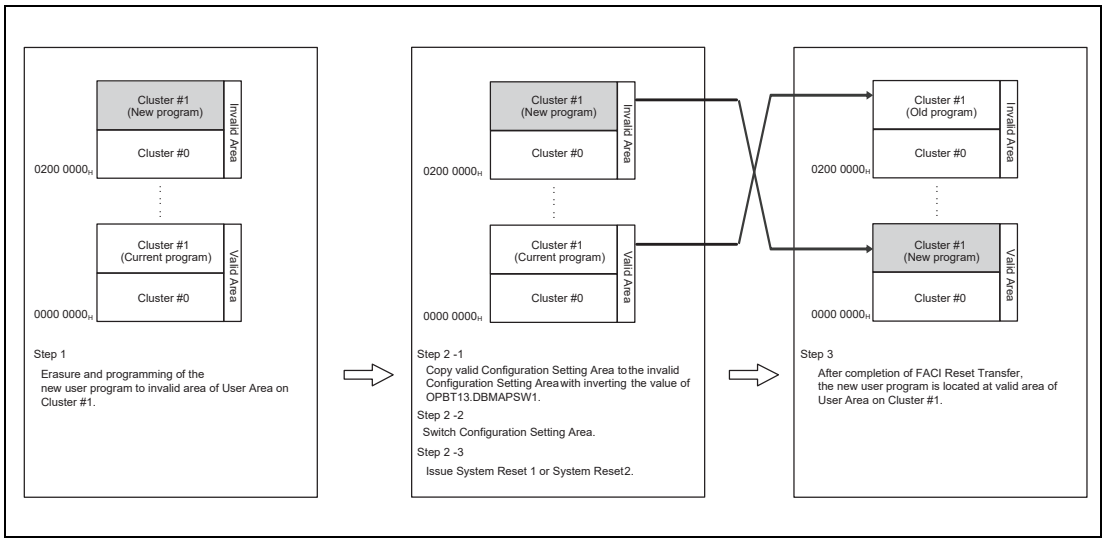


Figure 51.15 Example of User Program Update Procedure in the Field (In case of Update User Area on Cluster #1)

51.9 Reading Flash Memory

51.9.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in Normal Operation Mode and User Boot Mode. Data can simply be read out through access to addresses in the code flash memory.

Reading data bits from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) are all 1 and lead to the detection of an ECC 2-bit error and generation of the corresponding exception. That also lead to the detection of Address Parity Error depending on the address and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state. See **Section 44, Functional Safety** for the details of ECC and Address Parity.

51.9.2 Parallel Access to the Code Flash memory

The code flash memory is multi banked configuration. One bus master may access one bank and another bus master access other bank without a wait for arbitration of contention for the access path.

When multiple bus masters access the same bank, the accesses are arbitrated in round-robin fashion.

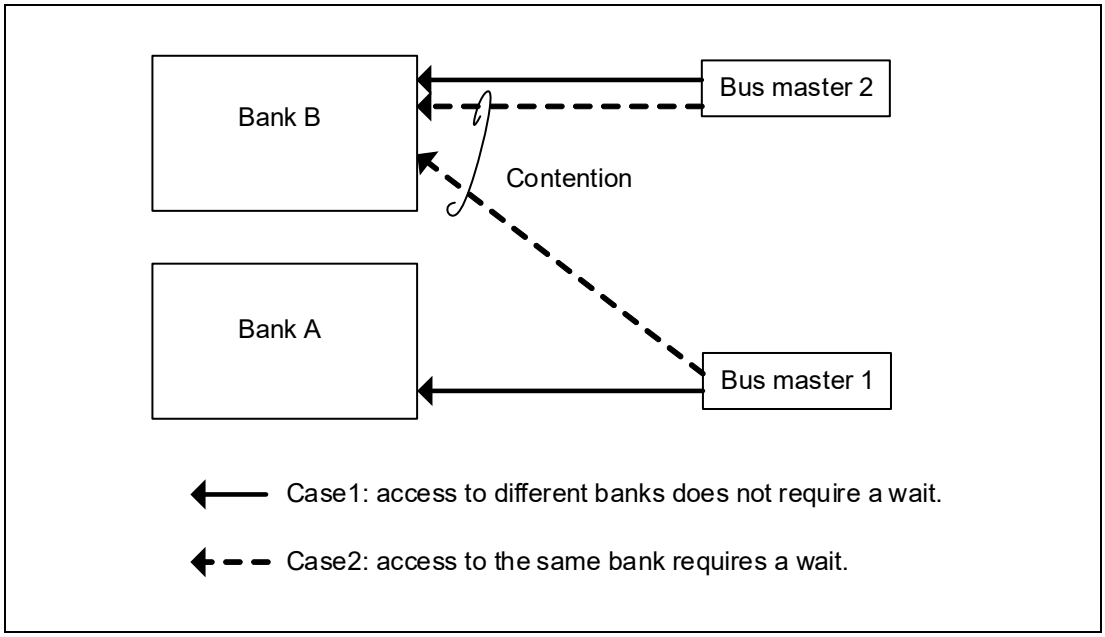


Figure 51.16 Operation in Access to Banks A and B (Example)

51.9.3 Reading Data Flash Memory

Reading data bits from an area of data flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) are all 1 and lead to the detection of an ECC 2-bit error. Use blank checking when you need to confirm that an area is in the non-programmed state. See **Section 44, Functional Safety** for the details of ECC.

51.10 Blank Checking Flash Memory

This product has Blank Check Area. This area is a mirror area of the flash memory and can read address parity and ECC field written in flash memory. In addition, blank flag is set if the read object address is blank.

51.10.1 Blank Check Area of Code Flash Memory

Table 51.29 Blank check address of code flash memory

Name	Address	Object Address	
		User Code Area*1	User Boot Area*1
Blank check address of code flash memory in Single Map Mode	0C00 0000_H + 20_H × n	0000 0000_H + 20_H × n to 0000 0000_H + 20_H × n + $1F_H$ n = 0,1,2,…,524287 for U2A-EVA (U2A16 mode) / U2A16 n = 0,1,2,…,262143 for U2A-EVA (U2A8 mode) / U2A8 n = 0,1,2,…,98303, 131072 + (0,1,2,…,98303) for U2A6	0800 0000_H + 20_H × n to 0800 0000_H + 20_H × n + $1F_H$ n = 0,1,2,…,2047 for Bank A n = 131072 + (0,1,2,…,2047) for Bank B
Blank check address of code flash memory in Double Map Mode	Valid area: 0C00 0000_H + 20_H × n Invalid area: 0E00 0000_H + 20_H × n	Valid area: 0000 0000_H + 20_H × n to 0000 0000_H + 20_H × n + $1F_H$ Invalid area: 0200 0000_H + 20_H × n to 0200 0000_H + 20_H × n + $1F_H$ n = 0,1,2,…,262143 for U2A-EVA (U2A16 mode) / U2A16 n = 0,1,2,…,131071 for U2A-EVA (U2A8 mode) / U2A8 n = 0,1,2,…,98303 for U2A6	Valid area: 0800 0000_H + 20_H × n to 0800 0000_H + 20_H × n + $1F_H$ Invalid area: 0A00 0000_H + 20_H × n to 0A00 0000_H + 20_H × n + $1F_H$ n = 0,1,2,…,2047

Note 1. For details of this function, see **Section 51.11.2.1, BLCHKSELR — Blank Check Area Selection Register**.

Table 51.30 Blank check of code flash memory Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, all ‘0’ is returned.
15	BF	Blank Flag. 0: the object address is not blank. 1: the object address is blank.
14 to 11	Reserved	When read, all ‘0’ is returned.
10	AP	Address parity bit Address[27] and [24:5] are used for generating an address parity, and generation type is even parity. If object address is blank, AP bit is “1”.
9 to 0	ECC[9:0]	ECC bits for Code Flash 256bit data. If object address is blank, all bit of ECC is “1”.

51.10.2 Blank Check Area of Data Flash Memory

Table 51.31 Blank check address of data flash memory

Name	Address	Object Address
Blank check address of Data Area	FF40 0000_H + 4_H × n	FF20 0000_H + 4_H × n to FF20 0000_H +4_H × n + 3_H n = 0,1,2,…,147455 for U2A-EVA (U2A16 mode) / U2A16 n = 0,1,2,…,81919 for U2A-EVA (U2A8 mode) / U2A8 n = 0,1,2,…,49151, 65536 + (0,1,2,…,16383) for U2A6
Blank check address of Extended Data Area	FF52 0000_H + 4_H × n	FF32 0000_H + 4_H × n to FF32 0000_H +4_H × n + 3_H n = 0,1,2,…,511
Blank check address of Configuration Setting Area	FF52 0800_H + 4_H × n	FF32 0800_H + 4_H × n to FF32 0800_H +4_H × n + 3_H n = 0,1,2,…,1023
Blank check address of Security Setting Area	FF52 1800_H + 4_H × n	FF32 1800_H + 4_H × n to FF32 1800_H +4_H × n + 3_H n = 0,1,2,…,1023
Blank check address of Block Protection Area for FPSYS0	FF52 2800_H + 4_H × n	FF32 2800_H + 4_H × n to FF32 2800_H +4_H × n + 3_H n = 0,1,2,…,1023
Blank check address of Block Protection Area for FPSYS1	FF54 0000_H + 4_H × n	FF34 0000_H + 4_H × n to FF34 0000_H +4_H × n + 3_H n = 0,1,2,…,1023 for U2A-EVA (U2A16 mode) / U2A16
Blank check address of Switch Area	FF57 3800_H + 4_H × n	FF37 3800_H + 4_H × n to FF37 3800_H +4_H × n + 3_H n = 0,1,2,…,1023
Blank check address of TAG Area	FF57 4800_H + 4_H × n	FF37 4800_H + 4_H × n to FF37 4800_H +4_H × n + 3_H n = 0,1,2,…,511

Table 51.32 Blank check of data flash memory Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, all ‘0’ is returned.
15	BF	Blank Flag. 0: the object address is not blank. 1: the object address is blank.
14 to 7	Reserved	When read, all ‘0’ is returned.
6 to 0	ECC[6:0]	ECC bits for Data Flash 32bit data. If object address is blank, all bit of ECC is “1”.

51.11 Description of Registers

51.11.1 Registers Related to Product Information

In Product Info Area in Code Flash Memory, Product information (Product Name, Product Number, Unique Chip ID) is stored.

Table 51.33 shows the list of registers related to product information. Other addresses in this area are reservations. Do not access them.

Table 51.33 List of Registers Related to Product Information

Register Name	Symbol	Address*1	Access Size
Product number / Product version storage register	PRDNUM	<PRDINFk_base> +0040_H	32
Product name storage register	PRDNAME1-3	<PRDINFk_base> +($00A0_H$ to $00A8_H$)	32
Unique Chip Identifier	UCID0-3	<PRDINFk_base> +(1000_H to $100C_H$)	32

Note 1. <PRDINFk_base> (k = f, b)

51.11.1.1 PRDNUM — Product Number/Product Version Storage Register

Access: This register can be read in 32-bit units.

Address: <PRDINFk_base> + 0040_H

Value after reset: See **Table 51.35**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	DID_RN[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DID_PN[15:0]															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This value after reset is dependent on products. See **Table 51.35**.

Table 51.34 PRDNUM Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19 to 16	DID_RN[3:0]	Revision Number These bits indicate the revision. The value is the same as SDID[31:28] (see **Section 53, Boundary Scan**).
15 to 0	DID_PN[15:0]	Product Number These bits indicate the product number. The value is the same as SDID[27:12] (see **Section 53, Boundary Scan**).

Table 51.35 Relationship between Product Name and Value of PRDNUM (1/2)

Product Name	MSPI restriction*1	SVR restriction*2	PRDNUM
R7F702Z19AEDBG	Yes	No	0001 840B$_H$
R7F702Z19BFDBG	No	No	0001 8449$_H$
R7F702300EBBG-C	Yes	Yes	0001 8406$_H$
R7F702300EBBB-C	Yes	Yes	0001 8413$_H$
R7F702300EABA-C	Yes	Yes	0001 8407$_H$
R7F702300AEBBC-C	No	Yes	0001 8453$_H$
R7F702300AEBBB-C	No	Yes	0001 844C$_H$
R7F702300AFABA-C	No	Yes	0001 844D$_H$
R7F702300BEBBC-C	No	No	0001 847D$_H$
R7F702300BEBBB-C	No	No	0001 847E$_H$
R7F702300BFABA-C	No	No	0001 847F$_H$
R7F702301EBBA-C	Yes	Yes	0001 8414$_H$
R7F702301EABG-C	Yes	Yes	0001 8408$_H$
R7F702301AEBBA-C	No	Yes	0001 844E$_H$
R7F702301AFABG-C	No	Yes	0001 844F$_H$
R7F702301BEBBA-C	No	No	0001 8480$_H$
R7F702301BFABG-C	No	No	0001 8481$_H$
R7F702302FABB-C	No	No	0001 8420$_H$
R7F702302FABD-C	No	No	0001 8422$_H$

Table 51.35 Relationship between Product Name and Value of PRDNUM (2/2)

Product Name	MSPI restriction*1	SVR restriction*2	PRDNUM
R7F702302FAFK-C	No	No	0001 8423$_H$
R7F702302FAFM-C	No	No	0001 8424$_H$

Note 1. Refer **Section 19.8, MSPI restrictions**.

Note 2. Refer **Section 10.10, SVR restriction**.

51.11.1.2 PRDNAMEn — Product Name Storage Register (n = 1 to 3)

Access: This register can be read in 32-bit units.

Address: <PRDINFk_base> + $00A0_H$ + 04_H x (n – 1) (n = 1 to 3)

Value after reset: See **Table 51.37**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRDNAMEn[31:16] (n = 1 to 3)															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRDNAMEn[15:0] (n = 1 to 3)															
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value after reset is dependent on products. See **Table 51.37**.

Table 51.36 PRDNAMEn Contents

Bit Position	Bit Name	Function
31 to 0	PRDNAMEn[31:0]	Product name: Indicate product model names by using 12-byte ASCII code. PRDNAME1[31:0]: Fourth to first bytes of product model name PRDNAME2[31:0]: Eighth to fifth bytes of product model name PRDNAME3[31:0]: Twelfth to ninth bytes of the product model name

Table 51.37 Relationship between Product Name and Value of PRDNAME

Product Name	MSPI restriction*1	SVR restriction*2	PRDNAME3	PRDNAME2	PRDNAME1
R7F702Z19AEDBG	Yes	No	2020 4139_H	315A 3230_H	3746 3752_H
R7F702Z19BFDBG	No	No	2020 4239_H	315A 3230_H	3746 3752_H
R7F702300EBBG-C	Yes	Yes	2020 2030_H	3033 3230_H	3746 3752_H
R7F702300EBBB-C	Yes	Yes	2020 2030_H	3033 3230_H	3746 3752_H
R7F702300EABA-C	Yes	Yes	2020 2030_H	3033 3230_H	3746 3752_H
R7F702300AEBBC-C	No	Yes	2020 4130_H	3033 3230_H	3746 3752_H
R7F702300AEBBB-C	No	Yes	2020 4130_H	3033 3230_H	3746 3752_H
R7F702300AFABA-C	No	Yes	2020 4130_H	3033 3230_H	3746 3752_H
R7F702300BEBBC-C	No	No	2020 4230_H	3033 3230_H	3746 3752_H
R7F702300BEBBB-C	No	No	2020 4230_H	3033 3230_H	3746 3752_H
R7F702300BFABA-C	No	No	2020 4230_H	3033 3230_H	3746 3752_H
R7F702301EBBA-C	Yes	Yes	2020 2031_H	3033 3230_H	3746 3752_H
R7F702301EABG-C	Yes	Yes	2020 2031_H	3033 3230_H	3746 3752_H
R7F702301AEBBA-C	No	Yes	2020 4131_H	3033 3230_H	3746 3752_H
R7F702301AFABG-C	No	Yes	2020 4131_H	3033 3230_H	3746 3752_H
R7F702301BEBBA-C	No	No	2020 4231_H	3033 3230_H	3746 3752_H
R7F702301BFABG-C	No	No	2020 4231_H	3033 3230_H	3746 3752_H
R7F702302FABB-C	No	No	2020 2032_H	3033 3230_H	3746 3752_H
R7F702302FABD-C	No	No	2020 2032_H	3033 3230_H	3746 3752_H
R7F702302FAFK-C	No	No	2020 2032_H	3033 3230_H	3746 3752_H
R7F702302FAFM-C	No	No	2020 2032_H	3033 3230_H	3746 3752_H

Note 1. Refer **Section 19.8, MSPI restrictions**.
Note 2. Refer **Section 10.10, SVR restriction**.

51.11.1.3 Unique Chip IDn — Unique Chip Identifier (n= 0 to 3)

Unique Chip ID is flash part of unique chip Identifier.

Access: This register can be read in 32-bit units.

Address: <PRDINFk_base> + 1000_H + 04_H x n (n = 0 to 3)

Value after reset: The value after reset is unique chip by chip.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Unique Chip IDn[31:16]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Unique Chip IDn[15:0]															
Value after reset	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value after reset is unique chip by chip.

Table 51.38 Unique Chip IDn Contents

Bit Position	Bit Name	Function
31 to 0	Unique Chip IDn[31:0]	Unique Chip IDn

Note: Bit 31 to 24 of Unique Chip ID2 is all ‘1’.

51.11.2 Registers Related to Read Function

Table 51.39 shows the list of registers related to Read Function.

Table 51.39 Registers Related to Read Function

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
CCIBH0	Blank Check Area Selection Register	BLCHKSELR	<CCIB0_base> + C_H	32	PBG00#3	—

Table 51.40 Register Reset Conditions

Register	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
BLCHKSELR	√	√	√	√	√	—	—

51.11.2.1 BLCHKSELR — Blank Check Area Selection Register

The Blank Check Area Selection Register (BLCHKSELR) select memory mapping of Blank Check Area.

Access: This register can be read / written in 32-bit units.

Address: <CCIB0_base> + C_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	BLNKSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*2	W*2	W*2	W*2	W*2	W*2	W*2	W*2	R	R	R	R	R	R	R	R/W*1

Note 1. Writing is enabled only when $A5_H$ is written to the KEY bit.

Note 2. Write only. When read, 0 is returned.

Table 51.41 BLCHKSELR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	KEY[7:0]	These bits control the modification of the BLNKSEL bits. $A5_H$: BLNKSEL can be written Other than above: BLNKSEL cannot be written, then retain previous value These bits must be specified simultaneously with BLNKSEL writing. These bits are write only. When read, 0 is returned.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BLNKSEL	Memory map selection bit of Blank Check Area for Code Flash. 0: Blank Check Area is mapped to User Area. 1: Blank Check Area is mapped to User Boot Area.

51.11.3 Registers Related to Code Flash Memory Mapping

Table 51.42 shows the list of registers related to Code Flash Memory Mapping.

Table 51.42 Registers Related to Code Flash Memory Mapping

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
SCDS	Code Flash Memory Mapping Mode Select Status	CFMAPSTT	<SCDS_base> + 0450_H	32	—	—
SCDS	Code Flash Memory Valid Area Switching Status	CFSWSTT	<SCDS_base> + 0454_H	32	—	—

Table 51.43 Register Reset Conditions

Register	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
CFMAPSTT	√	√	√	—	√	—	—
CFSWSTT	√	√	√	—	√	—	—

51.11.3.1 CFMAPSTT — Code Flash Memory Mapping Mode Status Register

CFMAPSTT indicates Code Flash Memory Mapping Mode.

Access: This register can be read in 32-bit units.

Address: <SCDS_base> + 0450_H

Value after reset: Specified by the user
The value after reset is the value of OPBT12 in valid area of Configuration Setting Area.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MAPMODE[1:0]	
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This value is dependent on the value in the flash memory which is specified by the user

Table 51.44 CFMAPSTT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	MAPMODE[1:0]	Code Flash Memory Mapping Mode Select Status $0\ 0_B$: Double Map Mode. $0\ 1_B$: Single Map Mode. $1\ 0_B$: setting prohibited. $1\ 1_B$: setting prohibited.

51.11.3.2 CFSWSTT — Code Flash Memory Valid Area Switching Status Register

CFSWSTT indicates Code Flash Memory Valid Area by the cluster. The setting of this register is valid when Code Flash Memory Mapping Mode is Double Map Mode.

Access: This register can be read in 32-bit units.

Address: <SCDS_base> + 0454_H

Value after reset: Specified by the user
The value after reset is the value of OPBT13 in valid area of Configuration Setting Area.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBMAP SW1	DBMAP SW0
Value after reset	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This value is dependent on the value in the flash memory which is specified by the user

Table 51.45 CFSWSTT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DBMAPSW1	[For U2A-EVA/U2A16] Double Map Mode Switching of Cluster 1 0: Bank D is mapped in valid area. 1: Bank C is mapped in valid area. [For U2A8/U2A6] Set the value of valid area at the shipping.
0	DBMAPSW0	Double Map Mode Switching of Cluster 0 0: Bank B is mapped in valid area. 1: Bank A is mapped in valid area.

51.11.4 Registers Related to Hardware Property Area

Table 51.46 shows the list of registers related to Hardware Property Area.

Table 51.46 Registers Related to Hardware Property Area

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
FACI0	Switch Area Status Register	FSWASTAT_0	<FACI0_base> + $00A8_H$	8	PBG6L1#1	—
FACIn	FACI Reset Transfer Status n (n = 0,1)	FRTSTAT_n	<FACIn_base> + 0098_H	8	*1	—
FACIn	FACI Reset Transfer Warning Interrupt Enable n (n = 0,1)	FRTEINT_n	<FACIn_base> + $009C_H$	8	*1	—

Note 1. n = 0: PBG6L1#1
n = 1: PBG6L1#2

Table 51.47 Register Reset Conditions

Register	Reset Category						
	Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
FSWASTAT_0	√	√	√	—	√	—	—
FRTSTAT_n	√	√	√	—	√	—	—
FRTEINT_n	√	√	√	—	√	—	—

51.11.4.1 FSWASTAT_0 — Switch Area Status Register

FSWASTAT_0 indicates valid area and status of Switch Area.

For details of this register, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

51.11.4.2 FRTSTAT_n — FACI Reset Transfer Status Register (n = 0,1)

FRTSTAT_n indicates FACI reset transfer warning status of FPSYSn.

[For U2A8/U2A6] No warning from FPSYS1 because there is no Hardware Property Area in FPSYS1.

See **Section 51.17.1, FACI reset transfer** for details of FACI reset transfer.

Access: This register can be read in 8-bit units.

Address: <FACIn_base> + 0098_H (n = 0,1)

Value after reset: $0X_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTEDTCT	RTECRCT
Value after reset	0	0	0	0	0	0	0/1	0/1
R/W	R	R	R	R	R	R	R	R

Table 51.48 FRTSTAT_n Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	RTEDTCT	Indicates that ECC 2-bit error has been detected during FACI reset transfer. 0: No error has been detected. 1: An error has been detected. RTEDTCT bit will be updated when it is reset and FACI reset transfer is finished.
0	RTECRCT	Indicates that ECC 1-bit error has been corrected during FACI reset transfer. 0: No error has been corrected. 1: An error has been corrected. RTECRCT bit will be updated when it is reset and FACI reset transfer is finished.

51.11.4.3 FRTEINT_n — FACI Reset Transfer Warning Interrupt Enable Register (n = 0,1)

FRTEINT_n enables FPSYSn FACI reset transfer warning interrupt.

FPSYSn FACI reset transfer warning interrupt is notified to ECM.

[For U2A8/U2A6] No warning from FPSYS1 because there is no Hardware Property Area in FPSYS1.

See **Section 51.17.1, FACI reset transfer** for details of FACI reset transfer.

Access: This register can be read / written in 8-bit units.

Address: <FACIn_base> + $009C_H$ (n = 0,1)

Value after reset: 03_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTEDIE	RTECIE
Value after reset	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W

Table 51.49 FRTEINT_n Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTEDIE*1	Enables FPSYSn FACI reset transfer warning interrupt when ECC 2-bit error occurs during FACI reset transfer and RTEDTCT bit in each FRTSTAT_n register becomes 1. 0: Does not generate FPSYSn FACI reset transfer warning interrupt when RTEDTCT = 1. 1: Generates FPSYSn FACI reset transfer warning interrupt when RTEDTCT = 1.
0	RTECIE*1	Enables FPSYSn FACI reset transfer warning interrupt when ECC 1-bit error occurs during FACI reset transfer and RTECRCT bit in each FRTSTAT_n register becomes 1. 0: Does not generate FPSYSn FACI reset transfer warning interrupt when RTECRCT = 1. 1: Generates FPSYSn FACI reset transfer warning interrupt when RTECRCT = 1.

Note 1. It is necessary to clear RTEDIE and RTECIE before clearing the ECM error source status for FACI reset transfer warning interrupt because FRTSTAT_n register is updated only when it is reset and FACI reset transfer is finished.

51.12 Configuration Setting Area (Option Bytes, Reset Vector)

Configuration Setting Area consists of two areas (Area 0 and Area 1). Valid area is mapped in front side, Invalid area is mapped in back side. Valid area cannot be updated whereas Invalid area can be updated.

Area 0 is selected as valid area at the shipping from RENESAS.

Area 1 is selected as invalid area and erased at the shipping from RENESAS.

The flash memory has the area to store a data specified by the user for configuration settings. Changes in settings become effective after TAG Update and release from the reset state (see **Section 51.17, Reset Transfer**).

OTP Setting of Configuration Setting Area becomes effective without reset. Valid / Invalid area switching for read access also becomes effective without reset.

Table 51.50 shows Configuration Setting Area. For setting and reading this area, see the *Renesas Flash Programmer Flash Programming Software User's Manual*, or the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 51.50 Configuration Setting Area (1/2)

Name	Address*9	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	CSAVOF/ CSAVOFC Number
Valid Option Byte Flag (CSAVOF0) *6 *8	<CSAk_base>+ 0000_H	5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0004_H to $001C_H$)	Erased	Can not Write	—	—
Valid Option Byte Flag (CSAVOF8-19) *6 *8	<CSAk_base>+ (0020_H to $004C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0050_H to $00FC_H$)	Erased	Can not Write	—	—
VOF Program completion flag (CSAVOFC0) *6 *8	<CSAk_base>+ 0100_H	5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0104_H to $011C_H$)	Erased	Can not Write	—	—
VOF Program completion flag (CSAVOFC8-19) *6 *8	<CSAk_base>+ (0120_H to $014C_H$)	All 5AA5 A55A$_H$	Can not Write	—	—
Reserved*7	<CSAk_base>+ (0150_H to $01FC_H$)	Erased	Can not Write	—	—
OTP Setting of Configuration setting Area*4	<CSAk_base>+ (0200_H to 0208_H)	All FFFF FFFF$_H$	—	—	0
Reserved (need to be programmed)*5	<CSAk_base>+ ($020C_H$ to $021C_H$)	All FFFF FFFF$_H$	—	—	0
Reserved*7	<CSAk_base>+ (0220_H to $02FC_H$)	Erased	Can not Write	—	—
Software Configuration Option Byte	<CSAk_base>+ 0300_H + 04_H x n (n = 0 to 31)	All 0000 0000_H	Customer ID A	—	8 to 11
Reset Vector (PE0)	<CSAk_base>+ 0380_H	0000 0000_H	Customer ID A	—	12
Reset Vector (PE1)	<CSAk_base>+ 0384_H	0000 0000_H	Customer ID A	—	12
Reset Vector (PE2)	<CSAk_base>+ 0388_H	0080 0000_H	Customer ID A	—	12
Reset Vector (PE3)	<CSAk_base>+ $038C_H$	0080 0000_H	Customer ID A	—	12
Reserved (need to be programmed) *5	<CSAk_base>+ (0390_H to $039C_H$)	All FFFF FFFF$_H$	Customer ID A	—	12
Option byte 0 (OPBT0)	<CSAk_base>+ $03A0_H$	3FF3 0010_H	Customer ID A	—	13
Option byte 1 (OPBT1)	<CSAk_base>+ $03A4_H$	F0FB 0000 $_H$	Customer ID A	—	13
Option byte 2 (OPBT2)	<CSAk_base>+ $03A8_H$	7FFF FFFF$_H$	Customer ID A	—	13
Option byte 3 (OPBT3)	<CSAk_base>+ $03AC_H$	F1FF FEFE$_H$	Customer ID A	—	13
Option byte 4 (OPBT4)	<CSAk_base>+ $03B0_H$	0C0C 0C0F$_H$	Customer ID A	—	13
Reserved (need to be programmed) *5	<CSAk_base>+ $03B4_H$	FFFF FFFF$_H$	Customer ID A	—	13
Option byte 6 (OPBT6)	<CSAk_base>+ $03B8_H$	FFFF 0FC3$_H$	Customer ID A	—	13
Option byte 7 (OPBT7)	<CSAk_base>+ $03BC_H$	FFFF FFFF$_H$	Customer ID A	—	13
Option byte 8 (OPBT8)	<CSAk_base>+ $03C0_H$	FFFF FFFE$_H$	Customer ID A	—	14
Option byte 9 (OPBT9)	<CSAk_base>+ $03C4_H$	FFF1 FFFF$_H$	Customer ID A	—	14
Option byte 10 (OPBT10)	<CSAk_base>+ $03C8_H$	FBFD 288E$_H$	Customer ID A	—	14
Option byte 11 (OPBT11)	<CSAk_base>+ $03CC_H$	FFFF FFFF$_H$	Customer ID A	—	14
Option byte 12 (OPBT12)	<CSAk_base>+ $03D0_H$	FFFF FFFD$_H$	Customer ID A	—	14
Option byte 13 (OPBT13)	<CSAk_base>+ $03D4_H$	FFFF FFFF$_H$	Customer ID A	—	14

Table 51.50 Configuration Setting Area (2/2)

Name	Address*9	State at the shipping*1	Write Protection ID*2	Read Protection ID*3	CSAVOF/ CSAVOFC Number
Option byte 14 (OPBT14)	<CSAk_base>+ $03D8_H$	$0000\ 0002_H$ (U2A-EVA) $0000\ 0192_H$ (U2A16/U2A8/ U2A6)	Customer ID A	—	14
Reserved (need to be programmed) *5	<CSAk_base>+ $03DC_H$	$FFFF\ FFFF_H$	Customer ID A	—	14
Option byte 16 (OPBT16)	<CSAk_base>+ $03E0_H$	$6FFF\ FFFF_H$	Customer ID A	—	15
Option byte 17 (OPBT17)	<CSAk_base>+ $03E4_H$	$FFFF\ FFFF_H$	Customer ID A	—	15
Option byte 18 (OPBT18)	<CSAk_base>+ $03E8_H$	$FFFF\ FFFF_H$	Customer ID A	—	15
Option byte 19 (OPBT19)	<CSAk_base>+ $03EC_H$	$FFFF\ FFFF_H$	Customer ID A	—	15
Option byte 20 (OPBT20)	<CSAk_base>+ $03F0_H$	$F088\ FF00_H$	Customer ID A	—	15
Option byte 21 (OPBT21)	<CSAk_base>+ $03F4_H$	$0000\ 00CC_H$	Customer ID A	—	15
Option byte 22 (OPBT22)	<CSAk_base>+ $03F8_H$	$C000\ C000_H$	Customer ID A	—	15
Option byte 23 (OPBT23)	<CSAk_base>+ $03FC_H$	$FFFF\ C000_H$	Customer ID A	—	15
Reserved (need to be programmed) *5	<CSAk_base>+ (0400_H to $047C_H$)	All $FFFF\ FFFF_H$	Customer ID A	—	16 to 19
Reserved*7	<CSAk_base>+ (0480_H to $07FC_H$)	Erased	Can not Write	—	—

Note 1. This is the value of valid area at the shipping. Invalid area is erased at the shipping.
Note 2. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary for programming the data.
Note 3. When On-chip debug and S_OPBT3.CFRPF is “0”, the Customer ID A authentication is needed to read.
Note 4. For details of this function, see **Section 47.7, Configuration Setting Area (Option Bytes, Reset Vector)**.
Note 5. For this reserved area, program $FFFF\ FFFF_H$ when programming Configuration Setting Area.
Note 6. This area will be updated automatically when programming Configuration Setting Area.
Note 7. This area cannot be programmed. When this area is read, it behaves as if it is erased.
Note 8. For details of this function, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.
Note 9. <CSAk_base> (k = f, b)

51.12.1 CSAVOFn — Valid Option Byte Flag n for Configuration Setting Area (n = 0, 8 to 19)

This flag indicates that the related option bytes are programmed or not.

The relation between each CSAVOFn and option bytes are shown in **Table 51.50**.

This flag is referred by flash sequencer in order to check whether the related option bytes are programmed or not. This flag will be updated automatically when programming Configuration Setting Area. The updated value will be 5AA5 $A55A_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0000_H + 04_H x n

Value at the shipping: 5AA5 $A55A_H$ (valid area)
Erased (invalid area)

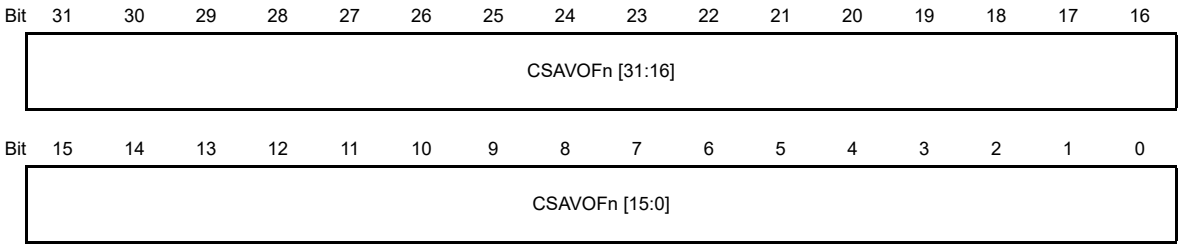


Table 51.51 CSAVOFn Contents

Bit Position	Bit Name	Function
31 to 0	CSAVOFn [31:0]	Program completion flag. 5AA5 $A55A_H$: The related option bytes are programmed. Other than above: The related option bytes are not programmed.

51.12.2 CSAVOFCn — VOF Program completion flag n for Configuration Setting Area (n = 0, 8 to 19)

This flag indicates that CSAVOFn is programmed or not.
The relation between each CSAVOFCn and option bytes are shown in **Table 51.50**.
This flag is prepared to check whether the related option bytes are programmed or not by user program.
This flag will be updated automatically when programming Configuration Setting Area. The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0100$_H$ + 04$_H$ x n

Value at the shipping: 5AA5 A55A$_H$ (valid area)
Erased (invalid area)

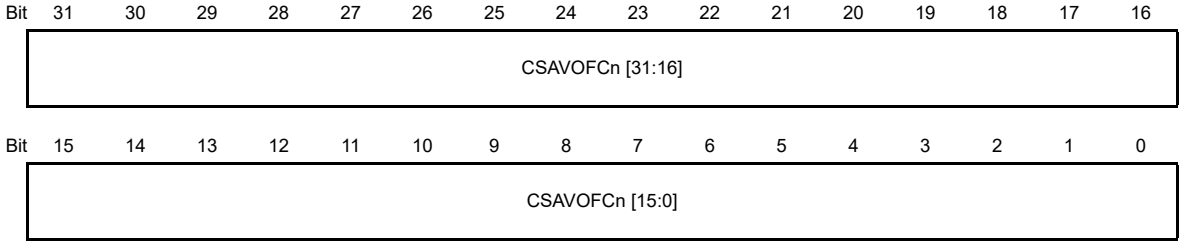


Table 51.52 CSAVOFCn Contents

Bit Position	Bit Name	Function
31 to 0	CSAVOFCn [31:0]	CSAVOFn Program completion flag. 5AA5 A55A$_H$: CSAVOFn is programmed. Other than above: CSAVOFn is not programmed.

51.12.3 Software Configuration Option Byte (n = 0 to 31)

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + 0300_H + 04_H x n (n = 0 to 31)

Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

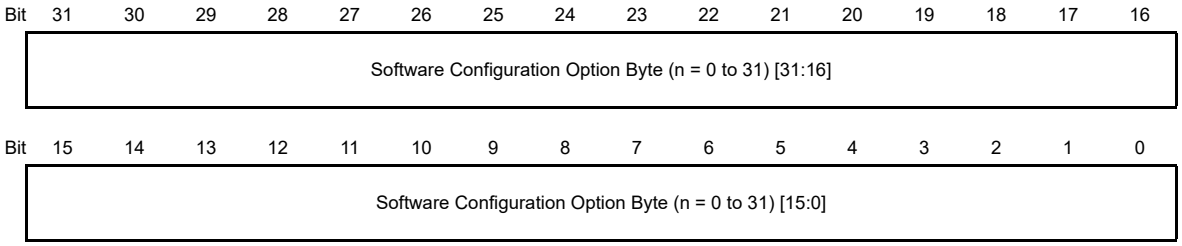


Table 51.53 Software Configuration Option Byte (n = 0 to 31) Contents

Bit Position	Bit Name	Function
31 to 0	—	Software Configuration Option Byte (n = 0 to 31) These data do not affect hardware behavior.

51.12.4 Reset Vector PE0

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Address: <CSAk_base> + 0380_H

Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	OPT_RBASE0[23:16]							

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPT_RBASE0[15:9]							—	—	—	—	—	—	—	—	—

Table 51.54 Reset Vector PE0 Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	Set the value of valid area at the shipping.
23 to 9	OPT_RBASE0 [23:9]	These bits indicate the reset vector base address of PE0 in Normal Operation Mode.
8 to 0	Reserved	Set the value of valid area at the shipping.

51.12.5 Reset Vector PE1

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + 0384_H

Value at the shipping: $0000\ 0000_H$ (valid area)
Erased (invalid area)

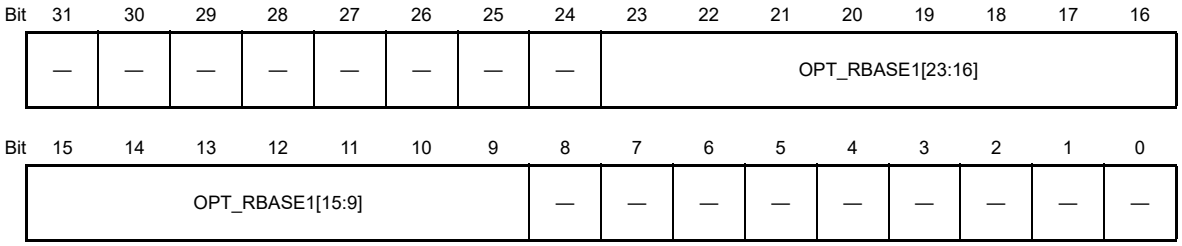


Table 51.55 Reset Vector PE1 Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	Set the value of valid area at the shipping.
23 to 9	OPT_RBASE1 [23:9]	These bits indicate the reset vector base address of PE1 in Normal Operation Mode and User Boot Mode.
8 to 0	Reserved	Set the value of valid area at the shipping.

51.12.6 Reset Vector PEn (n = 2, 3)

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + 0380_H + 04_H x n (n = 2, 3)

Value at the shipping: $0080\ 0000_H$ (valid area)
Erased (invalid area)

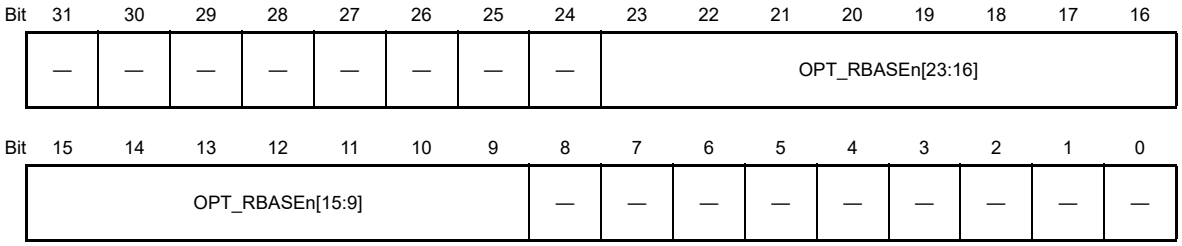


Table 51.56 Reset Vector PEn Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	Set the value of valid area at the shipping.
23 to 9	OPT_RBASEn[23:9]	[For U2A-EVA/U2A16] These bits indicate the reset vector base address of PEn (n=2,3) in Normal Operation Mode and User Boot Mode. [For U2A8/U2A6] Set the value of valid area at the shipping.
8 to 0	Reserved	Set the value of valid area at the shipping.

51.12.7 OPBT0 — Option Byte 0

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03A0_H$

Value at the shipping: 3FF3 0010_H (valid area)
Erased (invalid area)

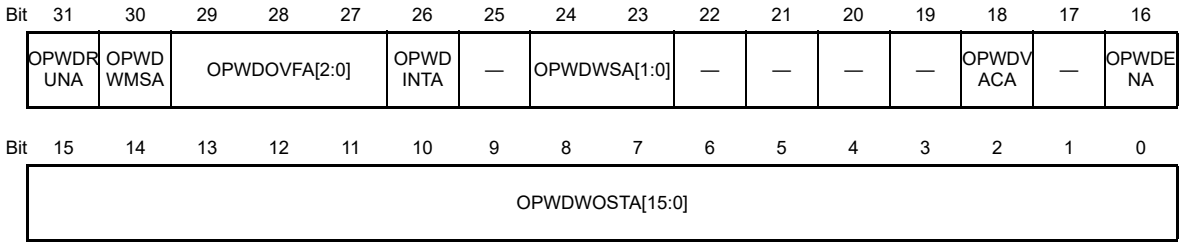


Table 51.57 OPBT0 Contents (1/2)

Bit Position	Bit Name	Function
31	OPWDRUNA	This bit sets the start mode of WDTBA. 0: WDTBA software trigger start mode 1: WDTBA default start mode
30	OPWDWMSA	This bit selects Window Open function mode of WDTBA 0: Window Size of Window Open function is set by WDTBAWS[1:0]. WDTBATIT outputs When the counter reaches 75% of the overflow setting defined by WDTBAMD.WDTBAOVF[2:0] 1: Window Size of Window Open function is set by WDTBAWOST WDTBATIT outputs When WDTBAWIS matches WDT counter.
29 to 27	OPWDOVFA[2:0]	These bits select the overflow interval time of WDTBA.

OPWDOVFA[2:0]	Overflow Interval Time
000_B	2^9 / WDTBTCKI
001_B	2^{10} / WDTBTCKI
010_B	2^{11} / WDTBTCKI
011_B	2^{12} / WDTBTCKI
100_B	2^{13} / WDTBTCKI
101_B	2^{14} / WDTBTCKI
110_B	2^{15} / WDTBTCKI
111_B	2^{16} / WDTBTCKI

26	OPWDINTA	This bit enables or disables a 75% interrupt request of WDTBA (WDTBATIT). 0: WDTBATIT disabled 1: WDTBATIT enabled
25	Reserved	Set the value of valid area at the shipping.
24, 23	OPWDWSA[1:0]	These bits select the window open period of WDTBA.

OPWDWSA[1:0]	Window Open Period
00_B	25%
01_B	50%
10_B	75%
11_B	100%

Table 51.57 OPBT0 Contents (2/2)

Bit Position	Bit Name	Function
22 to 19	Reserved	Set the value of valid area at the shipping.
18	OPWDVACA	This bit specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTBAWDTE (fixed) 1: WDTBAEVAC (variable)
17	Reserved	Set the value of valid area at the shipping.
16	OPWDENA	This bit enables/disables the WDTBA 0: WDTBA is disabled 1: WDTBA is enabled
15 to 0	OPWDWOSTA	These bits are Window Open Start register for setting the start timing of the window open.

For details of these data, see **Section 31, Window Watchdog Timer (WDTB)**.

51.12.8 OPBT1 — Option Byte 1

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03A4_H$

Value at the shipping: F0FB 0000_H (valid area)
Erased (invalid area)

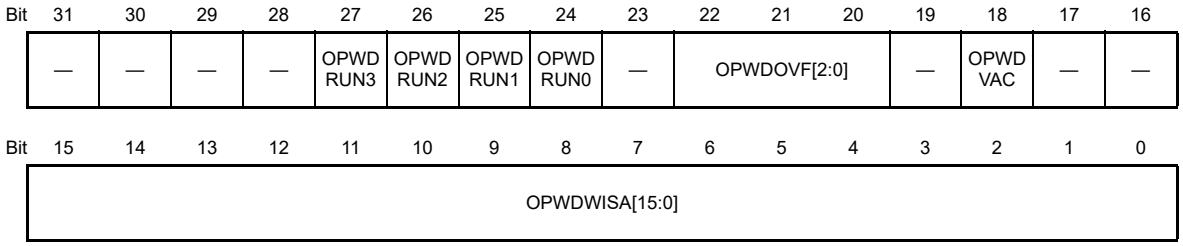


Table 51.58 OPBT1 Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	Set the value of valid area at the shipping.
27	OPWDRUN3	[For U2A-EVA/U2A16] This bit sets the start mode of WDTB3. 0: WDTB3 software trigger start mode 1: WDTB3 default start mode [For U2A8/U2A6] Arbitrary value can be specified.
26	OPWDRUN2	[For U2A-EVA/U2A16] This bit sets the start mode of WDTB2. 0: WDTB2 software trigger start mode 1: WDTB2 default start mode [For U2A8/U2A6] Arbitrary value can be specified.
25	OPWDRUN1	This bit sets the start mode of WDTB1. 0: WDTB1 software trigger start mode 1: WDTB1 default start mode
24	OPWDRUN0	This bit sets the start mode of WDTB0. 0: WDTB0 software trigger start mode 1: WDTB0 default start mode
23	Reserved	Set the value of valid area at the shipping.
22 to 20	OPWDOVF[2:0]	These bits select the overflow interval time of WDTBn (n = 0 to 3). OPWDOVF[2:0] / Overflow Interval Time 000_B / 2^9 / WDTBTCKI 001_B / 2^{10} / WDTBTCKI 010_B / 2^{11} / WDTBTCKI 011_B / 2^{12} / WDTBTCKI 100_B / 2^{13} / WDTBTCKI 101_B / 2^{14} / WDTBTCKI 110_B / 2^{15} / WDTBTCKI 111_B / 2^{16} / WDTBTCKI
19	Reserved	Set the value of valid area at the shipping.
18	OPWDVAC	This bit specifies the trigger register of WDTBn (n = 0 to 3) for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTBnWDTE (fixed) 1: WDTBnEVAC (variable)

Table 51.58 OPBT1 Contents (2/2)

Bit Position	Bit Name	Function
17, 16	Reserved	Set the value of valid area at the shipping.
15 to 0	OPWDWISA	These bits specify WDTBA Interrupt Output Timing Setting Register

For details of these data, see **Section 31, Window Watchdog Timer (WDTB)**.

51.12.9 OPBT2 — Option Byte 2

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

For details of this OPBT2, see the *RH850/U2A-EVA Group Security User’s Manual: Hardware*.

51.12.10 OPBT3 — Option Byte 3

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03AC_H$

Value at the shipping: F1FF $FEFE_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PE3_DISABLE	PE2_DISABLE	PE1_DISABLE	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HWBIST	—	—	—	TESTSET[1:0]		LBISTSEL[1:0]		—	—	—	—	—	—	STMSEL1	STMSEL0

Table 51.59 OPBT3 Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	Set the value of valid area at the shipping.
27	PE3_DISABLE	[For U2A-EVA/U2A16] PE3_DISABLE bit 0: PE3 enable 1: PE3 disable [For U2A8/U2A6] Arbitrary value can be specified.
26	PE2_DISABLE	[For U2A-EVA/U2A16] PE2_DISABLE bit 0: PE2 enable 1: PE2 disable [For U2A8/U2A6] Arbitrary value can be specified.
25	PE1_DISABLE	PE1_DISABLE bit 0: PE1 enable 1: PE1 disable
24 to 16	Reserved	Set the value of valid area at the shipping.
15	HWBIST	Power On BIST enable 0: BIST is skipped 1: BIST is executed
14 to 12	Reserved	Set the value of valid area at the shipping.
11, 10	TESTSET[1:0]	BIST selection 0 0_B: Prohibited 0 1_B: LBIST Only 1 0_B: MBIST Only 1 1_B: LBIST and MBIST For details, see **Section 44, Functional Safety**.
9, 8	LBISTSEL[1:0]	BIST scenario selection 0 0_B: LBIST scenario 1/MBIST scenario 1 0 1_B: LBIST scenario 2/MBIST scenario 1 1 0_B: LBIST scenario 3/MBIST scenario 1 1 1_B: Prohibited For details, see **Section 44, Functional Safety**.
7 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	STMSEL1, STMSEL0	These bits select operating mode and startup area. When FLMD0 pin is 0, the operating mode and startup area are selected depending on the combination of the STMSEL1 and STMSEL0. For details, see **Section 5, Operating Modes**.

51.12.11 OPBT4 — Option Byte 4

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03B0_H$

Value at the shipping: 0C0C 0C0F$_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ISOVDDFLTW [1:0]		ISOVDDCLKSEL [1:0]		ISOVDD FLTEN	—	ISOVDD HDE	ISOVDD LDE	VCCFLTW [1:0]		VCCCLKSEL [1:0]		VCCFL TEN	—	VCCHD E	VCCLD E

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E0VCCFLTW [1:0]		E0VCCCLKSEL [1:0]		E0VCC FLTEN	—	E0VCC HDE	E0VCC LDE	AWOVDDFLTW [1:0]		AWOVDDCLKS EL[1:0]		AWOVD DFLTEN	DSDET EN	—	—

Table 51.60 OPBT4 Contents (1/2)

Bit Position	Bit Name	Function
31, 30	ISOVDDFLTW[1:0]	VMON configurations for ISOVDD Select the minimum filtering width of digital noise filter.
29, 28	ISOVDDCLKSEL [1:0]	VMON configurations for ISOVDD Select a clock of the digital noise filter.
27	ISOVDDFLTEN	VMON configurations for ISOVDD Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
26	Reserved	Set the value of valid area at the shipping.
25	ISOVDDHDE	VMON configurations for ISOVDD ISOVDD High voltage detection enable.
24	ISOVDDLDE	VMON configurations for ISOVDD ISOVDD Low voltage detection enable.
23, 22	VCCFLTW[1:0]	VMON configurations for VCC Select the minimum filtering width of digital noise filter.
21, 20	VCCCLKSEL[1:0]	VMON configurations for VCC Select a clock of the digital noise filter.
19	VCCFLTEN	VMON configurations for VCC Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
18	Reserved	Set the value of valid area at the shipping.
17	VCCHDE	VMON configurations for VCC VCC High voltage detection enable.
16	VCCLDE	VMON configurations for VCC VCC Low voltage detection enable.
15, 14	E0VCCFLTW[1:0]	VMON configurations for E0VCC Select the minimum filtering width of digital noise filter.
13, 12	E0VCCCLKSEL[1:0]	VMON configurations for E0VCC Select a clock of the digital noise filter.
11	E0VCCFLTEN	VMON configurations for E0VCC Enable output filter for $\overline{\text{VMONOUT}}$ and VMONF.
10	Reserved	Set the value of valid area at the shipping.
9	E0VCCHDE	VMON configurations for E0VCC E0VCC High voltage detection enable.
8	E0VCCLDE	VMON configurations for E0VCC E0VCC Low voltage detection enable.

Table 51.60 OPBT4 Contents (2/2)

Bit Position	Bit Name	Function
7, 6	AWOVDDFLTW[1:0]	VMON configurations for AWOVDD Select the minimum filtering width of digital noise filter.
5, 4	AWOVDDCLKSEL [1:0]	VMON configurations for AWOVDD Select a clock of the digital noise filter.
3	AWOVDDFLTEN	VMON configurations for AWOVDD Enable output filter for VMONOUT and VMONF.
2	DSDETEN	VMON control during DeepSTOP
1, 0	Reserved	Set the value of valid area at the shipping.

For details of these data, see **Section 11, Power Supply Voltage Monitor**.

51.12.12 OPBT6 — Option Byte 6

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03B8_H$

Value at the shipping: FFFF $0FC3_H$ (valid area)
Erased (invalid area)

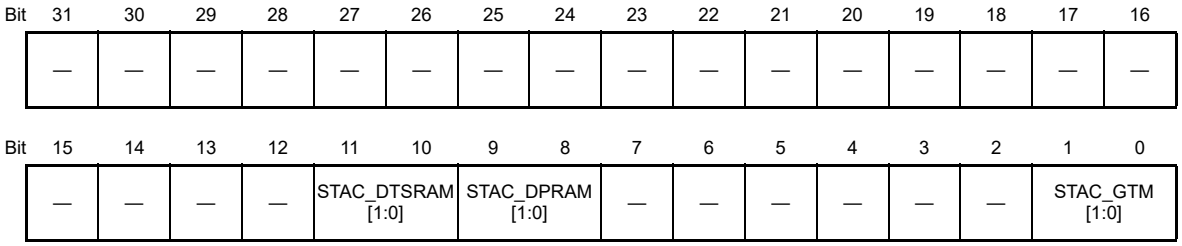


Table 51.61 OPBT6 Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	Set the value of valid area at the shipping.
11, 10	STAC_DTSRAM[1:0]	RAM Initialization Mode for DTSRAM. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
9, 8	STAC_DPRAM[1:0]	RAM Initialization Mode for DPRAM. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
7 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	STAC_GTM[1:0]	RAM Initialization Mode for GTM. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled

For details of these data, see **Section 9, Reset Controller**.

51.12.13 OPBT7 — Option Byte 7

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03BC_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	STAC_MMCA [1:0]		STAC_MSPI9 [1:0]		STAC_MSPI8 [1:0]	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STAC_MSPI7 [1:0]		STAC_MSPI6 [1:0]		STAC_MSPI5 [1:0]		STAC_MSPI4 [1:0]		STAC_MSPI3 [1:0]		STAC_MSPI2 [1:0]		STAC_MSPI1 [1:0]		STAC_MSPI0 [1:0]	

Table 51.62 OPBT7 Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	Set the value of valid area at the shipping.
21, 20	STAC_MMCA[1:0]	RAM Initialization Mode for MMCA. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
19, 18	STAC_MSPI9[1:0]	RAM Initialization Mode for MSPI9. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
17, 16	STAC_MSPI8[1:0]	RAM Initialization Mode for MSPI8. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
15, 14	STAC_MSPI7[1:0]	RAM Initialization Mode for MSPI7. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
13, 12	STAC_MSPI6[1:0]	RAM Initialization Mode for MSPI6. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
11, 10	STAC_MSPI5[1:0]	RAM Initialization Mode for MSPI5. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
9, 8	STAC_MSPI4[1:0]	RAM Initialization Mode for MSPI4. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
7, 6	STAC_MSPI3[1:0]	RAM Initialization Mode for MSPI3. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled
5, 4	STAC_MSPI2[1:0]	RAM Initialization Mode for MSPI2. X 0_B: Disabled 0 1_B: Prohibited 1 1_B: Enabled

Table 51.62 OPBT7 Contents (2/2)

Bit Position	Bit Name	Function
3, 2	STAC_MSPI1[1:0]	RAM Initialization Mode for MSPI1. X 0 $_B$: Disabled 0 1 $_B$: Prohibited 1 1 $_B$: Enabled
1, 0	STAC_MSPI0[1:0]	RAM Initialization Mode for MSPI0. X 0 $_B$: Disabled 0 1 $_B$: Prohibited 1 1 $_B$: Enabled

For details of these data, see **Section 9, Reset Controller**.

51.12.14 OPBT8 — Option Byte 8

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03C0_H$

Value at the shipping: FFFF $FFFE_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ETN_SGMII_SEL	—	ETN_RMII_SEL	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 51.63 OPBT8 Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	Set the value of valid area at the shipping.
27	ETN_SGMII_SEL	[For U2A-EVA/U2A16(BGA516)] ETNB1 MII/SGMII select 0: MII 1: SGMII [For other devices] Set the value of valid area at the shipping.
26	Reserved	Set the value of valid area at the shipping.
25	ETN_RMII_SEL	ETNB0 MII/RMII select 0: MII 1: RMII
24 to 0	Reserved	Set the value of valid area at the shipping.

51.12.15 OPBT9 — Option Byte 9

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03C4_H$

Value at the shipping: $FFF1\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PE3_DCLS_DIS	PE2_DCLS_DIS	PE1_DCLS_DIS	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PE3_HVE	PE2_HVE	PE1_HVE	PE0_HVE	—	—	—	—	—	—	—	EVA_PRD_EMU*1

Note 1. [For U2A-EVA]

Table 51.64 OPBT9 Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	Set the value of valid area at the shipping.
19	PE3_DCLS_DIS	[For U2A-EVA/U2A16] PE3 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled [For U2A8/U2A6] Arbitrary value can be specified.
18	PE2_DCLS_DIS	[For U2A-EVA/U2A16] PE2 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled [For U2A8/U2A6] Arbitrary value can be specified.
17	PE1_DCLS_DIS	PE1 DCLS Disable bit 0: DCLS Enabled 1: DCLS Disabled
16 to 12	Reserved	Set the value of valid area at the shipping.
11	PE3_HVE	[For U2A-EVA/U2A16] PE3 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled [For U2A8/U2A6] Arbitrary value can be specified.
10	PE2_HVE	[For U2A-EVA/U2A16] PE2 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled [For U2A8/U2A6] Arbitrary value can be specified.
9	PE1_HVE	PE1 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled
8	PE0_HVE	PE0 Hypervisor Enable 0: Hypervisor Disabled 1: Hypervisor Enabled
7 to 1	Reserved	Set the value of valid area at the shipping.
0	EVA_PRD_EMU	[For U2A-EVA] Emulation Device mode 0: U2A-EVA U2A8 mode emulation 1: U2A-EVA U2A16 mode emulation [For U2A16/U2A8/U2A6] Arbitrary value can be specified.

51.12.16 OPBT10 — Option Byte 10

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Address: <CSAk_base> + $03C8_H$

Value at the shipping: FBFD $288E_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	MOSC_EXCLK INPUT	—	MOSC_FREQ[2:0]			—	MOSC_AMP_SEL_A[2:0]			—	MOSC_AMP_SEL_B[2:0]		

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSC_CAP_SEL[3:0]				—	MOSC_RD_SEL_A[2:0]			—	MOSC_RD_SEL_B[2:0]			—	—	MOSC_SHTSTBY_A	MOSC_SHTSTBY_B

Table 51.65 OPBT10 Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	Set the value of valid area at the shipping.
28	MOSC_EXCLKINPUT	Main OSC input clock select. 0: Direct clock input to X1 (EXCLK mode). Main OSC amplifier is disabled. 1: Normal crystal oscillation. Main OSC amplifier is enabled.
27	Reserved	Set the value of valid area at the shipping.
26 to 24	MOSC_FREQ[2:0]	Main OSC frequency selection bit 000_B: 16 MHz 001_B: 20 MHz 010_B: 24 MHz 011_B: 40 MHz $1xx_B$: setting prohibited (need to configure all bit)
23	Reserved	Set the value of valid area at the shipping.
22 to 20	MOSC_AMP_SEL_A [2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation destabilization.
19	Reserved	Set the value of valid area at the shipping.
18 to 16	MOSC_AMP_SEL_B [2:0]	Main OSC trimming configuration These bits control OSC drivability during oscillation stabilization.
15 to 12	MOSC_CAP_SEL [3:0]	Main OSC trimming configuration These bits control internal capacitance.
11	Reserved	Set the value of valid area at the shipping.
10 to 8	MOSC_RD_SEL_A [2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation destabilization.
7	Reserved	Set the value of valid area at the shipping.
6 to 4	MOSC_RD_SEL_B [2:0]	Main OSC trimming configuration These bits control Damping resistor during oscillation stabilization.
3, 2	Reserved	Set the value of valid area at the shipping.
1	MOSC_SHTSTBY_A	Main OSC trimming configuration This bit controls OSC drivability during oscillation destabilization. MOSC_SHTSTBY_A must be set to 1.
0	MOSC_SHTSTBY_B	Main OSC trimming configuration This bit controls OSC drivability during oscillation stabilization. MOSC_SHTSTBY_B must be set to 0.

For details of these data, see **Section 13, Clock Controller**.

51.12.17 OPBT11 — Option Byte 11

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03CC_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CKDIVMD[1:0]		—	START UPPLL	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 51.66 OPBT11 Contents

Bit Position	Bit Name	Function
31, 30	CKDIVMD[1:0]	Products of CPU Frequency & CPU System Clock Setting 0 x $_B$: 240 MHz 1 0 $_B$: 320 MHz 1 1 $_B$: 400 MHz
29	Reserved	Set the value of valid area at the shipping.
28	STARTUPPLL	Start Up of Main OSC and PLL after reset released except DeepSTOP Reset. This setting has no effect in Serial Programing Mode. 0: Main OSC and PLL are enabled 1: Main OSC and PLL are disabled
27 to 0	Reserved	Set the value of valid area at the shipping.

For details of these data, see **Section 13, Clock Controller**.

51.12.18 OPBT12 — Option Byte 12

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03D0_H$

Value at the shipping: FFFF $FFFD_H$ (valid area)
Erased (invalid area)

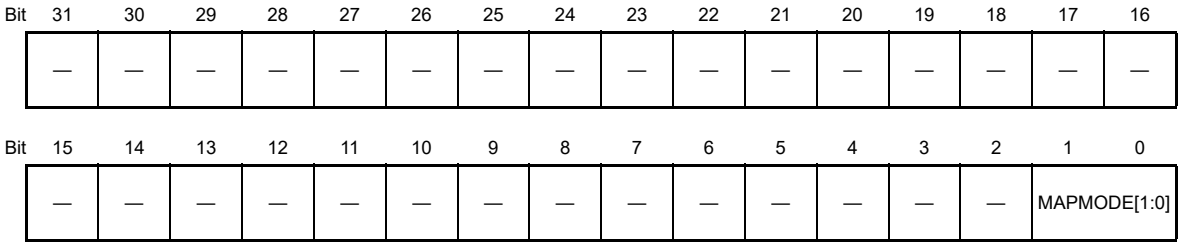


Table 51.67 OPBT12 Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	Set the value of valid area at the shipping.
1, 0	MAPMODE[1:0]	Code Flash Memory Mapping Mode Select $0\ 0_B$: Double Map Mode. $0\ 1_B$: Single Map Mode. $1\ 0_B$: setting prohibited. $1\ 1_B$: setting prohibited.

51.12.19 OPBT13 — Option Byte 13

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03D4_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

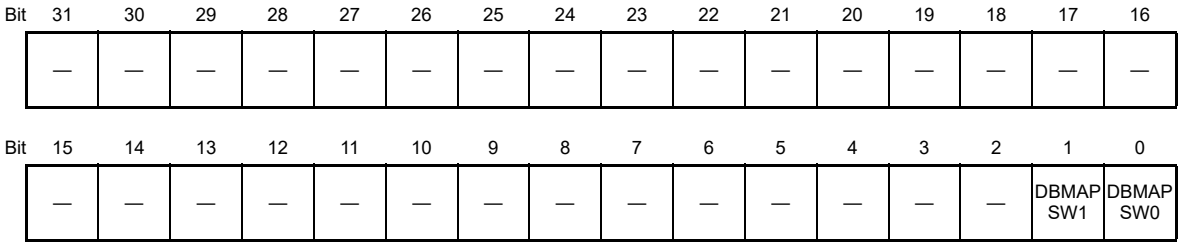


Table 51.68 OPBT13 Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	Set the value of valid area at the shipping.
1	DBMAPSW1	[For U2A-EVA/U2A16] Double Map Mode Switching of Cluster 1 0: Bank D is mapped in valid area. 1: Bank C is mapped in valid area. [For U2A8/U2A6] Set the value of valid area at the shipping.
0	DBMAPSW0	Double Map Mode Switching of Cluster 0 0: Bank B is mapped in valid area. 1: Bank A is mapped in valid area.

51.12.20 OPBT14 — Option Byte 14

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03D8_H$

Value at the shipping: 0000 0002_H (valid area) (U2A-EVA), 0000 0192_H (valid area) (U2A16/U2A8/U2A6)
Erased (invalid area)

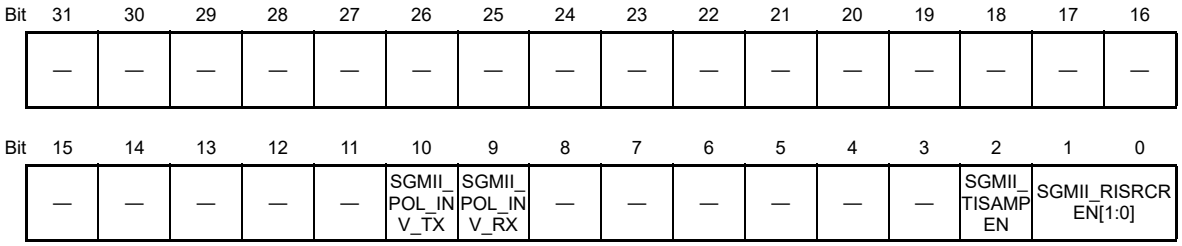


Table 51.69 OPBT14 Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	Set the value of valid area at the shipping.
10	SGMII_POL_INV_TX	SGMII transmitter polarity inversion 0: Polarity of TX_DATAP/TX_DATAN pin is not inverted. 1: Polarity of TX_DATAP/TX_DATAN pin is inverted. This bit is valid for U2A16/U2A8 only. For U2A-EVA, the polarity cannot be inverted. [For U2A-EVA/U2A6] Arbitrary value can be specified.
9	SGMII_POL_INV_RX	SGMII receiver polarity inversion 0: Polarity of RX_DATAP/RX_DATAN pin is not inverted. 1: Polarity of RX_DATAP/RX_DATAN pin is inverted. This bit is valid for U2A16/U2A8 only. For U2A-EVA, the polarity cannot be inverted. [For U2A-EVA/U2A6] Arbitrary value can be specified.
8 to 3	Reserved	[For U2A-EVA/U2A6] Arbitrary value can be specified. [For U2A16/U2A8] Set the value of valid area at the shipping.
2	SGMII_TISAMPEN	SGMII output amplitude control 0: 300mVpp (600mVppd) 1: 600mVpp (1200mVppd) This bit is valid for U2A-EVA only. For U2A16/U2A8, the amplitude is fixed to 300mVpp (600mVppd). [For U2A16/U2A8/U2A6] Arbitrary value can be specified.
1, 0	SGMII_RISRCREN [1:0]	SGMII receiver termination selectors $0\ 0_B$: Hi-Z $0\ 1_B$: Setting prohibited $1\ 0_B$: 50ohm + 50ohm between RIDP and RIDN (DC coupled) $1\ 1_B$: 50ohm to Vcm (AC coupled with IEEE802.3-2015 compliant capacitor) These bits are valid for U2A-EVA/U2A16/U2A8 only. [For U2A6] Arbitrary value can be specified.

51.12.21 OPBT16 — Option Byte 16

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03E0_H$

Value at the shipping: $6FFF\ FFFF_H$ (valid area)
Erased (invalid area)

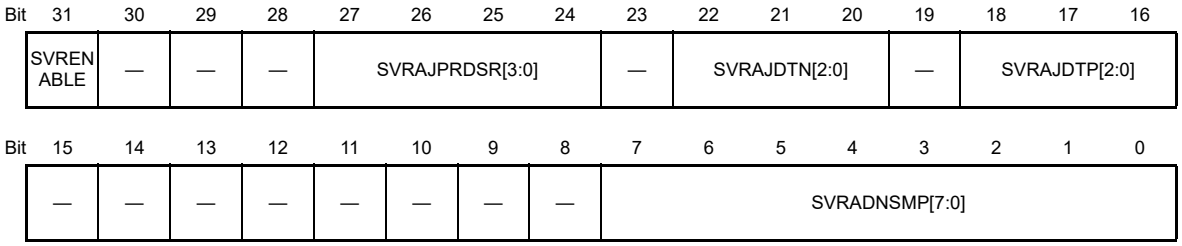


Table 51.70 OPBT16 Contents

Bit Position	Bit Name	Function
31	SVRENABLE	SVR Enable setting. 0: Disabled (default) 1: Enabled **CAUTION** Make sure that the all SVR parameters to be set to OPBT16-23 are correct before enabling SVR. Otherwise, the output voltage of Power MOSFET may be unintentional value.
30 to 28	Reserved	Set the value of valid area at the shipping.
27 to 24	SVRAJPRDSR[3:0]	SVR Adjusting slew rate (drive ability).
23	Reserved	Set the value of valid area at the shipping.
22 to 20	SVRAJDTN[2:0]	SVR Adjusting dead time of Lo-side:OFF->Hi-side:ON.
19	Reserved	Set the value of valid area at the shipping.
18 to 16	SVRAJDTP[2:0]	SVR Adjusting dead time of Hi-side:OFF->Lo-side:ON.
15 to 8	Reserved	Set the value of valid area at the shipping.
7 to 0	SVRADNSMP[7:0]	SVR ADC Sampling time.

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.22 OPBT17 — Option Byte 17

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03E4_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

Table 51.71 OPBT17 Contents

Bit Position	Bit Name	Function
31 to 24	SVRADTHRESHE[7:0]	SVR ADC conversion completion state setting.
23 to 16	SVRADTHRESH[7:0]	SVR ADC sampling start state setting.
15 to 8	SVRMAXDUTY[7:0]	SVR Maximum on-duty setting.
7, 6	Reserved	Set the value of valid area at the shipping.
5, 4	SVRMINSKIPDUTY[1:0]	SVR Skip minimum pulse setting.
3, 2	Reserved	Set the value of valid area at the shipping.
1, 0	SVRFSWMODE[1:0]	SVR Switching frequency setting.

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.23 OPBT18 — Option Byte 18

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03E8_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

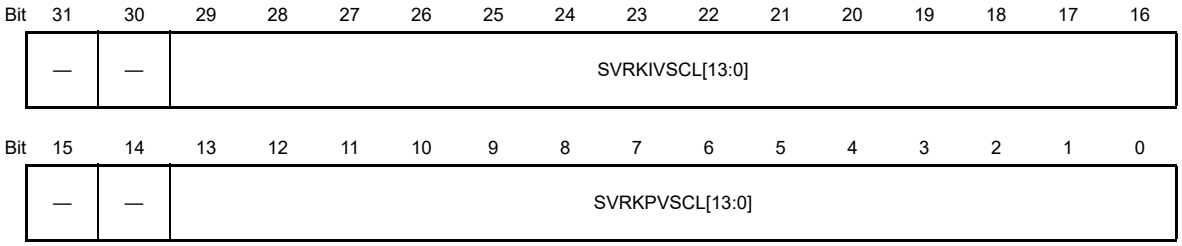


Table 51.72 OPBT18 Contents

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29 to 16	SVRKIVSCL[13:0]	SVR Scaled KI value of VPID.
15, 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKPVSCL[13:0]	SVR Scaled KP value of VPID.

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.24 OPBT19 — Option Byte 19

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03EC_H$

Value at the shipping: $FFFF\ FFFF_H$ (valid area)
Erased (invalid area)

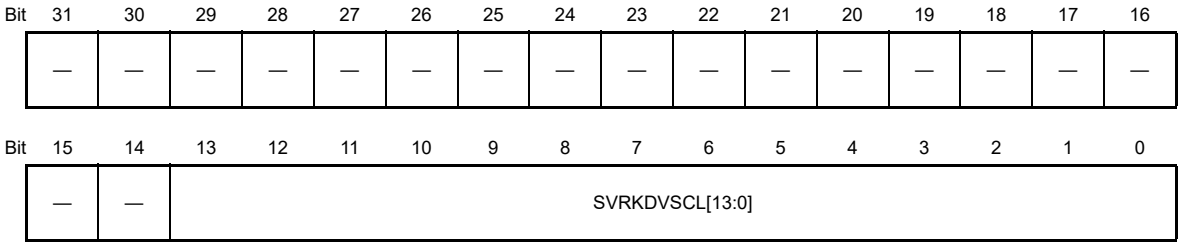


Table 51.73 OPBT19 Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKDVSCL[13:0]	SVR Scaled KD value of VPID.

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.25 OPBT20 — Option Byte 20

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03F0_H$

Value at the shipping: F088 $FF00_H$ (valid area)
Erased (invalid area)

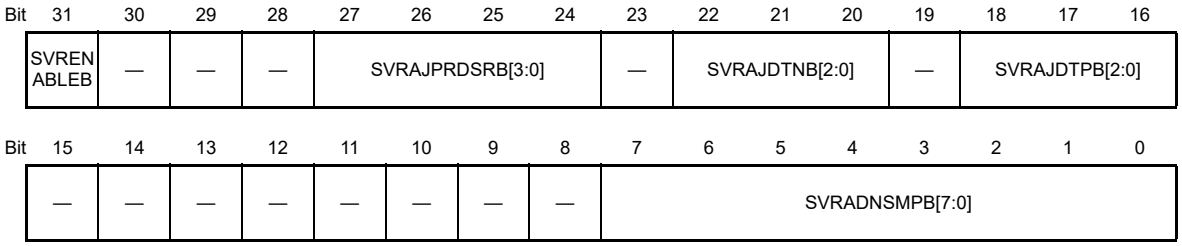


Table 51.74 OPBT20 Contents

Bit Position	Bit Name	Function
31	SVRENABLEB	Program inverted value of SVRENABLE.
30 to 28	Reserved	Set the value of valid area at the shipping.
27 to 24	SVRAJPRDSRB[3:0]	Program inverted value of SVRAJPRDSR[3:0].
23	Reserved	Set the value of valid area at the shipping.
22 to 20	SVRAJDTNB[2:0]	Program inverted value of SVRAJDTN[2:0].
19	Reserved	Set the value of valid area at the shipping.
18 to 16	SVRAJDTPB[2:0]	Program inverted value of SVRAJDTP[2:0].
15 to 8	Reserved	Set the value of valid area at the shipping.
7 to 0	SVRADNSMPB[7:0]	Program inverted value of SVRADNSMP[7:0].

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.26 OPBT21 — Option Byte 21

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03F4_H$

Value at the shipping: 0000 $00CC_H$ (valid area)
Erased (invalid area)

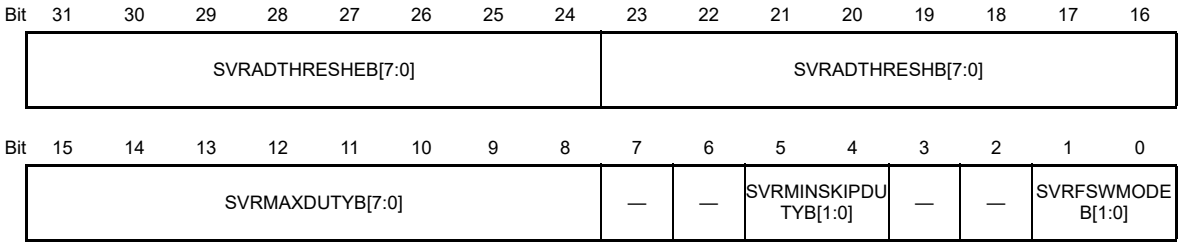


Table 51.75 OPBT21 Contents

Bit Position	Bit Name	Function
31 to 24	SVRADTHRESHEB [7:0]	Program inverted value of SVRADTHRESHE[7:0].
23 to 16	SVRADTHRESHB [7:0]	Program inverted value of SVRADTHRESH[7:0].
15 to 8	SVRMAXDUTYB[7:0]	Program inverted value of SVRMAXDUTY[7:0].
7, 6	Reserved	Set the value of valid area at the shipping.
5, 4	SVRMINSKIPDUTYB [1:0]	Program inverted value of SVRMINSKIPDUTY[1:0].
3, 2	Reserved	Set the value of valid area at the shipping.
1, 0	SVRFSWMODEB [1:0]	Program inverted value of SVRFSWMODE[1:0].

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.27 OPBT22 — Option Byte 22

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03F8_H$

Value at the shipping: $C000\ C000_H$ (valid area)
Erased (invalid area)

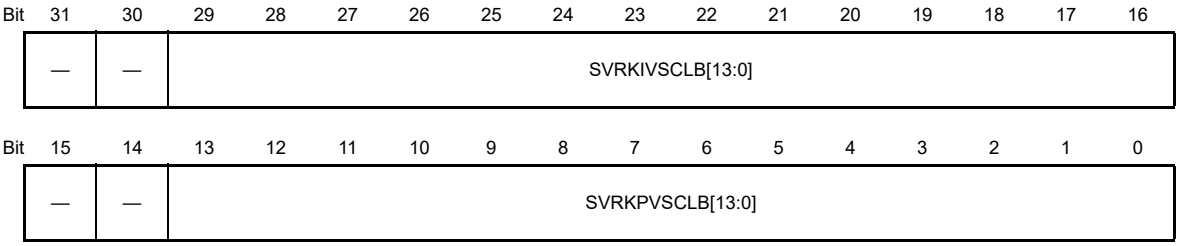


Table 51.76 OPBT22 Contents

Bit Position	Bit Name	Function
31, 30	Reserved	Set the value of valid area at the shipping.
29 to 16	SVRKIVSCLB[13:0]	Program inverted value of SVRKIVSCL[13:0].
15, 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKPVSCLB[13:0]	Program inverted value of SVRKPVSCL[13:0].

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.28 OPBT23 — Option Byte 23

Customer ID A authentication is necessary to program these data. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary to program these data.

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware*.

Address: <CSAk_base> + $03FC_H$

Value at the shipping: $FFFF\ C000_H$ (valid area)
Erased (invalid area)

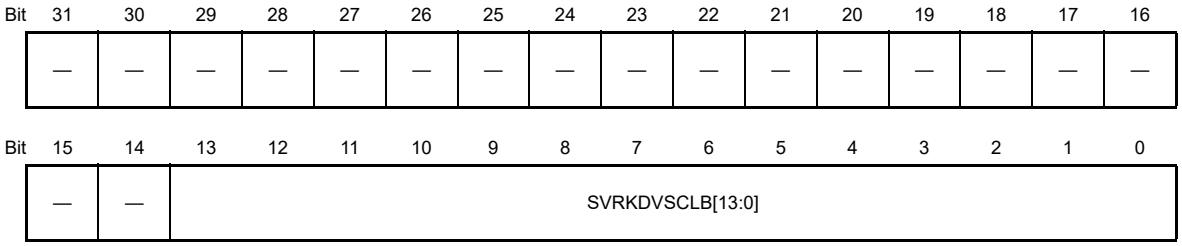


Table 51.77 OPBT23 Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	Set the value of valid area at the shipping.
13 to 0	SVRKDVSCLB[13:0]	Program inverted value of SVRKDVSCL[13:0].

For details of these data, see **Section 10, Power Supply Circuit**.

51.12.29 OTP (One Time Programmable) setting for Configuration Setting Area.

The details of this function, see **Section 47.7, Configuration Setting Area (Option Bytes, Reset Vector)**.

51.13 Switch Area

Switch Area consists of two areas (Area 0 and Area 1). Valid area is mapped in front side, Invalid area is mapped in back side. Valid area cannot be updated whereas Invalid area can be updated.

Area 0 is selected as valid area at the shipping from RENESAS.

Area 1 is selected as invalid area at the shipping from RENESAS.

Changes in settings become effective after TAG update without reset. Valid / Invalid area switching for read access also becomes effective without reset.

Table 51.78 and **Table 51.79** show Switch Area. For setting and reading these areas, see the *Renesas Flash Programmer Flash Programming Software User's Manual*, or the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 51.78 Switch Area in case of Area 0 is valid (FSWASTAT_0.SWVA = 0) (1/2)

Area	Name	Address	State at the Shipping*1	Write/Erase Protection ID*2	Read Protection ID*3
Area 0	A0PC (Area 0 Program Complete Flag)	<SWAf_base>+ 0000_H	5AA5 A55A$_H$	Can not Write	—
Area 0	A1ES (Area 1 Erase Start Flag)	<SWAf_base>+ 0004_H	Erased	Can not Write	—
Area 0	A1EC (Area 1 Erase Complete Flag)	<SWAf_base>+ 0008_H	Erased	Can not Write	—
Area 0	Reserved*6	<SWAf_base>+ ($000C_H$ to $001C_H$)	Erased	Can not Write	—
Area 0	TES (TAG Erase Start Flag)	<SWAf_base>+ 0020_H	5AA5 A55A$_H$	Can not Write	—
Area 0	TEC (TAG Erase Complete Flag)	<SWAf_base>+ 0024_H	5AA5 A55A$_H$	Can not Write	—
Area 0	Reserved*6	<SWAf_base>+ (0028_H to $003C_H$)	Erased	Can not Write	—
Area 0	Reserved*6	<SWAf_base>+ (0040_H to $01FC_H$)	Erased	Can not Write	—
Area 0	CVA (Configuration setting Valid Area)	<SWAf_base>+ 0200_H	A55A 5AA5$_H$	Can not Write	—
Area 0	SVA (Security setting Valid Area)	<SWAf_base>+ 0204_H	A55A 5AA5$_H$	Can not Write	—
Area 0	BVA0 (Block protection setting Valid Area for FPSYS0)	<SWAf_base>+ 0208_H	A55A 5AA5$_H$	Can not Write	—
Area 0	BVA1 (Block protection setting Valid Area for FPSYS1)	<SWAf_base>+ $020C_H$	A55A 5AA5$_H$	Can not Write	—
Area 0	Reserved	<SWAf_base>+ (0210_H to $021C_H$)	All FFFF FFFF$_H$	Can not Write	—
Area 0	Reserved*6	<SWAf_base>+ (0220_H to $07FC_H$)	Erased	Can not Write	—
Area 1	A1PC (Area 1 Program Complete Flag) *5	<SWAb_base>+ 0000_H	5AA5 A55A$_H$	Can not Write	—
Area 1	A0ES (Area 0 Erase Start Flag)*5	<SWAb_base>+ 0004_H	5AA5 A55A$_H$	Can not Write	—
Area 1	A0EC (Area 0 Erase Complete Flag)*5	<SWAb_base>+ 0008_H	5AA5 A55A$_H$	Can not Write	
Area 1	Reserved *6	<SWAb_base>+ ($000C_H$ to $001C_H$)	Erased	Can not Write	—
Area 1	TES (TAG Erase Start Flag)*5	<SWAb_base>+ 0020_H	5AA5 A55A$_H$	Can not Write	—
Area 1	TEC (TAG Erase Complete Flag)*5	<SWAb_base>+ 0024_H	5AA5 A55A$_H$	Can not Write	—
Area 1	Reserved*6	<SWAb_base>+ (0028_H to $003C_H$)	Erased	Can not Write	—
Area 1	Reserved*6	<SWAb_base>+ (0040_H to $01FC_H$)	Erased	Can not Write	—
Area 1	CVA (Configuration setting Valid Area)	<SWAb_base>+ 0200_H	A55A 5AA5$_H$	Customer ID A	—
Area 1	SVA (Security setting Valid Area)	<SWAb_base>+ 0204_H	A55A 5AA5$_H$	Customer ID A	—
Area 1	BVA0 (Block protection setting Valid Area for FPSYS0)	<SWAb_base>+ 0208_H	A55A 5AA5$_H$	Customer ID A	—

Table 51.78 Switch Area in case of Area 0 is valid (FSWASTAT_0.SWVA = 0) (2/2)

Area	Name	Address	State at the Shipping*1	Write/Erase Protection ID*2	Read Protection ID*3
Area 1	BVA1 (Block protection setting Valid Area for FPSYS1)	<SWAb_base>+ $020C_H$	A55A 5AA5$_H$	Customer ID A	—
Area 1	Reserved (need to be programmed)*4	<SWAb_base>+ (0210_H to $021C_H$)	All FFFF FFFF$_H$	Customer ID A	—
Area 1	Reserved*6	<SWAb_base>+ (0220_H to $07FC_H$)	Erased	Can not Write	—

Table 51.79 Switch Area in case of Area 1 is valid (FSWASTAT_0.SWVA = 1) (1/2)

Area	Name	Address	State at the Shipping*1	Write/Erase Protection ID*2	Read Protection ID*3
Area 1	A1PC (Area 0 Program Complete Flag)	<SWAf_base>+ 0000_H	—	Can not Write	—
Area 1	A0ES (Area 1 Erase Start Flag)	<SWAf_base>+ 0004_H	—	Can not Write	—
Area 1	A0EC (Area 1 Erase Complete Flag)	<SWAf_base>+ 0008_H	—	Can not Write	
Area 1	Reserved*6	<SWAf_base>+ ($000C_H$ to $001C_H$)	—	Can not Write	—
Area 1	TES (TAG Erase Start Flag)	<SWAf_base>+ 0020_H	—	Can not Write	—
Area 1	TEC (TAG Erase Complete Flag)	<SWAf_base>+ 0024_H	—	Can not Write	—
Area 1	Reserved*6	<SWAf_base>+ (0028_H to $003C_H$)	—	Can not Write	—
Area 1	Reserved*6	<SWAf_base>+ (0040_H to $01FC_H$)	—	Can not Write	—
Area 1	CVA (Configuration setting Valid Area)	<SWAf_base>+ 0200_H	—	Can not Write	—
Area 1	SVA (Security setting Valid Area)	<SWAf_base>+ 0204_H	—	Can not Write	—
Area 1	BVA0 (Block protection setting Valid Area for FPSYS0)	<SWAf_base>+ 0208_H	—	Can not Write	—
Area 1	BVA1 (Block protection setting Valid Area for FPSYS1)	<SWAf_base>+ $020C_H$	—	Can not Write	—
Area 1	Reserved	<SWAf_base>+ (0210_H to $021C_H$)	—	Can not Write	—
Area 1	Reserved*6	<SWAf_base>+ (0220_H to $07FC_H$)	—	Can not Write	—
Area 0	A0PC (Area 1 Program Complete Flag)*5	<SWAb_base>+ 0000_H	—	Can not Write	—
Area 0	A1ES (Area 0 Erase Start Flag)*5	<SWAb_base>+ 0004_H	—	Can not Write	—
Area 0	A1EC (Area 0 Erase Complete Flag)*5	<SWAb_base>+ 0008_H	—	Can not Write	
Area 0	Reserved*6	<SWAb_base>+ ($000C_H$ to $001C_H$)	—	Can not Write	—
Area 0	TES (TAG Erase Start Flag)*5	<SWAb_base>+ 0020_H	—	Can not Write	—
Area 0	TEC (TAG Erase Complete Flag)*5	<SWAb_base>+ 0024_H	—	Can not Write	—
Area 0	Reserved*6	<SWAb_base>+ (0028_H to $003C_H$)	—	Can not Write	—
Area 0	Reserved*6	<SWAb_base>+ (0040_H to $01FC_H$)	—	Can not Write	—

Table 51.79 Switch Area in case of Area 1 is valid (FSWASTAT_0.SWVA = 1) (2/2)

Area	Name	Address	State at the Shipping*1	Write/Erase Protection ID*2	Read Protection ID*3
Area 0	CVA (Configuration setting Valid Area)	<SWAb_base>+ 0200_H	—	Customer ID A	—
Area 0	SVA (Security setting Valid Area)	<SWAb_base>+ 0204_H	—	Customer ID A	—
Area 0	BVA0 (Block protection setting Valid Area for FPSYS0)	<SWAb_base>+ 0208_H	—	Customer ID A	—
Area 0	BVA1 (Block protection setting Valid Area for FPSYS1)	<SWAb_base>+ $020C_H$	—	Customer ID A	—
Area 0	Reserved (need to be programmed)*4	<SWAb_base>+ (0210_H to $021C_H$)	—	Customer ID A	—
Area 0	Reserved*6	<SWAb_base>+ (0220_H to $07FC_H$)	—	Can not Write	—

Note 1. This is the value of valid area at the shipping. Invalid area is erased at the shipping.

Note 2. When all bits of Customer ID A are all “1” or all “0”, the ID authentication is not necessary for programming/erasure the data.

Note 3. When On-chip debug and S_OPBT3.CFRPF is “0”, the Customer ID A authentication is needed to read.

Note 4. For this reserved area, program FFFF $FFFF_H$ when programming invalid area of Switch Area.

Note 5. This area will be updated automatically when programming invalid area of Switch Area.

Note 6. This area cannot be programmed. When this area is read, it behaves as if it is erased.

51.13.1 AnPC — Area n Program Complete Flag (n = 0, 1)

This flag indicates that the Switch Area n is programmed or not.
This flag is prepared to check whether the Switch Area n is programmed or not by user program.
It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when programming Switch Area. The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: FSWASTAT_0.SWVA = 0 (Area 0 is valid)
A0PC : <SWAf_base> + 0000$_H$
A1PC : <SWAb_base> + 0000$_H$
FSWASTAT_0.SWVA = 1 (Area 1 is valid)
A1PC : <SWAf_base> + 0000$_H$
A0PC : <SWAb_base> + 0000$_H$

Value at the shipping: 5AA5 A55A$_H$ (valid area)
5AA5 A55A$_H$ (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AnPC[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AnPC[15:0]															

Table 51.80 AnPC Contents

Bit Position	Bit Name	Function
31 to 0	AnPC[31:0]	Program completion flag. 5AA5 A55A$_H$: The Switch Area n is programmed. Other than above: The Switch Area n is not programmed.

51.13.2 AnES — Area n Erase Start Flag (n = 0, 1)

This flag indicates that the Erasure of the other Switch Area is started or not.
Note that Switch Area 0 contains A1ES. Whereas Switch Area 1 contains A0ES.

This flag is prepared to check whether the Switch Area Status is dirty or not by user program.

It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when Erasure of the other Switch Area is started. The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: FSWASTAT_0.SWVA = 0 (Area 0 is valid)
A1ES : <SWAf_base> + 0004$_H$
A0ES : <SWAb_base> + 0004$_H$
FSWASTAT_0.SWVA = 1 (Area 1 is valid)
A0ES : <SWAf_base> + 0004$_H$
A1ES : <SWAb_base> + 0004$_H$

Value at the shipping: Erased (valid area)
5AA5 A55A$_H$ (invalid area)

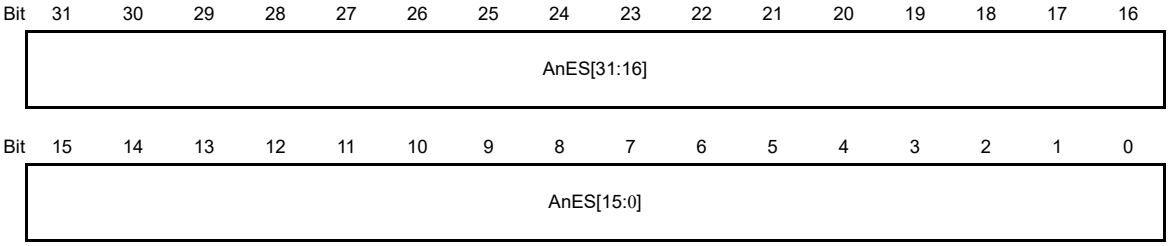


Table 51.81 AnES Contents

Bit Position	Bit Name	Function
31 to 0	AnES[31:0]	Erase Start flag of the other Switch Area. 5AA5 A55A$_H$: The erasure of the other Switch Area is started. Other than above: The erasure of the other Switch Area is not started.

51.13.3 AnEC — Area n Erase Complete Flag (n = 0, 1)

This flag indicates that the Erasure of the other Switch Area is completed or not.
Note that Switch Area 0 contains A1EC. Whereas Switch Area 1 contains A0EC.

This flag is prepared to check whether the Switch Area Status is dirty or not by user program.

It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when Erasure of the other Switch Area is completed. The updated value will be 5AA5 $A55A_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: FSWASTAT_0.SWVA = 0 (Area 0 is valid)
A1EC : <SWAf_base> + 0008_H
A0EC : <SWAb_base> + 0008_H
FSWASTAT_0.SWVA = 1 (Area 1 is valid)
A0EC : <SWAf_base> + 0008_H
A1EC : <SWAb_base> + 0008_H

Value at the shipping: Erased (valid area)
5AA5 $A55A_H$ (invalid area)

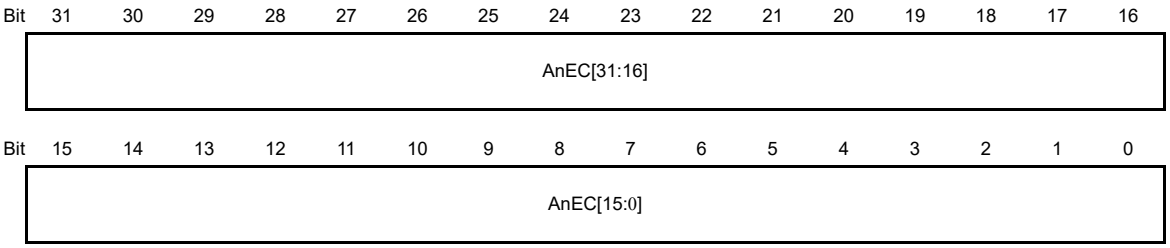


Table 51.82 AnEC Contents

Bit Position	Bit Name	Function
31 to 0	AnEC[31:0]	Erase Complete flag of the other Switch Area. 5AA5 $A55A_H$: The erasure of the other Switch Area is completed. Other than above: The erasure of the other Switch Area is not completed.

51.13.4 TES — TAG Erase Start Flag

This flag indicates that the Erasure of Tag Area is started or not.

This flag is prepared to check whether the Switch Area Status is dirty or not by user program.

It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when Erasure of TAG Area is started. The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SWAf_base> + 0020$_H$ (valid area)
<SWAb_base> + 0020$_H$ (invalid area)

Value at the shipping: 5AA5 A55A$_H$ (valid area)
5AA5 A55A$_H$ (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TES[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TES[15:0]															

Table 51.83 TES Contents

Bit Position	Bit Name	Function
31 to 0	TES[31:0]	TAG Erase Start flag. 5AA5 A55A$_H$: The erasure of TAG Area is started. Other than above: The erasure of TAG Area is not started.

51.13.5 TEC — TAG Erase Complete Flag

This flag indicates that the Erasure of TAG Area is completed or not.

This flag is prepared to check whether the Switch Area Status is dirty or not by user program.

It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when Erasure of TAG Area is completed. The updated value will be 5AA5 A55A$_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SWAf_base> + 0024$_H$ (valid area)
<SWAb_base> + 0024$_H$ (invalid area)

Value at the shipping: 5AA5 A55A$_H$ (valid area)
5AA5 A55A$_H$ (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC[15:0]															

Table 51.84 TEC Contents

Bit Position	Bit Name	Function
31 to 0	TES[31:0]	TAG Erase Complete flag. 5AA5 A55A$_H$: The erasure of TAG Area is completed. Other than above: The erasure of TAG Area is not completed.

51.13.6 CVA — Configuration Setting Valid Area Flag

This flag indicates the Valid Area of Configuration Setting Area.
For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SWAf_base> + 0200_H (valid area)
<SWAb_base> + 0200_H (invalid area)

Value at the shipping: A55A $5AA5_H$ (valid area)
A55A $5AA5_H$ (invalid area)

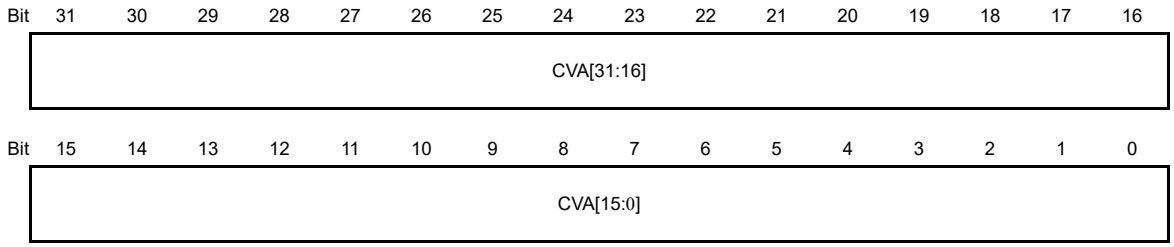


Table 51.85 CVA Contents

Bit Position	Bit Name	Function
31 to 0	CVA[31:0]	Configuration Setting Valid Area Flag 5AA5 $A55A_H$: Configuration Setting Area 1 is valid. A55A $5AA5_H$: Configuration Setting Area 0 is valid. Other than above : Setting prohibited. Valid Area of Configuration Setting Area is shown in FSWASTAT_0.CFGVA. For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

51.13.7 SVA — Security Setting Valid Area Flag

This flag indicates the Valid Area of Security Setting Area.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SWAf_base> + 0204_H (valid area)
<SWAb_base> + 0204_H (invalid area)

Value at the shipping: A55A $5AA5_H$ (valid area)
A55A $5AA5_H$ (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVA[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVA[15:0]															

Table 51.86 SVA Contents

Bit Position	Bit Name	Function
31 to 0	SVA[31:0]	Security Setting Valid Area Flag 5AA5 $A55A_H$: Security Setting Area 1 is valid. A55A $5AA5_H$: Security Setting Area 0 is valid. Other than above : Setting prohibited. Valid Area of Security Setting Area is shown in FSWASTAT_0.SECVA. For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

51.13.8 BVAn — Block Protection for FPSYSn Valid Area Flag (n = 0, 1)

This flag indicates the Valid Area of Block Protection Area for FPSYSn.
For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <SWAf_base> + $0208_H + 04_H$ x n (valid area)
<SWAb_base> + 0208_H + 04_H x n (invalid area)

Value at the shipping: A55A $5AA5_H$ (valid area)
A55A $5AA5_H$ (invalid area)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BVAn[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAn[15:0]															

Table 51.87 BVAn Contents

Bit Position	Bit Name	Function
31 to 0	BVAn[31:0]	Block Protection for FPSYSn Valid Area Flag 5AA5 $A55A_H$: Block Protection Area 1 for FPSYSn is valid. A55A $5AA5_H$: Block Protection Area 0 for FPSYSn is valid. Other than above: Setting prohibited. Valid Area of Block Protection Area for FPSYSn are shown in FSWASTAT_0.BPVAn. For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

51.14 TAG Area

TAG Area is used to specify the valid area of Switch Area.

Changes in settings become effective after TAG Update without reset. Valid/Invalid area switching for read access also becomes effective without reset.

Table 51.88 shows TAG Area. For setting and reading this area, see the *Renesas Flash Programmer Flash Programming Software User's Manual*, or the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Table 51.88 TAG Area

Name	Address	State at the shipping*1	Write/Erase Protection ID*2	Read Protection ID*3
VAPC (VAF Program Complete Flag) *4	<TAG_base>+ 0000_H	5AA5 A55A$_H$	Can not Write	—
Reserved*5	<TAG_base>+ (0004_H to $01FC_H$)	Erased	Can not Write	—
VAF (Valid Area Flag)*4	<TAG_base>+ 0200_H	A55A 5AA5$_H$	Customer ID A	—
Reserved*5	<TAG_base>+ (0204_H to $07FC_H$)	Erased	Can not Write	—

Note 1. This is the value of valid area at the shipping.

Note 2. When all bits of Customer ID A are all "1" or all "0", the ID authentication is not necessary for programming/erasure the data.

Note 3. When On-chip debug and S_OPBT3.CFRPF is "0", the Customer ID A authentication is needed to read.

Note 4. This area will be updated automatically by TAG update command.

Note 5. This area cannot be programmed. When this area is read, it behaves as if it is erased.

51.14.1 VAPC — VAF Program Complete Flag

This flag indicates that the VAF is programmed or not.
This flag is prepared to check whether the Switch Area Status is dirty or not by user program.
It is also referred by flash sequencer in order to check whether Switch Area Status is valid or dirty.

This flag will be updated automatically when VAF is updated. The updated value will be 5AA5 $A55A_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.
Address: <TAG_base> + 0000_H
Value at the shipping: 5AA5 $A55A_H$

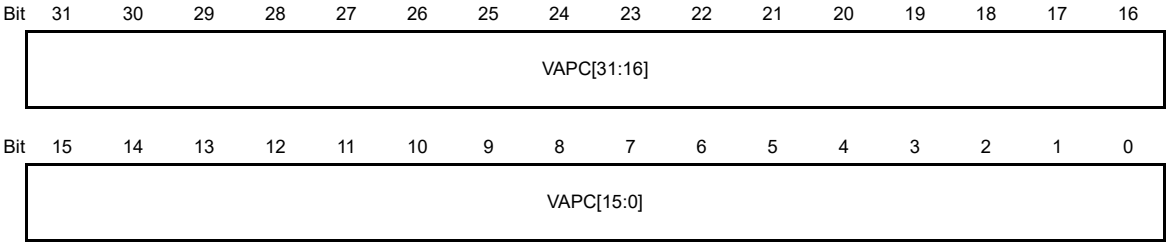


Table 51.89 VAPC Contents

Bit Position	Bit Name	Function
31 to 0	VAPC[31:0]	VAF Program completion flag. 5AA5 $A55A_H$: VAF is programmed. Other than above: VAF is not programmed.

51.14.2 VAF —Valid Area Flag of Switch Area

This flag indicates the Valid Area of Switch Area.

This flag will be updated automatically by TAG Update command. The updated value will be 5AA5 $A55A_H$ or A55A $5AA5_H$.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

Access: For setting these data, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

Address: <TAG_base> + 0200_H

Value at the shipping: A55A $5AA5_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VAF[31:16]															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VAF[15:0]															

Table 51.90 VAF Contents

Bit Position	Bit Name	Function
31 to 0	VAF[31:0]	Valid Area Flag of Switch area 5AA5 A55AH: Switch Area 1 is valid. A55A 5AA5H: Switch Area 0 is valid. Other than above: Switch Area Status is Dirty. Valid Area of Switch Area is shown in FSWASTAT_0.SWVA. For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware.*

51.15 Erase Counter Area

Each block of User Area has 16-bit Erase Counter.

Maximum count of Erase Counter is $03E9_H$ (1001), and the counter value is not increased if an “Erase” command is issued when an Erase Counter is $03E9_H$.

Erase Counter Area for User Area 0 and User Boot Area 0 are located in FF32 5000_H to FF32 $67FF_H$.

Erase Counter Area for User Area 1 and User Boot Area 1 are located in FF32 6800_H to FF32 $7FFF_H$.

Erase Counter Area for User Area 2 is located in FF34 1000_H to FF34 $27FF_H$.

Erase Counter Area for User Area 3 is located in FF34 2800_H to FF34 $3FFF_H$.

When an “Erase” command is issued to User Area or User Boot Area, Erase Counter Area is updated first, after that code flash is erased.

Table 51.91 and **Table 51.92** show mapping of Erase Counter Area. Un_m indicates block number of User Area. For example, U0_1 indicates block 1 of User Area 0. U0_uba indicates User Boot Area 0. When Area 0/1 Valid Flag for Erase Counter equal to 5AA5 $A55A_H$, Area 1 is valid. When Area 0/1 Valid Flag for Erase Counter is other values, Area 0 is valid.

The offset of Area 0 and Area 1 are fixed (not flipped by Hardware). It is necessary to check the Area 0/1 Valid Flag for Erase Counter to judge the valid area.

For details of this function, see the *RH850/U2A-EVA Group Flash Memory User’s Manual: Hardware.*

Table 51.91 Erase Counter Area for User Area n and User Boot Area n (n = 0, 1)

Area	Address (User Mode)	bit 31 bit16	bit15 bit0	State at the shipping	Read Protection ID*1
	<ECn_base> + 0000_H	Area 0/1 Valid Flag for Erase Counter		FFFF $FFFF_H$	—
	<ECn_base> + 0004_H to $07FC_H$	Reserved		Undefined	—
Area 0	<ECn_base> + 0800_H	Counter for Un_1	Counter for Un_0	0000 0000_H	—
	<ECn_base> + 0804_H	Counter for Un_3	Counter for Un_2	0000 0000_H	—
	…	…	…	…	…
	<ECn_base> + 0888_H	Counter for Un_69	Counter for Un_68	0000 0000_H	—
	<ECn_base> + $088C_H$ to $FF8_H$	Reserved*2		Erased	—
	<ECn_base> + FFC_H	Counter for Un_uba	Reserved	0000 $FFFF_H$	—
Area 1	<ECn_base> + 1000_H to $17FC_H$	Same as <ECn_base> + 0800_H to FFC_H (Valid area is decided by Area 0/1 Valid Flag for Erase Counter)			

Table 51.92 Erase Counter Area for User Area n (n ≠ 0, 1)

Area	Address (User Mode)	bit 31 bit16	bit15 bit0	State at the shipping	Read Protection ID*1
	<ECn_base> + 0000_H	Area 0/1 Valid Flag for Erase Counter		FFFF $FFFF_H$	—
	<ECn_base> + 0004_H to $07FC_H$	Reserved		Undefined	—
Area 0	<ECn_base> + 0800_H	Counter for Un_1	Counter for Un_0	0000 0000_H	—
	<ECn_base> + 0804_H	Counter for Un_3	Counter for Un_2	0000 0000_H	—
	…	…	…	…	…
	<ECn_base> + 0888_H	Counter for Un_69	Counter for Un_68	0000 0000_H	—
	<ECn_base> + $088C_H$ to $FF8_H$	Reserved*2		Erased	—
	<ECn_base> + FFC_H	Reserved		FFFF $FFFF_H$	—
Area 1	<ECn_base> + 1000_H to $17FC_H$	Same as <ECn_base> + 0800_H to FFC_H (Valid area is decided by Area 0/1 Valid Flag for Erase Counter)			

Note 1. When On-chip debug and S_OPBT3.CFRPF is "0", the Customer ID A authentication is needed to read.

Note 2. When this area is read, it behaves as if it is erased.

51.16 Extended Data Area

Extended Data Area is used to store any data to use by user software.

This area is 1 block of the data flash. But, protection specifications are different.

Table 51.93 Extended Data Area

Name	Address	State at the shipping.*1	Write/Erase Protection ID*2	Read Protection ID*3
Extended Data Area	FF32 0000_H to FF32 $07FF_H$	Erased	Customer ID A/B/C	—

Note 1. This is the value at the shipping.

Note 2. The ID (Customer ID A or B or C) which protects the Extended Data Area can be selected by the setting of Block Protection.
For details, see **Section 47.9, Block Protection Area for FPSYS0/FPSYS1.**
When all bits of the ID are all “1” or all “0”, the ID authentication is no needed to set.

Note 3. When On-chip debug and S_OPBT3.CFRPF is “0”, the Customer ID A authentication is needed to read.

51.17 Reset Transfer

Reading the setting of this product (such as System Configuration Parameters) from Flash Memory is executed by Power On Reset, System Reset 1, System Reset 2 and DeepSTOP Reset automatically (so called “Reset Transfer”).

Reset Transfer consists of two parts, FACI reset transfer and BIST parameter transfer.

51.17.1 FACI reset transfer

System Configuration Parameters, Security Parameters, various flags in Hardware Property Area, etc are read by FACI reset transfer and stored to System Configuration Data Storage (SCDS).

If an ECC error is detected during FACI reset transfer, the ECC error is notified to ECM.

The ECC error is categorized into two types dependent on the severity.

One is FACI reset transfer error and the other is FACI reset transfer warning.

The microcontroller will be kept in reset state and will not start up if the FACI reset transfer error is detected in Normal Operation Mode or User Boot Mode.

The microcontroller will start up even if the FACI reset transfer warning is detected.

In this case, Switch Area Status which is indicated by the FSWASTAT_0.SWAS may be dirty or the data in invalid area of Block Protection Area for FPSYS1 may be undefined. It is necessary to recover the Switch Area Status to valid in the startup program by the user. It is necessary to recover the data in invalid area of Block Protection Area for FPSYS1 to defined data (completely programmed or erased) in the application program by the user.

For details, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

51.17.2 BIST parameter transfer

BIST parameters are read by BIST parameter transfer.

If an ECC error is detected during BIST parameter transfer, the ECC error is notified to ECM (BIST parameter transfer error). In this case, the result of BIST is not reliable. See **Section 44.6, BIST** for details. The microcontroller will start up even if the BIST parameter transfer error is detected.

51.18 GCFU

NOTE

The GCFU mentioning this section is same hardware resource as debug function “Global Calibration Function Unit(GCFU)”.

So when using GCFU, it should confirm the GCFU usage for debugging and should be considered avoid confliction of resources.

For details on debugging, see the user's manual of the debugger.

51.18.1 Features of GCFU

51.18.1.1 Number of Units and Channels

This product has only one GCFU unit.

Table 51.94 Number of Units

	RH850/U2A-EVA, U2A16, U2A8, U2A6
Number of Channels	1
Name	GCFU

51.18.1.2 Register Base Address

Table 51.95 Register Base Address

Base Address Name	Base Address	Bus Group
<GCFU_base>	FFFB 1400_H	P-Bus Group 0

51.18.1.3 Clock Supply

The clocks supplied to the GCFU are listed in the following table.

Table 51.96 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
GCFU	PCLK	CLK_HBUS

51.18.1.4 Interrupt Requests and Error Notifications

No interrupt requests, no error notifications.

51.18.1.5 Reset Sources

Table 51.97 Reset Sources

Unit Name	Register Name	Reset Category						
		Power Up Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
GCFU	All registers	√	√ *1	√ *1	√*1	√*1	—	—

Note 1. When debugging tools are used, initialization of the on-chip debug registers by system reset may be protected so that debugging can be performed at system reset assertion by debug setting. Protect debugging Registers (Reset mask function).
For details on debugging, see the user's manual of the debugger.

51.18.2 Overview

51.18.2.1 Functional Overview

The GCFU is a function unit to remap a code flash memory area to code flash memory area or other memory area.

- Number of memory mapping blocks

  Up to 32 memory mapping blocks are supported.

- Block size

  Remap size of each memory mapping block is configurable from 8 KB to 1024 KB.

- Target area

  Non-secure area of User Area can be set to a source address. An address of non-secure area of User Area or other memory area can be set to a destination address. For details of non-secure area or secure area, see the *RH850/U2A-EVA Group Security User's Manual: Hardware.*

- Mapping error detection

  This function can detect mapping error and prohibit mapping when the source address mapping or destination address mapping is not correct.

NOTE

- The address is remapped whether it is instruction fetch or data access.
- The accesses from all bus masters are address conversion target.
- Access latency is changed if address of the access is remapped.
  It is increased in the case of remapping to same cluster of Code Flash.
- The access from flash memory application command interface (FACI) for flash memory programming in user's applications is out of address remap target.
- [For U2A6] It is prohibited to set address of Cluster RAM to a destination address.

- Automatic cache clear

  Each time the mapping setting is updated, a cache clear is issued by GCFU automatically for instruction cache and data buffer to avoid unintended instruction or data use.

NOTE

- Automatic cache clear effects all CPU cores except ICUMHA.
  To avoid unintended instruction or data use for ICUMHA, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

51.18.3 Registers

This section lists function registers (data registers) which control operation of the GCFU.

51.18.3.1 List of Register

GCFU registers are given as offsets from the base address <GCFU_base>.

The GCFU has the registers listed in **Table 51.98**.

Table 51.98 List of Registers (1/3)

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GCFU	Memory Mapping Enable Register	TM_ME	<GCFU_base> +0010$_H$	32	PBG0#1	—
GCFU	Memory Mapping Status Register	TM_MS	<GCFU_base> +0014$_H$	32	PBG0#1	—
GCFU	Memory Mapping Error Register	TM_ERR	<GCFU_base> +0024$_H$	32	PBG0#1	—
GCFU	Memory Mapping Size Configuration Register 0	TM_SZC0	<GCFU_base> +0030$_H$	32	PBG0#1	—
GCFU	Memory Mapping Size Configuration Register 1	TM_SZC1	<GCFU_base> +0034$_H$	32	PBG0#1	—
GCFU	Memory Mapping Size Configuration Register 2	TM_SZC2	<GCFU_base> +0038$_H$	32	PBG0#1	—
GCFU	Memory Mapping Size Configuration Register 3	TM_SZC3	<GCFU_base> +003C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 00	TM_MSA00	<GCFU_base> +0040$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 01	TM_MSA01	<GCFU_base> +0044$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 02	TM_MSA02	<GCFU_base> +0048$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 03	TM_MSA03	<GCFU_base> +004C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 04	TM_MSA04	<GCFU_base> +0050$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 05	TM_MSA05	<GCFU_base> +0054$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 06	TM_MSA06	<GCFU_base> +0058$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 07	TM_MSA07	<GCFU_base> +005C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 08	TM_MSA08	<GCFU_base> +0060$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 09	TM_MSA09	<GCFU_base> +0064$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 10	TM_MSA10	<GCFU_base> +0068$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 11	TM_MSA11	<GCFU_base> +006C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 12	TM_MSA12	<GCFU_base> +0070$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 13	TM_MSA13	<GCFU_base> +0074$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 14	TM_MSA14	<GCFU_base> +0078$_H$	32	PBG0#1	—

Table 51.98 List of Registers (2/3)

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GCFU	Memory Mapping Source Address Register 15	TM_MSA15	<GCFU_base> +007C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 16	TM_MSA16	<GCFU_base> +0080$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 17	TM_MSA17	<GCFU_base> +0084$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 18	TM_MSA18	<GCFU_base> +0088$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 19	TM_MSA19	<GCFU_base> +008C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 20	TM_MSA20	<GCFU_base> +0090$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 21	TM_MSA21	<GCFU_base> +0094$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 22	TM_MSA22	<GCFU_base> +0098$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 23	TM_MSA23	<GCFU_base> +009C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 24	TM_MSA24	<GCFU_base> +00A0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 25	TM_MSA25	<GCFU_base> +00A4H	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 26	TM_MSA26	<GCFU_base> +00A8$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 27	TM_MSA27	<GCFU_base> +00AC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 28	TM_MSA28	<GCFU_base> +00B0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 29	TM_MSA29	<GCFU_base> +00B4$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 30	TM_MSA30	<GCFU_base> +00B8$_H$	32	PBG0#1	—
GCFU	Memory Mapping Source Address Register 31	TM_MSA31	<GCFU_base> +00BC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 00	TM_MDA00	<GCFU_base> +00C0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 01	TM_MDA01	<GCFU_base> +00C4$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 02	TM_MDA02	<GCFU_base> +00C8$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 03	TM_MDA03	<GCFU_base> +00CC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 04	TM_MDA04	<GCFU_base> +00D0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 05	TM_MDA05	<GCFU_base> +00D4$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 06	TM_MDA06	<GCFU_base> +00D8$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 07	TM_MDA07	<GCFU_base> +00DC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 08	TM_MDA08	<GCFU_base> +00E0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 09	TM_MDA09	<GCFU_base> +00E4$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 10	TM_MDA10	<GCFU_base> +00E8$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 11	TM_MDA11	<GCFU_base> +00EC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 12	TM_MDA12	<GCFU_base> +00F0$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 13	TM_MDA13	<GCFU_base> +00F4$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 14	TM_MDA14	<GCFU_base> +00F8H	32	PBG0#1	—

Table 51.98 List of Registers (3/3)

Unit Name	Register Name	Symbol	Address	Access Size	Access Protection	
					PBG	Other
GCFU	Memory Mapping Destination Address Register 15	TM_MDA15	<GCFU_base> +00FC$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 16	TM_MDA16	<GCFU_base> +0100$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 17	TM_MDA17	<GCFU_base> +0104$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 18	TM_MDA18	<GCFU_base> +0108$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 19	TM_MDA19	<GCFU_base> +010C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 20	TM_MDA20	<GCFU_base> +0110$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 21	TM_MDA21	<GCFU_base> +0114$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 22	TM_MDA22	<GCFU_base> +0118$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 23	TM_MDA23	<GCFU_base> +011C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 24	TM_MDA24	<GCFU_base> +0120$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 25	TM_MDA25	<GCFU_base> +0124$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 26	TM_MDA26	<GCFU_base> +0128$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 27	TM_MDA27	<GCFU_base> +012C$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 28	TM_MDA28	<GCFU_base> +0130$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 29	TM_MDA29	<GCFU_base> +0134$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 30	TM_MDA30	<GCFU_base> +0138$_H$	32	PBG0#1	—
GCFU	Memory Mapping Destination Address Register 31	TM_MDA31	<GCFU_base> +013C$_H$	32	PBG0#1	—

51.18.3.2 TM_ME — Memory Mapping Enable Register

The Memory Mapping Enable Register is used to control the address remap function by memory mapping setting.

NOTES

1. Cache clear request is executed by update of mapping enable bit (TMEn) (not including update to the same value (without changing value)) (n = 00 to 31).
2. The mapping setting of block n is disabled regardless of TMEn setting if TM_ERR.ERn is “1” (mapping error) (n = 00 to 31).
3. For details of updating register setting, see **Section 51.18.4.1, Register Setting Sequence**.

Access: This register can be read / written in 32-bit units.

Address: <GCFU_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TME31	TME30	TME29	TME28	TME27	TME26	TME25	TME24	TME23	TME22	TME21	TME20	TME19	TME18	TME17	TME16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TME15	TME14	TME13	TME12	TME11	TME10	TME09	TME08	TME07	TME06	TME05	TME04	TME03	TME02	TME01	TME00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 51.99 TM_ME Register Contents

Bit Position	Bit Name	Function
31 to 0	TME31 to TME00	Memory Mapping Block n Mapping Enable (n = 00 to 31) Requests mapping of memory mapping block n. (n = 00 to 31) 0: Disable mapping. 1: Enable mapping.

51.18.3.3 TM_MS — Memory Mapping Status Register

The Memory Mapping Status Register is used to indicate the mapping state of memory mapping blocks 00 to 31.

NOTES

1. Following TM_ME register update, be sure to wait until the read value of the TM_MS register becomes the same as the value set to the TM_ME register (until the mapping setting set to the TM_ME register is actually applied).
2. The mapping setting of block n is disabled regardless of the value shown in MESn bit if TM_ERR.ERn is “1” (mapping error) (n = 00 to 31).

Access: This register can be read in 32-bit units.

Address: <GCFU_base> + 0014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MES31	MES30	MES29	MES28	MES27	MES26	MES25	MES24	MES23	MES22	MES21	MES20	MES19	MES18	MES17	MES16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MES15	MES14	MES13	MES12	MES11	MES10	MES09	MES08	MES07	MES06	MES05	MES04	MES03	MES02	MES01	MES00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 51.100 TM_MS Register Contents

Bit Position	Bit Name	Function
31 to 0	MES31 to MES00	Memory Mapping Block n Enable Status (n = 00 to 31) Indicates the mapping state of memory mapping block n. (n = 00 to 31) 0: Mapping is disabled. 1: Mapping is enabled.

51.18.3.4 TM_ERR — Memory Mapping Error Register

The Memory Mapping Error Register is used to indicate the mapping error of TM_MSAn/TM_MDAn (n = 00 to 31).

TM_ERR register indicate the mapping error when the address in TM_MSAn (n = 00 to 31) registers or TM_MDAn (n = 00 to 31) registers are not correct. For details of mapping error condition, see **Section 51.18.4.2, Mapping Error Detection**.

GCFU forcibly disables mapping setting of the block that detects mapping error.

Access: This register can be read in 32-bit units.

Address: <GCFU_base> + 0024_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ER31	ER30	ER29	ER28	ER27	ER26	ER25	ER24	ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ER15	ER14	ER13	ER12	ER11	ER10	ER09	ER08	ER07	ER06	ER05	ER04	ER03	ER02	ER01	ER00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 51.101 TM_ERR Register Contents

Bit Position	Bit Name	Function
31 to 0	ER31 to ER00	Memory Mapping Block n Error Status (n = 00 to 31) These bits indicate "1" when the address set in the TM_MSAn register / TM_MDAn register is a mapping error. 0: No mapping error. 1: Mapping error.

51.18.3.5 TM_SZCn — Memory Mapping Size Configuration Registers (n = 0 to 3)

The Memory Mapping Size Configuration Registers are used to set the mapping size of memory mapping blocks 00 to 31.

NOTES

1. Setting mapping sizes that exceed the implementation size is prohibited (the operation is not guaranteed).
2. Each entry (blocks 00 to 31) can be mapped only in mapping size alignment units.
3. Before updating the memory mapping size configuration bits MMSZn, be sure to set the corresponding mapping enable bit TMEn to 0 (mapping disabled) (n = 00 to 31).
4. For details of updating register setting, see **Section 51.18.4.1, Register Setting Sequence**.

Access: This register can be read / written in 32-bit units.

Address: <GCFU_base> + 0030_H + n × 4_H (n = 0 to 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMSZ(8n+7)				MMSZ(8n+6)				MMSZ(8n+5)				MMSZ(8n+4)			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MMSZ(8n+3)				MMSZ(8n+2)				MMSZ(8n+1)				MMSZ(8n+0)			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 51.102 TM_SZCn Register Contents

Bit Position	Bit Name	Function
31 to 28	MMSZ (8n+7)	Memory Mapping Block 8n+7 Size Configuration (n = 0 to 3)
27 to 24	MMSZ (8n+6)	Memory Mapping Block 8n+6 Size Configuration (n = 0 to 3)
23 to 20	MMSZ (8n+5)	Memory Mapping Block 8n+5 Size Configuration (n = 0 to 3)
19 to 16	MMSZ (8n+4)	Memory Mapping Block 8n+4 Size Configuration (n = 0 to 3)
15 to 12	MMSZ (8n+3)	Memory Mapping Block 8n+3 Size Configuration (n = 0 to 3)
11 to 8	MMSZ (8n+2)	Memory Mapping Block 8n+2 Size Configuration (n = 0 to 3)
7 to 4	MMSZ (8n+1)	Memory Mapping Block 8n+1 Size Configuration (n = 0 to 3)
3 to 0	MMSZ (8n+0)	Memory Mapping Block 8n+0 Size Configuration (n = 0 to 3)

Set the mapping size of each memory mapping block.

Value	Block Size	Value	Block Size
0_H	Reserved (setting prohibited)	8_H	Reserved (setting prohibited)
1_H	1024 Kbytes	9_H	8 Kbytes
2_H	Reserved (setting prohibited)	A_H	16 Kbytes
3_H	Reserved (setting prohibited)	B_H	32 Kbytes
4_H	Reserved (setting prohibited)	C_H	64 Kbytes
5_H	Reserved (setting prohibited)	D_H	128 Kbytes
6_H	Reserved (setting prohibited)	E_H	256 Kbytes
7_H	Reserved (setting prohibited)	F_H	512 Kbytes

51.18.3.6 TM_MSAn — Memory Mapping Source Address Registers (n = 00 to 31)

Memory Mapping Source Address Registers 00 to 31 are used to set the addresses of User Area.

Any address in non-secure area of User Area (up to 32 areas) can be set. For details of the address range which can be set to this register, see **Section 51.18.4.2, Mapping Error Detection**.

The address specified with memory mapping source address register n (TM_MSAn) is remapped with the address specified with memory mapping destination address register n (TM_MDAn).

The address range to be mapped can be specified from 8 KB/block to 1024KB/block.

Up to 32 blocks can be specified.

NOTES

1. Before updating Memory Mapping Source Address Register n (TM_MSAn register), be sure to set the corresponding mapping enable bit TMEn to 0 (mapping disabled). (n = 00 to 31)
2. Do not overlap mapping setting (source area) of the mapping enabled blocks. The operation is not guaranteed if mapping setting of the mapping enabled blocks are overlapped.
3. For details of updating register setting, see **Section 51.18.4.1, Register Setting Sequence**.

Access: This register can be read / written in 32-bit units.

Address: <GCFU_base> + 0040_H + n × 4_H (n = 00 to 31)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMSAn[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MMSAn[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 51.103 TM_MSAn Register Contents

Bit Position	Bit Name	Function
31 to 0	MMSAn[31:0]	Memory Mapping Source Address n (n = 00 to 31) These bits specify the address in User Area for memory mapping block n. **NOTES** 1. For the mapping address, specify an address inside the non-secure area of User Area. 2. TM_MSAn31 to 26 and 12 to 0 are fixed to 0. 3. The address bits for mapping differs according to the value of the mapping size setting (MMSZn bit). See **Table 51.104, Mapping Size and Mapping Source Address bits** for details.

Table 51.104 Mapping Size and Mapping Source Address bits

MMSZn	Mapping Size	Valid Bits	Ignored Bits
0_H	Reserved (setting prohibited)	—	—
1_H	1024 KB	MMSAn 25 to 20	MMSAn 31 to 26, 19 to 0
2_H to 8_H	Reserved (setting prohibited)	—	—
9_H	8 KB	MMSAn 25 to 13	MMSAn 31 to 26, 12 to 0
A_H	16 KB	MMSAn 25 to 14	MMSAn 31 to 26, 13 to 0
B_H	32 KB	MMSAn 25 to 15	MMSAn 31 to 26, 14 to 0
C_H	64 KB	MMSAn 25 to 16	MMSAn 31 to 26, 15 to 0
D_H	128 KB	MMSAn 25 to 17	MMSAn 31 to 26, 16 to 0
E_H	256 KB	MMSAn 25 to 18	MMSAn 31 to 26, 17 to 0
F_H	512 KB	MMSAn 25 to 19	MMSAn 31 to 26, 18 to 0

51.18.3.7 TM_MDAn — Memory Mapping Destination Address Registers (n = 00 to 31)

Memory Mapping Destination Address Registers 00 to 31 are used to set the addresses of memory area.

Any address in non-secure area of User Area (up to 32 areas) can be set. Address of other memory area also can be set.

For details of the address range which can be set to this register, see **Section 51.18.4.2, Mapping Error Detection**.

The address specified with memory mapping source address register n (TM_MSAn) is remapped with the address specified with memory mapping destination address register n (TM_MDAn). The address range to be mapped can be specified from 8 KB/block to 1024 KB/block. Up to 32 blocks can be specified.

NOTES

1. Before updating Memory Mapping Destination Address Register n (TM_MDAn register), be sure to set the corresponding mapping enable bit TMEn to 0 (mapping disabled). (n = 00 to 31)
2. For details of updating register setting, see **Section 51.18.4.1, Register Setting Sequence**.
3. [For U2A6] It is prohibited to set address of CRAM to TM_MDAn.

Access: This register can be read / written in 32-bit units.

Address: <GCFU_base> + $00C0_H$ + n × 4_H (n = 00 to 31)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMDAn[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MMDAn[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 51.105 TM_MDAn Register Contents

Bit Position	Bit Name	Function
31 to 0	MMDAn [31:0]	Memory Mapping Destination Address n (n = 00 to 31) These bits specify the address of memory area for memory mapping block n. **NOTES** 1. TM_MDAn12 to 0 are fixed to 0. 2. The address bits for mapping differs according to the value of the mapping size setting (MMSZn bit). See **Table 51.106, Mapping Size and Mapping Destination Address bits** for details.

Table 51.106 Mapping Size and Mapping Destination Address bits

MMSZn	Mapping Size	Valid Bits	Ignored Bits
0_H	Reserved (setting prohibited)	—	—
1_H	1024 KB	MMDAn 31 to 20	MMDAn 19 to 0
2_H to 8_H	Reserved (setting prohibited)	—	—
9_H	8 KB	MMDAn 31 to 13	MMDAn 12 to 0
A_H	16 KB	MMDAn 31 to 14	MMDAn 13 to 0
B_H	32 KB	MMDAn 31 to 15	MMDAn 14 to 0
C_H	64 KB	MMDAn 31 to 16	MMDAn 15 to 0
D_H	128 KB	MMDAn 31 to 17	MMDAn 16 to 0
E_H	256 KB	MMDAn 31 to 18	MMDAn 17 to 0
F_H	512 KB	MMDAn 31 to 19	MMDAn 18 to 0

51.18.4 Remap Function

51.18.4.1 Register Setting Sequence

The remap setting requires that the registers of GCFU be set in the appropriate sequence. During the sequence, do not access source area of the mapping block whose mapping will be changed by the sequence. It is necessary to prepare new data to memory area which will be destination of the remap in advance.

Table 51.107 Register Setting Sequence

(1)	**Mapping status checking**	
(1-1)	CPU core	Check the mapping setting of block n whose mapping is to be changed with mapping status bit MESn (if “0”, go to (3). if “1”, go to (2)). (n = 00 to 31).
(2)	**Mapping disable processing**	
(2-1)	CPU core	Set mapping enable bit TMEn of block n to “0” (mapping disabled). (n = 00 to 31).
(2-2)	GCFU *1	Request the code flash memory bus use right from GCFU upon updating of the mapping enable register (TM_ME register).
(2-3)	GCFU *1	Following acquisition of the bus use right, switch the memory mapping (to apply the setting of the mapping enable register (TM_ME register)).
(2-4)	GCFU *1	Following acquisition of the bus use right, output the cache clear request signal from GCFU to the Instruction cache and data buffer.
(2-5)	GCFU *1	Following the mapping switch/cache clear request signal output, GCFU releases the code flash memory bus use right.
(2-6) *2	CPU core	Check whether the mapping setting has been applied with mapping status bits MESn (if “0”, go to (3). if “1”, go to (2-6)). (n = 00 to 31).
(3)	**Remap address Setting**	
(3-1)	CPU core	Set mapping size configuration bits MMSZn of the block n. (n = 00 to 31).
(3-2)	CPU core	Set source address of User Area to TM_MSAn register of the block n. (n = 00 to 31).
(3-3)	CPU core	Set destination address of User Area to TM_MDAn register of the block n. (n = 00 to 31).
(3-4)	CPU core	Check error status bit ERn of block n (if “0”, go to (4). if “1”, confirm MMSZn, TM_MSAn, TM_MDAn setting and go to (3-1)). (n = 00 to 31).
(4)	**Mapping enable processing**	
(4-1)	CPU core	Set mapping enable bit TMEn of block n for which remap settings have been completed to “1” (mapping enabled).
(4-2)	GCFU *1	Request the code flash memory bus use right from GCFU upon updating of the mapping enable register (TM_ME register).
(4-3)	GCFU *1	Following acquisition of the bus use right, switch the memory mapping (to apply the setting of the mapping enable register (TM_ME register)).
(4-4)	GCFU *1	Following acquisition of the bus use right, output the cache clear request signal from GCFU to the Instruction cache and data buffer.
(4-5)	GCFU *1	Following the mapping switch/cache clear request signal output, the GCFU releases the code flash memory bus use right.
(4-6) *2	CPU core	Check whether the mapping setting has been applied with mapping status bits MESn (if “0”, go to (4-6). if “1”, end the sequence). (n = 00 to 31).

Note 1. This processing is performed by hardware.

Note 2. The processing for synchronization is also necessary. See **Section 3.9.1.2, When the Updated Results in the Control Registers and Memories are Reflected in the Instruction Fetch of a Subsequent Instruction:**.
Cache clear request of GCFU does not affect ICUMHA. To avoid unintended instruction or data use for ICUMHA, see the *RH850/U2A-EVA Group Security User’s Manual: Hardware*.

51.18.4.2 Mapping Error Detection

Mapping error is detected if the source address mapping or destination address mapping is not correct.

GCFU forcibly disables mapping setting of the block that detects mapping error.

Detail condition of Mapping Error Detection is shown in below.

Table 51.108 Mapping Error Detection

	region start address	region end address	Mapping Error Detection
TM_MSAn*2	0000 0000H	03FF FFFF$_H$	Secure area: Error is detected. Others: No Error is detected.
	0400 0000H	FFFF FFFF$_H$	Not available (Cannot set this region to TM_MSAn).
TM_MDAn*2	0000 0000H	03FF FFFF$_H$	Secure area: Error is detected. Others: No Error is detected.
	0400 0000H	FAFF FFFF$_H$	Error is detected.
	FB00 0000H	FBFF FFFF$_H$	No Error is detected.*1
	FC00 0000H	FDFF FFFF$_H$	Error is detected.
	FE00 0000H	FEFF FFFF$_H$	No Error is detected.*1
	FF00 0000H	FFFF FFFF$_H$	Error is detected.

Note 1. For details, see **Section 4, Address Space**.

Note 2. Valid Bits of Mapping Source Address and Mapping Destination Address are same between Mapping and Error Detection. See **Table 51.104, Mapping Size and Mapping Source Address bits** and **Table 51.106, Mapping Size and Mapping Destination Address bits** for details.

51.19 Usage Notes

(1) Reading areas where programming or erasure was interrupted
When programming or erasure of an area of flash memory is interrupted intentionally or unintentionally, the data stored in the area become undefined. To avoid undefined data that are read out becoming the source of faulty operation, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.

(2) Prohibition of additional writing
Writing to a given area twice is not possible. If you want to update data in an area of flash memory after writing to the area has been completed, erase the area first.

(3) Resets during programming and erasure
In the case of an external reset during programming and erasure, wait for at least width of reset pulse more than the min value by an external reset once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.

(4) Allocation of vectors for interrupts and other exceptions during programming and erasure
Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory. For how to change the address for vector fetching, see **Section 3, CPU System** and **Section 6, Interrupts**. For ICUMHA, see the *RH850/U2A-EVA Group Security User's Manual: Hardware.*

(5) Checking the areas where programming or erasure was interrupted
If programming or erasure is interrupted intentionally or unintentionally, the programming or erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming or erasure was interrupted, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to prove that the corresponding area is completely erased before using.

(6) Items prohibited during Flash sequencer processing
Do not perform the following operations while the Flash sequencer is processing (FSTATR_n.FRDY=0).
For details of FSTATR_n register, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.

- Have the operating voltage from the power supply go beyond the allowed range.
- Update the values of FHVE15FPn and FHVE3FPn for the FPSYSn of which Flash sequencer is processing.
  For details of these registers, see the *RH850/U2A-EVA Group Flash Memory User's Manual: Hardware*.
- Change the operating frequency of peripheral clock.
- Entering STOP mode / DeepSTOP mode.

(7) Securing coherency of instruction cache and data buffer
After the code flash memory is programmed or erased, a reset or instruction cache and data buffer clearing are required to secure coherency of the instruction cache and data buffer. For details of instruction cache and data buffer, see **Section 3, CPU System**. For ICUMHA, see the *RH850/U2A-EVA Group Security User's Manual: Hardware*.

(8) Setting of OTP.
Set OTP after programing target data, because OTP setting becomes effective without reset.

(9) STOP mode and suspended programming / erasure
It is prohibited to enter STOP mode with suspended programming / erasure.

(10) Reprogramming of User Area or Data Area after Emulation Device mode change. [For U2A-EVA]
Do not issue reprogramming of User Area or Data Area without reset after changing the Emulation Device mode which affects the address mapping of User Area or Data Area.
EVA_PRD_EMU in OPBT9 will change the Emulation Device mode after the reset which causes the Reset Transfer. For details of the reset, see **Section 51.17, Reset Transfer**.
When changing each product setting in EVA_PRD_EMU, it is necessary to reprogram the entire area of Flash as well because each address map of Flash is also changed.

(11) Reprogramming of User Area after Code Flash Memory Mapping Mode change.
Do not issue reprogramming of User Area without reset after changing the Code Flash Memory Mapping Mode which affects the address mapping of User Area.
MAPMODE in OPBT12 will change Code Flash Memory Mapping Mode after the reset which causes the Reset Transfer. For details of the reset, see **Section 51.17, Reset Transfer**.

(12) Code Flash Memory Mapping Mode or Emulation Device mode change after once it is set.
If Code Flash Memory Mapping Mode or Emulation Device mode is changed by mis-operation, the area which is protected in old mode may be revealed in new mode unintentionally.
EVA_PRD_EMU in OPBT9 will change Emulation Device mode. [For U2A-EVA]
MAPMODE in OPBT12 will change Code Flash Memory Mapping Mode.
For product device, it is recommended to set OTP for MAPMODE in OPBT12 once the configuration of MAPMODE is set.

(13) U2A8 mode and Block Protection Area for FPSYS1 [For U2A-EVA]
Make sure that both valid and invalid area of Block Protection Area for FPSYS1 are programmed completely and the contents are the same before changing the device mode to U2A8 mode.
EVA_PRD_EMU in OPBT9 will change Emulation Device Mode.

(14) Two User Boot Area for OTA(Over-the-Air) update support function
Code Flash Memory Mapping Mode of Single Map Mode and Double Map Mode have two User Boot Area. These two User Boot Area are arranged in address space. One is located at Bank A and the other is at Bank B. In Double Map Mode, the stored data in two User Boot Area can be swapped with each other by setting of User Area on Cluster 0 (OPBT13.DBMAPSW0). It has no setting to swap only User Boot Area. Therefore, it is necessary to store same data on both sides of User Boot Area in advance when using User Boot Area in Double Map Mode.

Section 52 RAM

This section describes the RAM mounted on RH850/U2A-EVA Group.

52.1 List of On-Chip RAMs

Table 52.1 shows the list of On-Chip RAMs.

Table 52.1 List of On-Chip RAMs

		RH850/U2A-EVA	RH850/U2A16	RH850/U2A8	RH850/U2A6
Local RAM per CPU		64 KB	64 KB	64 KB	64 KB
Cluster0 RAM		512 KB	512 KB	512 KB	512 KB
Cluster1 RAM		512 KB	512 KB	No	No
Cluster2 RAM		2048 KB	2048 KB	1024 KB	No
Cluster3 RAM*1 (AWO, Retention RAM)		256 KB	256 KB	128 KB	128 KB
Cluster0 ERAM*2		1 MB	No	No	32 KB
Cluster1 ERAM *2		1 MB	No	No	No
Global ERAM (GERAM)*2		2 MB	No	No	No
Instrumentation RAM*2		96 KB	No	No	No
Instruction cache RAM (tag)		128 lines × 4/core	128 lines × 4/core	128 lines × 4/core	128 lines × 4/core
Instruction cache RAM (data)		4 KB × 4/core	4 KB × 4/core	4 KB × 4/core	4 KB × 4/core
DTSRAM		4 KB	4 KB	4 KB	4 KB
sDMAC RAM	sDMAC Descriptor RAM (DPRAM)	4 KB/module	4 KB/module	4 KB/module	4 KB/module
	sDMAC Data RAM (DATARAM)	2 KB/module	2 KB/module	2 KB/module	2 KB/module
FLXA RAM	FLXA Message RAM (MRAM)	8 KB/module	8 KB/module	8 KB/module	8 KB/module
	FLXA RTBFRAM (TBF A/B)	512 B × 2/module	512 B × 2/module	512 B × 2/module	512 B × 2/module
ETNB RAM	Transmit FIFO	16 KB × 1/module	16 KB × 1/module	16 KB × 1/module	16 KB × 1/module
	Receive FIFO	8 KB × 1/module	8 KB × 1/module	8 KB × 1/module	8 KB × 1/module
RS-CANFD RAM	RS-CANFD AFLRAM	1.5 KB × 2/channel	1.5 KB × 2/channel	1.5 KB × 2/channel	1 KB × 2/channel
	RS-CANFD MBRAM	24.2 KB/channel	24.2 KB/channel	24.2 KB/channel	7.5 KB/channel
GTM MCS RAM (0/1)		8 KB × 4 4 KB × 4	8 KB × 4 4 KB × 4	8 KB × 4 4 KB × 4	8 KB × 4 4 KB × 4
MSPI RAM		128 B/module	128 B/module	128 B/module	512 B/module
MMCA RAM		512 B × 2	512 B × 2	512 B × 2	512 B × 2

Note 1. All data is retained in STOP mode and DeepSTOP mode (without MBIST execution).

Note 2. All data is retained in STOP/DeepSTOP mode in debug mode.

52.2 Features

Access:

The CPUn*1, DMA and H-Bus Master can access the Local RAM (CPUn*1), the Cluster 0/1/2/3 RAM*2, the Cluster 0/1 ERAM, and the Global ERAM. The access latency from CPUn to the Local RAM for CPUn*1 is the same as the access to the Local RAM (self).

The access latency from CPU0 or CPU1 to Cluster0 RAM is faster than the access latency to Cluster1/2/3 RAM*2, because CPU0 and CPU1 are in Cluster0. The access latency from CPU2 or CPU3 to Cluster1 RAM*2 is faster than the access latency to Cluster0/2/3 RAM, because CPU2 and CPU3 are in Cluster1*2. Cluster 2 only contains Cluster2 RAM, Cluster 3 only contains Cluster3 RAM, no CPU included.

Note 1. RH850/U2A-EVA and RH850/U2A16: n = 0 to 3
RH850/U2A8 and RH850/U2A6: n = 0, 1

Note 2. Cluster 1 and Cluster 1 RAM are not implemented in RH850/U2A8.
Cluster 1, 2 and Cluster 1, 2 RAM are not implemented in RH850/U2A6.

RAM Retention:

In RH850/U2A-EVA Group, Cluster3 RAM are retained in Standby mode.

For details, see **Section 15, Standby Controller (STBC)**.

ECC:

The Local RAM(CPU0/CPU1/CPU2*3/CPU3*3) and Cluster RAM include the ECC.

The Local RAM(CPU0/CPU1/CPU2*3/CPU3*3) and Cluster RAM also include the address feedback function. For details, see **Section 44, Functional Safety**.

Note 3. CPU2 and CPU3 are not implemented in RH850/U2A8 and RH850/U2A6.

RAM Initialization:

To avoid long initialization phases by software, a hardware mechanism is implemented to initialize the following RAMs. This initialization includes correct setting of the related ECC bits.

- GTM MCS RAM
- MMCA RAM
- MSPI RAM
- DTSRAM
- sDMAC Descriptor RAM (DPRAM)

For details of RAM initialization function, see **Section 9.5.6, RAM Initialization**.

52.3 RAM Data Retention

When reset*1 and write to RAM*1 occur at the same time, undefined data will not be written, and the data before writing or the data after writing will be retained depending on the timing.

When reset*1 and read from RAM*1 occur at the same time, undefined data will not be written and the data will be retained.

Note 1. See **Table 52.2** for the target reset factor and RAM. Regarding detail information of reset, see **Section 9, Reset Controller**.

Table 52.2 RAM Data Retention by each Reset factor

No	Category	Source	Retention RAM (included in Cluster RAM)	Local RAM per CPU, Cluster RAM *1	Cluster ERAM, Global ERAM, Instrumentation RAM
1	Power On Reset	POC Reset	—	—	—
		Debugger Reset	√	√	√
2	System Reset 1	External Reset	√	√	√
		Primary VMON Reset	—	—	—
		Debugger Disconnection Reset	√	√	√
3	System Reset 2	Software System Reset	√	√	√
		ECM Reset (if RESC0 = 0)	√	√	√
		WDTBA Reset	√	√	√
4	Application Reset	Software Application Reset	√	√	√
		ECM Reset (if RESC0 = 1)	√	√	√
5	DeepSTOP Reset	DeepSTOP Reset	√	—	√
6	Module Reset	Software Module Reset	√	√	√
7	JTAG Reset	JTAG Reset (TRST)	√	√	√

Note 1. In case MBIST is executed, RAM data is not retained.

52.4 Usage Notes

When ECC error detection/correction is enabled for the Local RAM and Cluster RAM, initialize the RAM using a write instruction with the maximum bit length of its access size before using the RAM.

For the maximum bit length of the RAM access size, see the applicable data width in **Section 44, Functional Safety**.

If the RAM is accessed before its initialization, an ECC error may be detected. Also, if initialization with the maximum bit length is not performed (e.g. if a 32-bit RAM is initialized by an 8-bit or 16-bit access), an ECC error may be detected.

Section 53 Boundary Scan

This section contains a generic description of boundary scan.

This product has a JTAG interface and provides a boundary scan function.

53.1 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1 that is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of this product conforms to IEEE Std 1149.1-2001.

53.2 Features

- Five control signals (TCK, TDI, TDO, TMS, and $\overline{\text{TRST}}$)
- TAP controller
- Instruction register
- Bypass register
- Boundary scan register

The JTAG interface has six instructions.

- BYPASS
  Test mode conforming to the IEEE 1149.1
- EXTEST
  Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD
  Test mode conforming to the IEEE 1149.1
- CLAMP
  Test mode conforming to the IEEE 1149.1
- HIGHZ
  Test mode conforming to the IEEE 1149.1
- IDCODE
  Test mode conforming to the IEEE 1149.1

Figure 53.1 shows a block diagram of the JTAG interface.

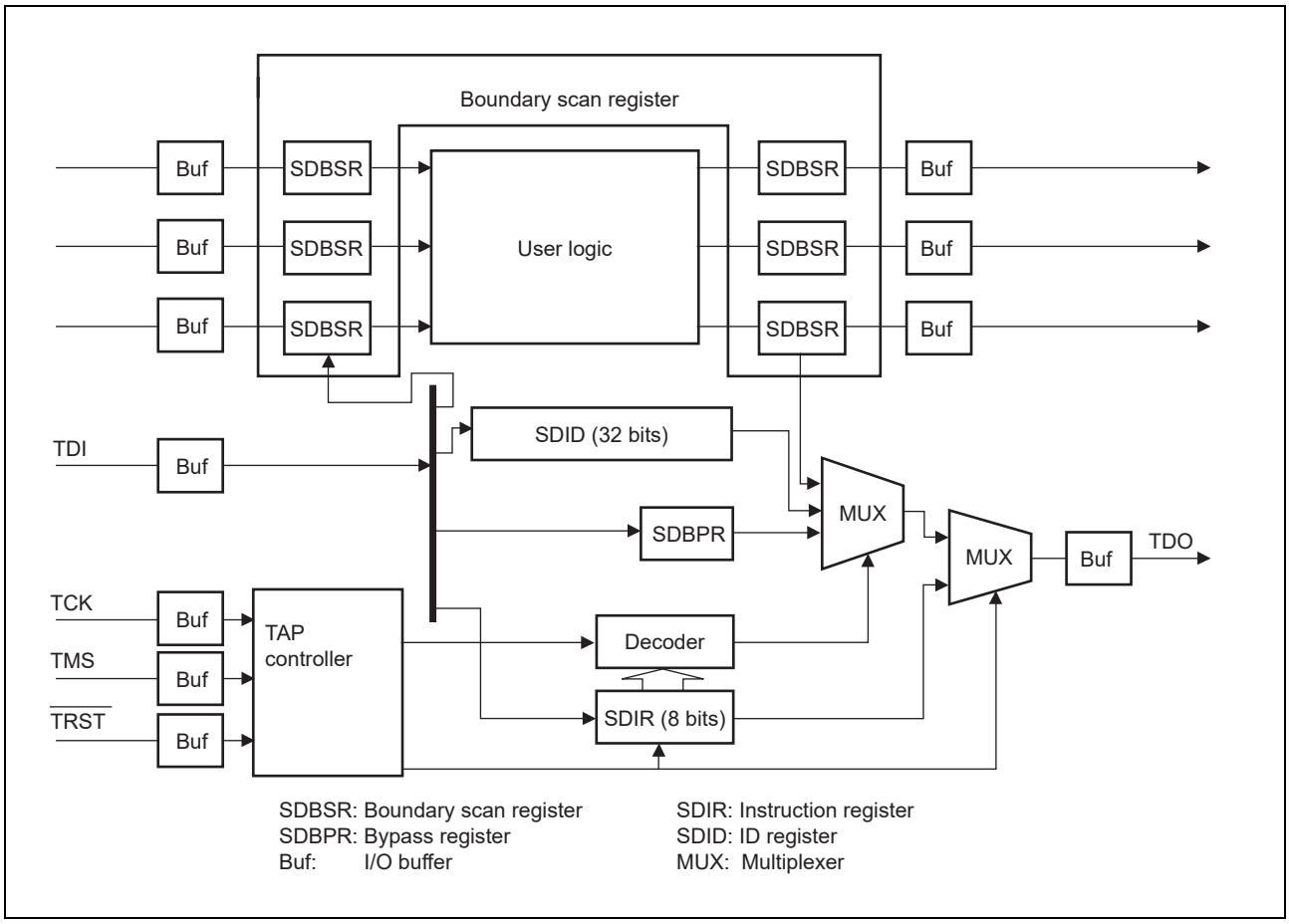


Figure 53.1 Block Diagram of JTAG Interface

53.3 External Input/Output Pins

There are five JTAG control signals: TCK, TDI, TMS, TDO, and $\overline{\text{TRST}}$.

Table 53.1 shows the pin configuration.

Table 53.1 Pin Configuration

Pin Name	Description
TCK	Serial data input/output clock pin Data is input to TDI and is output from TDO in synchronization with this clock signal.
TMS	Mode select input pin Changing the level of this signal in synchronization with TCK changes the state of the TAP controller. For the protocol, see **Figure 53.2, TAP Controller State Transition Diagram**.
$\overline{\text{TRST}}$	Reset input pin A low-level input of this signal resets the JTAG interface. This signal is accepted asynchronously with TCK.
TDI	Serial data input pin Data is input in synchronization with TCK and sent to the JTAG interface.
TDO	Serial data output pin Data to be read from the JTAG interface is output in synchronization with TCK.

53.4 Register Descriptions

The JTAG interface has the following registers. None of the registers can be accessed by the CPU.

- SDIR: Instruction register
- SDID: ID register
- SDBPR: Bypass register
- SDBSR: Boundary scan register

Table 53.2 Register Configuration

Register Name	Symbol	Access Size	Initial Value*1
Instruction register	SDIR	8	55_H
ID register	SDID	32	*2
Bypass register	SDBPR	1	Undefined
Boundary scan register	SDBSR	—	Undefined

Note 1. Registers are initialized when $\overline{\text{TRST}}$ pin is 0 or when TAP is in the Test-Logic-Reset state.

Note 2. The initial value differs depending on the device. Refer to **Table 53.3, List of ID Register (SDID) Value**.

Table 53.3 List of ID Register (SDID) Value

Products	MSPI restriction*1	SVR restriction *2	Product Name	ID Register Code			
				31 to 28	27 to 12	11 to 1	0
RH850/U2A-EVA/BGA516	Yes	No	R7F702Z19AEDBG	0001	1000 0100 0000 1011	0100 0100 011	1
RH850/U2A-EVA/BGA516	No	No	R7F702Z19BFDBG	0001	1000 0100 0100 1001	0100 0100 011	1
RH850/U2A16/BGA516	Yes	Yes	R7F702300EBBG-C	0001	1000 0100 0000 0110	0100 0100 011	1
RH850/U2A16/BGA373	Yes	Yes	R7F702300EBBB-C	0001	1000 0100 0001 0011	0100 0100 011	1
RH850/U2A16/BGA292	Yes	Yes	R7F702300EABA-C	0001	1000 0100 0000 0111	0100 0100 011	1
RH850/U2A16/BGA516	No	Yes	R7F702300AEBBC-C	0001	1000 0100 0101 0011	0100 0100 011	1
RH850/U2A16/BGA373	No	Yes	R7F702300AEBBB-C	0001	1000 0100 0100 1100	0100 0100 011	1
RH850/U2A16/BGA292	No	Yes	R7F702300AFABA-C	0001	1000 0100 0100 1101	0100 0100 011	1
RH850/U2A16/BGA516	No	No	R7F702300BEBBC-C	0001	1000 0100 0111 1101	0100 0100 011	1
RH850/U2A16/BGA373	No	No	R7F702300BEBBB-C	0001	1000 0100 0111 1110	0100 0100 011	1
RH850/U2A16/BGA292	No	No	R7F702300BFABA-C	0001	1000 0100 0111 1111	0100 0100 011	1
RH850/U2A8/BGA373	Yes	Yes	R7F702301EBBA-C	0001	1000 0100 0001 0100	0100 0100 011	1
RH850/U2A8/BGA292	Yes	Yes	R7F702301EABG-C	0001	1000 0100 0000 1000	0100 0100 011	1
RH850/U2A8/BGA373	No	Yes	R7F702301AEBBA-C	0001	1000 0100 0100 1110	0100 0100 011	1
RH850/U2A8/BGA292	No	Yes	R7F702301AFABG-C	0001	1000 0100 0100 1111	0100 0100 011	1
RH850/U2A8/BGA373	No	No	R7F702301BEBBA-C	0001	1000 0100 1000 0000	0100 0100 011	1
RH850/U2A8/BGA292	No	No	R7F702301BFABG-C	0001	1000 0100 1000 0001	0100 0100 011	1
RH850/U2A6/BGA292	No	No	R7F702302FABB-C	0001	1000 0100 0010 0000	0100 0100 011	1
RH850/U2A6/BGA156	No	No	R7F702302FABD-C	0001	1000 0100 0010 0010	0100 0100 011	1
RH850/U2A6/QFP176	No	No	R7F702302FAFK-C	0001	1000 0100 0010 0011	0100 0100 011	1
RH850/U2A6/QFP144	No	No	R7F702302FAFM-C	0001	1000 0100 0010 0100	0100 0100 011	1

Note 1. Refer to **Section 19.8, MSPI restrictions**.

Note 2. Refer **Section 10.10, SVR restriction**.

Note: Bit[31:28]: Revision Number. These bits indicate the revision of the product.
Bit[27:12]: Product Number. These bits indicate the product number.
Bit[11:1]: Manufacture ID. These bits indicate the manufacturer number (manufacturer ID).
223_H: Renesas Electronics.

Instructions can be serially transferred from the serial data input pin (TDI) and input to the instruction register (SDIR). The bypass register (SDBPR) is a 1-bit register, to which TDI and TDO are connected in BYPASS mode, CLAMP mode and HIGHZ mode. The boundary scan register (SDBSR) is connected to TDI and TDO in SAMPLE/PRELOAD mode and EXTEST mode. The ID code register (SDID) is a 32-bit register, from which the ID code is output via TDO in IDCODE mode.

Table 53.4 shows the serial transfer types possible with the JTAG interface registers.

Table 53.4 Serial Transfer Types

Register	Serial Input	Serial Output
SDIR	Possible	Impossible*1
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDID	Impossible	Possible

Note 1. A fixed value is read out.

53.4.1 Instruction Register (SDIR)

SDIR is an 8-bit register that holds a boundary scan instruction. SDIR is initialized by a low-level input of $\overline{\text{TRST}}$ or in the TAP Test-Logic-Reset state. Operation is not guaranteed if a reserved instruction is set in this register.

Table 53.5 Boundary Scan Instructions

Instruction Code								Description
0	0	0	0	0	0	0	0	JTAG EXTEST
0	1	0	0	0	0	0	0	JTAG SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	JTAG IDCODE (initial value)
1	1	0	1	0	0	0	0	JTAG CLAMP
1	0	0	0	0	0	0	0	JTAG HIGHZ
1	1	1	1	1	1	1	1	JTAG BYPASS
Other than above								Reserved

53.4.2 ID Register (SDID)

SDID is a 32-bit register with a device specific ID.

SDID can be read from the JTAG interface when the IDCODE instruction is set, but cannot be accessed from the CPU.

For the read values, see **Table 53.2, Register Configuration**.

53.4.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register to go through the boundary scan register. When SDIR is set to BYPASS mode, CLAMP mode or HIGHZ mode, SDBPR is connected between TDI and TDO. The initial value is undefined. SDBPR is not initialized by a power-on reset or by a low level input of $\overline{\text{TRST}}$.

53.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register for controlling the external Input/Output pins. When SDIR is set to SAMPLE/PRELOAD or EXTEST mode, SDBSR is connected to the position between TDI and TDO. The initial value is undefined. SDBSR is not initialized by a power-on reset or a low-level input of $\overline{\text{TRST}}$.

53.5 Operation

53.5.1 TAP Controller

Table 53.2 shows the state transition of the TAP controller. Transition is triggered by the TMS value at the rising edge of TCK.

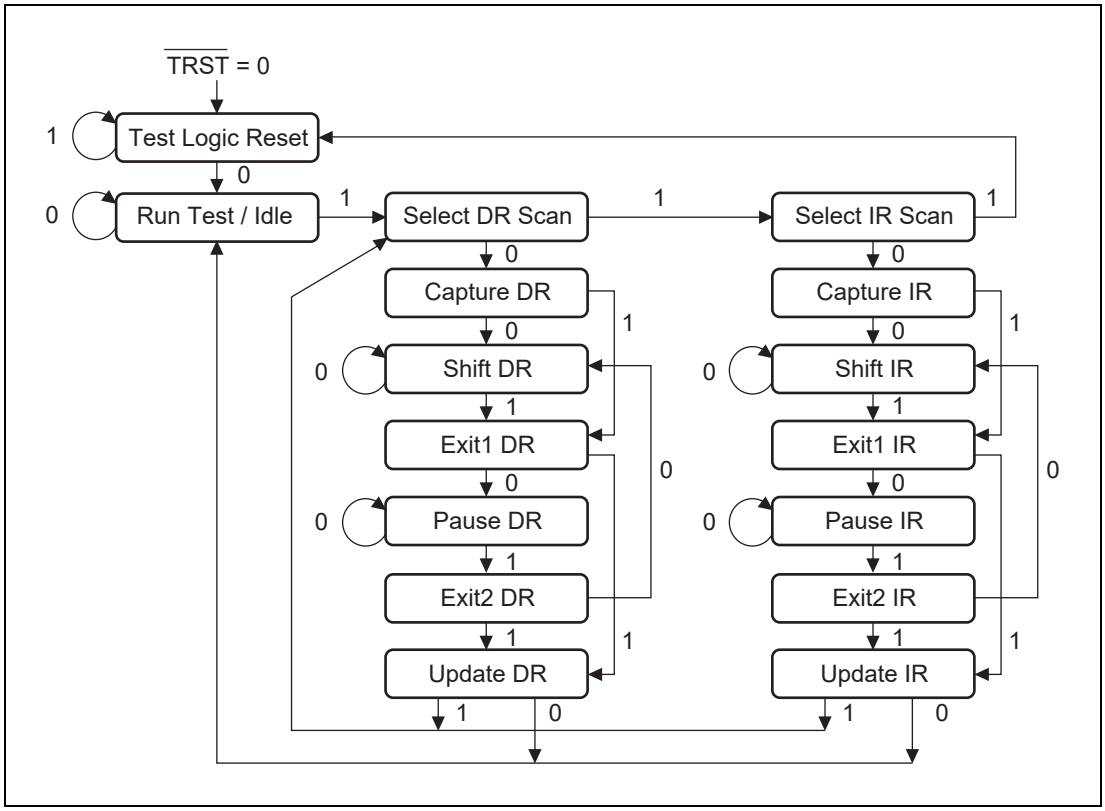


Figure 53.2 TAP Controller State Transition Diagram

NOTE

The TDI value is sampled at the rising edge of TCK and is shifted at the falling edge. TDO is in the high-impedance state in the states other than Shift-DR and Shift-IR. A low-level input of $\overline{\text{TRST}}$ causes transition to Test-Logic-Reset state asynchronously with TCK.

53.5.2 Supported Instructions

53.5.2.1 BYPASS

The BYPASS instruction is a standard instruction indispensable to bypass register operation. This instruction shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this instruction, the test circuit has no effect on the system circuit.

53.5.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input the value to the boundary scan register from the internal circuits of this device; to output the value from the scan path; and to load data onto the scan path. During execution of this instruction, the level of the input pin of this device is sent to the internal circuits as is, and the value of the internal circuits is output to the outside via the output pin as is. Executing this instruction has no effect on the system circuit of this device.

The SAMPLE operation allows taking in the snapshots of the value to be transferred to the internal circuits from the input pin or the value to be transferred to the output pin from the internal circuits to the boundary scan register and allows reading the snapshots from the scan path. Snapshots can be taken in without preventing the normal operation of this device.

The PRELOAD operation allows setting the initial value to the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. If the EXTEST instruction is executed without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transfer to the output latch) because the parallel output latch value is always output to the output pin by the EXTEST instruction.

53.5.2.3 EXTEST

The EXTEST instruction is used to test the external circuits when this device is mounted on the printed-circuit board. When this instruction is executed, the output pin is used to output the test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed-circuit board; whereas the input pin is used to take in the test result from the printed-circuit board to the boundary scan register. When the EXTEST instruction is executed N times for testing, the test data for the Nth execution is scanned in at the (N - 1)th scan-out.

If the data is loaded onto the boundary scan register of the output pin in the Capture-DR state of this instruction, it is not used for testing the external circuits (replaced through shift operation).

53.5.2.4 IDCODE

The IDCODE instruction sets the JTAG interface pins to IDCODE mode, which is defined by the JTAG standard. When the JTAG interface is initialized (by a low-level input of $\overline{\text{TRST}}$ or placing TAP in the Test-Logic-Reset state), IDCODE mode is set.

53.5.2.5 CLAMP instruction

When the CLAMP instruction is selected, the output pin outputs the boundary scan register value that has been previously set by the SAMPLE/PRELOAD instruction. While the CLAMP instruction is selected, the boundary scan register retains the previous state regardless of the TAP controller state. The bypass register is connected between TDI and TDO and operates in the same manner as when the BYPASS instruction is selected.

53.5.2.6 HIGHZ instruction

When the HIGHZ instruction is selected, all the output pins go to the high-impedance state. While the HIGHZ instruction is selected, the boundary scan register retains the previous state regardless of the TAP controller state. The bypass register is connected between TDI and TDO and operates in the same manner as when the BYPASS instruction is selected.

53.5.3 Pins Subject to Boundary Scan

All pins, excluding pins such as external clock input pins or power supply pins, are subject to boundary scan.

The pins which are not subjected to boundary scan are listed in **Table 53.6**.

For the supported terminals of the product, refer to Appendix "E02_01_List_of_Pin_Assignment.xlsx".

Table 53.6 Pins not Subject to Boundary Scan

Type	Pins
JTAG interface	JP0_0/TDI, JP0_1/TDO, JP0_2/TCK, JP0_3/TMS, $\overline{\text{TRST}}$
Mode pins	P6_13/FLMD1, P6_14/FLMD2, P5_3/MODE0
Power supply pins	SYSVCC, VCC, AWOVCL, VSS, VDD, E0VCC, E1VCC, E2VCC, SVRDRVCC, SVRDRVSS, LVDVCC, DVCC, DVDD, EMUVCC, EMUVDD, ERAMVDD, ERAMVCC, GETH0BVCC, GETH0RVCC, GETH0PVCC, GETH0VCL, SVRAVCC, SVRAVSS
Power supply pins (A/D converter)	A0VCC, A1VCC, A2VCC, A0VREFH, A1VREFH, A2VREFH, A0VSS, A1VSS, A2VSS
Clock signal	X1, X2
System pins	$\overline{\text{VMONOUT}}$, $\overline{\text{ERROROUT\_M}}$, PWRCTL, SVRNGATE, SVRPGATE
Debug pins	$\overline{\text{MSYN}}$, $\overline{\text{AURORES1}}$, $\overline{\text{AURORES2}}$, $\overline{\text{AURORESPD}}$, TODP0 to TODP3, TODN0 to TODN3, $\overline{\text{EVTI}}$, $\overline{\text{EVTO}}$, $\overline{\text{AUDRST}}$, AUDCK, $\overline{\text{AUDSYNC}}$, AUDATA0 to AUDATA3, CICREFP, CICREFN, $\overline{\text{ERAMRES2}}$, $\overline{\text{ERAMRESPD}}$
SGMII pins	RX_DATAN, RX_DATAP, TX_DATAN, TX_DATAP, RX_CLKN, RX_CLKP

The following signals are only sampled in boundary scan mode.

Table 53.7 Pins Subject to Boundary Scan (Sampling Only)

Function	Pin Name
Reset	$\overline{\text{RESET}}$
MODE	FLMD0

The following pins are shared by the analog buffer. Accordingly, boundary scan only applies to general I/O pins.

Table 53.8 Pins Subject to Boundary Scan (Only General I/O Pins)

Function	Pin Name
ADCJ0 input	P24_4 to P24_13, AP0_0 to AP0_15, AP1_0 to AP1_3
ADCJ1 input	P10_8 to P10_14, P17_0 to P17_6, AP2_0 to AP2_15, AP3_0 to AP3_3
ADCJ2 input	P6_0, P6_2 to P6_7, P6_9, P6_11, P6_12, AP4_0 to AP4_15, AP5_0 to AP5_3

NOTE

In boundary scan mode, the level of the following pins must be fixed:
FLMD1: High, FLMD2: High, MODE0: Low

53.6 Usage Notes

1. Once a instruction is set, it is not modified until another instruction is issued. To issue the same instruction twice in a row, insert a instruction that has no effect on chip operation (such as BYPASS) between the instructions.
2. To start the system in boundary scan mode, de-assert $\overline{\text{TRST}}$ while $\overline{\text{RESET}}$ is high. Also be sure to set TMS to high before de-asserting $\overline{\text{TRST}}$ and ensure that TMS remains high for 600 ns + five TCK clock cycles after de-asserting $\overline{\text{TRST}}$.
3. For details of boundary scan timing constraint, see **Section 55, Electrical Characteristics**.
4. If serial transfer is performed exceeding the number of bits of the register connected between TDI and TDO, the data that is input from TDI is output from TDO as is.
5. If the serial transfer sequence is corrupted, be sure to assert $\overline{\text{TRST}}$. In this case, transfer starts again from the beginning regardless of the point of transfer corruption.
6. Data is output via TDO at the falling edge of TCK.
7. To facilitate debugging, route $\overline{\text{TRST}}$ on the board in such a way that patterns can be easily cut.

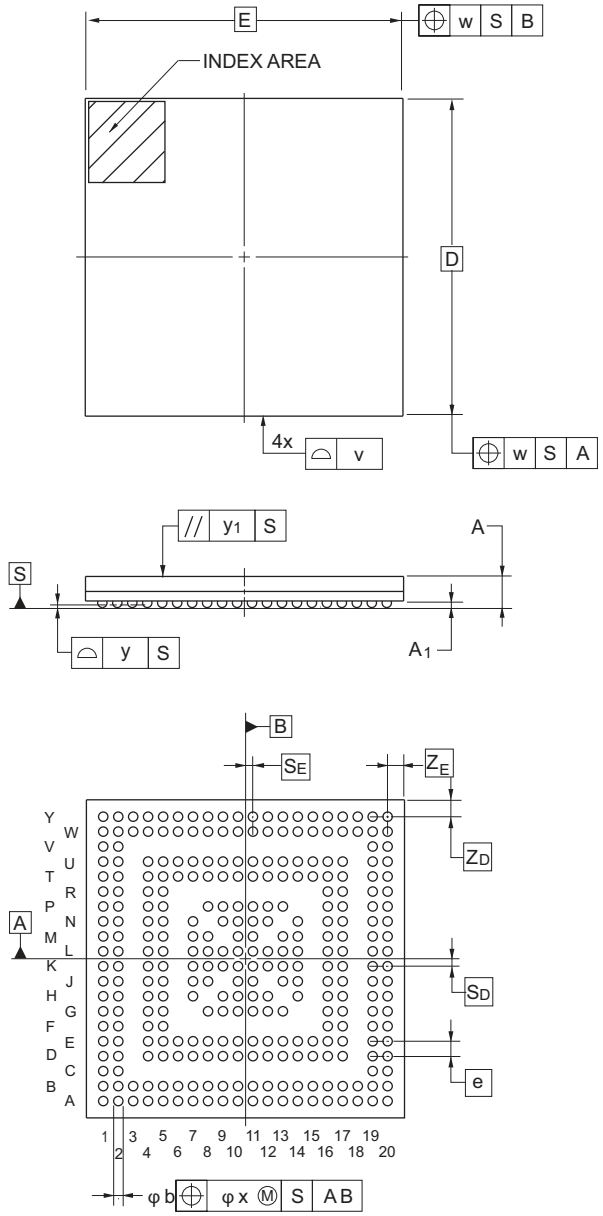
Section 54 Package

54.1 Package Outline

54.1.1 FPBGA (292pin) Package Drawing

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-FBGA292-17x17-0.80	PRBG0292GC-A	T292F1-80-GNP	0.9

Unit: mm



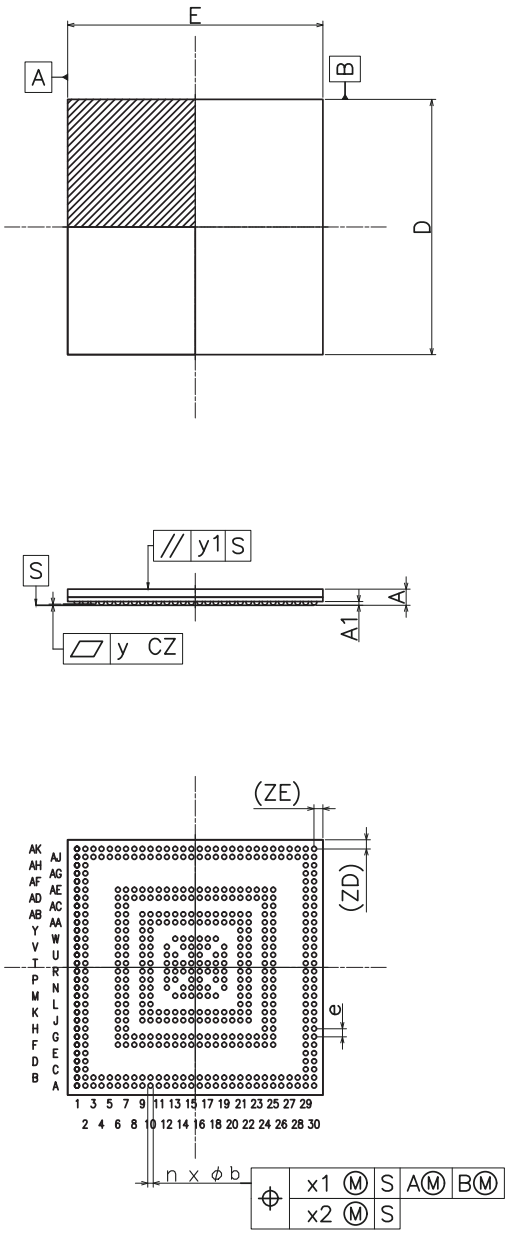
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	17.00	—
E	—	17.00	—
A	—	—	1.90
A_1	0.30	0.35	0.40
e	—	0.80	—
b	0.50	0.55	0.60
x	—	—	0.12
y	—	—	0.10
y_1	—	—	0.20
Z_D	—	0.90	—
Z_E	—	0.90	—
v	—	—	0.15
w	—	—	0.20
S_D	—	0.40	—
S_E	—	0.40	—

Note: INDEX AREA is on the corner near a ball A1 (Top View side).

Figure 54.1 FPBGA (292pin) outline

54.1.2 FPBGA (516pin (RENESAS Code = PRBG0516GC-A)) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-FBGA516-25x25-0.80	PRBG0516GC-A	2.00



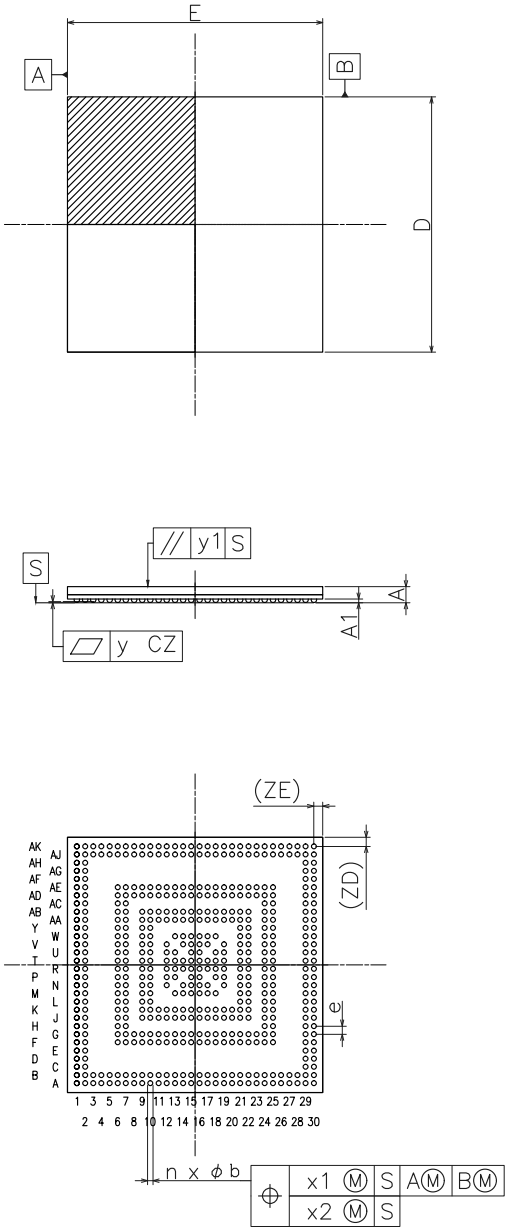
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	24.90	25.00	25.10
E	24.90	25.00	25.10
A	—	—	2.00
A_1	0.29	0.34	0.39
e	—	0.80	—
b	0.50	0.55	0.60
x_1	—	—	0.20
x_2	—	—	0.08
y	—	—	0.15
y_1	—	—	0.20
n	—	516	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Note: INDEX AREA is on the corner near a ball A1 (Top View side).

Figure 54.2 FPBGA (516pin (RENESAS Code = PRBG0516GC-A)) outline

54.1.3 FPBGA (516pin (RENESAS Code = PRBG0516GD-A)) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-FBGA516-25x25-0.80	PRBG0516GD-A	2.20



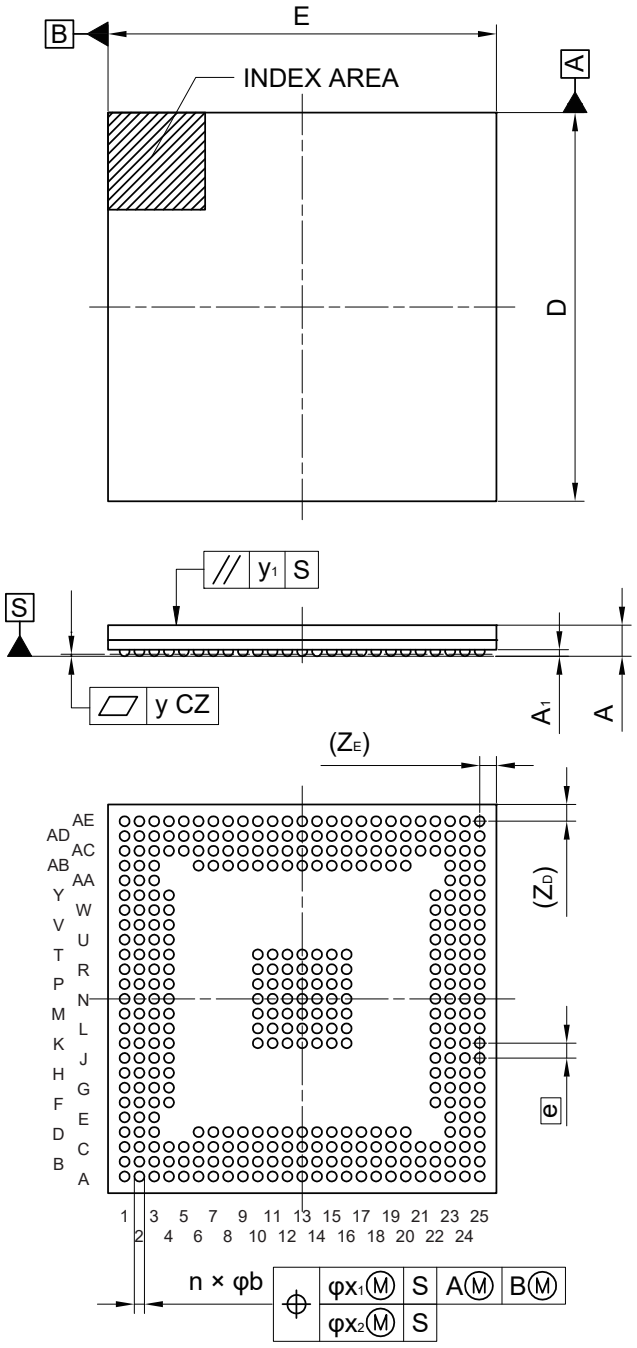
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	24.90	25.00	25.10
E	24.90	25.00	25.10
A	—	—	2.00
A_1	0.29	0.34	0.39
e	—	0.80	—
b	0.50	0.55	0.60
x_1	—	—	0.20
x_2	—	—	0.08
y	—	—	0.15
y_1	—	—	0.20
n	—	516	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Note: INDEX AREA is on the corner near a ball A1 (Top View side).

Figure 54.3 FPBGA (516pin (RENESAS Code = PRBG0516GD-A)) outline

54.1.4 FPBGA (373pin) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-FBGA373-21x21-0.80	PRBG0373GB-A	1.4



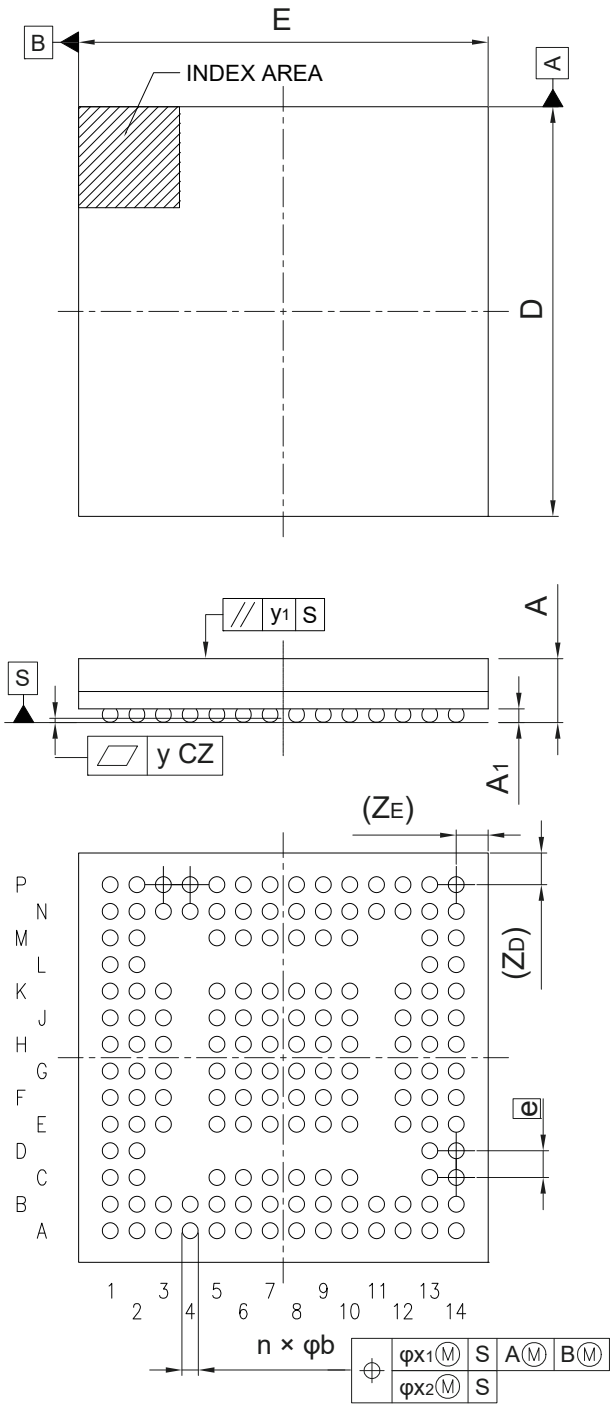
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	20.90	21.00	21.10
E	20.90	21.00	21.10
A	—	—	2.00
A_1	0.30	0.35	0.40
e	—	0.80	—
b	0.49	0.54	0.59
x_1	—	—	0.20
x_2	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
Z_D	—	0.90	—
Z_E	—	0.90	—
n	—	373	—

Note: INDEX AREA is on the corner near a ball A1 (Top View side).

Figure 54.4 FPBGA (373pin) outline

54.1.5 LFPBGA (156pin) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA156-10x10-0.65	PLBG0156JC-A	0.30



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	9.92	10.00	10.08
E	9.92	10.00	10.08
A	—	—	1.70
A_1	0.21	0.26	0.31
e	—	0.65	—
b	0.40	0.45	0.50
x_1	—	—	0.15
x_2	—	—	0.08
y	—	—	0.10
y_1	—	—	0.20
n	—	156	—
Z_D	—	0.775	—
Z_E	—	0.775	—

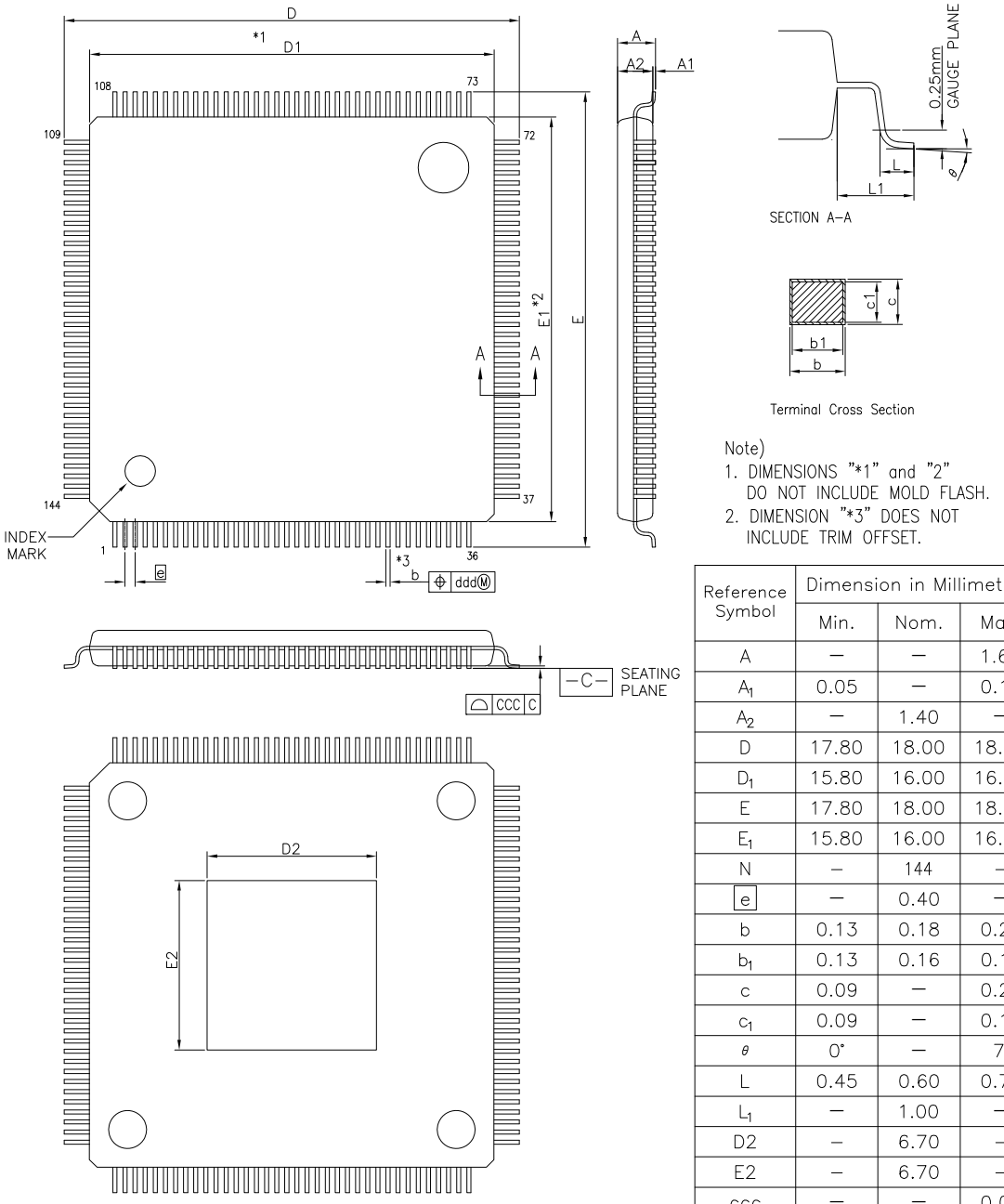
Note: INDEX AREA is on the corner near a ball A1 (Top View side).

Figure 54.5 LFPBGA (156pin) outline

54.1.6 HLQFP (144pin) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HLFQFP144-16x16-0.40	PLQP0144LC-B	0.90

UNIT:MM



Note)
1. DIMENSIONS "*1" and "2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

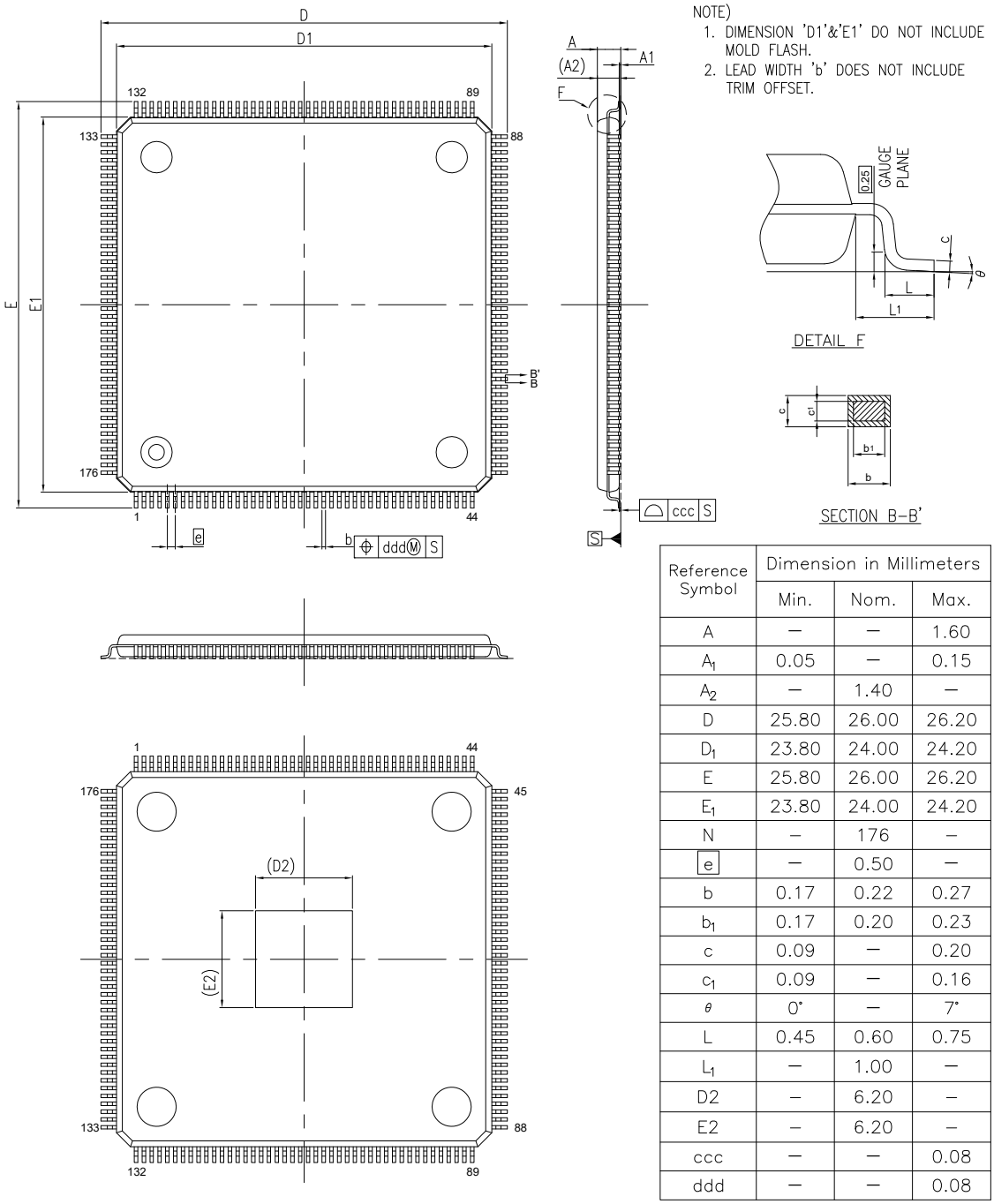
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	–	1.40	–
D	17.80	18.00	18.20
D_1	15.80	16.00	16.20
E	17.80	18.00	18.20
E_1	15.80	16.00	16.20
N	–	144	–
e	–	0.40	–
b	0.13	0.18	0.23
b_1	0.13	0.16	0.19
c	0.09	–	0.20
c_1	0.09	–	0.16
θ	0°	–	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
D2	–	6.70	–
E2	–	6.70	–
ccc	–	–	0.08
ddd	–	–	0.07

Figure 54.6 HLQFP (144pin) outline

54.1.7 HLQFP (176pin) Package Drawing

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HLFQFP176-24x24-0.50	PLQP0176KH-B	1.97

UNIT: MM



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	–	1.40	–
D	25.80	26.00	26.20
D_1	23.80	24.00	24.20
E	25.80	26.00	26.20
E_1	23.80	24.00	24.20
N	–	176	–
e	–	0.50	–
b	0.17	0.22	0.27
b_1	0.17	0.20	0.23
c	0.09	–	0.20
c_1	0.09	–	0.16
θ	0°	–	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
D2	–	6.20	–
E2	–	6.20	–
ccc	–	–	0.08
ddd	–	–	0.08

Figure 54.7 HLQFP (176pin) outline

Section 55 Electrical Characteristics

The specifications in this section are for devices operating under the conditions shown in **Section 55.2.1.1, Supply Voltage Characteristics**. Where a special condition is required for a given specification, the condition will be indicated. Furthermore, the specifications in this section are not guaranteed unless the conditions listed below are met.

55.1 Absolute Maximum Ratings

Conditions:

- VSS = SVRDRVSS = SVRAVSS = AnVSS
- Reference ground potential: VSS = 0 V.

Table 55.1 Absolute Maximum Ratings (1/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Power Voltage	$V_{AMRSYSVCC}$	SYSVCC	–0.5		6.5*10	V
	V_{AMRVCC}	VCC	–0.5		6.5*10	V
	V_{AMRVDD}	VDD	–0.5		1.42*11	V
	$V_{AMRE0VCC}$	E0VCC	–0.5		6.5*10	V
	$V_{AMRE1VCC}$	E1VCC	–0.5		6.5*10	V
	$V_{AMRE2VCC}$	E2VCC	–0.5		6.5*10	V
	$V_{AMRLVDVCC}$	LVDVCC	–0.5		6.5*10	V
	$V_{AMRA0VCC}$	A0VCC	–0.5		6.5*10	V
	$V_{AMRA1VCC}$	A1VCC	–0.5		6.5*10	V
	$V_{AMRA2VCC}$	A2VCC	–0.5		6.5*10	V
	$V_{AMRSVRDRVCC}$	SVRDRVCC	–0.5		6.5*10	V
	$V_{AMRSVRAVCC}$	SVRAVCC	–0.5		6.5*10	V
	$V_{AMRDVCC}$	DVCC	–0.5		4.6*12	V
	$V_{AMRDVDD}$	DVDD	–0.5		1.42*11	V
	$V_{AMREMUVCC}$	EMUVCC	–0.5		4.6*12	V
	$V_{AMREMUVDD}$	EMUVDD	–0.5		1.42*11	V
	$V_{AMRERAMVCC}$	ERAMVCC	–0.5		4.6*12	V
	$V_{AMRERAMVDD}$	ERAMVDD	–0.5		1.42*11	V
	$V_{AMRGETH0RVCC}$	GETH0RVCC	–0.5		6.5*10	V
	$V_{AMRGETH0BVCC}$	GETH0BVCC	–0.5		4.6*12	V
	$V_{AMRGETH0PVCC}$	GETH0PVCC	–0.5		6.5*10	V
Input Voltage	V_{AMRI}	E0VCC pin*1	–0.5		E0VCC + 0.5	V
		E1VCC pin*2	–0.5		E1VCC + 0.5	V
		E2VCC pin*3	–0.5		E2VCC + 0.5	V
		$\overline{\text{RESET}}$, FLMD0, X1, X2 PWRCTL pin	–0.5		SYSVCC + 0.5	V
		CICREFN, CICREFP,	–0.5		DVDD+0.3 *9	V
		$\overline{\text{AURORES1}}$, $\overline{\text{AURORES2}}$	–0.5		EMUVCC + 0.5	V
		$\overline{\text{MSYN}}$, $\overline{\text{AURORESPD}}$, $\overline{\text{EVTI}}$, $\overline{\text{EVTO}}$, AUDR I/F*4, $\overline{\text{VMONOUT}}$	–0.5		E0VCC + 0.5	V

Table 55.1 Absolute Maximum Ratings (2/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input Voltage	V_{AMRI}	$\overline{ERAMRESPD}$	–0.5		E1VCC + 0.5	V
		$\overline{ERAMRES2}$	–0.5		ERAMVCC + 0.5	V
		RHSIF, MSPI*5	–0.5		LVDVCC + 0.5	V
		SGMII I/F*6	–0.5		GETH0BVCC + 0.5	V
		P3_8 [For U2A-EVA, U2A16 and U2A8 Only]	–0.5		GETH0RVCC + 0.5	V
Analogue reference voltage	$V_{AMRAVREF}$	A0VREFH	–0.5		A0VCC + 0.5	V
		A1VREFH	–0.5		A1VCC + 0.5	V
		A2VREFH	–0.5		A2VCC + 0.5	V
Analogue input voltage	V_{AMRIAN}	A0VCC pins	–0.5		A0VCC + 0.5	V
		A1VCC pins	–0.5		A1VCC + 0.5	V
		A2VCC pins	–0.5		A2VCC + 0.5	V
VSS differential voltage	$V_{VSSDIFF}$		–0.1		0.1	V
Injection current per digital input	I_{INJ_DIN}	*13	–25		25	mA
Injection current per analog input	I_{INJ_AIN}	*13	–25		25	mA
Total Injection current of the device	$\lvert I_{INJ\_TOT} \rvert$	*13			120	mA
Output low current*7	I_{OL1p}	per pin			10	mA
	I_{OLall}	Sum of all output low currents of all EnVCC/ AnVCC pins			200	mA
Output high current*7	I_{OH1p}	per pin			–10	mA
	I_{OHall}	Sum of all output high currents of all EnVCC/ AnVCC pins			–200	mA
Junction temperature	T_j	U2A16 BGA516 U2A16 BGA373 U2A8 BGA373	–40		150	°C
		Other than above	–40		160	°C
Storage temperature*8	T_{stg}	U2A16 BGA516 U2A16 BGA373 U2A8 BGA373	–55		150	°C
		Other than above	–55		160	°C

Note 1. JP0, P0, P1, P2, P3 (except P3_8 [For U2A-EVA, U2A16 and U2A8 Only]), P4, P5, P6, P8

Note 2. P9, P10, P11, P12, P18, P19, P20, 21, P22, P23

Note 3. P24

Note 4. $\overline{AUDRST}$, $\overline{AUDSYNC}$, AUDCK, AUDATA0-AUDATA3

Note 5. RHSIF: HSIF0_TXDP, HSIF0_TXDN, HSIF0_RXDP, HSIF0_RXDN
MSPI: MSPI0_SOP, MSPI0_SON, MSPI0_SIP, MSPI0_SIN, MSPI0_SCKP, MSPI0_SCKN

Note 6. RX_CLKN, RX_CLKP, RX_DATAN, RX_DATAP, TX_DATAN, TX_DATAP and REFCLK.

Note 7. Given specification includes injected currents.

Note 8. After mounting

Note 9. Do not exceed maximum 1.42 V.

Note 10. Voltage overshoot 5.8 V to 6.5 V is permissible, cumulative time is less than 2 h. Voltage overshoot 5.5 V to 5.8 V is permissible, cumulative time is less than 100 h.

Note 11. Voltage overshoot 1.205 V to 1.42 V is permissible, cumulative time is less than 2 h. Voltage overshoot 1.155 V to 1.205 V is permissible, cumulative time is less than 100 h.

Note 12. Voltage overshoot 3.9 V to 4.6 V is permissible, cumulative time is less than 2 h. Voltage overshoot 3.6 V to 3.9 V is permissible, cumulative time is less than 100 h.

Note 13. Input voltage must be kept within -0.8 V ≤ Vin ≤ 6.5 V. Power supply voltage must be kept within rated values. Injection current must be kept within rated values on all states include ramp-up/down, and power-on/off. Injection current effects power dissipation in the package for thermal characteristics.

CAUTION

- **Even momentarily exceeding the absolute maximum rating for just one item creates a threat of failure in the reliability of the products. That is, the absolute maximum ratings are the levels that raise a threat of physical damage to the products. Be sure to use the products only under conditions that do not exceed the ratings. The quality and normal operation of the product are guaranteed under the standards and conditions given as DC and AC characteristics.**
- **Input voltage, analog reference voltage and analog input voltage must not exceed 6.5 V.**
- **Even in case when input voltage does not meet the specified characteristics, it is accepted if the injected current characteristics specified in Section 55.2.7 are met.**

55.2 General & DC Characteristics

55.2.1 Operational Condition

55.2.1.1 Supply Voltage Characteristics

Conditions:

- Temperature range: Tj (min) to Tj (max)
- VSS = SVRDRVSS = SVRAVSS = AnVSS
- Reference ground potential: VSS = 0 V

Table 55.2 Supply Voltage Characteristics (1/2)*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
System control supply voltage*6	V_{SYSVCC}	SVR unused*2	3.0		5.5	V
		SVR used*2	3.0		3.6	V
			4.5		5.5	V
Regulator supply voltage	V_{VCC}	SVR unused	3.0		5.5	V
		SVR used	3.0		3.6	V
			4.5		5.5	V
Core supply voltage	V_{VDD}		1.025	1.09	1.155	V
IO supply voltage	V_{E0VCC}		3.0		5.5	V
	V_{E1VCC}					
	V_{E2VCC}					
RHSIF/MSPI LVDS supply voltage	V_{LVDVCC}		3.0		3.6	V
			4.5		5.5	V
ADC supply voltage	V_{A0VCC}		3.0		5.5	V
	V_{A1VCC}					
	V_{A2VCC}					
ADC reference voltage supply	$V_{A0VREFH}$		3.0		5.5	V
	$V_{A1VREFH}$					
	$V_{A2VREFH}$					
SVR supply voltage	$V_{SVRDRVCC}$	*3	3.0		3.6	V
			4.5		5.5	V
	$V_{SVRAVCC}$		3.0		3.6	V
			4.5		5.5	V
Aurora I/F supply voltage (Analog)	V_{DVCC}		3.0		3.6	V
Aurora I/F supply voltage (Digital)	V_{DVDD}	Aurora used	1.04	1.09	1.14	V
		Aurora unused	1.025	1.09	1.155	V
Aurora control IO supply voltage	V_{EMUVCC}		3.0		3.6	V
Aurora control core supply voltage	V_{EMUVDD}	Aurora used	1.04	1.09	1.14	V
		Aurora unused	1.025	1.09	1.155	V
ERAM IO supply voltage*7	$V_{ERAMVCC}$		3.0		3.6	V
ERAM core supply voltage*7	$V_{ERAMVDD}$	Aurora used	1.04	1.09	1.14	V
		Aurora unused	1.025	1.09	1.155	V

Table 55.2 Supply Voltage Characteristics (2/2)*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SGMII supply voltage*5	$V_{GETH0RVCC}$	SGMII used	3.14	3.3	3.46	V
		SGMII unused	3.0		5.5	V
	$V_{GETH0PVCC}$	SGMII used	3.14	3.3	3.46	V
		SGMII unused	3.0		5.5	V
	$V_{GETH0BVCC}$	SGMII used	3.14	3.3	3.46	V
		SGMII unused		*4		V

Note 1. Use this device with the following conditions.
[For U2A-EVA, U2A16 and U2A8 Only]
SVRDRVCC ≥ SYSVCC = VCC = SVRAVCC
A0VCC ≥ A0VREFH ≥ VCC
A0VCC ≥ A0VREFH ≥ E0VCC = LVDVCC
A1VCC ≥ A1VREFH
A2VCC ≥ A2VREFH
GETH0RVCC = GETH0PVCC = GETH0BVCC, when SGMII use.
[For U2A6 Only]
SVRDRVCC ≥ SYSVCC = VCC = SVRAVCC
A0VCC = A0VREFH ≥ VCC
A0VCC = A0VREFH ≥ E0VCC
A1VCC = A1VREFH
A2VCC = A2VREFH

Note 2. SYSVCC is monitored by POC, see **Section 55.2.13, Voltage Detector (POC, VLVI) Characteristics**.

Note 3. Maximum allowable noise for SVRDRVCC is 500mV peak to peak.

Note 4. Input 3.0V to 3.6V voltage or connect to VSS with 1 kΩ or more pull-down resistance.

Note 5. Maximum allowable noise for SGMII power is 50mV peak to peak.

Note 6. Maximum allowable noise for SYSVCC is 100mV peak to peak.When MainOSC is used as the clock source of SGMII, allowable noise for SYSVCC will be 50mV peak to peak.

Note 7. U2A6 has only ERAM core and its supply voltage is V_{VDD}.

CAUTION

During operations, supply the specified voltages to all power lines. When stopping operation, turn off all of the power supplies.

55.2.1.2 Main Oscillator Characteristics

Conditions:

- Supply voltage range: Refer to **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.3 Main Oscillator Characteristics (1/3)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
MainOSC frequency	f_{MOSC} *1,*2	Crystal/Ceramic		16 – 1%		40 + 1%	MHz
		Crystal*5		20 – 100 ppm		20 + 100 ppm	MHz
MainOSC oscillation operating point	V_{MOSCOP}	Crystal/Ceramic			1.217		V
MainOSC oscillation amplitude	$V_{MOSCAMP}$	Crystal/Ceramic		0.9			V
MainOSC oscillation stabilization time	t_{MSTB}	Crystal/Ceramic				5	ms
MainOSC oscillation amplifier reaction time	$t_{MOSCAMP}$	Crystal/Ceramic				200	µs
Internal Capacitor size selectable by MOSC_CAP_SEL setting (OPBT10 setting)	C_{capsel}	MOSC_CAP_SEL [3:0]	= 0000_B	*4	0 (4.0), 0 (5.6)*3	*4	pF
			= 0001_B	*4	1 (4.9), 1 (6.5)*3	*4	pF
			= 0010_B	*4	2 (6.0), 2 (7.6)*3	*4	pF
			= 0011_B	*4	3 (6.9), 3 (8.5)*3	*4	pF
			= 0100_B	*4	4 (7.7), 4 (9.3)*3	*4	pF
			= 0101_B	*4	5 (8.7), 5 (10.2)*3	*4	pF
			= 0110_B	*4	6 (9.8), 6 (11.3)*3	*4	pF
			= 0111_B	*4	7 (10.6), 7 (12.1)*3	*4	pF
			= 1000_B	*4	8 (11.5), 8 (13.1)*3	*4	pF
			= 1001_B	*4	9 (12.6), 9 (14.1)*3	*4	pF
			= 1010_B	*4	10 (13.5), 10 (15.0)*3	*4	pF
Internal damping resistor size selectable by MOSC_RD_SEL_A and MOSC_RD_SEL_B setting (OPBT10 setting)*2	R_{rdsel}	MOSC_RD_SEL_A [2:0] MOSC_RD_SEL_B [2:0] [U2A-EVA only]	= 000_B	169	338	608	Ω
			= 001_B	718	1435	2440	Ω
			= 010_B	1267	2533	4306	Ω
			= 011_B	1711	3422	5475	Ω
			= 100_B	1711	3422	5475	Ω
			= 101_B	1711	3422	5475	Ω
			= 110_B	1711	3422	5475	Ω
			= 111_B	1711	3422	5475	Ω
	R_{rdsel}	MOSC_RD_SEL_A [2:0] MOSC_RD_SEL_B [2:0] U2A16, U2A8 and U2A6]	= 000_B	238	340	578	Ω
			= 001_B	378	540	1026	Ω
			= 010_B	574	820	1435	Ω
			= 011_B	756	1080	1782	Ω
			= 100_B	959	1370	2124	Ω
			= 101_B	1764	2520	3402	Ω
			= 110_B	1764	2520	3402	Ω
			= 111_B	1764	2520	3402	Ω

Table 55.3 Main Oscillator Characteristics (2/3)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Trans conductance size (g_m) and Output conductance size (g_{ds}) selectable by MOSC_SHTSTBY_A, MOSC_AMPSEL_A, MOSC_SHTSTBY_B and, MOSC_AMPSEL_B setting (OPBT10 setting)*2	g_m, g_{ds}	{MOSC_SHTSTBY_A, MOSC_AMPSEL_A [2:0]} or {MOSC_SHTSTBY_B, MOSC_AMPSEL_B [2:0]} SYSVCC = 3.0 V to 3.6 V	= 0000_B	1.05, 0.079	2.28, 0.153	3.69, 0.452	mS
			= 0001_B	1.66. 0.108	3.58. 0.204	5.78. 0.521	mS
			= 0010_B	2.25, 0.134	4.85, 0.251	7.80, 0.585	mS
			= 0011_B	2.82, 0.158	6.08, 0.294	9.76, 0.645	mS
			= 0100_B	3.73, 0.196	8.07, 0.362	12.91, 0.739	mS
			= 0101_B	4.76, 0.238	10.32, 0.437	16.49, 0.846	mS
			= 0110_B	5.56, 0.271	12.10, 0.496	19.35, 0.929	mS
			= 0111_B	6.33, 0.302	13.81, 0.553	22.08, 1.007	mS
			= 1000_B	3.94, 0.194	8.45, 0.371	13.52, 0.592	mS
			= 1001_B	4.46, 0.215	9.58, 0.409	15.32, 0.651	mS
			= 1010_B	4.97, 0.236	10.69, 0.446	17.07, 0.708	mS
			= 1011_B	5.46, 0.255	11.76, 0.481	18.78, 0.764	mS
			= 1100_B	6.24, 0.286	13.49, 0.538	21.54, 0.852	mS
			= 1101_B	7.12, 0.321	15.46, 0.603	24.71, 0.953	mS
			= 1110_B	7.82, 0.351	17.04, 0.653	27.24, 1.031	mS
			= 1111_B	8.49, 0.377	18.56, 0.701	29.68, 1.105	mS
Trans conductance size (g_m) and Output conductance size (g_{ds}) selectable by MOSC_SHTSTBY_A, MOSC_AMPSEL_A, MOSC_SHTSTBY_B and MOSC_AMPSEL_B setting (OPBT10 setting)*2	g_m, g_{ds}	{MOSC_SHTSTBY_A, MOSC_AMPSEL_A [2:0]} or {MOSC_SHTSTBY_B, MOSC_AMPSEL_B [2:0]} SYSVCC = 4.5 V to 5.5 V	= 0000_B	1.15, 0.089	2.47, 0.163	4.06, 0.535	mS
			= 0001_B	1.80, 0.119	3.85, 0.211	6.27, 0.592	mS
			= 0010_B	2.44, 0.145	5.19, 0.254	8.41, 0.645	mS
			= 0011_B	3.05, 0.170	6.48, 0.295	10.47, 0.697	mS
			= 0100_B	4.04, 0.209	8.57, 0.358	13.80, 0.781	mS
			= 0101_B	5.17, 0.251	10.96, 0.428	17.58, 0.876	mS
			= 0110_B	6.06, 0.284	12.87, 0.482	20.61, 0.952	mS
			= 0111_B	6.91, 0.314	14.70, 0.535	23.53, 1.023	mS
			= 1000_B	4.28, 0.204	8.98, 0.365	14.44, 0.580	mS
			= 1001_B	4.85, 0.226	10.18, 0.401	16.34, 0.633	mS
			= 1010_B	5.41, 0.246	11.36, 0.435	18.20, 0.684	mS
			= 1011_B	5.95, 0.266	12.50, 0.467	20.01, 0.734	mS
			= 1100_B	6.83, 0.297	14.35, 0.522	22.96, 0.815	mS
			= 1101_B	7.82, 0.333	16.49, 0.581	26.35, 0.903	mS
			= 1110_B	8.61, 0.363	18.20, 0.627	29.08, 0.973	mS
			= 1111_B	9.37, 0.389	19.86, 0.672	31.72, 1.040	mS
X1 clock Input frequency	f_{EX}*1	EXCLK mode		15.8		40.0	MHz
X1 clock Input cycle time	t_{EXCYC}	EXCLK mode		25.0		63.1	ns
X1 High level Input voltage	V_{IH}			0.8 × SYSVCC		SYSVCC + 0.3	V
X1 Low level Input voltage	V_{IL}			–0.3		0.2 × SYSVCC	V
X1 Input leakage current	I_{LIH}	VI = SYSVCC				0.5	µA
	I_{LIL}	VI = 0 V				–0.5	µA
X1 clock Input low level pulse width	t_{EXL}	EXCLK mode	f_{EX} = 16 MHz	26			ns
			f_{EX} = 20 MHz	20			ns
			f_{EX} = 24 MHz	16			ns
			f_{EX} = 40 MHz	10			ns

Table 55.3 Main Oscillator Characteristics (3/3)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
X1 clock Input high level pulse width	t_{EXH}	EXCLK mode	f_{EX} = 16 MHz	26			ns
			f_{EX} = 20 MHz	20			ns
			f_{EX} = 24 MHz	16			ns
			f_{EX} = 40 MHz	10			ns
X1 clock rise time	t_{EXR}	EXCLK mode	f_{EX} = 16 MHz			4	ns
			f_{EX} = 20 MHz			4	ns
			f_{EX} = 24 MHz			3	ns
			f_{EX} = 40 MHz			2.5	ns
X1 clock fall time	t_{EXF}	EXCLK mode	f_{EX} = 16 MHz			4	ns
			f_{EX} = 20 MHz			4	ns
			f_{EX} = 24 MHz			3	ns
			f_{EX} = 40 MHz			2.5	ns
X1 clock Input total jitter (Dj + 14 * rms Random jitter)	t_{CITJ}	The clock source of SGMII used @ 20MHz				73*6	ps peak to peak
X1 Input capacitance	C_{X1}	Main OSC. EXCLK mode, MOSC_CAP_SEL = 0010				10	pF

Note 1. To reach internal usable clocks only following 4 frequencies are supported: 16MHz, 20MHz, 24MHz and 40MHz. Tolerance of external quartz crystal is assumed as +/–1%.

Note 2. The StartUp is supported without external components under following conditions:
- See **Section 51.12.16, OPBT10 — Option Byte 10** for default values of drivability, damping resistance and internal capacitance.
- Specification covers a maximum external stray capacitance of up to 6pF to each pin X1 and X2.
- After StartUp drivability and capacitance will be configured by OPBT10
- A possible exceeding of the recommended maximum drive level for a crystal for all start-up phases has to be agreed with the crystal manufacturers separately.

Note 3. Ccapsel_x1 (Ccapsel_x1 including parasitic capacitance to ground at the X1 side), Ccapsel_x2 (Ccapsel_x2 including parasitic capacitance to ground at the X2 side).

Note 4. The capacitor tolerance is ±15%.

Note 5. The clock source of SGMII used. Ethernet grade crystal is mandatory. Example: CX3225GA(KYOCERA)

Note 6. 12 kHz to 20 MHz rms jitter = 3 ps.

CAUTION

Oscillation stabilization times differ according to matching with the resonator. Secure an oscillation stabilization time determined through evaluation of matching.

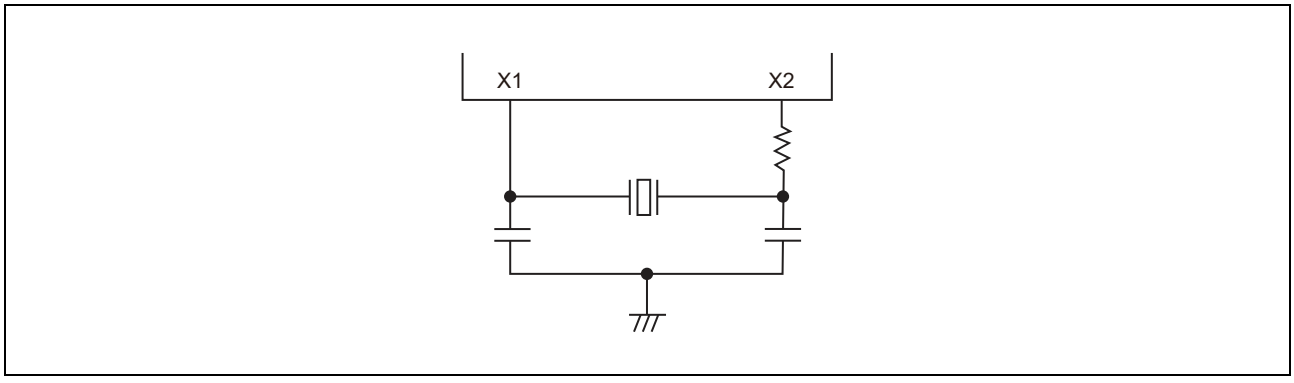


Figure 55.1 Oscillator Circuit Diagram (Crystal with External Components)

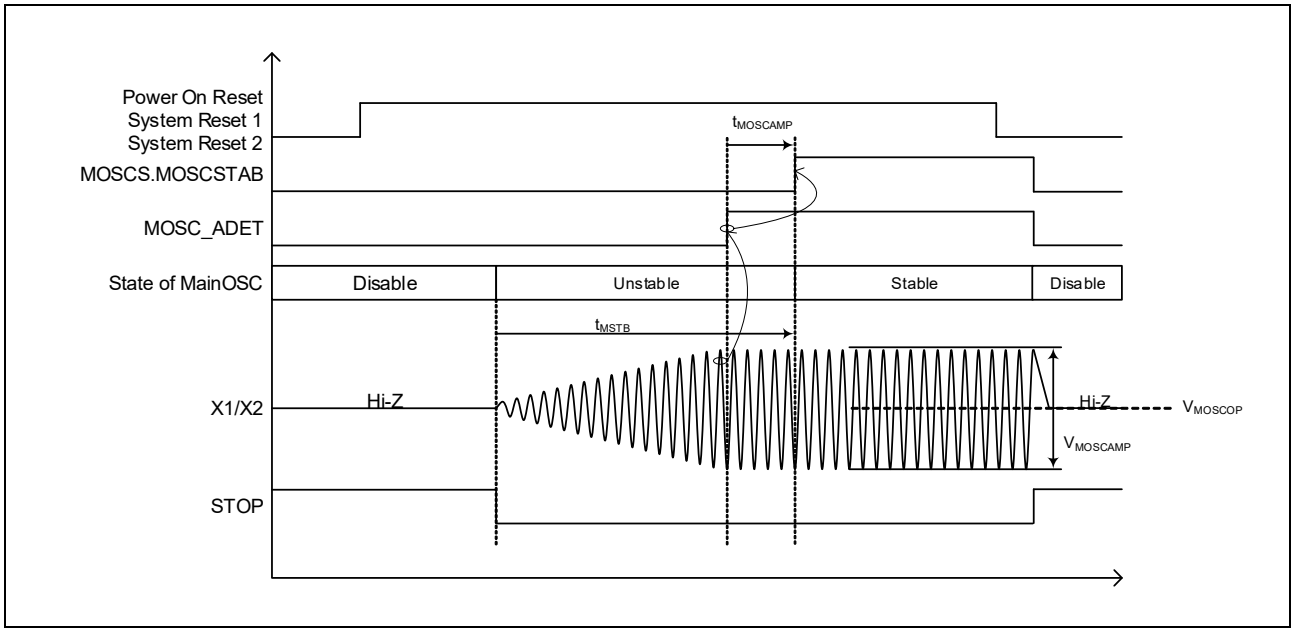


Figure 55.2 Main OSC Stabilization

55.2.1.3 Internal Oscillator Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.4 Internal Oscillator Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LS IntOSC frequency	f_{RL}		216	240	264	kHz
HS IntOSC frequency	f_{RH}		190	200	210	MHz
HV IntOSC frequency	f_{RHV}		8	16	24	MHz

55.2.1.4 PLL Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.5 PLL Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
PLL output long term jitter*1	t_{PLLLTJ}	Term = 1 µs	–500		500	ps
		Term = 10 µs	–1		1	ns
		Term = 20 µs	–2		2	ns
PLL lock time*2	t_{PLLLCT}				112	µs

Note 1. This characteristic is not tested in production.
Note 2. Lock time is time until being set “1” in PLLS.PLLCLKSTAB bit after PLLE.PLLENTRG bit is written “1”.

Table 55.6 RHSIF PLL Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
PLL lock time	$t_{RHPLLLCT}$				50	µs

55.2.1.5 Regulator Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.7 Regulator Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage	V_{AWOVCL}	AWOVCL pin	1.025	1.09	1.155	V
	$V_{GETH0VCL}$	GETH0VCL pin	1.025	1.09	1.155	V
Capacitance	C_{AWOVCL}	AWOVCL pin*1	0.140	0.2	0.286	µF
	$C_{GETH0VCL}$	GETH0VCL pin	0.154	0.22	0.286	µF
Equivalent series resistance for load capacitance	R_{VRAWO}	for C_{AWOVCL}			40*2	mΩ
	$R_{VRGETH0}$	for $C_{GETH0VCL}$			40*2	mΩ
Inrush current during power-on	$I_{RUSYSVCC}$	SYSVCC*3			350*2*4	mA
	$I_{RUGETH0PVCC}$	GETH0PVCC			100*2	mA

Note 1. The capacitors has to be mounted with one of the following connections:
- Distribute evenly the total capacitance for the device to all AWOVCL pins.
- Concentrate the total capacitance for the device to one AWOVCL pin.
In this case, the capacitor has to be connected to F23(BGA516), A22(BGA373), A18(BGA292), 126(QFP176), A13(BGA156) and 103(QFP144).
And other AWOVCL pins which capacitor not connected has to be left open.

Note 2. This is reference value.

Note 3. When returning from DeepSTOP.

Note 4. The time of current flow is less than 2.5 µs.

Table 55.8 Switching Voltage Regulator Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage	VDD		1.025	1.09	1.155	V
Switching frequency	f_{SVRSW}	FSWMODE[1:0] = 00_B	368	434	500	kHz
		FSWMODE[1:0] = 01_B	736	868	1000	kHz
SVR load transient response		Less than ΔI (max)	–33.5*1		33.5*1	mV
SVR efficiency		FSWMODE[1:0] = 00_B		80*1		%
		FSWMODE[1:0] = 01_B		70*1		%
Capacitance	$C_{SVRDRVCC}$	SVRDRVCC pin	9	14		µF
	$C_{SVRAVCC}$	SVRAVCC pin	0.7	1	1.3	µF
	C_{VDD0}	VDD pin. Close to LX		*2		µF
Equivalent series resistance for load capacitance	$R_{VRSVRDRVCC}$	for $C_{SVRDRVCC}$ (@0.1MHz to 5MHz)			10	mΩ
	$R_{VRSVRAVCC}$	for $C_{SVRAVCC}$ (@0.1MHz to 5MHz)			10	mΩ
	R_{VRVDD}	for C_{VDD0} (@0.1MHz to 5MHz)			10	mΩ
Inductance	LX	Close to external MOSFETs		*2		µH
DC resistance for inductance	R_{LX}	for LX			*2	mΩ
Inrush current during power-on	SVRDRVCC	include external circuit			4000*1	mA
Operating current variation*2*4*5	ΔI	I_{ISOVDD_R}, U2A16			440	mA/100 µs
		I_{ISOVDD_R}, U2A8			220	
		I_{ISOVDD_R}, U2A6			220	
Ramp-up time with soft start	t_{RUT}				3.6	ms

Note 1. This is reference value.
Note 2. Contact our sale office for the method of setting parameters.
Note 3. The total capacitance for the device has to be distributed around thermal ball.
Note 4. Operating current variation filtered by simple moving average of 20 µs window can be ignored.
Note 5. Operating current variation with smaller than 1 mA/µs slope is not necessary to be counted as "ΔI".

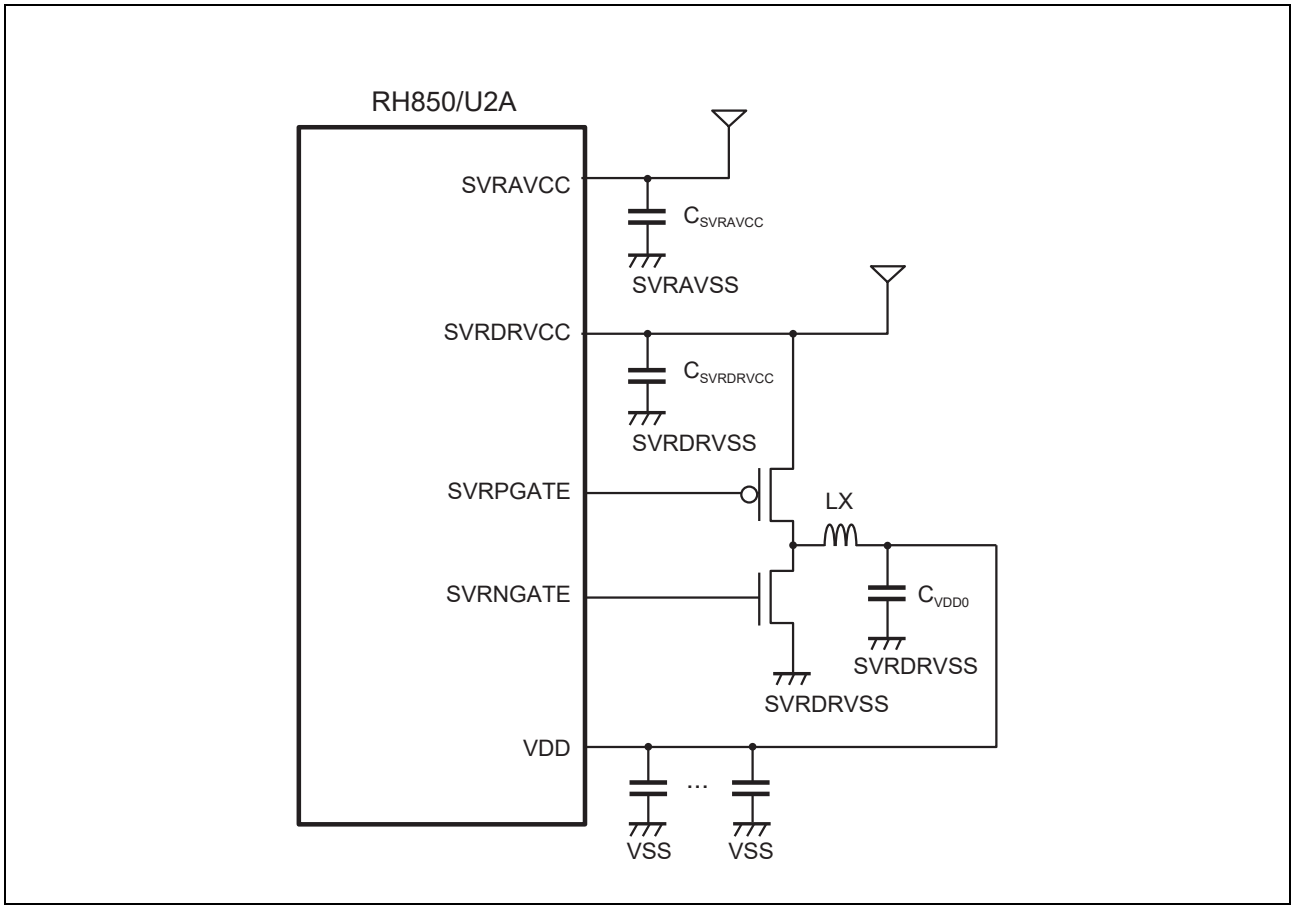


Figure 55.3 External Components of Switching Voltage Regulator

55.2.2 Input Voltage Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.9 Input Voltage Characteristics

Item	Symbol	Condition*1	MIN.	TYP.	MAX.	Unit
High level input voltage	V_{IH1}	SHMT1, IOVCC = 3.0 to 5.5 V	0.65 × IOVCC		IOVCC + 0.3	V
	V_{IH2}	SHMT2, IOVCC = 3.0 to 5.5 V	0.75 × IOVCC		IOVCC + 0.3	V
	V_{IH3}	SHMT4, IOVCC = 3.0 to 5.5 V	0.80 × IOVCC		IOVCC + 0.3	V
	V_{IH4}	TTL, IOVCC = 3.0 to 5.5 V	2.2		IOVCC + 0.3	V
Low level input voltage	V_{IL1}	SHMT1, IOVCC = 3.0 to 5.5 V	–0.3		0.35 × IOVCC	V
	V_{IL2}	SHMT2, IOVCC = 3.0 to 5.5 V	–0.3		0.25 × IOVCC	V
	V_{IL3}	SHMT4, IOVCC = 3.0 to 5.5 V	–0.3		0.50 × IOVCC	V
	V_{IL4}	TTL, IOVCC = 3.0 to 5.5 V	–0.3		0.8	V
Input hysteresis for Schmitt*3	V_{HS1}	SHMT1, IOVCC = 4.5 to 5.5 V	0.4			V
		SHMT1, IOVCC = 3.0 to 4.5 V	0.3			V
	V_{HS2}	SHMT2, IOVCC = 3.0 to 5.5 V	0.2 × IOVCC			V
	V_{HS3}	SHMT4, IOVCC = 3.0 to 5.5 V	0.1			V

Note 1. "IOVCC" means power supply voltage for I/O ports. See the appendix "*E02_01_List_of_Pin_Assignment.xlsx*" for the power supply of each pin.

Note 2. For X1 pin, see **Section 55.2.1.2, Main Oscillator Characteristics**.

Note 3. The configured drive strength of the output terminals in a power group might affect the hysteresis characteristics of the input terminals in the same power group. See Appendix file "*E02_01_List_of_Pin_Assignment.xlsx*" for the assignment of the power group.

55.2.3 Input Leakage Current

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.10 Input Leakage Current

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input leakage current	I_{LIH1}	Vin = 0 to EnVCC • Pn_m (Except P2_0, P2_5, P2_10 to P2_15, P4_5 to P4_10, P22_0 and P22_4)	Tj ≤ 105°C			0.5	µA
			105°C < Tj ≤ 150°C			1.0	µA
			150°C < Tj			2.0	µA
	I_{LIH2}	Vin = 0 to EnVCC • P2_0, P2_5, P2_10 to P2_15, P4_5 to P4_10, P22_0 and P22_4	Tj ≤ 105°C			0.5	µA
			105°C < Tj			2.0	µA
	I_{LIH3}	Vin = 0 to AnVCC • APn_m	Tj ≤ 150°C			0.5	µA
			150°C < Tj			1.0	µA
	I_{LIL1}	Vin = 0 to EnVCC • Pn_m (Except P2_0, P2_5, P2_10 to P2_15, P4_5 to P4_10, P22_0 and P22_4)	Tj ≤ 105°C			–0.5	µA
			105°C < Tj ≤ 150°C			–1.0	µA
			150°C < Tj			–2.0	µA
	I_{LIL2}	Vin = 0 to EnVCC • P2_0, P2_5, P2_10 to P2_15, P4_5 to P4_10, P22_0 and P22_4	Tj ≤ 105°C			–0.5	µA
			105°C < Tj			–2.0	µA
	I_{LIL3}	Vin = 0 to AnVCC • APn_m	Tj ≤ 150°C			–0.5	µA
			150°C < Tj			–1.0	µA

55.2.4 Pull-up/Pull-down Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.11 Pull-up/Pull-down Characteristics

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input pull-up current source	I_{PU1}		IOVCC*1 = 3.0 to 5.5V, Vin = 0V	30		350	µA
Input pull-down current source	I_{PD1}	Other than the below	IOVCC*1 = 3.0 to 5.5V, Vin = IOVCC*1	30		350	µA
	I_{PD2}	$\overline{\text{RESET}}$, $\overline{\text{AURORES1}}$, $\overline{\text{AURORES2}}$, $\overline{\text{AURORESPD}}$, $\overline{\text{ERAMRES2}}$, $\overline{\text{ERAMRESPD}}$	IOVCC*1 = 3.0 to 5.5V, Vin = IOVCC*1			110	µA

Note 1. "IOVCC" means power supply voltage for I/O ports. See the appendix "*E02_01_List_of_Pin_Assignment.xlsx*" for the power supply of each pin.

55.2.5 Output Voltage Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.12 Output Voltage Characteristics*1

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Output resistance	R_{O11}	Drive strength = 1	IOVCC – V_{OH} = 0.4 V, V_{OL} = 0.4 V	6		40	Ω
	R_{O12}	Drive strength = 2	IOVCC – V_{OH} = 0.4 V, V_{OL} = 0.4 V	11		72	Ω
	R_{O13}	Drive strength = 3	IOVCC – V_{OH} = 0.4 V, V_{OL} = 0.4 V	18		100	Ω
	R_{O14}	Drive strength = 4	IOVCC – V_{OH} = 0.4 V, V_{OL} = 0.4 V	32		200	Ω
	R_{O15}	Drive strength = 5	IOVCC – V_{OH} = 0.4 V, V_{OL} = 0.4 V	61		400	Ω
High level output voltage	V_{OH1}	Drive strength = 1 IOVCC = 3.0 to 5.5 V	I_{OH} = –4 mA / pin (4 output pins)*2	IOVCC–0.8			V
			I_{OH}= –2 mA / pin (4 output pins)*2	IOVCC–0.5			V
	V_{OH2}	Drive strength = 2 IOVCC = 3.0 to 3.6 V	I_{OH} = –4 mA / pin (8 output pins)*2	IOVCC–0.8			V
			I_{OH}= –2 mA / pin (8 output pins)*2	IOVCC–0.5			V
		Drive strength = 2 IOVCC = 3.6 to 5.5 V	I_{OH} = –4 mA / pin (2 output pins)*2	IOVCC–0.8			V
			I_{OH}= –2 mA / pin (2 output pins)*2	IOVCC–0.5			V
	V_{OH3}	Drive strength = 3 IOVCC = 3.0 to 5.5 V	I_{OH} = –4 mA / pin (8 output pins)*2	IOVCC–0.8			V
			I_{OH} = –2 mA / pin (8 output pins)*2	IOVCC–0.5			V
	V_{OH4}	Drive strength = 4 IOVCC = 3.0 to 5.5 V	I_{OH} = –2 mA / pin (16 output pins)*2	IOVCC–0.7			V
			I_{OH} = –1 mA / pin (16 output pins)*2	IOVCC–0.5			V
	V_{OH5}	Drive strength = 5 IOVCC = 3.0 to 5.5 V	I_{OH} = –1 mA / pin (16 output pins)*2	IOVCC–0.7			V
			I_{OH} = –500 µA / pin (16 output pins)*2	IOVCC–0.5			V
Low level output voltage	V_{OL1}	Drive strength = 1 IOVCC = 3.0 to 5.5 V	I_{OL} = 4 mA / pin (4 output pins)*2			0.8	V
			I_{OL} = 2 mA / pin (4 output pins)*2			0.5	V
	V_{OL2}	Drive strength = 2 IOVCC = 3.0 to 3.6 V	I_{OL} = 4 mA / pin (8 output pins)*2			0.8	V
			I_{OL} = 2 mA / pin (8 output pins)*2			0.5	V
		Drive strength = 2 IOVCC = 3.0 to 5.5 V	I_{OL} = 4 mA / pin (2 output pins)*2			0.8	V
			I_{OL} = 2 mA / pin (2 output pins)*2			0.5	V
	V_{OL3}	Drive strength = 3 IOVCC = 3.0 to 5.5 V	I_{OL} = 4 mA / pin (8 output pins)*2			0.7	V
			I_{OL} = 2 mA / pin (8 output pins)*2			0.5	V
	V_{OL4}	Drive strength = 4 IOVCC = 3.0 to 5.5 V	I_{OL} = 2 mA / pin (16 output pins)*2			0.7	V
			I_{OL} = 1 mA / pin (16 output pins)*2			0.5	V
	V_{OL5}	Drive strength = 5 IOVCC = 3.0 to 5.5 V	I_{OL} = 1 mA / pin (16 output pins)*2			0.7	V
			I_{OL} = 500 µA / pin (16 output pins)*2			0.5	V

Note 1. "IOVCC" means power supply voltage for I/O ports. See the appendix "*E02_01_List_of_Pin_Assignment.xlsx*" for the power supply of each pin.

Note 2. The number of pin indicates simultaneous ON in one power. The influence of the noise emission should be considered when switching the output level.

55.2.6 Output Current

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.13 Output Current Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
High level output current	$\|I_{OH}\|$	per pin			8	mA
	$\Sigma\|I_{OH}\|$ EnVCC	Each EnVCC			60	mA
	$\Sigma\|I_{OH}\|$ AnVCC	Each AnVCC			16	mA
Low level output current	I_{OL}	per pin			8	mA
	ΣI_{OL} EnVCC	Each EnVCC			60	mA
	ΣI_{OL} AnVCC	Each AnVCC			16	mA

55.2.7 Injection Current Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.14 Injection Current Operating Conditions

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DC injection current (per pin)	I_{INJ_DIN}	Digital pin *1	–2		3	mA
	I_{INJ_AIN}	Analogue pin	–3		3	mA
DC injection current (each power supply)	I_{INJ_E0VCC}	E0VCC			50	mA
	I_{INJ_E1VCC}	E1VCC			50	mA
	I_{INJ_E2VCC}	E2VCC			30	mA
	I_{INJ_A0VCC}	A0VCC			20	mA
	I_{INJ_A1VCC}	A1VCC			20	mA
	I_{INJ_A2VCC}	A2VCC			20	mA
DC injection current (total)	I_{INJ_TOTAL}				80	mA

Note 1. Injection current to the logic pin multiplied with LVDS function is prohibited when LVDS function is used. When LVDS function is not used, Injection current to the logic pin multiplied with LVDS function causes at maximum additional 1 µA leakage current at the P/N combinational pin. For example, injection current to HSIF0_TXDP causes at maximum additional 1 µA leakage current at HSIF0_TXDN.

NOTE

- Injection current to P3_8 pin is prohibited when SGMII function is used.
- Input voltage must be kept within –0.8V ≤ Vin ≤ 6.0V, if injection current is kept within rated value.
- Power supply voltage must be kept within operating conditions.
- Injection current effects power dissipation in the package for thermal characteristics.

55.2.8 LVDS Characteristics

55.2.8.1 LVDS Driver Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.15 LVDS Driver Characteristics (based on IEEE 1596.3-1996)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output low-level voltage	V_{LVDSOL}		1000			mV
Output high-level voltage	V_{LVDSOH}				1400	mV
Output differential voltage	V_{LVDSOD}		150		250	mV
Offset voltage	V_{LVDSOS}		1.125	1.2	1.275	V
Output impedance	R_{LVDSOI}		40		300	Ω

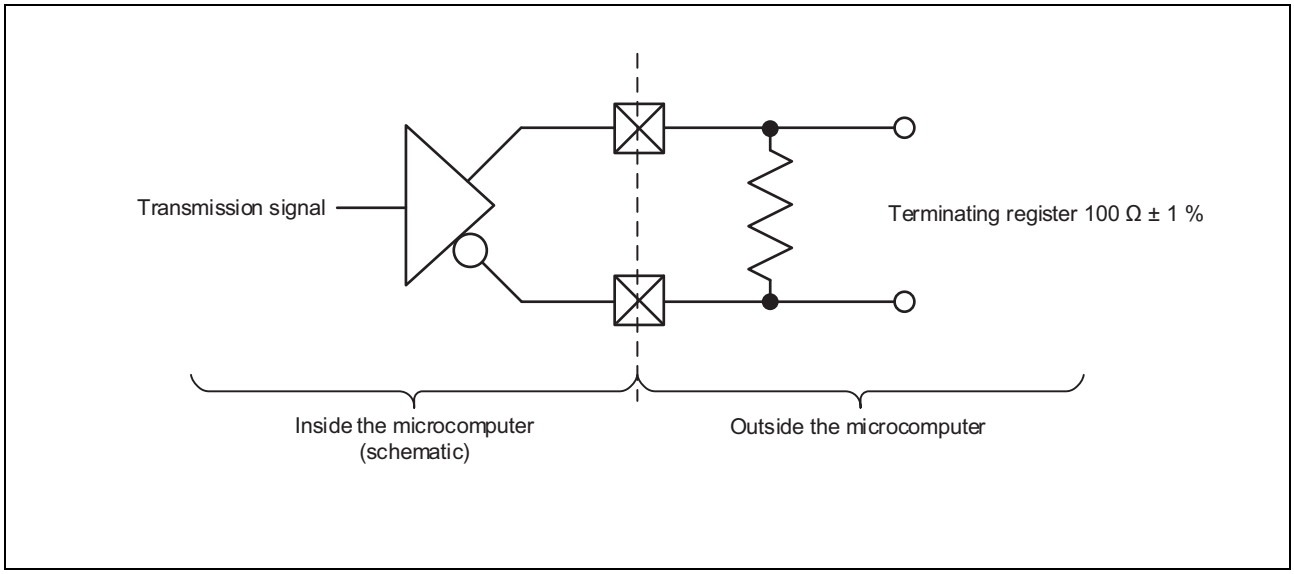


Figure 55.4 LVDS Driver Measurement Conditions

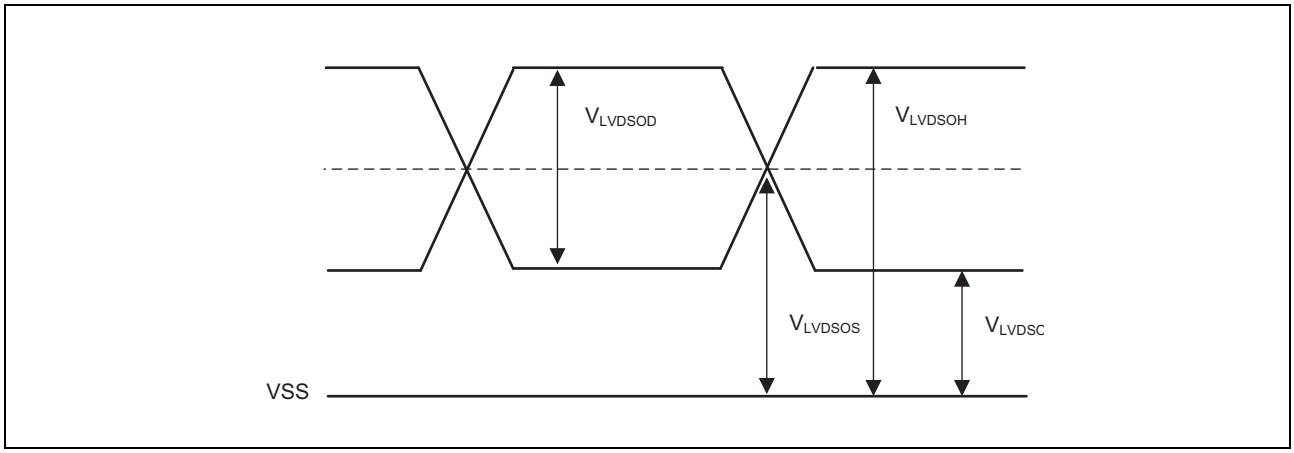


Figure 55.5 Meaning of LVDS Driver Symbols

55.2.8.2 LVDS Receiver Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.16 LVDS Receiver Characteristics (based on IEEE 1596.3-1996)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Differential Input threshold voltage	$V_{LVDSIDTH}$		–100		100	mV
Input voltage range	V_{LVDI}		0.825		1.575	V

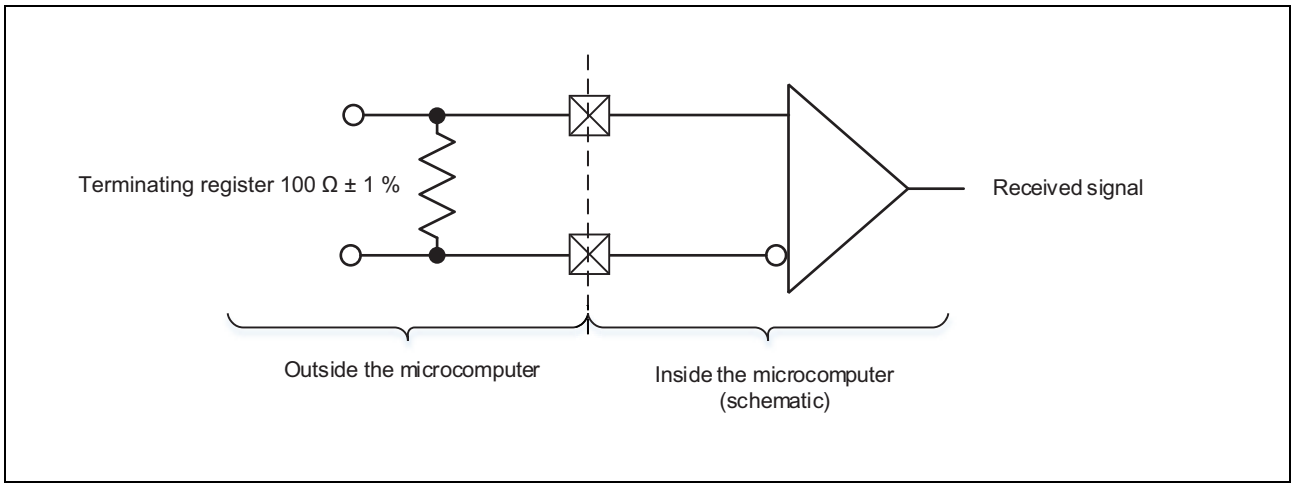


Figure 55.6 LVDS Receiver Measurement Conditions

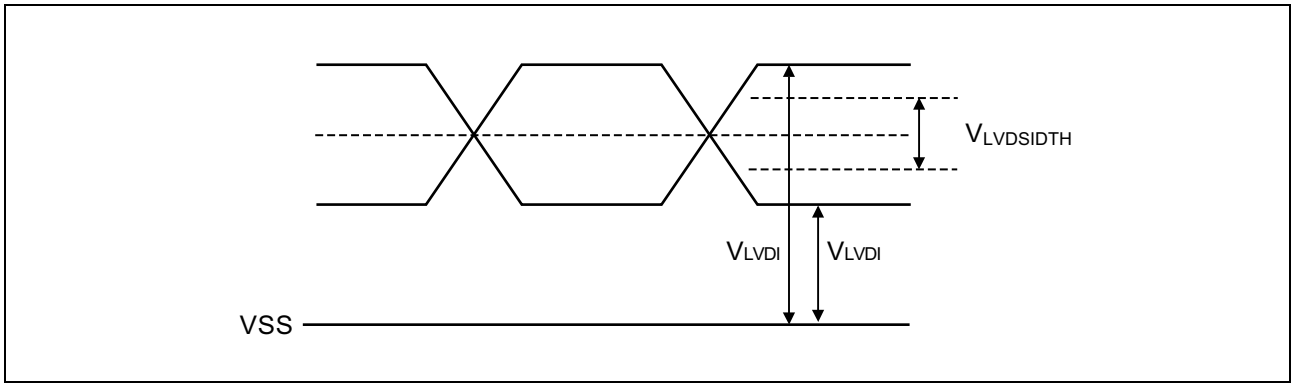


Figure 55.7 Meaning of LVDS Receiver Symbols

55.2.9 SGMII Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.17 SGMII REFCK Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
High level input voltage	V_{IH}		2.0		GETH0RVCC + 0.3	V
Low level input voltage	V_{IL}		–0.3		0.8	V

Table 55.18 SGMII Tx Buffer Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage high	V_{OH}				1525	mV
Output voltage low	V_{OL}		875			mV
Output Differential Voltage	$\|V_{OD}\|$		150		400	mV
Output Offset Voltage	V_{OS}		1075		1325	mV
Change in V_{OD} between “0” and “1”	$\Delta\|V_{OD}\|$				25	mV
Change in V_{OS} between “0” and “1”	ΔV_{OS}				25	mV
Output current on Short to GND	I_{sa}, I_{sb}				40	mA
Output current when a, b are shorted	I_{sab}				12	mA

Note 1. For details of the symbols, refer to the IEEE1596.3-1996 standard.
Note 2. All parameters measured at Rload = 100 Ω ±1 % load

Table 55.19 SGMII Rx Buffer Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input Voltage range a or b	V_I		675		1725	mV
Input differential threshold	V_{idth}		–50		50	mV
Receiver differential input impedance	R_{in}		80		120	Ω

Note 1. For details of the symbols, refer to the IEEE1596.3-1996 standard.

55.2.10 Aurora Interface Clock Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.20 Aurora Interface Clock Characteristics (CICREFP, CICREFN)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Differential input voltage	$V_{DIFFCIC}$ *1		200		1600	mV
External AC coupling capacitor	C_{ACC}		75	100	200	nF
Differential input resistance	$Z_{DIFFCIC}$		70	100	130	Ω

Note 1. Peak to peak differential input voltage.

Table 55.21 Aurora Interface Characteristics (TODP0-3/TODN0-3)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Differential output voltage	$V_{DIFFTOD}$ *1		800		1600	mV
Differential output resistance	$Z_{DIFFTOD}$		70		130	Ω

Note 1. Peak to peak differential input voltage.

55.2.11 IO Capacitances

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.22 IO Capacitances *1, *2

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f = 1 MHz 0 V for non measurement pins			10	pF
Input/output capacitance	C_{IO}				10	pF
Output capacitance	C_O				10	pF

Note 1. This capacity is the capacity value per BGA_Ball (QFP_Pin) 1 terminal.

Note 2. For analog input pins (ADCJnIm and ADCJnImS), refer to **Section 55.4, A/D Converter Characteristics**.

55.2.12 Supply Current Characteristics

55.2.12.1 General Definition

Power consumption except for the current of I/O buffer (I_{EnVCC}) is specified below.

The parameter I_{EnVCC} depends on customer's application and thus need to be defined by customer in order to determine the total power dissipation of a device.

55.2.12.2 Power Supply Currents

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.23 Current consumption for RH850/U2A-EVA*2, *3

Item	Symbol	Power supply	CPU frequency	MIN.	TYP.*1	MAX.	Unit
RUN mode current	I_{SYSVCC_R}	SYSVCC	400 MHz		6	34	mA
			320 MHz		6	34	mA
			240 MHz		6	34	mA
	I_{VCC_R}	VCC*5	400 MHz		22	45	mA
			320 MHz		22	45	mA
			240 MHz		22	45	mA
	I_{ISOVDD_R}	ISOVDD	400 MHz		480	1870	mA
			320 MHz		405	1694	mA
			240 MHz		330	1518	mA
STOP mode current*6	I_{SYSVCC_S}	SYSVCC			3	20	mA
	I_{VCC_S}	VCC			0.005	1	mA
	I_{ISOVDD_S}	ISOVDD			385	1300	mA
DeepSTOP mode current*6	I_{SYSVCC_DS}	SYSVCC			0.52	12	mA
	I_{VCC_DS}	VCC			2	400	µA
	I_{ISOVDD_DS}	ISOVDD			14*4	900*4	mA
Cyclic RUN mode current*6	I_{SYSVCC_CR}	SYSVCC	100 MHz		4	32	mA
	I_{VCC_CR}	VCC	100 MHz		0.008	1.2	mA
	I_{ISOVDD_CR}	ISOVDD	100 MHz		405	1350	mA
Cyclic STOP mode current*6	I_{SYSVCC_CS}	SYSVCC			3	20	mA
	I_{VCC_CS}	VCC			0.005	1	mA
	I_{ISOVDD_CS}	ISOVDD			385	1300	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- SYSVCC = VCC = EnVCC = LVDVCC = AnVCC = AnVREFH = SVRDRVCC = SVRAVCC = 5.0 V
- GETH0PVCC = GETH0BVCC = GETH0RVCC = 3.3 V
- ISOVDD = 1.09 V

Note 2. The above value does not include the current of SVR converter.

Note 3. The above value does not include the current of I/O buffer.

Note 4. Indicated leak current will be applied if the power continues supplying to ISOVDD during DeepSTOP mode.

Note 5. Additional current (max.60 mA for Code Flash per unit, max.35 mA for Data Flash per unit) will be applied when data writing/erasing to Code or Data Flash is processing.

Note 6. MainOSC and all peripherals are stopped.

Table 55.24 Current consumption for RH850/U2A-EVA (specific power supply)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
EMUVCC	I_{EMUVCC}					20	mA
DVCC	I_{DVCC}					20	mA
ERAMVCC	$I_{ERAMVCC}$					5	mA
EMUVDD	I_{EMUVDD}					20	mA
DVDD	I_{DVDD}					350	mA
ERAMVDD	$I_{ERAMVDD}$					250	mA

Table 55.25 Current consumption for RH850/U2A16[*2, *3]

Item	Symbol	Power supply	CPU frequency	MIN.	TYP.[*1]	MAX.	Unit
RUN mode current	I_{SYSVCC_R}	SYSVCC	400 MHz		6	34	mA
			320 MHz		6	34	mA
			240 MHz		6	34	mA
	I_{VCC_R}	VCC[*5]	400 MHz		11	35	mA
			320 MHz		11	35	mA
			240 MHz		11	35	mA
	I_{ISOVDD_R}	ISOVDD	400 MHz		420	1650	mA
			320 MHz		355	1483	mA
			240 MHz		290	1316	mA
STOP mode current[*6]	I_{SYSVCC_S}	SYSVCC			3	20	mA
	I_{VCC_S}	VCC			0.005	1	mA
	I_{ISOVDD_S}	ISOVDD			335	1130	mA
DeepSTOP mode current[*6]	I_{SYSVCC_DS}	SYSVCC			0.52	12	mA
	I_{VCC_DS}	VCC			2	400	µA
	I_{ISOVDD_DS}	ISOVDD			12[*4]	730[*4]	mA
Cyclic RUN mode current[*6]	I_{SYSVCC_CR}	SYSVCC	100 MHz		4	32	mA
	I_{VCC_CR}	VCC	100 MHz		0.008	1.2	mA
	I_{ISOVDD_CR}	ISOVDD	100 MHz		350	1180	mA
Cyclic STOP mode current[*6]	I_{SYSVCC_CS}	SYSVCC			3	20	mA
	I_{VCC_CS}	VCC			0.005	1	mA
	I_{ISOVDD_CS}	ISOVDD			335	1130	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- SYSVCC = VCC = EnVCC = LVDVCC = AnVCC = AnVREFH = SVRDRVCC = SVRAVCC = 5.0 V
- GETH0PVCC = GETH0BVCC = GETH0RVCC = 3.3 V
- ISOVDD = 1.09 V

Note 2. The above value does not include the current of SVR converter.

Note 3. The above value does not include the current of I/O buffer.

Note 4. Indicated leak current will be applied if the power continues supplying to ISOVDD during DeepSTOP mode.

Note 5. Additional current (max.60 mA for Code Flash per unit, max.35 mA for Data Flash per unit) will be applied when data writing/erasing to Code or Data Flash is processing.

Note 6. MainOSC is stopped.

Table 55.26 Current consumption for RH850/U2A8[*2, *3]

Item	Symbol	Power supply	CPU frequency	MIN.	TYP.[*1]	MAX.	Unit
RUN mode current	I_{SYSVCC_R}	SYSVCC	400 MHz		6	30	mA
			320 MHz		6	30	mA
			240 MHz		6	30	mA
	I_{VCC_R}	VCC[*5]	400 MHz		9	28	mA
			320 MHz		9	28	mA
			240 MHz		9	28	mA
	I_{ISOVDD_R}	ISOVDD	400 MHz		245	870	mA
			320 MHz		210	780	mA
			240 MHz		175	690	mA
STOP mode current[*6]	I_{SYSVCC_S}	SYSVCC			3	16	mA
	I_{VCC_S}	VCC			0.005	0.6	mA
	I_{ISOVDD_S}	ISOVDD			170	580	mA
DeepSTOP mode current[*6]	I_{SYSVCC_DS}	SYSVCC			0.46	8	mA
	I_{VCC_DS}	VCC			2	320	µA
	I_{ISOVDD_DS}	ISOVDD			5[*4]	385[*4]	mA
Cyclic RUN mode current[*6]	I_{SYSVCC_CR}	SYSVCC	100 MHz		4	28	mA
	I_{VCC_CR}	VCC	100 MHz		0.008	0.8	mA
	I_{ISOVDD_CR}	ISOVDD	100 MHz		180	620	mA
Cyclic STOP mode current[*6]	I_{SYSVCC_CS}	SYSVCC			3	16	mA
	I_{VCC_CS}	VCC			0.005	0.6	mA
	I_{ISOVDD_CS}	ISOVDD			170	580	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- SYSVCC = VCC = EnVCC = LVDVCC = AnVCC = AnVREFH = SVRDRVCC = SVRAVCC = 5.0 V
- GETH0PVCC = GETH0BVCC = GETH0RVCC = 3.3 V
- ISOVDD = 1.09 V

Note 2. The above value does not include the current of SVR converter.

Note 3. The above value does not include the current of I/O buffer.

Note 4. Indicated leak current will be applied if the power continues supplying to ISOVDD during DeepSTOP mode.

Note 5. Additional current (max.60 mA for Code Flash per unit, max.35 mA for Data Flash per unit) will be applied when data writing/erasing to Code or Data Flash is processing.

Note 6. MainOSC is stopped.

Table 55.27 Current consumption for RH850/U2A6[*2, *3]

Item	Symbol	Power supply	CPU frequency	MIN.	TYP.[*1]	MAX.	Unit
RUN mode current	I_{SYSVCC_R}	SYSVCC	400 MHz		6	30	mA
			320 MHz		6	30	mA
			240 MHz		6	30	mA
	I_{VCC_R}	VCC[*5]	400 MHz		9	28	mA
			320 MHz		9	28	mA
			240 MHz		9	28	mA
	I_{ISOVDD_R}	ISOVDD	400 MHz		240	840	mA
			320 MHz		207	755	mA
			240 MHz		173	670	mA
STOP mode current[*6]	I_{SYSVCC_S}	SYSVCC			3	16	mA
	I_{VCC_S}	VCC			0.005	0.6	mA
	I_{ISOVDD_S}	ISOVDD			168	570	mA
DeepSTOP mode current[*6]	I_{SYSVCC_DS}	SYSVCC			0.46	8	mA
	I_{VCC_DS}	VCC			2	320	µA
	I_{ISOVDD_DS}	ISOVDD			4[*4]	375[*4]	mA
Cyclic RUN mode current[*6]	I_{SYSVCC_CR}	SYSVCC	100 MHz		4	28	mA
	I_{VCC_CR}	VCC	100 MHz		0.008	0.8	mA
	I_{ISOVDD_CR}	ISOVDD	100 MHz		178	610	mA
Cyclic STOP mode current[*6]	I_{SYSVCC_CS}	SYSVCC			3	16	mA
	I_{VCC_CS}	VCC			0.005	0.6	mA
	I_{ISOVDD_CS}	ISOVDD			168	570	mA

Note 1. The condition of “TYP.” shows the specification with the following conditions. Also, the value is just for reference only.
- Tj = 25°C
- SYSVCC = VCC = EnVCC = AnVCC = AnVREFH = SVRDRVCC = SVRAVCC = 5.0 V
- ISOVDD = 1.09 V

Note 2. The above value does not include the current of SVR converter.

Note 3. The above value does not include the current of I/O buffer.

Note 4. Indicated leak current will be applied if the power continues supplying to ISOVDD during DeepSTOP mode.

Note 5. Additional current (max.60 mA for Code Flash per unit, max.35 mA for Data Flash per unit) will be applied when data writing/erasing to Code or Data Flash is processing.

Note 6. MainOSC is stopped.

55.2.12.3 Power Supply Currents for specific features

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.28 Current consumption for specific features*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SVR converter	I_{SVR}	Current of SYSVCC			4	mA
	I_{SVRA}	Current of SVRAVCC			12	mA
LVDS for RHSIF/MSPI	I_{LVDS}	Current of LVDVCC			16	mA
Gigabit Ethernet	I_{GBETH}	Current of GETH0BVCC, GETH0RVCC and GETH0PVCC			90	mA
ADCJ0	I_{ADC0}*2	Current of A0VCC 4ch T&H function is used			5.7	mA
	$I_{ADC0REF}$*2	Current of A0VREFH			0.2	mA
ADCJ1	I_{ADC1}*2	Current of A1VCC 4ch T&H function is used			5.7	mA
	$I_{ADC1REF}$*2	Current of A1VREFH			0.2	mA
ADCJ2	I_{ADC2}*2	Current of A2VCC			2.9	mA
	$I_{ADC2REF}$*2	Current of A2VREFH			0.2	mA

Note 1. This table shows additional current when the specific function indicated above is used.

Note 2. For QFP package and BGA156 package, the current of AnVREFH (IADCnREF) is total of AnVREFH current (IADCnREF) and AnVCC current (IADCn).

55.2.13 Voltage Detector (POC, VLVI) Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.29 Voltage Detector (POC, VLVI) Characteristics*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Detection voltage (SYSVCC)	V_{POC}		2.5	2.63	2.75	V
	V_{VLVI}		1.9	2.0	2.1	V
Response time	t_{DPOC1}	Rise			1.2	ms
	t_{DPOC2}	Fall			10	µs
SYSVCC minimum pulse width	t_{WPOC}		0.2			ms
SYSVCC voltage ramp	t_{SYSVS}		0.002		550	ms/V

Note 1. POC monitors SYSVCC supply voltage.

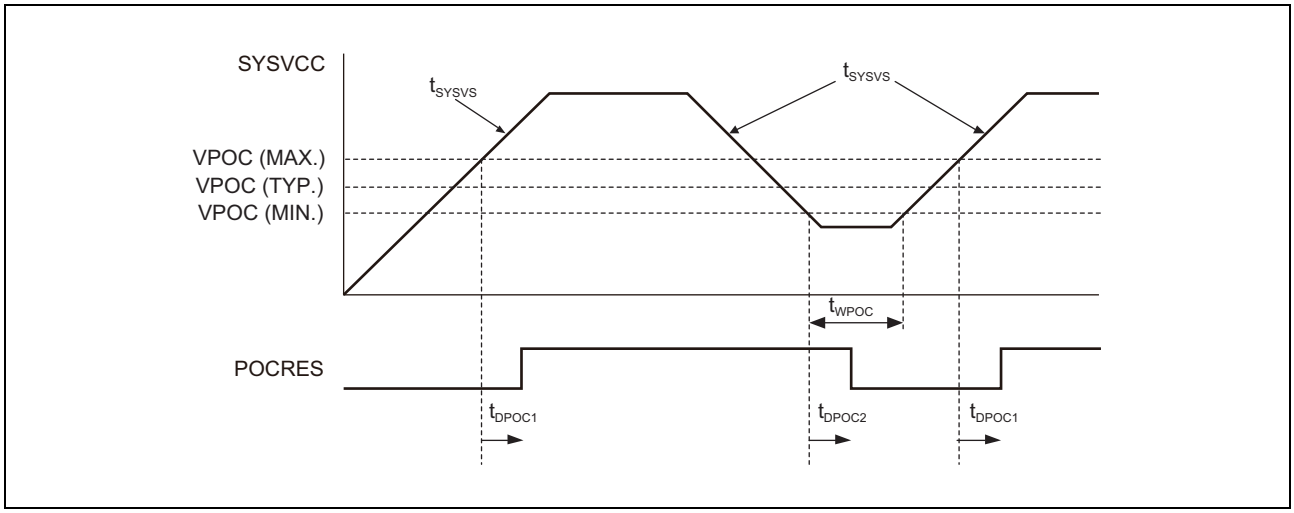


Figure 55.8 POC Characteristics

55.2.14 VMON Characteristics

Conditions:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.

Table 55.30 VMON Characteristics*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
VDD primary high detection level	V_{VDDMAH}	AWOVDD	1.155	1.17	1.205	V
	V_{VDDMIH}	ISOVDD	1.155	1.17	1.205	V
VDD primary low detection level	V_{VDDMAL}	AWOVDD	0.975	1.01	1.025	V
	V_{VDDMIL}	ISOVDD Enable assist by Delay Monitor (DMON)	0.985	1.01	1.025	V
VCC primary high detection level	V_{VCCMH}		5.5	5.64	5.8	V
VCC primary low detection level	V_{VCCML}		2.8	2.9	3	V
E0VCC primary high detection level	V_{EVCCMH}		5.5	5.64	5.8	V
E0VCC primary low detection level	V_{EVCCML}		2.8	2.9	3	V
$\overline{\text{VMONOUT}}$ delay time	t_{DVMON}				10+Filter time*1	µs
VMON minimum pulse width	t_{WVMON}		0.2			ms
Voltage ramp	t_{VS}		0.002		550	ms/V

Note 1. See **Section 11.3.6, Registers** for details specification of filter time.

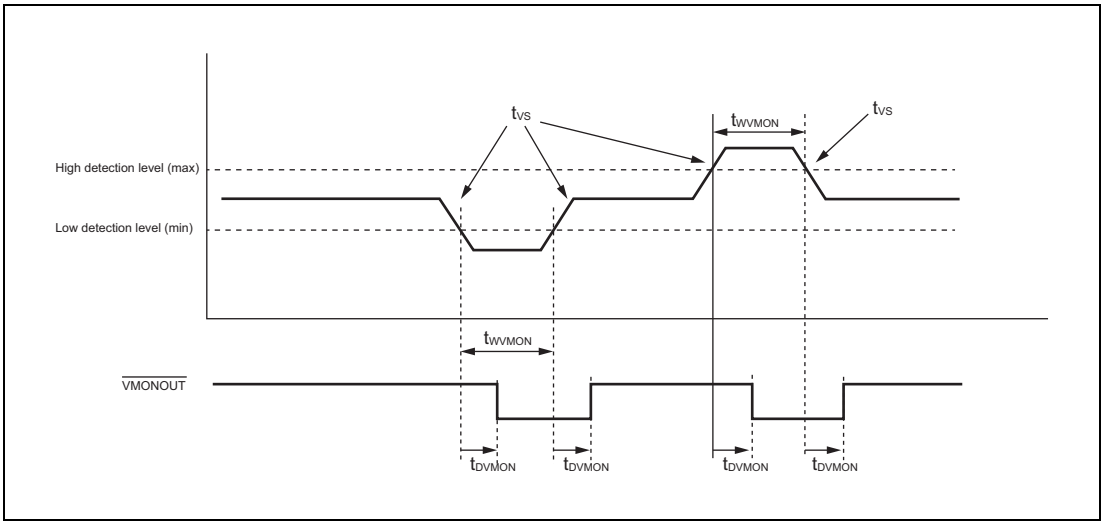


Figure 55.9 VMON Characteristics

55.3 AC Characteristics

55.3.1 AC Characteristic Measurement Condition

55.3.1.1 General Conditions

Below conditions are valid for all subsequent timing specifications if not noted otherwise:

- See **Section 55.2.1.1, Supply Voltage Characteristics**.
- Drive strength = 3
- CL = 30 pF
- All bits of PINVn/JPINV0 are set as 0.
- All bits of PODCn/JPODC0 are set as 0 except for specification on RIIC timing.

NOTE

Even though AC characteristics correspond to the nominal frequency, a main oscillator tolerance of up to 1000 ppm is considered with regard to timing characteristics for communication modules.

55.3.1.2 Input Measurement Points

If not stated otherwise, the below given AC timing specification is based on the measurements points as follows:

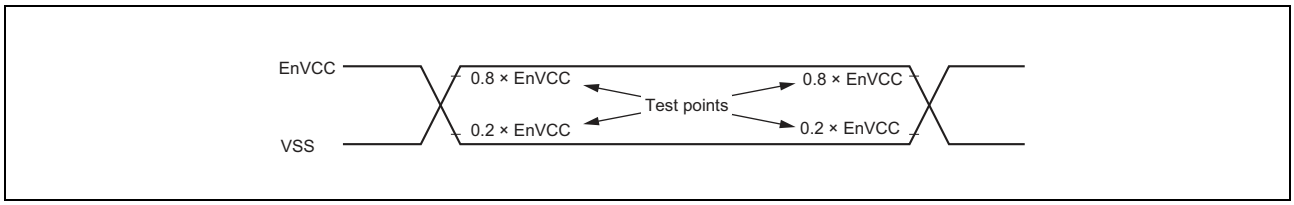


Figure 55.10 AC Input Measurement Points

55.3.1.3 Output Measurement Points

If not stated otherwise, the below given AC timing specification is based on the measurements points as follows:

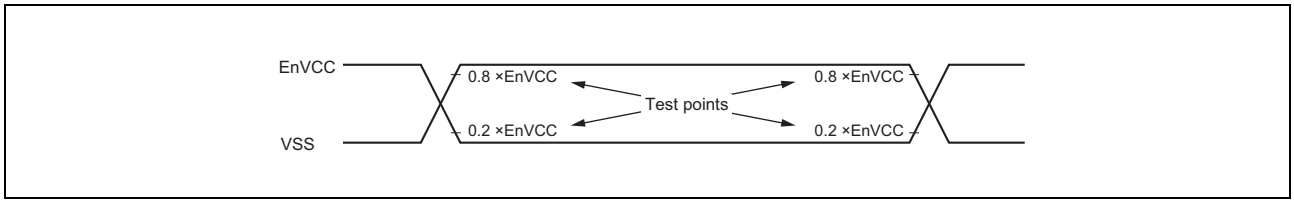


Figure 55.11 AC Output Measurement Points

55.3.1.4 Load conditions

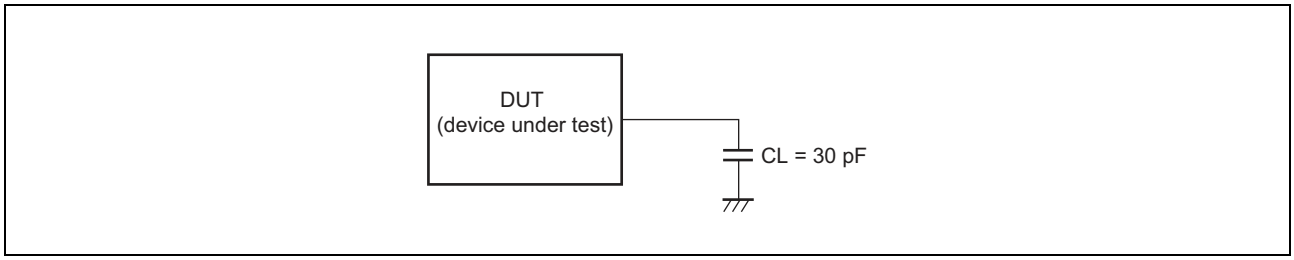


Figure 55.12 AC Load Conditions

55.3.2 Power On/Off Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.31 Power On/Off Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Reset hold time at power-on	t_{RESH1}*6	Power-up*1 SVR is used	1.3 + t_{RUT}			ms
		Power-up*1 SVR is not used	1.3			ms
Power hold time at reset assertion*5	t_{PWH}	Power-down*2 (ETNB1 (SGMII) used)	64			µs
		Power-down*2 (ETNB1 (SGMII) not used)	10.5			µs
Operation mode low level hold time at power-on	t_{MDL1}*6	SVR is used	0.1 + t_{RUT}			ms
		SVR is not used	0.1			ms
Operating mode setup time at power-on	t_{MDS1}		1			ms
Operating mode setup time at reset assert	t_{MDS2}		1			ms
Operating mode hold time at reset negate	t_{MDH1}		1			ms
Operating mode hold time at power-off	t_{MDH2}		0			µs
$\overline{\text{TRST}}$ setup time at reset	t_{TRMDS}		2			µs
$\overline{\text{TRST}}$ hold time at reset negate*4	t_{TRMDH}		30			ms
$\overline{\text{TRST}}$ hold time at power-on	t_{TRSTH1}*6	SVR is used	1.29 + t_{RUT}			ms
		SVR is not used	1.29			ms
$\overline{\text{TRST}}$ hold time at power-off	t_{TRSTH3}	Time since SYSVCC and VDD were power-off.			10	µs
Oscillator stabilization time	t_{OSC}				5.5	ms
PLL lock in time	t_{PLLL}	*3			1	ms
EMUVDD hold time at power-on	$t_{EMUVDDH1}$		0			ms
EMUVDD setup time at power-off	$t_{EMUVDDS1}$		0			ms

Note 1. t_{RESH1} is the reset time required for the supply of internal clock signals to become stable after power supplies are turned on. There are no restrictions on the rising order of each power supply.

Note 2. t_{PWH} is the time from assertion of the reset signal until any of the power voltages have dropped below the lower-limit voltages.
There are no restrictions on the falling order of each power supply.

Note 3. t_{PLLL} is the time required for PLL to lock in after MOSC oscillation has become stable.

Note 4. Access by the Nexus, LPD and BSCAN during t_{TRMDH} duration is prohibited (both of High and low condition of $\overline{\text{TRST}}$).

Note 5. The device can withstand up to 1000 uncontrolled power down cycles without impact on lifetime.
Uncontrolled means not according to power down timing requirements.

Note 6. Objection of power supplies for t_{RESH1}, t_{TRSTH1} and t_{MDL1} is changed by power configuration.
When SVR is used:All power supplies except for ISOVDD
When SVR is not used: All power supplies

CAUTION

The states of I/O pins are not reset during the noise cancellation interval of the reset signal following its assertion while power is being turned off. During that time, do not allow any input of mid-range potential to the pin or contention of output data.

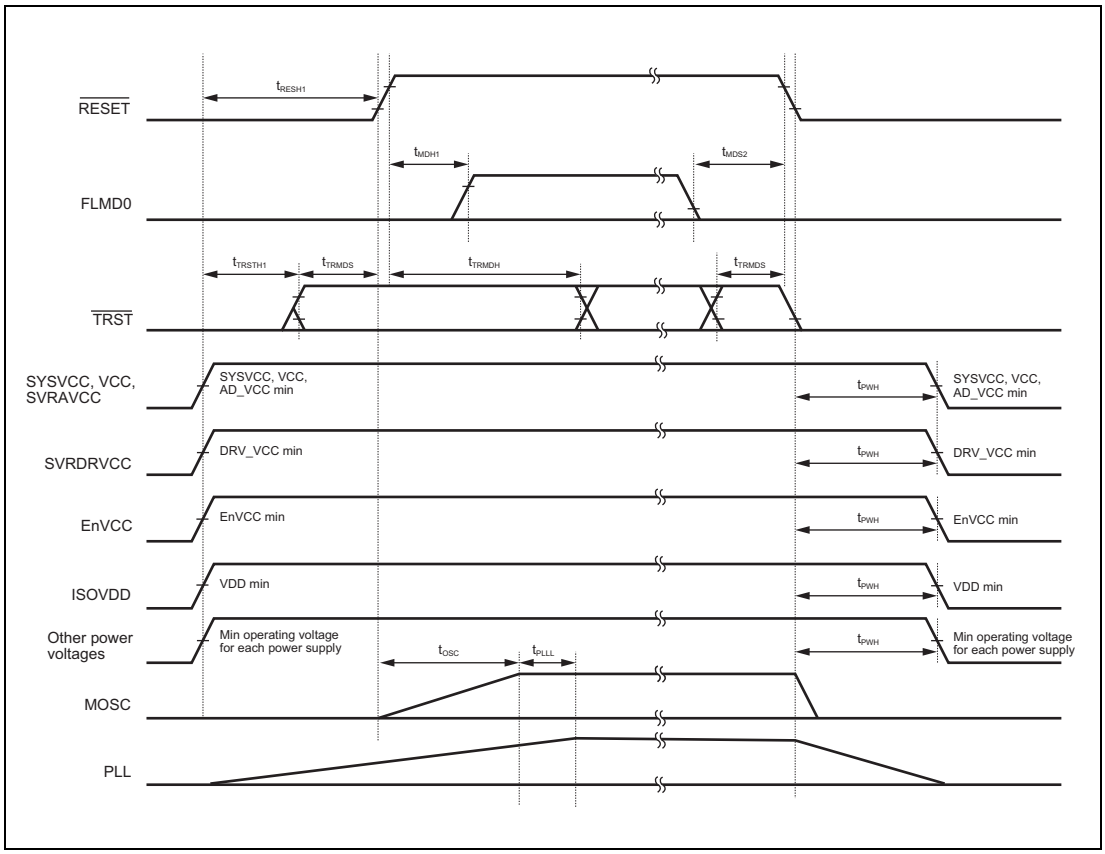


Figure 55.13 Power On/Off Timings (Normal operating mode and User boot mode 0)

NOTE

DVCC, DVDD, ERAMVCC, ERAMVDD and EMUVDD are not included in the other power voltages.

For power on/off timing about those power sources, see **Figure 55.18, EMUVCC, EMUVDD Power On / Off Timings** and **Section 55.3.30, Debug Resource (Aurora, ERAM) Specific Reset Timing**.

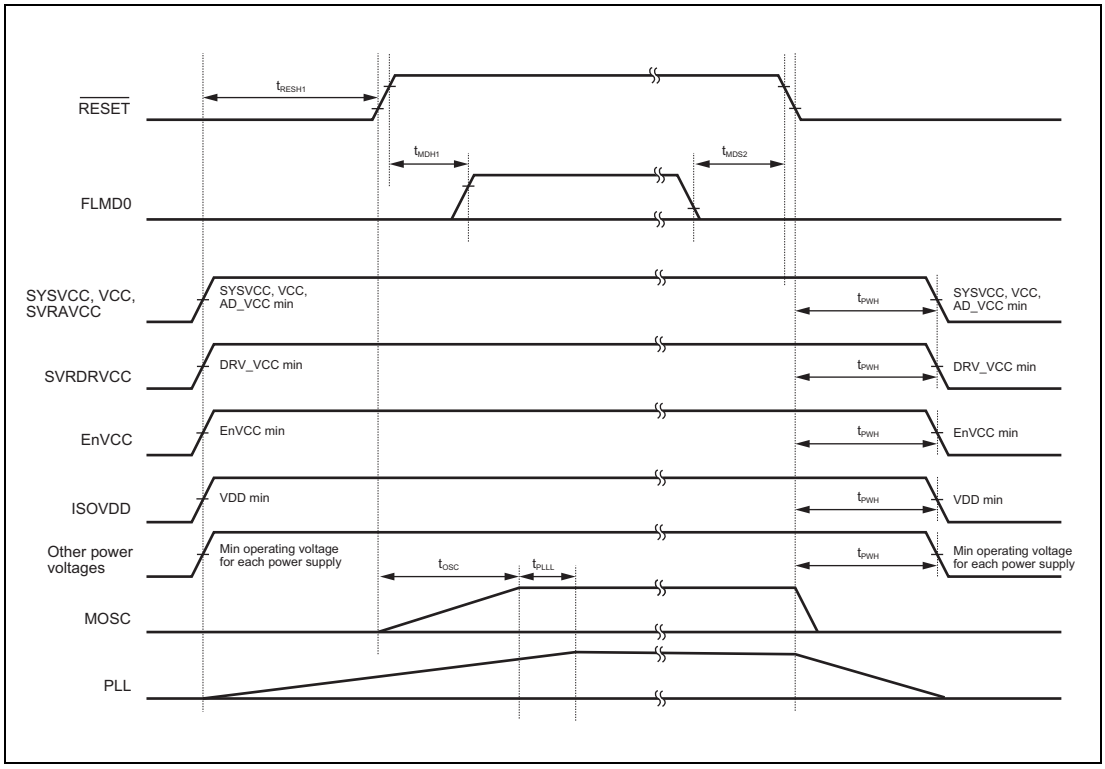


Figure 55.14 Power On/Off Timings (Serial programming mode 0)

NOTE

DVCC, DVDD, ERAMVCC, ERAMVDD and EMUVDD are not included in the other power voltages.

For power on/off timing about those power sources, see **Figure 55.18, EMUVCC, EMUVDD Power On / Off Timings** and **Section 55.3.30, Debug Resource (Aurora, ERAM) Specific Reset Timing**.

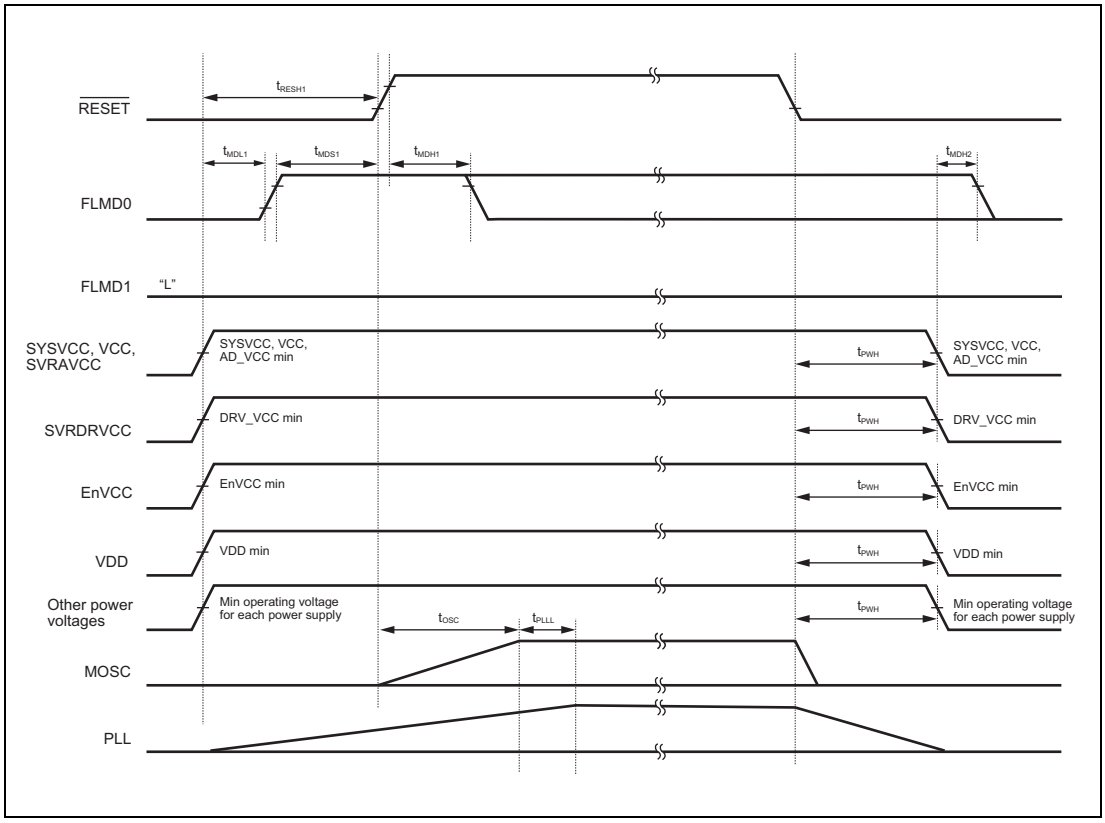


Figure 55.15 Power On/Off Timings (Serial programming mode 1)

NOTE

DVCC, DVDD, ERAMVCC, ERAMVDD and EMUVDD are not included in the other power voltages.

For power on/off timing about those power sources, see **Figure 55.18, EMUVCC, EMUVDD Power On / Off Timings** and **Section 55.3.30, Debug Resource (Aurora, ERAM) Specific Reset Timing**.

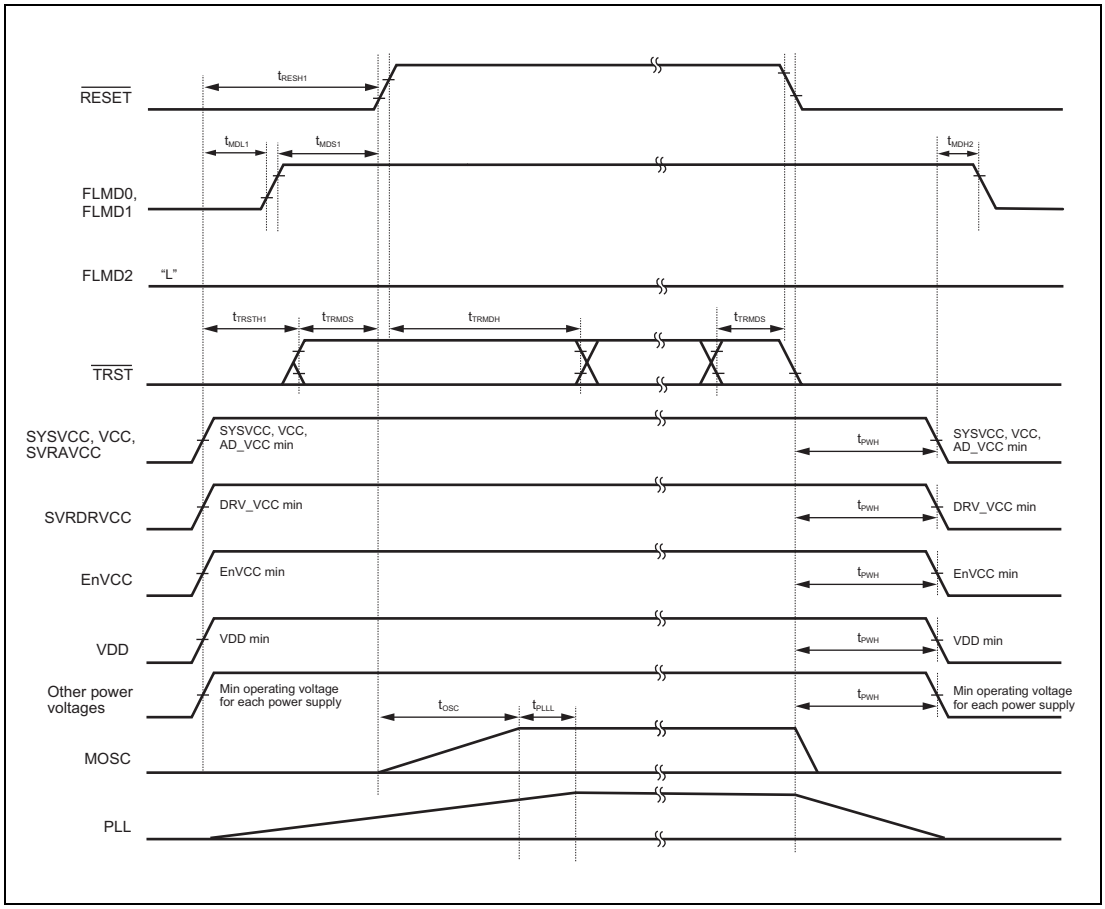


Figure 55.16 Power On/Off Timings (User boot mode 1)

NOTE

DVCC, DVDD, ERAMVCC, ERAMVDD and EMUVDD are not included in the other power voltages.

For power on/off timing about those power sources, see **Figure 55.18, EMUVCC, EMUVDD Power On / Off Timings** and **Section 55.3.30, Debug Resource (Aurora, ERAM) Specific Reset Timing**.

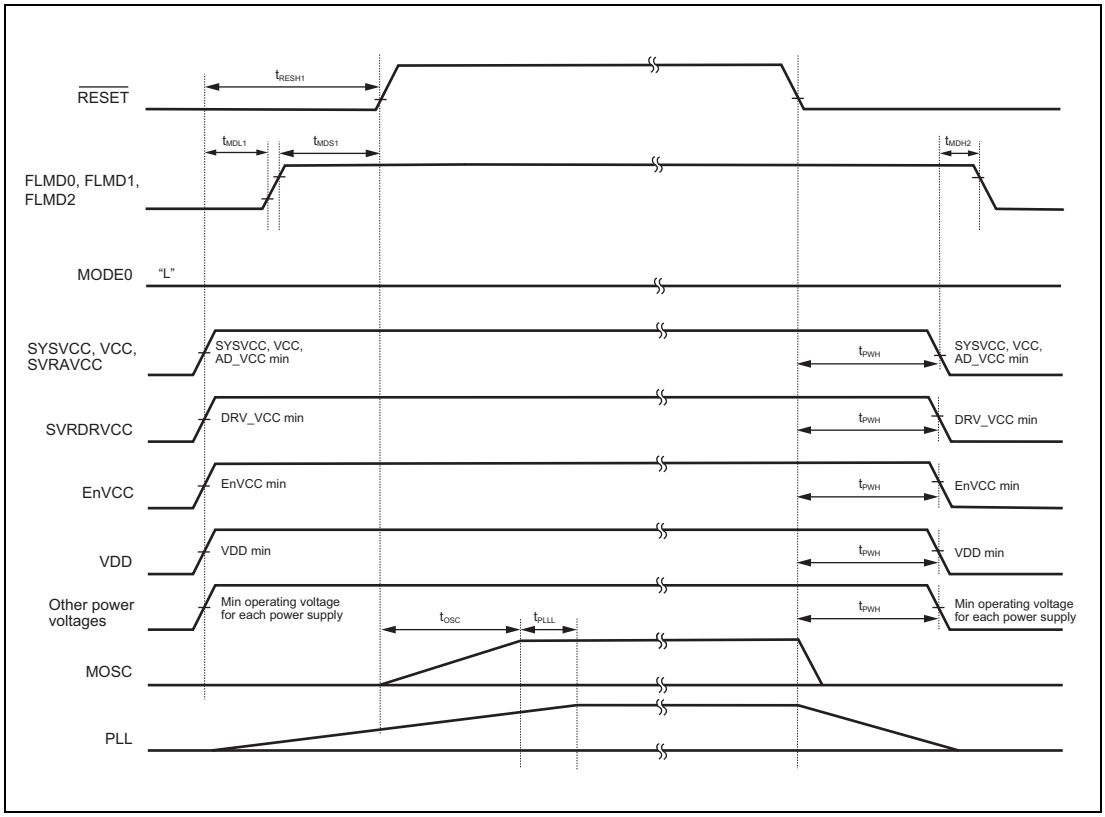


Figure 55.17 Power On/Off Timings (Boundary scan mode)

NOTE

DVCC, DVDD, ERAMVCC, ERAMVDD and EMUVDD are not included in the other power voltages.

For power on/off timing about those power sources, see **Figure 55.18, EMUVCC, EMUVDD Power On / Off Timings** and **Section 55.3.30, Debug Resource (Aurora, ERAM) Specific Reset Timing**.

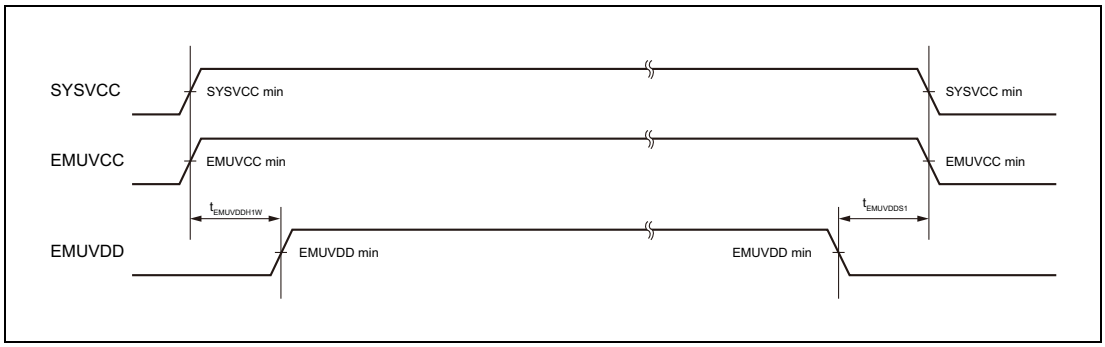


Figure 55.18 EMUVCC, EMUVDD Power On / Off Timings

55.3.2.1 Power Up Sequencing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**

Table 55.32 Power Up Sequencing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Runtime from release of $\overline{\text{RESET}}$ to start of instruction fetch					1.5 + t_{BIST}*1	ms

Note 1. t_{BIST} means BIST run time. Refer to **Table 44.490, BIST scenario selection at the next System Reset 2** on **Section 44.6.2.41, BSEQ0SEL — BIST Scenario Select Register** for details. Note that the given BIST run time is the typical value and the tolerance of the corresponding clock needs to be taken into account for the maximum value.

55.3.3 Standby Transition/Return Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**

Table 55.33 DeepSTOP Transition/Return Timing In case of VDD External Supply

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Wake-up trigger to PWRCTL delay time	$t_{DPWRCTL}$				100	µs
VDD hold time	t_{HDD}		0			µs
VDD power-on start time	t_{PDD}		0			µs

55.3.4 Clock Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.34 Clock Output Timing*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Clock output period time	t_{CLKOUT}		41.6			ns
Clock output high level width	t_{WCOH}		t_{CLKOUT} / 2 – 10			ns
Clock output low level width	t_{WCOL}		t_{CLKOUT} / 2 – 10			ns

Note 1. There is a function to output the internal clock via EXTCLKnO pin.

NOTE

For base clock, refer to related **Section 13, Clock Controller**.

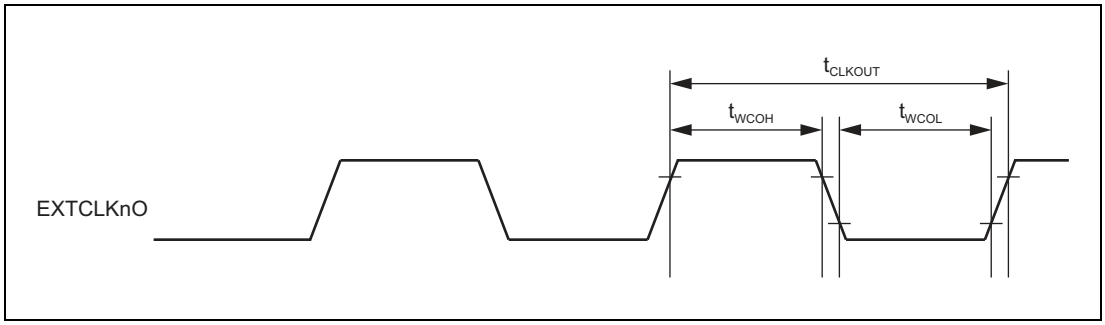


Figure 55.19 Clock Output Timing

55.3.5 Output Slew Rate

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.35 Output Slew Rate

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Output rise and fall time	t_{KRP11}/t_{KFP11}	Drive strength = 1 20% to 80%	CL = 100 pF			6.7	ns
	t_{KRP12}/t_{KFP12}	Drive strength = 2 20% to 80%	CL = 30 pF			4.0	ns
			CL = 50 pF			6.7	ns
			CL = 100 pF			13.4	ns
	t_{KRP13}/t_{KFP13}	Drive strength = 3 20% to 80%	CL = 30p F			6.7	ns
			CL = 50 pF			11.1	ns
			CL = 100 pF			21.0	ns
	t_{KRP14}/t_{KFP14}	Drive strength = 4 20% to 80%	CL = 30 pF			13.5	ns
			CL = 50 pF			22.5	ns
			CL = 100 pF			45.0	ns
	t_{KRP15}/t_{KFP15}	Drive strength = 5 20% to 80%	CL = 30 pF			27.0	ns
			CL = 50 pF			45.0	ns
			CL = 100 pF			90.0	ns

55.3.6 Control Signal Timing

55.3.6.1 $\overline{\text{RESET}}$ Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.36 $\overline{\text{RESET}}$ Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{RESET}}$ input low level width	t_{WRSL}*1		800			µs
$\overline{\text{RESET}}$ pulse rejection width*2	t_{WRRJ}		400		2400	ns

Note 1. t_{WRSL} is the minimum required time to complete the reset state generated by an external reset signal. When an external reset signal is input that is shorter than this time, the reset state will continue even after the external reset release. This microcontroller is completely reset after completion of the reset state. An external reset request is accepted with a reset input width of more than the maximum time of t_{WRRJ}. When the reset pulse width is less than the minimum value of the reset noise rejection width, the reset request is not accepted. When the reset signal is input during DeepSTOP mode, the reset state will continue during the wait time set by PWRGD_CNT even when the external reset signal is released.

Note 2. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered out or not). This characteristic is not tested in production.

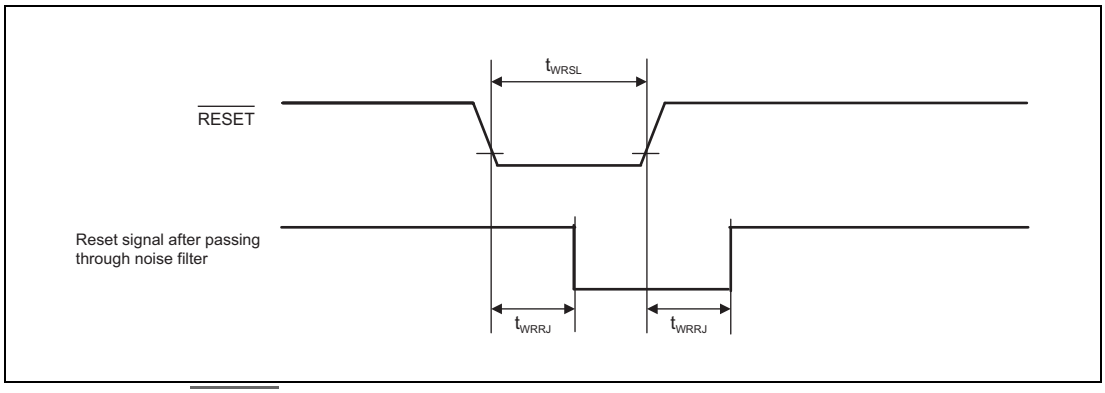


Figure 55.20 $\overline{\text{RESET}}$ Timing

55.3.6.2 Interrupt, Wake-up and Error Input Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.37 Interrupt Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
NMI input high level width	t_{WNIH}	*3	600			ns
		*4	20			µs
NMI input low level width	t_{WNIL}	*3	600			ns
		*4	20			µs
NMI pulse rejection width*2	t_{WNIRJ}		100		600	ns
INTPn input high level width	t_{WITH}	*3	600			ns
		*4	20			µs
INTPn input low level width	t_{WITL}	*3	600			ns
		*4	20			µs
INTPn pulse rejection width*2	t_{WIRJ}		100		600	ns
RLIN3nRX wake-up input high level width	t_{WRLINH}		(S + 1) x 1/fs*1			ns
RLIN3nRX wake-up input low level width	t_{WRLINL}		(S + 1) x 1/fs*1			ns
RLIN3nRX wake-up pulse rejection width*2	$t_{WRLINRJ}$		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
CANnRX wake-up input high level width	t_{WCANH}		(S + 1) x 1/fs*1			ns
CANnRX wake-up input low level width	t_{WCANL}		(S + 1) x 1/fs*1			ns
CANnRX wake-up pulse rejection width*2	t_{WCANRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
FLXnRXDA wake-up input high level width	t_{WFLXH}		(S + 1) x 1/fs*1			ns
FLXnRXDA wake-up input low level width	t_{WFLXL}		(S + 1) x 1/fs*1			ns
FLXnRXDA wake-up pulse rejection width*2	t_{WFLXRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
$\overline{\text{ERRORINn}}$ wake-up input high level width	t_{WERRH}		(S + 1) x 1/fs*1			ns
$\overline{\text{ERRORINn}}$ wake-up input low level width	t_{WERRL}		(S + 1) x 1/fs*1			ns
$\overline{\text{ERRORINn}}$ wake-up pulse rejection width*2	t_{WERRRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns

Note 1. S: Number of sampling times
fs: The value given by following formula

$$fs = \frac{f_{DNFCK}}{PRS}$$

f_{DNFCK}: frequency of CLK_LSB
PRS: 1, 2, 4, 8, … , 128

Note 2. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered out or not). This characteristic is not tested in production.

Note 3. Edge Detection or Level Detection (CLKA_LPS is operated by CLK_HSIOSC/20)

Note 4. Level Detection (CLKA_LPS is operated by CLK_LSIOSC)

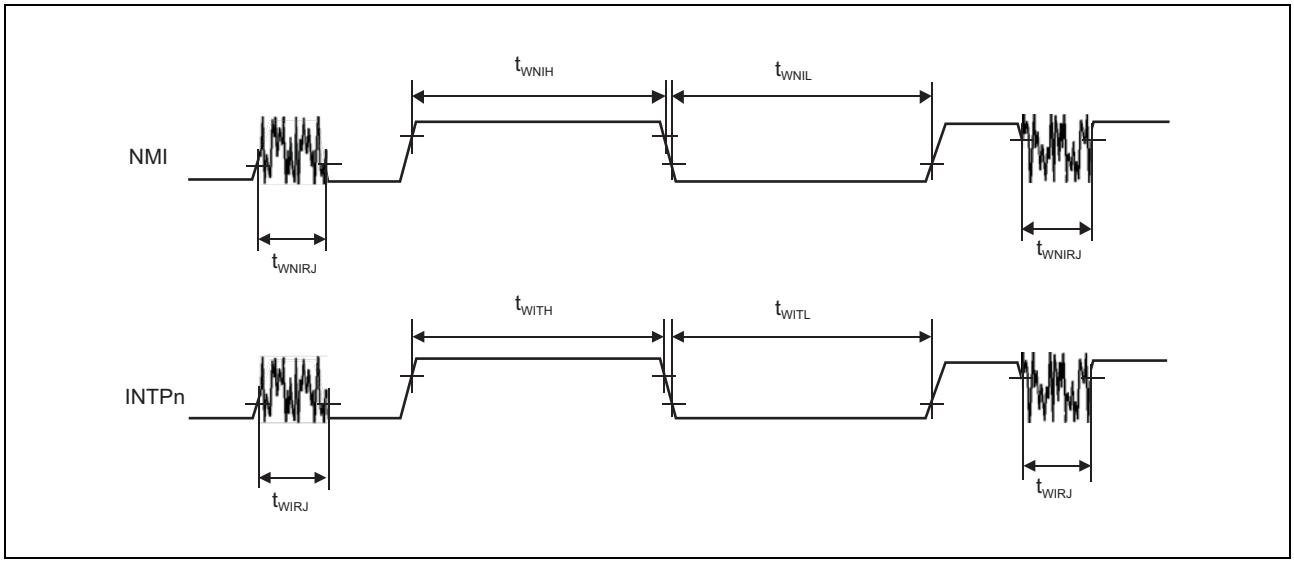


Figure 55.21 Interrupt, wake-up and error input timing

55.3.6.3 Mode Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.38 Mode Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1, FLMD2, MODE0 input high level width	t_{WMDH}		600			ns
FLMD0, FLMD1, FLMD2, MODE0 input low level width	t_{WMDL}		600			ns
FLMD0, FLMD1, FLMD2, MODE0 pulse rejection width*1	t_{WMDRJ}		100		600	ns

Note 1. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered out or not). This characteristic is not tested in production.

NOTE

Switching of FLMD0 after rising edge of $\overline{\text{RESET}}$ is prohibited except for serial programming mode.

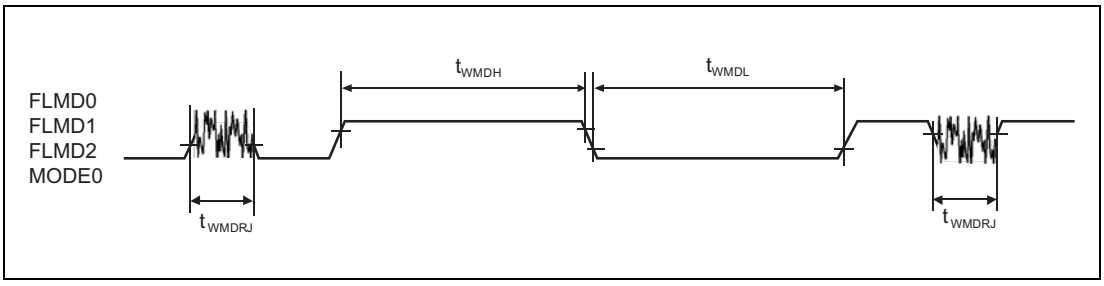


Figure 55.22 Mode Timing

55.3.6.4 ADTRG Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.39 ADCJnTRGm Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ADCJnTRGm input high level width	t_{WADH}		(S + 1) x 1/fs*1			ns
ADCJnTRGm input low level width	t_{WADL}		(S + 1) x 1/fs*1			ns
ADCJnTRGm pulse rejection width*2	t_{WADRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns

Note 1. S: Number of sampling times
fs: The value given by the following formula

$$fs = \frac{f_{DNFCK}}{PRS}$$

f_{DNFCK}: frequency of CLK_ADC, CLKA_ADC
PRS: 1, 2, 4, 8, … , 128

Note 2. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered out or not). This characteristic is not tested in production.

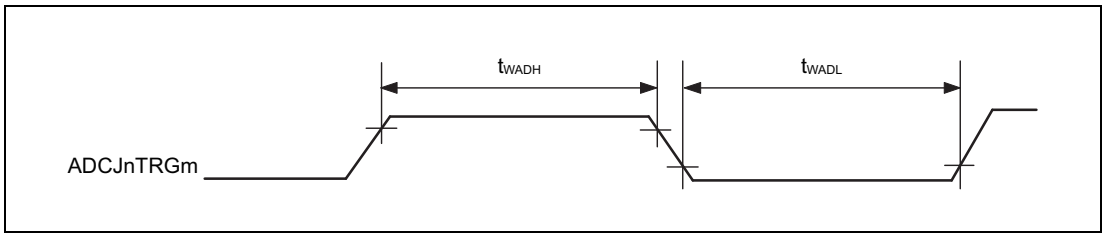


Figure 55.23 ADCJnTRGm Timing

55.3.6.5 Communication Signal Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.40 Control Signal

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SENTnRX input high level width	$t_{WSENTIH}$	Analog filter	600			ns
		Digital filter	(S + 1) x 1/fs*1			ns
SENTnRX input low level width	$t_{WSENTIL}$	Analog filter	600			ns
		Digital filter	(S + 1) x 1/fs*1			ns
SENTnRX pulse rejection width	$t_{WSENTIRJ}$	Analog filter	100		600	ns
		Digital filter	(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
PSI5nRX input high level width*2	$t_{WPSI5IH}$	Analog filter	600			ns
		Digital filter	(S + 1) x 1/fs*1			ns
PSI5nRX input low level width	$t_{WPSI5IL}$	Analog filter	600			ns
		Digital filter	(S + 1) x 1/fs*1			ns
PSI5nRX pulse rejection width*2	$t_{WPSI5IRJ}$	Analog filter	100		600	ns
		Digital filter	(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns

Note 1. S: Number of sampling times
fs: The value given by the following formula

$$fs = \frac{f_{DNFCK}}{PRS}$$

f_{DNFCK}: frequency of CLK_HSB
PRS: 1, 2, 4, 8, … , 128

Note 2. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered out or not). This characteristic is not tested in production.

55.3.7 Low Power Sampler (DPIN input) Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 4 (SELDP2-0)

Table 55.41 Low Power Sampler Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
DPINn input delay time	t_{DSDDI}				150	ns

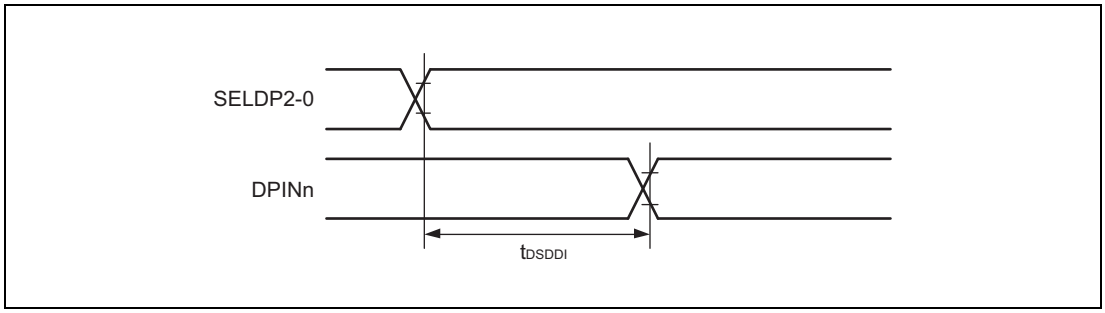


Figure 55.24 Low Power Sampler Timing

55.3.8 SFMA Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 2
- Buffer type = Sch1

Table 55.42 SFMA Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SFMA0CLK clock cycle	$t_{SFMAcyc}$		25			ns
SFMA0CLK high pulse width	t_{SFMAWH}		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
SFMA0CLK low pulse width	t_{SFMAWL}		0.4 × $t_{SFMA0cyc}$		0.6 × $t_{SFMA0cyc}$	ns
Data input setup time	$t_{SFMADIS}$		9.0			ns
Data input hold time	t_{SFMDIH}		0.0			ns
SFMA0SSL setup time	t_{SFMASS}		1 × $t_{SFMA0cyc}$ – 12.5		8 × $t_{SFMA0cyc}$	ns
SFMA0SSL hold time	t_{SFMASH}		1.5 × $t_{SFMA0cyc}$		8.5 × $t_{SFMA0cyc}$ + 12.5	ns
Continuous transfer delay time	$t_{SFMACTD}$		1 × $t_{SFMA0cyc}$		8 × $t_{SFMA0cyc}$	ns
Data output delay time	$t_{SFMADOD}$				6.6	ns
Data output hold time	$t_{SFMADOH}$		–4.6			ns
Data output buffer on time	$t_{SFMADBON}$				6.6	ns
Data output buffer off time	$t_{SFMADBOFF}$		–7.0		3.0	ns

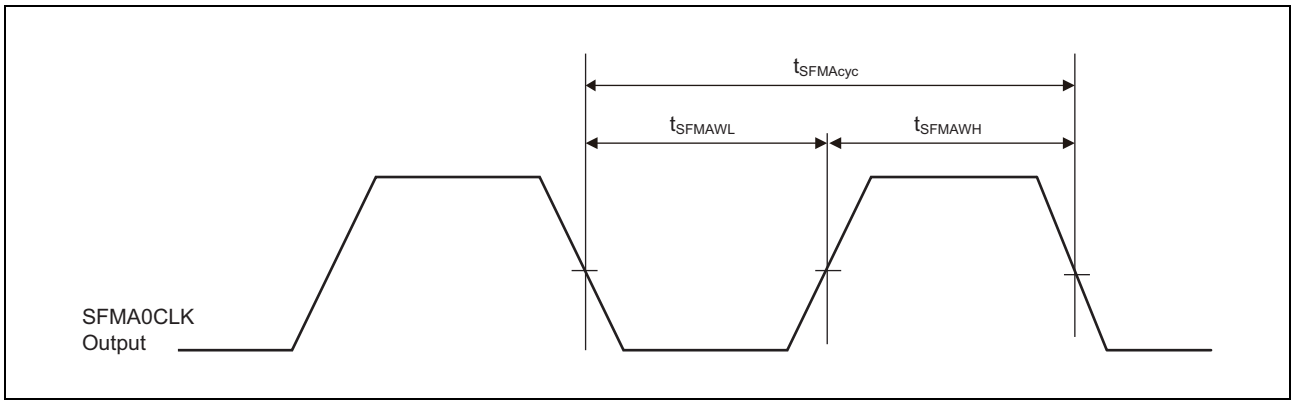


Figure 55.25 SFMA Clock Output Timing

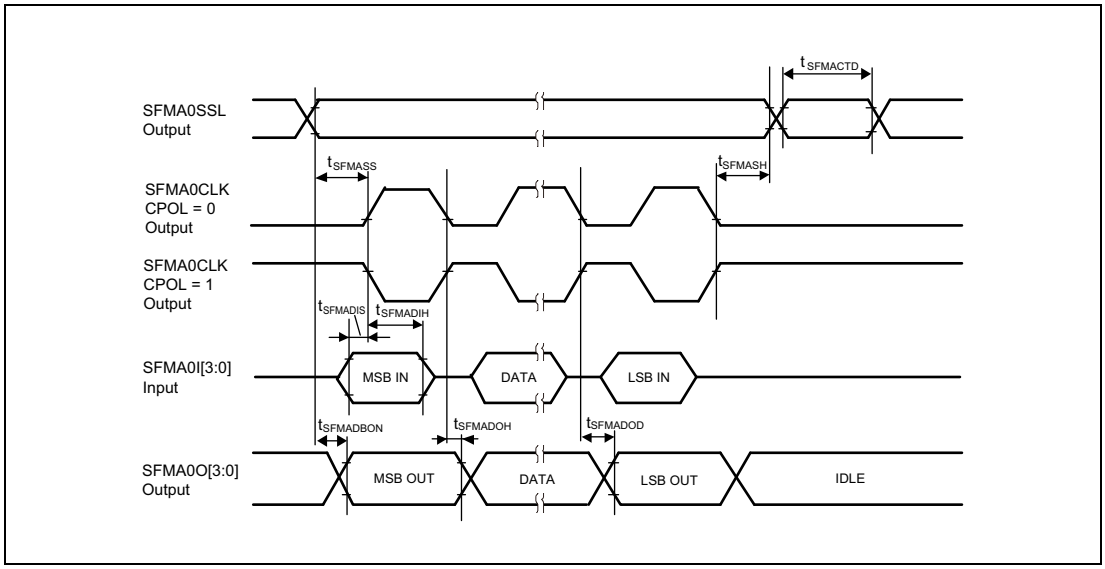


Figure 55.26 SFMA Transmission and Reception Timing (CPHAT = 0, CPHAR = 0)

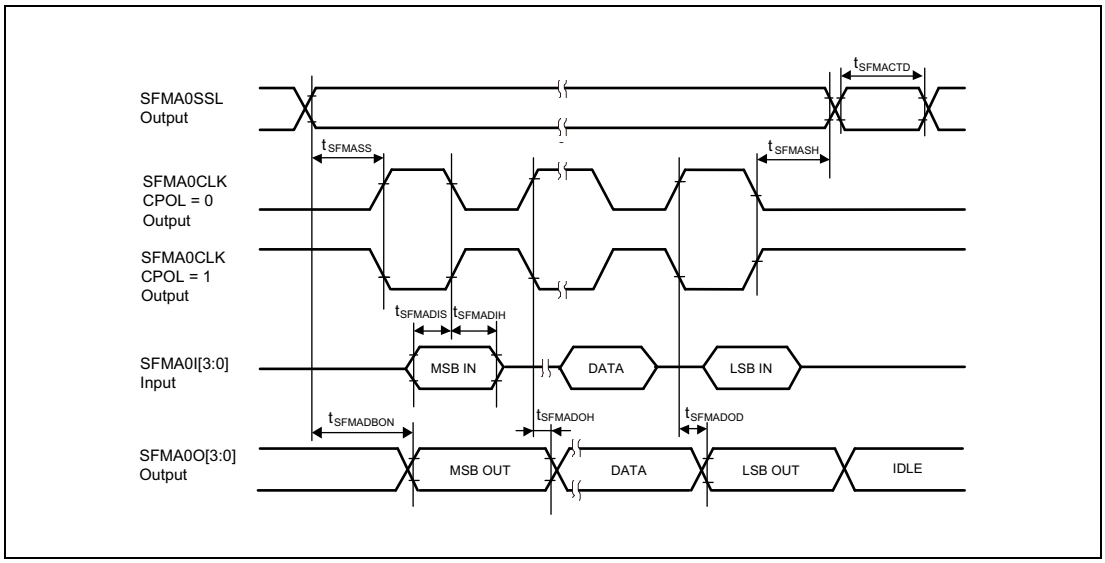


Figure 55.27 SFMA Transmission and Reception Timing (CPHAT = 1, CPHAR = 1)

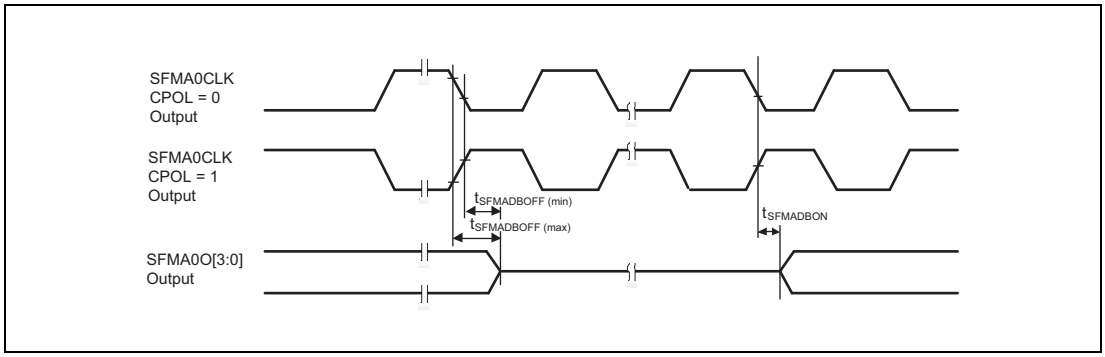


Figure 55.28 SFMA Timing Switching the Buffers on and off (CPHAT = 0, CPHAR = 0)

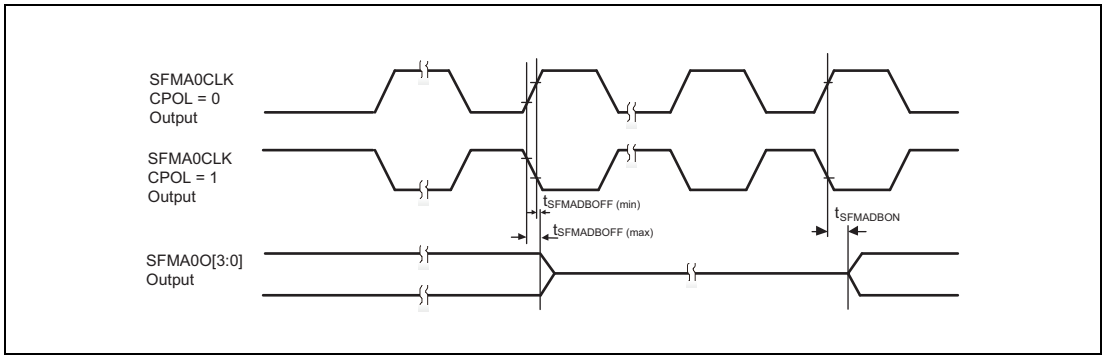


Figure 55.29 SFMA Timing for Switching the Buffers on and off (CPHAT = 1, CPHAR = 1)

55.3.9 MMCA Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 2
- Buffer type = Sch1

Table 55.43 MMCA Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MMCA0CLK clock cycle	$t_{MMCA0CYC}$		25			ns
MMCA0CLK high time	$t_{MMCA0WH}$		6.5			ns
MMCA0CLK low time	$t_{MMCA0WL}$		6.5			ns
MMCA0CMD output data delay time	$t_{MMCA0CMDD}$		– 6.5		6.5	ns
Data output delay time	$t_{MMCA0DADD}$		– 6.5		6.5	ns
MMCA0CMD input data setup time	$t_{MMCA0CMS}$		7.5			ns
MMCA0CMD input data hold time	$t_{MMCA0CMH}$		2.5			ns
Data input setup time	$t_{MMCA0DAS}$		7.5			ns
Data input hold time	$t_{MMCA0DAH}$		2.5			ns

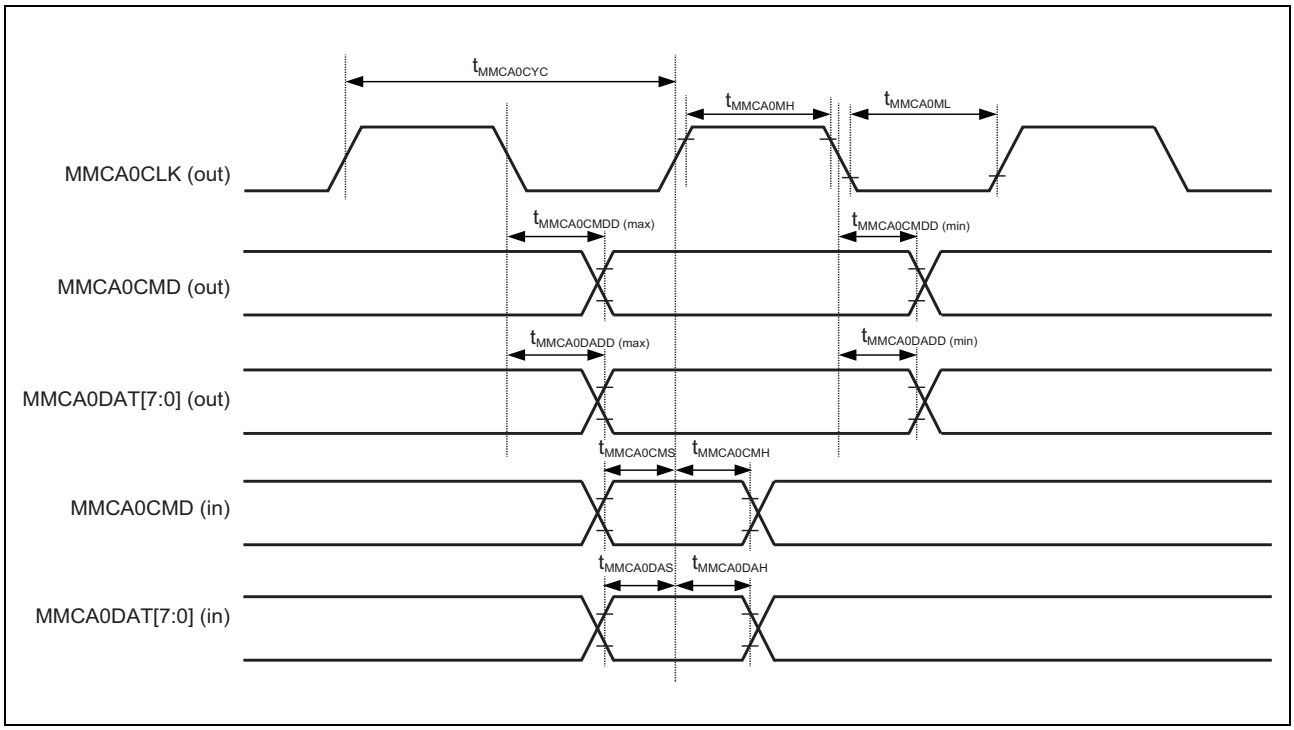


Figure 55.30 MMCA Timing

55.3.10 MSPI Timing

55.3.10.1 MSPI Communication Speed Overview

Table 55.44 MSPI Communication Speed Overview (1/3)

MSPIn pin name		Port	Max. Frequency
MSPI0	SC	P4_6	10 MHz
	SO	P4_5	10 MHz
	SI	P4_7	10 MHz
	SC	P6_8	10 MHz
	SO	P6_9	10 MHz
	SI	P6_7	10 MHz
	SC	P2_12	20 MHz
	SO	P2_13	20 MHz
	SI	P2_11	20 MHz
	SCKN / SCKP	P2_14, P2_15	40 MHz
	SON / SOP	P2_12, P2_13	40 MHz
	SIN / SIP	P2_10, P2_11	40 MHz
MSPI1	SC	P2_2	10 MHz
	SO	P5_2	10 MHz
	SI	P2_1	10 MHz
	SC	P6_8	10 MHz
	SO	P6_7	10 MHz
	SI	P6_9	10 MHz
	SC	P10_3	20 MHz
	SO	P10_2	20 MHz
	SI	P10_4	20 MHz
	SCKN / SCKP	P4_10, P4_9	40 MHz
	SON / SOP	P4_8, P4_7	40 MHz
	SIN / SIP	P4_6, P4_5	40 MHz
MSPI2	SC	P4_7	10 MHz
	SO	P4_6	10 MHz
	SI	P4_10	10 MHz
	SC	P10_11	10 MHz
	SO	P10_10	10 MHz
	SI	P10_12	10 MHz
	SC	P2_0	20 MHz (The selection of "Drive strength = 1" is available at equal or less than 10 MHz.)
	SO	P2_5	20 MHz (The selection of "Drive strength = 1" is available at equal or less than 10 MHz.)
	SI	P2_3	20 MHz

Table 55.44 MSPI Communication Speed Overview (2/3)

MSPIn pin name		Port	Max. Frequency
MSPI3	SC	P22_4	10 MHz (The selection of "Drive strength = 1" is available)
	SO	P22_0	10 MHz (The selection of "Drive strength = 1" is available)
	SI	P22_1	10 MHz
	SC	P24_10	10 MHz
	SO	P24_13	10 MHz
	SI	P24_12	10 MHz
	SC	P5_2	20 MHz
	SO	P5_4	20 MHz
	SI	P5_3	20 MHz
MSPI4	SC	P23_0	10 MHz
	SO	P23_2	10 MHz
	SI	P23_1	10 MHz
	SC	P21_0	20 MHz
	SO	P21_2	20 MHz
	SI	P21_1	20 MHz
MSPI5	SC	P12_0	10 MHz
	SO	P12_2	10 MHz
	SI	P12_1	10 MHz
	SC	P24_4	20 MHz
	SO	P24_6	20 MHz
	SI	P24_5	20 MHz
MSPI6	SC	P9_0	10 MHz
	SO	P9_2	10 MHz
	SI	P9_1	10 MHz
	SC	P11_0	10 MHz
	SO	P11_2	10 MHz
	SI	P11_1	10 MHz
MSPI7	SC	P11_8	10 MHz
	SO	P11_10	10 MHz
	SI	P11_9	10 MHz
	SC	P19_0	10 MHz
	SO	P19_2	10 MHz
	SI	P19_1	10 MHz
MSPI8	SC	P0_0	10 MHz
	SO	P0_2	10 MHz
	SI	P0_1	10 MHz
	SC	P18_1	10 MHz
	SO	P18_3	10 MHz
	SI	P18_2	10 MHz

Table 55.44 MSPI Communication Speed Overview (3/3)

MSPIn pin name		Port	Max. Frequency
MSPI9	SC	P1_0	10 MHz
	SO	P1_2	10 MHz
	SI	P1_1	10 MHz
	SC	P18_8	10 MHz
	SO	P18_10	10 MHz
	SI	P18_9	10 MHz

55.3.10.2 MSPI Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = Sch1

Table 55.45 MSPI Timing (Master mode : Communication Speed 10MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle*1*5	T_{SCKCYC}		100			ns
MSPInSCK Low/High width*1	T_{SCKWID}	CL = 30 pF Drive strength = 3	0.5 × T_{SCKCYC} – 9			ns
		CL = 50 pF Drive strength = 3	0.5 × T_{SCKCYC} – 12			ns
		CL = 100 pF Drive strength = 3	0.5 × T_{SCKCYC} – 23			ns
		CL = 30 pF Drive strength = 4	0.5 × T_{SCKCYC} – 15.5			ns
		CL = 50 pF Drive strength = 2	0.5 × T_{SCKCYC} – 10			ns
		CL = 100 pF Drive strength = 1*4	0.5 × T_{SCKCYC} – 10			ns
Chip select signal setup time*2*6	T_{MCSSU}	$T_{MSPInCLK}$ = 12.5 ns	MSPInSEUPm [11:0] x $T_{MSPInCLK}$ - 15			ns
Chip select signal hold time*3	T_{MCSHO}	$T_{MSPInCLK}$ = 12.5 ns	MSPInHOLDm [11:0] x $T_{MSPInCLK}$ - 5			ns
Receive data setup time (MSPInSAMP =0)	T_{MSISU1}		20			ns
Receive data setup time (MSPInSAMP =1)	T_{MSISU2}		20			ns
Receive data hold time (MSPInSAMP =0)	T_{MSIHO}		0			ns
Receive data hold time (MSPInSAMP =1)	T_{MSIHO}		0			ns
Transmit data delay time	T_{MSODL}				7	ns
Transmit data hold time	T_{MSOHL}		(T_{SCKCYC} / 2) – 5			ns

Note 1. This item is programmable, the value can be set by MSPInPRCSm[1:0] and MSPInCDIVm[4:0].
Note 2. This item is programmable, the value can be set by MSPInSEUPm[11:0] and it must be set to 002_H or above.
Note 3. This item is programmable, the value can be set by MSPInHOLDm.
Note 4. The selection of "Drive strength = 1" is available on P2_0 and P22_4 only.
Note 5. Use by “PCLK/2 ≥ MSPInSCK”.
Note 6. When using the MSPI at master mode with the setting MSPInCFGm1.MSPInCPHAm = 0, set the period from CS active to the first edge of SCK to 1/2 or more of the communication rate by MSPInSEUPm.

Table 55.46 MSPI Timing (Master mode : Communication Speed 20MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle*1*4	T_{SCKCYC}		50			ns
MSPInSCK Low/High width*1	T_{SCKWID}	CL = 15pF@50Ω Drive strength = 3	0.5 × T_{SCKCYC} – 6			ns
Chip select signal setup time*2*5	T_{MCSSU}	$T_{MSPInCLK}$ = 12.5 ns	MSPInSEUPm [11:0] x $T_{MSPInCLK}$ – 15			ns
Chip select signal hold time*3	T_{MCSHO}	$T_{MSPInCLK}$ = 12.5 ns	MSPInHOLDm [11:0] x $T_{MSPInCLK}$ – 5			ns
Receive data setup time (MSPInSAMP = 1)	T_{MSISU2}		20			ns
Receive data hold time (MSPInSAMP = 1)	T_{MSIHO}		0			ns
Transmit data delay time	T_{MSODL}				7	ns
Transmit data hold time	T_{MSOHL}		(T_{SCKCYC}/2) – 5			ns

Note 1. This item is programmable, the value can be set by MSPInPRCSm[1:0] and MSPInCDIVm[4:0].
Note 2. This item is programmable, the value can be set by MSPInSEUPm[11:0] and it must be set to 002_H or above.
Note 3. This item is programmable, the value can be set by MSPInHOLDm.
Note 4. Use by “PCLK/2 ≥ MSPInSCK”.
Note 5. When using the MSPI at master mode with the setting MSPInCFGm1.MSPInCPHAm = 0, set the period from CS active to the first edge of SCK to 1/2 or more of the communication rate by MSPInSEUPm.

Table 55.47 MSPI Timing (Master mode : Communication Speed 40MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle*1*4	T_{SCKCYC}		25			ns
MSPInSCK Low/High width*1	T_{SCKWID}	CL = 15pF@LVDS	0.5 × T_{SCKCYC} – 2.5			ns
Chip select signal setup time*2*5	T_{MCSSU}	$T_{MSPInCLK}$ = 12.5 ns	MSPInSEUPm [11:0] x $T_{MSPInCLK}$ – 15			ns
Chip select signal hold time*3	T_{MCSHO}	$T_{MSPInCLK}$ = 12.5 ns	MSPInHOLDm [11:0] x $T_{MSPInCLK}$ – 5			ns
Receive data setup time (MSPInSAMP = 1)	T_{MSISU2}		7			ns
Receive data hold time (MSPInSAMP = 1)	T_{MSIHO}		0			ns
Transmit data delay time	T_{MSODL}				5	ns
Transmit data hold time	T_{MSOHL}		(T_{SCKCYC}/2) – 5			ns

Note 1. This item is programmable, the value can be set by MSPInPRCSm[1:0] and MSPInCDIVm[4:0].
Note 2. This item is programmable, the value can be set by MSPInSEUPm[11:0] and it must be set to 002_H or above.
Note 3. This item is programmable, the value can be set by MSPInHOLDm.
Note 4. Use by “PCLK/2 ≥ MSPInSCK”.
Note 5. When using the MSPI at master mode with the setting MSPInCFGm1.MSPInCPHAm = 0, set the period from CS active to the first edge of SCK to 1/2 or more of the communication rate by MSPInSEUPm.

Table 55.48 MSPI Timing (Slave mode : Communication Speed 10MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle	T_{SCKCYC}		100			ns
MSPInSCK Low/High width	T_{SCKWID}		$0.5 \times T_{SCKCYC} - 23$			ns
Chip select signal setup time (MSPInCSIE = 1)	T_{SCSSU}		15			ns
Chip select signal hold time (MSPInCSIE = 1)	T_{SCSHO}		10			ns
Receive data setup time	T_{SSISU}		6			ns
Receive data hold time	T_{SSIHO}		5			ns
Transmit data delay time 1	T_{SSODL1}				54	ns
Transmit data delay time 2 (MSPInCSIE = 1)	T_{SSODL2}				40	ns
Transmit data hold time	T_{SSOHL}		$(T_{SCKCYC}/2) - 5$			ns
Transmit data release time (MSPInCSIE = 1)	T_{SSOREL}				40	ns

Table 55.49 MSPI Timing (Slave mode : Communication Speed 20MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle	T_{SCKCYC}		50			ns
MSPInSCK Low/High width	T_{SCKWID}		$0.5 \times T_{SCKCYC} - 6$			ns
Chip select signal setup time (MSPInCSIE = 1)	T_{SCSSU}		15			ns
Chip select signal hold time (MSPInCSIE = 1)	T_{SCSHO}		10			ns
Receive data setup time	T_{SSISU}		6			ns
Receive data hold time	T_{SSIHO}		5			ns
Transmit data delay time 1	T_{SSODL1}				24	ns
Transmit data delay time 2 (MSPInCSIE = 1)	T_{SSODL2}				40	ns
Transmit data hold time	T_{SSOHL}		$(T_{SCKCYC}/2) - 5$			ns
Transmit data release time (MSPInCSIE = 1)	T_{SSOREL}				40	ns

Table 55.50 MSPI Timing (Slave mode: Communication Speed 40MHz)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
MSPI operation clock cycle	$T_{MSPInCLK}$		12.5			ns
MSPInSCK cycle	T_{SCKCYC}		25			ns
MSPInSCK Low/High width	T_{SCKWID}		0.5 × T_{SCKCYC} – 2.5			ns
Chip select signal setup time (MSPInCSIE = 1)	T_{SCSSU}		15			ns
Chip select signal hold time (MSPInCSIE = 1)	T_{SCSHO}		10			ns
Receive data setup time	T_{SSISU}		6			ns
Receive data hold time	T_{SSIHO}		5			ns
Transmit data delay time 1	T_{SSODL1}				12	ns
Transmit data hold time	T_{SSOHL}		(T_{SCKCYC}/2) – 5			ns

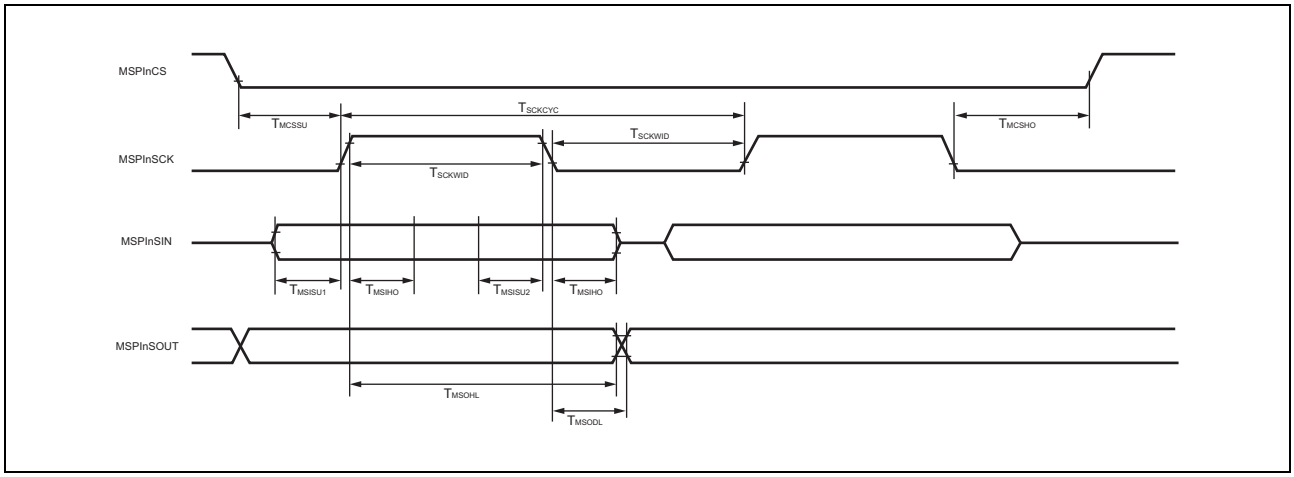


Figure 55.31 MSPIn Timing (Master Mode, MSPInCPHAm = 0)

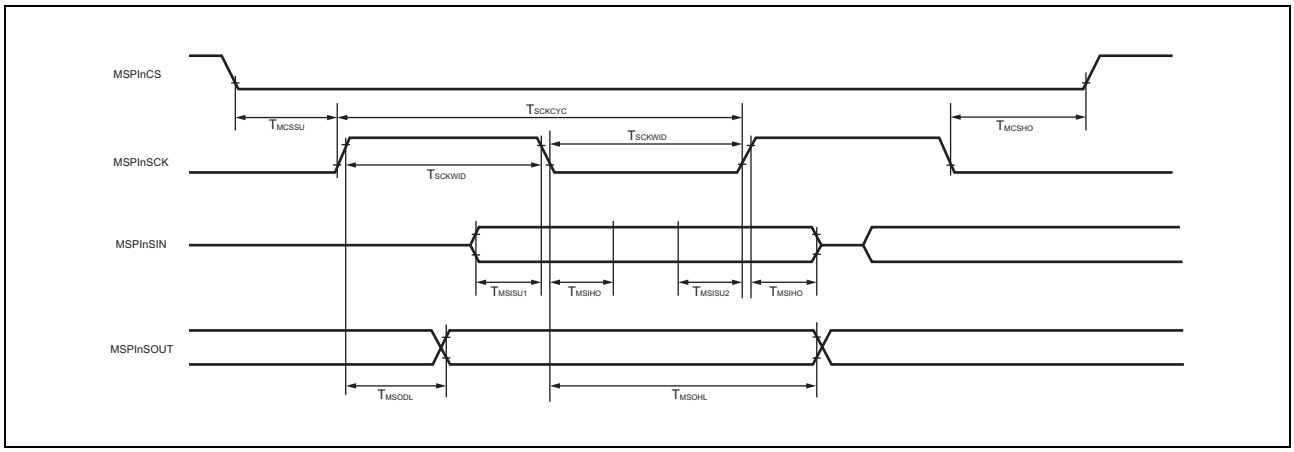


Figure 55.32 MSPIn Timing (Master Mode, MSPInCPHAm = 1)

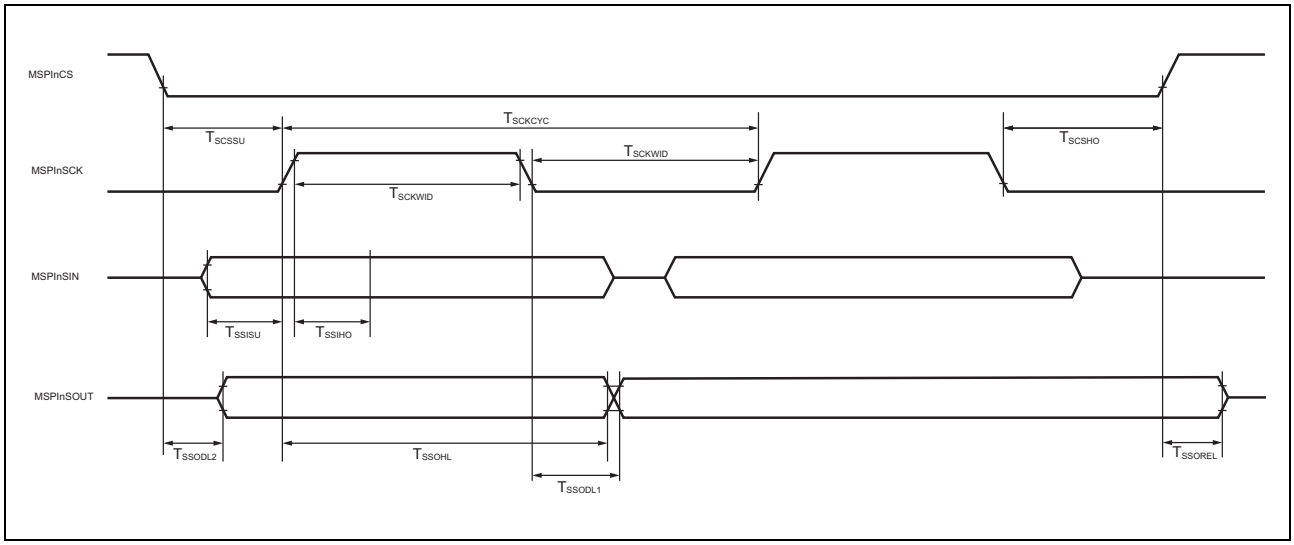


Figure 55.33 MSPIn Timing (Slave Mode, MSPInCPHA0 = 0)

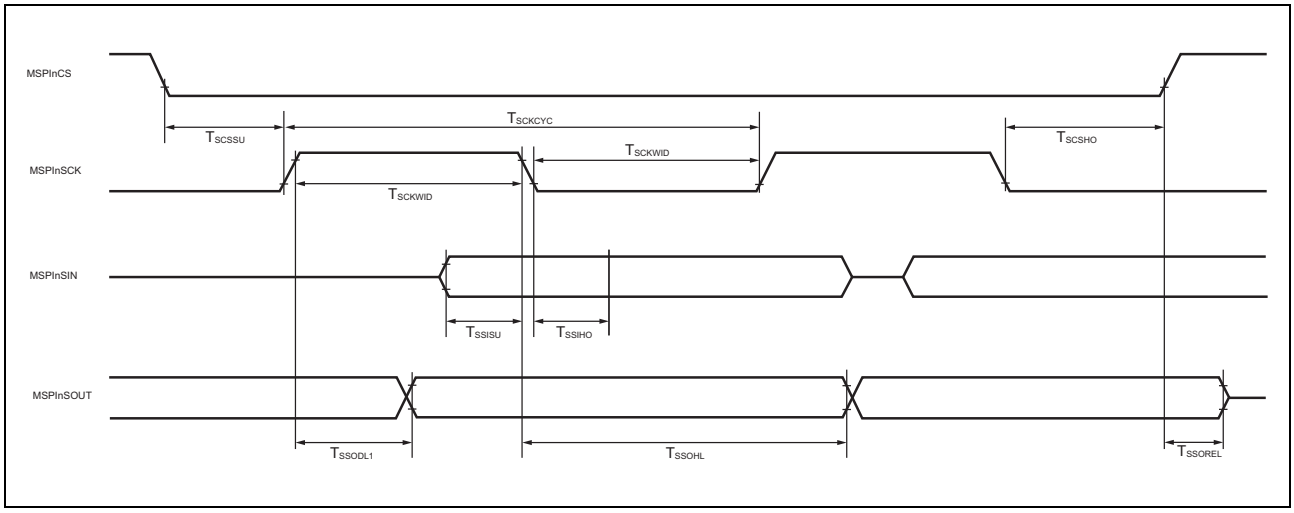


Figure 55.34 MSPIn Timing (Slave Mode, MSPInCPHA0 = 1)

55.3.11 SCI3 Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = Sch1

Table 55.51 SCI3 Timing (Master Mode, Asynchronous)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate	r_{SCI3AS}				10	Mbps

Table 55.52 SCI3 Timing (Master Mode, Synchronous)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output cycle time	$t_{SCI3Scyc}$		200			ns
Output clock pulse width	t_{SCI3SW}		0.4 × $t_{SCI3Scyc}$		0.6 × $t_{SCI3Scyc}$	ns
Transmit data delay time	$t_{SCI3TXD}$		–40		40	ns
Receive data setup time	$t_{SCI3RXS}$		4 × t_{PCLK}			ns
Receive data hold time	$t_{SCI3RXH}$		4 × t_{PCLK}			ns

Note: t_{PCLK} is the period of PCLK (CLK_HSB).

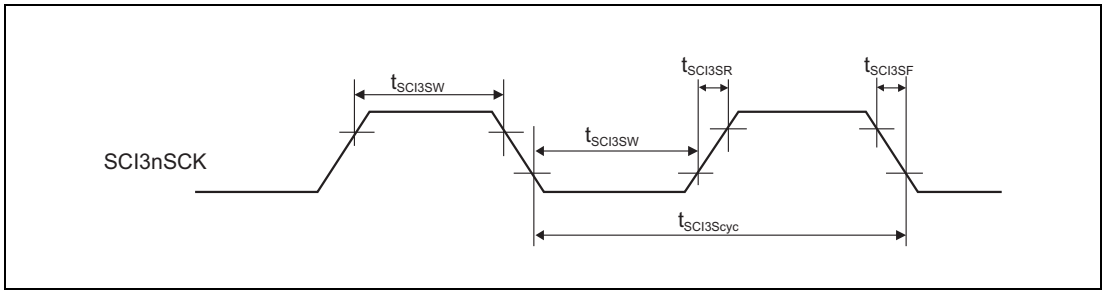


Figure 55.35 SCI Clock Input/Output Timing

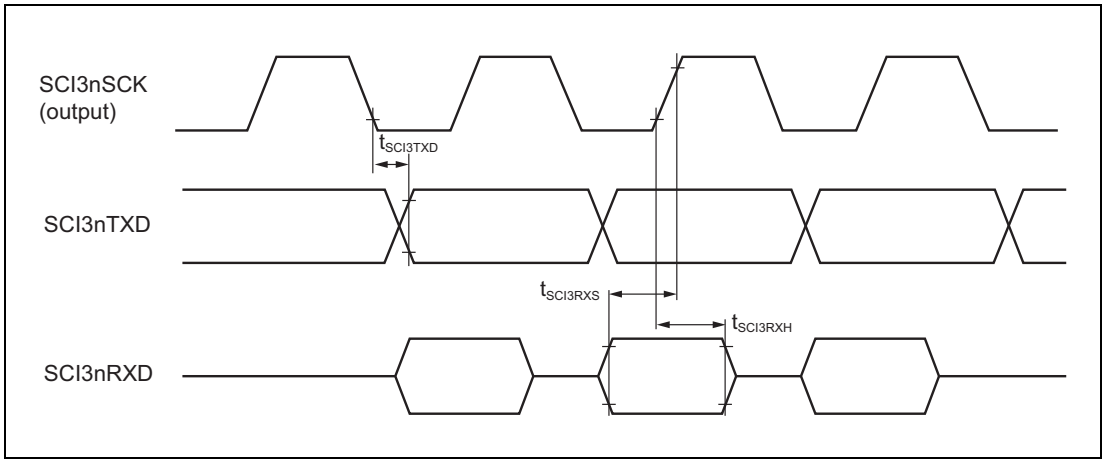


Figure 55.36 SCI3 Input/Output Timing, Clock Synchronous Mode (in Master Mode)

Table 55.53 SCI3 Timing (Slave Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Input cycle time	$t_{SCI3Scyc}$		200			ns
Input clock pulse width	t_{SCI3SW}		0.4 × $t_{SCI3Scyc}$		0.6 × $t_{SCI3Scyc}$	ns
Transmit data delay time*1	$t_{SCI3TXD}$		2 × t_{PCLK}		50 + 4 × t_{PCLK}	ns
Input clock rising time	t_{SCI3SR}				20	ns
Input clock falling time	t_{SCI3SF}				20	ns
Receive data setup time	$t_{SCI3RXS}$		4 × t_{PCLK}			ns
Receive data hold time	$t_{SCI3RXH}$		4 × t_{PCLK}			ns

Note 1. This does not apply to transmission of Data 0 (the first bit) which is transferred not in continuous transfer mode. Transmission of Data 0 (the first bit) which is transferred not in continuous transfer mode starts when TDRE becomes 0.

Note: t_{PCLK} is the period of PCLK (CLK_HSB).

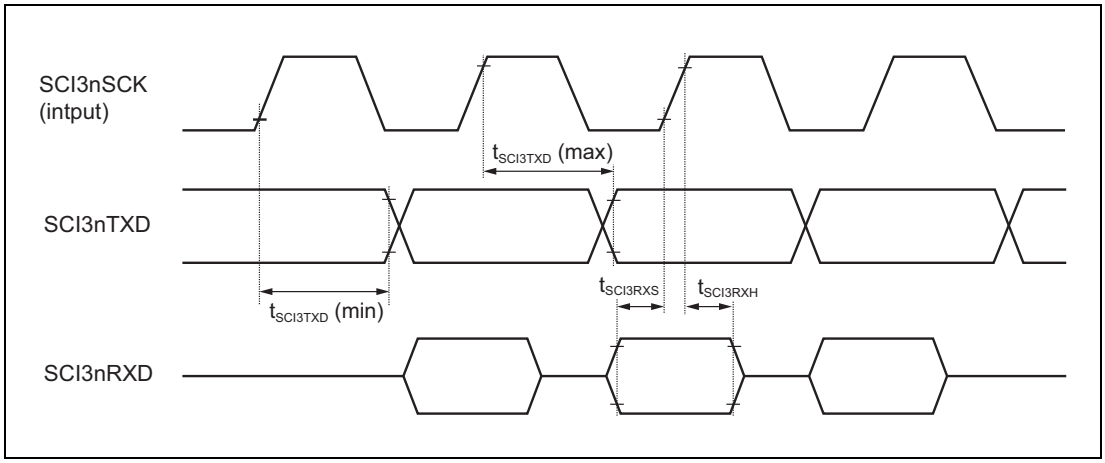


Figure 55.37 SCI3 Input/Output Timing, Clock Synchronous Mode (in Slave Mode)

55.3.12 RLIN3 Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 4
- Buffer type = Sch1

Table 55.54 RLIN3 Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RLIN3 transfer rate	r_{RLin}	LIN mode			20	kbps
	r_{RLine}	LIN extended baud rate*1			115.2	kbps
	r_{RLurt}	UART mode			8	Mbps

Note 1. The LIN extended baud rate is not part of the LIN standard specification.

55.3.13 RIIC Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = Sch1

Table 55.55 RIIC Timing (Normal Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				100	kHz
Bus free time (between stop/start condition)	t_{BUF}		4.7			µs
Hold time[*1]	t_{HD}: STA		4.0			µs
RIICnSCL clock low level width	t_{LOW}		4.7			µs
RIICnSCL clock high level time	t_{HIGH}		4.0			µs
Setup time for start/restart condition	t_{SU}: STA		4.7			µs
Data hold time	t_{HD}: DAT	CBUS compatible master	5.0			µs
		I^2C mode	0[*2]		[*3]	µs
Data setup time	t_{SU}: DAT		250			ns
Stop condition setup time	t_{SU}: STO		4.0			µs

Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Note 3. If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.

Table 55.56 RIIC Timing (Fast Mode)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
RIICnSCL clock period	f_{CLK}				400	kHz
Bus free time (between stop/start condition)	t_{BUF}		1.3			µs
Hold time[*1]	t_{HD}: STA		0.6			µs
RIICnSCL clock low level width	t_{LOW}		1.3			µs
RIICnSCL clock high level time	t_{HIGH}		0.6			µs
Setup time for start/restart condition	t_{SU}: STA		0.6			µs
Data hold time	t_{HD}: DAT	I^2C mode	0[*2]		[*5]	µs
Data setup time	t_{SU}: DAT		100[*3]			ns
Stop condition setup time	t_{SU}: STO		0.6			µs
Pulse width with spike suppressed by input filter	t_{SP}		0		[*4]	ns

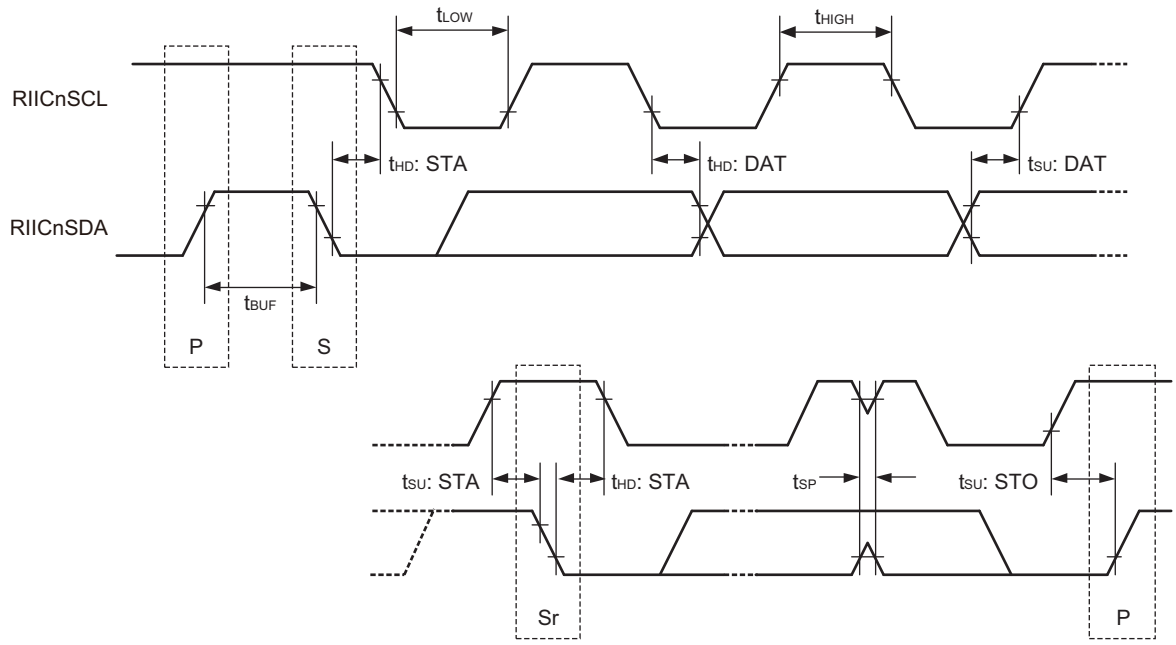
Note 1. At the start condition, the first clock pulse is generated after the hold time.

Note 2. The system requires a minimum of 300 ns hold time internally for the RIICnSDA signal (at VIH min. of RIICnSCL signal). In order to occupy the undefined area at the falling edge of RIICnSCL.

Note 3. The fast mode I^2C bus can be used in normal mode I^2C bus system. In this case, set the fast mode I^2C bus so that it meets the following conditions.
- If the system does not extend the RIICnSCL signal's low state hold time: t_{SU}: DAT ≥ 250 ns
- If the system extends the RIICnSCL signal's low state hold time:
Transmit the following data bit to the RIICnSDA line prior to releasing the RIICnSCL line (1250 ns: Normal mode I^2C bus specification).

Note 4. The filtered width is specified by the frequency of IICΦ and the value of RIICnMR3.NF[1:0].

Note 5. If the system does not extend the RIICnSCL signal low hold time (t_{Low}), only the maximum data hold time (t_{HD}: DAT) needs to be satisfied.



Note
P: Stop condition S: Start condition Sr: Restart condition

Figure 55.38 RIIC Timing

55.3.14 RS-CANFD Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = Sch1

Table 55.57 RS-CANFD Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate	r_{CN1}	Classical CAN mode			1	Mbps
	r_{CN2}	CAN FD mode (nominal bit rate)			1	Mbps
	r_{CN3}	CAN FD mode (data bit rate)			8	Mbps
Internal delay time	t_{DCIN}	$t_{INPUT} + t_{OUTPUT}$			50	ns

Note 1. For the configuration of the transfer speed, see **Section 23.5.1.3, Baud Rate**.

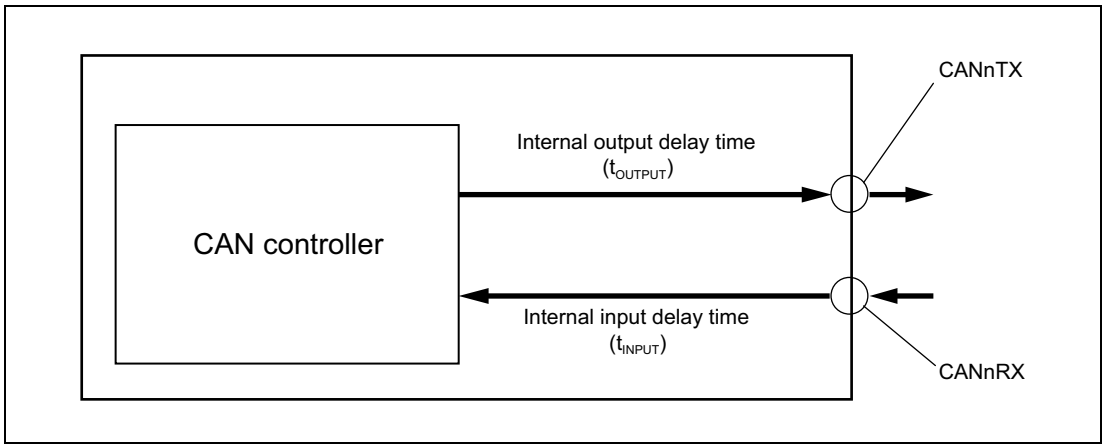


Figure 55.39 RS-CANFD Timing

55.3.15 FlexRay Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- CL =25 pF
- Buffer type = Sch1

Table 55.58 FlexRay Timing*1

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate	r_{FLXA}				10	Mbps

Note 1. Base of this specification is “FlexRay Electrical Physical Layer Specification V3.0.1, Oct-2010”

55.3.16 RSENT Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 4
- Buffer type = Sch1

Table 55.59 RSENT Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Tick Time			1		90	µs

55.3.17 Renesas High-speed Serial I/F Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- HSIF0_REFCLK: CL = 15 pF
- Buffer type = Sch1
- E0VCC = 5.0 ± 0.5 V or 3.3V ± 0.3 V

Table 55.60 External Reference Clock Input/Output Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
External Reference Clock Input frequency	f_{REFCKI}		10		20	MHz
External Reference Clock Input duty cycle	DCREFCKI		35		65	%
External Reference Clock output frequency	f_{REFCKO}		10		20	MHz
External Reference Clock output duty cycle	DCREFCKO		35		65	%

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- HSIF0_REFCLK: CL = 15 pF
- E0VCC = 5.0 ± 0.5 V or 3.3V ± 0.3 V
- HSIF0_TXDP, HSIF0_TXDN: CL = 5.0 pF

Table 55.61 RHSIF Transmit Data Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transmit data cycle	f_{TX}	Slow mode			5	M Baud
		Fast mode			320	M Baud
Transmit data delay time (HSIF0_REFCLK input)	$t_{REFITXDD}$	Slow mode	0		60	ns
		Fast mode*1				ns
Transmit data delay time (HSIF0_REFCLK output)	$t_{REFOTXDD}$	Slow mode RHSIFnREFCLK 20MHz	–20		20	ns
		Slow mode RHSIFnREFCLK 10MHz	15		85	ns
		Fast mode*1				ns

Note 1. Asynchronous

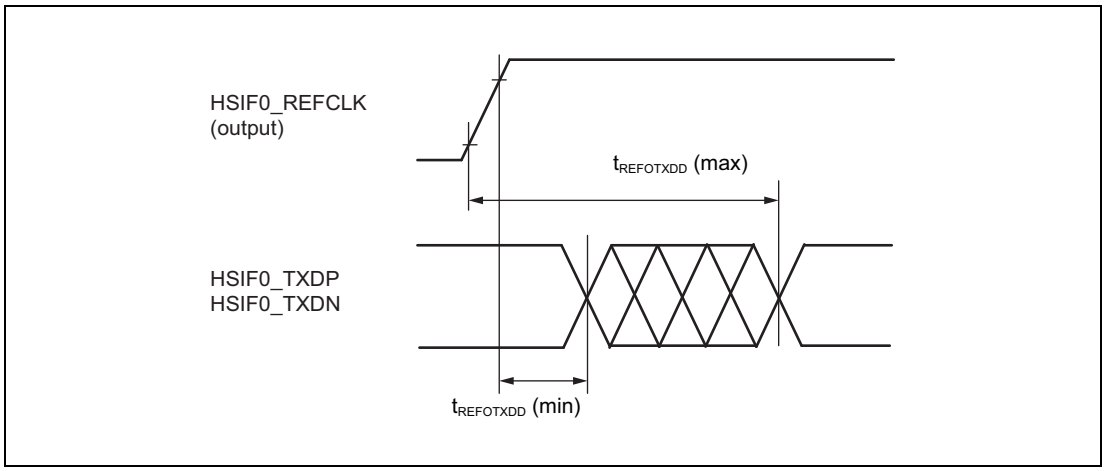


Figure 55.40 RHSIF Transmit Data Timing (HSIF0_REFCLK: output)

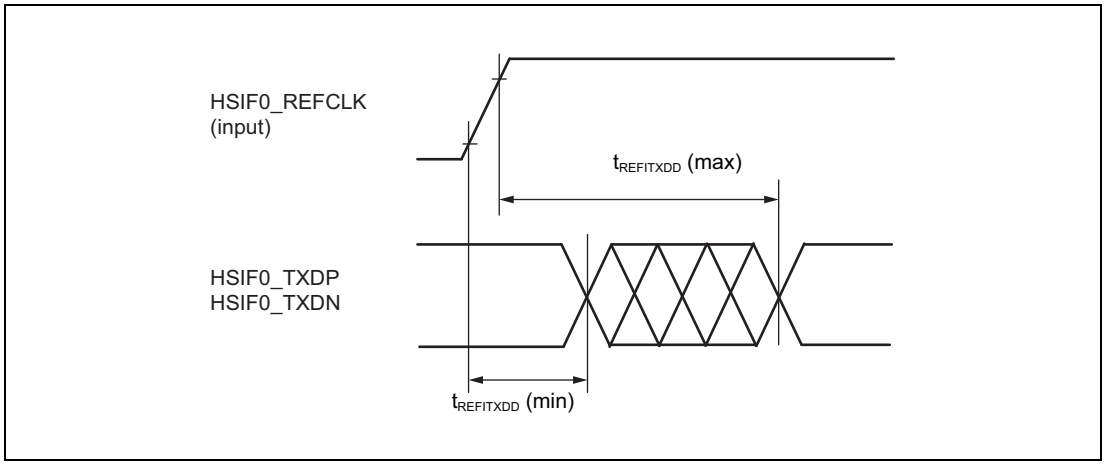


Figure 55.41 RHSIF Transmit Data Timing (HSIF0_REFCLK: input)

Table 55.62 RHSIF Receipt Data Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Receipt data cycle	f_{RX}	Slow mode			5	M Baud
		Fast mode			320	M Baud

55.3.18 CXPI Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 4
- Buffer type = Sch1

Table 55.63 CXPI Timing (CXPI-PWM Mode, Master node)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CXPI transfer rate	r_{CXPI}				20	kbps
CXPI cycle time	Tbit		50			µs
Required loop back delay time (from CXP1nTX to CXP1nRX)	t_{pwm_loop}				0.05 × Tbit	µs

Table 55.64 CXPI Timing (CXPI-PWM Mode, Slave node)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CXPI transfer rate	r_{CXPI}				20	kbps
CXPI cycle time	Tbit		50			µs
CXP1nTX output delay time (from CXP1nRX)	$t_{tx\_0\_pd}$		4 × t_{PCLK} *1		511 × t_{PCLK} + 4 × t_{PCLK} + 0.1 *1,*2	µs
Required loop back delay time (from CXP1nTX to CXP1nRX)	t_{pwm_loop}				0.05 × Tbit	µs

Note 1. t_{PCLK} is period of CLK_HSB

Note 2. "511 × t_{PCLK}" means maximum surge noise filter time. For details, see **Section 30.4.1.6, CXP1nFIL — CXPI Input Filter Setting Register**

Table 55.65 CXPI Timing (CXPI-NRZ Mode, Master or Slave node)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
CXPI transfer rate	r_{CXPI}				20	kbps
CXPI cycle time	Tbit		50			µs
Required loop back delay time (from CXP1nTX to CXP1nRX)	t_{nrz_loop}		1 × Tbit		4 × Tbit	µs

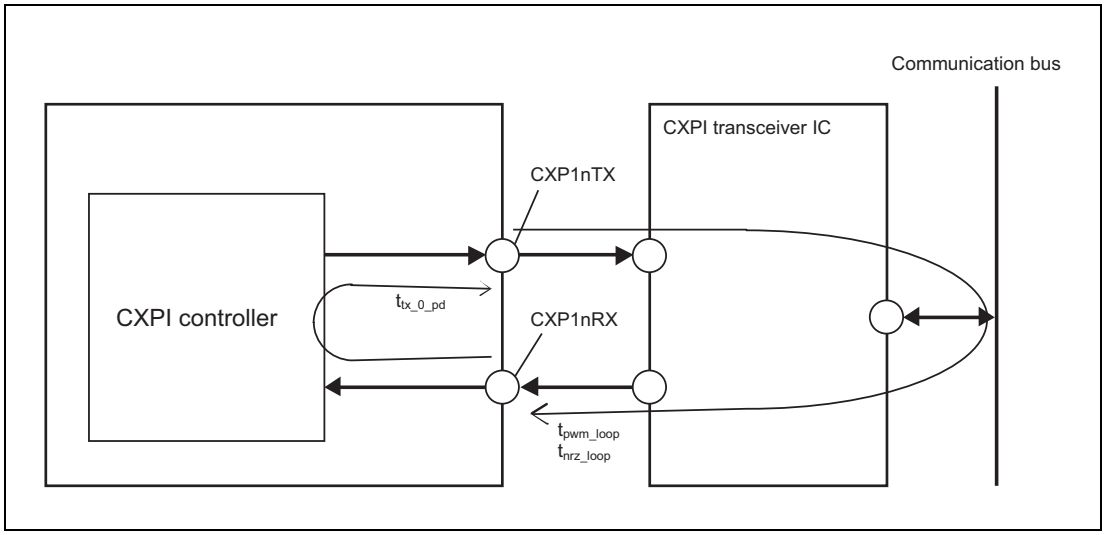


Figure 55.42 CXPI Block Diagram and Timing Path

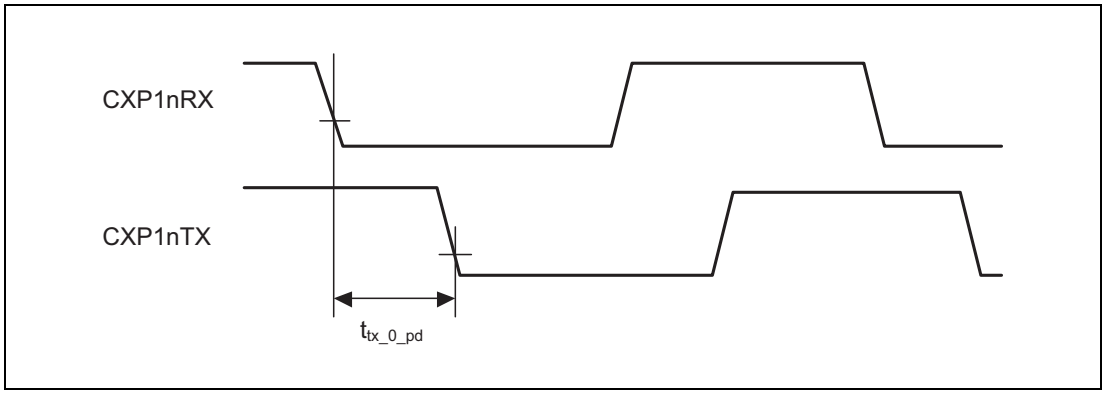


Figure 55.43 $t_{tx\_0\_pd}$ Waveform in CXPI PWM Mode

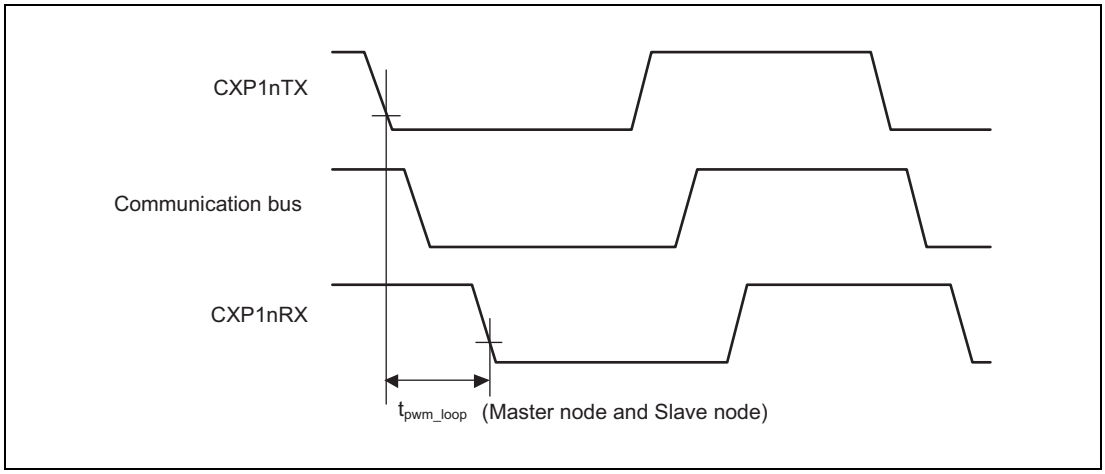


Figure 55.44 t_{pwm_loop} Waveform in CXPI PWM Mode

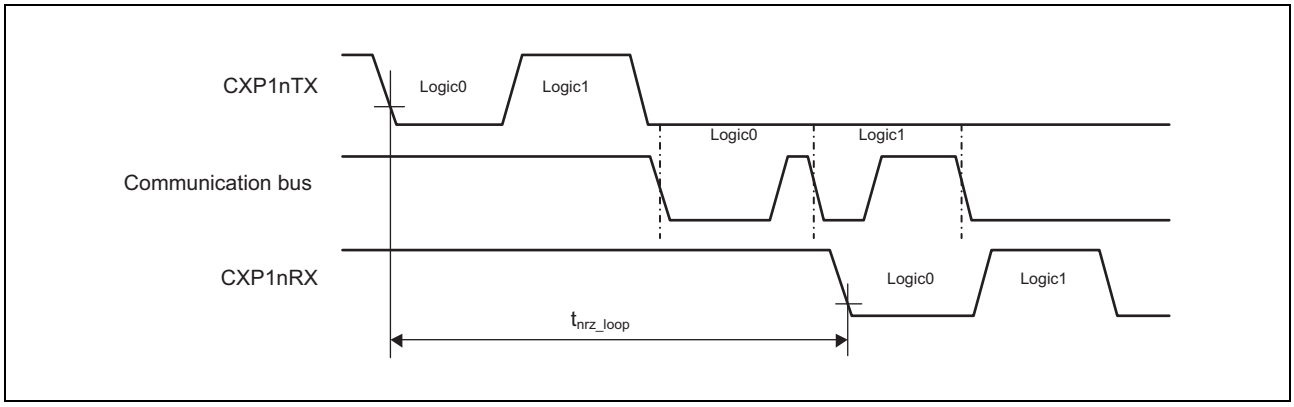


Figure 55.45 t_{nrz_loop} Waveform in CXPI NRZ Mode

55.3.19 Ethernet Timing

55.3.19.1 MII Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- CL = 15 pF
- Buffer type = TTL

Table 55.66 Ethernet Timing – 10 Mbit/s MII Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNBnTXCLK width	t_{MTC}		400 – 100 ppm	400	400 + 100 ppm	ns
ETNBnTXCLK high width	t_{MTCH}	*1	140	200	260	ns
ETNBnTXCLK low width	t_{MTCL}	*1	140	200	260	ns
ETNBnTXD [3:0] delay time	t_{MTXD}		0		25	ns
ETNBnTXEN delay time	t_{MTXE}		0		25	ns
ETNBnRXCLK width	t_{MRC}		400 – 100 ppm	400	400 + 100 ppm	ns
ETNBnRXCLK high width	t_{MRCH}	*1	140	200	260	ns
ETNBnRXCLK low width	t_{MRCL}	*1	140	200	260	ns
ETNBnRXD [3:0] setup time	t_{MRXDS}		10			ns
ETNBnRXD [3:0] hold time	t_{MRXDH}		10			ns
ETNBnRXDV, ETNBnRXER setup time	t_{MRDES}		10			ns
ETNBnRXDV, ETNBnRXER hold time	t_{MRDEH}		10			ns
ETNBnWOL delay time	t_{MWOL}		1		30	ns

Note 1. The duty cycle of ETNBnTXCLK and ETNBnRXCLK shall be between 35 to 65% (IEEE802.3).

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Cload = 15 pF
- Drive strength = 2
- Buffer type = TTL

Table 55.67 Ethernet Timing – 100 Mbit/s MII Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNBnTXCLK width	t_{MTC}		40 – 100 ppm	40	40 + 100 ppm	ns
ETNBnTXCLK high width	t_{MTCH}	*1	14	20	26	ns
ETNBnTXCLK low width	t_{MTCL}	*1	14	20	26	ns
ETNBnTXD [3:0] delay time	t_{MTXD}		0		25	ns
ETNBnTXEN delay time	t_{MTXE}		0		25	ns
ETNBnRXCLK width	t_{MRC}		40 – 100 ppm	40	40 + 100 ppm	ns
ETNBnRXCLK high width	t_{MRCH}	*1	14	20	26	ns
ETNBnRXCLK low width	t_{MRCL}	*1	14	20	26	ns
ETNBnRXD [3:0] setup time	t_{MRXDS}		10			ns
ETNBnRXD [3:0] hold time	t_{MRXDH}		10			ns
ETNBnRXDV, ETNBnRXER setup time	t_{MRDES}		10			ns
ETNBnRXDV, ETNBnRXER hold time	t_{MRDEH}		10			ns
ETNBnWOL delay time	t_{MWOL}		1		30	ns

Note 1. The duty cycle of ETNBnTXCLK and ETNBnRXCLK shall be between 35 to 65% (IEEE802.3).

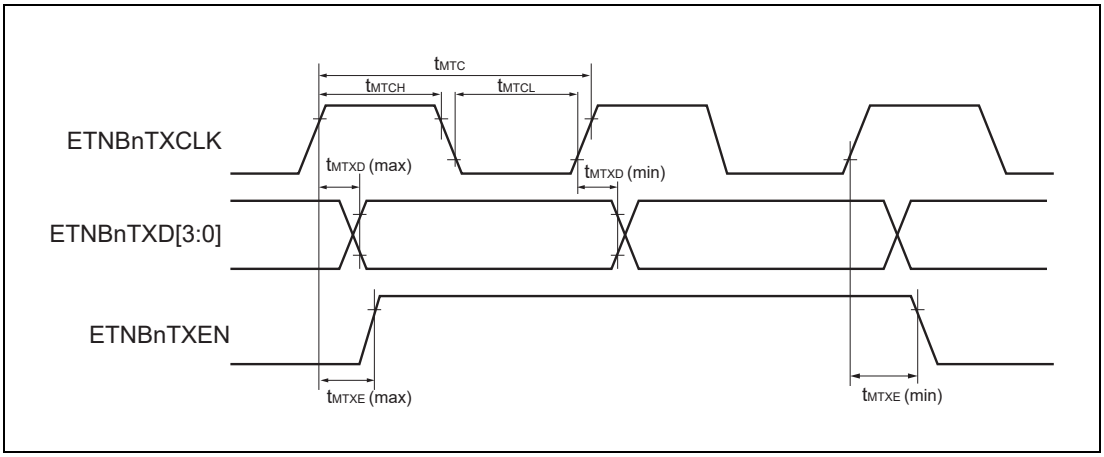


Figure 55.46 Ethernet Timing – MII Transmitter

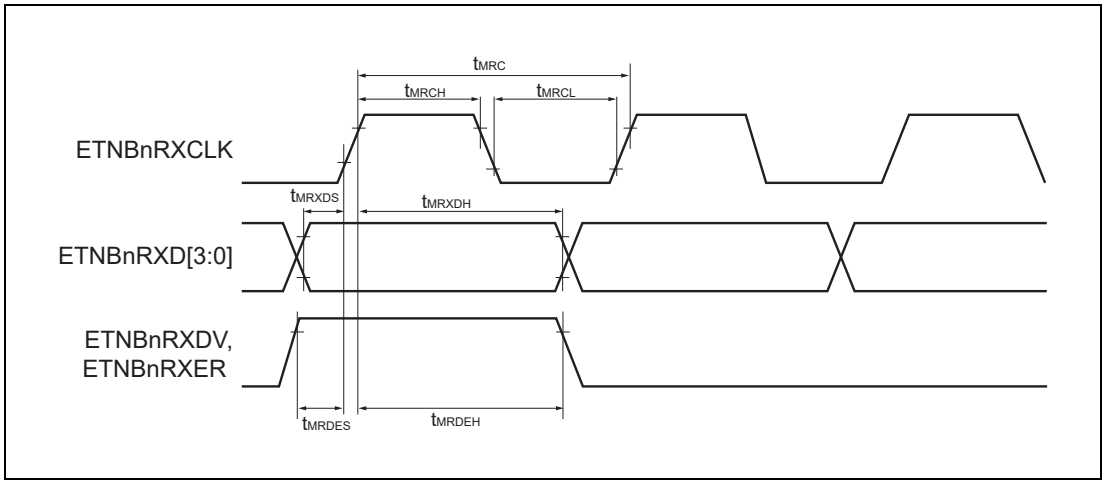


Figure 55.47 Ether net Timing – MII Receiver

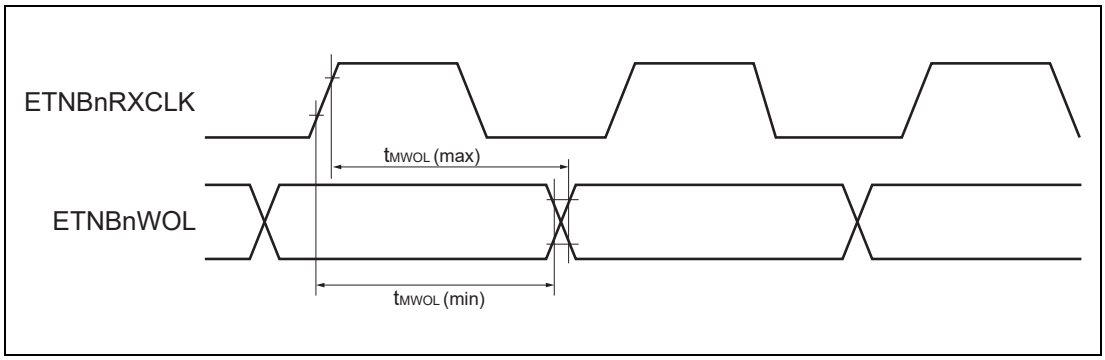


Figure 55.48 Ethernet Timing – MII WOL

55.3.19.2 RMII Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- CL = 15 pF
- Drive strength = 2
- Buffer type = TTL

Table 55.68 Ethernet Timing – RMII Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
ETNB0REF50CK width	t_{RMC}		20 – 50 ppm	20	20 + 50 ppm	ns
ETNB0REF50CK high width	t_{RMCH}	*1	7		13	ns
ETNB0REF50CK low width	t_{RMCL}	*1	7		13	ns
ETNB0RXD [1:0], ETNB0CRS _DV and ETNB0RXER setup time	t_{RMS}		4			ns
ETNB0RXD [1:0], ETNB0CRS_DV and ETNB0RXER hold time	t_{RMH}		2			ns
ETNB0TXD [1:0], ETNB0TXEN output delay time	t_{RMD}		2		16	ns
ETNB0WOL delay time	t_{RMWOL}		1		30	ns

Note 1. The duty cycle of ETNB0REF50CK shall be between 35 to 65% (RMII™ specification).

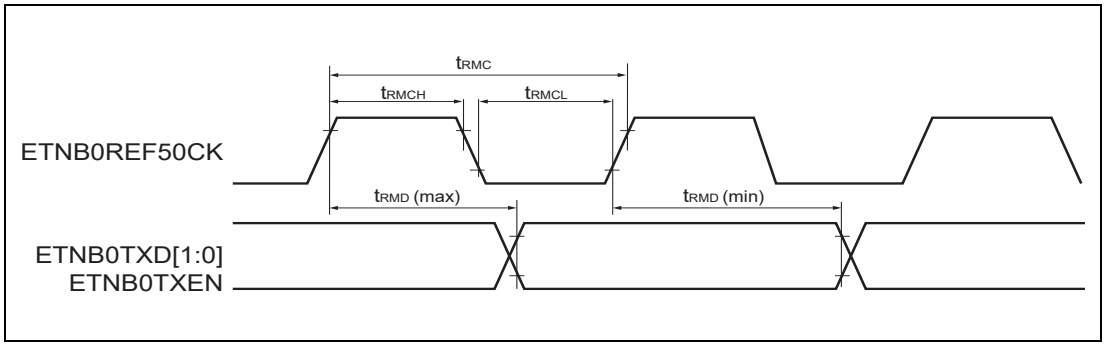


Figure 55.49 Ethernet Timing – RMII Transmitter

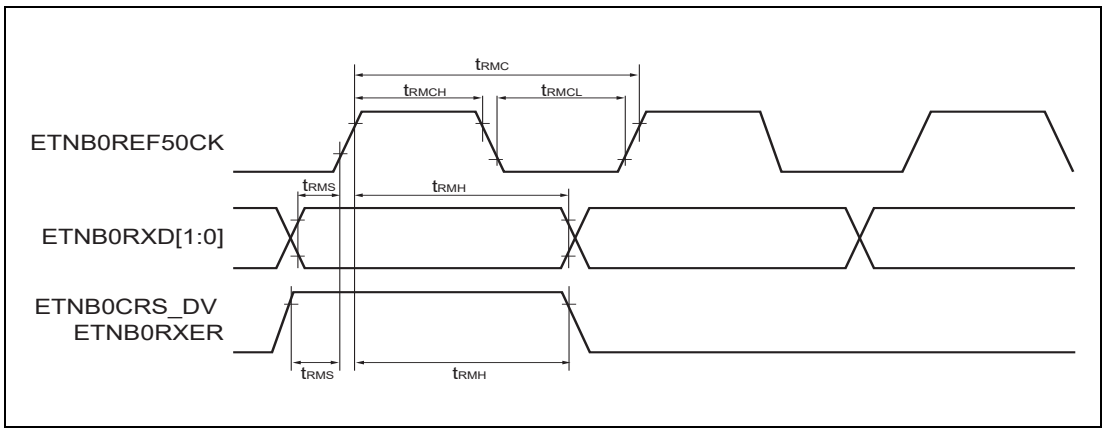


Figure 55.50 Ethernet Timing – RMII Receiver

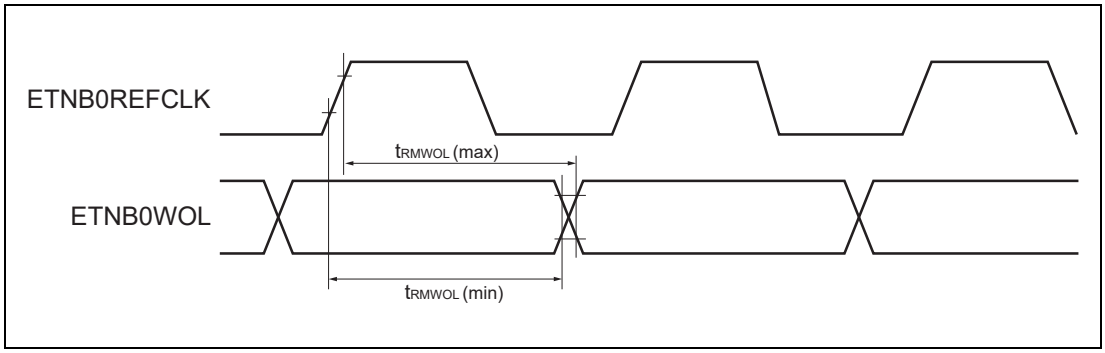


Figure 55.51 Ethernet Timing – RMII WOL

55.3.19.3 SGMII Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.69 Ethernet Timing – SGMII REFCK Characteristics

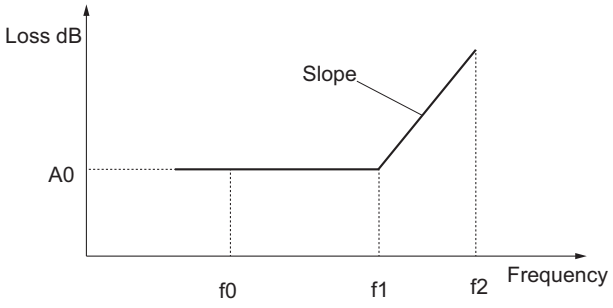
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Clock input frequency	t_{CIF}		40 – 100ppm		40 + 100 ppm	ns
Clock input duty cycle	t_{CIDC}		45		55	%
Clock input rising/falling time (20%-80%)	t_{CIRFT}				3	ns
Clock Input Total jitter (Dj+ 14* rms Random jitter)	t_{CITJ}				73*1	ps peak to peak

Note 1. 12 kHz to 20 MHz rms jittrer = 3 ps.

Table 55.70 Ethernet Timing – SGMII Tx buffer Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Signaling speed	t_{SS}		1.25 – 100 ppm		1.25 + 100 ppm	GBd
Total output jitter (Dj+14*rms random jitter)	t_{TOJ}				300	ps peak to peak
VOD rise/fall time (20% - 80%)	t_{VRFT}		60		250	ps
Differential output return loss (min)	DORL				*1	dB

Note 1. See the **Figure 55.52, Differential output return loss** for details of differential output return loss.



Parameter value		Unit
A0	-8	dB
f0	50	MHz
f1	0.625	GHz
f2	1.25	GHz
Slope	10	dB/dec

Figure 55.52 Differential output return loss

Table 55.71 Ethernet Timing – SGMII Rx buffer Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Signaling speed	t_{SS}		1.25 – 100 ppm		1.25 + 100 ppm	GBd
Total input jitter tolerance*2 (Dj+14*rms random jitter)	t_{TIJT}				400	ps peak to peak
setup/hold time*1, *3	t_{SHT}		120			ps

Note 1. Measured at 50% of the transition

Note 2. Only supported by RH850/U2A

Note 3. Only supported by RH850/U2A-EVA

Table 55.72 Ethernet Timing – Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
SGMII power stabilization time	t_{SPST}				1.2	ms

55.3.19.4 Management Interface

Timing of management interface (ETNBnMDC and ETNBnMDIO) depends on software. It is necessary to adjust wait time according to AC specification of PHY.

55.3.20 PSI5 Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Drive strength = 4

Table 55.73 PSI5 Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Bit time		125 kbps (Low speed)	7.6	8.0	8.4	µs
		189 kbps (High speed)	5.0	5.3	5.6	µs
		250 kbps (PAS compatibility)	3.8	4.0	4.2	µs
Gap time		125 kbps (Low speed)	8.4			µs
		189 kbps (High speed)	5.6			µs
		250 kbps (PAS compatibility)	2.0			µs

55.3.21 PSI5-S Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.74 PSI5-S Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
PSI5-S transfer rate	r_{PSI5S1}	PSI5-S mode			5.33	Mbps
	r_{PSI5S2}	UART mode			5.33	Mbps
Output clock cycle	$t_{PSIScyc}$	PSI5-S mode	37.5			ns
Output clock pulse width	$t_{PSISCKW}$	PSI5-S mode	0.3 x $t_{PSIScyc}$		0.5 x $t_{PSIScyc}$	ns

55.3.22 Timer Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.75 Timer Input Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TAUDnIm input high level width	t_{WTDIH}		(S + 1) x 1/fs*1			ns
TAUDnIm input low level width	t_{WTDIL}		(S + 1) x 1/fs*1			ns
TAUDnIm pulse rejection width*2	t_{WTDRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
TAUJnIm input high level width	t_{WTJIH}	Analog noise filter	600*3			ns
		Digital noise filter	(S + 1) x 1/fs*1			ns
TAUJnIm input low level width	t_{WTJIL}	Analog noise filter	600*3			ns
		Digital noise filter	(S + 1) x 1/fs*1			ns
TAUJnIm pulse rejection width*2	t_{WTJRJ}	Analog noise filter	100		600*3	ns
		Digital noise filter	(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
TAPAnESO input high level width*4	t_{WTPIH}	Analog noise filter	600			ns
		Digital noise filter	(S + 1) x 1/fs*1			ns
TAPAnESO input low level width*4	t_{WTPIL}	Analog noise filter	600			ns
		Digital noise filter	(S + 1) x 1/fs*1			ns
TAPAnESO pulse rejection width*2, *4	t_{WTPRJ}	Analog noise filter	100		600	ns
		Digital noise filter	(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
ENCAnTINm input high level width	t_{WENIH}		(S + 1) x 1/fs*1			ns
ENCAnTINm input low level width	t_{WENIL}		(S + 1) x 1/fs*1			ns
ENCAnTINm pulse rejection width*2	t_{WENRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns
TSG3nPTSIm/ENCAnEx, TSG3nCLKI high level width	t_{WTGIH}		(S + 1) x 1/fs*1			ns
TSG3nPTSIm/ENCAnEx, TSG3nCLKI low level width	t_{WTGIL}		(S + 1) x 1/fs*1			ns
TSG3nPTSIm/ENCAnEx, TSG3nCLKI pulse rejection width*2	t_{WTGRJ}		(S – 1) x 1/fs*1		(S + 1) x 1/fs*1	ns

Note 1. S: Number of sampling times
fs: The value given by the following formula

$$fs = \frac{f_{DNFCK}}{PRS}$$

f_{DNFCK}: frequency of CLKA_TAUJ (for TAUJ2 and TAUJ3), CLK_HSB (others)
PRS: 1, 2, 4, 8, … , 128

Note 2. Input pulse shorter than the given min. value will be filtered out. Input pulses between min. and max. value result in an undefined signal condition (i.e. pulses might be filtered or not.
This characteristic is not tested in production.

Note 3. When CLK_LSIOSC is selected by CLKA_TAUJ register, at least one clock period of CLK_LSIOSC (4.6 µs) is required for activation of input signal for that domain. Any input pulses with less than 4.6 µs width may be rejected.

Note 4. By-pass of filter is possible. For details, see **Section 2.7.2.10, ANF/DNF Type F1**.

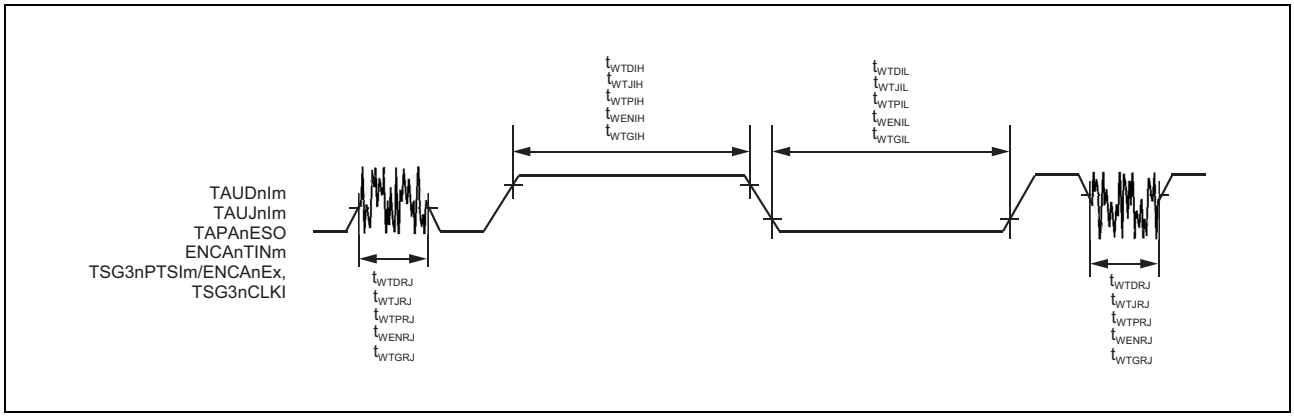


Figure 55.53 Timer Input Timing

55.3.23 GTM Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.76 GTM Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
GTM input high level width	t_{WGTIH}		4 × Tsamp*1			ns
GTM input low level width	t_{WGTIL}		4 × Tsamp*1			ns
GTM output cycle time	t_{CYGTO}		4 × Tsamp*1			ns
GTMECLKn output cycle	$t_{CYGTECLK}$		8 × Tsamp*1			ns
GTMECLKn output high level width	t_{WGTOH}		4 × Tsamp*1 – 10			ns
GTMECLKn output low level width	t_{WGTOL}		4 × Tsamp*1 – 10			ns

Note 1. Tsamp = $1/f_{CLK\_UHSB}$

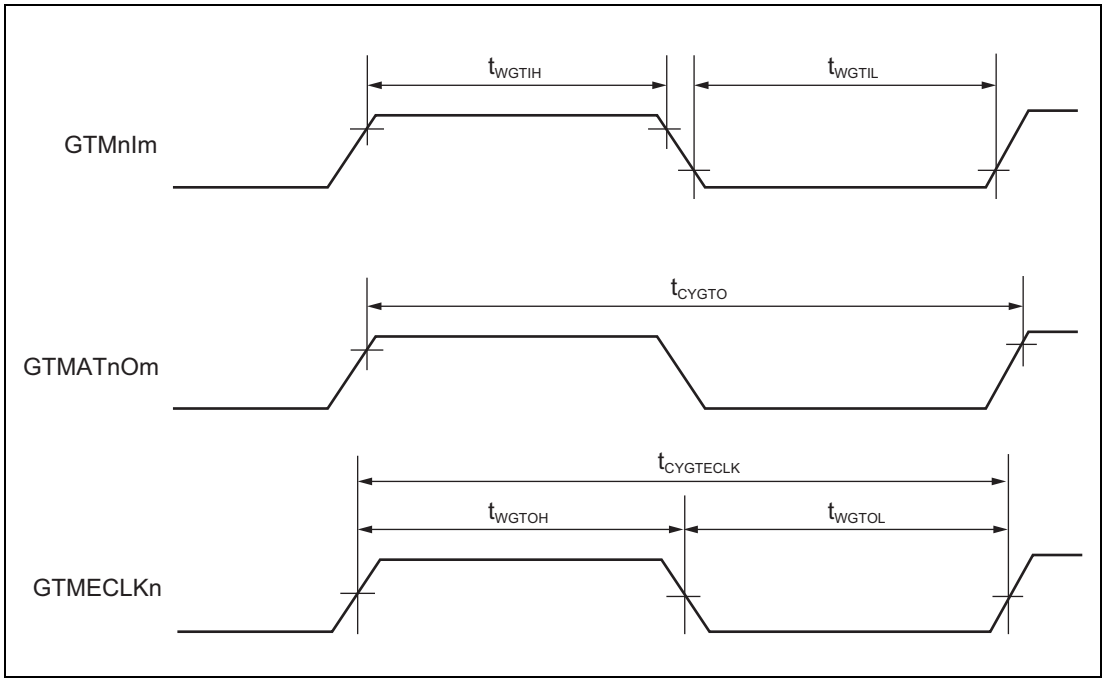


Figure 55.54 GTM Timing

55.3.24 Emergency shut-Off (ESO) Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.77 ESO Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
GTMATnOx Hi-z delay time	t_{DESO}				50	ns

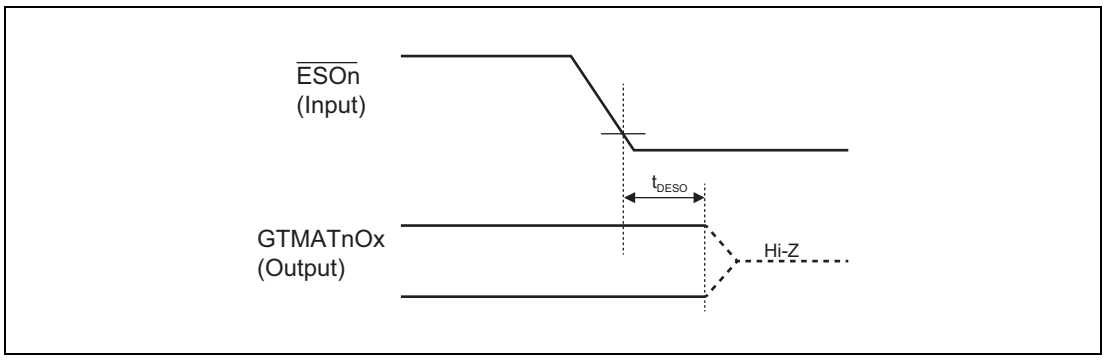


Figure 55.55 ESO Timing

55.3.25 Debug Reset Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.78 Debug Reset Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{TRST}}$ input low level width	t_{WTRL}		600			ns
$\overline{\text{TRST}}$ pulse rejection*1	t_{WTRRJ}		100		600	ns
$\overline{\text{AURORES1}}$, $\overline{\text{AURORES2}}$, $\overline{\text{AURORESPD}}$ input low level width	t_{WARSL}		600			ns
$\overline{\text{AURORES1}}$, $\overline{\text{AURORES2}}$, $\overline{\text{AURORESPD}}$ pulse rejection width*1	t_{WARRJ}		100		600	ns
$\overline{\text{ERAMRESPD}}$, $\overline{\text{ERAMRES2}}$ input low level width	t_{WERSL}		600			ns
$\overline{\text{ERAMRESPD}}$, $\overline{\text{ERAMRES2}}$ pulse rejection width*1	t_{WERRJ}		100		600	ns

Note 1. Input pulses shorter than the given min. value will be filtered out (resulting in no interrupt detected). Input pulses between min. and max. value result in an undefined interrupt signal condition (i.e. pulses might be filtered out or not).

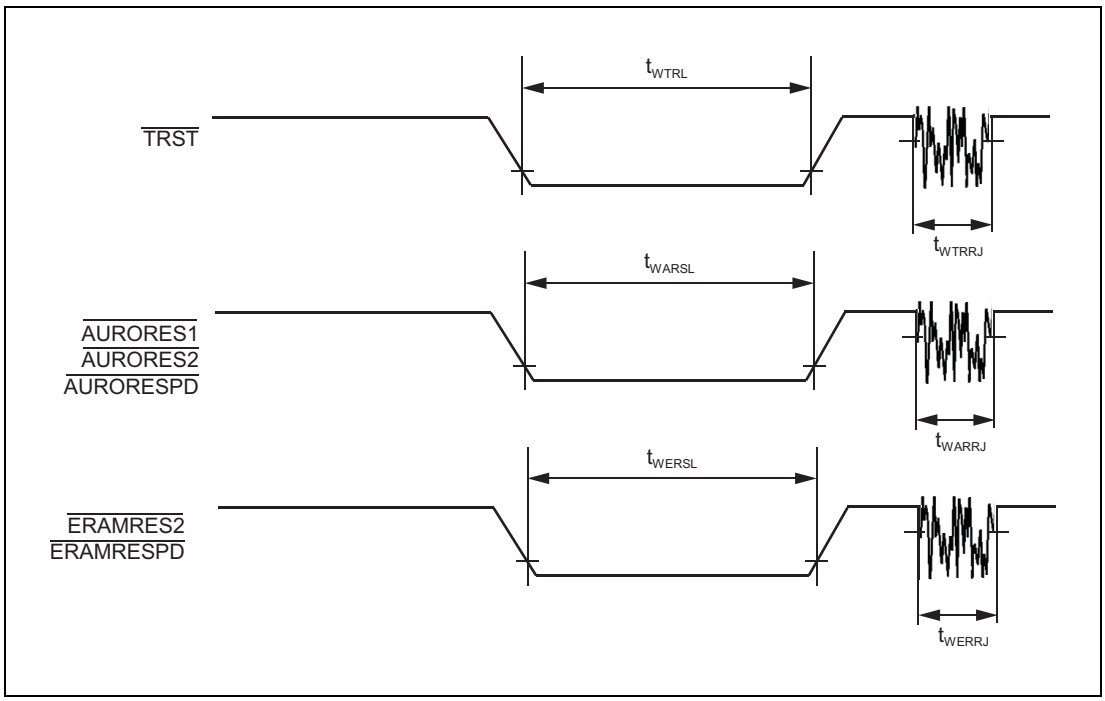


Figure 55.56 Debug Reset Timing

55.3.26 Nexus Interface Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = TTL

Table 55.79 Nexus Interface Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TCK Cycle width	t_{TCKW}		50			ns
TCK high level width	t_{TCKWH}		21			ns
TCK low level width	t_{TCKWL}		21			ns
TMS/TDI setup time	t_{TISU}		6			ns
TMS/TDI hold time	t_{TIH}		6			ns
TDO output delay time	t_{TDOD}				17	ns
$\overline{RDY}$ delay time	t_{RDYD}				17	ns
TCK/$\overline{TRST}$/TMS/TDI input rising time	t_{TIR}				4	ns
TCK/$\overline{TRST}$/TMS/TDI input falling time	t_{TIF}				4	ns

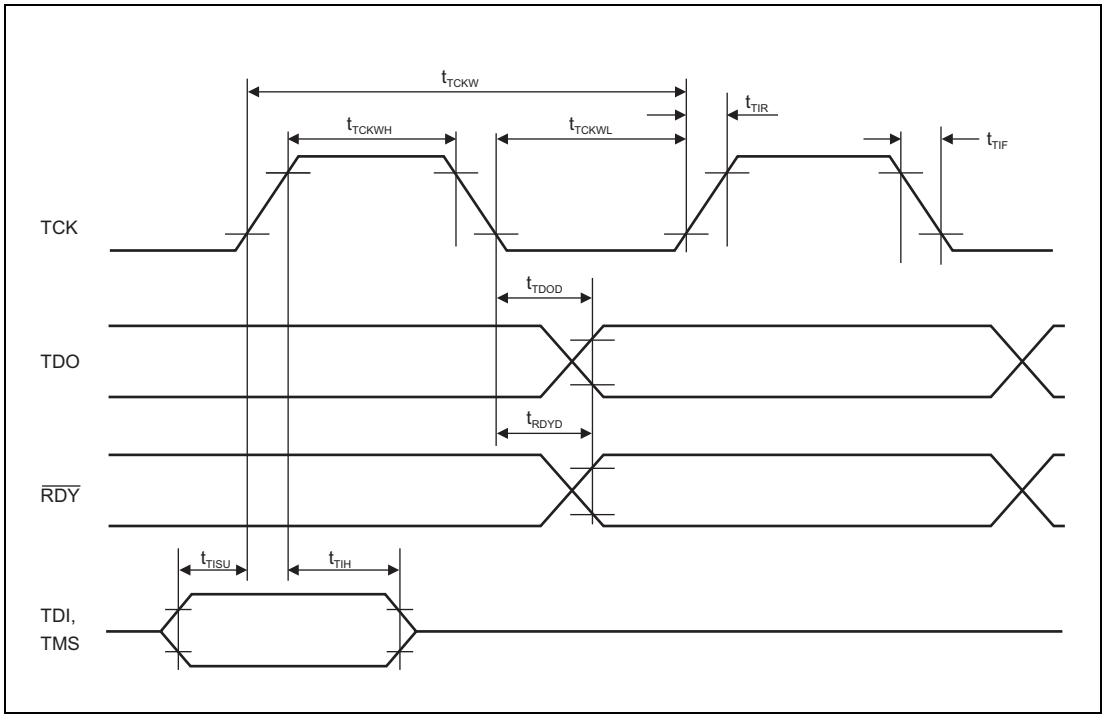


Figure 55.57 Nexus Interface Timing

55.3.27 LPD (4pin) Interface Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.80 LPD (4pin) Interface Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
LPDCLK cycle time	$t_{LPDCKCYC}$		25			ns
LPDCLK high/low level width	t_{LPDCKW}		4.5			ns
LPDCLK input rising/falling time	$t_{LPDCKRF}$				8	ns
LPDI setup time	t_{LPDSU}		3			ns
LPDI hold time	t_{LPDH}		3			ns
LPDCLKO cycle time	$t_{LPDCKOCYC}$		25			ns
LPDCLKO high/low level width	$t_{LPDCKOW}$		t_{LPDCKW} - 2			ns
LPDCLK to LPDCLKO delay time	$t_{LPDCKOD}$				44	ns
LPDO output delay	t_{LPDOD}		0		18	ns

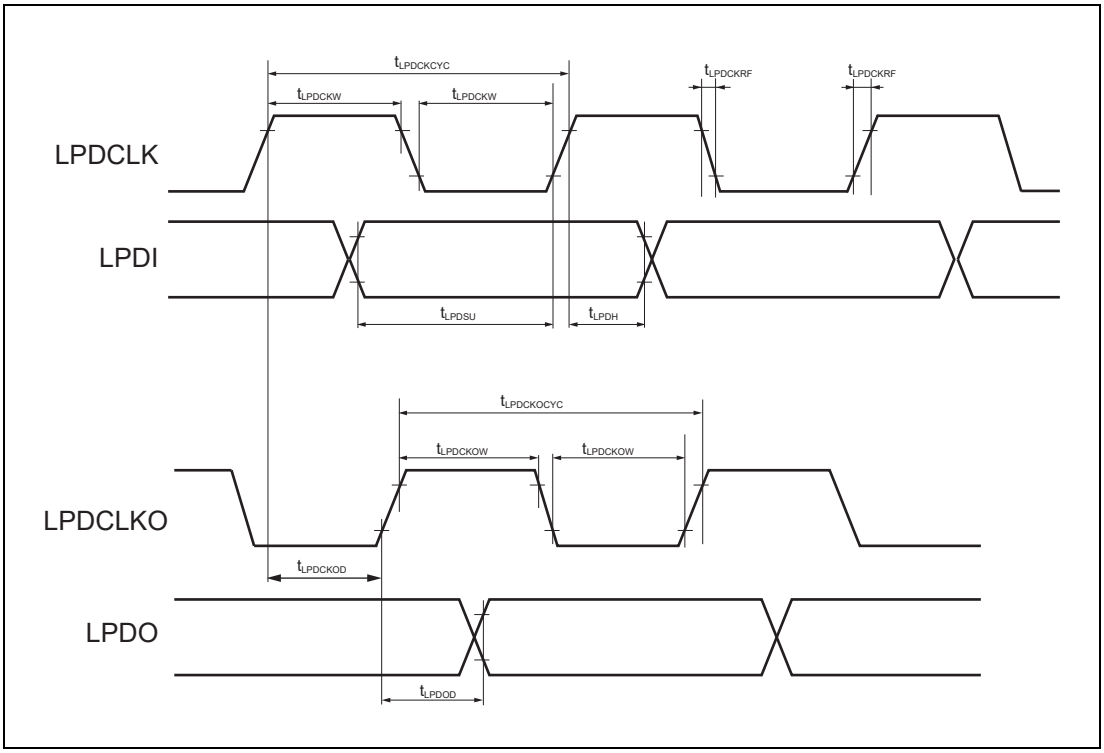


Figure 55.58 LPD (4pin) Interface Timing

55.3.28 AUDR Interface Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- CL = 10 pF
- Drive strength = 2
- Buffer type = SHMT1 ($\overline{\text{AUDRST}}$), TTL (Others)

Table 55.81 AUDR Interface Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
AUDCK cycle time	$t_{AUCKcyc}$		20			ns
AUDCK high level width	t_{AUCKH}		0.4 x $t_{AUCKcyc}$			ns
AUDCK low level width	t_{AUCKL}		0.4 x $t_{AUCKcyc}$			ns
$\overline{\text{AUDRST}}$ setup time	t_{AURSTS}		12			ns
$\overline{\text{AUDRST}}$ pulse width	t_{AURSTW}		5 x $t_{AUCKcyc}$			ns
$\overline{\text{AUDSYNC}}$ setup time	t_{AUSYS}		10			ns
$\overline{\text{AUDSYNC}}$ hold time	t_{AUSYH}		5			ns
AUDATA input setup time	t_{AUDTS}		8			ns
AUDATA input hold time	t_{AUDTH}		5			ns
AUDATA output delay	t_{AUDTD}				$t_{AUCKcyc}$ – 3.4	ns

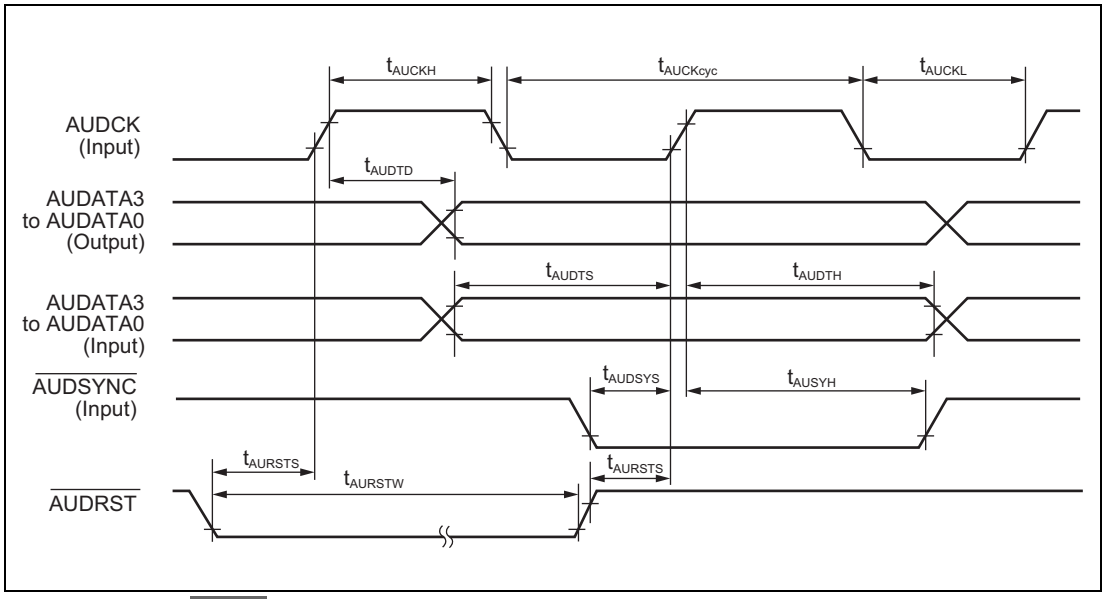


Figure 55.59 $\overline{\text{AUDR}}$ Interface Timing

55.3.29 Aurora Interface Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

55.3.29.1 Aurora Interface — 1.25 Gbps baud rate

Table 55.82 Aurora Interface Operating Condition – 1.25Gbps baud rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Rise/Fall Time	t_{AITRF}	*1	60			ps
Deterministic jitter	t_{AIDJ}				0.17	UI
Total jitter	t_{AITJ}				0.35	UI
Output skew	t_{AIOS}	*2			25	ps
Multiple output skew	t_{AIMOS}	*3			1000	ps
Unit interval	t_{AIUI}		800 - 100 ppm		800 +100 ppm	ps

Note 1. At driver output.
Note 2. Skew at transmitter output between the differential pair.
Note 3. Skew at transmitter output between lanes of a multi-lane channel.

55.3.29.2 Aurora Interface — 2.5 Gbps baud rate

Table 55.83 Aurora Interface Operating Condition – 2.5Gbps baud rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Rise/Fall Time	t_{AITRF}	*1	40			ps
Deterministic jitter	t_{AIDJ}				0.17	UI
Total jitter	t_{AITJ}				0.35	UI
Output skew	t_{AIOS}	*2			20	ps
Multiple output skew	t_{AIMOS}	*3			1000	ps
Unit interval	t_{AIUI}		400 - 100 ppm		400 + 100 ppm	ps

Note 1. At driver output.
Note 2. Skew at transmitter output between the differential pair.
Note 3. Skew at transmitter output between lanes of a multi-lane channel.

55.3.29.3 Aurora Interface — 3.125 Gbps baud rate

Table 55.84 Aurora Interface Operating Condition – 3.125Gbps baud rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Rise / Fall Time	t_{AITRF}	*1	30			ps
Deterministic jitter	t_{AIDJ}				0.17	UI
Total jitter	t_{AITJ}				0.35	UI
Output skew	t_{AIOS}	*2			15	ps
Multiple output skew	t_{AIMOS}	*3			1000	ps
Unit interval	t_{AIUI}		320 -100 ppm		320 + 100 ppm	ps

Note 1. At driver output.
Note 2. Skew at transmitter output between the differential pair.
Note 3. Skew at transmitter output between lanes of a multi-lane channel.

55.3.29.4 Aurora Interface — 5 Gbps baud rate

Table 55.85 Aurora Interface Operating Condition – 5Gbps baud rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Rise/Fall Time	t_{AITRF}	*1	30			ps
Deterministic jitter	t_{AIDJ}				0.25	UI
Total jitter	t_{AITJ}				0.35	UI
Output skew	t_{AIOS}	*2			15	ps
Multiple output skew	t_{AIMOS}	*3			1000	ps
Unit interval	t_{AIUI}		200 - 100 ppm		200 + 100 ppm	ps

Note 1. At driver output.
Note 2. Skew at transmitter output between the differential pair.
Note 3. Skew at transmitter output between lanes of a multi-lane channel.

55.3.29.5 Aurora Interface — 6.25 Gbps baud rate

Table 55.86 Aurora Interface Operating Condition – 6.25Gbps baud rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Rise / Fall Time	t_{AITRF}	*1	30			ps
Deterministic jitter	t_{AIDJ}				0.25	UI
Total jitter	t_{AITJ}				0.35	UI
Output skew	t_{AIOS}	*2			15	ps
Multiple output skew	t_{AIMOS}	*3			1000	ps
Unit interval	t_{AIUI}		160 - 100 ppm		160 +100 ppm	ps

Note 1. At driver output.
Note 2. Skew at transmitter output between the differential pair.
Note 3. Skew at transmitter output between lanes of a multi-lane channel.

55.3.29.6 Aurora Interface — Transmitter clock Timing

Table 55.87 Aurora Interface Transmitter clock timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Reference clock frequency	f_{AICLK}	1.25 Gbps baud rate		62.5		MHz
		2.5 Gbps baud rate		125		MHz
		3.125 Gbps baud rate		156.25		MHz
		5 Gbps baud rate		125		MHz
		6.25 Gbps baud rate		156.25		MHz
Reference clock rise time	t_{AICTR}			200	400	ps
Reference clock fall time	t_{AICTF}			200	400	ps
Reference clock duty cycle			45		55	%
Reference clock total jitter	t_{AICTJ}	*1			40*2	ps
Stability	$t_{AICSTAB}$				50	ppm

Note 1. Peak to peak.

Note 2. Phase noise of CICREF[P/N] should be below the line of **Figure 55.60**.

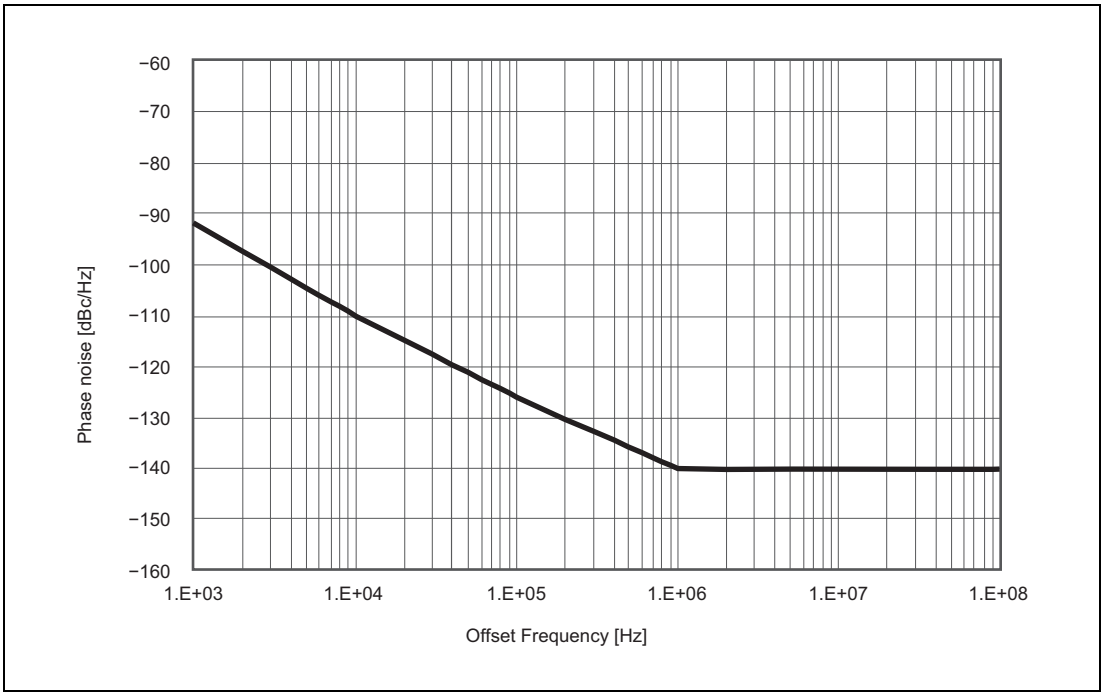


Figure 55.60 Phase noise of CICREF[P/N]

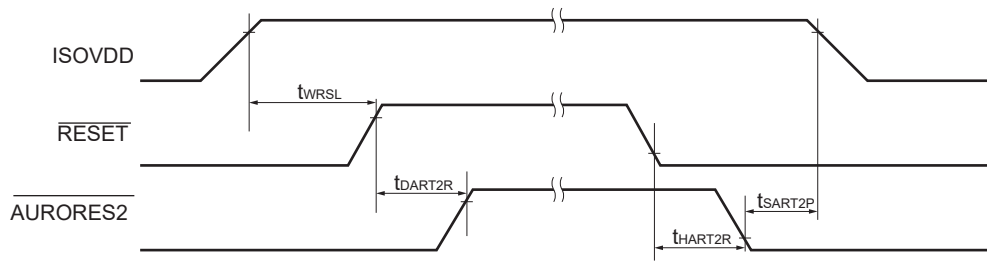
55.3.30 Debug Resource (Aurora, ERAM) Specific Reset Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

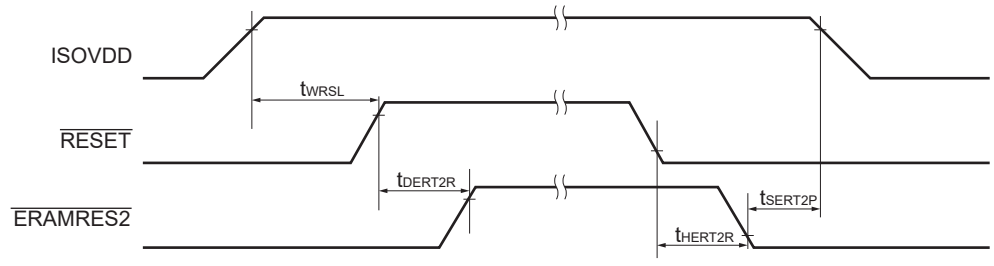
Table 55.88 Debug Resource (Aurora, ERAM) Specific Reset Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{AURORES2}}$ release timing delay	t_{DART2R}		0		400	ns
$\overline{\text{AURORES2}}$ inactive hold time	t_{HART2R}		2.4			µs
$\overline{\text{AURORES2}}$ setup time	t_{SART2P}		2.4			µs
$\overline{\text{ERAMRES2}}$ release timing delay	t_{DERT2R}		0		400	ns
$\overline{\text{ERAMRES2}}$ inactive hold time	t_{HERT2R}		2.4			µs
$\overline{\text{ERAMRES2}}$ assert setup time	t_{SERT2P}		2.4			µs
$\overline{\text{AURORES1}}$, $\overline{\text{AURORESPD}}$ release timing delay	$t_{DART2AVD}$		2.4			µs
$\overline{\text{AURORES1}}$, $\overline{\text{AURORESPD}}$ inactive hold time	$t_{SART2AVD}$		2.4			µs
$\overline{\text{ERAMRESPD}}$ release timing delay	$t_{DERT2EVD}$		2.4			µs
$\overline{\text{ERAMRESPD}}$ inactive hold time	$t_{SERT2EVD}$		2.4			µs



Note: About the parameter t_{WRSL}, refer to "$\overline{\text{RESET}}$ input level width" in **Section 55.3.6.1, RESET Timing**

Figure 55.61 $\overline{\text{AURORES2}}$ AC Timing



Note: About the parameter t_{WRSL}, refer to "$\overline{\text{RESET}}$ input low level width" in **Section 55.3.6.1, RESET Timing**

Figure 55.62 $\overline{\text{ERAMRES2}}$ AC Timing

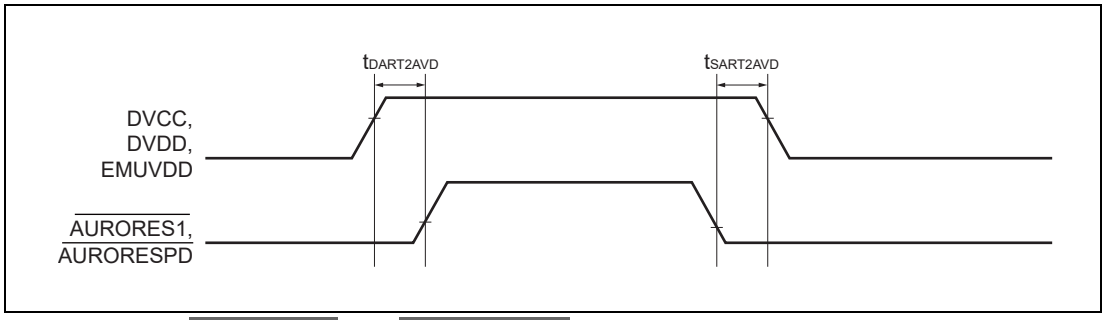


Figure 55.63 $\overline{\text{AURORES1}}$ and $\overline{\text{AURORESPD}}$ AC Timing

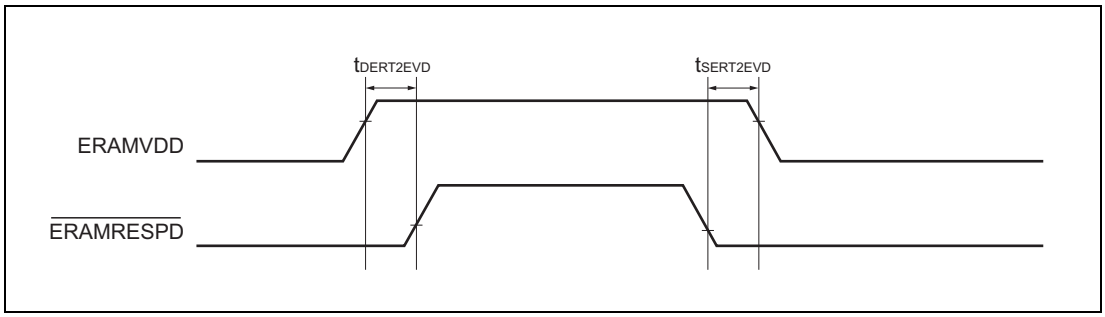


Figure 55.64 $\overline{\text{ERAMRESPD}}$ AC Timing

55.3.31 Debug Event Interface Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.89 Debug Event Interface Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
$\overline{\text{EVTI}}$ input high level width	t_{WEVIH}	*1	2 ×t_{MCKW} *3			ns
		*2	2 ×t_{TCKW}			ns
$\overline{\text{EVTI}}$ input low level width	t_{WEVIL}	*1	2 ×t_{MCKW} *3			ns
		*2	2 ×t_{TCKW}			ns
$\overline{\text{EVTO}}$ output high level width	t_{WEVOH}		t_{MCKW} *3			ns
$\overline{\text{EVTO}}$ output low level width	t_{WEVOL}		t_{MCKW} *3			ns
$\overline{\text{MSYN}}$ input high level width	t_{WMSNH}		2 ×t_{MCKW} *3			ns
$\overline{\text{MSYN}}$ input low level width	t_{WMSNL}		2 ×t_{MCKW} *3			ns

Note 1. When used as event trigger.

Note 2. When used as break input.

Note 3. t_{MCKW} is the cycle of the clock (MCKO) obtained by dividing the CPU clock (CLK_CPU) and the value must be divided by 16 (default). For details, refer to the *RH850/U2A-EVA Group User's Manual: Emulation*.

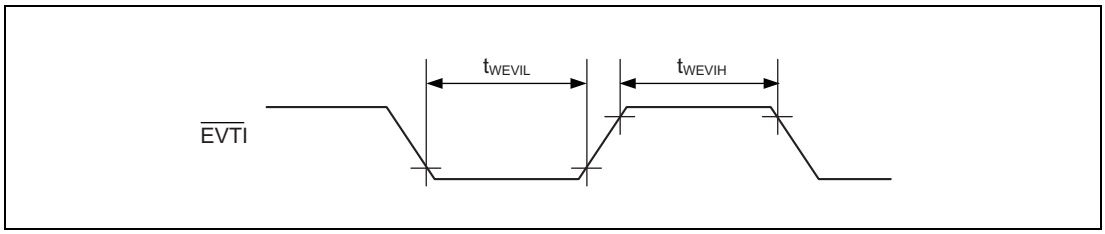


Figure 55.65 $\overline{\text{EVTI}}$ Timing

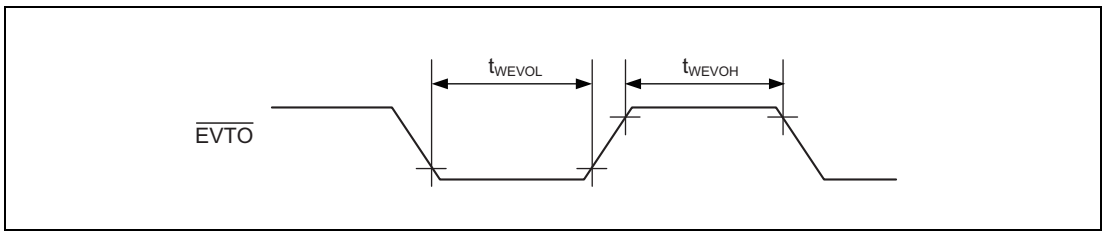


Figure 55.66 $\overline{\text{EVTO}}$ Timing

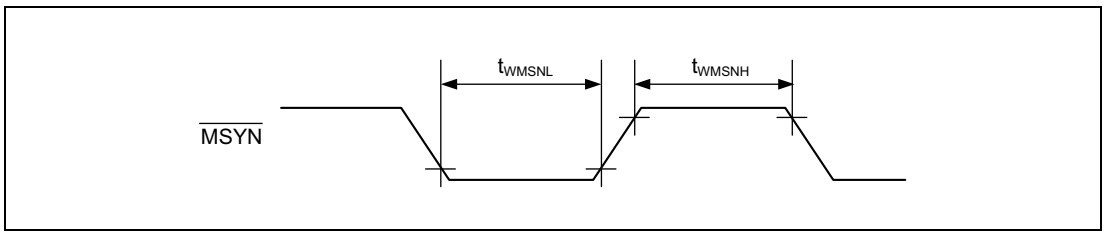


Figure 55.67 $\overline{\text{MSYN}}$ Timing

55.3.32 Debug Interface Mode Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.90 Debug Interface Mode Timing

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
TCK (JP0_2) input timing before $\overline{\text{TRST}}$	$t_{DBGIFSWCK}$		$10\times t_{LPDCKW}$ $10\times t_{TCKW}$			ns
TDI (JP0_0) setup time	$t_{DBGIFSWS}$		$10\times t_{LPDCKW}$ $10\times t_{TCKW}$			ns
TDI (JP0_0) hold time	$t_{DBGIFSWH}$		2			µs

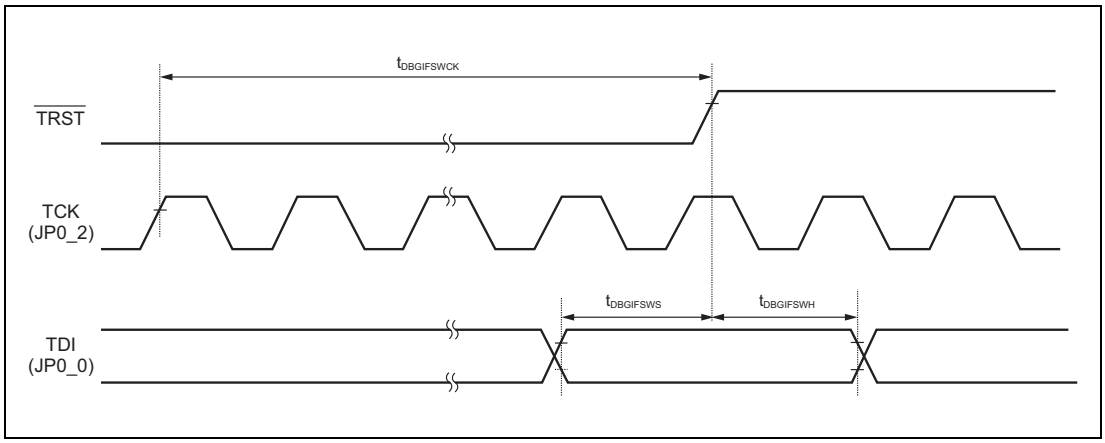


Figure 55.68 Debug Interface Mode Timing

55.3.33 Debug Wake-up Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.91 Debug wake-up timing

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
INTDCUTDI setup time	t_{IDCUS}		1			ms
INTDCUTDI hold time	t_{IDCUH}		3			ms

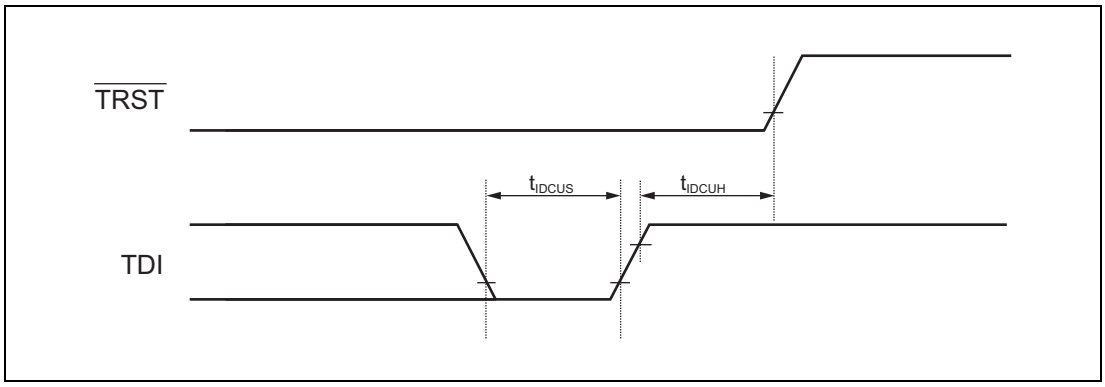


Figure 55.69 Debug wake-up timing

55.3.34 Flash Programming

55.3.34.1 Flash Programming Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.92 Flash Programming transfer rate

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Flash Programming transfer rate	r_{FP}	2-wired UART mode			2.5	Mbps
FPCK cycle time	t_{KCYSF}	3-wired Clock Sync mode	100*1			ns
FPCK high level width	t_{KWHSF}	3-wired Clock Sync mode	$t_{KCYSF}/2 - 10$			ns
FPCK low level width	t_{KWLSF}	3-wired Clock Sync mode	$t_{KCYSF}/2 - 10$			ns
FPDR setup time	t_{SSISF}	3-wired Clock Sync mode	$2 \times t_{FPcyc}$*2			ns
FPDR hold time	t_{HSISF}	3-wired Clock Sync mode	$2 \times t_{FPcyc}$*2			ns
FPDT output delay	t_{DSOSF}	3-wired Clock Sync mode	$2 \times t_{FPcyc}$*2		$3 \times t_{FPcyc}$*2 + 32	ns
FPDT hold time	t_{HSOSF}	3-wired clock sync mode	$2 \times t_{FPcyc}$*2			ns

Note 1. Input the external clock data is more than or equal to 8 clocks of CLK_HSB.
Note 2. t_{FPcyc} is a period of CLK_HSB.

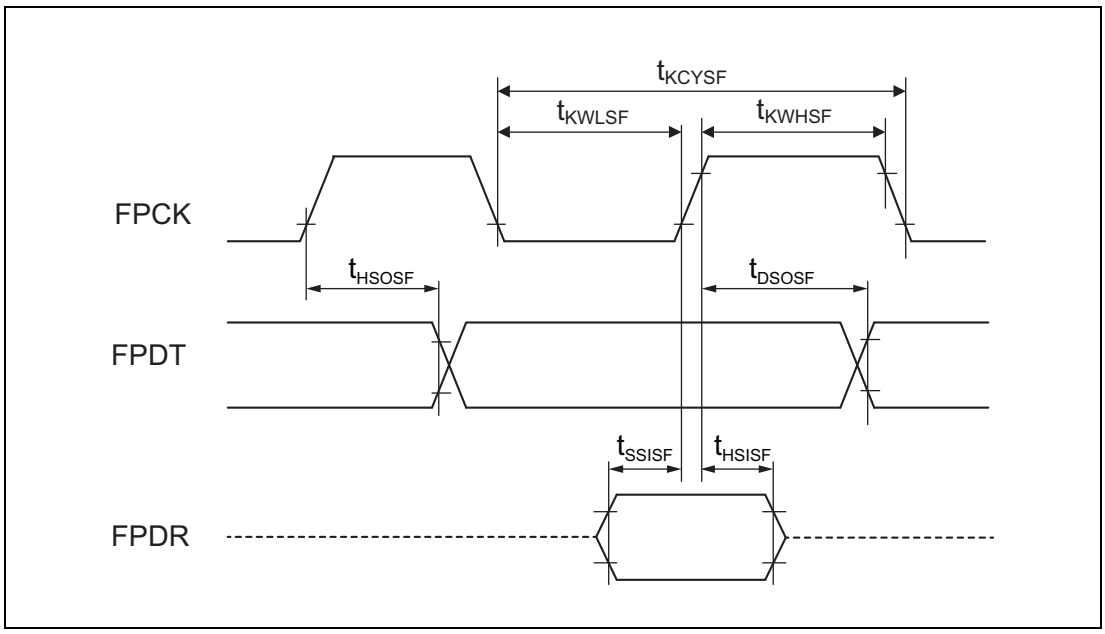


Figure 55.70 Flash Programming transfer rate

55.3.34.2 Serial Programming Setup Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.93 Serial Programming Setup Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
FLMD0 pulse input start time	t_{MD0IS}		2			ms
FLMD0 pulse input end time	t_{MD0IE}				100	ms
FLMD0 low/high level width	t_{MD0PWL}/ t_{MD0PWH}		4			µs
FLMD0 rise time	t_{MD0R}				20	ns
FLMD0 fall time	t_{MD0F}				20	ns

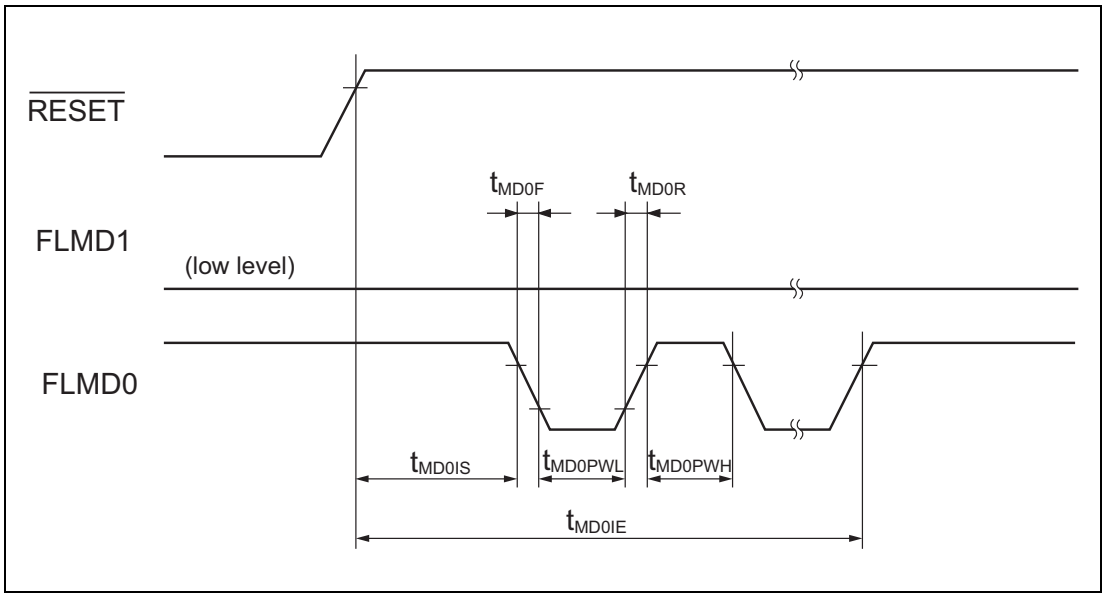


Figure 55.71 Flash Programming transfer rate

NOTE

For FLMD0 pulses, see **Section 51.7.3, Selection of Flash Programming Interface**.

55.4 A/D Converter Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.94 ADC Characteristics (1/2)

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Resolution	RESn		12		12	bit
Analog supply voltages	AnVCC		3.0		5.5	V
Reference voltages	AnVREFH		AnVCC–1.0 (Do not set to below 3.0 V) [For U2A-EVA, U2A16 and U2A8 Only]		AnVCC	V
			AnVCC [For U2A6 Only]			
Analog input voltage	V_{IAN}	ADCJnIm (T&H not used)	AnVSS		AnVREFH	V
		ADCJnIm (T&H used)	0.2		AnVREFH -0.2	V
		ADCJ0ImS (A0VREFH ≥ E2VCC)	A0VSS		E2VCC	V
		ADCJ1ImS (A1VREFH ≥ E1VCC)	A1VSS		E1VCC	V
		ADCJ2ImS (A2VREFH ≥ E0VCC)	A2VSS		E0VCC	V
Operation frequency	f_{ADCLK}	CLKA_ADC	10*3		40	MHz
		CLK_ADC	20		40	MHz
Conversion time	t_{CONV}	t_{SPL} + t_{SAR}*4	1.0			µs
Sampling time	t_{SPL}*4		0.45			µs
T&H sampling time	t_{THSMP}	In self-diagnosis	0.45			µs
		First conversion (including after self-diagnosis)	10			µs
		Other than above	0.45			µs
T&H hold time	t_{THHOLD}				10	µs
Slope of analog input voltage	t_{VSIAN}	T&H used	–5		5	kV/s
Total error	TOE	ADCJnIm (T&H not used)	–4.0		4.0	LSB
		ADCJnIm (T&H used), 3.0 V ≤ AnVREFH < 4.5 V	–8.0		8.0	LSB
		ADCJnIm (T&H used), 4.5 V ≤ AnVREFH < 5.5 V	–6.0		6.0	LSB
		ADCJnImS	–8.0		8.0	LSB
Integral non-linearity error*1	INL	ADCJnIm (T&H not used)	–2.0		2.0	LSB
		ADCJnIm (T&H used)	–3.0		3.0	LSB
		ADCJnImS	–6.0		6.0	LSB
Differential non-linearity error*1	DNL	ADCJnIm (T&H not used)	–1.0		2.0	LSB
		ADCJnIm (T&H used)	–1.0		2.0	LSB
		ADCJnImS	–1.0		4.0	LSB
Offset error*1 (Zero scale error)	OSE	ADCJnIm (T&H not used)	–3.5		3.5	LSB
		ADCJnIm (T&H used), 3.0 V ≤ AnVREFH < 4.5 V	–7.5		7.5	LSB
		ADCJnIm (T&H used), 4.5 V ≤ AnVREFH < 5.5 V	–5.5		5.5	LSB
		ADCJnImS	–7.5		7.5	LSB

Table 55.94 ADC Characteristics (2/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Full-scale error*1	FSE	ADCJnIm (T&H not used)		–3.5		3.5	LSB
		ADCJnIm (T&H used), 3.0 V ≤ AnVREFH < 4.5 V		–7.5		7.5	LSB
		ADCJnIm (T&H used), 4.5 V ≤ AnVREFH < 5.5 V		–5.5		5.5	LSB
		ADCJnImS		–7.5		7.5	LSB
Pin self diagnosis		for AnVCC/2		–60.0		60.0	LSB
		Other than above		–40.0		40.0	LSB
AD core self-diagnosis Function				–8.0		8.0	LSB
Secondary power supply voltage monitor absolute error*2		for E0VCC, VCC		–16		16	LSB
		for AWOVDD, ISOVDD		–10		10	LSB
Pull-up resistor for wiring break detection	R_{PU}	ADCJnIm pins	VIAN = AnVSS	10		34	kΩ
			VIAN = AnVCC/2	5		22	kΩ
			VIAN = AnVCC	3		16	kΩ
		ADCJnImS pins	VIAN = AnVSS	10		34	kΩ
			VIAN = EnVCC/2	5		22	kΩ
			VIAN = EnVCC	3		16	kΩ
Pull-down resistor for wiring break detection	R_{PD}	ADCJnIm pins	VIAN = AnVSS	1.5		10	kΩ
			VIAN = AnVCC/2	5		19	kΩ
			VIAN = AnVCC	10		34	kΩ
		ADCJnImS pins	VIAN = AnVSS	1.5		10	kΩ
			VIAN = EnVCC/2	5		19	kΩ
			VIAN = EnVCC	10		34	kΩ

Note 1. Quantization error is not included.

Note 2. Error of only voltage monitor.

Note 3. Include oscillation accuracy of HS IntOSC.

Note 4. For details of t_{SPL} and t_{SAR}, see **Section 43.4.24, Analog Input Sampling and Scan Group Processing Time**.

CAUTION

An analog input pin can also be used as a digital general-purpose input or output pin.

Changes in a digital input or output at the analog input pin during an AD conversion may reduce the precision of conversion.

The coupling noise from the changes in digital inputs or outputs of pins near the analog input pin during an AD conversion may also reduce the precision of conversion.

The power supply noise from the changes in digital outputs of pins of the same power supply as the analog input or the ADC that are performing AD conversion may also reduce the precision of conversion.

55.4.1 Sampling Errors in the External Circuit of the A/D Converter

Sampling error is error to which “Errors (Sampling error 1) which depend on input leakage current of analog pin” and “Errors (Sampling error 2) which depend on conversion cycles with charge sharing” were added.

$$Sampling\ error\ =\ Sampling\ error\ 1\ +\ Sampling\ error\ 2$$

The external circuits of the A/D pins which become the factor of sampling error (sampling error 1 and sampling error 2) are shown below.

(a) Errors (Sampling error 1) which depend on input leakage current of analog pin

The error depends on the input leakage current (ILeak) of analog pin and external resistance (Re), and occurs.

The error which depends on the input leakage current is given by the formula of the following.

$$Sampling\ error\ 1\ (LSB)\ =\ Re \times ILeak \times \frac{4096}{V_{avrefh}}$$

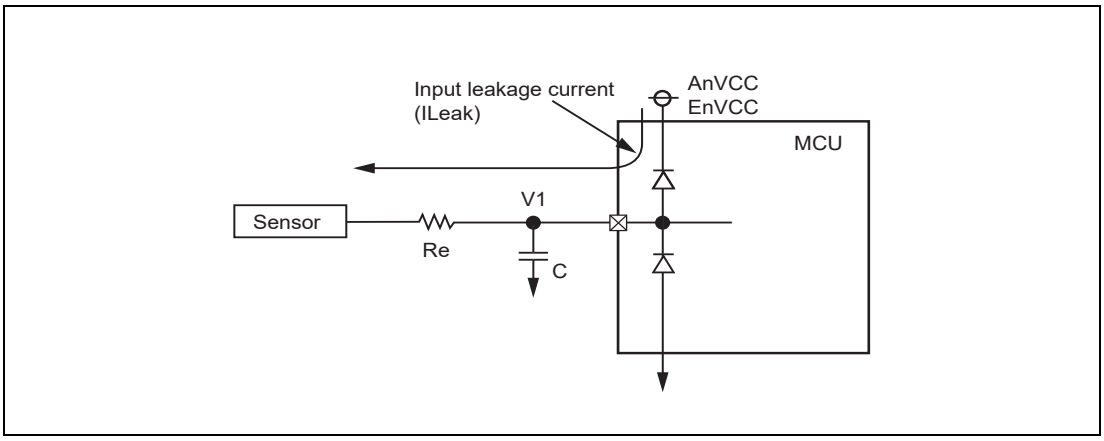


Figure 55.72 Errors (Sampling error 1) which Depend on Input Leakage Current of Analog Pin

(b) Errors (Sampling error 2) which depend on conversion cycles with charge sharing

A formula for errors in sampled values due to the external circuit of the A/D converter is given below. These errors will depend on the input circuit and conversion cycle. The formula given below for the errors is simplified for the calculation of sampling error based on internal stray capacitance, amplifier offset, resistance of the signal source, and conversion cycle. This formula can also be used to calculate the effects of the signal source resistance and conversion cycle on these errors.

The formula gives the error of analog input 2 as shown in the figure below when A/D conversion is performed in the order of analog input 1 then 2.

$$Sampling\ error\ 2\ (LSB) = \left[\left(\frac{|V2 - V1| \times CIN1}{Ce + CIN1} + \frac{|V_{vfaerr}| \times CIN2}{Ce + CIN2}\right) \times \frac{1}{1 - e^{(-T1)/(Re \times Ce)}} + \left(\frac{1}{T1} \times C1 \times V3 \times Re\right)\right] \times \frac{4096}{V_{avrefh}}$$

Table 55.95 Definition of the symbols for the Sampling Error Formula

Item	Symbol	Condition	Reference	Unit
Common capacitance of the final stage of channel multiplexer	CIN1	ADCJ0Im	1.5	pF
		ADCJ0ImS	5.2	pF
		ADCJ1Im	1.5	pF
		ADCJ1ImS	6.6	pF
		ADCJ2Im	1.5	pF
		ADCJ2ImS	5.2	pF
Common capacitance of the final stage of the amplifier and T&H control circuit	CIN2	ADCJ0Im, ADCJ0ImS	9.1	pF
		ADCJ1Im, ADCJ1ImS	9.1	pF
		ADCJ2Im, ADCJ2ImS	8.0	pF
External capacitor on analog input pin	Ce		Depends on customer's environment	µF
Signal source impedance	Re			kΩ
Conversions cycle of analog Input pins	T1			ms
AnVREFH voltage	Vavrefh			V
Potential difference between V1 and V2	V2-V1			V
Offset voltage of the amplifier and T&H control circuit	Vvfaerr		50	mV
Parasitic capacitance in the channel multiplexer	C1	ADCJnIm	2	pF
		ADCJnImS	4	
|AnVCC voltage / 2.5 - measured pin voltage (V2)|	V3		Depends on customer's environment	V

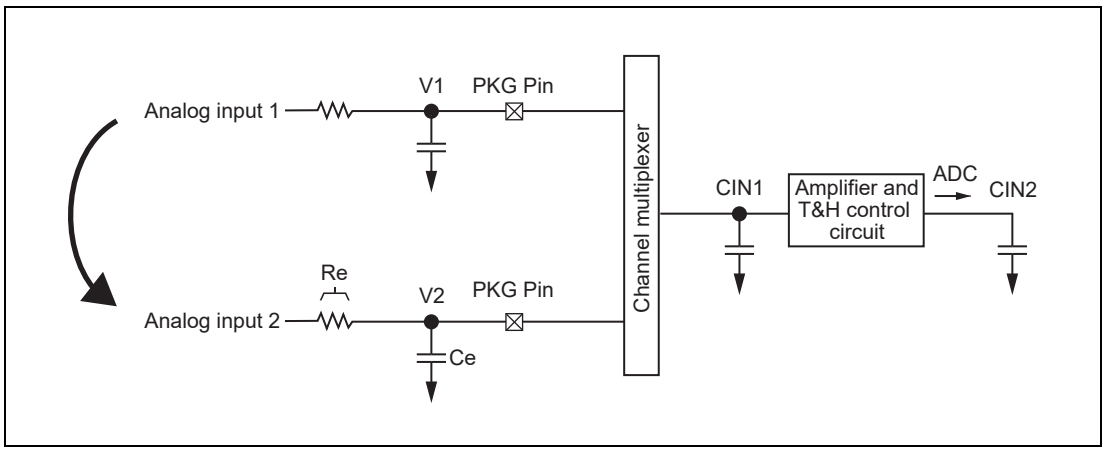


Figure 55.73 Schematic for Sampling Error 2 Formula

Values for conversion error calculated by using this formula do not include error (absolute error, etc.) specified in the A/D converter characteristics.

This formula is a desktop formula and is theoretical. When the signal source has an extremely high resistance or when the conversion cycle is too short, calculated and measured values may differ. Actual error depends on the external capacitor, external resistor, capacitance and resistance of board wiring, so evaluate and verify the error on the user board is no greater than the value produced by this formula (Condition of this formula is "Re < 1.5 MΩ and T1 ≥ 10 µs", or "1.5 MΩ ≤ Re ≤ 2 MΩ and T1 ≥ 512 µs").

55.5 Code Flash Characteristics

The code flash memory is shipped in the erased state. If the code flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Table 55.96 Code Flash Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{CLKFC} *1		8		40	MHz
Number of rewrites*2	CWRT	Data retention of 20 years*3	1000			times

Note 1. f_{CLKFC} is the frequency of CLK_LSB.

Note 2. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n”, the device can be erased “n” times for each block. For example, when a block of 64 KB is erased after 512 bytes of writing have been performed for different addresses 128 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 3. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the code flash memory.

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Only the processing time of the hardware. The overhead required by the software is not included.

Table 55.97 Code Flash Programming Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming time		512 B	CWRT < 100 times		0.31	2.0		0.25	1.8		0.24	1.7	ms
			CWRT ≥ 100 times		0.39	2.4		0.3	2.2		0.29	2.1	ms
		8 KB	CWRT < 100 times		5.0	17		4.0	15		3.8	14	ms
			CWRT ≥ 100 times		6.0	20		4.8	18		4.6	17	ms
		1 MB	CWRT < 100 times		0.64	2.2		0.5	1.9		0.47	1.8	s
			CWRT ≥ 100 times		0.77	2.7		0.6	2.3		0.57	2.2	s
Erasure time*1		16 KB	CWRT < 100 times		26	108		24	98		24	96	ms
			CWRT ≥ 100 times		32	130		29	118		29	116	ms
		64 KB	CWRT < 100 times		88	374		81	340		78	330	ms
			CWRT ≥ 100 times		106	449		98	408		94	396	ms
		1 MB	CWRT < 100 times		1.4	6.0		1.3	5.4		1.25	5.3	s
			CWRT ≥ 100 times		1.7	7.2		1.6	6.5		1.5	6.4	s

Note 1. When erase counter function is used, add the erase counter update time. For details, see **Table 55.101** and **Table 55.102**.

Table 55.98 Code Flash Programming/Erasure suspend latency Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming Suspend Latency	tSPD					160			120			120	µs
Erasure Suspend Latency	tSESD1		Priority on suspension The 1st suspend for the same erasing pulse			150			120			120	µs
	tSESD2		Priority on suspension The 2nd suspend for the same erasing pulse			1.7			1.7			1.7	ms
	tSEED		Priority on erasure			1.7			1.7			1.7	ms

Table 55.99 Code Flash Programming/Erasure resume latency Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming Resume Latency	tRPT					80			50			50	µs
Erasure Resume Latency	tREST1		Priority on suspension Resume after the 1st suspend for the same erasing pulse			1.7			1.7			1.7	ms
	tREST2		Priority on suspension Resume after the 2nd suspend for the same erasing pulse			110			80			80	µs
	tREET		Priority on erasure			110			80			80	µs

Table 55.100 Code Flash Forced Stop command latency

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Forced Stop command Latency	tFD					28			20			20	µs

Table 55.101 Erase Counter update time Characteristics [For U2A-EVA]

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Erase Counter update time					31.64	448.72		29.1	430.08		28.75	422.88	ms

Table 55.102 Erase Counter update time Characteristics [For U2A16/U2A8/U2A6]

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFC} < 20 MHz			20MHz ≤ f_{CLKFC} < 40 MHz			f_{CLKFC} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Erase Counter update time					16.64	193.72		15.0	181.08		14.65	176.88	ms

55.6 Data Flash Characteristics

The data flash memory is shipped in the erased state. If the data flash memory is read where it has not been written after erasure (no write condition), an ECC error is generated, resulting in the occurrence of an exception.

Table 55.103 Data Flash Basic Characteristics

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operation frequency	f_{CLKFD} *1		8		40	MHz
Number of rewrites*2	DWRT	Data retention 20 years*3	125k			Times
		Data retention 3 years*3	250k			Times

Note 1. f_{CLKFD} is the frequency of CLK_LSB.

Note 2. The number of rewrites is the number of erasures for each block. When the number of rewrites is “n”, the device can be erased “n” times for each block. For example, when a block of 4096 bytes is erased after 4 bytes of writing have been performed for different addresses 1024 times, the number of rewrites is counted as 1. However, multiple writing to the same address is not possible with 1 erasure (overwriting prohibited).

Note 3. Retention period under average Ta = 85°C. This is the period starting on completion of a successful erasure of the data flash memory.

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Only the processing time of the hardware. The overhead required by the software is not included.

Table 55.104 Data Flash Programming Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFD} < 20 MHz			20MHz ≤ f_{CLKFD} < 40 MHz			f_{CLKFD} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming time		4 B			0.12	0.82		0.09	0.70		0.08	0.66	ms
		8 B			0.13	0.88		0.09	0.73		0.08	0.67	ms
		16 B			0.14	0.97		0.09	0.76		0.08	0.70	ms
		32 B			0.14	0.97		0.09	0.76		0.08	0.70	ms
		64 B			0.14	0.97		0.10	0.76		0.09	0.70	ms
		128 B			0.17	1.20		0.13	0.95		0.11	0.89	ms
		2 KB			2.72	9.64		2.08	7.56		1.76	6.92	ms
		128 KB			0.174	0.62		0.13	0.48		0.11	0.44	s
Property Programming time		32 B			0.41	2.64		0.29	2.18		0.25	2.03	ms
Switch Programming time		32 B			0.26	1.79		0.18	1.46		0.16	1.36	ms
TAG Update time					0.30	1.70		0.21	1.43		0.18	1.34	ms
Erasure time		2 KB *1			11.1	105		9.9	95		9.6	92	ms
		4 KB			19.2	179		17	160		16.4	155	ms
		128 KB			0.62	5.73		0.55	5.12		0.53	4.96	s
Blank check command time		4 B				5.5			2.3			1.3	µs
		4 KB				1.8			0.8			0.5	ms
Property Erasure time		2 KB			11.1	105		9.9	95		9.6	92	ms
Switch Erasure time		2 KB			11.4	106.7		10.1	96.4		9.8	93.4	ms
TAG Erasure time		2 KB			11.4	106.7		10.1	96.4		9.8	93.4	ms

Note 1. Extended Data Area Only

Table 55.105 Data Flash Programming/Erasure suspend latency Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFD} < 20 MHz			20MHz ≤ f_{CLKFD} < 40 MHz			f_{CLKFD} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming Suspend Latency*1	tSPD					160			120			120	µs
Erasure Suspend Latency*1	tSESD1		Priority on suspension The 1st suspend for the same erasing pulse			150			120			120	µs
	tSESD2		Priority on suspension The 2nd suspend for the same erasing pulse			300			300			300	µs
	tSEED		Priority on erasure			300			300			300	µs

Note 1. Data Area and Extended Data Area Only

Table 55.106 Data Flash Programming/Erasure resume latency Characteristics

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFD} < 20 MHz			20MHz ≤ f_{CLKFD} < 40 MHz			f_{CLKFD} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Programming Resume Latency*1	tRPT					80			50			50	µs
Erasure Resume Latency*1	tREST1		Priority on suspension Resume after the 1st suspend for the same erasing pulse			300			300			300	µs
	tREST2		Priority on suspension Resume after the 2nd suspend for the same erasing pulse			100			70			70	µs
	tREET		Priority on erasure			100			70			70	µs

Note 1. Data Area and Extended Data Area Only

Table 55.107 Forced Stop command latency

Item	Symbol	Block size	Condition	8MHz ≤ f_{CLKFD} < 20 MHz			20MHz ≤ f_{CLKFD} < 40 MHz			f_{CLKFD} = 40 MHz			Unit
				MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Forced Stop command Latency	tFD					28			20			20	µs

55.7 Temperature Sensor Characteristics

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.

Table 55.108 Temperature Sensor Characteristics*1*2*3

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Temperature accuracy	A_{CCTS1}	– 40 °C ≤ Tj ≤ + 140 °C	–4		4	°C
	A_{CCTS2}	Tj > 140 °C	–2		2	°C
Temperature update period	t_{TSUP}		10			ms
Operation stabilization waiting time	t_{TSSB}				200	µs

Note 1. The temperature borders need to be set keeping the temperature range (Tj(min) to Tj(max)) including the temperature sensor accuracy.

Note 2. It does not include accuracy of measuring equipment.

Note 3. Temperature sensor cannot detect local heat generation inside the chip.

55.8 Thermal Characteristics

55.8.1 Thermal Characteristics Parameter

Table 55.109 Thermal Characteristics*1

Parameter	Estimated value										Unit	Remark
	U2A-EVA	U2A16			U2A8		U2A6					
	BGA516	BGA516	BGA373	BGA292	BGA373	BGA292	BGA292	QFP176	BGA156	QFP144		
θja	10.8	13.4	13.7	13.9	15.9	16.3	17.9	12.4	18.7	12.6	°C/W	JESD51-9 compliant (4 layers)
θjb	6.0	8.8	8.5	8.0	10.6	10.3	12.1	6.0	10.9	5.0	°C/W	JESD51-9 compliant (4 layers)
θjc	3.7	5.7	5.8	5.9	7.8	8.2	9.3	11.9	11.4	11.6	°C/W	JESD51-9 compliant (4 layers)
θjcbot	3.4	5.3	5.4	5.3	7.2	7.3	8.7	1.1	8.3	0.7	°C/W	JESD51-9 compliant (4 layers)
Ψjb	5.7	8.4	8.1	7.7	10.2	10.0	11.8	5.8	10.6	4.9	°C/W	JESD51-9 compliant (4 layers)
Ψjmb*2	3.1	4.8	4.6	4.9	6.3	6.5	7.9	0.9	7.8	0.6	°C/W	JESD51-9 compliant (4 layers)
Ψjt	0.2	0.2	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	°C/W	JESD51-9 compliant (4 layers)
4Lθja	12.7	15.8	16.1	16.7	18.5	19.3	21.0	14.3	23.1	15.1	°C/W	L board (4 layers)
4Lθjb	7.3	10.5	10.0	9.6	12.3	12.0	14.0	8.0	12.7	6.9	°C/W	L board (4 layers)
4LΨjb	6.7	9.9	9.4	9.1	11.7	11.5	13.4	7.9	12.3	6.7	°C/W	L board (4 layers)
4LΨjmb*2	3.1	4.7	4.4	4.8	6.0	6.3	7.7	0.9	8.0	0.5	°C/W	L board (4 layers)
4LΨjt	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.4	0.3	°C/W	L board (4 layers)
4LTb_inc	6.2	5.9	6.9	7.6	6.9	7.9	7.8	6.4	10.9	8.3	°C/W	L board (4 layers)

Note 1. The thermal characterization parameters depends on the usage environment.

Note 2. Ψjmb shows thermal characterization parameter from junction to board surface (center of the PKG on layer-1; Tmb).

Ψjmb = (Tj-Tmb)/Pd (Pd: power consumption of the chip)

55.8.2 Assumed Board

Table 55.110 JESD51-9 Compliant Board (4 layers)

	Board Size (mm)		
	X	Y	Area (mm^2)
Board Size	101.5	114.5	11621.75
Remaining copper rate		Conductor thickness	
50 – 95 – 95 – 50%		70 – 35 – 35 – 70 µm	

Table 55.111 L board (4 layers)

	Board Size (mm)		
	X	Y	Area (mm^2)
Board Size	90	160	14400
Remaining copper rate		Conductor thickness	
30 – 80 – 80 – 30%		35 – 35 – 35 – 35 µm	

55.9 BSCAN Timing

Conditions:

- See **Section 55.3.1, AC Characteristic Measurement Condition**.
- Buffer type = TTL

Table 55.112 BSCAN Timing

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
TCK cycle width	t_{DCKW}		100			ns
TDI setup time	t_{SDI}		12			ns
TDI hold time	t_{HDI}		3			ns
TMS setup time	t_{SMS}		12			ns
TMS hold time	t_{HMS}		3			ns
TDO delay time	t_{DDO}		0		30	ns

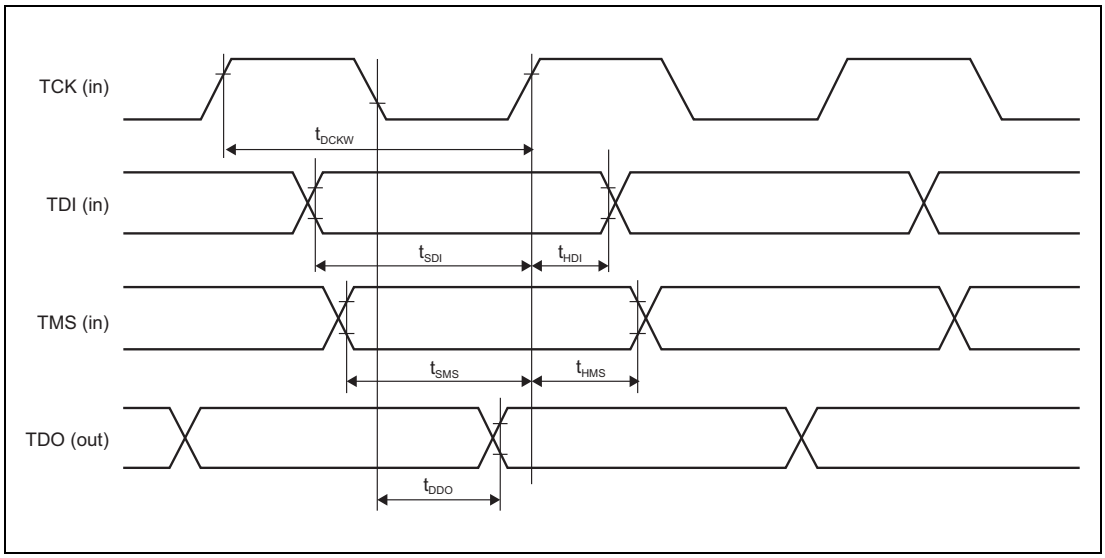


Figure 55.74 BSCAN Timing

Section 90 Long-Term System Counter (LTSC)

This section contains a generic description of the Long-Term System Counter (LTSC).
The first part in this section describes all RH850/U2A-EVA specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the LTSC.

90.1 Features LTSC for RH850/U2A-EVA

Long-Term System Counter (LTSC) is 64-bit counter that operates by long period. It enables the generation of a unique timestamp without overflow.

90.1.1 Number of Units and Channels

This microcontroller has the following number of LTSC units.

Table 90.1 Number of Units

Product Name	RH850/U2A-EVA (516 pins)	RH850/U2A16 (516 pins)	RH850/U2A16 (373 pins)	RH850/U2A16 (292 pins)	RH850/U2A8 (373 pins)	RH850/U2A8 (292 pins)	RH850/U2A6 (292 pins)	RH850/U2A6 (176 pins)	RH850/U2A6 (156 pins)	RH850/U2A6 (144 pins)
Number of Units	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)	1 (n = 0)
Name	LTSCn									

LTSC is connected to PCLK. Neither compare nor interrupt functions.

Table 90.2 Index

Index	Description
n	Throughout this section, the individual LTSC units are identified by the index “n” (n = 0).

90.1.2 Register Base Address

LTSCn base addresses are listed in the following table.
LTSCn register addresses are given as offsets from the base addresses.

Table 90.3 Register Base Address

Base Address Name	Base Address	Bus Group
<LTSC0_base>	FFC7 8100_H	P-Bus Group 5

90.1.3 Clock Supply

Clock supply by and to LTSCn is listed in the following table.

Table 90.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
LTSCn	PCLK	CLK_HSB

For details of the clock, see **Section 13, Clock Controller**.

90.1.4 Interrupt Request and Error Notifications

LTSC does not support the interrupt request and error notifications.

90.1.5 Reset Sources

LTSCn reset sources are listed in the following table. LTSCn is initialized by these reset source.

Table 90.5 Reset Sources

Unit Name	Register Name	Reset condition						
		Power On Reset	System Reset 1	System Reset 2	Application Reset	DeepSTOP Reset	Module Reset	JTAG Reset
LTSCn	LTSCnRMSK	√	√	√	—	√	—	—
	Other Registers	√	√	√	√*1	√	—	—

Note 1. Reset for “Other Registers” is masked when LTSCnRM bit in LTSCnRMSK register is set to 1.

90.1.6 External Input/Output Pins

LTSC has no external pins

90.2 Overview

90.2.1 Functional overview

- 1×64-bit counter
- Free-run up counting
- Atomic read/write access to all registers
- No compare functions
- Clock sources: Peripheral Clock PCLK
- No interrupt functions
- No functions to start/stop multiple counter channels synchronously (Only 1channel)
- Anytime read access to counter registers
- Application reset (SW reset) can be masked. When masked, counter keeps running on reset occurrence and counter register will not be initialized.
- Stop control by Supervisor

90.2.2 Counter channels

Long-Term System Counter is composed of one 64-bit up-counter channel. This counter is not equipped with compare registers.

The counter register can be read at any time.

The CPU architecture supports 64-bit write access on the 32-bit peripheral bus. However, in case of individual 32-bit write accesses to 64-bit counter, the updated register value will be processed after write to the upper 32-bit register. The appropriate write sequence must therefore be specified by software (i.e., that the register must be accessed to the lower 32 bits first, and then to the upper 32 bits.)

In case of reading the 64-bit counter value while the counter is running, the current 64-bit counter value will be captured first and read access will be performed on the captured value then. This is to avoid that the upper or lower 32-bit value changes between the two consecutive 32-bit read accesses.

90.2.3 Block Diagram

The following block diagram shows the main components of the Long-Term System Counter.

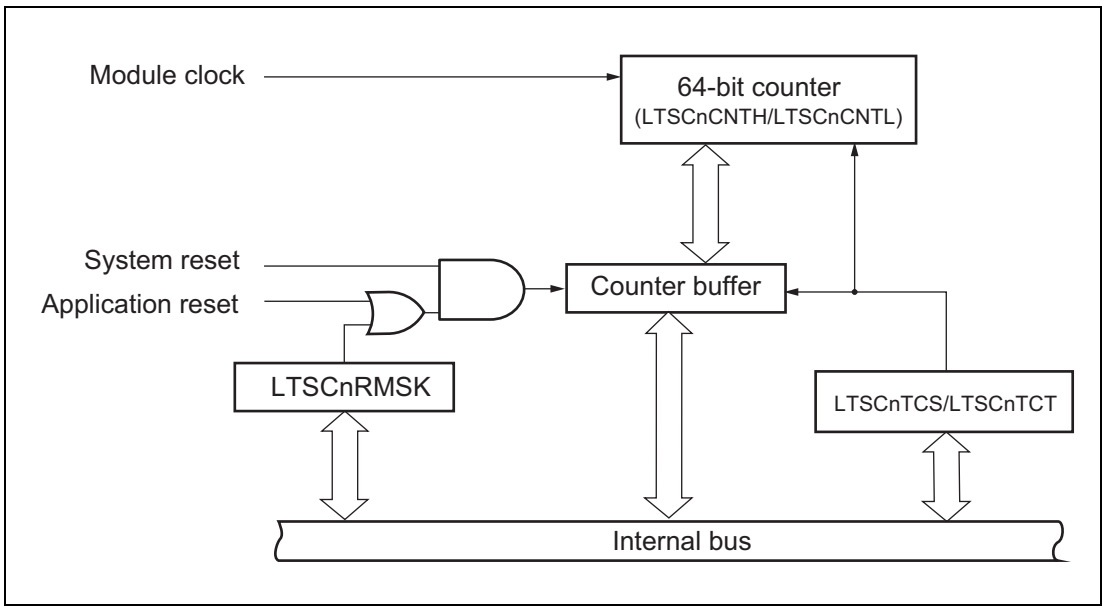


Figure 90.1 LTSC Block Diagram

90.3 Registers

This section contains a description of all registers of Long-Term System Counter.

90.3.1 List of Registers

LTSC registers are listed in the following table.

For details about <LTSCn_base>, see **Section 90.1.2, Register Base Address**.

Table 90.6 List of Registers

Module Name	Register Name	Symbol	Address	Access size	Access Protection	
					PBG	Other
LTSCn	LTSC Timer Counter Start Register	LTSCnTCS	<LTSCn_base>+0010$_H$	32/8	PBG50#6	—
	LTSC Timer Counter Stop Register	LTSCnTCT	<LTSCn_base>+0014$_H$	32/8	PBG50#6	—
	LTSC Timer Counter Status Register	LTSCnCSTR	<LTSCn_base>+0018$_H$	32/8	PBG50#6	—
	LTSC Timer SW Reset Mask Register	LTSCnRMSK	<LTSCn_base>+0034$_H$	32/8	PBG50#6	—
	LTSC Timer Counter Register L	LTSCnCNTL	<LTSCn_base>+0040$_H$	32	PBG50#6	—
	LTSC Timer Counter Register H	LTSCnCNTH	<LTSCn_base>+0044$_H$	32	PBG50#6	—

90.3.2 LTSCnTCS — LTSC timer counter start register

Access: This register is a write-only register that can be written in 32- or 8-bit units.

Address: <LTSCn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTSCnTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 90.7 LTSCnTCS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	LTSCnTS	This bit starts the counter channel. 0: No function 1: Starts the counter and sets LTSCnCSTR.LTSCnCST = 1. Setting this bit is ignored as long as LTSCnCSTR.LTSCnCST = 1.

NOTES

1. A reset from any source will initialize the bit unless the masking function has been selected in the LTSC SW reset mask register (LTSCnRMSK). If masking function is selected, the function related to register bits LTSCnTS is not affected by SW reset (Application Reset). Refer to **Section 90.4.4, Resets**.
2. LTSCnTCS is always read as $0000\ 0000_H$.

90.3.3 LTSCnTCT — LTSC timer counter stop register

Access: This register is a write-only register that can be written in 32- or 8-bit units.

Address: <LTSCn_base> + 0014_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTSCnTT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 90.8 LTSCnTCT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	LTSCnTT	This bit stops the counter. 0: No function 1: Stops the counter and clears LTSCnCSTR.LTSCnCST bit. Setting this bit is ignored as long as LTSCnCSTR.LTSCnCST = 0.

NOTES

1. A reset from any source will initialize the bit unless the masking function has been selected in the LTSC SW reset mask register (LTSCnRMSK). If masking function is selected, the function related to register bits LTSCnTT is not affected by SW reset (Application Reset). Refer to **Section 90.4.4, Resets**.
2. LTSCnTCT is always read as $0000\ 0000_H$.

90.3.4 LTSCnCSTR — LTSC timer counter status register

Access: This register is a read-only register that can be read in 32- or 8-bit units.

Address: <LTSCn_base> + 0018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTSCn CST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 90.9 LTSCnCSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	LTSCnCST	This bit indicates whether the counter is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to LTSCnTCS.LTSCnTS being set to 1. Setting LTSCnTCT.LTSCnTT to 1 re-sets this bit to 0.

NOTE

A reset from any source will initialize the bit unless the masking function has been selected in the LTSC SW reset mask register (LTSCnRMSK). If masking function is selected, the function related to register bits LTSCnCST is not affected by SW reset (Application Reset). Refer to **Section 90.4.4, Resets**.

90.3.5 LTSCnRMSK — LTSC timer SW reset mask register

Access: This register can be read or written in 32- or 8-bit units.

Address: <LTSCn_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LTSCnRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 90.10 LTSCnRMSK Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LTSCnRM	This bit masks the SW reset (Application Reset). When SW reset is masked, it behaves as follows: • The counter continues counting up operation. SW reset occurrence has no impact on the counter operation and the counter value. • The count enable status will not be changed by SW reset; no stop trigger will be generated by occurrence of SW reset. All other LTSC registers will be reset independent whether masking was selected or not. 0: Reset mask disabled (LTSC responds to SW reset (Application Reset)) 1: Reset mask enabled (LTSC does not responds to SW reset (Application Reset))

NOTE

This register will be initialized by only System reset, while it will not be initialized by SW reset (Application Reset) even though LTSCnRM bit is set to 0. Refer to **Section 90.4.4, Resets**.

90.3.6 LTSCnCNTL — LTSC timer counter register low (lower 32-bit)

Access: This register can be read or written in 32-bit units.

Address: <LTSCn_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTSCnCNTL[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTSCnCNTL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 90.11 LTSCnCNTL Register Contents

Bit Position	Bit Name	Function
31 to 0	LTSCnCNTL [31:0]	Lower 32-bit of 64-bit counter value

NOTES

1. A reset from any source will initialize the bits unless the masking function has been selected in the LTSC SW reset mask register (LTSCnRMSK) for this channel. If masking function is selected, the register contents will not be changed by SW reset (Application Reset). Refer to **Section 90.4.4, Resets**.
2. This register is part of a 64-bit register. The architecture supports 64-bit write access to the upper and lower register. However, in case of individual 32-bit write accesses the total 64-bit register value will become effective after the higher 32-bit value has been written to. The appropriate write sequence must therefore be specified by software (i.e., that the register must be accessed from the lower 32 bits, and then the upper 32 bits.).
3. Read access to counter registers is supported both while counter is running and stopped.
   In case of reading the 64-bit counter value while the counter is running, the current 64-bit counter value will be captured on read access to the lower 32-bit value. The captured upper 32-bit value can be read afterwards. This is to avoid that the upper or lower 32-bit value changes between the two consecutive 32-bit read accesses.
4. Write access is permitted only, while counter is stopped.

90.3.7 LTSCnCNTH — LTSC timer counter register high (upper 32-bit)

Access: This register can be read or written in 32-bit units.

Address: <LTSCn_base> + 0044_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTSCnCNTH[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LTSCnCNTH[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 90.12 LTSCnCNTH Register Contents

Bit Position	Bit Name	Function
31 to 0	LTSCnCNTH [31:0]	Upper 32-bit of 64-bit counter value

NOTES

1. A reset from any source will initialize the bits unless the masking function has been selected in the LTSC SW reset mask register (LTSCnRMSK) for this channel. If masking function is selected, the register contents will not be changed by SW reset (Application Reset). Refer to **Section 90.4.4, Resets**.
2. This register is part of a 64-bit register. The architecture supports 64-bit write access to the upper and lower register. However, in case of individual 32-bit write accesses the total 64-bit register value will become effective after the higher 32-bit value has been written to. The appropriate write sequence must therefore be specified by software (i.e., that the register must be accessed from the lower 32 bits, and then the upper 32 bits.).
3. Read access to counter registers is supported both while counter is running and stopped.
   In case of reading the 64-bit counter value while the counter is running, the current 64-bit counter value will be captured on read access to the lower 32-bit value. The captured upper 32-bit value can be read afterwards. This is to avoid that the upper or lower 32-bit value changes between the two consecutive 32-bit read accesses.
4. Write access is permitted only, while counter is stopped.

90.4 Operation

LTSC is a 64-bit timer/counter that operates by PCLK based period. When the counter value reached to FFFFFFFF FFFFFFFF$_H$, it will change to 00000000 00000000$_H$ at the next cycle and it will continue to count. In case of writing to timer counter register low and high (LTSCnCNTL/H) while the counter is running, the value of these registers is not updated.

90.4.1 Start and Stop LTSC

The basic operation of start and stop for LTSC is shown as follows.

(1) Start the counter

LTSC counter starts from the current count value when LTSCnTCS.LTSCnTS bit is written to 1. When LTSCnTCS.LTSCnTS is written to 1 again while the counter is running, the counter operation is not affected.

(2) Stop the counter

LTSC counter stops when LTSCnTCT.LTSCnTT bit is written to 1. The counter value keeps the value when the counter stopped.

90.4.2 How to update the counter value

It is possible to start the counter from the unique value by setting timer counter register low and high in advance. The way to update the counter is shown as follows

1. Read LTSCnCSTR register to check the status of the counter. Write 1 to LTSCnTCT.LTSCnTT bit to stop the counter if the value of LTSCnCSTR.LTSCnCST bit is 1.
2. Write the update value to LTSCnCNTL and LTSCnCNTH register. Refer to "**Section 90.4.3, Access to register** " about the procedure of writing these registers.
3. Write 1 to LTSCnTCS.LTSCnTS to start the counter.

90.4.3 Access to register

The bit width of the counter register in LTSC is 64-bit. To those register, 64-bit access is supported and the 64-bit access will be divided into twice accesses of the 32-bit peripheral bus. In case of individual 32-bit write or read accesses to 64-bit counter registers of LTSC, the upper 32-bit register will be processed after the lower 32-bit register access. Writing access to a 64-bit register, the register value will be updated when the upper 32-bit register is processed while reading access to a 64-bit register, the value of the register at that moment will be read out.

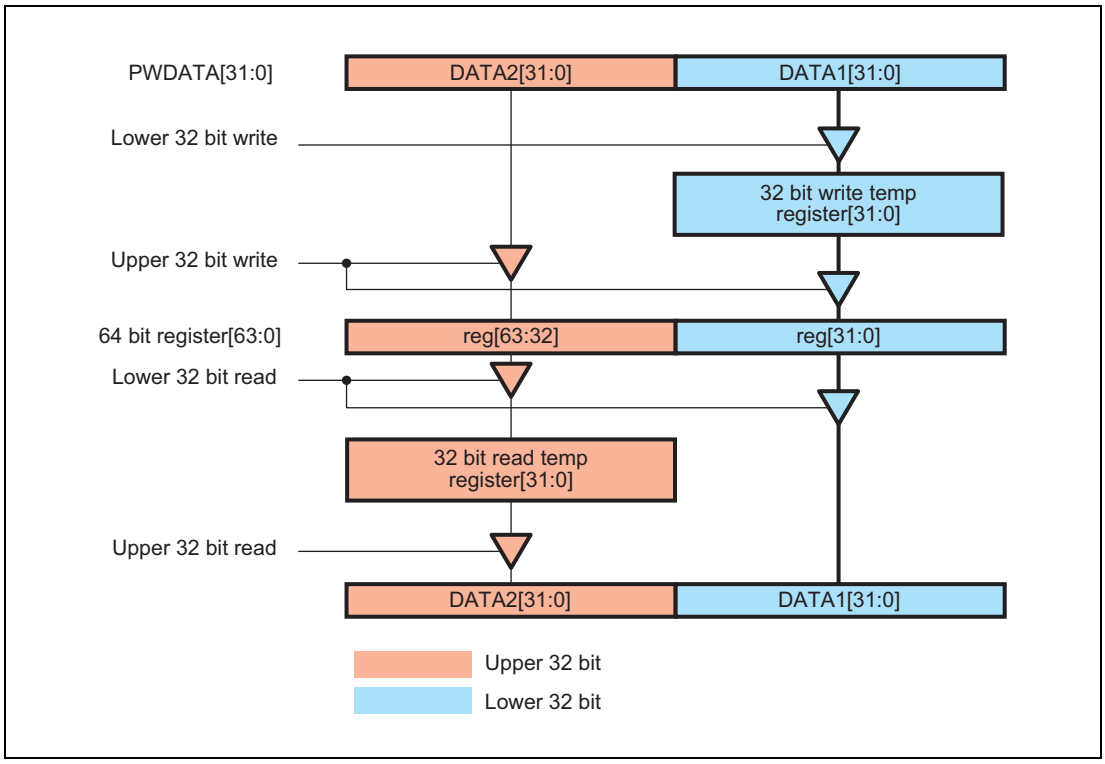


Figure 90.2 Timer counter register access sequence

90.4.4 Resets

There are Application reset and System reset for LTSC. To System reset, all of the LTSC registers will be initialized, to the Application reset, you can mask it or not. All the registers behave as shown in the below table upon the Application reset or System reset.

Table 90.13 Reset of LTSC registers

Registers or bits in LSTC		Application reset		System reset
		Masked (LTSCnRM =1)	UnMask (LTSCnRM =0)	
LTSCnTCS	LTSCnTS	KEPT	INIT	INIT
LTSCnTCT	LTSCnTT	KEPT	INIT	INIT
LTSCnCSTR	LTSCnCST	KEPT	INIT	INIT
LTSCnRMSK	LTSCnRM	KEPT	KEPT	INIT
LTSCnCNTL	LTSCnCNTL[31:0]	KEPT	INIT	INIT
LTSCnCNTH	LTSCnCNTH[31:0]	KEPT	INIT	INIT

Note: INIT: initialized
KEPT: values are kept

RH850/U2A-EVA Group User's Manual: Hardware

Publication Date: Rev.0.40 October 11, 2018
Rev.1.50 August 30, 2024

Published by: Renesas Electronics Corporation

RH850/U2A-EVA

Renesas Electronics Corporation

R01UH0864EJ0150